Microapps

Contents

Microapps

August 19, 2022

Application integrations extend Citrix Workspace and their microapps provide users with a cutting-edge experience and user interface. Deliver relevant, actionable notifications, combined with intuitive microapp workflows, to make the most important use-cases of business systems and applications directly accessible from a user's Workspace.

Save users time by reducing context switching and eliminating the need to learn how to use various applications for one-off interactions. This improves the user experience because they can focus on their primary responsibilities.

When you evaluate the Microapps service, the Citrix Service Operations team provides ongoing onboarding help. That team also communicates with you to ensure that the Microapps service is running and configured correctly. The onboarding steps are:

- Have a Citrix Cloud account and Citrix Workspace experience access.
- Have Microapps service entitlement.
- Download and use the Citrix Workspace app or use the workspace URL.
- Discuss integration requirements with Citrix.
- Review security whitepaper, the Citrix Workspace Microapps Technical security overview, and Secure Deployment Guide for Citrix Cloud Platform.
- Review the Integration Checklist to resolve security, legal, and development issues.
- Review our specifications documentation for target applications.

For a complete guide to onboarding Citrix Workspace Microapps service, see Getting started.

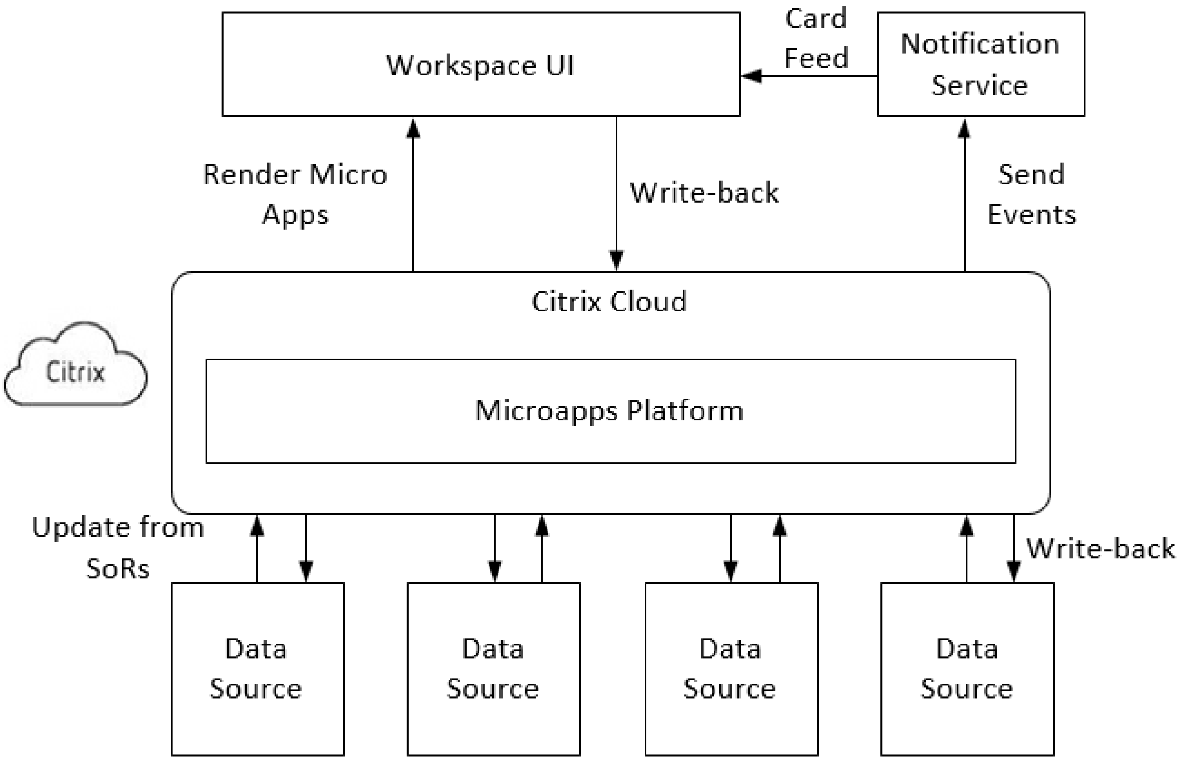
Microapps service belongs to the Intelligent Workspace features for Citrix Workspace. Intelligent Workspace features give users a single unified experience with microapps, notifications, actions, and workflows to guide and automate work. For the latest details on OS support for Microapps service, see the **Intelligent Workspace features** entry on the Citrix Workspace app feature matrix.

Overview

Citrix Workspace Microapps service is a solution focused on delivering actions and notifications from your applications right into your Workspace or other channels. You do this by building integrations from your application data sources to the Microapps service enabling you to pull actions from your applications into Workspace.

Microapps deliver actionable forms and notifications. Microapps can write back to source systems. OAuth 2.0 is the recommended authentication mechanism for writing to SaaS applications. An access token specific to each user is stored to enable a seamless user experience for user actions.

The following diagram provides a high-level overview of the integration schema:



Terminology

Citrix Microapps service is offered as part of Citrix Workspace Intelligence. To familiarize yourself with Citrix Workspace see Citrix Workspace documentation.

Citrix Workspace platform is a foundational component of Citrix Cloud that enumerates and delivers all your digital workspace resources to the Citrix Workspace user experience.

Microapps are small, task-specific applications that deliver highly targeted functionality. These apps allow users to accomplish single-purpose activities in a simple and quick manner. Microapps deliver actionable forms and notifications. Microapps can write back to source systems.

Microapps service refers to several components inside Citrix Cloud focused on delivering actions from your applications right into your Workspace or other channels. Microapps services include Microapps admin, the Microapps server, and cache.

Write-back describes how data is returned. When an action is taken, data is written back to integrated application and then resynchronized back to the cache.

System of record (SoR) is the target application that holds information of interest to Citrix Workspace users and uses JSON REST and any common authentication mechanism (OAuth 2.0, NTLM, Basic Auth, Bearer Auth).

HTTP integration is a custom built integration that you create from your application data sources to the Microapps service platform. These integrations enable you to pull data from your applications into your Workspace and performs writebacks to the target system.

Microapp builder is a no-code tool that allows developers to build event-driven microapps (event notifications) and user-initiated microapps (action pages).

Event notifications are event-driven microapps that automatically notify users when something requires their attention by creating feed notification (also known as a card) in the Workspace activity feed. For example, 'New Expense Report for Approval' and 'New Course Available for Registration'.

Action page refers to user-initiated microapps that are available as actions in Workspace and make it easy to initiate actions. For example, 'Request PTO', 'Submit a Help Desk Ticket', and 'Search the Directory'.

Channels are how microapps-related notifications are delivered, including your Notification feed in Workspace, Mobile notifications in Workspace Apps, and MS Teams events. Events are notifications pushed to your feed based on changes in data sources through the rendering of microapps pages.

Notification service enables system alerts from data sources that are pushed to Notification Feeds without a specific request from a user.

Data synchronization is pulling data from your integrated applications to the Microapps service platform so that a comparison can be made to the cache. Generally, full synchronization is performed every 24 hours and incremental syncs can be configured to pull every five minutes. Data synchronization is configured when setting up your data endpoints.

Set up integrations

Use our out-of-the-box microapps that are available with template integrations. For a full list of available template integrations and out-of-the-box microapps, see Set up integrations.

You can also create custom integrations to suit your needs. For more information, see Build a custom application integration.

Add and customize microapps

After you set up your integration, prepare your microapps and their pages and notifications.

Customize existing out-of-the-box microapps or build your own microapps to deliver the best end-user experience that meets your needs and streamlines daily workflows. For more information, see Create microapps.

Use cases

Simplify valuable workflows with Citrix Workspace, harnessing microapp technology with out-of-the-box templates available today. These use cases give employees a consistent and modern experience independent of the legacy systems they leverage, providing a simplified and effective way to perform important departmental workflows. For more information, see Optimize workflows with Citrix Workspace.

Check out these videos for demos of these workflows:

IT Self Service microapp Demo

HR Self Service microapp Demo

Other resources

Take the Citrix Workspace Microapps Essentials elearning course to learn about the fundamentals of web services, APIs, and systems integrations through building microapps designed to boost employee productivity and optimize the end user experience.

Check out this overview of Citrix Workspace Intelligence and the Microapps service at Video: Microapp Overview.

Learn about creating custom integrations and microapps at Video: Microapp Custom Integrations.

Find out more about getting a test instance at Citrix Workspace Developer Portal.

Here's a quick guide to setting up an RSS microapp: Get notifications when there is a Citrix security bulletin.

Visit the Microapps Discussions Forum.

Where to go next

For security compliance, see Secure Deployment Guide for Citrix Cloud Platform and Citrix Workspace Microapps Technical Security Overview.

To review Citrix Cloud connectivity requirements, see Internet Connectivity Requirements.

To get started with Microapps service, see Getting started.

To learn more about defining identity providers and accounts, see Identity and Access Management.

What's new

December 1, 2022

A goal of Citrix is to deliver new features and updates to Citrix Workspace customers when they are available. Workspace with intelligence releases regular updates to the Microapp platform, so check back here regularly to find out about new features and functionality.

> **Important!**
>
> We are continuously releasing new functionality and strive to announce these features when first available. This means newer features that you see at the top of this list might be in the release process and it can take a few days to be available for a particular customer.

For details about the Service Level Agreement for cloud scale and service availability, see the Citrix Cloud Service Level Agreement. See Citrix Workspace and Citrix Cloud for more information about Citrix Workspace.

December 2022

Deprecation announcement: Due to low usage, Citrix plans to deprecate the Microapps service in the Asia-Pacific (APS) geo location and migrate existing users to the US or EU geo locations, as applicable. All APS environments are scheduled for deprecation beginning December 30, 2022. Administrators of APS Microapps environments with active use are scheduled to receive an in-product message with further steps detailing the deprecation and migration process. For more information, see Deprecation

June 2022

Deprecation of Microapps service: Due to the current deprecation process of microapps applications, the Microapp service is removing unused environments from the platform from June 1, 2022 onwards. Any Microapp service environment that has no activity is scheduled for this deprovisioning. For more information, see Microapps service deprecation.

May 2022

Deprecation of Microapps integrations: Microapps integrations are deprecated as of May 16, 2022. Any deployed instance of the integration continues to function and the integration is available for creation in the catalog. The maintenance date for deprecated integrations is the deprecation date + 4 weeks. Upon maintenance, the integration will not be available in the integrations catalog. For more information, see Deprecation.

November 2021

The following announcements are a part of this week's release: MAS 1.130.0

New Employee Survey App integration template and microapps: Our new Employee Survey App integration template lets users send new survey form and manage existing survey forms from Citrix Workspace. For more information, see Integrate Employee Survey App.

New multiselect lookup page builder component Multi select lookup works in a similar manner to the Lookup component and allows users to search through, and select a large quantity of values by searching for an alternative value. For example users can search and add multiple users or user emails when scheduling a meeting, or add multiple labels when creating a Jira ticket. For more information, see Page builder.

The following announcements are part of previous releases:

October 2021

New Cherwell integration template and microapps: Our new Cherwell integration template lets Citrix Workspace users manage and receive notifications about changes to incidents and service requests. For more information, see Integrate Cherwell.

Updated RSS integration template: We've made updates to the RSS template to allow for easy customization for any channel that you want. For more information, see Customize RSS template.

September 2021

New Adobe Sign integration template and microapps: Our new Adobe Sign integration template allows Citrix Workspace users to securely view and sign agreements as well as manage templates. For more information, see Integrate Adobe Sign.

New page builder components: We've added two new components and a page template for building your microapp pages; **Embed**, **File upload**, and **Embed** page template. With **Embed** and **Embed** page template you can embed webpages in your microapps that can be displayed in Citrix Workspace. With the **File upload** component, you can upload raw files to your application System of record during submit or update actions. For more information, see Page builder.

Configure user providers: We've added a new feature for administrating your microapp subscribers and users. Configure user providers allows admins to configure user providers to collect user and group user data from an external system. You can use this data to manage microapp subscriptions in all of your integrations. For more information, see User providers.

New HTTP RSS integration template: Our new RSS integration template delivers a stronger integration and allows for more capability to configure the cached data structure. The set-up process is as easy as ever. For more information, see Integrate RSS.

August 2021

Change in synchronization behavior in the Workday integration: Workday synchronization has been improved to protect data loading. Rather than continue synchronizing when a failure is triggered, the synchronization reattempts for the failed API call five times. After a failed synchronization you can find information in the synchronization logs. For example, a failure state may occur if there is an expired password in the integration configuration, or a report URL in a workday module no longer exists or has a missing permission. For more information, see synchronization data.

New Oracle HCM integration template and microapp: Our new Oracle HCM integration template allows Citrix Workspace users to view and edit items and receive notifications directly in Workspace. For more information, see Integrate Oracle HCM.

New Covid-19 Self Certify integration template and microapp: Our new Covid-19 Self Certify integration template allows Citrix Workspace users to submit their Covid-19 vaccinated state using the self-certification response. For more information, see Integrate Covid-19 Self Certify.

New Relay state authentication option: Relay state provides an additional option to configure OAuth 2.0 authentication that enables users to access microapps without needing to reenter their credentials. This can be configured only if both you and your target System of Record use Okta. For more information, see Create HTTP integration.

Scripting support for webhooks: HTTP integration scripting now supports webhooks. You can configure scripts in your defined webhooks to achieve control over requests, response parsing, and transformation. For more information, see Integration scripting and Citrix Developer Portal.

July 2021

Canvas LMS integration template and microapps: Our new Canvas LMS integration template allows users to view courses, create course announcements, manage course enrollment, and view student's grades. For more information, see Integrate Canvas LMS.

New SAP Concur integration template and microapps: Our new SAP Concur integration template has a simplified configuration process and lets users submit requests and receive notifications about the status of requests. For more information, see Integrate SAP Concur.

New version of Citrix DaaS integration template: We have a new integration template available for Citrix DaaS that simplifies the set up process, improves synchronization, and extends the machines monitored. For more information see, Integrate Citrix DaaS. For a detailed article on upgrading to the new integration, see Upgrade your integration.

June 2021

New validation rules for service action parameter and template variable configuration: We've updated our integration validation rules for better future functionality. The following characters and phrases are no longer supported when defining your service action parameters and template variables during integration configuration:
`Whitespace ! " ## % & ' ( ) * + , . / ; < = > @ [ \ ] ^ { | } ~` **`true`**, **`false`**, **`else`**, **`null`**, `undefined`, **`this`**. For more information, see Configure the integration.

UPN support for filters, constraints, and service action parameters: Wherever you can select user email as a variable to extract data when building an action page, you can now use a user principal name (UPN) attribute. In Active Directory, the UPN attribute is a user identifier for logging in. For more information, see Configure UPN attribute for data filters.

SolarWinds integration template and microapps: Our new SolarWinds integration template allows users to submit and monitor tickets, service requests, and take action through Citrix Workspace. For more information, see Integrate SolarWinds.

May 2021

Smartsheet integration template and microapps: Our new Smartsheet integration template lets users manage sheets, discussions, update requests, and attachments. For more information, see Integrate Smartsheet.

Integration scripting: Integration scripting allows developers to write scripts that extend the capabilities of the platform for your HTTP integrations. Custom scripts can be bound to data endpoints and service actions to achieve full control over request preparation, response parsing, and transformation. For more information, see Integration scripting and Citrix Developer Portal.

New SAP SuccessFactors EC HTTP integration template: Our new SAP SuccessFactors EC template simplifies the integration configuration process using new scripting capabilities and adds a new **Skills** microapp. For more information, see Integrate SAP SuccessFactors.

New Tableau HTTP integration template: Our new Tableau HTTP template simplifies customization using new scripting capabilities. For more information, see Integrate Tableau.

Kronos Workforce Central integration template and microapps: Our new Kronos Workforce Central integration template allows users to view and respond to potential workforce management activities, perform time management tasks, and submit requests. For more information, see Integrate Kronos Workforce Central.

Updated DocuSign integration template: We've updated our DocuSign integration template. Implementing this new template requires admins to re-add the integration template. For more information, see Integrate DocuSign.

April 2021

New List/Grid component layouts We've added new preconfigured layouts to enable you to surface the right information in a way that suits your data the best. For more information, see the List/Grid entry under Display components.

Updated Citrix Cloud Status integration template: Our new Citrix Cloud Status integration template resolves performance issues. Implementing this new template requires admins to re-add the integration template. For more information, see Integrate Citrix Cloud Status Hub.

Blackboard Learn integration template and microapps: Our new Blackboard Learn integration template allows users to register for a new course and view the course and its related details as a student, and as an instructor to create a course announcement and view the course members and grades. For more information, see Integrate Blackboard Learn.

Scheduler improvements: Microapps synchronization has updated its scheduling mechanism for improved reliability. With this change, scheduled synchronization jobs will run at the interval defined after the last successful run. Before this change, scheduled jobs would try to run regardless if the synchronization was already running. For example, where the interval was set to 5 minutes beginning at 10.00, the job would try to run and fail at 10.05, 10.10, 10.15 until the synchronization job was finished. Now, if the interval is set to 5 minutes, the job starts at 10.00, runs (for example, for 15 minutes) and once successful pauses for an interval of five minutes and starts again. Therefore, starts at 10.00, runs successfully until 10.15, and then starts again at 10.20. For more information, see Synchronize data.

Webhook logs: To improve performance and stability, only the last 10 Webhook logs entries are kept for review in the Webhook logs screen. For more information, see Show Webhook logs.

New Workday HTTP integration template: Our new Workday HTTP template simplifies the integration configuration process using new scripting capabilities. All workflows are now available through one integration template. For more information, see Integrate Workday.

SAP Ariba HTTP integration template and microapps: Our new SAP Ariba HTTP integration template delivers a stronger integration to view and approve requisitions from Citrix Workspace. For more information, see Integrate SAP Ariba.

Updated Citrix Podio integration template: Our new Citrix Podio integration template resolves the integrity warning issue. The older template works as-is, but shows an integrity warning that is removed with this update. Implementing this new template requires admins to re-add the integration template. For more information, see Integrate Podio.

March 2021

List/Grid component:: We've released a new page builder component to display a list of data to users in Citrix Workspace. Select from preconfigured layouts to surface the right information in a way that suits your data the best. For more information, see the **List** component entry under Display components.

Script transformation improvements: Script transformation introduces extra improvements when dealing with scripts. You can now expand the text area for easier editing. Basic code validation is enabled to ensure your scripts have no mistakes, and general standard code interface improvements have been made for a better user experience. For more information, see Script transformation.

Ivanti integration template and microapps: Our new Ivanti integration template allows users to submit and monitor incidents, service requests, and take action through Citrix Workspace. For more information, see Integrate Ivanti.

New SocialChorus microapps We've added new microapps to our SocialChorus integration template. There's a new microapp **Featured Content** microapp. Also, new notifications for **Important Communications**. For more information, see SocialChorus microapps.

New Zoom integration template microapps: We've added new microapps to our Zoom integration template. Three new use cases have been addressed with these workflows; **Upcoming Meetings (Current Week):** View all upcoming meetings for the current week. User can edit and start the meeting, **My Office Hours:** Schedule Office Hours meeting according to preferences. User can choose the duration, start date, dial-in numbers, etc, and **Meeting Recordings:** View all the meeting recording for the last seven days. Also allows users to play recordings from any device. For more information, see Zoom microapps.

Slack integration template and microapps: Our new Slack integration template allows users to provide additional monitoring capabilities for critical channels that may not be traffic intensive but require the attention of its members. Users can initiate a new template for signature, be notified for any new pending documents, and view a list of envelopes previously sent or received in their DocuSign inbox. For more information, see Integrate Slack.

Qualtrics integration template and microapps: Our new Qualtrics integration template allows users to receive notifications about surveys that require a response, view active surveys requiring attention, and also to allow the survey manager to access survey statistics. For more information, see Integrate Qualtrics.

New OAuth grant type: HTTP Integration now supports the OAuth 2.0 implicit grant type flow when configuring service action execution. For more information, see Set up Service Authentication.

Zendesk HTTP integration template and microapps: Our new Zendesk HTTP integration template delivers a stronger integration to submit and monitor requests from Citrix Workspace. For more information, see Integrate Zendesk.

February 2021

New Citrix Podio FAQs microapp: Updates have been made to the Citrix Podio integration article for the new FAQs microapp. This microapp compiles a list of FAQs and make them available in Citrix Workspace. For more information, see Podio FAQs microapp.

Notification threshold: To improve performance a new feature has been added to limit the maximum number of notification cards that are generated per user per notification job. By default, this value is 50 and can be adjusted. For more information, see Increase notification threshold.

DocuSign integration template and microapps: Our new DocuSign integration template delivers a strong integration and allows users to send and receive envelopes for digital signatures from any device using Citrix Workspace. Users can initiate a new template for signature, be notified for any new pending documents, and view a list of envelopes previously sent or received in their DocuSign inbox. For more information, see Integrate DocuSign.

SocialChorus integration template and microapps: Our new SocialChorus integration template delivers a strong integration and allows users to communicate important announcements from management and share the content, such as articles, links and notes, between employees through different channels. For more information, see Integrate SocialChorus.

January 2021

Jira HTTP integration template and microapps: Our new Jira HTTP integration template provides more capability to configure the cached data structure and comes with a new microapp for creating epics from Workspace. For more information, see Integrate Jira.

MS Dynamics CRM HTTP integration template and microapps: Our new MS Dynamics CRM HTTP integration template delivers stronger integration and allows for more capability to configure the cached data structure. The set-up process is as easy as ever. For more information, see Integrate MS Dynamics CRM.

Script transformation: This feature allows you to enable inline script transformation for Data loading endpoints and Service actions. Scripts can be configured to receive a response object obtained from the HTTP response and transform it to another response object depending on your target integration System of Record (SoR). For more information, see Script transformation.

Citrix Podio integration template and microapps: Our new template delivers quick actions on Citrix Workspace utilizing the flexibility and diversified use cases on Podio. With this integration, you can easily connect our out-of-the-box microapps on Workspace to the corresponding Podio apps readily available on the Podio App Market, or your customized apps on Podio. For more information, see Integrate Podio.

Citrix DaaS integration template and microapps: Our new template allows you to search for and perform self-service actions from your Citrix Workspace. Users can check the status of their associated machines that are faulty. For more information, see Integrate DaaS.

Salesforce HTTP integration template: Our new Salesforce HTTP integration template delivers stronger integration and allows for more capability to configure the cached data structure. The set-up process is as easy as ever. For more information, see Integrate Salesforce.

New Salesforce out-of-the-box microapps: New Salesforce microapps enable notifications and workflows for anywhere access to leads, accounts, opportunities, cases, and contracts. For complete details about new Salesforce HTTP microapps, see Salesforce microapps.

Template variables: This feature increases flexibility when setting up your data endpoints and service actions during HTTP integration. Template variables allow you to insert dynamic values to your request configuration. For example, you can insert template variables into your Salesforce Object Query Language (SOQL) queries to download account objects easily. Incremental sync and action invocation have their own template variable definitions that enable you to override or change all parameters of the original endpoint definition in pre and post action update, action invocation, and incremental sync. For more information, see Configure the integration.

December 2020

Microsoft Outlook integration template (Preview): Our new Microsoft Outlook integration template delivers a strong integration and allows users to schedule events and office hours, edit events and office hours, and receive a notification an hour before an event's start time.

Pagination configuration: A new feature has been added to improve pagination configuration in HTTP integration: **Max pages to load**. Use this to set the limit of pages returned when handling large volumes of records. **Max pages to load** can be configured for each separate endpoint. Note: this does not affect current configurations unless you edit your endpoint and do not save, upon which the default value (1000) will be configured. For more information, see Configure the integration.

Google Calendar integration template (Preview): Our new Google Calendar integration template delivers a strong integration and allows users to schedule calendar events and list events and office hours of a user. For more information, see Integrate Google Calendar.

Component enhancements: We have significantly improved the performance of **Lookup component** queries. Also, for the **Table component** we've improved microapp tables with the user email filter on either the main or a joined table column. For more information about these and other table builder components, see Page builder components.

November 2020

MS Teams integration template (Preview): Our new MS Teams integration template delivers a strong integration and allows users to schedule Teams meetings, create a team from scratch or based on an existing team, add a new channel to an existing team, send a message to a specific channel and receive a notification for newly created channels. For more information, see Integrate MS Teams.

Citrix Cloud Status integration template and microapps: Our new template allows you to get updates on incidents and maintenance schedules which may impact some of the Citrix services. For more information, see Integrate Citrix Cloud Status Hub.

Workday HTTP integration template: Our new Workday HTTP integration template delivers stronger integration and allows for more capability to configure the cached data structure. This new template comes with microapps for change job requests, expense reports, and time off requests. For more information, see Integrate Workday.

New Workday HTTP out-of-the-box microapps: New Workday microapps enable notifications and workflows for change job requests, expense reports, and time off requests. For complete details about new Workday microapps, see Use Workday microapps.

Download attachments with new component: Our new **Attachment** component can list attachments from your data source and allows end users to download attachments. Images can be previewed directly. For more information, see Page builder components.

Google Meet integration template (Preview): Our new Google Meet HTTP integration template delivers a strong integration and allows users to schedule Google Meet meetings from any device or intranet. Users can select date, start time and end time, password, and co-organizers. The set-up process is as easy as ever. For more information, see Integrate Google Meet.

Power BI HTTP integration template: Our new Power BI HTTP integration template delivers stronger integration and allows for more capability to configure the cached data structure. The set-up process is as easy as ever. For more information, see Integrate Power BI.

New Power BI out-of-the-box microapps: New Power BI microapps enable notifications and workflows for viewing and managing users, and viewing groups. For complete details about new Power BI HTTP microapps, see Power BI microapps.

Power BI component enhancements: The Power BI component is needed to authorize the logged in user before they can view a dashboard, report, or tile. The out-of-the-box microapps that come with the Power BI template have the components configured as needed. To set this up from scratch, you need to configure a service action to generate a token for the user. For more information, see Configure Power BI component service actions.

October 2020

Component values as action parameters: You can now use component values as parameters in **Send Email** and **Go To URL** actions. This feature allows Workspace users to enter email recipients in a Workspace field for a given action and also user’s input from a Workspace form can be used as a part of a URL template opened in Workspace. For more information, see Page builder components - Actions.

Zoom integration template (Preview): Our new Zoom HTTP integration template delivers a strong integration and allows users to schedule meetings with their preferences from any device or intranet. For more information, see Integrate Zoom.

September 2020

Webhook logs: Use Webhook logs to view a history of requests and errors from all webhook endpoints. You can filter by webhook name and state, such as success, error, or all. For more information, see Show Webhook logs.

GoToMeeting integration (Preview): Our new GoToMeeting HTTP integration template delivers a strong integration and allows users to schedule GoToMeetings from any device or intranet. Users can select date, start time and end time, password, and co-organizers. The set-up process is as easy as ever. For more information, see Integrate GoToMeeting.

Webex integration (Preview): Our new Webex HTTP integration template delivers a strong integration and allows users to schedule Webex Meetings from any device or intranet. Users can host one-time/recurring meetings, add invitees and co-hosts, and select from different timezones. The set-up process is as easy as ever. For more information, see Integrate Webex.

August 2020

Custom icons: You can now add custom icons to better identify your integrations and improve user experience. For more information see Add custom icons.

Asynchronous data update for webhook listeners: Improvements to webhooks integration means that webhook request data is now synchronously validated, but asynchronously stored to the database. This leads to increased performance as webhook requests are protected from overloading the microapp database and more requests can be handled during peak periods. For more information, see Create HTTP integration.

July 2020

Google Directory HTTP integration: Our new HTTP Google Directory integration template delivers stronger integration and allows for more capability to configure the cached data structure. The set-up process is as easy as ever. For more information, see Integrate Google Directory.

New Google Directory out-of-the-box microapps: New Google Directory microapps enable notifications and workflows for viewing and managing users, and viewing groups. For complete details about new Google Directory microapps, see Google Directory microapps.

New notification option - Clear all notifications: You can now optionally remove all notifications from your microapps. Use this feature to clear your notifications if you need to reorganize or regenerate with a newer data structure. For more information, see Build event notifications.

Workday OOTB microapp enhanced - Create PTO Request: This Workday microapp now delivers an auto-generated comment in Workday explaining who created the PTO request and on behalf of whom the request was made. This feature provides more clarity into the request. This comment is seen only by managers, not the employees themselves.

Workday OOTB microapps enhanced - PTO Requests: Users are now directed to log in through SSO if they have SSO configured in Workday. Otherwise you are directed to the standard login page. For more information about all of our Workday OOTB microapps, see Workday microapps.

New Salesforce connector parameter - OAuth Authorization Base Url: Allows you to configure a custom OAuth login page for your Salesforce instance. For more information, see Add the Salesforce integration to Citrix Workspace Microapps.

SAP Concur OOTB microapp removed - Quick Expenses: This SAP Concur microapp is now deprecated. Users need to delete this microapp as it will become misconfigured. For more information, see Use SAP Concur microapps.

June 2020

Create integration data structures in depth: We've provided new examples and details on how to create table data structures beyond a relationship of one. Use the method described to create references to deeper data structures for use in your microapps. For more information, see Create integration data structures in depth.

On premises application support (General Availability): Deliver actions and notifications from your on-premises applications right into your Workspace or other channels. This capability is now generally available. For more information, see On-premises instance.

Webhook Listeners: You can now configure webhook listeners for your integration to enable your apps to provide near real-time data to your end users. Configuring a webhook allows your apps to deliver data to other applications at a much quicker rate than synchronization from the Microapp platform side. For more information, see Webhook listeners.

Data update before action execution: You can now configure Data Update before Action Execution to fetch fresh data from a system of records. This ensures your data is fully in sync before an action execution and builds trust in your microapps. For more information, see Configure the integration.

HTTP ServiceNow integration: Our new HTTP integration template delivers stronger integration and allows for more capability to configure the cached data structure. The set-up process is as easy as ever. For more information, see Integrate ServiceNow.

New ServiceNow out-of-the-box microapps: New ServiceNow microapps enable notifications and workflows for ServiceNow approvals, requests, change requests, delegates, incidents, and problems. For complete details about new ServiceNow microapps, see Use ServiceNow microapps.

HTTP Integration rate limiting: You can now optionally configure rate limiting when setting up your HTTP Integration. Use the feature when setting up your configuration based on the standard rate limits of your target integration system of record. For more information, see Create HTTP integration.

Bundle repository: Template integrations and their out-of-the-box microapps are now available through a bundle repository. When you want to add a new template integration in the Microapps service console, the add integration page is updated with the latest available template integrations for you to choose from. This interface now provides more information about the integrations. For more information, see Set up template integrations.

Product onboarding: In-product navigation has been introduced for the integration page in the Microapps service. This navigation delivers basic guidance and documentation links for core microapp developer tasks, such as service configuration, microapp creation, and subscriber management.

May 2020

Logic rule user improvement: We have introduced a usability improvement for creating logic rules in the microapp page builder. Easily create logic argument for your microapp components using the new streamlined and intuitive logic interface. For more information, see Customize microapps.

Italian language support: You can now localize your microapps to Italian when exporting and importing translated JSON files for the purposes of localization. For more information, see Localize microapps.

Pages update - Date/Time Properties: Updates to the Date/Time component in the page builder give you more control over your microapp time outputs. You can now configure the date and time display to show browser, 12-hour and 24 hour time outputs in addition to default time value displayed in the component. For more information, see Customize microapps.

Notification and page improvement - variables: Improved interface for configuring variables in the notification and page builders allows you to easily configure and control variables for your microapps. For more information, see Customize microapps.

Import new microapp version: You can now import newer versions of your microapps from each microapp’s option (ellipsis) menu. With this feature you can optionally keep your older microapps marked as end-of-life (EOL) with the newer version set to active. For more information, see Export and import integrations and microapps.

Extension of logic rules: The page builder now allows you to create logic rules for non-input components. For example, you can take a display component into account when deciding about the action of another component (hide, show, and so on). For more information, see Customize microapps.

Improvement to Datetime parameter in service actions: Working with the Datetime parameter in service action setup now produces the matching format in the resulting request.

April 2020

Notification improvements: The notification configuration process has been redesigned to improve your workflow when setting up your microapp notifications. You can now control your notification changes, execution and expiry settings, and general output of your published notifications from a single, unified configuration workflow. For more information, see Customize microapps.

HTTP integration table merging: The option to merge parent and child tables when configuring endpoints during HTTP integration is now available. With this feature you can choose whether the selected attributes in the merged table are taken from the parent or child table, or if both attributes are preserved. For more information, see Configure the integration.

Navigate to Power BI from report view in Citrix Workspace: Microapp pages that show reports and dashboards from Power BI now include link to the target source of record for a more detailed view. For more information, see Use Power BI microapps.

March 2020

On premises application support (tech preview): Deliver actions and notifications from your on-premises applications right into your Workspace or other channels. For more information, see On-premises instance (Tech Preview).

Data update after action execution: You can now configure Data Update after Action Execution to fetch fresh data from a system of records. This ensures your data is fully in sync after an action execution. This is an optional setting. For more information, see Configure the integration.

Performance improvement: Changes to microapp subscriptions are now applied immediately. This removes the need to wait when logging on and off.

ServiceNow OOTB microapp enhancement: ServiceNow microapps have been updated to include the *Lookup* page component, which has *type-ahead* searching capability. For more information, see Use ServiceNow microapps.

Data caching improvement: Enhancements to the HTTP integration feature allow users to check data more easily, from **Tables** after synchronization. For more information, see Configure the integration.

Jira OOTB microapp enhancement: Jira microapps have been updated to include the *Lookup* page component, which has *type-ahead* searching capability. For more information, see Use Jira microapps.

Workday OOTB microapps: New versions of Workday custom reports were released: staffing activities, expenses, change job, and time off request. We've added and updated our spreadsheets to let you create custom reports. For more information, see Create custom reports.

SAP Concur OOTB microapps: These SAP Concur microapps are now deprecated: **Submit Quick Expense**, **Create User**, and **User**. Users need to delete these microapps as they will become misconfigured. For more information, see Use SAP Concur microapps.

February 2020

New HTTP synchronization rules: When you define daily or weekly synchronization, synchronization occurs randomly within the timeslot you select. For example, selecting 00-04 daily full synchronize runs a full synchronize at a randomly selected time in that period. For more information, see Configure the integration.

Creating a new integration workflow includes importing: When creating a new integration for Microapps, you can now search from a variety of options to help you get started. Select from templates, create a customized HTTP integration, and now even import and export your own integration configurations and microapps. For more information about exporting and importing integrations and microapps, see Export and import integrations and microapps.

Enhancements to existing integrations and microapps: To support usability and adoption, we are constantly working to improve our template integrations and microapps. For example, we've implemented improvements to our out-of-the-box configurations to the RSS microapp.

Enhancements to page builder: The *Look Up* page component now has *type-ahead* searching capability.

Deprecation

December 1, 2022

The announcements in this article are intended to give you advanced notice of Microapps integrations and features that are being phased out. We provide this information so that you can make timely business decisions.

There are two important stages associated with the deprecation of a Microapp integration:

- Deprecation - On May 16, 2022, a **DEPRECATED** label will appear on the integration tile in the catalog. Any deployed instance of the integration continues to function and the integration is available for creation in the catalog.
- Maintenance - July 1, 2022. The integration is not available in the integrations catalog. Any deployed integration continues to function and the integration is available for creation in the

catalog. The integration is not fully supported and only critical bug fixes will continue to be made.

> **Note:**
>
> Integrations already deployed are unaffected and are still fully functional after the deprecation and maintenance dates.

Deprecation in Asia-Pacific (APS) geo location

Due to low usage, Citrix plans to deprecate the Microapps service in the Asia-Pacific (APS) geo location and migrate existing users to the US or EU geo locations, as applicable. All APS environments are scheduled for deprecation beginning December 30, 2022. Administrators of APS Microapps environments with active use are scheduled to receive an in-product message with further steps detailing the deprecation and migration process.

Deprecations and removals

The following integration templates are deprecated as of May 16, 2022. Integrations will be unavailable after July 1, 2022 01:00 AM where the integrations are end-of-life and no longer available for use:

- Adobe sign
- Blackboard learn
- Canvas LMS
- Cherwell
- Citrix Virtual Apps and Desktops
- DocuSign
- EmployeeSurvey
- Google Directory
- Google Meet
- GoToMeeting
- Ivanti
- Kronos Workforce Cen
- Microsoft Outlook
- Microsoft Teams
- MS Dynamics CRM
- Oracle HCM
- PowerBI
- Qualtrics
- Slack
- SocialChorus
- Solar Winds

- Tableau
- Upwork
- Webex Meetings
- Zendesk
- Zoom Meetings
- Google Calendar
- Smartsheet

The following integration templates are deprecated as of August 10th, 2022. Integrations will be unavailable after September 1, 2022 01:00 AM where the integrations are end-of-life and no longer available for use:

- Back To Work
- BambooHR
- COVID Vaccine Finder
- Digital Wellbeing
- Employee Recognition with Azure AD
- Freshdesk
- Grouproom
- New York Times
- Office Tracking log
- PlanFacts Integration by Zuric
- RunMy Process
- S2S Missed Call
- SharePoint Change Management
- Slack
- Spotify
- Virtual Metric
- Zoho CRM

Microapps service deprecation

June 1, 2022

The announcements in this article are intended to give you advance notice of Microapps integrations and features that are being phased out. We provide this information so that you can make timely business decisions.

Due to the current deprecation process of microapps applications, the Microapp service is removing unused environments from the platform from June 1, 2022 onwards. Any Microapp platform service environment that has no activity is scheduled for this deprovisioning.

If you believe you are wrongly affected by this process, you can request restoration of your environment. To do so, create a support task via your normal channel (customer service portal, phone, management escalation, and so on) with details and Support will create a ticket to validate the request and restore the environment.

- Support requests for environments raised within the first 30 days of deprovisioning can be restored to their previous status.
- Support requests raised after the first 30 days of deprovisioning cannot be restored as the database will be deleted. You can request a new, blank environment after this time period.

The microapps platform service reaches End of Life (EOL) on July 3, 2023. Existing customers actively using microapps can continue to use the service as-is until the EOL date. As part of the EOL process, we will be removing select integration templates from the Microapps catalog on July 1, 2022.

Deprecation schedule

Release Milestones and Dates for Microapps service:

- **June 1, 2022** - Notice of Change (NSC) - The date in which Citrix announces the intent to retire the Microapps service to impacted customers.
- **June 15, 2022** - Internal environments are removed.
- **July 1, 2022** - End of life (EOL) Microapps integration templates - Certain microapp integration templates are removed from the microapp catalog.
- **July 4, 2022** - External environments removal process begins.
- **July 3, 2023** - End of Life (EOL) Microapps service - Microapps entitlement will be removed from Workspace Service plans and customers will no longer have access to the Microapps service or functionality.

Deprecation FAQ

- **Which product is replacing Citrix Microapps service?**
  Citrix Work Collaboration Solutions is continuing to invest in digitizing and simplifying workflows and will unveil our product innovations in this area in the future. In the meantime, the Citrix Account Team is working with the Work Collaboration Solutions Product Management team to transition current Microapps customers to solutions that fit their use cases.

- **Can I continue using microapps until the EOL date?**
  You are entitled to the Microapps service through Workspace Service and can continue to access and use your microapps until the EOL date of July 3, 2023.

- **Can I still use microapp service templates after July 1, 2022?**
  You are entitled to the Microapps service through your Workspace Service and can continue to

build custom microapp workflows even after July 1, 2022. Removing the templates means customers not currently using the templates cannot start using these templated microapp workflows after July 1, 2022. Entitled customers keep the ability to build custom microapp workflows including the use cases the templates currently cover.
If you have Citrix Virtual Apps and Desktops microapp, you can still address the use case served by the templated microapps workflows without requiring the Microapps service. You can do this by using the available ITSM adapter or Connection Center in the Citrix Workspace app. Please reach out to your account team for more information on this.

Getting started

May 5, 2022

> **Note:**
>
> - Microapp integration templates are deprecated and will be unavailable after July 1, 2022.
> - After this period, the integration is end-of-life and no longer available for use.
> - For more information and a list of deprecated integrations, see Deprecation.

This article guides you through getting started with Citrix Workspace Microapps. Before you begin onboarding, make sure you consult the following articles about how credentials and data are handled and security guidance for deploying the service's management console:

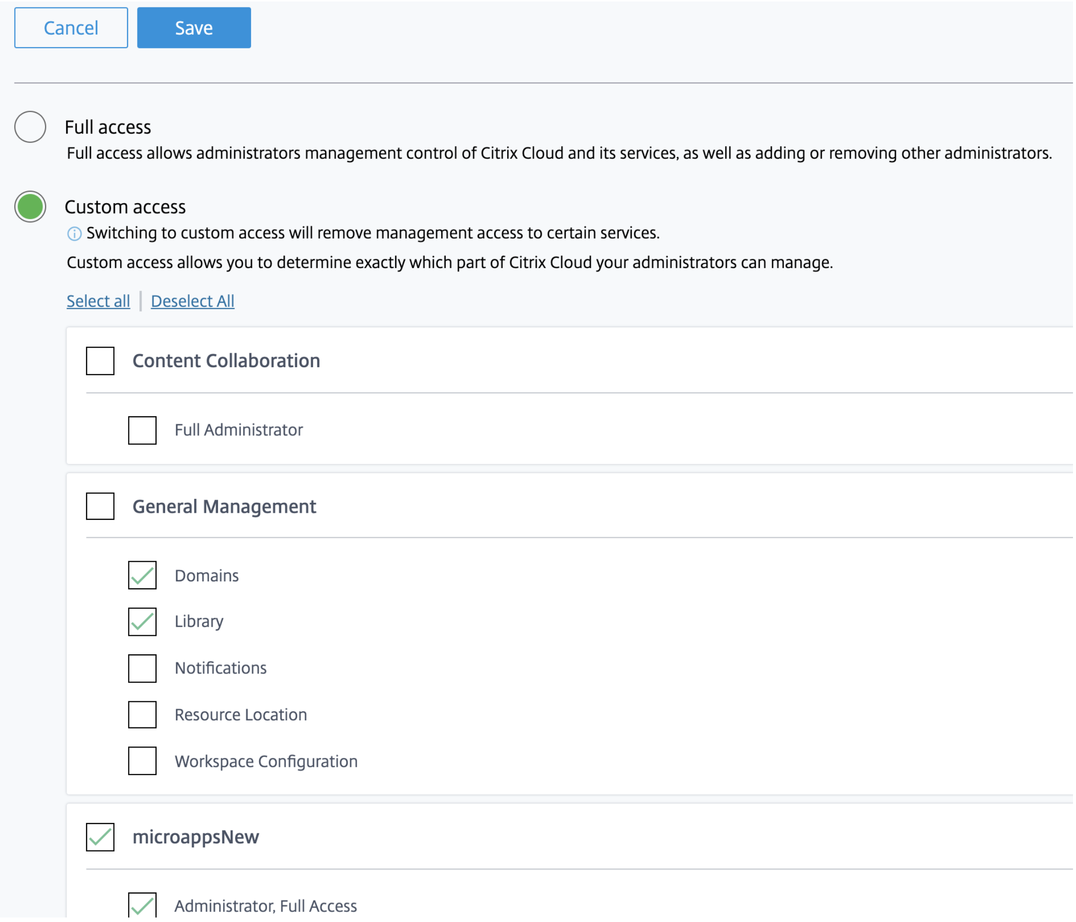
- Secure Deployment Guide for Citrix Cloud Platform
- Microapps Technical security overview
- Reference Architecture on Microapps service with Citrix Workspace

Add administrators

Enable correct administrator access to add subscribers to your microapps. Use this delegated permissions process to enable admins to add subscribers. For more information about managing access and subscribers, see Assign subscribers.

1. After signing in to Citrix Cloud, select **Identity and Access Management** from the menu, and select **Administrators**.
   The console shows all the current administrators in the account.
2. Locate the administrator that you want to manage, select the menu (ellipsis) button, and select **Edit Access**.
3. Select **Custom access**.
4. Ensure the following check boxes are selected and then select **Save**:
   - Under **General Management**, select **Domains** and **Library**.

- Under **microappsNew**, select **Administrator, Full Access**.



○ Full access

Full access allows administrators management control of Citrix Cloud and its services, as well as adding or removing other administrators.

● Custom access

ⓘ Switching to custom access will remove management access to certain services.

Custom access allows you to determine exactly which part of Citrix Cloud your administrators can manage.

Select all | Deselect All

□ Content Collaboration

□ Full Administrator

□ General Management

☑ Domains

☑ Library

□ Notifications

□ Resource Location

□ Workspace Configuration

☑ microappsNew

☑ Administrator, Full Access

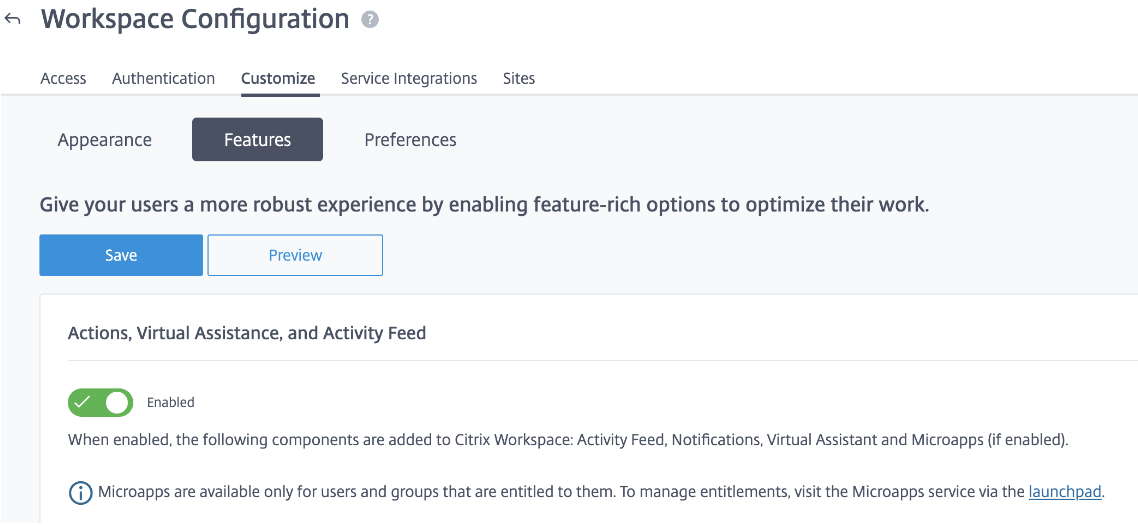
Repeat this procedure for all administrators who need to add subscribers. For more information about managing administrators including adding new administrators, see Add administrators to a Citrix Cloud account.

> **Note:**
>
> Granting domains and library admin access allows administrators to assign resources. For more information, see Assign users and groups to service offerings using Library.

Customize Workspace UI

Enable the Workspace UI for Microapps, notifications, and the activity feed by customizing the Workspace UI.

1. After signing in to Citrix Cloud, select **Workspace Configuration** from the menu.
2. Select **Customize** and then **Features**.
3. Toggle the switch to **Enabled**, and select **Save**.



Give your users a more robust experience by enabling feature-rich options to optimize their work.

Actions, Virtual Assistance, and Activity Feed

Enabled

When enabled, the following components are added to Citrix Workspace: Activity Feed, Notifications, Virtual Assistant and Microapps (if enabled).

Microapps are available only for users and groups that are entitled to them. To manage entitlements, visit the Microapps service via the launchpad.

Have users set up Workspace app

For your subscribers to use the workspace and access the apps, they must download and use the Citrix Workspace app or use the workspace URL. You must have a few SaaS apps published to your workspace to test the Microapps solution.

The Workspace app can be downloaded from Citrix Downloads. In the **Select a product** list, select **Citrix Workspace app**.

Whitelist URLs

If you have an outbound firewall configured, ensure that access to the following domains is allowed:

> - `https://*.cloud.com`
> - `https://*.citrixdata.com`

For more information, see Internet Connectivity Requirements.

Review prerequisites for integrations

Before setting up an integration, review the following:

- Review the best practices to resolve security, legal, and development issues before setting up integrations. For more information, see Best practices for application integrations.
- Review our specifications documentation for target applications. Each application integration article has a link to its respective technical specification.

Set up integrations

Now you're ready to set up an integration. Citrix Workspace Microapps has a large selection of out-of-the-box integrations for you to use. For more information about template integrations, see Template integrations and their microapps. Follow the steps to set up an application integration.

Or you can develop your own low-code HTTP integrations from scratch. For more information about planning and building a custom integration, see Build a custom application integration.

Where to go next

It's time to set up your microapps. For more information about customizing microapps, see Create microapps.

Manage access and subscribers. For more information, see Assign subscribers.

Technical security overview

July 20, 2021

This article applies to Citrix Workspace Microapps services hosted in Citrix Cloud.

Before you start deployment of Citrix Workspace Microapps services, review the Secure Deployment Guide for the Citrix Cloud Platform.

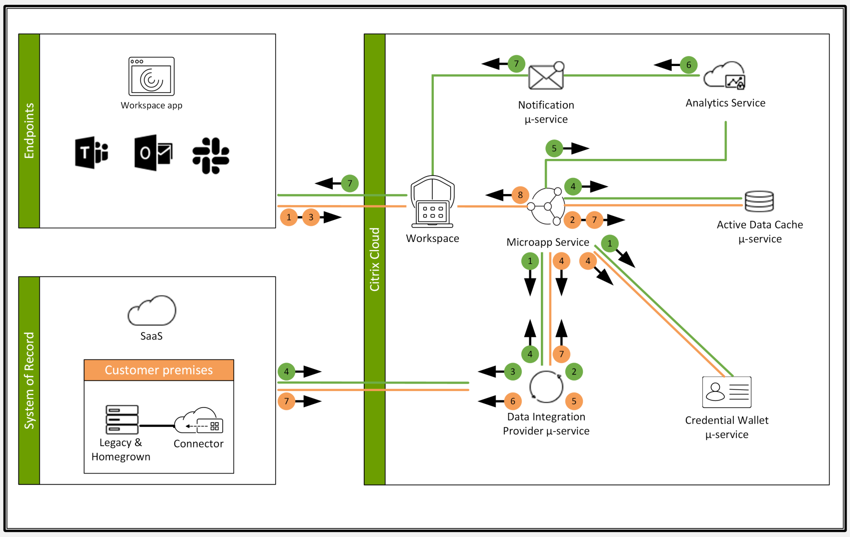
Security overview

The Microapps platform is integrated into Workspace and uses existing identity management. OAuth is the primary authentication mechanism for writing to SaaS applications. A SaaS-provided user access token specific to each user in Credential Wallet is stored to enable an efficient user experience for user actions.

This component is connected to the cloud service using an agent called the Citrix Cloud Connector. The following diagram illustrates the service and its security boundaries.

Citrix Workspace- Microapp Data Flow

Microapps enhance Citrix Workspace by increasing end-user productivity by streamlining functionality from complex enterprise application. This diagram depicts the data flow of how Citrix Workspace interacts and gets data from the systems of record as well as what happens once a user takes an action through a microapp.



System of record data sync & notification

1. Microapp service retrieves encrypted service account credentials for the system of record from the credential wallet and requests a sync from the data integration provider
2. Data integration provider decrypts the service account credentials
3. Data integration provider retrieves data from the system of record
4. Data integration provider streams system of record data to the microapp service which stores the data in active data cache
5. Microapp service sends raw events to Citrix Analytics
6. Citrix Analytics creates targeted scored notifications and sends to the notification service
7. Endpoints receive notifications

Microapp Action

1. Endpoint retrieves data from the microapp service to render microapp details
2. Microapp service retrieves data from active data cache to support microapp rendering
3. User invokes an action from the microapp. Microapp service receives action
4. Microapp service retrieves encrypted Oauth token from credential wallet and sends action to data integration provider
5. Data integration provider decrypts end user’s Oauth token for the system of record action that is being invoked upon
6. Data integration provider writes action to system of record under the identity of the end-user
7. Data integration provider reads back changed data; microapp service updates active data cache
8. Endpoint receives feedback on successful action

Credential handling

This service handles the following types of credential for Application Integrations:

- Basic user credentials
- OAuth
- OAuth2

General Security Overview

Microapps template apps and integrations are provided ‘as is’. All data, fields, and entities described in the connector guides are extracted in whole from the source system of record (SOR) and transferred to the Microapps cache. Templates are not configurable at this time.

Microapps stores a cache of data from the integrated system of record. For HTTP integrations, the scope of the data is fully configurable by the customer when setting up the configuration, and can be changed anytime.

A system account is used to retrieve the data for the cache using the target system of record API. The system account is typically required to have read-only access to all the data required for Microapps across all users using the microapps.

Credentials for the service account, in addition to OAuth tokens for individual user accounts are used for write actions are stored securely by Microapps using a service backed by Azure Key Vault.

Microapps also stores all the notification cards generated for users. Events triggering these notifications, in addition to content of these cards is also configurable by the administrator.

User interactions with the system are logged. Log data does not contain any system of records data or secrets.

Content is stored in the region selected when setting up your Citrix Cloud account. Regions currently available are US, EU, and AP/S (Asia Pacific South).

Customer content and logs are stored entirely within the Citrix platform and are not available to third parties.

Data backups of microapp for the purposes recovery are stored for the period of two weeks.

Microapp data is held for 30 days after the termination of an entitlement. After this grace period, all resources are deleted.

Data

Data is not explicitly seen, only entity properties and integration tables.

This data is displayed in two places:

1. On the Microapps admin/builder page (and therefore Workspace)
2. Data related to the record can be displayed in the feed cards (as defined variables). These definitions are in the microapp builder. Microapps can potentially be defined to export all data depending on administrator definition.

Example: if someone syncs or adds personal information (potentially PII) and sets to display all records and fields. The predefined templates are constructed according to security best practices and provided in a way that does not display sensitive data.

> **Note:**
>
> Creating integrations requires Administrator (or equivalent) privileges in the customer organization. Microapps provides the tools, it is up to the customer/administrators to utilize these tools to be as secure as possible based on the customer requirements.

Personally Identifiable Information (PII)

Personally Identifiable Information (PII) is designated on a case-by-case basis depending on geographical, business, security, and compliance considerations. PII is a matter for the administrator/customer to decide as each individual company/customer can have different definitions of what is ‘sensitive’ within their own organization.

Personally Identifiable Information changes depending on different geographical authorities but can include names, social security / national insurance numbers, biometric records, ID numbers, factors specific to physical, physiological, mental, economic, cultural, or social identity, or information that combined with other pieces of data, can identify an individual.

In terms of general best practices consider the following to identify and practice when handling data.

- Identify the PII your application integration stores
- Find all the places PII is stored
- Classify PII in terms of sensitivity

Identify PII

PII can include bank details and log-in information. Government agencies store PII including social security numbers, addresses, passport details, and license numbers.

PII Storage Location

PII can exist in a range of different locations such as file servers, cloud services, employee laptops, portals, and more including:

Data in use: The data employees use to do their jobs.

Data at rest: This is the data stored or archived in locations like hard drives, databases, laptops, Sharepoint, and web servers.

Data in motion: This is the data which is transitioning from one location to another. An example would be data moving from a local storage device to a cloud server or moving between employees and business partners via email.

PII Sensitivity Classification

PII must be classified based on its sensitivity. This is a vital part of PII protection. Consider if data is:

Identifiable: How unique is the PII data? If a single record can identify an individual by itself it is a sign that the data is highly sensitive.

Combined data: Try to identify two or more pieces of data that, when combined, can identify a unique individual.

Compliance: Depending on the type of organization you work for there are various regulations and standards for PII. Regulations you may be subject to include the Payment Card Industry Data Security Standard (PCI DSS), General Data Protection Regulation (GDPR), or HIPAA.

For more information regarding GDPR and Citrix see Citrix GDPR FAQ.

Read Access

The Microapps platform reads from the cache. The cache is encrypted at rest at the storage level, as provided by the cloud vendor. The READ access is not available in the source system. It is audited in Microapps platform.

Write Access

Write APIs are used. If services do not allow write APIs, we fall back to service accounts. In all cases, write access is auditable and attributed to the user in the source system.

Data flow

As the components hosted by the cloud service do not include the VDAs, the customer's application data and golden images required for provisioning are always hosted within the customer setup. The control plane has access to metadata, such as user names, machine names, and application shortcuts, restricting access to the customer's Intellectual Property from the control plane.

Data flowing between the cloud and customer premises uses secure Transport Layer Security (TLS) connections over port 443.

Citrix Cloud Connector network access requirements

The Citrix Cloud Connectors require only port 443 outbound traffic to the internet, and can be hosted behind an HTTP proxy.

The communication used in Citrix Cloud for HTTPS is TLS 1.0, 1.1, or 1.2. (See Deprecation of TLS versions for in-progress changes.)

IP whitelist configuration

You may need to configure third-party service providers, such as Salesforce, with designated IP addresses from which the microapps service accesses their APIs. The microapps service makes requests from an external IP address within the ranges listed below. Configure your third-party provider according to your microapp service region. Ensure you configure both listed ranges for the US and EU regions.

If you do configure designated IPs, we recommend bookmarking and revisiting this article regularly to stay informed with IP ranges and addresses that may be added or removed in the future.

US Region (32 addresses):

- 20.57.83.64/28
- 20.57.83.192/28

EU Region (32 addresses):

- 20.86.245.144/28
- 20.86.245.160/28

AP-S Region (16 addresses):

- 20.205.243.64/28

Security best practices

> **Note:**
>
> Your organization may need to meet specific security standards to satisfy regulatory requirements. This document does not cover this subject, because such security standards change over time. For up-to-date information on security standards and Citrix products, consult Citrix Security.

To ensure you have reviewed all security considerations:

- Grant users only the capabilities they require
- Review your organizations minimum security guidelines
- Use encrypted communication (HTTPS), with at minimum Transport Layer Security (TLS) 1.2
- Check your rate limits
- Use a system account with minimum permissions to fetch data (bulk reads)
- Consider how long you must retain data once the sync is not running
- Use public APIs for production
- OAuth2 is the preferred authentication protocol for this SaaS service
- Always use OAuth2 for write-backs
- Keep the integration updated. Verify it works and external APIs did not change

Where to go next

See the following resources for more security information and security guides for a similar Citrix Cloud service:

- Citrix Security
- Citrix ADC Secure Deployment guide
- Security considerations and best practices
- Smart cards
- Transport Layer Security (TLS)

> **Note:**
>
> This document is intended to provide the reader with an introduction to and overview of the security functionality of Citrix Cloud; and to define the division of responsibility between Citrix and customers regarding securing the Citrix Cloud deployment. It is not intended to serve as a configuration and administration guidance manual for Citrix Cloud or any of its components or services.

Set up template integrations

May 9, 2022

> **Note:**
>
> - Microapp integration templates are deprecated after May 16, 2022, and are unavailable after July 1, 2022.
> - After this period, the integrations are end-of-life and no longer available for use.
> - Integrations **already deployed** are unaffected and are still fully functional after the deprecation and maintenance dates.
> - For more information and a list of deprecated integrations, see Deprecation.
>
> You can set up a template integration and use the out-of-the-box microapps or build your own. Before you begin, make sure to review the best practices for configuring application integrations. For a comprehensive list of template integrations and their out-of-the-box microapps, see Template integrations and out-of-the-box microapps.

In addition to Citrix built template integrations, there are third party microapps. For a full list of these microapps, see Citrix Ready Marketplace.

If you want to build your own integration, see Deprecation.

Maintenance statement for Microapps integration templates

Citrix-provided integrations are templates. The developer of a template maintains it and ensures that it continues to work.

On release, new integration templates will typically be classified as **Citrix Labs**. This allows the functionality to mature as a result of initial customer feedback. For Citrix Labs templates, there is no commitment to support and support is provided by the developer on a best-effort basis. Citrix Labs integration templates are shared for the purpose of testing/validation. We do not advise deploying them in production environments. Refer to the listing of Citrix provided templates shown in the product when adding a new integration. Citrix Labs templates are listed in a separate section.

Customers can instantiate integration templates as integrations, and run them unchanged. Assuming the template has transitioned out of Citrix Labs, the developer of an integration will maintain it, keep it up-to-date, and fully working while the template is listed as Citrix-provided in the product. Reach out to the template developer to learn more about their support policies.

Customers can customize instantiated integrations. For Citrix's integration templates, there is no commitment to support customization and support is provided on a best effort basis. Reach out to the template developer to learn more about their support policies.

Template developers can remove templates from the product at any time. This will not remove instantiations from customer deployments. For integration templates that have left Citrix Labs, the developer commits to giving a minimum of four-week notice before ceasing maintenance of an integration template. Reach out to the template developer to learn more about their support policies.

Updates to the template integrations are not automatically rolled-out to customers. Customers decide when to take new versions of the template from the catalog. If the customer made any customizations based on the old version, these will need to be merged into the new version by the customer.

Best practices for application integrations

Review the following best practices when you prepare to implement a new integration for Citrix Workspace Microapps:

Development

Review development considerations:

- Consider fees for API service calls. Define what rate of API calls the use case requires. This includes who pays the fees, fee tiers, and an estimation of what your use cases' rate is.
- Ensure you have access to the target system.
- Verify that your app registration can be in your marketplace. Otherwise, every user has to create their own registration.
- Ensure that you are able to procure an instance of the target system.
- Do not use private APIs, and do not reverse engineer system-provided APIs.
- Continue to regression test connectors and track API agreement changes.
- Verify that data retention time periods that are required are compatible with the defaults for Citrix Cloud.
- If you use ad blocking software, disable it for Citrix Cloud and Workspace.

Legal

Review legal considerations:

- Review the user agreement.
- Review restrictions on use of service accounts.
- Review restrictions on distributions.
- Retain all agreements that you sign up for and review in a repository.
- For non-citrix microapps, read the microapp terms of service.

Security

Review security considerations:

- Review your organizations minimum security guidelines.
- Use encrypted communication (HTTPS), with at minimum Transport Layer Security (TLS) 1.2.
- Check your rate limits.
- Use a system account with minimum permissions to fetch data (bulk reads).
- Consider how long you retain data once the sync is not running.
- Use public APIs for production.
- OAuth2 is the preferred authentication protocol for this SaaS service.
- Always use OAuth2 for write-backs.
- Keep the integration updated. Verify it works and external APIs did not change.

Template integrations and their microapps

The following integrations are available for Citrix Microapps out-of-the-box. Select the target application to go to the details of setting up the integration:

Citrix Cloud Status

Incident: Receive Notifications for New Incidents and Incident updates. Also search for Incidents, and view their details.

Maintenance: Receive Notifications for New Maintenance schedules added to the Calendar, and when the Maintenance window starts and ends. Also search for Maintenance schedules, and view their details.

Citrix Podio

Broadcast: View all published (**Status: Live**) broadcasts that have a future **End Date**.

Create Broadcast: Create and publish new broadcasts.

Manage Broadcast: Administrators can view and update all created broadcasts. This view is not limited to published (**Status: Live**) broadcasts.

Citrix DaaS

My Desktops: Search for your Citrix Virtual desktops, and perform self-service actions from Citrix Workspace such as restarting.

My Sessions: Search for your Citrix Virtual sessions, and perform self-service actions from Citrix Workspace such as logging off and disconnecting.

Virtual Desktops (Admin Mode): Enables from Citrix Workspace CVAD Administrators to lookup machines, view their details, put machines into maintenance mode, and restart the machines.

Covid Self Certify

Self Certify: Submit your daily vaccination status response for covid-19 self-certification.

Employee Survey App

Survey Form: Provides notifications for new surveys and allows users to view and submit the survey forms.

Manage Survey: Allows survey app admin to manage the scheduled survey forms from Citrix Workspace. This microapp is only for Employee Survey App admin members.

Adobe Sign

Send Agreements: Send the template and upload an agreement to recipients for signature.

My Received Agreements: Provides a list of Adobe Sign agreements waiting for user's signature. A user can sign the agreement form

In Progress Agreements: List of in-progress Adobe Sign agreements for signature. User can also cancel the sent Agreement.

Completed Agreements: List of completed Adobe Sign agreements and details. User can also Share the received Agreements.

Canceled Agreements: List of Canceled Adobe Sign agreements and details.

Manage Agreements: Manage and edit template agreements.

SAP Ariba

Requisition Approval: View requisitions with details that are pending approval. Includes an approval action.

SAP Ariba (Legacy)

Requisition Approval: View requisitions with details that are pending approval, and an action to approve.

Blackboard Learn

Course Registration: Register for a course.

Create Course Announcement: Allows instructors to an create announcement for a course.

Instructor View: Allows instructors to view course members and grades.

My Courses: View course announcements and attachments of a course

My Grades: Allows students to view course grades.

Canvas LMS

Create Course Announcement: Teachers create an announcement for a course.

My Courses: View course announcements, assignments, and course files.

My Grades: Students view their course grades.

Teacher's View: Teachers view and add course members and view students.

Cherwell

Create Incident: Allows the user to create an incident.

Create Service Request: Create a new service request from the catalog or select predefined service requests templates.

My Open Incidents: View and manage open incidents that the workspace user owns.

My Open Serice Requests: View and manage open service requests that the workspace user owns.

SAP Concur

Approvals: Approve pending expense reports or send them back.

Expenses: Search and view expenses.

Itineraries: Search, view, and share itineraries.

Quick Expenses: Search and view quick expenses.

Reports: Search and view your submitted expense reports.

SAP Concur (Legacy)

Approvals: Approve pending expense reports or send them back.

Expenses: Search and view expenses.

Itineraries: Search, view, and share itineraries.

Quick Expenses: Search and view quick expenses.

Reports: Search and view your submitted expense reports.

Submit Quick Expense: Create a new quick expense report.

DocuSign

My Received Envelopes: View user's received envelopes, which are awaiting signature. User can authenticate himself and view the Envelope in DocuSign.

Send a Template: Send an existing template for signature.

My Sent Envelopes: View a list of envelopes sent by the user. Users also can view the details of the envelope, and also add, edit, and delete recipients.

Google Analytics

Session Metrics: View sessions and session metrics.

User Metrics: View user metrics.

Google Calendar

Create Event: Schedule an event according to user preferences.

My Calendar (Current Month): View list of the upcoming current month's one-time and recurring calendar events and the ability to edit the events.

My Office Hours: Create, edit, and view office hours.

Google Calendar (Legacy)

Calendar Events: Create and preview events.

Google Directory

Create User: Add a new user.

Directory Admin: Manage users and details.

Groups: View groups and details.

My Details: View your own details.

Users: View user details.

Google Directory (Legacy)

Directory Admin: Add a new user.

Directory Details: View details of teammates, including new employees and position changes.

Google Meet

Create Meeting: Schedule a meeting according to user preference.

Meeting Recordings: View list of all meeting recordings available for the user and watch the recorded videos.

GoToMeeting

Create a Meeting: This microapp is used to schedule a meeting according to user preference. The user has the option to select date, start time and end time, password, and co-organizers.

Ivanti

Agent Incidents: Self-service to manage incidents by an Agent. Subscribers must have agent access to Ivanti or resolutions will fail.

Create from Service Catalog: Create a service request based on the title of the microapp by using a deep link back to the self-service module of Ivanti.

Create Incident: Report a new incident in Ivanti. This is a quick create that can be submitted without an owner.

My Incidents: Self-service for my incidents.

My Service Requests: View existing service requests containing deep links to view the dynamic templates associated to the request.

Quick Service Request: Quickly create common service requests.

Jira

Create Epic: Create a new Jira epic with details.

Create Ticket: Create a new Jira ticket with details.

Tickets: View tickets, add comments, create subtasks, and change status and assignee.

Jira (Legacy)

Create ticket: Create a new Jira ticket with details.

Tickets: View tickets, add comments, create sub-tasks, and change status and assignee.

Kronos Workforce Central

My Accrual Balance: View accrual balance for different days instantly.

My Time Off History: Allows users to view their history of time off data available for current month.

Record Timestamp: Register their punch in and out date and time.

Request Time Off: Submit application for time off.

Request Vacation Approval: Receive notifications for all time off requests to managers and push notifications for all approved or refused time off requests back to the original requester.

Time Log: Receive notifications for all Work Time requested. Approve or refuse for the single user or group of users.

Microsoft Dynamics

Accounts: Search, view, and edit accounts.

Appointments: Search, view, and edit appointments.

Cases: Search, view, and edit cases.

Contacts: Search, view, and edit contacts.

Create Account: Create an account.

Create Appointment: Create an appointment.

Create Case: Create a case.

Create Contact: Create a contact.

Create Lead: Create a lead.

Create Opportunity: Create an opportunity.

Create Phone Call: Create a phone call.

Create Task: Create a task.

Leads: Search, view, and edit leads.

Opportunities: Search, view, and edit opportunities.

Phone Calls: Search, view, and edit phone calls.

Tasks: Search, view, and edit tasks.

Microsoft Dynamics (Legacy)

Accounts: Search, view, and edit accounts.

Appointments: Search, view, and edit appointments.

Cases: Search, view, and edit cases.

Contacts: Search, view, and edit contacts.

Create Account: Create an account.

Create Appointment: Create an appointment.

Create Case: Create a case.

Create Contact: Create a contact.

Create Lead: Create a lead.

Create Opportunity: Create an opportunity.

Create Phone Call: Create a phone call.

Create Task: Create a task.

Leads: Search, view, and edit leads.

Opportunities: Search, view, and edit opportunities.

Phone Calls: Search, view, and edit phone calls.

Tasks: Search, view, and edit tasks.

Microsoft Outlook

Create Event: Microapp is used to schedule an Event/Meeting as per the user preference.

My Calendar: Microapp is used to view and edit upcoming Events/Meetings.

My Office Hours: Microapp is used to create, view, and edit virtual office hours.

Microsoft Teams

Add Channel: Add a new channel to an existing team.

Create Meeting: Schedule an MS Teams meeting as per user preference.

Create Team: Create a team from scratch or based on an existing team as per user preference. Additionally, whenever a Channel is created for any team, the team owner will receive a notification.

Send Message: Send a message to a specific channel in any team.

Oracle HCM

Employee Directory: View and search employee directory.

Enter My Time: Enter daily time for selected entry types.

PTO: Create and submit absences.

Power BI

Dashboards: View details of Power BI dashboards.

Reports: View details of Power BI reports.

Qualtrics

My Surveys: Find all active surveys that are emailed to users. Notifications are sent when a survey is sent to a user and 24 hours before a survey expires.

Qualtrics Survey Statistics: For Brand Admins to manage surveys. Review statistics on active surveys. Notifications sent 24 hours before a survey expires with basic performance stats and the ability to deep link.

RSS

Item: Search for and view items.

Salesforce

Accounts: Search for, view, and edit accounts.

Cases: Search for, view, and edit cases that are assigned to you.

Contacts: Search for, view, and edit contacts.

Contracts: Search for, view, and edit contracts.

Create Account: Create a new account.

Create Case: Create a new case.

Create Contact: Create a new contact.

Create Contract: Create a new contract.

Create Event: Create a new event.

Create Lead: Create a new lead.

Create Opportunity: Create a new opportunity.

Create Task: Create a new task.

Events: Search for, view, and edit events.

Leads: Search for, view, edit, and convert leads.

Opportunities: Search for, view, and edit opportunities.

Tasks: Search for, view, and edit tasks.

Salesforce (Legacy)

Accounts: Search for, view, and edit accounts.

Cases: Search for, view, and edit cases that are assigned to you.

Contacts: Search for, view, and edit contacts.

Contracts: Search for, view, and edit contracts.

Create Account: Create a new account.

Create Case: Create a new case.

Create Contact: Create a new contact.

Create Contract: Create a new contract.

Create Event: Create a new event.

Create Lead: Create a new lead.

Create Opportunity: Create a new opportunity.

Create Task: Create a new task.

Events: Search for, view, and edit events.

Leads: Search for, view, edit, and convert leads.

Opportunities: Search for, view, and edit opportunities.

Pending Account Approvals: Search for and approve or reject accounts.

Pending Contact Approvals: Search for and approve or reject contacts.

Pending Contract Approvals: Search for and approve contracts.

Tasks: Search for, view, and edit tasks.

ServiceNow

Change Requests: Search for change requests, view their details, add comments, and update them.

Incidents: Search incidents, view their details, add comments, and update them.

Problems: Search for problems, view their details, add comments, and update them.

Request Approval: Search and view pending approvals, and approve or reject them.

Submit Change Request: Submit a new change request.

Submit Delegate: Submit a new delegate.

Submit Incident: Submit a new incident.

Submit Problem: Submit a new problem.

ServiceNow (Legacy)

Approvals: Search and view pending approvals, and approve or reject them.

Change Requests: Search for change requests, view their details, add comments, and update them.

Incidents: Search incidents, view their details, add comments, and update them.

Problems: Search for problems, view their details, add comments, and update them.

Submit Catalog Request: Select items and submit a new catalog request.

Submit Change Request: Submit a new change request.

Submit Delegate: Submit a new delegate.

Submit Incident: Submit a new incident.

Submit Problem: Submit a new problem.

Tasks: Search and view tasks including change requests, incidents, and problems.

Slack

My Favorite Channels: Receive notifications of activity in favorited channels.

Post to Slack: Post a message to the selected focused channel in slack.

Set My Slack Status: Set your slack status, create reminders, and enable **Do Not Disturb** for a set amount of time.

Smartsheet

Access Sheets: View sheets, share a sheet to a licensed or non-licensed user or group, add a sheet as favorite, and allow users to view their individual sheet.

Create a Sheet: Create a new sheet with fields and options such as sheet name, enter column title, select column type, and select primary column.

Discussion: Generate notifications to the discussion creator whenever there is a reply to theirs discussion thread.

My Update Requests: View sent and received update requests with details such as sent to, sent by, subject and status. Additionally, when a user sends an update request to recipients, the recipients receive a notification. Once the update request is complete,d the sender receive a completed notification.

Send Smartsheet as Attachment: Send Smartsheets as an attachment (PDF or Excel), with details such as To email, subject, and message.

Share with Admin: Used by Non-Admin users to share their sheets to admins with view only access, and to unlock other features such as Access sheets, My Update request, Discussions, and receive the respective notification. Share the sheet with Admin to unlock additional Smartsheet actions and notifications in Workspace for you, including: update request actions/notifications, discussion notifications, and viewing your sheets.

Start a Discussion: Start a discussion on sheet level.

SocialChorus

Important Communications: Search and view important communications from recommended channels that are posted within the last 7 days.

Latest Articles: Search and view content from subscribed channels that has been posted within the last 7 days.

Featured Content: Get recently featured communications from the subscribed channels with this microapp. Users can search, view images, and read content from Citrix Workspace.

SolarWinds

Create Ticket: Create a new incident.

Delete Ticket: Delete an incident.

My Assigned Tickets: View assigned tickets to update them and/or to change its status if needed.

My Open Tickets: Allows the user to see his requested incidents and update them if needed.

My Open Service Requests: Allows the user to see his service requests and update them if needed.

Service Catalog Request: Search for a service catalog item by name or create common service requests quickly.

SAP SuccessFactors EC

Directory: Search for employees and preview their details including skill set.

Skills: Search for skills and preview employees with corresponding skill set.

SAP SuccessFactors (Legacy)

Directory: Search, view, and edit employees with corresponding details.

Learning: Search, view, share, and register available learning courses.

Tableau

Reports: View details of Tableau reports.

Tableau (Legacy)

Reports: View details of Tableau reports.

Webex

Create a Meeting: Schedule a meeting with the option to select duration, time zones, invitees, and co-hosts.

Workday

Change Job: View and approve change job requests.

Create Change Job: Create a change job request.

Create Expense Report: Create an expense report.

Create Time Off Request: Submit a paid time-off (PTO) request.

My Time Off Request: View a personalized list of time-off requests.

Time Off Requests: View and approve paid time-off (PTO) requests.

Workday (Legacy)

Approval: Approve expense reports, time-off, and change job requests.

Create PTO Request: Submit a paid time-off (PTO) request.

My Expenses: View a personalized list of expense reports with report details and details of individual expense items.

My PTO Request: View a personalized list of time-off requests.

PTO Balance: View a personalized list of remaining time-off days.

Purchase Orders: View purchase orders with purchase order details.

Zendesk

Add Ticket: Submit Zendesk tickets.

Tickets: View Zendesk tickets with details.

Zendesk (Legacy)

Add Ticket: Submit Zendesk tickets.

Tickets: View Zendesk tickets with details.

Zoom

Create a Meeting: Schedule meetings according to your preference. User can choose the meeting title, duration, start date, co-organizers, and so forth.

Upcoming Meetings (Current Week): View all upcoming meetings for the current week. User can edit and start the meeting.

My Office Hours: Schedule Office Hours meeting according to preferences. User can choose the duration, start date, dial-in numbers, etc.

Meeting Recordings: View all the meeting recording for the last seven days. Also allows users to play recordings from any device.

End of Life process

This section describes the end of life (EOL) process for integrations that are replaced by newer HTTP integrations. When an integration is replaced with a newer HTTP integration, we refer to the older integration as legacy and the integration shows in the **Legacy** section of the admin console. Importantly:

- The legacy integration is available for three (3) months. After this period, the integration is no longer available for use.
- The legacy integration is supported for six (6) months. After this, there is no support for users running the integration.

Once an integration is replaced with a newer HTTP integration, we recommend migrating to the new integration. To migrate:

1. Set up the new HTTP integration.
2. Test the new integration and promote to production when ready.
3. Stop syncing old integration. For more information, see Set data synchronization.
4. Mark microapps of old integration as EOL.
5. Move over subscriptions. For more information, see Assign subscribers.

If no action is taken, the legacy integration continues functioning as-is. There will be a warning that the integration is no longer supported.

Verify needed entities

After you set up your template integration, use **Table** to verify your current list of tables stored in the cache and filters that are applied to those tables.

For template integrations, a large quantity of data can be pulled from your integrated applications to the Microapps platform. To better control and limit this amount of data, use the **Table** page to filter entities for your data synchronization to speed up synchronization.

1. From the Manage Microapps page, select the menu next to the integration that you want to verify entities for.
2. Select **Edit** and then **Table**.
3. Select **Edit Schema**.

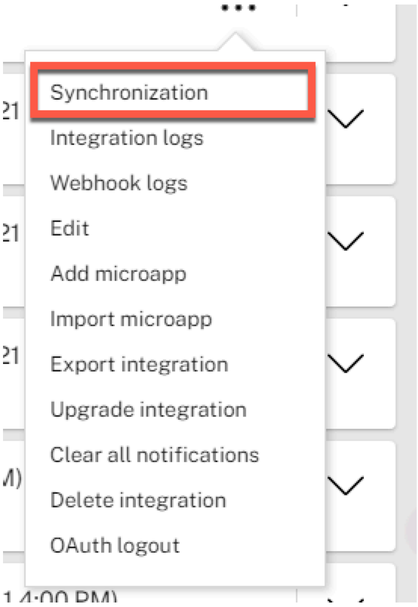
   The entities to sync for this integration, and filters that are applied to those tables, are listed. Entities with an information icon are required and cannot be edited.
4. You can unselect any entities that you do not require.
5. Select **Save**.

You are now ready to set and run your first data synchronization.

Set data synchronization

Pull data from your integrated applications to the Microapps platform so that a comparison can be made to the cache. As a best practice, full synchronization is performed every 24 hours and incremental syncs can be configured to pull every five minutes.

1. From the Manage Microapps page, select the menu next to the integration for which you want to set synchronization.
2. Select **Synchronization**.



3. Set **Full** and **Incremental** data synchronization values.

   - **Full** Drops the local cache and pulls all data from the source system.

     > **Important:**
     >
     > Running full synchronization can take a long time. We recommend running full synchronization at night or generally during off hours. You can cancel a data synchronization that is in progress at any time by selecting the *X* icon.

   - **Incremental** Pulls only changed (new and updated) records. Does not load deleted data.

     > **Important:**
     >
     > Not all APIs support incremental synchronization.
     >
     > When you define **daily** or **weekly** synchronization, synchronization occurs randomly within the timeslot you select. For example, selecting 00-04 daily full synchronize will run a full synchronize at a randomly selected time in that period.

4. Select **Save**.

> **Note:**
>
> You can also select the arrow icons to run the integrations on demand if necessary.

Trusted certificate authority

Trusted certificates are used to create secure connections to a server. A certificate authority (CA) certifies the ownership of a public key by the named subject of the certificate. CAs acts as a trusted third-party, being trusted both by the owner of the certificate and by the party relying upon the certificate.

For a list of trusted certificate authorities, see Trusted CA list.

Where to go next

If you want a quick start guide to onboarding Citrix Workspace Microapps, see Getting started.

If you want to build your own integration, see Build a custom application integration.

If you want to find out more about working with microapps, see Create microapps.

Integrate Citrix Cloud Status Hub

May 11, 2021

Integrate with Citrix Cloud Status to get updates on incidents and maintenance schedules, which may impact some of the Citrix services.

> **Note**
>
> This integration is built for Citrix Cloud Status, but it can also be used for any other StatusHub subscription by changing the Base URL, and subscribing to the respective Webhook update service.

For a comprehensive list of out-of-the-box Citrix Cloud Status microapps, see Use Citrix Cloud Status microapps.

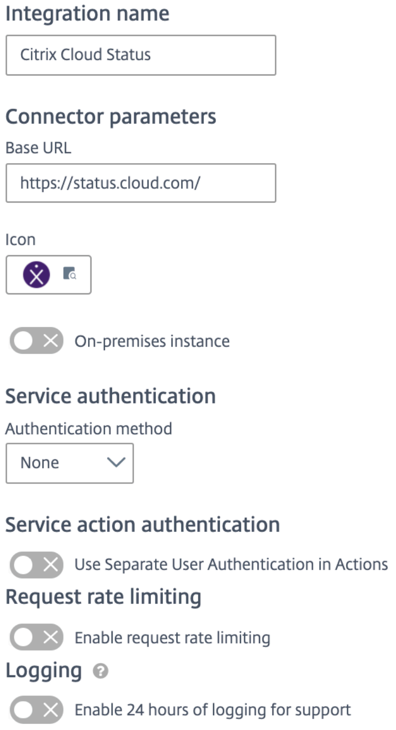
Add the Citrix Cloud Status integration

Follow these steps to set up the Citrix Cloud Status integration. The authentication options are preselected. Ensure that these options are selected as you complete the process.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Citrix Cloud Status tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Leave the instance **Base URL** as-is.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default Citrix Cloud Status icon.
5. Select **Add**.

The **Microapp Integrations** page opens with the Citrix Cloud Status integration and its microapps.



Integration name

Citrix Cloud Status

Connector parameters

Base URL

https://status.cloud.com/

Icon

On-premises instance

Service authentication

Authentication method

None

Service action authentication

Use Separate User Authentication in Actions

Request rate limiting

Enable request rate limiting

Logging

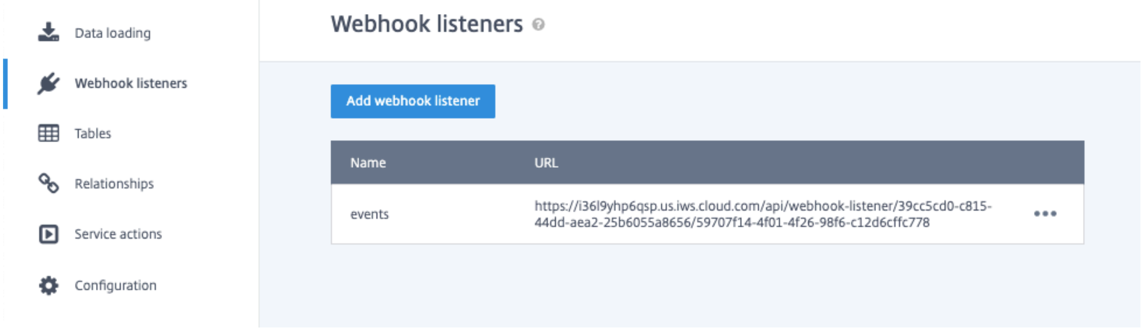
Enable 24 hours of logging for support

Set up the webhook listener

After you add the integration above, you need to set up the webhook listener.

Follow these steps:

1. From the **Microapp Integrations** page, navigate to the Citrix Cloud Status integration and select the **Edit** from the menu.
2. Select **Webhook Listeners** from the left-hand menu.
3. Copy the URL you see. Either use a shortcut, or select the ellipsis menu and select **Copy URL**.



Name	URL
events	https://i36l9yhp6qsp.us.iws.cloud.com/api/webhook-listener/39cc5cd0-c815-44dd-aea2-25b6055a8656/59707f14-4f01-4f26-98f6-c12d6cffc778

4. Navigate to https://status.cloud.com/subscribers/new and select **Webhook**.

5. Paste the webhook listener URL in the **Webhook URL** field. Accept the **terms & services** and select **Next**.
6. Select **All services**. If you want, for example, to subscribe separately for different regions, you can select **Selected services**, and then toggle **Aggregate by groups**.
7. If you do not want to receive all updates for an incident, select the check box **Only send me the minimum number of notifications per incident**.
8. Select **Save** to finish.

For more info about the synchronized entities, see Verify needed entities.

For more details of API endpoints and table entities, see Citrix Cloud Status connector specifications.

Use Citrix Cloud Status microapps

Existing Web/SaaS integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note**
>
> These microapps that are mapped to the Webhook tables provide new incidents that occurred after adding the integration.

Our Citrix Cloud Status integration template comes with the following preconfigured out-of-the-box microapps:

Incident: Receive Notifications for New Incidents and Incident updates. Also search for Incidents, and view their details.

Notification or Page	Use-case workflows
Incident - New (Notification)	When a new Incident occurs, subscribed users receive a notification.
Incident - Changed (Notification)	When an Incident is updated, subscribed users receive a notification.
Incident (Page)	Provides a read only view of an Incident when the user clicks an Incident notification.
Incidents (Action Page)	A table view with search and filter options for subscribed users to search for incidents in the events table (Webhook).
Incident Detail (Page)	Provides a read only view of an Incident when the user clicks an Incident in the Incidents (Action Page).

Notification or Page	Use-case workflows
Incident Update Detail (Page)	Provides a read only view of an Incident update when the user clicks an Incident update in the Incident/Incident Detail page.

Maintenance: Receive Notifications for New Maintenance schedules added to the Calendar, and when the Maintenance window starts and ends. Also search for Maintenance schedules, and view their details.

Notification or Page	Use-case workflows
Maintenance - New (Notification)	When a new Maintenance is scheduled and added to the calendar, subscribed users receive a notification.
Maintenance - Started (Notification)	When a Maintenance Window starts, subscribed users receive a notification.
Maintenance - Ended (Notification)	When a Maintenance Window ends, subscribed users receive a notification.
Maintenance Schedules (Action Page)	A table view with search and filter options for subscribed users to search for Maintenance schedules in the events table (Webhook).
Maintenance Detail (Page)	Provides a read only view of a Maintenance when the user clicks a Maintenance notification or Maintenance in the Maintenance Schedules (Action Page).

Integrate Citrix Podio

June 23, 2021

Integrate with Citrix Podio to deliver quick actions on Citrix Workspace utilizing the flexibility and diversified use cases on Podio. With this integration, you can easily connect our out-of-the-box microapps on Workspace to the corresponding Podio apps readily available on the Podio App Market.

- Podio Apps: Citrix has over 285 apps on Podio App Market or create your own apps as a business user with Podio. The app template lets you construct the app so that it suits your specific business needs and map to your team's unique process on Podio. For more information, see

Creating apps.

- This integration also allows you to create your own apps and custom solutions on Podio, build microapps, and deploy to Citrix Workspace. These microapps use Podio, a Citrix-owned service, as the System of Record (SoR). Your data transacted with these microapps resides with Citrix and follows Citrix guidelines and protocols.

Multiple Podio apps can use the same integration with Microapps. Consider the following when designing your Podio integrations:

- The App ID and App token are unique to the Workspace. This value can be used for all apps in a Podio Workspace.
- The Podio integration template is designed for push notifications. As this integration uses the App token rather than user permissions, user context writes are not supported.
- If the apps are in the same Podio Workspace, they're suitable for one integration with Microapps. If the apps are in different Podio Workspaces, then use different integrations.
- Also consider logical separation, user bases, and ease of management.

Review prerequisites

Review the following requirements. These are separated into two groups; integration level and Citrix Podio app level. Both are required for this integration template, although you can implement only the desired microapps at the app level.

Integration requirements

To set up this integration, you must have Admin permissions for the target Podio Workspace and have the Podio app.

These are the values that you enter in the integration configuration screen to set up the integration:

- **Base URL**: `https://api.podio.com`. This value is prefilled.
- **Podio app ID**: Each Podio workspace has an App ID. Enter this value as an **Access token parameter** when you set up the integration replacing the `podio_app_id` variable. You can use the App ID from any Podio app in the Podio workspace. See Collect App ID and App token.
- **App token**: Use this token to authenticate as an app rather than a user. Enter this value as an **Access token parameter** when you set up the integration replacing the `podio_token_id` variable. Collect this with the App ID.
- **Token URL**: This value is prefilled: `https://api.podio.com/oauth/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server. See Collect Client ID and Client Secret.
- **Client Secret**: The client secret is a unique string issued when setting up the target application integration.

Citrix Podio app requirements

These are the values you need to enter at the Podio app level to configure endpoints and service actions for each Podio app that you want to connect to through this integration. The following values can be required for each app:

- View ID: Template endpoints and service actions use Podio views as a filter to download a select quantity of records. Use this value to replace the `<podioapp>_view_id` template variable when you modify configurations. See Collect View ID.
- App ID: Each app has a unique ID. Use this value to replace the `<podioapp>_app_id` template variable when you modify configurations.

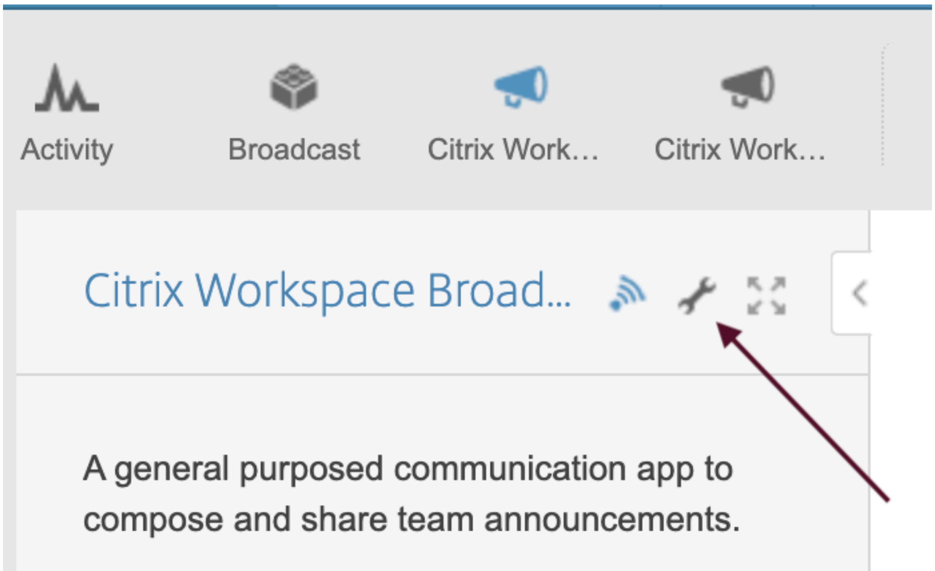
These are the unique values by Podio app:

Podio app	ID values	Get the app
Broadcast app	`broadcast_app_id` `broadcast_view_id`	Citrix Workspace Broadcast App
FAQs app	`faq_app_id` `faq_view_id`	Citrix Workspace FAQs app

Collect App ID and App token

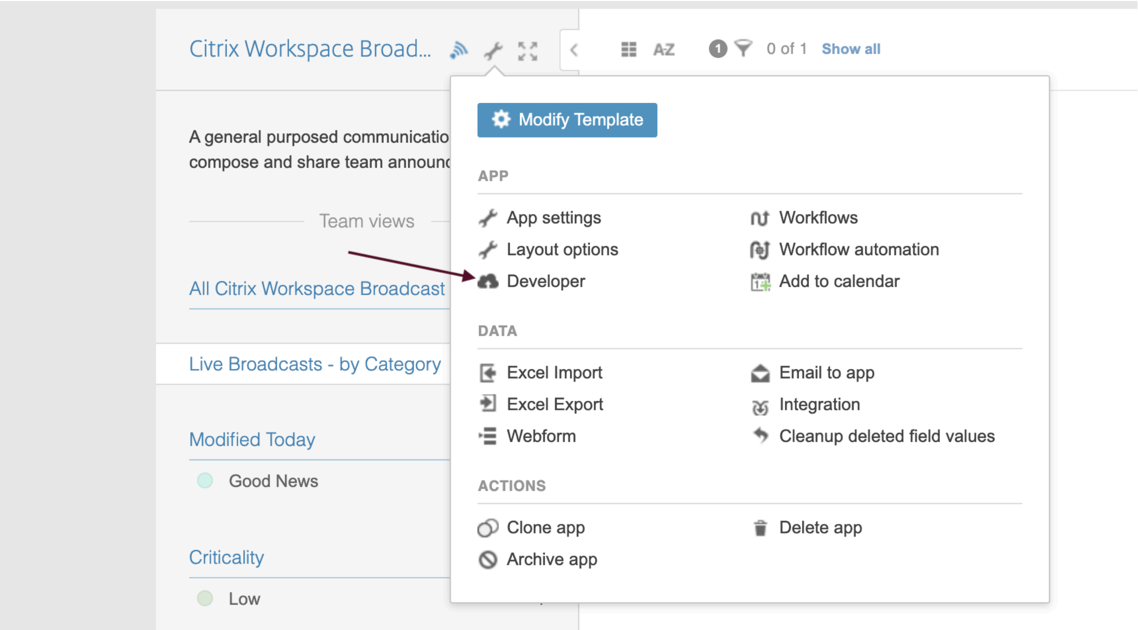
Collect the App ID and App token from Podio to authenticate as an app instead of a user. With App ID access, users can access only data of apps within the related Podio Workspace. You collect this ID from the Podio app for the Podio workspace. If you have multiple apps in a workspace, you can use any of the App IDs affiliated with the workspace.

1. Log in to Podio, and navigate to your Podio app.
2. Select the **Tools icon** next to the app name. We've shown the steps for the Broadcast app as an example in the screenshots.

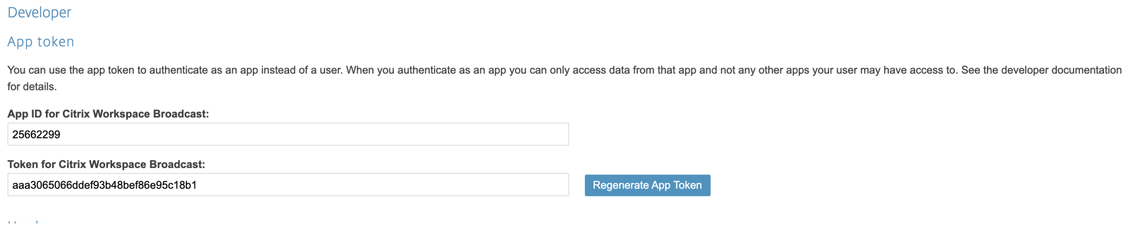


A general purposed communication app to compose and share team announcements.

3. Under **APP**, select **Developer**.



4. Copy and save the **App ID** and **Token** fields. You enter these values as **Access token parameter** when you set up the integration.



Developer

App token

You can use the app token to authenticate as an app instead of a user. When you authenticate as an app you can only access data from that app and not any other apps your user may have access to. See the developer documentation for details.

App ID for Citrix Workspace Broadcast:

25662299

Token for Citrix Workspace Broadcast:

aaa3065066ddef93b48bef86e95c18b1

Regenerate App Token

You can also view the fields and sample JSON values under **App fields** on this page. For more information about working with app entries in Podio, see Working with items.

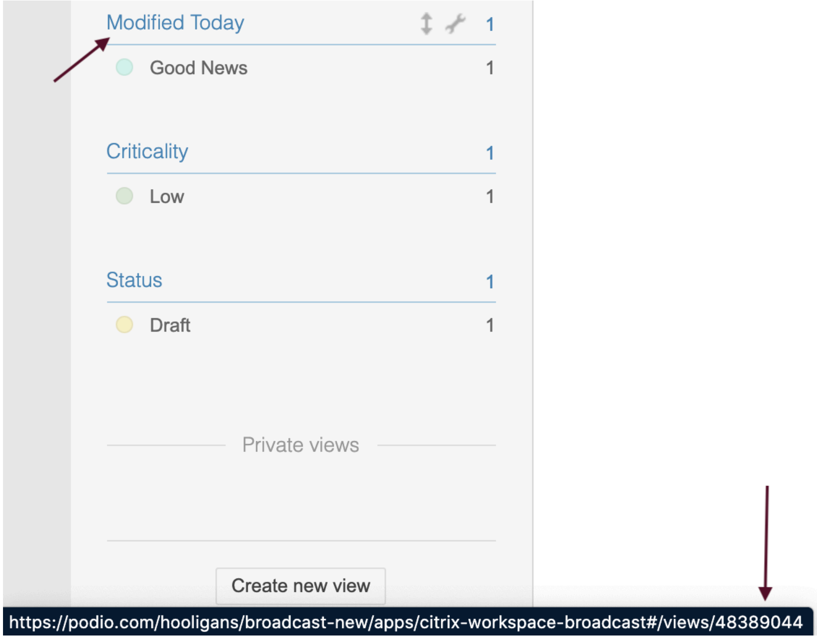
Collect View ID

Some Podio endpoints and service actions use Podio views as a filter to download a select quantity of records. For more information about using views and filters in Podio, see Views, filters and reports.

You need to find your view ID for the fields specified below, depending on which microapps you want to implement. You use this value to replace the `view_id` template variable when you replace service action variables below in Replace Data Loading and Service Action variables.

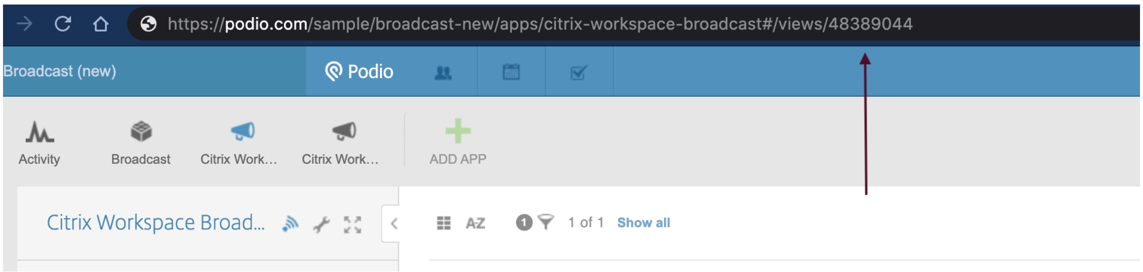
- For Broadcast app, the field is **Modified Today**
- For FAQs app, the field is **Published FAQs**

1. Log in to Podio, and navigate to your Podio app.
2. Mouse over the field specified above to see the view ID at the bottom-left of your browser. We've shown the field in the Broadcast app as an example in the screenshots.



https://podio.com/hooligans/broadcast-new/apps/citrix-workspace-broadcast#/views/48389044

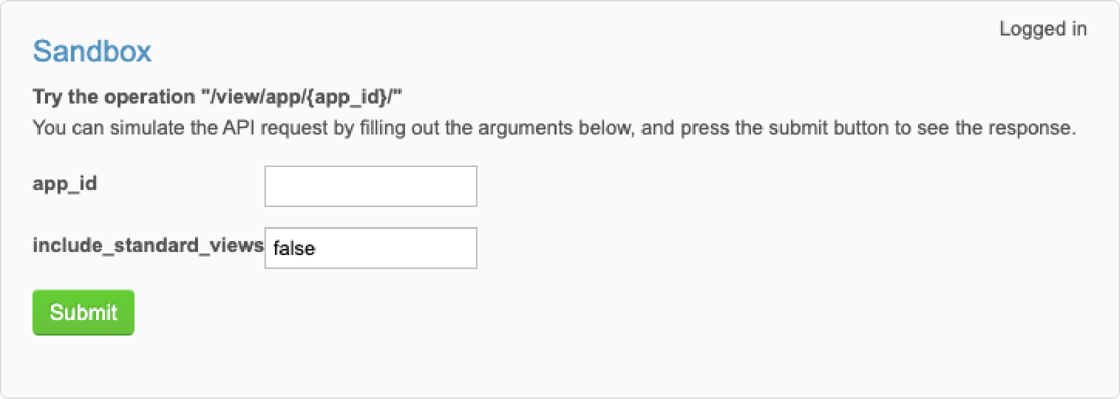
3. You can select the specified field to open the view. Copy the end part of the URL, and save for later use when you replace variables in service actions.



https://podio.com/sample/broadcast-new/apps/citrix-workspace-broadcast#/views/48389044

You can also find the View ID from the Podio developer's portal. Follow these steps:

1. Log in to Podio API Views.
2. Select **Get views**.
3. Scroll down to the **Sandbox** section and log in if required.
4. Enter the **app_id** that you collected in the Collect App ID and App token process.
5. Leave the **include_standard_views** value as *false*, and select **Submit**.



Sandbox

Logged in

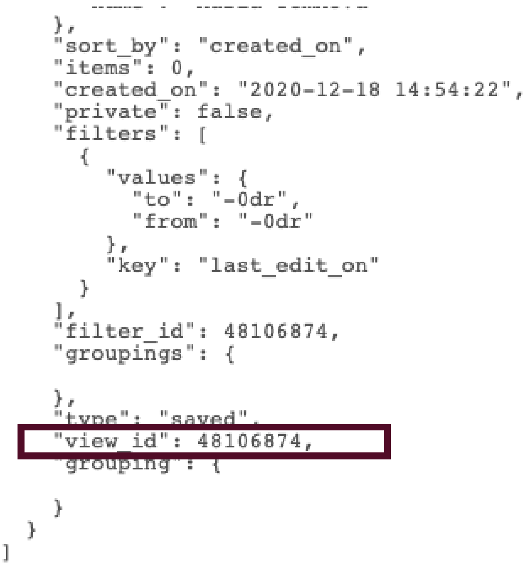
Try the operation "/view/app/{app_id}/"

You can simulate the API request by filling out the arguments below, and press the submit button to see the response.

app_id

include_standard_views: false

6. You can search (ctrl+F) through the output for **view_id**. It's near the bottom. Copy and save the value for later use.

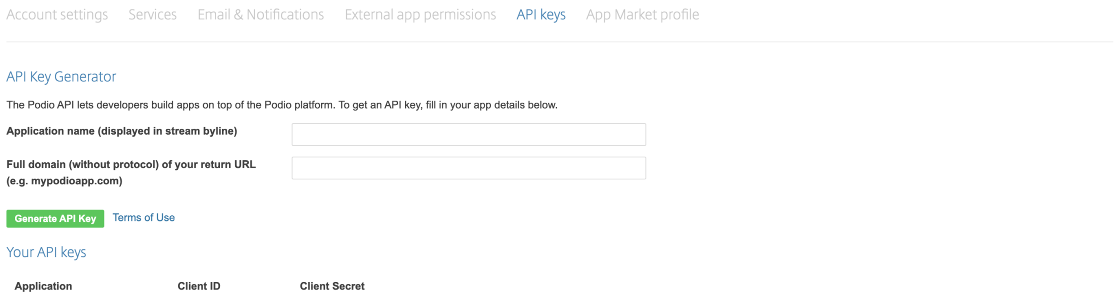


```
      },
      "sort_by": "created_on",
      "items": 0,
      "created_on": "2020-12-18 14:54:22",
      "private": false,
      "filters": [
        {
          "values": {
            "to": "-0dr",
            "from": "-0dr"
          },
          "key": "last_edit_on"
        }
      ],
      "filter_id": 48106874,
      "groupings": {

      },
      "type": "saved",
      "view_id": 48106874,
      "grouping": {

      }
    }
  ]
```

Collect Client ID and Client Secret

You need to collect a Client ID and Client Secret in Podio to enter in to Microapps set-up screen for authentication.

1. Log in to API keys. Complete the fields under **API Key Generator**.
2. Enter a name for **Application name**.
3. Enter your Microapps instance URL for **Full domain (without protocol) of your return URL**. This section of the URL `{ yourmicroappserverurl }` is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region(us/eu/ap-s)} .iws.cloud.com`.
4. Select **Generate API Key**.
5. Under **Your API keys**, copy and save the **Client ID** and **Client Secret** values for the application you just added. You enter these values when you set up the integration.



The Podio API lets developers build apps on top of the Podio platform. To get an API key, fill in your app details below.

Application name (displayed in stream byline)

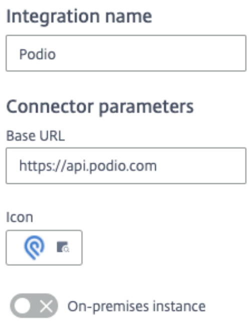
Full domain (without protocol) of your return URL (e.g. mypodioapp.com)

Add the integration to Citrix Workspace Microapps

Add the Podio integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Podio tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**:
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

Podio

Connector parameters

Base URL

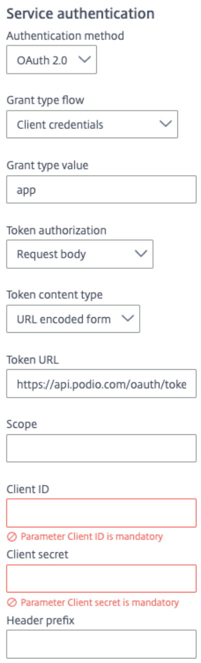
https://api.podio.com

Icon

On-premises instance

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

   a) Select **Client credentials** from the **Grant type flow** menu.
   b) Enter **app** in the **Grant type value** field.
   c) Select **Request body** from the **Token authorization** menu.
   d) Select **URL encoded form** from the **Token content type** menu.
   e) Confirm the **Token URL** field. This value is prefilled: `https://api.podio.com/oauth/token`
   f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the Client Secret when you generated an API key.
   g) Enter your **Client secret**. The client secret is a unique string issued with the Client ID while generating an API key.



Service authentication

Authentication method: OAuth 2.0

Grant type flow: Client credentials

Grant type value: app

Token authorization: Request body

Token content type: URL encoded form

Token URL: https://api.podio.com/oauth/toke

Scope:

Client ID:

Parameter Client ID is mandatory

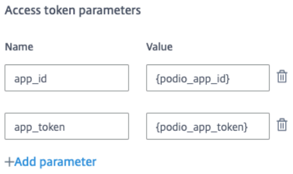
Client secret:

Parameter Client secret is mandatory

Header prefix:

6. Enter **Access token parameters** values. For more information, see Collect App ID and App token:

    - Enter the **App ID** that you collected next to **podio_app_id**. You can use the App ID from any Podio app in the Podio workspace.
    - Enter the **App token** that you collected next to **podio_app_token**.



Access token parameters

Name	Value
app_id	{podio_app_id}
app_token	{podio_app_token}

+Add parameter

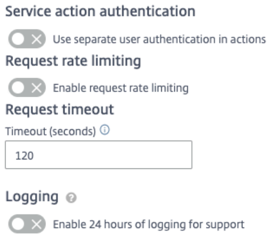
7. Do not enable the **Service action authentication** toggle.

8. Do not enable the **Request rate limiting** toggle.

9. Enter *120* in the **Request timeout** field.

10. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

11. Select **Save** to proceed.



Service action authentication

Use separate user authentication in actions

Request rate limiting

Enable request rate limiting

Request timeout

Timeout (seconds): 120

Logging

Enable 24 hours of logging for support

The **Microapp Integrations** page opens with your added integration and its microapps. Next you need to make some changes to template variables for endpoints and service actions as described below.

Replace Data Loading and Service Action variables

To complete this set up, you need to replace the app ID and view ID template variables in the Podio integration configuration with the App ID and View ID that you collected above for each of the Podio apps you are connecting. You replace these variables for data loading and service actions.

Quick guide to replacing variables

These are the values you need to enter at the Podio app level to configure endpoints and service actions for each Podio app that you want to connect to through this integration:

- `broadcast_app_id` Modify endpoints: **Broadcast**, **Broadcast Category**, **Broadcast Criticality**, **Broadcast Status**; and service actions: **Create Broadcast**, **Update Broadcast**.
- `broadcast_view_id` Modify endpoint: **Broadcast**; and service action: **Create Broadcast**.
- `faq_app_id` Modify endpoints: **FAQ Category** and **FAQs**.
- `faq_view_id` Modify endpoint: **FAQs**.

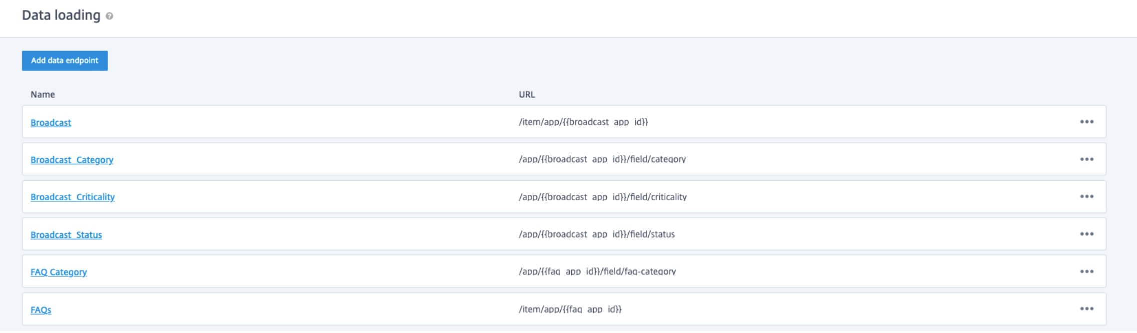
Modify the following variables in the locations given. Step-by-step guidance is provided below.

Replace Data Loading variables

For each data endpoint, you must manually add your variable for app ID and view ID. You do this for all six endpoints.

- `broadcast_app_id` Modify endpoints: **Broadcast**, **Broadcast Category**, **Broadcast Criticality**, **Broadcast Status**.
- `broadcast_view_id` Modify endpoint: **Broadcast**.
- `faq_app_id` Modify endpoints: **FAQ Category** and **FAQs**.
- `faq_view_id` Modify endpoint: **FAQs**.

1. From the **Microapp Integrations** page, select the menu next to the Podio integration, and then **Edit**. The **Data Loading** screen opens. If you are in the configuration screen, select **Data Loading** from the left side navigation column.
2. Select the menu next to the first endpoint and then select **Edit**, or select the name of the endpoint. Let's start with **Broadcast** endpoint.

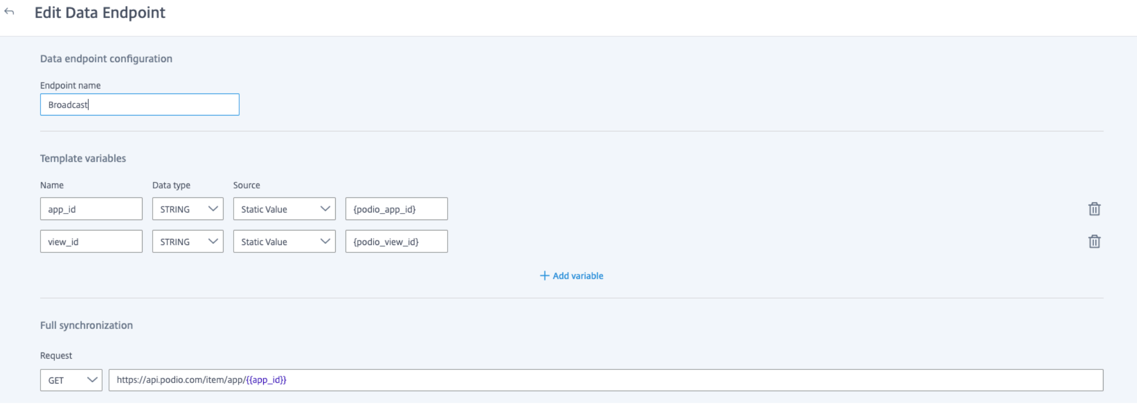


Data loading

Add data endpoint

Name	URL
Broadcast	/item/app/{{broadcast_app_id}}
Broadcast_Category	/app/{{broadcast_app_id}}/field/category
Broadcast_Criticality	/app/{{broadcast_app_id}}/field/criticality
Broadcast_Status	/app/{{broadcast_app_id}}/field/status
FAQ Category	/app/{{faq_app_id}}/field/faq-category
FAQs	/item/app/{{faq_app_id}}

3. In the **Edit Data Endpoint** screen, under **Template variables** replace the **{broadcast_app_id}** and **{broadcast_view_id}** variable with your App ID.



Edit Data Endpoint

Data endpoint configuration

Endpoint name: Broadcast

Template variables

Name	Data type	Source	
app_id	STRING	Static Value	{podio_app_id}
view_id	STRING	Static Value	{podio_view_id}

+ Add variable

Full synchronization

Request: GET https://api.podio.com/item/app/{{app_id}}

4. Select **Apply** at the bottom of the screen and confirm.

5. Now repeat this procedure for the other five endpoints, replacing the values as required. Don't forget to select **Apply** at the bottom of the screen and confirm to save for every endpoint.

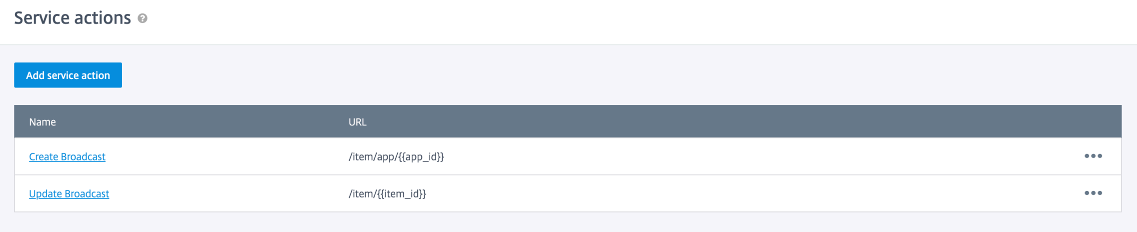
Replace Service Action variables

For each service action, you must manually add your details for `broadcast_app_id` and `broadcast_view_id`. The **Create Broadcast** service action requires you to replace both variables. The **Update Broadcast** service action only requires replacing the `broadcast_app_id` variable.

- `broadcast_app_id` Modify service actions: **Create Broadcast** and **Update Broadcast**.
- `broadcast_view_id` Modify service action: **Create Broadcast**.

1. While editing the integration configuration, select **Service Actions** from the left side navigation column.

2. Select the menu next to one of the service actions and select **Edit**, or select the name of the service action. Let's start with the **Create Broadcast** service action.

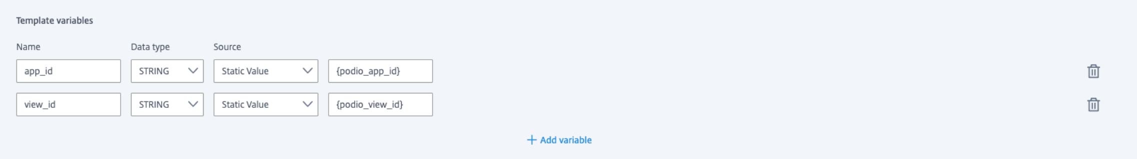


Service actions

Add service action

Name	URL
Create Broadcast	/item/app/{{app_id}}
Update Broadcast	/item/{{item_id}}

3. In the **Edit Service Action** screen, under **Template variables** replace the **{broadcast_app_id}** and **{broadcast_view_id}** variables with your App ID and View ID that you collected earlier.



Template variables

Name	Data type	Source	
app_id	STRING	Static Value	{podio_app_id}
view_id	STRING	Static Value	{podio_view_id}

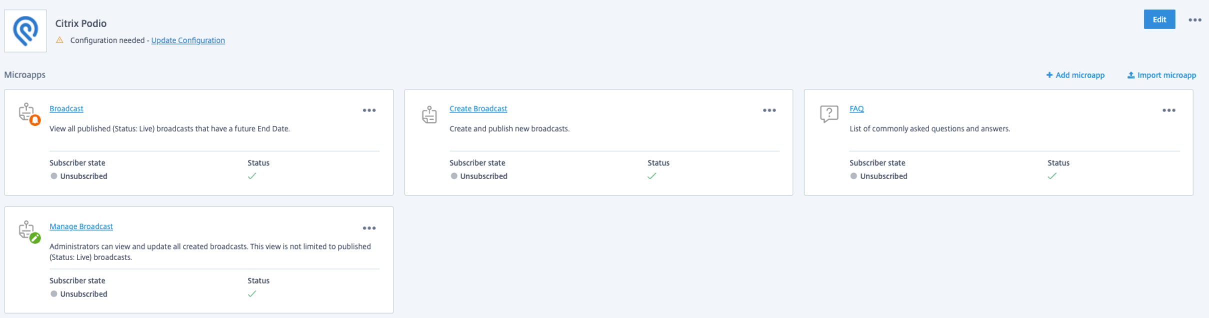
4. Select **Save** to finish.

5. Now repeat this for the other service action: **Update Broadcast**, but only replacing the `broadcast_app_id` variable.

For more details of API endpoints and table entities, see Podio connector specifications.

Use Podio microapps

This Podio integration template comes with these out-of-the-box microapps. Start with these microapps and customize them for your needs.



Citrix Podio

⚠ Configuration needed - Update Configuration

Microapps

Broadcast: View all published (Status: Live) broadcasts that have a future End Date. Subscriber state: Unsubscribed. Status: ✓

Create Broadcast: Create and publish new broadcasts. Subscriber state: Unsubscribed. Status: ✓

FAQ: List of commonly asked questions and answers. Subscriber state: Unsubscribed. Status: ✓

Manage Broadcast: Administrators can view and update all created broadcasts. This view is not limited to published (Status: Live) broadcasts. Subscriber state: Unsubscribed. Status: ✓

Podio Broadcast microapps

Use these microapps to compose and share important announcements with your team. These announcements might be for general, facility or IT updates, change in processes, critical crisis communication, or just engaging your team with some entertainment and well-being news.

The Citrix Workspace Broadcast app is a pre-requisite for the Broadcast microapps. You need to install this app from the Podio App market. This Podio app template comes with a predefined set of values for the Category and Criticality fields. Use the Modify Template feature on the app to add or modify options for these fields to suit your business needs. This modification would not require any changes to the corresponding microapps. You can also modify the Podio app by adding or editing other fields. However, this would require you to reflect the corresponding changes in the microapp too. Get the Podio app: Citrix Workspace Broadcast App.

Familiarize yourself with the following considerations:

- The **Create Broadcast** and **Manage Broadcast** microapps described below are admin microapps.
- The **Live** status in these microapps means publishing the message. When a message is published, it is made available in the Broadcast microapp for standard (non-admin) users.
- If any Podio app status field is modified to have different values, then the correct value needs to be updated in the microapp. For example, if you want to use *Published* as a status instead of *Live*, the microapp needs to be customized to the same.

Broadcast: View all published (**Status: Live**) broadcasts that have a future **End Date**.

Notification or Page	Use-case workflows
Changed Broadcast notification	When a broadcast is changed and live, subscribers receive a notification.
New Broadcast notification	When a new broadcast is live, subscribers receive a notification.
Broadcast page	Provides a read-only page with broadcast details.
Broadcasts page	Provides a list of your broadcasts with a link to details.

Create Broadcast: Create and publish new broadcasts.

Notification or Page	Use-case workflows
Create Broadcast page	Provides a page for creating and publishing broadcasts.

Manage Broadcast: Administrators can view and update all created broadcasts. This view is not limited to published (**Status: Live**) broadcasts.

Notification or Page	Use-case workflows
Changed Broadcast notification	When a broadcast is updated, subscribers receive a notification.
New Broadcast notification	When a new broadcast is live, subscribers receive a notification.
Broadcasts page	Provides a searchable list of created broadcasts with a link to details where you can update broadcasts.
Broadcast page	Provides a read-only page similar to the page used in the Broadcast microapp that shows how the broadcast looks to the public. There is an **Update** button that leads to the Update Broadcast page where admins can modify this broadcast.
Update Broadcast	Provides a page for updating an existing broadcast.

Clean up expired broadcasts

Enable this workflow to automate identification of broadcasts whose end date has expired. The broadcast is marked as completed and is no longer shown to subscribers.

This workflow is available with the Podio Premium plan. See Workflow Automation for more information. The admin that set up this integration enables the flow by refreshing from Podio after logging in to https://workflow-automation.podio.com and refreshing the related Podio workspace.

Podio FAQs microapp

Compile a list of FAQs and make them available in Citrix Workspace. The Citrix Workspace FAQs app is a pre-requisite for the FAQs microapp. You need to install this app from the Podio App market. Get the Podio app: Citrix Workspace FAQs app.

FAQs: List of commonly asked questions and answers.

Notification or Page	Use-case workflows
New Article notification	When a new article is published, subscribers receive a notification.

Notification or Page	Use-case workflows
FAQ page	Provides a searchable list of FAQs, which can be filtered by category, with a link to the Question and Answers page.
Question and Answers page	Provides a detailed view of an FAQ, with a link to source material.

Integrate Citrix DaaS

March 31, 2022

Deploy the Citrix DaaS (CVADs) integration to add microapps that help people perform self-service actions from Citrix Workspace. Users can check the status of their associated machines and sessions, and perform operations such as restart, disconnect, and log off. Admins can also turn on maintenance mode for machines with an Administrator Mode microapp

A short 98-second Tech Insight video showing the functionality:

This is an embedded video. Click the link to watch the video

For more information about the microapps in this integration, see Use Citrix DaaS microapps. For more information about upgrading this Citrix DaaS integration, see Upgrade your integration.

Deployment steps

1. Create a Secure Client, which is used by the integration to communicate with the CVAD service APIs
2. Choose an appropriate API proxy location for the Token URL
3. Collect required information that must be filled in to the integration configuration
4. Add the CVAD service integration, and configure it
5. Subscribe people to the microapps, so that they can be used

Review prerequisites

To set up this integration, you must have **Citrix Workspace**, **Microapps**, and **Citrix DaaS** enabled in Citrix Cloud.

This integration template supports on-premises and cloud VDAs if you use **Citrix DaaS** for brokering.

This integration does not support VDAs brokered from on-premises CVAD, because on-premises CVAD does not have public APIs available, and this integration relies on the CVAD Service Public APIs.

These are the values that you enter when configuring the CVAD service integration in Citrix Workspace Microapps:

- **Customer ID**: The customer ID used when calling the CVAD service APIs. Your customer ID is found in Citrix Cloud on the Secure Clients page.
- **Token URL**: Defaults to the US API Proxy URL. See API Proxy for other regional API Proxy URLs.
- **Client ID**: The clientID created on the Citrix Cloud Identity and Access Management website. This is required to obtain the bearer token needed for authentication to the CVAD service APIs. See Generate Client ID and Client Secret.
- **Client Secret**: The secret key created on the Citrix Cloud Identity and Access Management website. This is required to obtain the bearer token needed for authentication to the CVAD service APIs.

Permissions for the Client ID and Secret (Secure Client)

When creating the Secure Client from Identity and Access Management, the account you are logged in with when generating the Secure Client must have the following Citrix DaaS permissions:

- **Read-only Administrator, All** - To pull data from Citrix DaaS.
- **Session Administrator, All** - To perform log off and restart actions.
- **Help Desk Administrator, All** - To enable and disable maintenance mode.

The Secure Client credentials inherit the permissions of the logged in user. If the permissions of the user that was logged in when the Secure Client was created change, then those new permissions apply to the Integration, too.

Generate Client ID and Client Secret (Secure Client)

A Client ID and Client Secret (Secure Client) are required to obtain the bearer token to authenticate to and use the CVAD service APIs.

Create a Secure Client from the Citrix Cloud Identity and Access Management page and store the client ID and client Secret securely, as the Client ID and Client Secret are needed when configuring the integration.

The Secure Client Name helps to quickly identify what the Client is used for. For this Integration a name like "DaaS integration microapps" may be suitable. The name is not needed to configure the CVAD service integration.

A client ID is some numbers and letters, separated by hyphens. For example: `91132682-26af-460c-af73-18c0d2e95121`

A client Secret looks similar to this: `DTcs_w_akE6mKlberYMgtg==`

A step-by-step guide to creating a Secure Client can be followed from the CVAD migration guide here: Generate the customer ID, client ID, and secret key.

Choose an API Proxy

Citrix provides API proxies in multiple regions. Choose a proxy closest to the region that your Citrix Cloud instance resides in:

- US: `https://api-us.cloud.com/cctrustoauth2//tokens/clients`
- EU: `https://api-eu.cloud.com/cctrustoauth2//tokens/clients`
- AP-S: `https://api-ap-s.cloud.com/cctrustoauth2//tokens/clients`

The integration configuration defaults to US. Copy and paste another region's URL into the **Token URL** field if needed.

Gather configuration data to be entered

When configuring the integration you will need the following information:

1. The Customer ID of the Citrix Cloud instance
2. The Client ID and Client Secret generated when the Secure Client was created
3. The API proxy URL to enter in the Token URL field, if it differs from the default US proxy

Add the integration

Add the Citrix DaaS integration to Citrix Workspace Microapps.

This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace after Subscribing users or groups to them

Follow these steps:

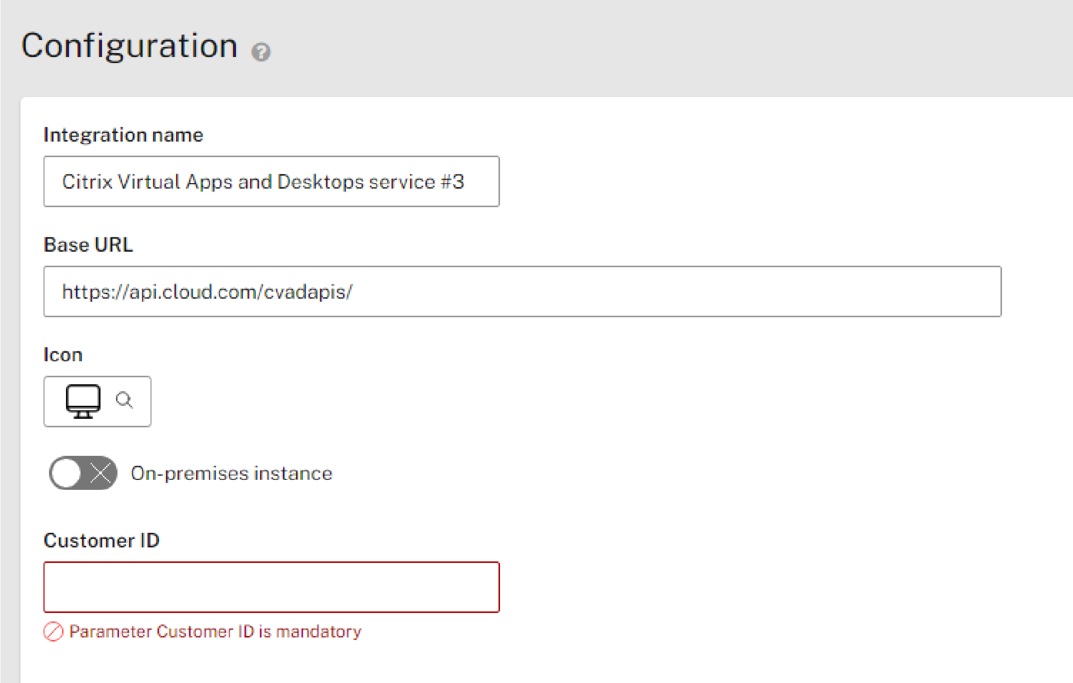
1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

2. Choose the **Citrix DaaS** tile.

3. Click **Add**

4. From the Configuration page:

    a) Enter your **Customer ID** in the Customer ID field. To find your customer ID, see Generating the customer ID.



Configuration

Integration name

Citrix Virtual Apps and Desktops service #3

Base URL

https://api.cloud.com/cvadapis/

Icon

On-premises instance

Customer ID

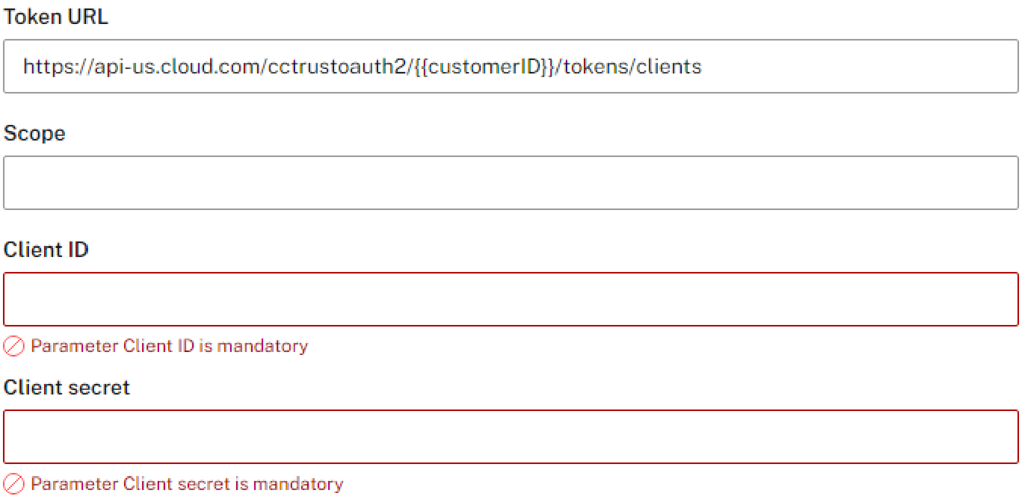
Parameter Customer ID is mandatory

b) Confirm the **Token URL** field is the desired API Proxy URL collected earlier. This value is prefilled with the US API Proxy.

c) In the Token URL field replace with your customer ID. For example, if the customer ID was *acmecorp*, the URL would look like: `https://api-us.cloud.com/cctrustoauth2/acmecorp/tokens/clients`

d) Enter your **Client ID**, generated when the Secure Client was created

e) Enter your **Client secret**. The client secret is the unique string issued with the Client ID when creating the Secure Client.



Token URL

https://api-us.cloud.com/cctrustoauth2/{{customerID}}/tokens/clients

Scope

Client ID

Parameter Client ID is mandatory

Client secret

Parameter Client secret is mandatory

5. Select **Save** to proceed.

The microapps service will now synchronize with the CVAD service APIs, and load data. For complete information about synchronization, see Synchronize data.

Subscribe Groups or Users to the Microapps

After configuring the Integration, people must be Subscribed to the microapps to see them in Workspace. Find out how to assign Subscribers

Who you subscribe to each microapp depends on your needs. However, as a general guide:

1. Subscribe all users who use Virtual Apps or Desktops to the **My Sessions** and **My Desktops** microapps
2. Subscribe CVAD administrators to the **Virtual Desktops (Admin Mode)** microapp

Use DaaS microapps

The Citrix DaaS integration comes with out-of-the-box microapps. Start with these microapps and customize them for your needs.

My Desktops: Search for your Citrix Virtual Desktops, and perform self-service actions from Citrix Workspace such as restarting.

Notification or Page	Use-case workflows
Virtual Desktop Detail page	Provides a page with desktop details, and options to **Restart Desktop** and **Force Restart Desktop**.
Virtual Desktops page	Provides a searchable list of Citrix Virtual Desktops associated with the user, with a link to the Virtual Desktop Detail page.

My Sessions: Search for your Citrix Virtual sessions, and perform self-service actions from Citrix Workspace such as logging off and disconnecting.

Notification or Page	Use-case workflows
List of Sessions page	Provides a searchable list of Virtual Desktop sessions associated with the user, with a link to the Session Detail page.

Notification or Page	Use-case workflows
Session Detail page	Provides a page with Virtual Desktop session and machine details, and options to **Log Off Session** and **Disconnect Session**.

Virtual Desktops (Admin Mode): Enables from Citrix Workspace CVAD Administrators to lookup machines, view their details, put machines into maintenance mode, and restart the machines.

Notification or Page	Use-case workflows
A Machine in a Fault State notification	When there's a new record of a machine reporting a faulty state, all subscribers receive a notification that links to the Virtual Desktop Detail page.
List of Virtual Desktops page	Provides a searchable list of Citrix Virtual Desktops with a link to the Virtual Desktop Detail page. You can search by machine or user.
Virtual Desktop Detail page	Provides a page with desktop details, and options to **Restart Desktop**, **Force Restart Desktop**, **Enable Maintenance Mode**, and **Disable Maintenance Mode**.

Upgrade your integration

If you are already using the Citrix Virtual Apps and Desktops integration, use this process to upgrade to the latest version. With this process you avoid having to resubscribe all your users.

Enhancements include:

- Site ID is obtained automatically.
- All machines are displayed - Not limited to faulty machines.
- Incremental synchronization runs faster. Synchronization does not delete Deleted Sessions. Also, synchronization does not update Machines, rather only Sessions.
- Added updates after actions that delete Sessions after logging off, updates Sessions after disconnecting, and sets restarting state for Machines.
- Added option to update information for details on Machines and Sessions so that users can ensure that what's seen is the same as in Citrix DaaS.

Follow these steps:

1. Download and add this script into the Scripting tab of your old Citrix DaaS integration.
    a) From the integration configuration screen of the integration, select **Scripting** from the left-hand navigation.
    b) Download this script: Upgrade Citrix-Virtual-Apps-and-Desktops-service
    c) Select **Upload script**. Alternatively, you can input your script directly into the text area by selecting **Edit**.
    d) Drag your script onto the import pop-up. Select **Import**.
2. Add the new Citrix DaaS integration from the catalog of Citrix-provided templates, but do not configure the integration.
3. After adding the integrations, export each microapp individually. For each microapp, select the menu next to the microapp and select **Export**. Perform this procedure for all three microapps.
4. In the old integration (that is the integration you are currently using), import the files you exported.
    a) Select the menu next to the microapp and select **Import new version**.
    b) Drag or browse your computer to add the files, each for the corresponding microapp.
    c) Enable the **Delete existing feed cards** toggle to replace the original microapp.
    d) Select **Import** at the bottom of the screen.
5. Delete all data endpoints except the scripted endpoints.
    a) Select **Edit** next to the integration.
    b) On the **Data loading** page select the menu next to an integration that is not scripted, select **Delete**, and confirm.
6. Update the integration configuration. Specifically, the **Base URL**, **Customer ID**, and **Header prefix** according to the instructions in Add the integration. Remember to select **Save** to complete the procedure

You upgraded the Citrix Virtual Apps and Desktops integration.

How the Integration works

The CVAD service integration uses the Citrix Virtual Apps and Desktops REST APIs and a Citrix Cloud Client ID and Secret (referred to as a Secure Client) to function.

Bearer token refresh requirements from the CVAD service APIs are handled by a Citrix-provided API Proxy. The API proxy uses the Client ID and Secret to refresh the bearer token automatically and uses the token to authenticate to the CVAD service APIs when microapp actions are performed or data is synchronized.

The microapps service - part of Citrix Workspace - synchronizes data from the CVAD service APIs, using the bearer token from the API Proxy.

The synchronized data is then presented in Workspace through microapps, and allows users of Workspace to perform actions on CVAD service Sessions and Machines, in addition to showing information to the user about their sessions and machines - all from Workspace.

Further reading and viewing

Citrix Tech Zone Live session covering the CVAD service integration - covering the origins of the microapps, what they do, how they work, and a demo (~16 minutes): Tools needed for users to self-service their VDI sessions

This is an embedded video. Click the link to watch the video

For more details of CVAD service API endpoints and table entities, see Citrix DaaS connector specifications.

Integrate Covid-19 Self-Certify

August 25, 2022

Deploy the Covid-19 Self-Certify integration to submit your Covid-19 self-certification status using the self-certification response from Citrix Workspace.

We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details about this microapp, see Use Covid Self Certify microapps.

Review prerequisites

The COVID Self-Certification Podio app pack is required for this integration. For more information, see Get and configure the Covid-Self Certify app in Citrix Podio.

After you set up this integration with Covid-19 Self Certify, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://api.podio.com`
- **Covid Self Certification App Id**: Enter your Podio Workspace App ID. See Collect App ID and App token.
- **Covid Self Certification View Id**: Enter your Podio Workspace App View ID. See Collect View ID.
- **Country Data App Id**: Enter your Country Data App ID available in Covid-19 self-certification. See Collect Country Data App ID.
- **Authentication method**: OAuth 2.0

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

- **Grant type flow**: Authorization Code
- **Grant type value**: authorization_code
- **Callback URL**: `https://{ yourmicroappserverurl } .us.iws.cloud.com/admin/api/gwsc/auth/serverContext`
- **Token authorization**: Request body
- **Token content type**: URL encoded form
- **Authorization URL**: `https://podio.com/oauth/authorize`
- **Token URL**: `https://podio.com/oauth/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server.
- **Client secret**: The client secret is a unique string issued when setting up the target application integration.
- **Refresh token URL**: NA
- **Scope**: NA
- **Relay state**: NONE
- **Access token parameters**: Enter the name and value parameters as per the instructions given below:
    1. **app_id**: Enter your Podio Workspace App ID. See Collect App ID and App token.
    2. **app_token**: Enter Podio Workspace App Token. See Collect App ID and App token.
    3. **grant_type**: app
- **Request rate limiting**: 15 request per minute
- **Request timeout**: 120

Create service account

The integration requires regular access to your Covid Self Certify instance, so we recommend creating a dedicated user account. This account must be an administrator or member of the Covid Self Certify Podio Workspace app.

To add multiple admin/members to the Covid Self Certify Podio Workspace, see Inviting and adding members to a Podio workspace.

Get and configure the Covid Self Certify app in Citrix Podio

The COVID Self-Certification app pack is required for this integration. The app pack includes the Covid Self Certification app and the Country Data app. Install this app pack from the Podio App Market: COVID Self-Certification app pack.

To configure the app, follow these steps:

1. Log in to your Citrix Podio instance: https://podio.com/.
2. Create a Workspace with an appropriate name.
3. Add the **Country Data** app into your workspace and insert the following four fields into the Country Data app:
    a) Region: Add a **Text** component named `Region` which stores the region name.
    b) Countries: Add a **Text** component named `Countries` which stores the countries or cities that you want to display to end users, regarding region.
    c) Guidelines: Add a **Text** component named `Guidelines` which stores the guidelines or conditions that end users need to follow.
    d) Image: Add a **Link** component named `Image` which stores any image that suits your region.
4. Add the **Covid Self Certify** app into your workspace and insert the following six fields into the Covid Self Certify app:
    a) Name: Add a **Text** component named `Name` which stores the name of the employee/user.
    b) Email: Add a **Email** component named `Email` which stores the email of the employee/user.
    c) Condition: Add a **Category** component named `Condition` and insert two category options: `Agree` and `Disagree`.
    d) CreatedOn: Add a **Date** component named `CreatedOn` which stores the agreed/disagreed date and time value.
    e) Region: Add a **Text** component named `Region` which stores the region value selected by the user/employee.
    f) Country: Add a **Text** component named `Country` which stores the country value selected by the user/employee.

Configure the OAuth client app

1. Log in to the Podio Developer site as Podio Workspace Application Admin: https://developers.podio.com.
2. Select **Generate Your API Key**.
3. Select **Get an API key now** and enter the **Application name** field and the **Domain** field as per the instructions given below.
    a) **Application name:** Enter your app name.

    b) **Domain:** Enter your Microapps instance URL for **Full domain (without protocol) of your return URL**. This section of the URL `{ yourmicroappserverurl }` is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region (us/eu/ap-s)} .iws.cloud.com`.

4. Select **Generate API key**.
5. Under **Your API keys**, copy and save the **Client ID** and **Client Secret** values for future reference.

For more information, see https://docs.citrix.com/en-us/citrix-microapps/set-up-template-integrations/integrate-podio.html#collect-client-id-and-client-secret.

Collect App ID and App token

Collect the App ID and App Token from Podio to authenticate as an app instead of a user. With App ID access, users can access only data of apps within the related Podio Workspace. You collect this ID from the Podio app for the Podio workspace. If you have multiple apps in a workspace, you can use any of the App IDs affiliated with the workspace.

Follow these steps:

1. Log in to Podio, and navigate to your Podio app.
2. Select the **Tools icon** next to the app name.
3. Under **App**, select **Developer**.
4. Copy and save the **App ID** and **Token** fields. You enter these values as **Access token parameters** when you set up the integration.

For more information, see https://docs.citrix.com/en-us/citrix-microapps/set-up-template-integrations/integrate-podio.html#collect-app-id-and-app-token.

Collect View ID

1. Log in to Podio, and navigate to your Podio app.
2. Mouse over the field specified above to see the view ID at the bottom-left of your browser. For covid-19 self-certify, place your cursor over **Agreed/Disagreed** and look for your **View ID** in the URL link.

For more information, see https://docs.citrix.com/en-us/citrix-microapps/set-up-template-integrations/integrate-podio.html#collect-view-id.

Collect Country Data App ID

1. Log in to Podio, and navigate to your Podio app.
2. Select the **Tools** icon next to the app name.
3. Under **Country Data App**, select **Developer**.

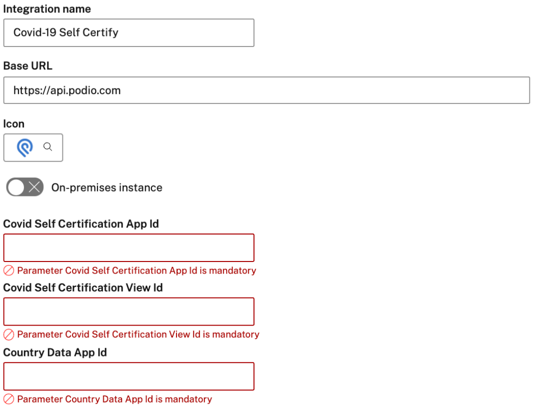
4. Copy and save the **App ID** and **Token** fields. You enter these values as **Access token parameters** when you set up the integration.

Add the integration to Citrix Workspace Microapps

Add the Covid Self Certify integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the **Covid Self Certify** tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**: `https://api.podio.com`
    - Enter your **Podio App Id**.
    - Enter your Podio Workspace **App View Id**.



Integration name

Covid-19 Self Certify

Base URL

https://api.podio.com

Icon

On-premises instance

Covid Self Certification App Id

Parameter Covid Self Certification App Id is mandatory

Covid Self Certification View Id

Parameter Covid Self Certification View Id is mandatory

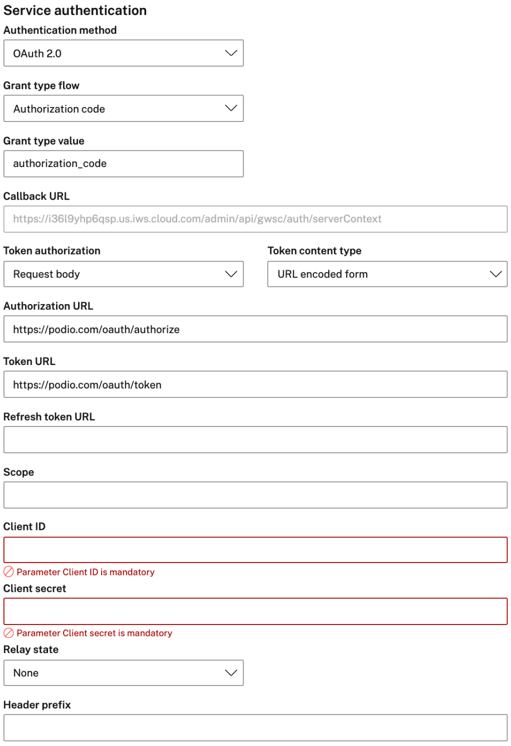
Country Data App Id

Parameter Country Data App Id is mandatory

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
    a) Select **Authorization Code** from the **Grant type flow** menu.
    b) Enter **authorization_code** in the **Grant type values** field.
    c) Enter `https://{ yourmicroappserverurl } .us.iws.cloud.com/admin/api/gwsc/auth/serverContext` in the **Callback URL** field.

d) Select **Request body** from the **Token authorization** menu.
e) Select **URL encoded form** from the **Token content type** menu.
f) Enter `https://podio.com/oauth/authorize` in the **Authorization URL** field.
g) Confirm the **Token URL** field. This value is prefilled: `https://podio.com/oauth/token`.
h) Enter your **Client ID**: The client ID is the string representing client registration information unique to the authorization server. You collect this and the Client secret when you generated an API key. See Configure the OAuth client app.
i) Enter your **Client secret**: The client secret is a unique string issued when setting up the target application integration. You collect this and the Client ID when you generated an API key. See Configure the OAuth client app.

6. **Refresh token URL**: NA
7. **Scope**: NA
8. **Relay state**: NONE



Service authentication

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/auth/serverContext

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://podio.com/oauth/authorize

Token URL: https://podio.com/oauth/token

Refresh token URL:

Scope:

Client ID:

Parameter Client ID is mandatory

Client secret:

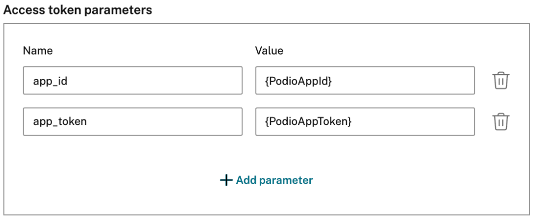
Parameter Client secret is mandatory

Relay state: None

Header prefix:

9. Add the following **Access token parameters**. For more information, see Collect App ID and App token.

   a) **app_id**: Enter Podio Workspace App ID
   b) **app_token**: Enter Podio Workspace App Token
   c) **grant_type**: app



Access token parameters

Name	Value
app_id	{PodioAppId}
app_token	{PodioAppToken}

1. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Google login screen.
   a) Enter your **Service Account username** and your **Service Account password** and select **Log in**.
   b) Select **Accept**.
2. Enter *15 requests per minute* in the **Request rate limiting** field.
3. Enter `120` in the **Request timeout** field.
4. Select **Save** to proceed.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

> **Note:**
>
> We recommend triggering incremental synchronization once every hour and full synchronization two or three times a day.

For more details of API endpoints and table entities, see Covid-19 Self Certify connector specifications.

Use Covid Self Certify microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Self Certify: Submit your daily covid-19 self-certification status response from Citrix Workspace.

Notification or Page	Use-case workflows
Covid Self Certify - Agree: Item Change notification	Get notified for all your agreed covid-19 self-certification response submission.
Covid Self Certify - Disagree: Item Change notification	Get notified for all your disagreed covid-19 self-certification response submission.
New Self Certify - Agree: Item New notification	Get notified for all your agreed covid-19 self-certification response submission (New user).

Notification or Page	Use-case workflows
New Self Certify - Disagree: Item New notification	Get notified for all your disagreed covid-19 self-certification response submission (New user).
Select Region: Item page	Choose their working location.
Self Certification Guidelines page	Submit the response for daily self-certification.
Self Certification Agreed: Item page	View agreed response details for daily self-certification.
Self Certification Confirmed page	View agreed response details for daily self-certification from their notification feed.
Self Certification Denied page	View disagreed response details for daily self-certification.
Self Certification Disagree page	View disagreed response details for daily self-certification from their notification feed.

Integrate Employee Survey App

November 8, 2021

Deploy the Employee Survey App to send a new survey form and manage existing survey forms from Citrix Workspace. The following workflows are addressed with the microapps:

- The **Manage Survey** microapp is for Citrix Podio admins to maintain the existing survey apps in Citrix Workspace.
- The **Survey Form** microapp sends notifications for all surveys scheduled in Citrix Podio to all the subscribers and receives user feedback in Citrix Workspace.

Regardless of when you roll out your technology solution, it's important for you to understand your end users' experience so that you know if your solution is meeting their needs while driving productivity through an optimal digital workspace experience. The employee survey integration allows you to easily receive feedback from your end users at any time you choose. Not only does the survey give you the ability to hear from users and provide required support, the input also assists you in staying informed and managing their experience accordingly. The template includes pre-built, customizable surveys based on Employee and End User Experience best practices. You can easily add more surveys to gather feedback on any aspect of Employee Experience. To learn more about End User Experience, check out the End User Experience Kit on the Citrix Success Center.

We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details about the microapps, see Use Employee Survey App microapps.

Review prerequisites

After you set up this integration with the Survey App, you need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://api.podio.com`
- **Survey Admin App Id**: Enter the "My Surveys" app id. See Collect App ID and App token.
- **Callback URL**: `https://{ yourmicroappserverurl } .us.iws.cloud.com/admin/api/gwsc/auth/serverContext`
- **Authorization URL**: `https://podio.com/oauth/authorize`
- **Token URL**: `https://podio.com/oauth/token`
- **Client ID**: The Client ID is the string representing client registration information unique to the authorization server.
- **Client secret**: The Client secret is a unique string issued when setting up the target application integration.
- **Access token parameters**: Enter the name and value parameters following the instructions given below:
  - **app_id**: Enter your Podio Workspace App ID. See Collect App ID and App token.
  - **app_token**: Enter Podio Workspace App Token. See Collect App ID and App token.
  - **grant_type**: app

> **Note**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Create service account

The integration requires regular access to your Survey App instance. We recommend creating a dedicated user account. This account must have full administrator permissions. For more information, see Podio API documentation.

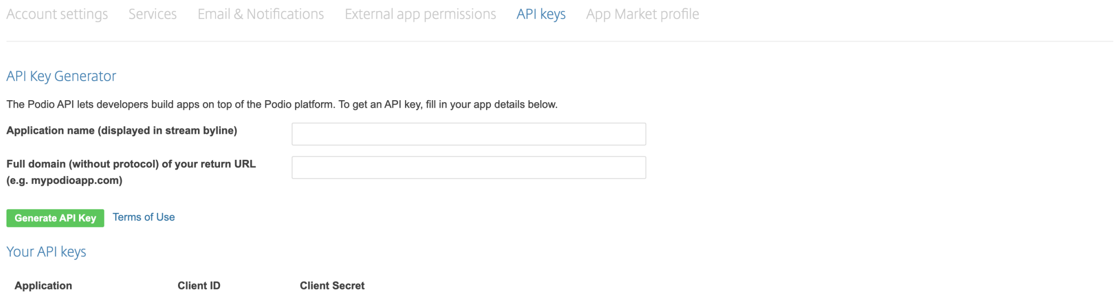
Enable APIs

The number of API requests that can be made to specific resources is limited. We recommend the following: API limitation documentation.

Configure the OAuth Client

Collect the Client ID and Client Secret in Podio to enter in to the **Microapps integration configuration** screen for authentication.

1. Log in to the Podio Developer site as Podio Workspace Application Admin: Podio
2. Select **Generate Your API Key**.
3. Select **Get an API key now**.
4. Enter a name for **Application name**.
5. Enter your Microapps instance URL for **Full domain (without protocol) of your return URL**. This section of the URL `{ yourmicroappserverurl }` is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region(us/eu/ap-s)} .iws.cloud.com`.
6. Select **Generate API Key**.
7. Under **Your API keys**, copy and save the **Client ID** and **Client Secret** values for the application you just added. You enter these values when you set up the integration.



API Key Generator

The Podio API lets developers build apps on top of the Podio platform. To get an API key, fill in your app details below.

Application name (displayed in stream byline)

Full domain (without protocol) of your return URL (e.g. mypodioapp.com)

Your API keys

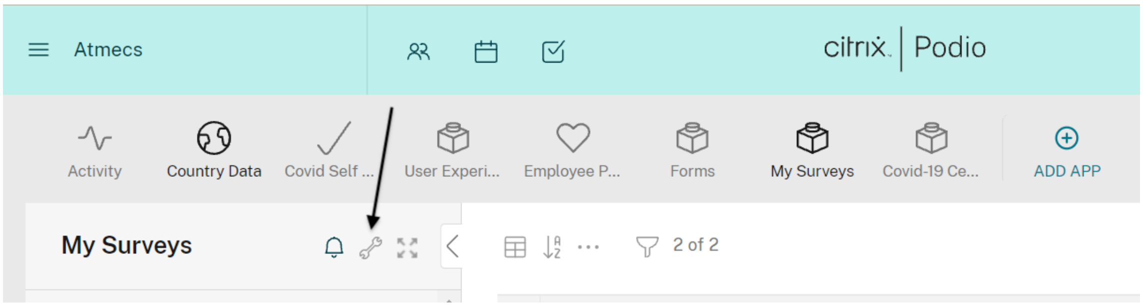
Application	Client ID	Client Secret

For more information, see Collect Client ID and Client secret.

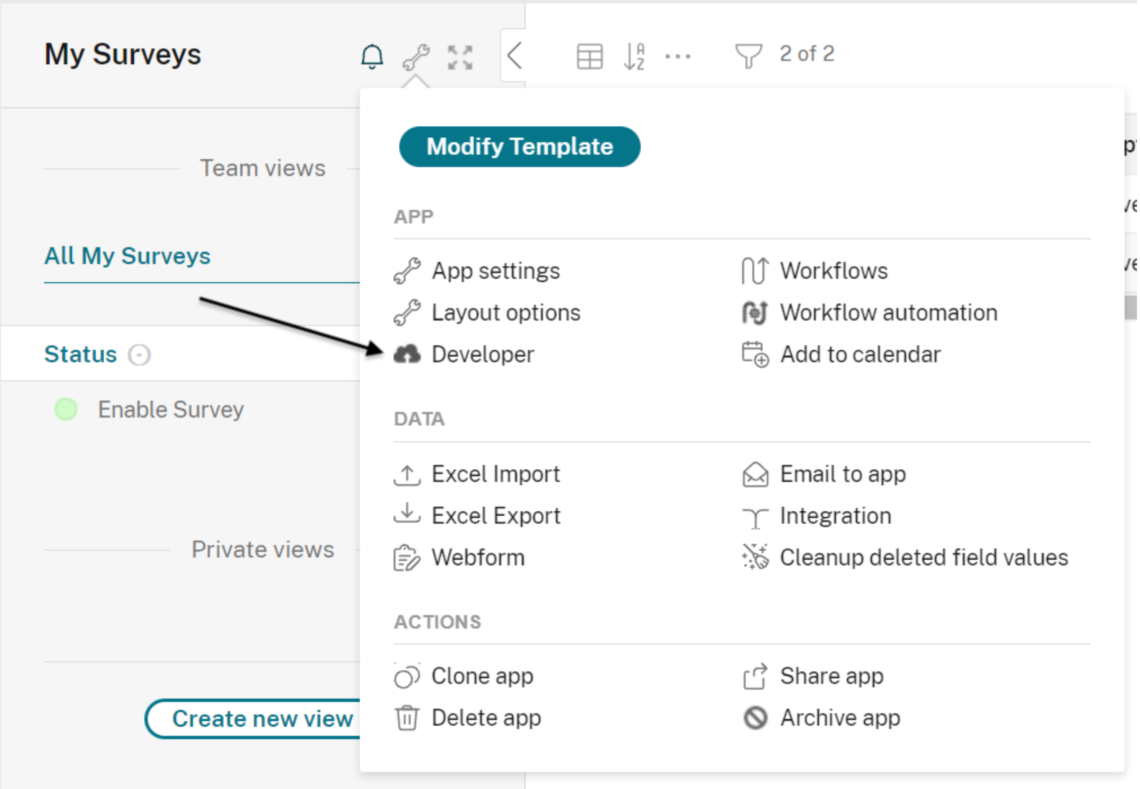
Collect App ID and App token

Collect the App ID and App Token from Podio to authenticate as an app instead of a user. With App ID access, users can access only data of apps within the related Podio Workspace. You collect this ID from the Podio app for the Podio workspace. If you have multiple apps in a workspace, you can use any of the App IDs affiliated with the workspace.

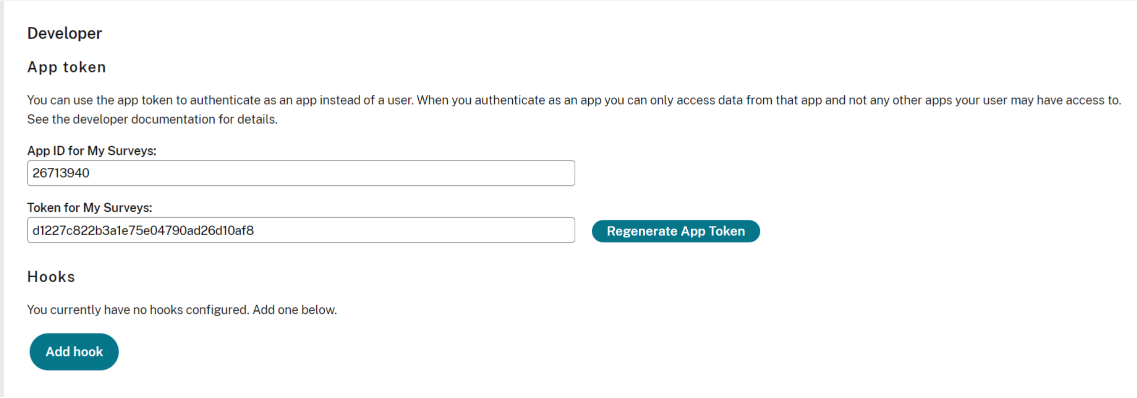
1. Log in to Podio, and navigate to your Podio app.
2. Select the **Tools icon** next to the app name. We've shown the steps for the Broadcast app as an example in the screenshots.



3. Under **APP**, select **Developer**.



4. Copy and save the **App ID** and **Token** fields. You enter these values as **Access token parameter** when you set up the integration.



Developer

App token

You can use the app token to authenticate as an app instead of a user. When you authenticate as an app you can only access data from that app and not any other apps your user may have access to. See the developer documentation for details.

App ID for My Surveys:
26713940

Token for My Surveys:
d1227c822b3a1e75e04790ad26d10af8

Hooks

You currently have no hooks configured. Add one below.

You can also view the fields and sample JSON values under **App fields** on this page. For more information about working with app entries in Podio, see Working with items.

How to send New Survey App notifications

Use the Employee Survey app to schedule multiple webform/survey feedback forms.

> **Note**
>
> We recommended adding any self-identification fields within the survey form, such as **Email** or **Name**, as required fields for the purpose of identification.

Follow these steps to schedule a survey in Citrix Workspace.

1. Log in to your Citrix Podio instance, and go to the Podio App Market and get the Citrix Workspace Employee Surveys App Pack: Podio App Market - Citrix Workspace Employee Surveys
2. Select the **My Survey App** from under **Apps in this pack**.
3. Select **Add Survey** and enter a **Survey Name** that you want to send and get feedback on. Select a **Launch Date** on which survey notification to trigger.
4. Select **Enable_Survey** as **Status** to generate notification and **Disable_Survey** to disable the same instantly from Citrix Workspace.
5. Enter the notification body content in **Survey Description**. This content is shown in the feed card for subscribers.
6. Enter the survey form **Response App Id**. This value is the ID that the admin wants to send to users for notification.
7. Select the **Expire After** date to disable notifications to subscribers automatically on the selected date.
8. Go to Podio created webform apps that the admin wants to schedule and get feedback from users, and select **Settings** and choose the **Webforms** option.
9. Under the **Allowed Domain** section, add the WSI workspace domain in the survey apps webform configuration. For example, `{ yourmicroappserverurl } .cloud.com`.
10. Select **Save** and then **Share**.

Add the integration to Citrix Workspace Microapps

Add the Podio integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

2. Choose the Podio tile.

3. Enter an **Integration name** for the integration.

4. Enter **Connector parameters**.

   - Enter the instance **Base URL**: `https://api.podio.com`
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.

5. Enter the **Survey Admin App Id**. See Collect App ID and App token.

6. Enter the **Service Authentication** and **Connector parameters** that you collected in the previous procedures.

7. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

   a) Select **Authorization Code** from the **Grant type flow** menu.
   b) Enter **authorization_code** in the **Grant type value** field.
   c) The **Callback URL** is prefilled: `https://{ yourmicroappserverurl } .us.iws.cloud.com/admin/api/gwsc/auth/serverContext`.
   d) Select **Request body** from the **Token authorization** menu.
   e) Select **URL encoded form** from the **Token content type** menu.
   f) The **Authorization URL** is prefilled: `https://podio.com/oauth/authorize`.
   g) The **Token URL** is prefilled: `https://podio.com/oauth/token`.
   h) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collected this ID and the Client Secret in the procedure Configure the OAuth Client.
   i) Enter your **Client secret**. The client secret is a unique string issued with the Client ID in the procedure Configure the OAuth Client.

8. Enter **Access token parameters** values. For more information, see Collect App ID and App token:

   - Enter the **App ID** that you collected next to **app_id**.
   - Enter the **App token** that you collected next to **app_token**.
   - Enter *app* for **grant_type**.

9. The **Request rate limiting** toggle is enabled and set to *1000 request per minute*.

10. Enter *120* in the **Request timeout** field.

11. Select **Save** to proceed.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Employee Survey App connector specifications.

Use Employee Survey App microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Survey Form: Provides notifications for new surveys and allows users to view and submit the survey forms.

Notification or Page	Use-case workflows
New Survey notification	Allows users to view all new forms initiated through the Survey Admin app into Citrix Workspace.
Survey Form page	Allows the users to complete and submit a survey from Citrix Workspace.

Manage Survey: Allows survey app admin to manage the scheduled survey forms from Citrix Workspace. This microapp is only for Employee Survey App admin members.

> **Note**
>
> We recommended maintaining a limited number of survey records in the **My Surveys** Podio admin app.

Notification or Page	Use-case workflows
My Survey page	Lists all the scheduled surveys forms created by an admin.
Manage Survey page	Allows admins to manage the scheduled survey forms such as change in status, launch date, expire after date, and so forth.

Integrate Adobe Sign

October 18, 2021

Deploy the AdobeSign integration to quickly and securely make every agreement and approval digital.

With Citrix Adobe Sign Integration users can perform the following actions:

- Send a template and upload agreements to recipients for signature.
- List of Adobe Sign agreements waiting for a user's signature. Users can sign the agreement from Citrix Workspace.
- List sent Adobe Sign agreements for signature. Users can also cancel a sent agreement.
- List completed Adobe Sign agreements with details. Users can also share received agreements.
- List canceled Adobe Sign agreements with details.
- Manage and edit template agreements.

We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details about the microapps, see Use Adobe Sign microapps.

Review prerequisites

After you set up this integration with Adobe Sign, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://api.in1.adobesign.com/`
- **Adobe Sign Group ID**: This Group id represents your user group id for your admin account.
- **Agreements per user**: This is the number of agreements to store per user in the cache. The maximum value is 100. We recommend 15–30 agreements per user.
- **Authorization URL**: `https://{ HostName } .com/public/oauth`
- **Token URL**: `https://{ HostName } .com/oauth/token`
- **Refresh token URL**: `https://{ HostName } .com/oauth/refresh`
- **Scope**: *user_read:self user_write:self user_login:self agreement_read:self agreement_write:self agreement_send:self application_write:self library_write:self*
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server.
- **Client secret**: The client secret is a unique string issued when setting up the target application integration.
- **Relay state**: NONE
- **Request rate limiting**: 500 request per second
- **Request timeout**: 120

> **Note**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with your configured microapp.

Create service account

The integration requires regular access to your Adobe Sign instance. We recommend creating a dedicated user account. This account must be an Adobe Sign administrator account with Full administrator privileges.

To set up a new account, see Create a free Adobe Sign account.

Enable APIs

There is no API limitation mentioned in Adobe Sign. Adobe Sign transaction limits are mentioned in the following link: https://helpx.adobe.com/in/sign/using/transaction-limits.html. Plans can be chosen according to your requirements.

Configure the OAuth client app

Configure the OAuth client to write data back through the integration. For more information about working with Adobe Sign, see Get started guide (Adobe Sign).

1. Log in to the Adobe Sign Admin account: https://secure.echosign.com/public/login
2. Select **Account** and select **API Applications** under the section **Adobe Sign APIs**.
3. Select **Create** and create two new **Apps**.
4. For the first app, paste the redirect URLs in the **Redirect Uri** field.
   - `https://{ yourmicroappserverurl } .us.iws.cloud.com/app/api/auth/serviceAction/callback`
   - `https://{ yourmicroappserverurl } .us.iws.cloud.com/admin/api/gwsc/auth/serverContext`
5. Select and choose the **OAuth Configure** option and enable all the scopes with modifier as **Group**.
6. Repeat the same for the second OAuth app.
7. Collect the **Client Id** and **Client secret** details from both OAuth apps and save them for future use. You use these values for Service Authentication and Service Action Authentication while configuring the integration.

Configure group and get Group Id

Groups allow you to have separate subsets of users that have access to different Library Documents and different settings. For more information, see Create and manage groups (Adobe Sign.

1. Log in to the Adobe Sign Admin account: https://secure.echosign.com/public/login
2. Select **Account**, and select **Groups** section.
3. Select **Create New Group** to add a new group.
4. Enter an appropriate name for the group.
5. Select and choose the **Group Setting** option to find the **Group Id** on the configuration page.
6. Add or import users to the groups.

Add the integration to Citrix Workspace Microapps

Add the Adobe Sign integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions, which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

2. Choose the **Adobe Sign** tile.

3. Enter a name for the integration.

4. Enter **Connector parameters**:

   - Enter Instance Base URL: `https://api.in1.adobesign.com/`
   - Select an Icon for the integration from the Icon Library, or leave this as the default icon.
   - Enter your **Adobe Sign Group ID**. For more information, see: Configure group and get Group Id.
   - Enter the **Agreements per user**. This is the number of agreements to store per user in the cache. The maximum value is 100. We recommend 15–30 agreements per user.

5. Under Service authentication, select OAuth 2.0 from the Authentication method menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with your configured microapp.

   a) Select **Authorization code** from the Grant type menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the Callback URL, which you use when registering your application.

b) Select **Request body** from the **Token authorization** menu and **Token content Type** as **URL Encoded Form**.
c) Enter the **Refresh token URL**: `https://{ HostName } /oauth/refresh`.
d) Ensure that the following is entered for **Scope**: *user_read:self user_write:self user_login:self agreement_read:self agreement_write:self agreement_send:self application_write:self library_write:self*
e) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. Add the Callback URL you see on the integration configuration page.
f) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

6. Under Service Action Authentication, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the Callback URL, which you use when registering your application.
   c) Select **Request body** from the **Token authorization** menu.
   d) The **Authorization URL** is prefilled.
   e) The **Token URL** is prefilled.
   f) Ensure that the following is entered for **Scope**: *user_read:self user_write:self user_login:self agreement_read:self agreement_write:self agreement_send:self application_write:self library_write:self library_read:self*
   g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth client. Add the Callback URL you see on the integration configuration page.
   h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

7. Select **Save** to proceed.

8. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Google login screen.

a) Enter your Service Account **User name** and password and select **Log in**.
b) Select **Accept**.

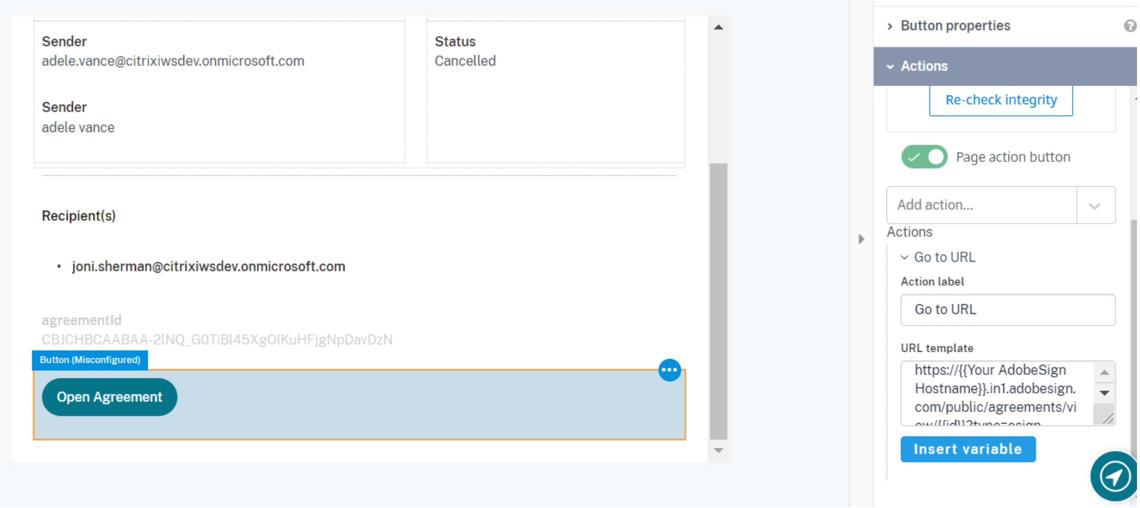
Add redirect action to Open Agreement buttons

Configure this Go to URL action with your host name for the following buttons to enable the Open Agreement button capability for these microapps. This Go to URL configuration redirects users to the specific envelope detail page in DocuSign. Navigate to each button listed below to perform this action:

Microapp	Page	Button
In progress Agreements	In progress Agreements Details	**Open Agreement**
Canceled Agreements	Canceled Agreements Details	**Open Agreement**
Canceled Agreements	Agreement Canceled	**Open Agreement**
Completed Agreements	Completed Agreement Details	**Open Agreement**
Completed Agreements	Agreement Completed	**Open Agreement**

Follow these steps:

1. Navigate to the first button listed above, select the button, and select the **Actions** tab on the right.
2. In the **Add action** field, select **Go To URL**.
3. In the **URL Template** field, replace the `{ Your Adobe Sign Host }` portion of the URL with your host name. Find your Adobe Sign host name in your Adobe Sign instance URL following this model: `https://{ hostname } .in1.adobesign.com/public/login`. For example, `https://citrixwsi.in1.adobesign.com/public/login`.
4. Now repeat this procedure for the other buttons, replacing the `{ Your Abobe Sign Host }` portion of the URL with your host name.



Sender
adele.vance@citrixiwsdev.onmicrosoft.com

Sender
adele vance

Status
Cancelled

Recipient(s)

- joni.sherman@citrixiwsdev.onmicrosoft.com

agreementId
CBJCHBCAABAA-2lNQ_G0TiBl45XgOIKuHFjgNpDavDzN

Action label: Go to URL

URL template: https://{{Your AdobeSign Hostname}}.in1.adobesign.com/public/agreements/vi

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Adobe Sign connector specifications.

Use Adobe Sign microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Send Agreements: Send the template and upload an agreement to recipients for signature.

Notification or Page	Use-case workflows
My Template page	Provide a list of template agreements available for the user with options to **Create** and send to users for agreement using the upload option.
Send Agreement page	Allows user to send an agreement to recipients using the upload option.
Send Template page	Allows user to send template agreements to recipients.

My Received Agreements: Provides a list of Adobe Sign agreements waiting for a user's signature. A user can sign the agreement form

Notification or Page	Use-case workflows
New Agreement Received notification	Recipients are notified for all new agreements created.
New Agreement Changed notification	Recipients are notified for all new agreements created with an **Order By** option.
Agreement Reminders notification	Recipients are notified of unsigned agreements that are pending for a week.
My Received Agreement page	Provides a personalized list of all received agreements.
My Received Agreement Details page	Details of a received agreement are shown with options to **Sign Agreement** and **Send To Other Recipient**.
Agreement Received page	The details of a received agreement are shown with options to **Sign Agreement** and **Send To Other Recipient**.
Share Agreement page	Allows recipients to share received agreements with other users.
Sign Agreement page	Allows recipients to sign and complete agreements.

In progress Agreements: List of in-progress Adobe Sign agreements for signature. User can also cancel the sent Agreement.

Notification or Page	Use-case workflows
My Sent Agreement page	Provides a personalized list of sent agreements.
In progress Agreement Details page	Allows users to view detailed information of selected sent agreement with options to **Cancel Agreement** and **Open Agreement**.

Completed Agreements: List of completed Adobe Sign agreements and details. User can also Share the received Agreements.

Notification or Page	Use-case workflows
Signed Agreement notification	Sender of the agreement is notified about signed agreements.
Completed Agreements page	Provides a personalized list of completed agreement.
Completed Agreement Details page	Allows users to view detailed information of a selected completed agreement with options to **Send To Other Recipient** and **Open Agreement** from the action page.
Agreement Completed page	Allows users to view detailed information of a selected completed agreement with options to **Send To Other Recipient** and **Open Agreement**. from the notification feed.
Share Agreement page	Allows recipients to share the received agreements to other users.

Canceled Agreements: List of Canceled Adobe Sign agreements and details.

Notification or Page	Use-case workflows
Canceled Agreement notification	Recipients of the agreement are notified for Canceled agreements.
Canceled Agreement page	Provides a personalized list of canceled agreements.
Canceled Agreement Details page	Allows users to view detailed information of a selected canceled agreement with the option to **Open Agreement**.
Agreement Canceled page	Allows users to view detailed information of a selected canceled agreement with the option to **Open Agreement**.

Manage Agreements: Manage and edit template agreements.

Notification or Page	Use-case workflows
My Templates page	Provides a personalized list of templates.

Notification or Page	Use-case workflows
Manage Template page	Allows users to edit selected templates in Citrix Workspace.

Integrate SAP Ariba

June 25, 2021

Integrate with SAP Ariba to review and approve requisition requests without requiring any additional logins.

> **Note:**
>
> We provide two SAP Ariba integration templates for your use. We recommend using the newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

For a comprehensive list of out-of-the-box SAP Ariba microapps, see Use Ariba microapps.

Review prerequisites

You need these values to add the HTTP integration in Citrix Workspace Microapps:

- **Base URL**: `https://openapi.ariba.com/api/approval/v1/prod`
- **Token URL**: `https://api.ariba.com/v2/oauth/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server, called OAuth Client ID in SAP. See Collect the OAuth Client ID and Application Key.
- **Client Secret**: The client secret is a unique string issued when setting up the target application integration, issued by SAP.
- **Ariba APIkey**: Referred to as the **Application key**, which replaces a variable in the scripting process. See Collect the OAuth Client ID and Application Key.
- **Ariba Realm**: Replaces a variable in the scripting process. To find this, contact your SAP admin to confirm your **Realm ID**.
- **Ariba lastChangeId**: Specifies the last change received in the previous response. The response includes all changes since this one. For example, use *1* if you want to load all changes from the beginning.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

- Your organization must be in the United States of America or Europe.
- You must have a current license for an SAP Ariba Solution or an Ariba Network Solution component.
- You must have an SAP Ariba Open APIs Administrator Account. Your organization's SAP Ariba Administrator can request access to the SAP Ariba Open APIs Developer Portal from one of the following links:
  - United States of America: https://developer.ariba.com/api/
  - Europe: https://eu.developer.ariba.com/api/
- Configure Citrix Gateway to support single sign-on for SAP Ariba so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Ariba single sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Set up the SAP Ariba integration

1. Log in to https://developer.ariba.com/api/ with an administrator account.
2. On the welcome page, select **Create application**.
3. To create a new application, enter an **Application Name** and **Description**, and select **Submit**.

   Your application is sent to the SAP Ariba Open APIs Team for approval.

Collect the OAuth Client ID and Application Key

After your application is approved (described in section above), you receive an email with the OAuth secret token. The SAP Ariba Open APIs Team sets up and enables the back end with your Ariba Realm ID.

> **Note:**
>
> You cannot proceed until you receive the approval email.

1. Log in and go to the **Manage Applications** page.
2. Select the application your created.
3. Copy the **OAuth Client ID** and **Application key**.
4. Save them in a secure place for later use when you add the integration to Citrix Workspace Microapps.

Add the integration to Citrix Microapps

Follow these steps to set up the SAP Ariba HTTP integration. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the SAP Ariba tile under **Integrations**.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**: `https://openapi.ariba.com/api/approval/v1/prod`
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.
5. Enter the **Ariba APIkey**. This application key replaces a variable in the scripting process. See Collect the OAuth Client ID and Application Key.
6. Enter the **Ariba Realm**. The Ariba realm Id replaces a variable in the scripting process. To find this, contact your SAP admin to confirm your **Realm ID**.
7. Enter the **Ariba lastChangeId**. Specifies the last change received in the previous response. The response includes all changes since this one. For example, use *1* if you want to load all changes from the beginning.
8. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
   a) Select **Client Credentials** from the **Grant type flow** menu.
   b) Enter **client_credentials** for **Grant type value**.
   c) Select **Authorization header** from the **Token authorization** menu.
   d) Select **URL encoded form** from the **Token content type** menu.
   e) Enter the **Token URL**: `https://api.ariba.com/v2/oauth/token`
   f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configure the OAuth server. See Collect the OAuth Client ID and Application Key.

   g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration. See Collect the OAuth Client ID and Application Key.

9. Enable the **Request rate limiting** toggle. Enter *3* for **Number of requests** and *1 second* for **Time interval**.
10. In the **Request Timeout** field, enter *120*.
11. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
12. Select **Save** to proceed.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Ariba connector specifications.

Use Ariba microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our SAP Ariba integration comes with the following preconfigured out-of-the-box microapps:

Requisition Approval: View requisitions with details that are pending approval, and an action to approve.

Notification or Page	Use-case workflows
Pending Approval notification	When a new purchase requisition is awaiting approval, the approver receives a notification.
Pending Next Approval notification	When a new purchase requisition is awaiting next level approval, the next approver receives a notification.
Requisition Created notification	When a new requisition is created, the requisition submitter receives a notification with details.
Requisition Changed notification	When the status of a new purchase requisition is changed, the requisition submitter receives a notification.
Approval Request page	Provides a searchable list of requests awaiting approval and a link to more details.
Requisition Detail Approver page	Provides a detailed view of a requisition with actions to **Approve** and **Deny** the request.

Notification or Page	Use-case workflows
Requisition Detail Requestor page	Provides a detailed view of a requisition to the requester.

Add the Legacy SAP Ariba integration

Add the SAP Ariba integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace. After you set up this integration in SAP Ariba, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- API URL
- OAuth Token URL
- OAuth Client ID
- OAuth Client Secret
- Application Key
- Realm ID

Follow these steps:

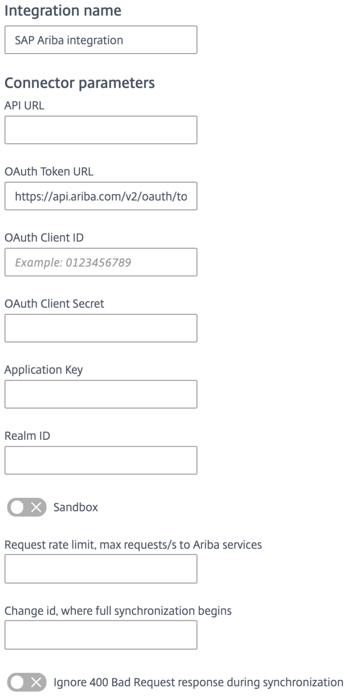
1. From the Microapps overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

3. Choose the Ariba tile.

4. Enter a name for the integration.



Integration name

SAP Ariba integration

Connector parameters

API URL

OAuth Token URL

https://api.ariba.com/v2/oauth/to

OAuth Client ID

Example: 0123456789

OAuth Client Secret

Application Key

Realm ID

Sandbox

Request rate limit, max requests/s to Ariba services

Change id, where full synchronization begins

Ignore 400 Bad Request response during synchronization

5. Enter the **Connector parameters** that you collected in the previous procedures.
   - Enter your **API URL** and **OAuth Token URL** credentials for your target systems service authentication.
   - Enter your **OAuth Client ID** and **Client Secret**.
   - Enter your **Application Key** and **Realm ID**.
   - Toggle **Sandbox** if you require your data to load into a sandbox environment.
   - Enter a value for **Request rate limit, max requests/s to Ariba services**. This field is mandatory and determines the number of calls per second. Speak with your Ariba representative about the limits for your instance to configure the value properly. We recommend 10 calls/second or less as an initial safe rate if the rate limit is not known. Zero can be used to fully disable all limitations. However, too high of a request rate can result in access denial.
   - Select a value for **Change id, where full synchronization begins**. When full synchronization starts limiting the amount of data loading, older changes are skipped during full synchronization. Leave this value empty to load everything.
   - Optional) Enable **Ignore 400 Bad Request response during synchronization** toggle to recover from 400 Bad Request response errors. If any Ariba records are deleted during synchronization, the Ariba API returns an error as the data requested does not exist anymore. A warning is generated in the log.
6. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Ariba connector specifications.

Legacy SAP Ariba microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our Legacy SAP Ariba integration comes with the following preconfigured out-of-the-box microapps:

Requisition Approval: View requisitions with details that are pending approval, and an action to approve.

Notification or Page	Use-case workflows
Pending Approval (change) notification	When details of a purchase requisition awaiting approval are changed, the approver receives a notification.
Pending Approval (new) notification	When a new purchase requisition is awaiting approval, the approver receives a notification.
Requisition Change notification	When the status of a new purchase requisition is changed, the user receives a notification.
Requisition Created notification	When a new purchase requisition is created (meaning submitted or being composed), the user receives a notification.
Approval Requests page	Provides a personalized list of approval requests and a link to more details.
Requisition Detail page	Provides a detailed view of a requisition and an actionable approval button.

Integrate Blackboard Learn

May 17, 2021

Deploy the Blackboard Learn integration to register for a new course and view the course and its related details as a student, and as an instructor to create a course announcement and view the course members and grades.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of out-of-the-box microapps for Blackboard Learn, see Use Blackboard Learn microapps.

Review prerequisites

You need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://{ host_name } .com`
- **Authorization URL**: `https://{ host_name } .com/learn/api/public/v1/oauth2/authorizationcode`
- **Token URL**: `https://{ host_name } .com/learn/api/public/v1/oauth2/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server. You collect this as **Application Key** when you configure the OAuth server.
- **Secret**: The client secret is a unique string issued when setting up the target application integration. You collect this as **Secret** when you configure the OAuth client.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service https://docs.citrix.com/en-us/citrix-gateway-service/.

Create user account

The integration requires regular access to your Blackboard instance. We recommend creating a dedicated user account: **Blackboard Partner account** https://docs.blackboard.com/partners/become-a-partner.

The following permissions are required: Full administrator privileges.

API access

Blackboard APIs are enabled by default, we need either Blackboard Developer AMI or Blackboard Partner Account to access the APIs. The number of API requests that can be made to specific resources is limited. We recommend reviewing the following information: https://docs.blackboard.com/learn/rest/admin/groups-quotas-rates

> **Important**
>
> The pagination limit is set to 100. Administrators can extend this limit up to 200 based on APIs.
>
> Blackboard Learn developer server supports up to 10000 API calls/Site/24hours.
>
> Blackboard Learn production server supports up to 75000 API calls/Site/24hours.
>
> Due to limit in API call, incremental synchronization is setup to retain only **Course Memberships** and **Course Announcements**. Remaining endpoints will be triggered as part of full synchronization.

Configure OAuth server

Configure the OAuth server to read data through the Blackboard Learn integration.

1. Sign in to https://developer.blackboard.com/.
2. Select **My Apps** and select the **+** icon to Create a New App. Alternatively, navigate to: `https://developer.blackboard.com/portal/applications/create`.
3. Complete the required fields and select **Register application and generate API key**.
4. Copy and save the **Application ID**, **Application Key** and **Secret** shown on the screen. Use these details for **Service Authentication** while configuring the integration.
5. Log in to the Blackboard Learn application as an Administrator.
6. Navigate to **System Admin**.
7. Select **REST API Integrations** under the **Integrations** table.
8. Select **Create Integration**.
9. Complete the required fields:

   a) Paste the **Application ID** that you selected in step 4 above.
   b) Enter your administrator user name in the **Learn User** field.
   c) Select **Yes** for **End User Access** and **Authorized To Act As User**.

Configure OAuth client

Configure the OAuth client to writing back data through the Blackboard Learn integration.

1. Sign in to https://developer.blackboard.com/.
2. Select **My Apps** and select the **+** icon to Create a New App. Alternatively, navigate to: `https://developer.blackboard.com/portal/applications/create`.
3. Complete the required fields and select **Register application and generate API key**.
4. Copy and save the **Application ID**, **Application Key** and **Secret** shown on the screen. Use these details for **Service Action Authentication** while configuring the integration.
5. Log in to the Blackboard Learn application as an Administrator.
6. Navigate to **System Admin**.
7. Select **REST API Integrations** under the **Integrations** table.
8. Select **Create Integration**.
9. Complete the required fields:
    a) Paste the **Application ID** that you selected in step 4 above.
    b) Enter your administrator user name in the **Learn User** field.
    c) Select **Yes** for **End User Access** and **Authorized To Act As User**.

Add the integration to Citrix Workspace Microapps

Add the Blackboard Learn integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions that are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration** and **Add a new integration from Citrix-provided templates**.
2. Choose the Blackboard Learn tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**:
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.
5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that

these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

a) Select **Client credentials** from the **Grant type** menu.
b) Select **Authorization Header** from the **Token authorization** menu.
c) The **Token URL** is prefilled: `https://{ host_name } .com/learn/api/public/v1/oauth2/token`
d) Ensure the following is entered for Scope: *read*
e) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this as **Application Key** when you configure the OAuth server.
f) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration. You collect this as **Secret** when you configure the OAuth server.

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication** in Actions toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization Header** from the **Token authorization** menu.
   c) The **Authorization URL** is prefilled: `https://{ host_name } .com/learn/api/public/v1/oauth2/authorizationcode`
   d) The **Token URL** is prefilled: `https://{ host_name } .com/learn/api/public/v1/oauth2/token`
   e) Ensure the following is entered for Scope: *write*
   f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this as **Application Key** when you configure the OAuth client.
   g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration. You collect this as **Secret** when you configure the OAuth client.

7. The **Enable request rate limiting** toggle is enabled. Leave *60* for **Number of requests** and *1 minute* for **Time interval**.

8. **Request timeout** is set to *120* by default.

9. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

10. Select **Save** to proceed.

Configure Service action parameters

After you configure the template above, you need to configure a service action correctly. For the **My Grades microapp**, update their `host_name` in the **View on Blackboard** button of **My Grades page**.

Follow these steps:

1. From the **Microapp Integrations** page, navigate to the Blackboard Learn integration and select the **My Grades** microapp.
2. Select **Pages**, and then the **My Grades** page.
3. In the page builder, select the **View on Blackboard** button element, and then the **Actions** tab in the right pane.
4. Under **Actions**, select **Go to URL**.
5. In the **URL template** field, replace `host_name` with your instance host name. This form is used: `https://{ host_name } /webapps/bb-social-learning-BBLEARN/execute/mybb?cmd=display&toolId=MyGradesOnMyBb_____MyGradesTool`

   When finished, leave the screen. Changes are saved automatically for the builder.

Edit table attributes

To finish configuring the integration, you need to change the data type of the endpoints listed below.

Follow these steps:

1. From the **Microapp Integrations** page, select the menu next to the Blackboard Learn integration, and then **Edit**. The **Data Loading** screen opens. If not, select **Data Loading** from the left side navigation column.
2. For each **Endpoint** listed in the table below, you need to change the **Data type** in the menu for the given **Attribute**.

Endpoint	Attribute	Data type change
Course Announcements	Body	Binary
User Grades	Display Score	Double
Grade Score	Score Possible	Double

3. For each endpoint, select the menu next to the endpoint and **Edit**.

4. In the **Edit Data Endpoint** screen, scroll to the bottom of the page. In the table under **Data structure**, select the pencil icon to edit the table.
5. In the **Edit table attributes** screen, change the **Data type** by selecting the new value from the menu. For each change made for each endpoint, select **Save** and then **Apply**.
6. Repeat for the other data endpoints.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Blackboard Learn connector specifications.

> **Notes**
>
> The pagination limit is set to 100. Administrators can extend this limit up to 200 based on APIs.
>
> Due to limit in API call, incremental synchronization is setup to retain only **Course Memberships** and **Course Announcements**. Remaining endpoints will be triggered as part of full synchronization.

Use Blackboard Learn microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note**
>
> For the **My Grades microapp**, an administrator needs to update their "host_name" in the **View on Blackboard** button of **My Grades page**. For complete steps, see Configure Service action parameters.

Course Registration: Register for a course.

Notification or Page	Use-case workflows
New Course Registration notification	When a student enrolls in a course, the enrolled student receives a notification.
Course Registration Detail page	Provides a read only view of enrolled courses with course and instructor details.
List Courses page	Provides a list of available courses.
Course Details page	View course details, instructor details, with a **Quick Enroll** option.

Create Course Announcement: Allows instructors to an create announcement for a course.

Notification or Page	Use-case workflows
Create Course Announcement page	Provides a form to create an announcement for a course with the following details: Course (Courses drop-down menu), Title, Message, and Publish Date.

Instructor View: Allows instructors to view course members and grades.

Notification or Page	Use-case workflows
Course Details page	Allows the instructor to view a list of available courses.
Member Details page	Allows the instructor to view a list of members enrolled in a course.
Grade Details page	Allows the instructor to view the grades of specific course members.

My Courses: View course announcements and attachments of a course

Notification or Page	Use-case workflows
New Course Announcements notification	When there's a new announcement, students enrolled in the course receive a notification.
Course Announcement Detail page	Provides a read only view of new announcements for a course with details.
My Courses page	Provides a list of courses the user is enrolled in.
Course Details page	Provides a list of announcements and attachments of a course.
Announcement Detail page	View a course announcement and its details.
Attachment Detail page	View and download the attachment.

My Grades: Allows students to view course grades.

Notification or Page	Use-case workflows
New Grades notification	When there's a new grade posted, students enrolled in the course receive a notification.
My Grades page	Allow students to view the grades by selecting the course.

Integrate Canvas LMS

July 21, 2021

Deploy the Canvas LMS integration to view courses, create course announcements, manage course enrollment, and view student's grades.

- As a student, view courses and their related details.
- As a teacher, create a course announcement, view/add course members, and view student's grades.

We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapps for Canvas LMS, see Use Canvas LMS microapps.

Review prerequisites

After you set up this integration with Canvas LMS, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://{ host_name } .com`
- **Authorization URL**: `https://{ host_name } .com/login/oauth2/auth`
- **Token URL**: `https://{ host_name } .com/login/oauth2/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server.
- **Client secret**: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with

> your configured microapp.

Configure Citrix Gateway to support single sign-on for Canvas LMS so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.

Enable APIs

Canvas APIs are enabled by default. You need a Canvas **Partner Account** to access the APIs. The number of API requests that can be made to specific resources is limited. We therefore recommend API limit throttling, as described in this article: https://canvas.instructure.com/doc/api/file.throttling.html.

Create a New Service Account

The integration requires regular access to your Canvas LMS instance, so we recommend creating a dedicated user account. This account must have the following permissions: Full administrator privileges

It is recommended to have a Canvas Partner account: https://www.instructure.com/canvas/become-partner.

Configure OAuth server

Configure the OAuth server to read data through the Canvas integration.

1. Sign in to `https://{ host_name } .com/accounts` as an administrator.
2. Select the account that you want to integrate.
3. Select **Developer Keys** and the select **+ Developer Key**.
4. Select **+ API Key** and complete the required fields including Key Name, Owner Email, Redirect URIs.
5. Enter the following authorized redirect URLs for this app in the Redirect URL field: `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
6. Make the Client Credentials Audience as Canvas and click on **Save**.
7. Make the State as **ON**.
8. Copy the **ClientID** and **Secret** from the details. You use these values for Service Authentication while configuring the integration.

Configure OAuth client

Configure the OAuth client to write data back through the Canvas integration.

1. Sign in to `https://{ host_name } .com/accounts` as an administrator.

2. Select the account that you want to integrate.
3. Select **Developer Keys** and the select **+ Developer Key**.
4. Select **+ API Key** and complete the required fields including Key Name, Owner Email, Redirect URIs.
5. Enter the following authorized redirect URLs for this app in the Redirect URL field: `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serviceAction/callback`
6. Make the Client Credentials Audience as Canvas and click on **Save**.
7. Make the State as **ON**.
8. Copy the **ClientID** and **Secret** from the details. You use these values for Service Action Authentication while configuring the integration.

Add the integration to Citrix Workspace Microapps

Add the Canvas LMS integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

2. Choose the **Canvas LMS** tile.

3. Enter a name for the integration.

4. Enter **Connector parameters**:

   - Enter the instance **Base URL:** `https://{ host_name } .com`
   - Select an Icon for the integration from the Icon Library, or leave this as the default icon.

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

   a) Select **Authorization code** from the **Grant type** menu.
   b) The Authorization URL is prefilled: `https://{ host_name } .com/login/oauth2/auth`.
   c) The Token URL is prefilled: `https://{ host_name } .com/login/oauth2/token`.
   d) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this when you configure the OAuth server.

   e) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration. You collect this when you configure the OAuth server.

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication** in Actions toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization code** from the **Grant type** menu.
   c) The Authorization URL is prefilled: `https://{ host_name } .com/login/oauth2/auth`.
   d) The Token URL is prefilled: `https://{ host_name } .com/login/oauth2/token`.
   e) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collected this value when you configure the OAuth client.
   f) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration. You collected this value when you configure the OAuth client.

7. Enable the **Request rate limiting** toggle. Enter *500* for **Number of requests** and *1 minute* for **Time interval**.

8. Leave **Request timeout** empty.

9. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

10. Select **Save** to proceed.

Configure Service action parameters

After you configure the template above, you need to configure a service action correctly. For the **My Grades** microapp, update their `host_name` in the **View On Canvas** button of **My Grades** page.

Follow these steps:

1. From the Microapp Integrations page, navigate to the **Canvas LMS** integration and select the **My Grades** microapp.
2. Select **Pages**, and then the **My Grades** page.
3. In the page builder, select the **View on Canvas** button component, and then the **Actions** tab in the right pane.
4. Under **Actions**, select **Go to URL**.
5. In the **URL template** field, replace `host_name` with your instance host name. This form is used: `https://{ host_name } /courses/\\{ \\{ id\ } \ } /grades`
6. When finished, leave the screen. Changes are saved automatically for the builder.

Edit table attributes

To finish configuring the integration, you need to change the data type of the endpoints as shown in the following list. For each **Endpoint** listed in the following table, you must change the **Data type** in the menu for the given attribute.

Table	Column	Data type
Assignments	points_possible	Binary
Assignments	description	Binary
Grades	current_grade	Binary

Follow these steps:

1. From the **Microapp Integrations** page, select the menu next to the Canvas LMS integration, and then **Edit**. The Data Loading screen opens. If not, select Data Loading from the left side navigation column.
2. For each endpoint, select the menu next to the endpoint and **Edit**.
3. In the **Edit Data Endpoint** screen, scroll to the bottom of the page. In the table under **Data structure**, select the pencil icon to edit the table.
4. In the **Edit table attributes** screen, change the **Data type** by selecting the new value from the menu. For each change made for each endpoint, select **Save** and then **Apply**.
5. Repeat this procedure for the other data endpoints.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

> **Note:**
>
> We recommend setting the **Full Synchronization** interval to every 8 hours and **Incremental Synchronization** interval as hourly.

For more details of API endpoints and table entities, see Canvas LMS connector specifications.

> **Note:**
>
> Due to a limit in the number of API calls, incremental synchronization is set up to retain only **Announcements**. Remaining endpoints are triggered as part of full synchronization.

Use Canvas LMS microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Create Course Announcement: Teachers create an announcement for a course.

Notification or Page	Use-case workflows
Create Course Announcement page	Provides a form to create an announcement for a course with the following details: Course (Courses drop-down menu), Title, Message, and Publish Date.

My Courses: View course announcements, assignments, and course files.

Notification or Page	Use-case workflows
Course Invitation notification	When a student is invited to a course, they receive a notification.
New Course Announcements notification	When there is a new announcement for a course, students enrolled to the course receive a notification.
New Course Announcements (Future) notification	When there is a new announcement for a course with a future publish date, students enrolled to the course receive a notification.
Accept/Reject Invitation page	Provides a read-only view of a course invitation with options to **Accept** or **Reject**.
List Courses page	Allows users to view their list of enrolled courses.
Course Details page	Allows users to view a list of announcements and files of a course.
Announcement Details page	Allows users to view the course announcement and its details.
Assignment Details page	Allows users to view an assignment and its details.
File Details page	Allows users to view and download files.

My Grades: Students view their course grades.

Notification or Page	Use-case workflows
New Grades notification	When new grades are posted for a course, students enrolled in the course receive a notification.
List Courses page	Allows users to view their list of enrolled courses.
My Grades page	Allow users to view the grades of the selected course.

Teacher's View: Teachers view and add course members and view students.

Notification or Page	Use-case workflows
List Courses page	Allows teachers to view a list of available courses.
List Members page	Allows teachers to view a list of members enrolled for a course, with an **Add** button that navigates to the **Add User** page.
Member Details page	Allows teachers to view the grades and related details of members of a course.
Add User page	Allows teachers to add a student to the selected course.

Integrate SAP Concur

November 5, 2021

Integrate with SAP Concur to submit requests and receive notifications about approvals, expenses, itineraries, and reports.

> **Note**
>
> We provide two SAP Concur integration templates for your use. We recommend using the newer template in the **Integrations** category for most use-cases as it provides more power to configure the cached data structure. Find the Legacy (Deprecated) integration template details at SAP Concur - Add the legacy integration.

For a comprehensive list of out-of-the-box SAP Concur microapps, see Use SAP Concur microapps.

Review prerequisites

These prerequisites assume you administer the SAP Concur instance of your organization to set up the integration. You must have these details to add the integration in Citrix Workspace Microapps:

- **Base URL**: This is the base URL model: `https://{ data_center } .api.concursolutions .com`. For more information, see SAP Concur - Base URIs.
- **Token URL**: `https://{ data_center } .api.concursolutions.com/oauth2/v0/ token`
- **Authorization URL**: `https://{ data_center } .api.concursolutions.com/oauth2 /v0/authorize`
- **Username**: This and Password are the credentials of the service account with access to SAP Concur.
- **Password**: This and Username are the credentials of the service account with access to SAP Concur.
- **Client ID**: You collect the Client ID by registering the OAuth client in your SAP Concur account. The Client ID and the Client Secret are the same for both Service authentication and Service action authentication. See Request Client ID and Client secret.
- **Clent secret**: You collect the Client secret by registering the OAuth client in your SAP Concur account. The Client ID and the Client secret are the same for both Service authentication and Service action authentication. See Request Client ID and Client secret.
- **Scope**: CCARD creditcardaccount.read receipts.read COMPD USER user_read EMERG JOBLOG company.read ERECPT ITINER FISVC LIST PASSV mileage.vehicle.writeonly CONFIG FOP mileage.rate.writeonly receipts.writeonly GHOST user.read CONREQ user.write COMPANY mileage.journey.read EVS TRVPTS ATTEND INVPO NOTIF TRVREQ SUPSVC company.write EXPRPT EXTRCT PAYBAT INVPMT mileage.vehicle.read IMAGE TAXINV RCTIMG UNUTX TWS TMCSP quickexpense.writeonly BANK INVVEN openid receipts.write travelrequest.write MTNG mileage.rate.read mileage.journey.writeonly INSGHT TRVPRF INVTV MEDIC TSAI

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

- You must have the Web Services component purchased from SAP Concur. If your company hasn't purchased the Web Services component, ask the SAP Concur administrator in your organization to contact your Account Manager at SAP Concur.
- Configure Citrix Gateway to support single sign-on for SAP Concur so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Concur Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Request Client ID and Client secret

Obtain a new oauth2 client_id and client_secret and define the scope of client's application following these instructions.

> **Important:**
>
> Create a support ticket with Concur to configure the integration using OAuth 2.0 and register correct authentication callback URLs. We recommend using the following text for your support ticket:
>
> We request OAuth 2.0 credentials to enable an integration with Citrix Workspace. This tool downloads recent expense reports and line item details using a service account to present pending approvals to managers. It allows employees to submit, approve, and reject expenses. Actions taken by end users are directed to the 3LO authorization page where they enter their Concur credentials and approve access by the Workspace app.
>
> Add the following authorized redirect URLs to Concur for this integration with authorization grant access granted to allow access to private data and enable OAuth authenticated user actions. The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com:
>
> - `<https://microapps_server_URL/admin/api/external-services/com.sapho.services.concur.ConcurService/auth/serverContext>`
> - `<https://microapps_server_URL/app/api/auth/serviceAction/callback>`

Add the integration to Citrix Workspace Microapps

Add the SAP Concur HTTP integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the **SAP Concur** tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.

- Enter the instance **Base URL**. This is the base URL model: `https://{ data_center } .api.concursolutions.com`. For more information, see SAP Concur - Base URIs.
- Select an **Icon** for the integration from the Icon Library, or leave this as the default Salesforce icon.
- Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

    a) Select **OAuth 2.0** from the **Authentication method** menu.
    b) Select **Resource Owner Password** from the **Grant type flow** menu. Provide the correct credentials to authorize resource server provision of an access token.
    c) Enter **password** in the **Grant type value** menu.
    d) Select **Request body** from the **Token authorization** menu.
    e) Select **URL encoded form** from the **Token content type** menu.
    f) The **Token URL** is prefilled: `https://{ data_center } .api.concursolutions.com/oauth2/v0/token`
    g) Leave the **Refresh token URL** and **Scope** fields empty.
    h) Enter your **Username** and **Password**.
    i) Enter your **Client ID** and **Clent secret**.
    j) Leave **Header prefix** empty.

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level.

    a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
    b) Select **Authorization code** from the **Grant type flow** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.
    c) Enter **authorization_code** in the **Grant type value** menu.
    d) The **Callback URL** field is prefilled.
    e) Select **Request body** from the **Token authorization** menu.
    f) Select **URL encoded form** from the **Token content type** menu.
    g) The **Authorization URL** is prefilled: `https://{ data_center } .api.concursolutions.com/oauth2/v0/authorize`

h) The **Token URL** is prefilled: `https://{ data_center } .api.concursolutions.com/oauth2/v0/token`

i) Leave the **Refresh token URL** field empty.

j) The **Scope** field contains the following scopes: `CCARD creditcardaccount.read receipts.read COMPD USER user_read EMERG JOBLOG company.read ERECPT ITINER FISVC LIST PASSV mileage.vehicle.writeonly CONFIG FOP mileage.rate.writeonly receipts.writeonly GHOST user.read CONREQ user.write COMPANY mileage.journey.read EVS TRVPTS ATTEND INVPO NOTIF TRVREQ SUPSVC company.write EXPRPT EXTRCT PAYBAT INVPMT mileage.vehicle.read IMAGE TAXINV RCTIMG UNUTX TWS TMCSP quickexpense.writeonly BANK INVVEN openid receipts.write travelrequest.write MTNG mileage.rate.read mileage.journey.writeonly INSGHT TRVPRF INVTV MEDIC TSAI`

k) Enter your **Client ID** and **Clent secret**.

l) Leave **Header prefix** empty.

7. By default the **Request rate limiting** field is set to **50 requests per second**.
8. **Request timeout** is set to *120* by default.
9. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
10. Select **Save**.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

Use SAP Concur microapps

Our SAP Concur integration comes with the following preconfigured out-of-the-box microapps:

Approvals: Approve pending expense reports or send them back.

Notification or Page	Use-case workflows
Approval Reminder notification	When an expense report is pending approval and passes a defined threshold before deadline (for example, 3 days by default), the approver receives a notification reminder.
New Request notification	When a new request for approval is assigned to an approver, the approver receives a notification.

Notification or Page	Use-case workflows
New Request notification	When there is a change to a request for approval, the approver receives a notification.
Expense Detail page	Provides a read only view of an expense with details.
Expense Report Detail page	Provides a detailed view of a report with a list of expenses with options to **Approve** and **Send Back**.
My Pending Approvals page	Provides a personalized list of pending approvals.

Expenses: Search and view expenses.

Notification or Page	Use-case workflows
My Expense Detail page	Provides a detailed view of an expense with attachments.
My Expense Report Detail page	Provides a detailed view of a report with a list of expenses with the option to **Recall**.
My Expenses page	Provides a searchable, personalized list of expenses.

Itineraries: Search, view, and share itineraries.

Notification or Page	Use-case workflows
Changed Itinerary successfully booked notification	When a user’s itinerary is changed, the user receives a notification.
Changed Shared Itinerary notification	When a changed itinerary is shared with a user, they receive a notification.
New Itinerary successfully booked notification	When a user’s new itinerary is booked and ticketed, the user receives a notification.
New Shared Itinerary notification	When an itinerary is shared with a user, they receive a notification.
Itinerary Detail page	Provides a view of a user’s itinerary and an actionable share button.

Notification or Page	Use-case workflows
My Itineraries page	Provides a personalized list of itineraries.

Reports: Search and view your submitted expense reports.

Notification or Page	Use-case workflows
Request Approved notification	When a user's expense report is approved, they receive a notification.
Request Send Back notification	When a user's expense report is sent back, they receive a notification.
My Expense Detail page	Provides a detailed view of an expense with attachments.
My Expense Report Detail page	Provides a detailed view of a report with a list of expenses with the option to **Recall**.
My Expense Reports page	Provides a list of expense reports.

Add the legacy integration

As a System Admin, you use the following process to enable the Concur Integration. Ensure you meet the prerequisites then set up the Concur integration. After you set up the integration, you must provide employees with the employee name, email address, employee id, or log-in id of the dedicated account and ask them to adjust their SAP Concur profile settings. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

Review prerequisites

These prerequisites assume you administer the SAP Concur instance of your organization to set up the integration. You must have these details to add the integration in Citrix Workspace Microapps:

- Instance URL
- Username
- Password

For OAuth 2.0:

- OAuth Client ID
- OAuth Client Secret
- Authorization URL

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

- Obtain a new oauth2 client_id and client_secret and define the scope of client's application.

> **Important:**
>
> Create a support ticket with Concur to configure the integration using OAuth 2.0 and register correct authentication callback URLs. We recommend using the following text for your support ticket:
>
> We request OAuth 2.0 credentials to enable an integration with Citrix Workspace. This tool downloads recent expense reports and line item details using a service account to present pending approvals to managers. It allows employees to submit, approve, and reject quick expenses. Actions taken by end users are directed to the 3LO authorization page where they enter their Concur credentials and approve access by the Workspace app.
>
> Add the following authorized redirect URLs to Concur for this integration with authorization grant access granted to allow access to private data and enable OAuth authenticated user actions. The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com:
>
> - `<https://microapps_server_URL/admin/api/external-services/com.sapho.services.concur.ConcurService/auth/serverContext>`
> - `<https://microapps_server_URL/app/api/auth/serviceAction/callback>`

- You must have the Web Services component purchased from SAP Concur. If your company hasn't purchased the Web Services component, ask the SAP Concur administrator in your organization to contact your Account Manager at SAP Concur.
- Configure Citrix Gateway to support single sign-on for SAP Concur so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Concur Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Configure the legacy template

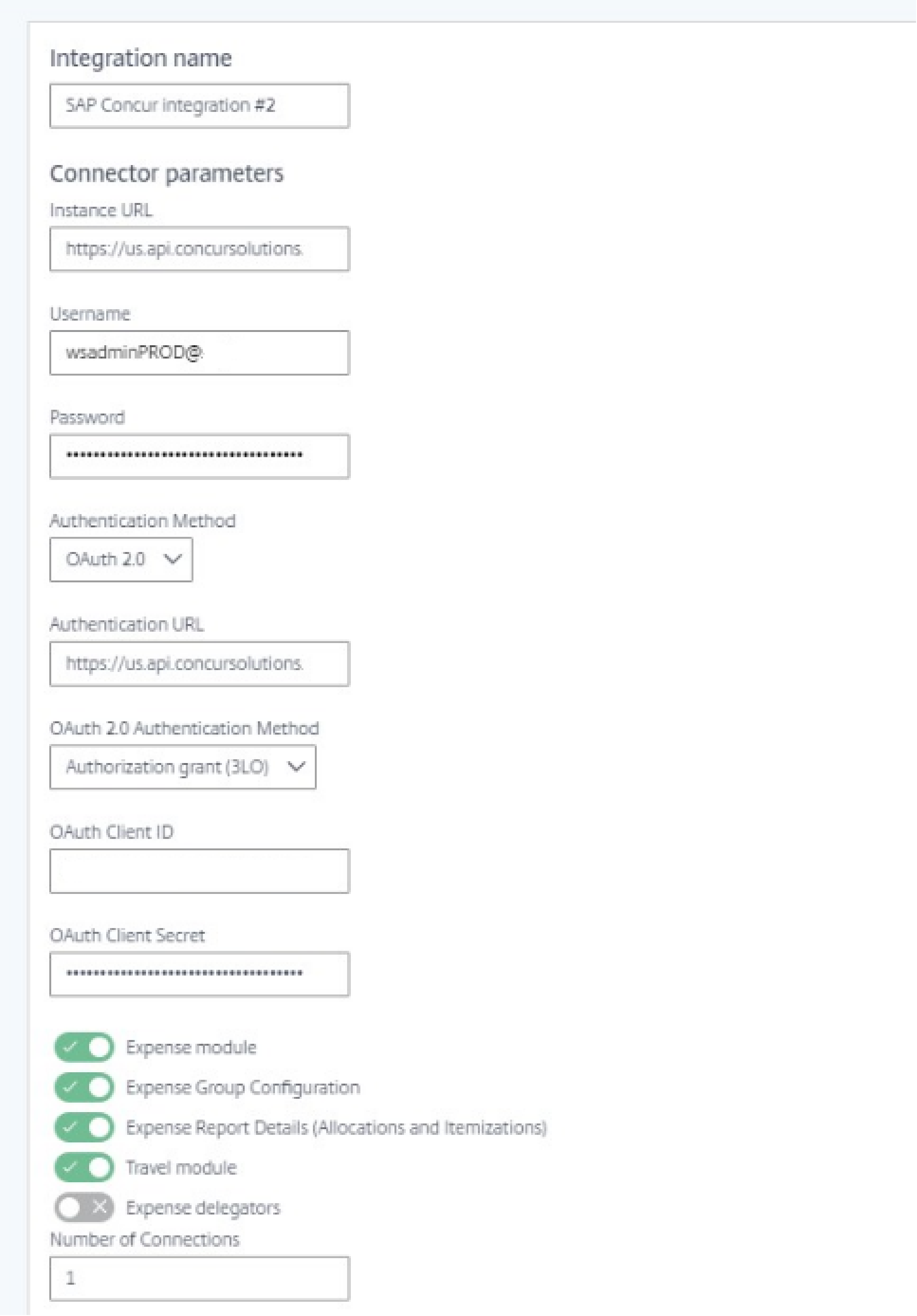
Add the SAP Concur integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the SAP Concur tile.
4. Enter a name for the integration.



Integration name

SAP Concur integration #2

Connector parameters

Instance URL

https://us.api.concursolutions.

Username

wsadminPROD@

Password

••••••••••••••••••••••••••••••••••••

Authentication Method

OAuth 2.0

Authentication URL

https://us.api.concursolutions.

OAuth 2.0 Authentication Method

Authorization grant (3LO)

OAuth Client ID

OAuth Client Secret

••••••••••••••••••••••••••••••••••••

✓ Expense module

✓ Expense Group Configuration

✓ Expense Report Details (Allocations and Itemizations)

✓ Travel module

✕ Expense delegators

Number of Connections

1

5. Enter the **Connector parameters** that you collected as prerequisites.
   - Enter your **Instance URL**.
   - Enter the **Username** and **Password**.
   - Select an **Authentication Method**. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access.
   - Enter an **Authorization URL** to define the authorization server url provided when setting up the target application integration.

- Select an **OAuth 2.0 Authentication Method**.
  - **Credentials** The client's credentials are used instead of the resource owner's. The access token is associated either with the client itself, or delegated authorization from a resource owner.
  - **Authorization grant (3LO)** The resource owner allows access.
- Enter the **OAuth Client ID** and **OAuth Client Secret** that you collected in the prerequisites procedure.

6. Select the following radio buttons as required:

   - **Expense module** Provides access to employee expense tracking.
   - **Expense Group Configuration** Retrieves the list of Expense Polices, Expense Types, and Payment Types for the Expense Group that the OAuth access token is assigned to.
   - **Expense Report Details (allocations and Itemizations)** Allows for the retrieval of allocation information as it relates to a Report ID, Entry ID, or Itemization ID.
   - **Travel module** Provides access to travel data such as itineraries, travel profiles, and travel requests.
   - **Expense delegators** Retrieves a list of users that have granted delegate permissions to the user specified in the OAuth access token.

7. Enter a quantity for **Number of Connections**. This value determines the number of strings the data sync initiates.

   > **Note:**
   >
   > The default number of connections is one. Opening more connections reduces the time for data synchronization, but increases the load on the Microapps server and can influence its performance. If you require, we recommend no more than 10.

8. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

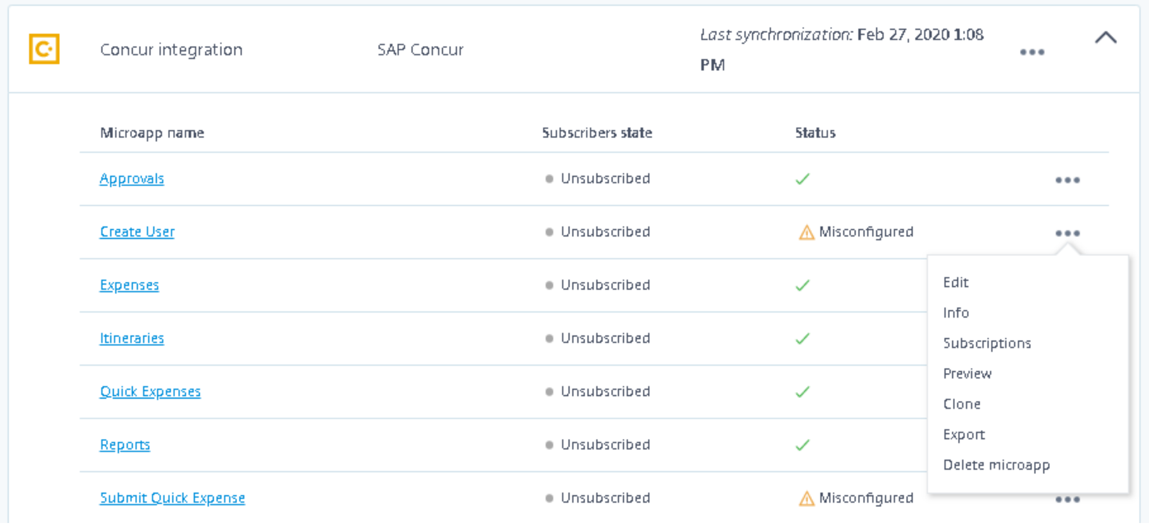
You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Concur connector specifications.

Use SAP Concur legacy microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Important!**
>
> These SAP Concur microapps are now deprecated: **Submit Quick Expense**, **Create User**, and **User**. Users need to delete these microapps as they will become misconfigured. To delete the microapp, from the Microapps Integrations page select the menu next to the microapp that you want to delete. Select **Delete microapp** and confirm.



Concur integration SAP Concur *Last synchronization:* Feb 27, 2020 1:08 PM

Microapp name	Subscribers state	Status
Approvals	Unsubscribed	✓
Create User	Unsubscribed	⚠ Misconfigured
Expenses	Unsubscribed	✓
Itineraries	Unsubscribed	✓
Quick Expenses	Unsubscribed	✓
Reports	Unsubscribed	✓
Submit Quick Expense	Unsubscribed	⚠ Misconfigured

Our SAP Concur integration comes with the following preconfigured out-of-the-box microapps:

Approvals: Approve pending expense reports or send them back.

Notification or Page	Use-case workflows
Approval Reminder notification	When an expense report is pending approval and passes a defined threshold before deadline (for example, 3 days by default), the approver receives a notification reminder.
New Request notification	When a new request for approval is assigned to an approver, the approver receives a notification.
New Request (changed) notification	When there is a change to a request for approval, the approver receives a notification.
Expense Detail page	Provides a read only view of an expense with details.

Notification or Page	Use-case workflows
My Pending Approvals page	Provides a personalized list of pending approvals.
Review Report page	Provides an actionable form with a detailed view of an expense report pending approval.

Expenses: Search and view expenses.

Notification or Page	Use-case workflows
Expense Detail page	Provides a read only view of an expense with details.
My Expenses page	Provides a personalized list of expenses.
Report Detail page	Provides a detailed view of an expense report with a field for commenting.

Itineraries: Search, view, and share itineraries.

Notification or Page	Use-case workflows
New Itinerary successfully booked notification	When a user's new itinerary is booked and ticketed, the user receives a notification.
New Shared Itinerary notification	When an itinerary is shared with a user, they receive a notification.
Itinerary Detail page	Provides a view of a user's itinerary and an actionable share button.
My Itineraries page	Provides a personalized list of itineraries.
Share Itinerary page	Provides a form for sharing a user's itinerary.

Reports: Search and view your submitted expense reports.

Notification or Page	Use-case workflows
Request Approved notification	When a user's expense report is approved, they receive a notification.

Notification or Page	Use-case workflows
Request SendBack notification	When a user's expense report is sent back, they receive a notification.
Expense Detail page	Provides a read only view of an expense with details.
Expense Report page	Provides a detailed view of an expense report with a field for commenting.
My Pending Expense Reports page	Provides a personalized list of expense reports.

Integrate Cherwell

October 26, 2021

Deploy the Cherwell integration template to manage and be notified about changes to incidents and service requests.

We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapp for Cherwell, see Use Cherwell microapps.

Review prerequisites

After you set up this integration with Cherwell, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://{ Cherwell_Instance_URL } /CherwellAPI/api/`. `{ Cherwell_Instance_UR` `}` is obtained from Cherwell when purchased.
- **Token URL**: `https://{ Cherwell_Instance_URL } /CherwellAPI/token`. `{ Cherwell_Instance_URL }` is obtained from Cherwell when purchased.
- **Username**: This and Password are the credentials of the service account with access to Cherwell.
- **Password**: This and Username are the credentials of the service account with access to Cherwell.
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server.
- **Client secret**: The client secret is a unique string issued when setting up the target application integration.

It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

The integration requires regular access to your Cherwell instance, so we recommend creating a dedicated user account. This account must have the following permissions:

- Administrator security group

Create Client Id and Client Secret

Before performing this procedure, navigate to `https://{ Cherwell_Instance_URL } / CherwellAutoDeploy/` and download Cherwell Service Management and install it.

1. Log in with your User ID and Password.
2. On the Cherwell Service Management APP screen, select **Security**.
3. Go to **Edit REST API client settings**.
4. Select **Create new client**.
5. Enter a value for:
    - **Name**
    - **Description**
    - Enter *360* for **Token lifespan**
    - Enter *1440* for **Refresh Token lifespan
    - Select the **API Access enabled** checkbox.
6. Select **Save current client**.
7. Copy and save the **Client Key**. This is the value for **Client Id** and **Client Secret** for Service Authentication while configuring the integration.

Create Username and Password

Again before performing this procedure, navigate to `https://{ Cherwell_Instance_URL } / CherwellAutoDeploy/` and download Cherwell Service Management and install it.

1. Log in with your User ID and Password.
2. On the Cherwell Service Management APP screen, select **Security**.
3. Select **Edit Users**.
4. Select **Create new user**.
5. Enter a value for:

    - **Login ID**
    - **Password**
    - **Confirm password**

6. Select Security Group: Admin, and enter **Full name** and **E-Mail**.

7. Select the **Password never expires** checkbox.

8. Select **Save current user**.

> **Note:**

Add the integration to Citrix Workspace Microapps

Add the Cherwell integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Cherwell tile.
3. Enter a name for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**: `https://{ Cherwell_Instance_URL } /CherwellAPI /api/`. `{ Cherwell_Instance_URL }` is obtained from Cherwell when purchased.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.
5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
    a) Select **Resource owner password** from the **Grant type** menu.
    b) Enter *password* for **Grant type value**.
    c) Select **Request body** as **Token authorization**.
    d) Select **URL encoded form** as **Token content type**.
    e) Enter your **Token URL**: `https://{ Cherwell_Instance_URL } /CherwellAPI/ token`. `{ Cherwell_Instance_URL }` is obtained from Cherwell when purchased.
    f) Enter your **Username**. This and Password are the credentials of the service account with access to Cherwell.
    g) Enter your **Password**. This and Username are the credentials of the service account with access to Cherwell.

h) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server.
i) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.
j) Select **Bearer** from the **Header Prefix** menu.
k) Under **Access Token Parameters** enter *auth_mode* as **Name** and *internal* as **Value**.

6. Leave the **Service Action Authentication** toggle as disabled.
7. Leave the **Enable request rate limiting** toggle as disabled.
8. (Optional) Enable the **Logging** toggle to keep 24 hours of logging for support purposes.
9. Select **Save** to proceed.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Cherwell connector specifications.

Use Cherwell microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Create Incident: Allows the user to create an incident.

Notification or Page	Use-case workflows
Create Incident page	Provides a form to create an incident.

Create Service Request: Create a new service request from the catalog or select predefined service requests templates.

Notification or Page	Use-case workflows
Create Service Request page	Provides a form to select a predefined template or create a request from the service catalog.

My Open Incidents: View and manage open incidents that the workspace user owns.

Notification or Page	Use-case workflows
Incident Created notification	When a new incident is created, the owner of the incident receives a notification.
Incident Resolution SLA Thresholds Hit notification	When an incident's resolution SLA is missed, the owner of the incident receives a notification.
Incident Response SLA Thresholds Hit notification	When an incident's response SLA is missed, the owner of the incident receives a notification.
Incident Status Changed notification	When an incident's status is changed, the owner of the incident receives a notification.
My Open Incidents page	Provides a list of open incidents owned by the workspace user.
Incident Details page	Provides incident details including comments with the option to add comments and withdraw the incident.

My Open Serice Requests: View and manage open service requests that the workspace user owns.

Notification or Page	Use-case workflows
Service Request Created notification	When a new service request is created, the owner of the incident receives a notification.
Service Request Resolution SLA Thresholds Hit notification	When a new service request's resolution SLA is missed, the owner of the incident receives a notification.
Service Request Response SLA Thresholds Hit notification	When a new service request's response SLA is missed, the owner of the incident receives a notification.
Service Request Status Changed notification	When a new service request's status is changed, the owner of the incident receives a notification.
My Open Service Requests	Provides a list of open service requests owned by the workspace user.
Service Request Details	Provides service request details including comments with the option to add comments and withdraw the service request.

Integrate DocuSign

September 21, 2021

Deploy the DocuSign integration to send and receive envelopes for digital signatures from any device using Citrix Workspace. Users can initiate a new template for signature, be notified for any new pending documents, and view a list of envelopes previously sent or received in their DocuSign inbox.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapps for DocuSign, see Use DocuSign microapps.

Review prerequisites

These prerequisites assume that the administrator is part of the DocuSign integration set up of the organization. This DocuSign admin account must have full read privileges for user information.

After you set up this integration with DocuSign, you will need these artifacts to add the integration in Citrix Workspace Microapps, specifically the following list of parameters for setting up OAuth integration:

- **Base URL**: The Base URL to which the endpoint paths are appended. You can collect the Base URL from the Service Account under **Apps and Keys**. Base URL format: `{ baseURL fetched from sevice account } /restapi/2.1/`. For how to find Base URL, see Configure OAuth server.
- **DocuSign Account Id**: Your docusign admin account Id. See Configure OAuth server.
- **Environmental URL**: For example from the base URL `https://{ yourEnvironmentalURL } .docusign.net/restapi/v2.1/`.
- **Authorization URL**: `https://account-d.docusign.com/oauth/auth`
- **Token URL**: `https://account-d.docusign.com/oauth/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server. Additionally, all application keys for DocuSign must pass the DocuSign go live process. This applies to the **Integration Key** (**ClientId**). This can take up to a week. For more information about the DocuSign go live process, see https://developers.docusign.com/docs/esign-rest-api/go-live/.
- **Client Secret**: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for DocuSign so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

DocuSign API limitation form link: https://developers.docusign.com/docs/esign-soap-api/esign101/security/call-limits/

Create a new service account

Sign up here: https://admin.docusign.com. You can view User Permission Profiles here: https://support.docusign.com/en/guides/ndse-admin-guide-permission-sets. If you want to access the demo version of eSignature Admin, use this URL instead: https://admindemo.docusign.com. For more information on new service accounts, see: https://support.docusign.com/en/guides/org-admin-guide-getting-started.

Configure OAuth server

Configure the OAuth server to read data through the DocuSign integration.

1. Log in to the eSignature Admin with your service account: https://admin.docusign.com. For the demo version of eSignature Admin: https://admindemo.docusign.com. For more information, see: https://support.docusign.com/en/guides/ndse-admin-guide-access and https://support.docusign.com/en/guides/ndse-admin-guide-welcome-to-administration.
2. Go to **Setting** in the navigation bar.
3. On the settings page, select **Apps and Keys** and then **ADD APP & INTEGRATION KEY**.
4. Enter an **APP Name** and select **ADD**.
5. Select **+Add Secret Key** to generate a Secret Key.
6. Enter the Redirect URI: `https://{ yourmicroappserverurl } /admin/api/gwsc/auth /serverContext`
7. Copy and save the **Integration Key** and **Secret Key** (**ClientId** and **Secret**) shown on the screen. You use these details for **Service Authentication** while configuring the integration.
8. Copy and save the **Account's Base URI**.
9. Copy and save the **DocuSign Account Id**.

Additionally, all application keys for DocuSign must pass the DocuSign go live process. This applies to the **Integration Key** (**ClientId**). This can take up to a week. For more information about the DocuSign go live process, see https://developers.docusign.com/docs/esign-rest-api/go-live/.

Configure OAuth client

Configure the OAuth client for writing back data through the DocuSign integration.

1. Log in to the eSignature Admin with your service account: https://admin.docusign.com. For the demo version of eSignature Admin: https://admindemo.docusign.com. For more information, see: https://support.docusign.com/en/guides/ndse-admin-guide-access and https://support.docusign.com/en/guides/ndse-admin-guide-welcome-to-administration.
2. Go to **Setting** in the navigation bar.
3. On the settings page, select **Apps and Keys** and then **ADD APP & INTEGRATION KEY**.
4. Enter an **APP Name** and select **ADD**.
5. Select **+Add Secret Key** to generate a Secret Key.
6. Enter the Redirect URI: `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
7. Copy and save the **Integration Key** and **Secret Key** (**ClientId** and **Secret**) shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

Additionally, all application keys for DocuSign must pass the DocuSign go live process. This applies to the **Integration Key** (**ClientId**). This can take up to a week. For more information about the DocuSign go live process, see https://developers.docusign.com/docs/esign-rest-api/go-live/.

Configure Organization

Link accounts to your organization. The account that you used to create the organization is automatically linked.

1. From the eSignature Admin home page, select **GET STARTED**. If you do not see this option in your DocuSign account, then DocuSign Admin is not enabled on the account. Contact your DocuSign account manager for assistance.
2. Enter an **Organization Name** and an optional **Description** in the fields provided, then select **NEXT**.
3. Select **CREATE** to finish creating your organization. For more information, see https://support.docusign.com/en/guides/org-admin-guide-create-org.
4. Go to the **Organization Admin** dashboard. Select **Connected Apps**, and select **AUTHORIZE APPLICATION**.
5. Select the App name that you entered and created while configuring the OAuth server and client. Select **ADD** to provide permissions.

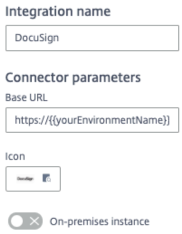
6. Enter these scopes. For more information, see: https://developers.docusign.com/platform/auth/reference/scopes.
    - For service authentication, select the App Name that you created in Configure OAuth server, provide the *user_read* permission, and select **ADD**.
    - For service action authentication, select the App Name that you created in Configure OAuth client, provide the *user_write* permission, and select **ADD**.

Add the integration to Citrix Workspace Microapps

Add the DocuSign integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the DocuSign tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**:
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

DocuSign

Connector parameters

Base URL

https://{{yourEnvironmentName}}

Icon

5. Enter your **DocuSign Account Id**. See Configure OAuth server.
6. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
    a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization

server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.

b) Select **Request body** from the **Token authorization** menu.

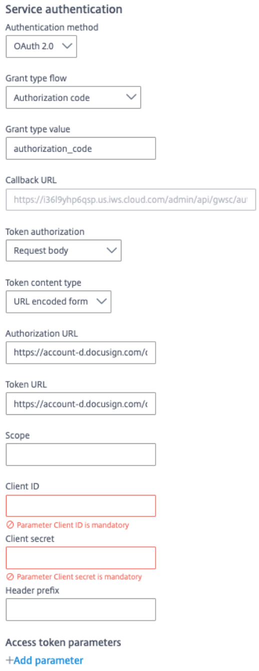
c) The **Authorization URL** is prefilled.

d) The **Token URL** is prefilled.

e) Ensure the following is entered for Scope: *user_read*

f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service authentication

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://account-d.docusign.com/(

Token URL: https://account-d.docusign.com/(

Scope:

Client ID:

⊘ Parameter Client ID is mandatory

Client secret:

⊘ Parameter Client secret is mandatory

Header prefix:

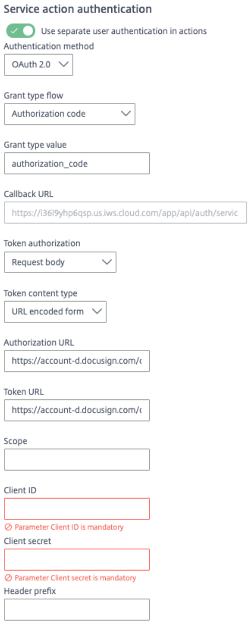
Access token parameters

7. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.

   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.

c) Select **Request body** from the **Token authorization** menu.
d) The **Authorization URL** is prefilled.
e) The **Token URL** is prefilled
f) Ensure the following is entered for Scope: *user_write*
g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.
h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service action authentication

Use separate user authentication in actions

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://account-d.docusign.com/(

Token URL: https://account-d.docusign.com/(

Scope

Client ID

⊘ Parameter Client ID is mandatory

Client secret

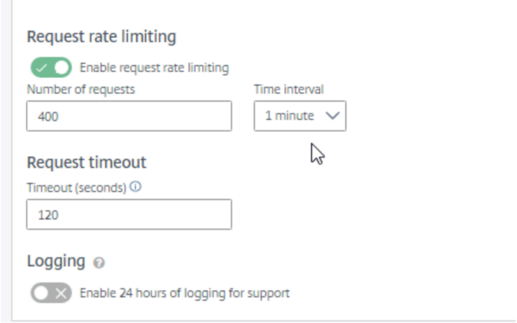
⊘ Parameter Client secret is mandatory

Header prefix

8. The **Enable request rate limiting** toggle is enabled. Leave *400* for **Number of requests** and *1 minute* for **Time interval**.

   1.. **Request timeout** is set to *120* by default.

9. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

Enable request rate limiting

Number of requests: 400

Time interval: 1 minute

Request timeout

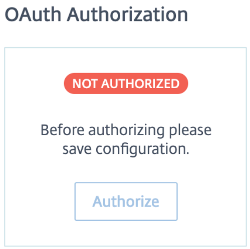
Timeout (seconds): 120

Logging

Enable 24 hours of logging for support

10. Select **Save** to proceed.
11. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Google login screen.

    a) Enter your Service Account user name and password and select **Log in**.
    b) Select **Accept**.

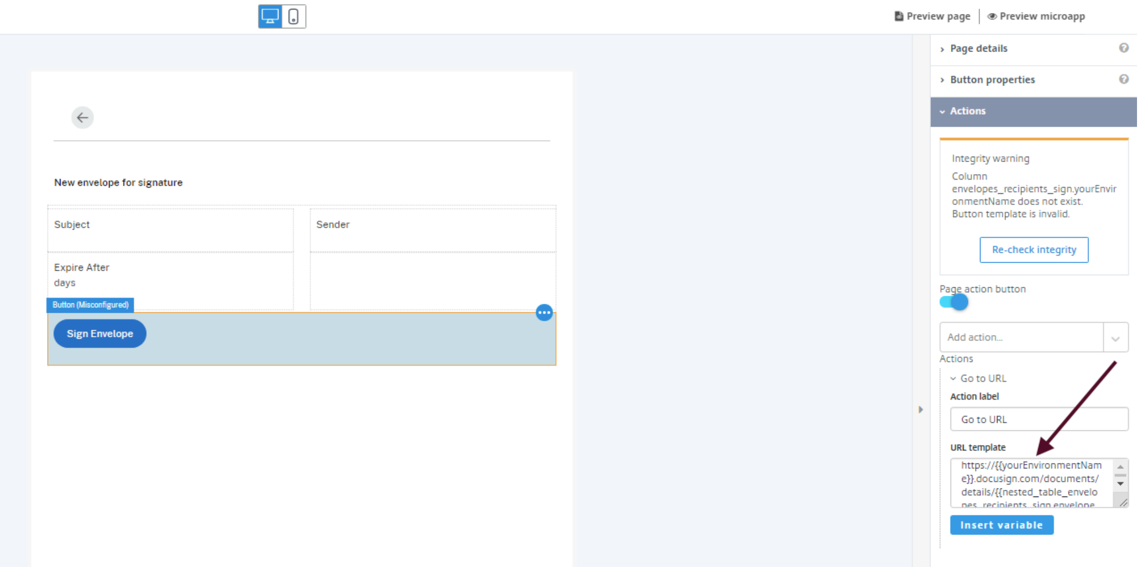

OAuth Authorization

Before authorizing please save configuration.

Add actions to buttons

Configure **Go to URL** actions for the following buttons to enable **Sign Envelope** and **View Envelope** button capabilities. This **Go to URL** configuration redirects the user action to the specific envelope detail page in DocuSign.

1. Navigate to and then select the following buttons in the page builder. Perform this procedure for each button below:
   - **My Received Envelopes** microapp: **Pending Envelope Details** page: **Sign Envelope** button
   - **My Sent Envelopes** microapp: **Recipient View of Envelope** page: **Sign Envelope** button
   - **My Sent Envelopes** microapp: **Sender View of Envelope** page: **View Envelope** button
2. Select the **Actions** tab on the right.
3. Under **Actions**, select **Go to URL**.
4. In the **URL template** field, replace the `{ yourEnvironmentName }` portion of the URL with your environmental URL. Find your environmental URL in your base URL: `https://{ yourEnvironmentName } .docusign.net/restapi/v2.1/`. For example: `https://appdemo.docusign.com/documents/details/\\{ \\{ nested_table_envelopes_recipients_ .envelope_id\ } \ }`.



Integrity warning

Column envelopes_recipients_sign.yourEnvironmentName does not exist. Button template is invalid.

URL template: https://{{yourEnvironmentName}}.docusign.com/documents/details/{{nested_table_envelopes_recipients_sign.envelope

5. Now, repeat this procedure for the other buttons.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see DocuSign connector specifications.

Use DocuSign microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note:**
>
> To enable **Sign Envelope** and **View Envelope** button capabilities, you must replace the button **Action** for Pending Envelope Details page, Recipient View of Envelope page and Sender View of Envelope page with your environmental URL. See Add actions to buttons.

My Received Envelopes: View user's received envelopes, which are awaiting signature. User can authenticate himself and view the Envelope in DocuSign.

Notification or Page	Use-case workflows
My Received Envelopes page	Provides a table with all envelopes that are pending the user's signature, with a search option to filter envelopes by email subject or the sender's name. Users can view envelopes for sent and delivered status.

Notification or Page	Use-case workflows
Received Envelope Details page	Provides details for an envelope pending signature, such as email subject, sender name, and expire after (number of days). Users can view the envelope in Workspace using the **Sign Envelope** button.
Sign Envelope page	Provides a form to sign an envelope from Citrix Workspace.

Send a Template: Send an existing template for signature.

Notification or Page	Use-case workflows
Select Template page	Provides a table with all the templates for the user. The integration doesn't list *Shared* templates in the list of templates for the signed in user.
Send Template page	Provides a form to send the template to recipients. The sender provides the following details to send an envelope: Name, Email, and Role. Envelopes can be sent to at maximum three (3) recipients at a time. A template must contain a document to send an envelope using that template.

My Sent Envelopes: View a list of envelopes sent by the user. Users also can view the details of the envelope, and also add, edit, and delete recipients.

> **Note:**
>
> Notifications expire two days after their creation.

Notification or Page	Use-case workflows
Envelope Status Change notification	When an envelope's status is changed to either *Completed* or *Declined*, a notification is generated and sent to the sender of the envelope. If an envelope's status is already *Completed* or *Declined* when the record is added to the Citrix Microapps Database, then the notification is not generated for the sender.
New Envelope notification	When a new envelope is received, the envelope status and recipient status must be either *Sent* or *Delivered*, and a notification is generated and sent to the associated recipient.
Add Recipient page	Provides a form to add a recipient to an envelope. Contains input fields such as Name, Email, and Role.
Edit Recipient page	Provides a form to edit input fields such as name and email. When a user modifies a name or email, a **Save and Resend** button is displayed and an email is sent to the recipient.
Envelope Details page	Provides envelope details such as email subject, created on, and status. Also shows a list of recipients associated with the envelope, and an **Add Recipient** button. Users can't edit or add recipients to voided or declined envelopes.
My Sent Envelopes page	Provides the list of all envelopes sent by the user having a status (sent, delivered, completed, declined, voided) with a search bar to filter envelopes by email subject or envelope status.
Recipient Details page	Provides recipient details such as name, email, and status with these buttons: edit recipient, resend envelope, and delete recipient.

Notification or Page	Use-case workflows
Sign Envelope page	Provides details for an envelope pending signature, such as email subject, sender name, and expire after (number of days). Users can view the envelope in DocuSign using the **Sign Envelope** button to sign the envelope.
View Envelope page	Provides envelope details such as email subject, created on, and status. Also contains a list of recipients associated to the envelope, and a **View Envelope** button.

Integrate Google Analytics

April 28, 2021

Integrate with Google Analytics to monitor traffic via your configured microapp. Acting as a news aggregator, Citrix Workspace Microapps allows users to track traffic in a single place, removing the need to manually check your analytics via website. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

For a comprehensive list of out-of-the-box Google Analytics microapps, see Use Google Analytics microapps.

Review prerequisites

These prerequisites assume you administer the Google Analytics instance of your organization to set up the integration. You must have these details to add the integration in Citrix Workspace Microapps:

- Google Analytics account
- Client ID and Secret
- View ID
- OAuth 2.0 authorization credentials

Add callback URLs

Add a custom URL to your instance configuration to grant access to private data and enable OAuth authenticated user actions. The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com.

Log in to Google Analytics as an admin and add the following authorized redirect URLs for this integration:

- `https://{ yourmicroappserverurl } /admin/api/external-services/com.sapho.services.googleanalytics.GoogleAnalyticsService/auth/serverContext`
- `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`

Add the integration to Citrix Workspace Microapps

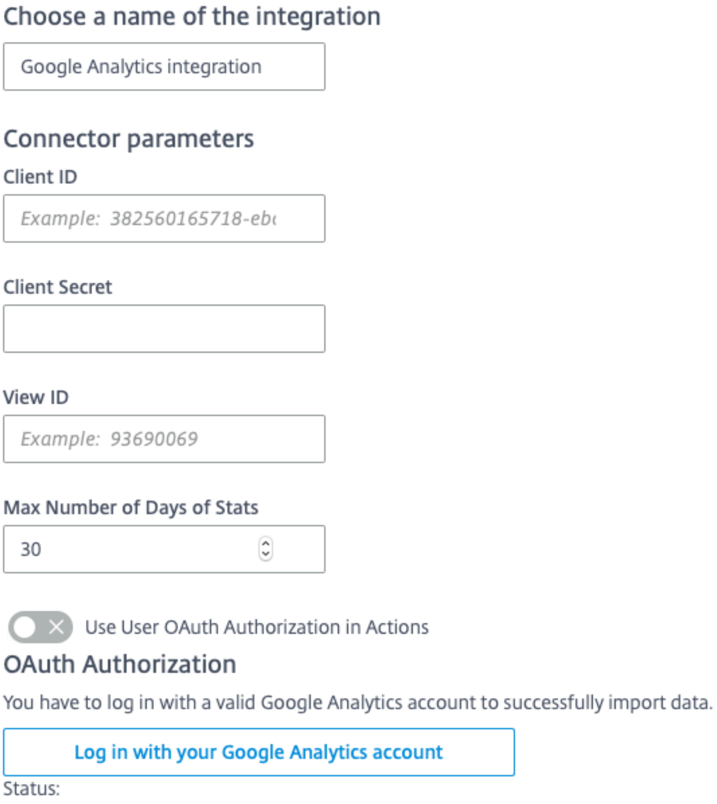
Add the Google Analytics integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Google Analytics tile.
4. Enter a name for the integration.



Choose a name of the integration

Google Analytics integration

Connector parameters

Client ID

Example: 382560165718-ebı

Client Secret

View ID

Example: 93690069

Max Number of Days of Stats

30

○ Use User OAuth Authorization in Actions

OAuth Authorization

You have to log in with a valid Google Analytics account to successfully import data.

Status:

5. Enter the **Connector parameters** that you collected in the previous procedures:
   - Enter the **Client ID** and **Client Secret**.
   - Enter the **View ID**.
   - Enter the **Max Number of Days of Stats** (default 30).
6. Toggle **Use User OAuth Authorization in Actions** if necessary.
7. Select **Log in with your Google Analytics account** to enable OAuth Authorization. A Google sign-in page opens in a new tab. You are prompted to enter an account name, enter a password, and confirm access.
8. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization.

For more information, see Verify needed entities and Set data synchronization in the Configure the integration article.

For more details of API endpoints and table entities, see Google Analytics connector specifications.

Use Google Analytics microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our Google Analytics integration comes with the following preconfigured out-of-the-box microapps:

Session Metrics: View sessions and session metrics.

Notification or Page	Use-case workflows
Daily Sessions Decreased notification	When the number of daily sessions decreases by a defined amount (by default, 10%), users receive a notification.
Daily Sessions Increased notification	When the number of daily sessions increases by a defined amount (by default, 10%), users receive a notification.
Weekly Sessions Decreased notification	When the number of weekly sessions decreases by a defined amount (by default, 10%), users receive a notification.
Weekly Sessions Increased notification	When the number of weekly sessions increases by a defined amount (by default, 10%), users receive a notification.
Sessions page	Provides a list of all sessions.

User Metrics: View user metrics.

Notification or Page	Use-case workflows
Daily Users Decreased notification	When the number of daily users decreases by a defined amount (by default, 10%), users receive a notification.
Daily Users Increased notification	When the number of daily users increases by a defined amount (by default, 10%), users receive a notification.
Weekly Users Decreased notification	When the number of weekly users decreases by a defined amount (by default, 10%), users receive a notification.

Notification or Page	Use-case workflows
Weekly Users Increased notification	When the number of weekly users increases by a defined amount (by default, 10%), users receive a notification.
Users page	Provides a list of all user metrics.

Integrate Google Calendar

April 28, 2021

Deploy the Google Calendar integration to schedule calendar events and list events and office hours of a user from any device or intranet. This integration delivers these four use cases:

> **Note:**
>
> We provide two Google Calendar integration templates for your use. The java-based template provides more use-cases, including notifications. This is under the **Integrations** category in the catalog. The set-up instructions follow immediately below. We also have a new HTTP-based template under the **Citrix Labs** category in the catalog. This integration is more flexible and provides more power to configure the cached data structure. For more information, see Citrix Labs Google Calendar integration.

For comprehensive details of out-of-the-box microapps for Google Calendar, see Use Google Calendar microapps.

> **Note:**
>
> We want your feedback! Please provide feedback for this preview integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

Review prerequisites

These prerequisites assume you administer the Google Calendar instance of your organization to set up the integration.

- This integration requires a dedicated Google account which is used to synchronize calendar data with Workspace. This account must have Admin API privilege Users/Read or a standard Admin role which includes this privilege.
- If your internal server hosting Workspace is behind a firewall, you must allow access to host name www.google.com with port 443, so Workspace can connect.

- Obtain a new oauth2 client_id and client_secret and define the scope of client's application.
- Configure Citrix Gateway to support single sign-on for Google Calendar so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Google CalendarSingle Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

You must have these details to add the Google Calendar integration in Citrix Workspace Microapps:

- OAUTH Private Key JSON
- Impersonated Admin User account

For User Consent (3LO) Authentication for Google Calendar:

- Client ID
- Client Secret

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for Google Meet so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service https://docs.citrix.com/en-us/citrix-gateway-service/.

The integration requires regular access to your Google Calendar instance, so we recommend creating a dedicated user account. This account must have the following permissions:

- Permissions required for Service Account: Full administrator privileges

- Scopes required for Service Account:

  ```
  https://www.googleapis.com/auth/calendar
  https://www.googleapis.com/auth/calendar.events
  https://www.googleapis.com/auth/admin.directory.user
  ```

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Google API limitation form link: https://developers.google.com/calendar/pricing
- Recommended plan: Enterprise Plus

Create a new service account

Sign in here: https://workspace.google.com/intl/en/pricing.html

How to enable APIs

Google Calendar APIs are enabled for access via web services for paid accounts by default.

Configure OAuth

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **ENABLE APIS AND SERVICES**, and enable all the required APIs from Google Workspace . Recommended APIs: **Google Calendar API** and **Admin SDK**.
5. Go back to the **APIs and Services** page, and select **OAuth consent screen** on the left screen.
6. Select the **User Type** according to your requirement (we recommend: **Internal**), and then select **Create**.
7. Complete the required fields including **Scopes required for Service Account**, and save the details. These are the required scopes:
   - https://www.googleapis.com/auth/calendar
   - https://www.googleapis.com/auth/calendar.events
   - https://www.googleapis.com/auth/admin.directory.user

Configure callback URL server

Configure the OAuth server to read data through the Google Calendar integration.

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **Credentials** on the left screen.
5. Select **CREATE CREDENTIALS**, and select **OAuth client ID** from the list.
6. Select **Web Application** from the **Application type** list, and enter a **Name**.
7. Select **ADD URI** under **Authorized redirect URIs**.
8. Enter the following authorized redirect URLs for this integration in the **URIs** field:
   - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
9. Select **Create**.

10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Authentication** while configuring the integration.

Configure callback URL client

Configure the OAuth client for writing back data through the Google Calendar integration.

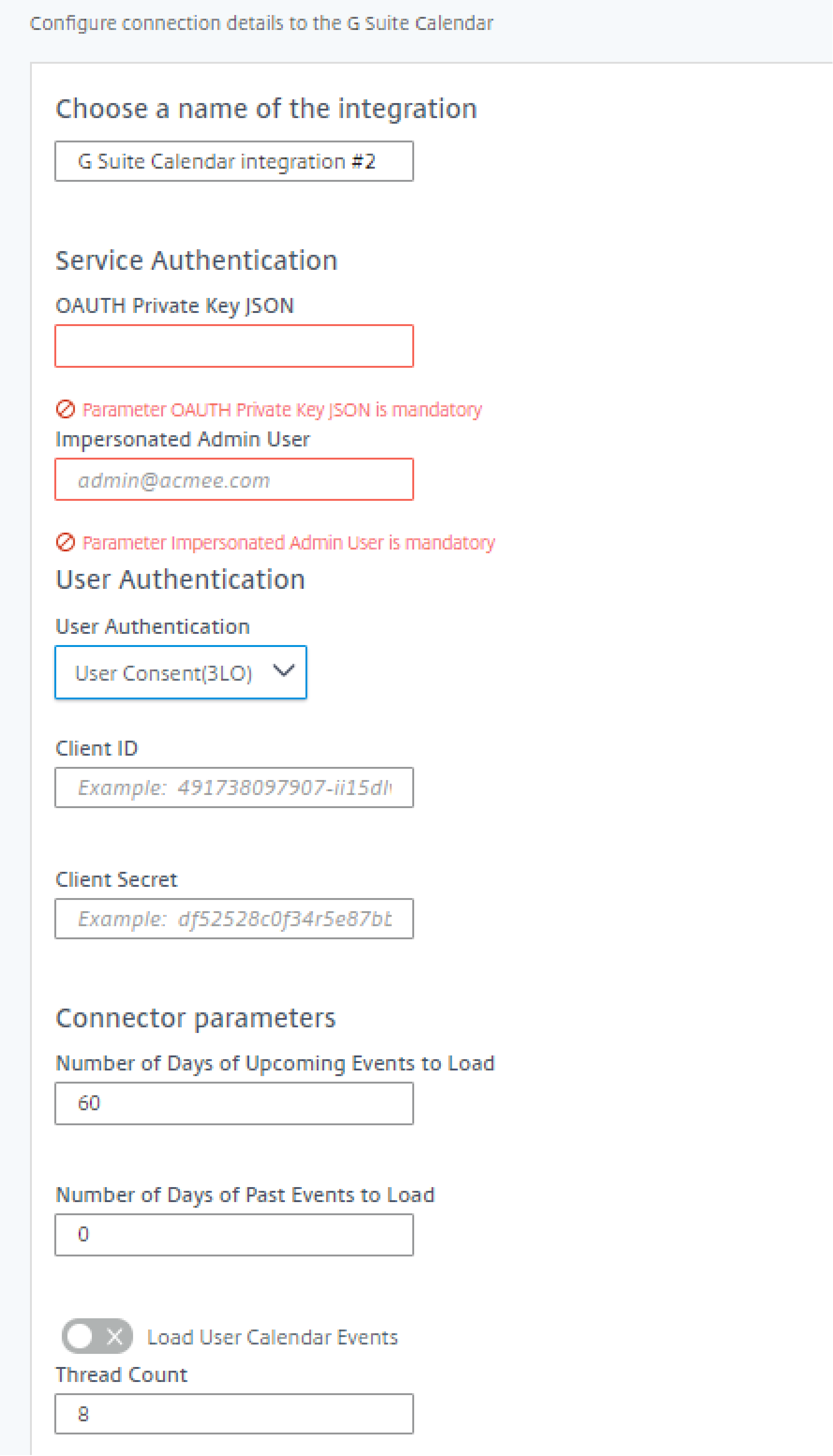
1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **Credentials** on the left screen.
5. Select **CREATE CREDENTIALS**, and select **OAuth client ID** from the list.
6. Select **Web Application** from the **Application type** list, and enter a **Name**.
7. Select **ADD URI** under **Authorized redirect URIs**.
8. Enter the following authorized redirect URLs for this integration in the **URIs** field:
   - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
9. Select **Create**.
10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

Add the Google Calendar integration

Follow these steps to set up the Google Calendar integration:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.
2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Google Calendar tile from the **Integrations** category of the catalog.
4. Enter a name for the integration.



Configure connection details to the G Suite Calendar

Choose a name of the integration

G Suite Calendar integration #2

Service Authentication

OAUTH Private Key JSON

⊘ Parameter OAUTH Private Key JSON is mandatory

Impersonated Admin User

admin@acmee.com

⊘ Parameter Impersonated Admin User is mandatory

User Authentication

User Authentication

User Consent(3LO)

Client ID

Example: 491738097907-ii15dl

Client Secret

Example: df52528c0f34r5e87bt

Connector parameters

Number of Days of Upcoming Events to Load

60

Number of Days of Past Events to Load

0

Load User Calendar Events

Thread Count

8

5. Enter the **Service Authentication** parameters that you collected in the previous procedures.

   - Copy and paste the entire **OAUTH Private Key JSON**. Copy the whole key, including the {} brackets.

    - Enter the **Impersonated Admin User**.

6. Select a **User Authentication** method.
    - **Admin**
    - **User**
    - **User Consent (3LO)** The resource owner allows access.
7. For User Consent (3LO), enter the **Client ID** and **Client Secret** that you collected in the prerequisites procedure.
8. Enter **Connector Parameters**.
    - **Number of Days of Upcoming Events to Load** - Defines the length of time to cache upcoming calendar events to send notifications.
    - **Number of Days of Past Events to Load** - Defines the length of time to cache past events.
    - Select the **Load User Calendar Events** radio button if necessary.
    - **Thread Count** - Enter a value.
9. Select **Add**.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Google Calendar connector specifications.

Use Google Calendar microapps

Our Google Calendar integration comes with the following preconfigured out-of-the-box microapps.

Calendar Events: Create and preview events.

Notification or Page	Use-case workflows
Event Reminder notification	When an event is upcoming, all subscribers receive a reminder notification.
All Events page	Provides a personalized list of upcoming events.
Create Event page	Provides a form for adding a new event with details.
Event Detail page	Provides a detailed view of an event including a list of guests.

Citrix Labs Google Calendar integration

Deploy the Google Calendar integration to schedule calendar events and list events and office hours of a user from any device or intranet.

> **Note:**
>
> This integration template is in Preview and marked as **Preview** in the list of available templates that are shown in the product when adding a new integration. While in Preview, there is no commitment to support, and support is provided by the developer on a best-effort basis. Preview integration templates are shared for the purpose of testing and validation. We do not advise deploying them in production environments. For more information, see Maintenance statement for Microapps integration templates.

This integration delivers these four use cases:

- With the Create Event microapp, users can host one-time or recurring meetings, add invitees, and select different timezones. The microapp also follows up with an email to all invitees with the corresponding event object for easy calendar integration.
- With the My Calendar (Current Month) microapp, users can view all upcoming events for the current month.
- With the My Office Hours microapp, users can set up virtual office hours or list all the current month office hours of the user.

Review prerequisites

These prerequisites assume that administrator is a part of the Google Calendar integration set up of the organization. This Google Calendar admin account must have full read privileges for user information. After you set up this integration with Google Calendar, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- BASE URL: `https://www.googleapis.com/`
- TOKEN URL: `https://oauth2.googleapis.com/token`
- AUTHORIZATION URL: `https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&prompt=consent`
- CLIENT ID: The client ID is the string representing client registration information unique to the authorization server.
- SECRET: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with

> your configured microapp.

Configure Citrix Gateway to support single sign-on for Google Meet so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service https://docs.citrix.com/en-us/citrix-gateway-service/.

The integration requires regular access to your Google Calendar instance, so we recommend creating a dedicated user account. This account must have the following permissions:

- Permissions required for Service Account: Full administrator privileges
- Scopes required for Service Account:

  ```
  https://www.googleapis.com/auth/calendar
  https://www.googleapis.com/auth/calendar.events
  https://www.googleapis.com/auth/admin.directory.user
  ```

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Google API limitation form link: https://developers.google.com/calendar/pricing
- Recommended plan: Enterprise Plus

Create a new service account

Sign in here: https://workspace.google.com/intl/en/pricing.html

Enable APIs

Google Calendar APIs are enabled for access via web services for paid account by default.

Configure OAuth

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **ENABLE APIS AND SERVICES**, and enable all the required APIs from Google Workspace . Recommended APIs: **Google Calendar API** and **Admin SDK**.
5. Go back to the **APIs and Services** page, and select **OAuth consent screen** on the left screen.
6. Select the **User Type** according to your requirement (we recommend: **Internal**), and then select **Create**.

7. Complete the required fields including **Scopes required for Service Account**, and save the details. These are the required scopes:
    - https://www.googleapis.com/auth/calendar
    - https://www.googleapis.com/auth/calendar.events
    - https://www.googleapis.com/auth/admin.directory.user

Configure callback URL server

Configure the OAuth server to read data through the Google Calendar integration.

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **Credentials** on the left screen.
5. Select **CREATE CREDENTIALS**, and select **OAuth client ID** from the list.
6. Select **Web Application** from the **Application type** list, and enter a **Name**.
7. Select **ADD URI** under **Authorized redirect URIs**.
8. Enter the following authorized redirect URLs for this integration in the **URIs** field:
    - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
9. Select **Create**.
10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Authentication** while configuring the integration.

Configure callback URL client

Configure the OAuth client for writing back data through the Google Calendar integration.

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **Credentials** on the left screen.
5. Select **CREATE CREDENTIALS**, and select **OAuth client ID** from the list.
6. Select **Web Application** from the **Application type** list, and enter a **Name**.
7. Select **ADD URI** under **Authorized redirect URIs**.

8. Enter the following authorized redirect URLs for this integration in the **URIs** field:
   - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
9. Select **Create**.
10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

Add the Citrix Labs integration to Citrix Workspace Microapps

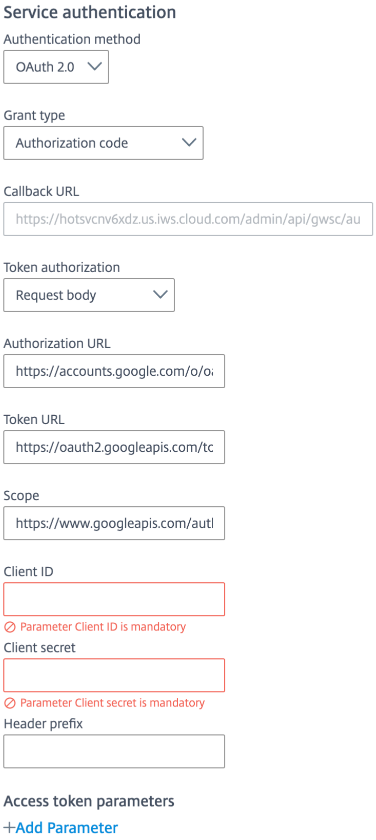
Add the Preview Google Calendar integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Google Calendar tile from the **Preview** category of the catalog.
3. Enter a name for the Integration.
   - Enter the instance **Base URL**: `https://www.googleapis.com/`.
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.
4. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
   a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.
   b) Select **Request body** from the **Token authorization** menu.
   c) The **Authorization URL** is prefilled: `https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&prompt=consent`
   d) The **Token URL** is prefilled: `https://oauth2.googleapis.com/token`
   e) Ensure the following is entered for Scope: `https://www.googleapis.com/auth/calendar https://www.googleapis.com/auth/calendar.events https://www.googleapis.com/auth/admin.directory.user`

f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service authentication

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://hotsvcnv6xdz.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Request body

Authorization URL: https://accounts.google.com/o/oa

Token URL: https://oauth2.googleapis.com/tc

Scope: https://www.googleapis.com/autl

Client ID:

Parameter Client ID is mandatory

Client secret:

Parameter Client secret is mandatory

Header prefix:

Access token parameters

5. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.

   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.

   c) Select **Request body** from the **Token authorization** menu.

   d) The **Authorization URL** is prefilled: `https://accounts.google.com/o/oauth2/v2/`

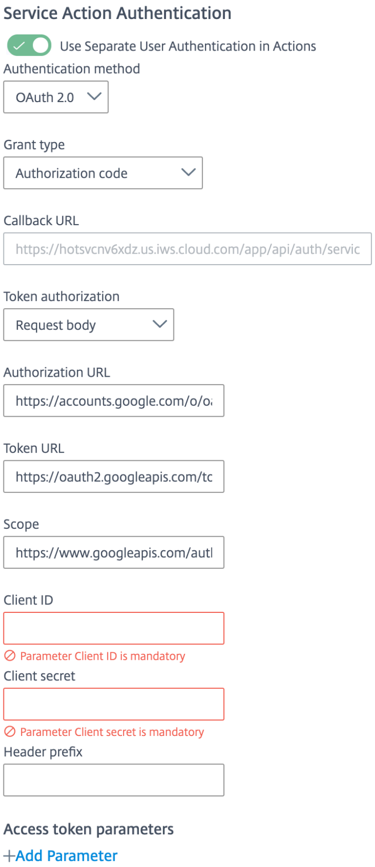
auth?access_type=offline&prompt=consent

e) The **Token URL** is prefilled: `https://oauth2.googleapis.com/token`

f) Ensure the following is entered for Scope: `https://www.googleapis.com/auth/calendar https://www.googleapis.com/auth/calendar.events`

g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.

h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service Action Authentication

Use Separate User Authentication in Actions

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://hotsvcnv6xdz.us.iws.cloud.com/app/api/auth/servic

Token authorization: Request body

Authorization URL: https://accounts.google.com/o/o

Token URL: https://oauth2.googleapis.com/tc

Scope: https://www.googleapis.com/autl

Client ID:

Parameter Client ID is mandatory

Client secret:

Parameter Client secret is mandatory

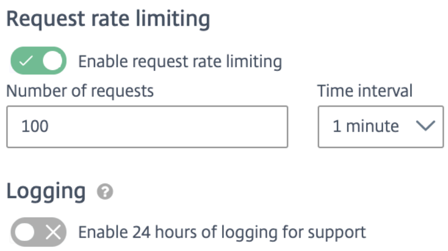
Header prefix:

Access token parameters

+Add Parameter

6. Enable the **Enable request rate limiting** toggle. Enter *100* for **Number of requests** and *1 minute* for **Time interval**.

7. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

Enable request rate limiting: on

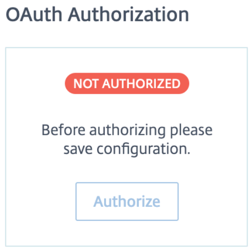
Number of requests: 100

Time interval: 1 minute

Logging

Enable 24 hours of logging for support: off

8. Select **Save** to proceed.
9. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Google login screen.
   a) Enter your Service Account user name and password and select **Log in**.
   b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

> **Note:**
>
> It is recommended to set the "Full Synchronization" interval as **Weekly** to remove the canceled or deleted events from Microapps platform and subsequently from the user's calendar.

For more details of API endpoints and table entities, see Preview Google Calendar connector specifications.

Use Labs Google Calendar microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note:**
>
> Since the currently available time zones are hardcoded in the **Create Event**, **My Office Hours** and **My Calendar (Current Month)** microapp, the addition of any other time zone would require the admin to add them manually.

Create Event: Schedule an event according to user preferences.

Notification or Page	Use-case workflows
Create Event page	Provides a form to schedule an event with these details: Event Title, Google Meet Video link, Start and End Time, Time Zone, Recurrence (One Time, Daily, Weekly, Monthly), Description, Location, and Event Attendees.

My Calendar (Current Month): View list of the upcoming current month's one-time and recurring calendar events and the ability to edit the events.

Notification or Page	Use-case workflows
Event Notification Details page	Detail page which shows all appropriate details of an event and **Join now** button to start the meeting. User can also redirect to Google Calendar by selecting **Open My Calendar**.
Load Events page	Provides a list of one-time and recurring calendar events available for the user with **Refresh List** button to refresh the list. Users can view event details by clicking on the events available in the list.
List Events page	Provides up to date list of one-time and recurring calendar events available for the user. Users can view event details by clicking on the events available in the list.
Event Details page	Detail page which shows all the appropriate details of the event. User can start/join the meeting by selecting **Join Event** and users can redirect to Google Calendar by selecting **Open My Calendar**.

Notification or Page	Use-case workflows
All Day Event Details page	Detail page which shows all the appropriate details of all day events. User can start (join) the meeting by selecting **Join Event** and users can redirect to Google Calendar by selecting **Open My Calendar**.
Edit OneTime Event page	Form page to edit the one-time event.
Edit Recurring Event page	Form page to edit the existing recurring event.

My Office Hours: Create, edit, and view office hours.

> **Note:**
>
> The refresh button is used to sync the cache with the most recent data, in lieu of full or incremental synchronization.

Notification or Page	Use-case workflows
Create Office Hours page	Provides a form to create office hours with these details: Start and End Time, Time Zone, Recurrence (One Time, Daily, Weekly, Monthly).
My Office Hours List page	Provides a list of user office hours available with a **Refresh List/Load My Hours** button to update/refresh the list. User can add office hours by selecting **Add Office Hours**.
Edit My Office Hours page	Form page to edit user office hours.

Integrate Google Directory

April 28, 2021

Integrate with Google Directory to share employee contact information with your entire organization on any device, intranet, or messenger. Ensure you meet the prerequisites, enable the APIs, and create the service account. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

> **Note:**
>
> We provide two Google Directory integration templates for your use. We recommend using the newer HTTP integration for most use-cases, specifically Google Directory workflows. The HTTP integration provides more power to configure the cached data structure. At the end of this article you can find documentation for the Legacy Google Directory integration template. For details of the Google Calendar integration, see Integrate Google Calendar.

Review prerequisites

These prerequisites assume you administer the Google Directory instance of your organization to set up the integration.

- This integration requires a dedicated Google account which is used to synchronize calendar data with Workspace. This account must have Admin API privilege Users/Read or a standard Admin role which includes this privilege.
- If your internal server hosting Workspace is behind a firewall, you must allow access to host name www.google.com with port 443, so Workspace can connect.
- Obtain a new oauth2 client_id and client_secret and define the scope of client's application.
- Configure Citrix Gateway to support single sign-on for Google Directory so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Google Directory Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

You must have these details to add the Google Directory integration in Citrix Workspace Microapps:

- Client ID
- Client Secret
- Domain
- Valid Google Directory account and password

Enable APIs

Enable the APIs for the services you require.

Follow these steps:

1. Log in to https://console.developers.google.com with an administrator account and select **Create** to create a new project. You can also update an existing project.
2. Select **Enable APIs and Services** and search for **Admin SDK**. Select it and select **Enable**.
3. Search for the **Google Calendar API**. Select it and select **Enable**.

Create service account

1. Select the **Settings** icon at the top left, mouseover **IAM & admin**, and select **Service accounts**.
2. Select **CREATE SERVICE ACCOUNT**.
3. Enter your **Service account name**, a **Service account ID** (by default, automatically generated), a **Service account description**, and click **CREATE**.
4. Select the **Select a role** menu, and choose an **Owner Role**.
5. Select **Continue** and then select **Done**.

Enable Google delegation and create Service Account Key

To enable Google domain-wide delegation and create a service account key follow these steps:

1. In your service account list, find the account you created. Select **Actions > Edit**.
2. Select **Show domain-wide delegation**. Select the **Enable Google Domain-wide Delegation** check box.
3. To create your private key, select **+Create key**, select **JSON**, and select **CREATE**.

   A private key is saved to your computer.
4. Store the JSON file in a secure location. It is required when you configure the Calendar integration.
5. Select **CLOSE** and select **SAVE**.

Enable and manage API access

1. Navigate to https://admin.google.com, select **Security > API reference**. Ensure **Enable API access** is selected.
2. Select **Advanced settings > Manage API client access**. Add the **Service account name** into the list of Authorized API clients.
3. Under **Client Name**, enter the **client_id** from the private key JSON file that you downloaded.
4. Enter the following comma delimited list of scopes into the **One or More API Scopes** field:

~~~
```,```<https://www.googleapis.com/auth/admin.directory.user.
readonly><!--NeedCopy-->
~~~

5. Select **Authorize**.

Add callback URLs to Google API Console

Grant access to private data and provide a link to terms of service and privacy policy. The callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com.

1. Go to https://console.developers.google.com and log in using your credentials.
2. Select **OAuth consent screen** from the left navigation.
3. Under **Authorized domains**, add this domain: `cloud.com`, press return, and select **Save**.
4. To create an OAuth client ID, select **Credentials** from the left navigation. Select **Create credentials** and **Oauth Client ID**.
5. Select **Web application** and add the following URIs following the style of those previously added to allow access to private data and enable OAuth authenticated user actions:

   Authorized redirect URLs:

   ```
   https://{ yourmicroappserverurl } /admin/api/external-services/com.
   sapho.services.googlecalendar.GoogleCalendarService/auth/serverContext,
   https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback
   ```

   For Google Directory, use:

   ```
   https://{ yourmicroappserverurl } /admin/api/external-services/com.
   sapho.services.googleforwork.GoogleForWorkService/auth/serverContext,
   https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback
   ```

6. After adding each URL, press Enter. After adding all desired URIs, scroll down, and select **Create**.

   > **Note:**
   >
   > If you do not have access, give yourself permissions to accept OAuth permissions. Go to **Admin console > Security > API Permissions**. Under **Internal App Settings**, select the **Trust domain owned apps** check box.

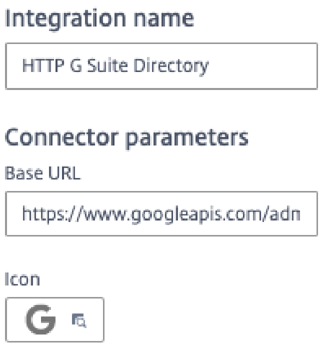
Add the integration to Citrix Workspace Microapps

Add the Google Directory integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace. We provide two Google Directory integration templates for your use. We recommend using the newer HTTP integration for most use-cases.

Follow these steps to set up the Google Directory HTTP integration. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Google Directory tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL** or simply replace { `customer-id` } in the example with your customer ID.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default Google Directory icon.



Integration name

HTTP G Suite Directory

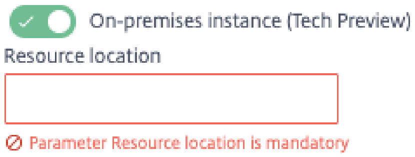
Connector parameters

Base URL

https://www.googleapis.com/adn

Icon

    - Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.



On-premises instance (Tech Preview)

Resource location

Parameter Resource location is mandatory

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

    a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization

server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application

b) Select **Authorization header** from the **Token authorization** menu.

c) Enter your **Authorization URL** or simply replace `{ customer-id }` in the example with your customer ID. This is the authorization server URL provided when setting up the target application integration.

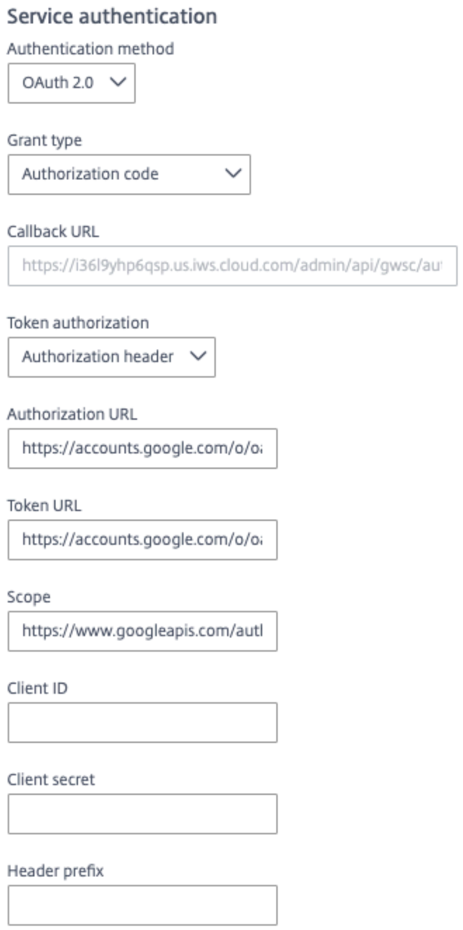
d) Enter your **Token URL** or simply replace `{ customer-id }` in the example with your customer ID. This is the URL of the access authorization token.

e) Ensure the following is entered for **Scope**. To synchronize additional entities, you must add scopes here. Use the following, separated by a space: `https://www.googleapis.com/auth/admin.directory.user https://www.googleapis.com/auth/admin.directory.orgunit https://www.googleapis.com/auth/admin.directory.group`.

f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Google account. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

h) Enter your **Header prefix**. (optional) Enter the header prefix if your bearer prefix is different from the default header.



Service authentication

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Authorization header

Authorization URL: https://accounts.google.com/o/o

Token URL: https://accounts.google.com/o/o

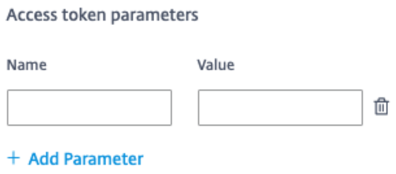
Scope: https://www.googleapis.com/autl

Client ID:

Client secret:

Header prefix:

i) If you selected **OAuth 2.0** authentication method, you can select **+ Add Parameter** to include **Access token parameters**. Access token parameters define the access token parameters as required by the target application authorization server if necessary.



Access token parameters

Name:

Value:

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

    a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.

    b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application

c) Select **Authorization header** from the **Token authorization** menu.

d) Enter your **Authorization URL** or simply replace `{ customer-id }` in the example with your customer ID. This is the authorization server URL provided when setting up the target application integration.

e) Enter your **Token URL** or simply replace `{ customer-id }` in the example with your customer ID. This is the URL of the access authorization token.

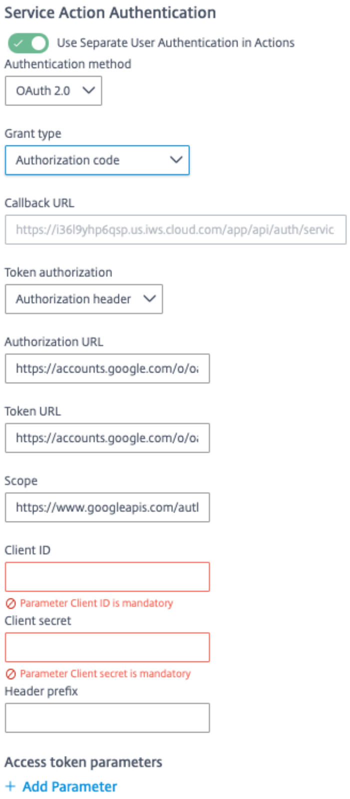
f) Ensure the following is entered for **Scope**. To synchronize additional entities, you must add scopes here. Use the following: `https://www.googleapis.com/auth/admin.directory.user`.

g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Google account. You need to add the **Callback URL** you see on the integration configuration page.

h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

i) (Optional) Enter your **Header prefix** if your bearer prefix is different from the default header.

j) If you selected **OAuth 2.0** authentication method, you can select **+ Add Parameter** to include **Access token parameters**. Access token parameters define the access token parameters as required by the target application authorization server if necessary.



Service Action Authentication

Use Separate User Authentication in Actions

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Authorization header

Authorization URL: https://accounts.google.com/o/o:

Token URL: https://accounts.google.com/o/o:

Scope: https://www.googleapis.com/autl

Client ID:

Parameter Client ID is mandatory

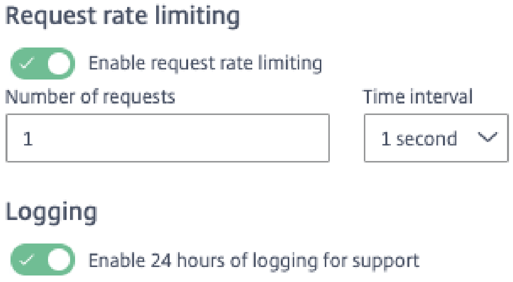
Client secret:

Parameter Client secret is mandatory

Header prefix:

Access token parameters

7. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

Enable request rate limiting

Number of requests: 1

Time interval: 1 second

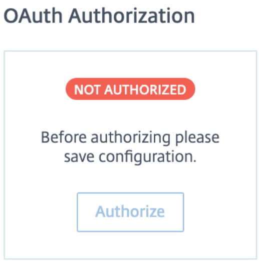
Logging

Enable 24 hours of logging for support

9. Select **Save** to proceed.
10. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Webex login screen.

    a) Enter your Service Account user name and password and select **Log in**.

    b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Google Directory connector specifications.

Use Google Directory microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs. Our Google Directory HTTP integration comes with the following preconfigured out-of-the-box microapps.

Google Directory connector specifications

Create User: Add a new user.

Notification or Page	Use-case workflows
Create User page	Provides a form for adding a new user with details.

Directory Admin: Manage users and details.

Notification or Page	Use-case workflows
Delete User page	Provides a form for removing a user.
Update User page	Provides a form for editing the details of a user.
User Detail page	Provides a detailed view of an employee with buttons to update or delete the user.

Notification or Page	Use-case workflows
Users page	Provides a searchable list of all employees with a link to individual user details.

Groups: View groups and details.

Notification or Page	Use-case workflows
Group Detail page	Provides a detailed view of a group.
Groups page	Provides a searchable list of all groups with a link to individual group details.

My Details: View your own details.

Notification or Page	Use-case workflows
My Details page	Provides a detailed, read-only view of a user's own employee details.

Users: View user details.

Notification or Page	Use-case workflows
New Employee notification	When a new teammate joins, all subscribers receive a notification.
User Detail page	Provides a detailed view of an employee with buttons to update or delete the user.
Users page	Provides a searchable list of all employees with a link to individual user details.

Legacy Google Directory integration

You must have these details to add the Google Directory integration in Citrix Workspace Microapps and review the prerequisites above:

- Client ID
- Client Secret

- Domain
- Valid Google Directory account and password

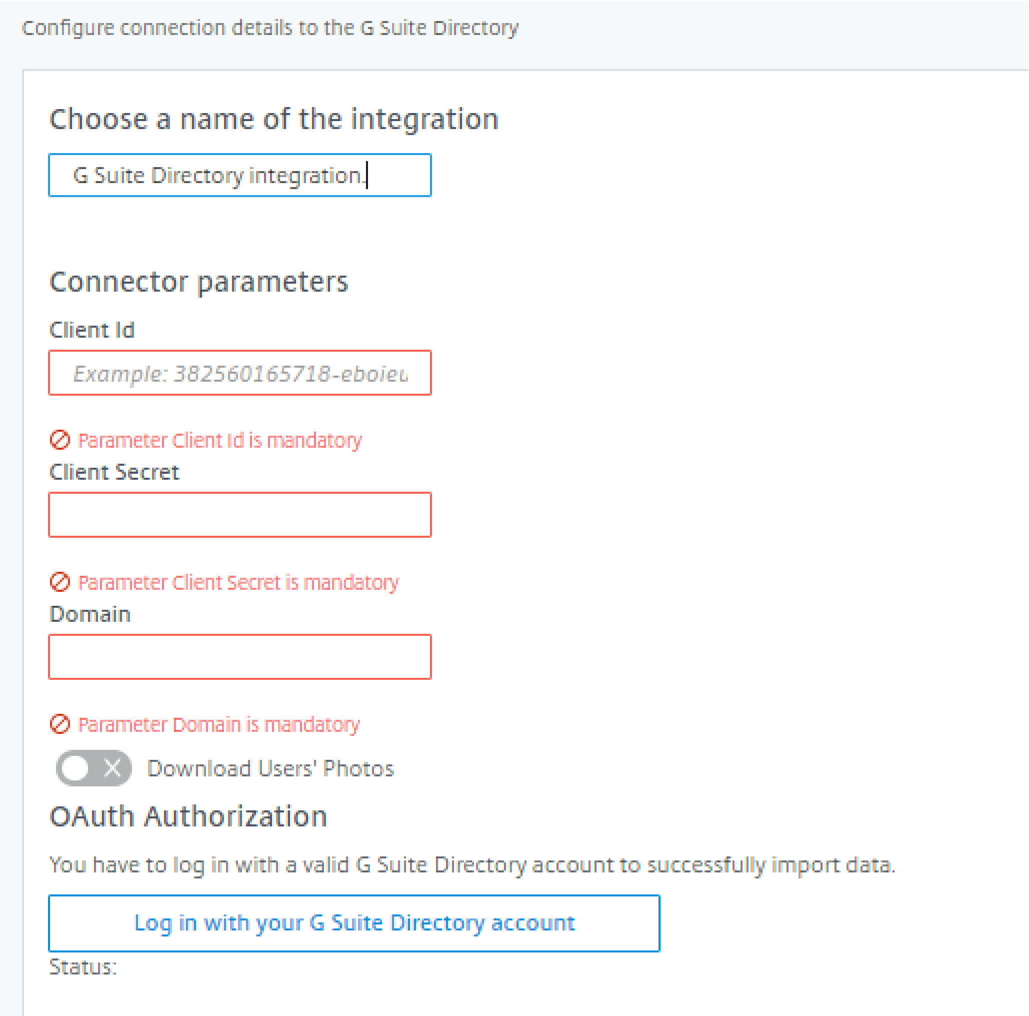
Add the Legacy Google Directory integration

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Google Directory tile.
4. Enter a name for the integration that you collected as prerequisites.



Configure connection details to the G Suite Directory

Choose a name of the integration

G Suite Directory integration

Connector parameters

Client Id

Example: 382560165718-eboieu

Parameter Client Id is mandatory

Client Secret

Parameter Client Secret is mandatory

Domain

Parameter Domain is mandatory

Download Users' Photos

OAuth Authorization

You have to log in with a valid G Suite Directory account to successfully import data.

Status:

5. Enter **Connector Parameters**.

   - Enter **Client Secret**.

- Enter **Domain**.
- Select the **Download Users' Photos** radio button if you want to cache users photos.

6. Select **Log in with your Google Directory account** to enable OAuth Authorization. A Google sign-in page opens in a new tab. You are prompted to enter an account name, confirm access, and enter a password.
7. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Google Directory connector specifications.

Legacy Google Directory microapps

Our Google Directory integration comes with the following preconfigured out-of-the-box microapps.

Directory Admin: Add a new user.

Notification or Page	Use-case workflows
Create User page	Provides a form for adding a new user with details.

Directory Details: View details of teammates, including new employees and position changes.

Notification or Page	Use-case workflows
New Employee notification	When a new teammate joins, all subscribers receive a notification.
Position Change notification	When the title of an employee changes, all subscribers receive a notification.
All Users page	Provides a list of all employees with a link to details.
User Detail page	Provides a detailed view of an employee.

Integrate Google Meet

April 20, 2021

Deploy the Google Meet integrations to schedule Google Meet meetings and list recordings from any device or intranet. This integration addresses two use cases:

- With the Create Meeting microapp, users can host one-time/recurring meetings, add invitees and select different timezones. There is also a follow-up email to all invitees with the corresponding meeting object for easy calendar integration.
- With the Meeting Recordings microapp, users can view all meeting recordings that they have access to.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapp for Google Meet, see Use Google Meet microapps.

Review prerequisites

These prerequisites assume that the administrator is part of the Google Meet integration set up of the organization. This Google Meet admin account must have full read privileges for user information. After you set up this integration with Google Meet, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- BASE URL: `https://www.googleapis.com/`
- TOKEN URL: `https://oauth2.googleapis.com/token`
- AUTHORIZATION URL: `https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&prompt=consent`
- CLIENT ID: The client ID is the string representing client registration information unique to the authorization server.
- SECRET: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for Google Meet so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service https://docs.citrix.com/en-us/citrix-gateway-service/.

The integration requires regular access to your Google Meet instance, so we recommend creating a dedicated user account. This account must have the following permissions:

- Permissions required for Service Account: Full administrator privileges
- Scopes required for Service Account:

  ```
  https://www.googleapis.com/auth/calendar
  https://www.googleapis.com/auth/calendar.events
  https://www.googleapis.com/auth/admin.directory.user
  ```

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Google API limitation form link: https://developers.google.com/calendar/pricing
- Recommended plan: Google Workspace Enterprise edition

Create a new service account

Sign in here: https://workspace.google.co.in/intl/en_in/pricing.html

Enable APIs

Google Meet APIs are enabled for access via web services for paid account by default.

Configure OAuth

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **ENABLE APIS AND SERVICES**, and enable all the required APIs from G-Suite . Recommended APIs: **Google Calendar API** and **Admin SDK**.
5. Go back to the **APIs and Services** page, and select **OAuth consent screen** on the left screen.
6. Select the **User Type** according to your requirement (we recommend: **Internal**), and then select **Create**.
7. Complete the required fields including **Scopes required for Service Account**, and save the details. These are the required scopes:
   - https://www.googleapis.com/auth/calendar

- https://www.googleapis.com/auth/calendar.events
- https://www.googleapis.com/auth/admin.directory.user

Configure callback URL server

Configure the OAuth server to read data through the Google Meet integration.

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **Credentials** on the left screen.
5. Select **CREATE CREDENTIALS**, and select **OAuth client ID** from the list.
6. Select **Web Application** from the **Application type** list, and enter a **Name**.
7. Select **ADD URI** under **Authorized redirect URIs**.
8. Enter the following authorized redirect URLs for this integration in the **URIs** field:
   - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
9. Select **Create**.
10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Authentication** while configuring the integration.

Configure callback URL client

Configure the OAuth client for writing back data through the Google Meet integration.

1. Log in with service account to: https://console.cloud.google.com
2. Select **APIs and Services** on left-hand menu.
3. Select the appropriate project from the project list on the navigation menu.
4. Select **Credentials** on the left screen.
5. Select **CREATE CREDENTIALS**, and select **OAuth client ID** from the list.
6. Select **Web Application** from the **Application type** list, and enter a **Name**.
7. Select **ADD URI** under **Authorized redirect URIs**.
8. Enter the following authorized redirect URLs for this integration in the **URIs** field:
   - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`

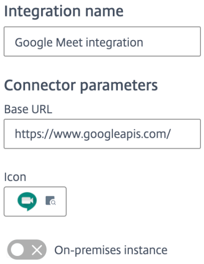
9. Select **Create**.
10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

Adding the integration to Citrix Workspace Microapps

Add the Google Meet integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Google Meet tile.
3. Enter a name for the Integration.
   - Enter the instance **Base URL**: `https://www.googleapis.com/`.
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

Google Meet integration

Connector parameters

Base URL

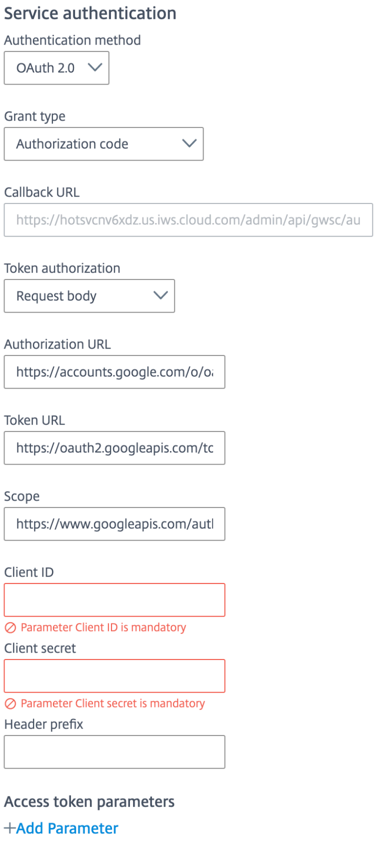
https://www.googleapis.com/

Icon

On-premises instance

4. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
   a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.
   b) Select **Request body** from the **Token authorization** menu.
   c) The **Authorization URL** is prefilled: `https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&prompt=consent`

d) The **Token URL** is prefilled: `https://oauth2.googleapis.com/token`
e) Ensure the following is entered for Scope: `https://www.googleapis.com/auth/calendar https://www.googleapis.com/auth/calendar.events https://www.googleapis.com/auth/admin.directory.user`
f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.
g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service authentication

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://hotsvcnv6xdz.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Request body

Authorization URL: https://accounts.google.com/o/o

Token URL: https://oauth2.googleapis.com/tc

Scope: https://www.googleapis.com/autl

Client ID:

Parameter Client ID is mandatory

Client secret:

Parameter Client secret is mandatory

Header prefix:

Access token parameters

+Add Parameter

5. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type en-

ables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.

c) Select **Request body** from the **Token authorization** menu.

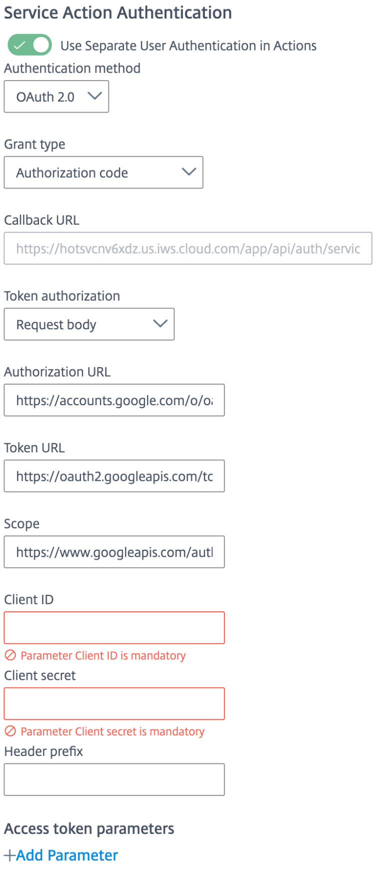
d) The **Authorization URL** is prefilled: `https://accounts.google.com/o/oauth2/v2/auth?access_type=offline&prompt=consent`

e) The **Token URL** is prefilled: `https://oauth2.googleapis.com/token`

f) Ensure the following is entered for Scope: `https://www.googleapis.com/auth/calendar` `https://www.googleapis.com/auth/calendar.events`

g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.

h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service Action Authentication

✓ Use Separate User Authentication in Actions

Authentication method

OAuth 2.0

Grant type

Authorization code

Callback URL

https://hotsvcnv6xdz.us.iws.cloud.com/app/api/auth/servic

Token authorization

Request body

Authorization URL

https://accounts.google.com/o/oa

Token URL

https://oauth2.googleapis.com/tc

Scope

https://www.googleapis.com/autl

Client ID

⊘ Parameter Client ID is mandatory

Client secret

⊘ Parameter Client secret is mandatory

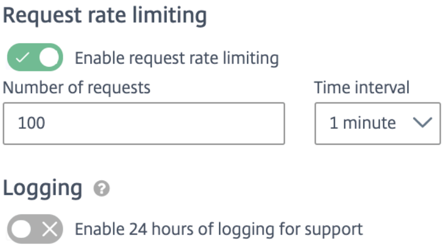
Header prefix

Access token parameters

+Add Parameter

6. Enable the **Enable request rate limiting** toggle. Enter *100* for **Number of requests** and *1 minute* for **Time interval**.

7. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

☑ Enable request rate limiting

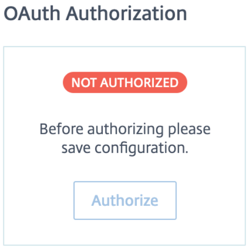
Number of requests: 100

Time interval: 1 minute

Logging

☐ Enable 24 hours of logging for support

8. Select **Save** to proceed.
9. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Google login screen.
   a) Enter your Service Account user name and password and select **Log in**.
   b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

> **Note:**
>
> The Citrix Google Meet integration uses **Data Update After Action** to pull in the most recent data for the logged in user via the **Refresh List** button in the **Meeting Recordings** service action. We recommend to use this approach as is. Utilize the full synchronization manually once every two (2) months for retaining an optimum amount of data for the user. Additionally, the integration doesn't support incremental synchronization and relies solely on **Data Update After Action** to pull in the most recent data.

For more details of API endpoints and table entities, see Google Meet connector specifications.

Use Google Meet microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Create Meeting: Schedule a meeting according to user preference.

Notification or Page	Use-case workflows
Create Meeting page	Provides a form to schedule a meeting with details such as meeting title, start and end time, time zone, recurrence (once, daily, weekly, monthly), password, and meeting attendees.

Meeting Recordings: View list of all meeting recordings available for the user and watch the recorded videos.

Notification or Page	Use-case workflows
Recordings page	Provides a list of meeting recordings available for the user and a **Refresh List** button to refresh the List. View the recording details by clicking on the recordings available from the list.
RecordingDetails page	A detailed page of the Meeting Recordings and a **Play Recording** option to watch the recorded videos. Click on the button to watch the recording

Integrate GoToMeeting

April 20, 2021

Deploy the GoToMeeting integration to schedule GoToMeetings from any device or intranet. With the **Create a Meeting** microapp from GoToMeeting, any user can host one-time meetings.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For

> any issues, our team will also monitor our dedicated forum on a daily basis.

For a comprehensive list of out-of-the-box GoToMeeting microapps, see Use GoToMeeting microapps.

Review prerequisites

These prerequisites assume that the administrator is part of the GoToMeeting integration set up of the organization. This GoToMeeting admin account must have full read privileges for user information.

After you set up this integration with GoToMeeting, you will need these artifacts to add the integration in Citrix Workspace Microapps, specifically the following list of parameters for setting up OAuth integration:

- **Base URL**: `https://api.getgo.com/`
- **Authorization URL**: `https://api.getgo.com/oauth/v2/authorize`
- **Token URL**: `https://api.getgo.com/oauth/v2/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server.
- **Secret**: The client secret is a unique string issued when setting up the target application integration.
- **Account ID**: This value replaces the `your_accountId` parameter in **All Organizers** endpoint. See Collect account ID.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for GoToMeeting so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.

The integration requires regular access to your GoToMeeting instance, so we recommend creating a dedicated user account. You can view the permission/privileges using https://goto-developer.logmeininc.com/admin/#section/Overview/Users-Roles-Licenses-and-Groups.

- Permissions required for Service Account: Full administrator privileges

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- GoTo Meeting API limitation form link: https://goto-developer.logmeininc.com/guides/FAQ/Ref-Rate-Limits/

Enable APIs

The GoTo Meeting APIs are enabled by default through web services for paid accounts.

Create a new service account

Sign up here: https://developer.goto.com/. For more information on new service accounts, see: https://support.goto.com/meeting/new-gotomeeting-guide.

Configure OAuth server

Configure the OAuth server to read data through the GoTo Meeting integration.

1. Log in with your service account to: https://developer.goto.com/GoToMeetingV1.
2. Select **OAuth Clients** in the top navigation bar.
3. Select **Create a new client**.
4. Fill in the details and enter the following authorized redirect URLs for this integration in the **Redirect URL** field:
   - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
5. Under **Scopes** section, select the **Scopes** check box.
6. Select **Save**.
7. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Authentication** while configuring the integration.

Configure OAuth client

Configure the OAuth client for writing back data through the GoTo Meeting integration.

1. Log in with your service account, as above: https://developer.goto.com/GoToMeetingV1.
2. Select **OAuth Clients** in the top navigation bar.
3. Select **Create a new client**.
4. Fill in the details and enter the following authorized redirect URLs for this integration in the **Redirect URL** field:
   - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
5. Under **Scopes** section, select the **Scopes** check box.

6. Select **Save**.

7. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

Collect account ID

Collect the account ID, and use this value to replace the `your_accountId` parameter in **All Organizers** endpoint.

1. Log in with admin credentials to https://admin.logmeininc.com/portal/
2. Look at the URL on the homepage. The account ID is found using this model: `https://admin.logmeininc.com/portal/##accounts/<accountID>/users?filterType=usertype`
3. Copy and save the account ID for use during set up process. See Replace Data Loading value.

Add the integration to Citrix Workspace Microapps

Add the GoToMeeting integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

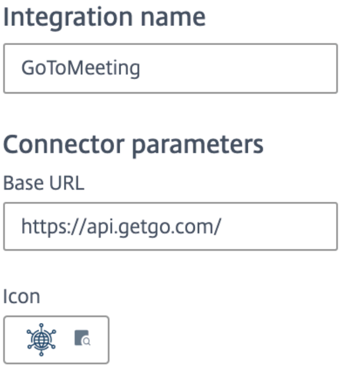
1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

2. Choose the GoToMeeting tile.

3. Enter an **Integration name** for the integration.

4. Enter **Connector parameters**.

   - Enter the instance **Base URL**: `https://api.getgo.com/`.
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

GoToMeeting

Connector parameters

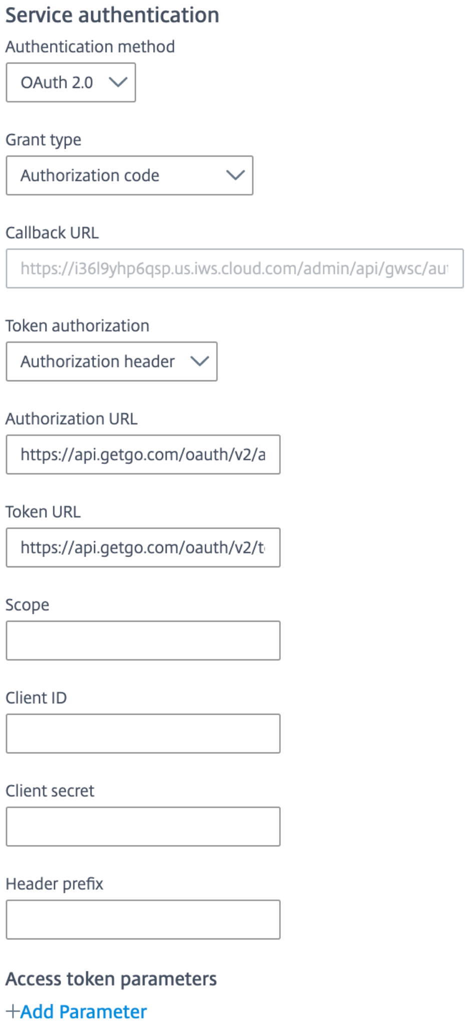
Base URL

https://api.getgo.com/

Icon

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

   a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.
   b) Select **Authorization header** from the **Token authorization** menu.
   c) The **Authorization URL** is prefilled: `https://api.getgo.com/oauth/v2/token`.
   d) The **Token URL** is prefilled: `https://api.getgo.com/oauth/v2/authorize`.
   e) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.
   f) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service authentication

Authentication method

OAuth 2.0

Grant type

Authorization code

Callback URL

https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/au

Token authorization

Authorization header

Authorization URL

https://api.getgo.com/oauth/v2/a

Token URL

https://api.getgo.com/oauth/v2/t

Scope

Client ID

Client secret

Header prefix

Access token parameters

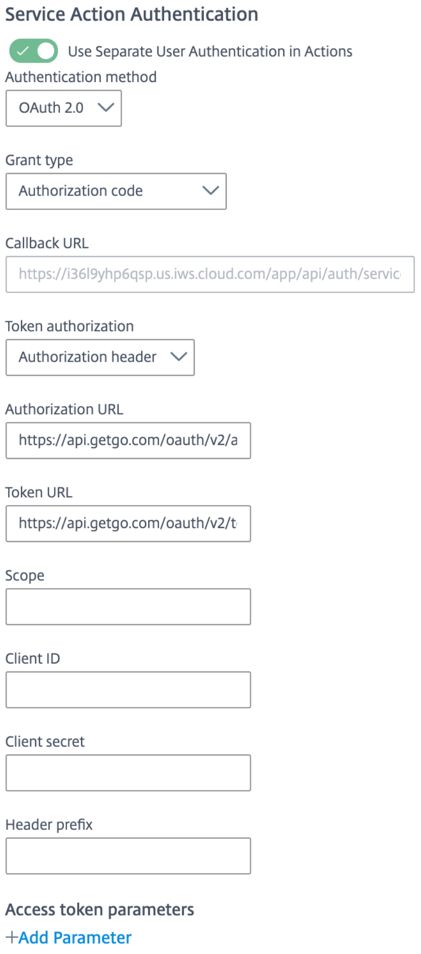
+Add Parameter

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.
   c) Select **Authorization header** from the **Token authorization** menu.
   d) The **Authorization URL** is prefilled: `https://api.getgo.com/oauth/v2/token`.
   e) The **Token URL** is prefilled: `https://api.getgo.com/oauth/v2/authorize`.
   f) Enter your **Client ID**. The client ID is the string representing client registration information

unique to the authorization server. You collect this and the secret when you configured the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service Action Authentication

✓ Use Separate User Authentication in Actions

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Authorization header

Authorization URL: https://api.getgo.com/oauth/v2/a

Token URL: https://api.getgo.com/oauth/v2/t

Scope:

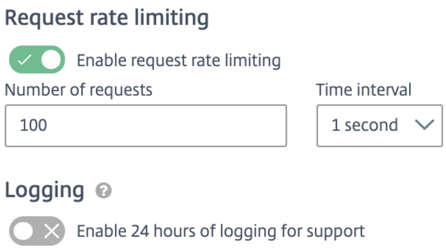
Client ID:

Client secret:

Header prefix:

Access token parameters

7. Enable the **Enable request rate limiting** toggle. Enter *100* for **Number of requests** and *1 minute* for **Time interval**.



Request rate limiting

✓ Enable request rate limiting

Number of requests: 100

Time interval: 1 second

Logging

✕ Enable 24 hours of logging for support

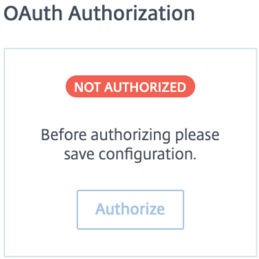
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

9. Select **Save** to proceed.

10. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Webex login screen.

    a) Enter your Service Account user name and password and select **Log in**.

    b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

Replace Data Loading value

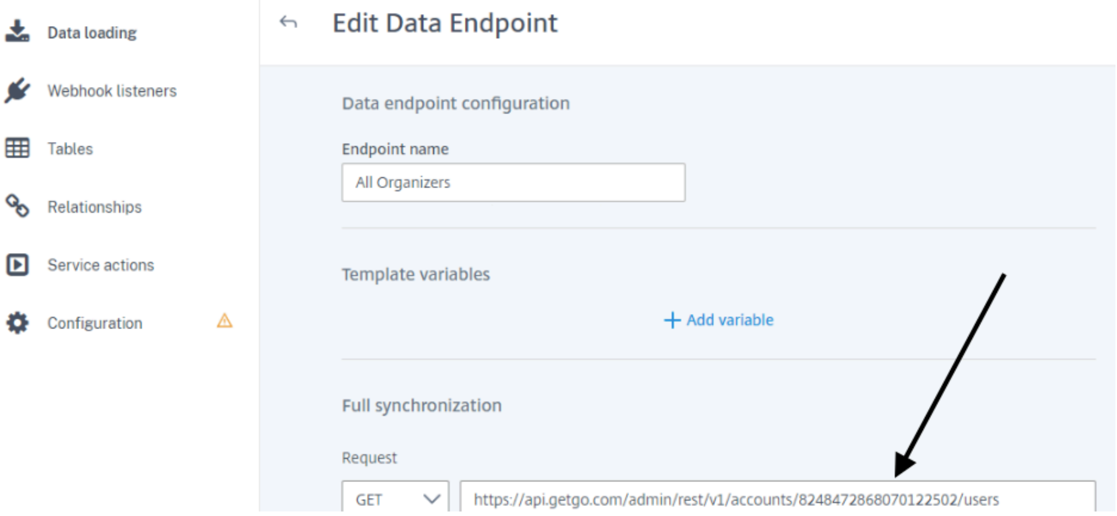
Replace the `your_accountId` parameter in **All Organizers** endpoint. Use the account ID value that you collected in Collect account ID.

1. From the **Microapp Integrations** page, select the menu next to the GoToMeeting integration, and then **Edit**. The **Data Loading** screen opens. If you are in the configuration screen, select **Data Loading** from the left side navigation column.

2. Select the menu next to the **All Organizers** endpoint and then select **Edit**, or select the name of the endpoint: **All Organizers**.

3. In the **Edit Data Endpoint** screen, find the value as shown in the screenshot. Replace this value with you account ID that you collected earlier.

4. Select **Apply** at the bottom of the screen and confirm.



Edit Data Endpoint

Data endpoint configuration

Endpoint name: All Organizers

Template variables

Full synchronization

Request: GET https://api.getgo.com/admin/rest/v1/accounts/8248472868070122502/users

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Ta-**

ble page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see GoToMeeting HTTP connector specifications.

Use GoToMeeting microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Create a Meeting: Schedule a meeting according to user preference. The user has the option to select date, start time and end time, password, and co-organizers.

Notification or Page	Use-case workflows
Create Meeting page	Provides a form to schedule a meeting with the following details according to user preference: Meeting Subject, Start and End Time, Password, and Co-Organizers for the meeting.
New Meeting page	Provides a success message along with option to view meeting details (with a View Details button) once a meeting is scheduled successfully.
Meeting Details page	Provides detailed information about the meeting scheduled by the user.

Integrate Ivanti

April 20, 2021

Deploy the Ivanti integration to submit and monitor incidents, service requests, and take action through Citrix Workspace. It also provides microapps for higher tier agents with the ability to review incidents. This integration creates requests using a delegated model that performs the action for incidents on behalf of the user. Service requests are implemented through deep links to Ivanti and allows users to create their own requests.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For

> any issues, our team will also monitor our dedicated forum on a daily basis.

Review prerequisites

A user account with global admin rights must be created with the last name *tenant* and the first name defined as the `TenantID` of your Ivanti implementation. See Create service account.

You need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://{ TenantID } .trysaasit.com/api/odata/businessobject/`
- **TenantID**: Find your `TenantID` using this model: `https://{ TenantID } .trysaasit.com`.
- **Token**: The token permits API access. See Collect access token.

Create user account

This account must be created with:

1. Enter the `TenantID` of your Ivanti implementation in the field: **First Name**. Find your `TenantID` using this model: `https://{ TenantID } .trysaasit.com`
2. Enter the string *tenant* in the field: **Last Name**.

Collect access token

Follow this process to obtain an access token. This access token is valid during its lifetime or until a new access token is requested.

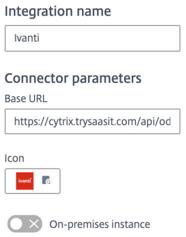
1. While logged in to your Ivanti instance as an administrator, select the setting icon (wrench) in the upper right corner of the screen.
2. Select **Security Controls** in the left navigation menu, and select **API Keys**.
3. Select "**Add Key Group**.
4. Enter a **Name** and a **Description** for the group, and select **Save Key Group**. Let's call our group *WSi Citrix*.
5. Select the newly created **WSi Citrix Key Group** and select **Add API Key**.
6. Set **On Behalf Of** to *Administrator Role*. Set **In Role** to *Administrator*. Select **Save Key**.
7. Copy the **Key** to be used as the token when you set up this integration.

Add the integration to Citrix Workspace Microapps

Add the Ivanti integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions that are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration** and **Add a new integration from Citrix-provided templates**.
2. Choose the Ivanti tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**:
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

Ivanti

Connector parameters

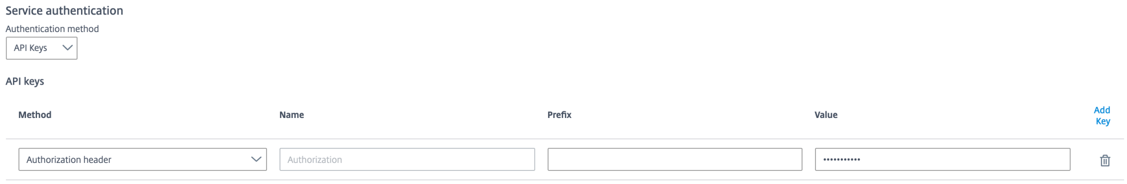
Base URL

https://cytrix.trysaasit.com/api/od

Icon

On-premises instance

5. Under **Service Authentication**, select **API Keys** from the **Authentication method** menu and enter the token value that you collected in the **Value** field. API Keys ensure that your integration meets the maximum security compliance.



Service authentication

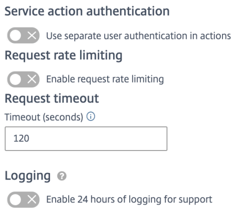
Authentication method

API Keys

API keys

Method	Name	Prefix	Value	Add Key
Authorization header	Authorization		•••••••••••	

6. Leave the **Enable request rate limiting** toggle as disabled.
7. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
8. Select **Save** to proceed.



Service action authentication

Use separate user authentication in actions

Request rate limiting

Enable request rate limiting

Request timeout

Timeout (seconds)

120

Logging

Enable 24 hours of logging for support

The **Microapp Integrations** page opens with your added integration and its microapps. You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Ivanti connector specifications.

Use Ivanti microapps

Our Ivanti integration template comes with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Agent Incidents: Self-service to manage incidents by an Agent. Subscribers must have agent access to Ivanti or resolutions will fail.

Notification or Page	Use-case workflows
Incident Status Change notification	When the status of an incident changes, the owner receives a notification with a link to incident details.
New Incident notification	When a new incident is created, the owner receives a notification with a link to incident details.
Resolution Target Breached notification	When an incident breaches the resolution target, the owner receives a notification with a link to incident details.
Response Target Breached notification	When an incident breaches the response target, the owner receives a notification with a link to incident details.
Incident Detail page	View details of an incident with an option to add a note and **Resolve Incident**.
My Incidents page	Provides a list of incidents assigned to a user with a link to incident details.

Create from Service Catalog: Create a service request based on the title of the microapp by using a deep link back to the self-service module of Ivanti.

Notification or Page	Use-case workflows
Create Service Request page	Provides a form to create a new service request with a searchable service list from the service catalog.

Create Incident: Report a new incident in Ivanti. This is a quick create that can be submitted without an owner.

Notification or Page	Use-case workflows
Create Incident page	Provides a form to create an incident.

My Incidents: Self-service for my incidents.

Notification or Page	Use-case workflows
My Incident has Breached its Resolution Target notification	When a user's incident breaches the resolution target, they receive a notification with a link to incident details.
My Incident has Breached its Response Target notification	When a user's incident breaches the response target, they receive a notification with a link to incident details.
My Incident Status has changed notification	When the status of a user's incident changes, they receive a notification with a link to incident details.
My New Incidents notification	When a new incident is logged for a user, they receive a notification with a link to incident details.
Incident Detail page	View details of an incident with an option to **Resolve Incident**.
My Active Incidents page	Provides a list of incidents assigned to a user with a link to incident details.

My Service Requests: View existing service requests containing deep links to view the dynamic templates associated to the request.

Notification or Page	Use-case workflows
New service request notification	When a new service request is submitted, the user receives a notification with a link to service request details.
Service Request Status Change notification	When the status of a service request changes, the owner receives a notification with a link to service request details.
My Service Requests page	Provides a list of service requests assigned to a user with a link to incident details.

Notification or Page	Use-case workflows
Service Request Detail page	View details of a service request with an option to add a note and **Open Service Request**.

Quick Service Request: Quickly create common service requests.

Notification or Page	Use-case workflows
Quick Service Request List page	Provides a form for submitting common service requests, such as **Request Equipment**, **Request Computer**, **Reset Password**, and **Request Access**.

Integrate Jira

April 28, 2021

Integrate with Jira to track issues and get automated updates about tasks on any device, intranet, or messenger. Use the following process to enable the integration with Jira. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

> **Note:**
>
> We provide two Jira integration templates for your use. We recommend using the newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure. For full details of the microapps available in each integration, see Use Jira microapps.

Review prerequisites

All user accounts that you want to use through this integration must have visibility for their email in account settings set to **Anyone**. This means service accounts as well as the accounts that users log in to Citrix Workspace through OAuth. Navigate to https://id.atlassian.com/manage-profile/profile-and-visibility, log in if necessary, in **Contact** section and next to the email account select **Anyone** under the **Who can see this?** menu.

> **Note:**
>
> Jira no longer supports Internet Explorer 11. Configuring this microapp with Internet Explore 11 will result in errors. Switch to another browser (Chrome, Edge, and so on) to configure correctly.

After you prepare this integration in Jira, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `templatebaseurl` Replace `{ cloud-id }` with your cloud-ID. If you need to find your cloud-ID, log in as an administrator of your JIRA instance and look at the URL.
- **Authorization URL**: This is predefined. This is the authorization server URL provided when setting up the target application integration.
- **Token URL**: This is predefined. This is the URL of the access authorization token.
- **Epic-Name-customFieldId**: This is found in the Jira admin portal. See Replace Service Action variables.
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server.
- **Client Secret**: The client secret is a unique string issued when setting up the target application integration.
- **Username**: This is your service account username.
- **Password**: This is your service account password.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

The following prerequisites must be met before you begin the integration process:

- A dedicated user account that has Browse Projects access to the Jira projects that you would like to manage.

  For more information about managing users, see https://www.atlassian.com/software/jira/guides#ManagingUsers-Addingusers.

  For more information about how to add a user to a permission scheme for a Jira project, see Managing project permissions.

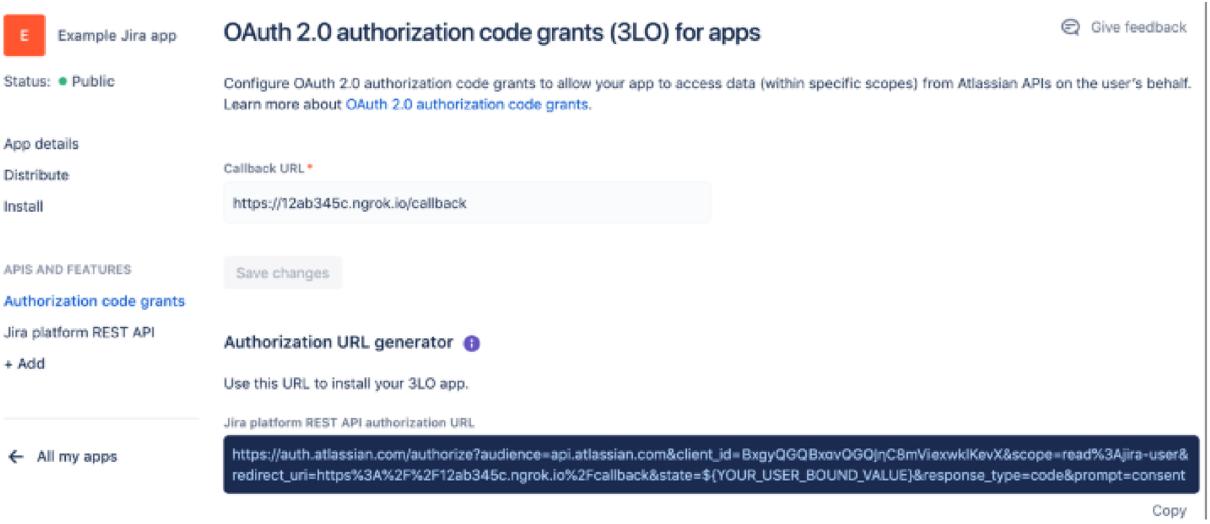
- Configure Citrix Gateway to support single sign-on for Jira so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Jira Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Enabling OAuth 2.0 (3LO)

Before you implement OAuth 2.0 (3LO), you need to enable it for your app in Jira app management.

1. Navigate to App management.
2. Create a new app by selecting **Create new app**, enter a name, agree to the terms, and select **Create**.
3. Copy the **Client ID** and **Secret** for later use
4. In the **APIS AND FEATURES** section in the side navigation, click **+Add**
5. In the Features section of the **APIs and features page** find **OAuth 2.0 (3LO)**, and select **Add** and then **Configure**.
6. Enter the **Callback URL**. Set this URL to any URL that is accessible by the app. When you implement OAuth 2.0 (3LO) in your app (see next section), the redirect_uri must match this URL.
7. Click Save changes.

Your screen looks something like this:



OAuth 2.0 authorization code grants (3LO) for apps

Configure OAuth 2.0 authorization code grants to allow your app to access data (within specific scopes) from Atlassian APIs on the user's behalf. Learn more about OAuth 2.0 authorization code grants.

Callback URL *

https://12ab345c.ngrok.io/callback

Authorization URL generator

Use this URL to install your 3LO app.

Jira platform REST API authorization URL

```
https://auth.atlassian.com/authorize?audience=api.atlassian.com&client_id=BxgyQGQBxovOGQjnC8mViexwklKevX&scope=read%3Ajira-user&
redirect_uri=https%3A%2F%2F12ab345c.ngrok.io%2Fcallback&state=${YOUR_USER_BOUND_VALUE}&response_type=code&prompt=consent
```

Create API Token

A script or other process can use an API token to perform basic authentication with Jira Cloud applications or Confluence Cloud. You must use an API token if the Atlassian account you authenticate with has had two-step verification enabled.

1. While logged in to the Atlassian account, go to API Tokens.
2. Select **Create API token** and enter a name for the token in the **Label** field.
3. **Copy to clipboard** and save for later use.

You must enter the **API token** as your **Password** when you add the integration to Citrix Workspace Microapps.

Add the Jira platform REST API

If you haven't already added the Jira platform REST API, do this now.

1. In the **APIS AND FEATURES** section in the side navigation, click **+Add**.
2. In the **APIs** section of the **APIs and features page** find **OAuth 2.0 (3LO)**, and select **Add** and then **Configure**.
3. Add the desired scopes for your app.

Add callback URL

Add a custom URL to your instance configuration to grant access to private data and enable OAuth authenticated user actions. To find your microapp server URL, sign in to Citrix Cloud, and select the **Microapps** tile. In the URL bar, copy the first section of the URL. This is your microapps server URL. The section `{ yourmicroappserverurl }` is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region(us/eu/ap-s)} .iws.cloud.com`.

For the HTTP Jira integration, you must add two different callback URLs. However, a Jira application can have only one callback URL. This means you need to register two applications; one for user actions and the other for synchronisation. They must have different callback URLs.

`https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`

`https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`

> **Note:**
>
> Pay attention to the callback URLs when adding the Jira integration. Service Authentication and Service Action Authentication have different Callback URLs. The Client ID and Secret must be the ones appropriate to either the Service Authentication or Service Action Authentication callback URL.

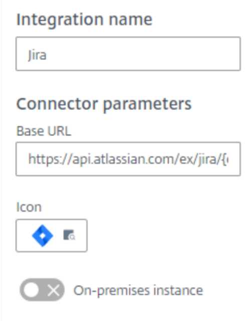
Add the integration to Citrix Workspace Microapps

Follow these steps to set up the Jira HTTP integration. We recommend using the newer HTTP integration for most use-cases. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

> **Note:**
>
> By default, this integration synchronizes data for a six (6) month time period. We recommend that you modify this value based on your needs and usual age of your tickets. The filter is based on last updated, not created. To change this you must modify the `timeToSync` variable in a data loading endpoint. See Replace Data Loading variable.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Jira tile under **Integrations**.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL** or simply replace `{ cloud-id }` in the example with your cloud ID. If you need to find your cloud-ID, log in as an administrator of your JIRA instance and look at the URL. This cloud-ID is a universally unique identifier (UUID), that is an 8-4-4-4-12 digit hexadecimal number which is part of the URL. Alternatively, you can authenticate using admin credentials and send a GET request to `https://api.atlassian.com/oauth/token/accessible-resources`. The Cloud-ID is part of the response.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default Jira icon.
    - Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance. Due to differences between Jira Cloud API v2 and Jira Server API v2 of your Jira instance, you must also update some parts of the integration manually. Contact support.



Integration name

Jira

Connector parameters

Base URL

https://api.atlassian.com/ex/jira/{

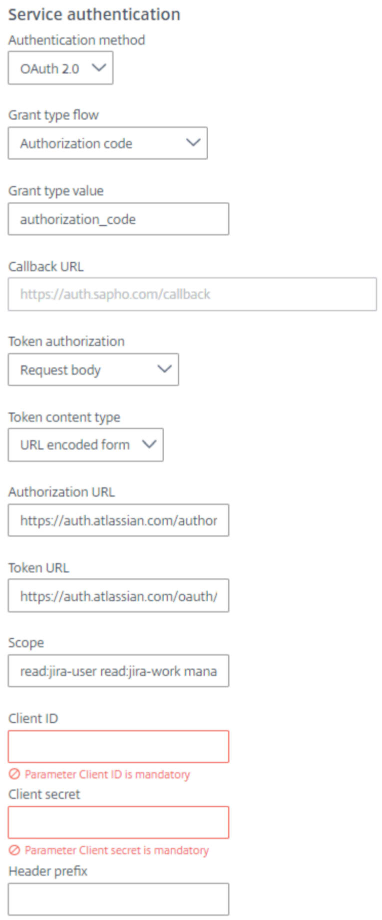
Icon

On-premises instance

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

    a) Select **Authorization code** from the **Grant type flow** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application. Service Authentication and Service Action Authentication have different Callback URLs.

b) Enter **authorization_code** in the **Grant type value** field.
c) Select **Request body** from the **Token authorization** menu.
d) Select **URL encoded form** from the **Token content type** menu.
e) Your **Authorization URL** is predefined. This is the authorization server URL provided when setting up the target application integration.
f) Your **Token URL** is predefined. This is the URL of the access authorization token.
g) Ensure the following is entered for **Scope**. This string is defined by the authorization server when setting up your target integration application. To synchronize other entities, you must add scopes here. Use the following, separated by a space: `read:jira-user read:jira-work manage:jira-project manage:jira-configuration write:jira-work manage:jira-data-provider offline_access`.
h) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Jira account. Client ID and Secret must be the ones appropriate to the Service Authentication callback URL.
i) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.
j) Enter your **Header prefix**. (optional) Enter the header prefix if your bearer prefix is different from the default header.



Service authentication

Authentication method
OAuth 2.0

Grant type flow
Authorization code

Grant type value
authorization_code

Callback URL
https://auth.sapho.com/callback

Token authorization
Request body

Token content type
URL encoded form

Authorization URL
https://auth.atlassian.com/author

Token URL
https://auth.atlassian.com/oauth/

Scope
read:jira-user read:jira-work mana

Client ID

⊘ Parameter Client ID is mandatory

Client secret

⊘ Parameter Client secret is mandatory

Header prefix

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

    a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
    b) Select **Authorization code** from the **Grant type flow** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application. Service Authentication and Service Action Authentication have different Callback URLs.
    c) Enter **authorization_code** in the **Grant type value** field.
    d) Select **Request body** from the **Token authorization** menu.
    e) Select **URL encoded form** from the **Token content type** menu.
    f) Your **Authorization URL** is predefined. This is the authorization server URL provided when setting up the target application integration.

g) Your **Token URL** is predefined. This is the URL of the access authorization token.

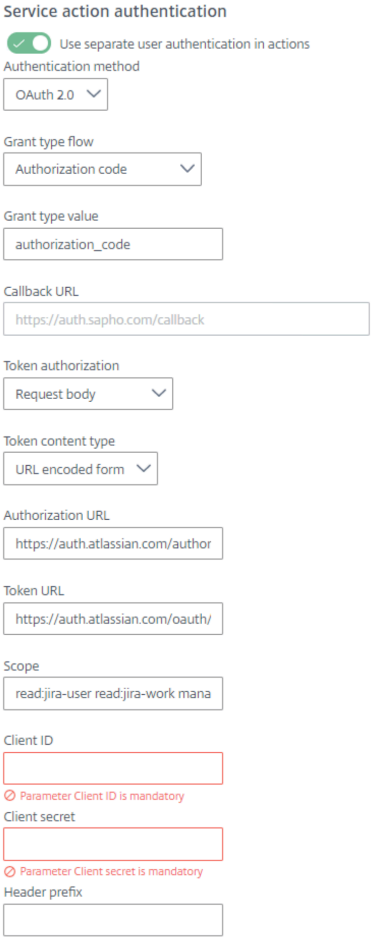
h) Ensure the following is entered for **Scope**. This string is defined by the authorization server when setting up your target integration application. To synchronize other entities, you must add scopes here. Use the following, separated by a space: `read:jira-user read:jira-work manage:jira-project manage:jira-configuration write:jira-work manage:jira-data-provider offline_access`.

i) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Jira account. Client ID and Secret must be the ones appropriate to the Service Action Authentication callback URL.

j) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

k) Enter your **Header prefix**. (optional) Enter the header prefix if your bearer prefix is different from the default header.

l) If you selected **OAuth 2.0** authentication method, you can select **+ Add Parameter** to include **Access token parameters**. Access token parameters define the access token parameters as required by the target application authorization server if necessary.



Service action authentication

☑ Use separate user authentication in actions

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://auth.sapho.com/callback

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://auth.atlassian.com/author

Token URL: https://auth.atlassian.com/oauth/

Scope: read:jira-user read:jira-work mana

Client ID:

Parameter Client ID is mandatory

Client secret:

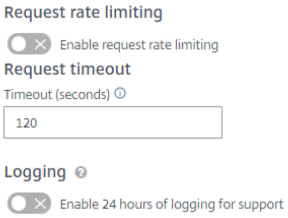
Parameter Client secret is mandatory

Header prefix:

7. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.

8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

9. The **Request timeout** field is set to 120 by default.



Request rate limiting

Enable request rate limiting

Request timeout

Timeout (seconds)

120

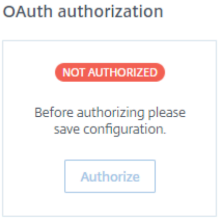
Logging

Enable 24 hours of logging for support

10. Select **Save** to proceed.

11. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Webex login screen.

    a) Enter your Service Account user name and password and select **Log in**.
    b) Select **Accept**.



OAuth authorization

NOT AUTHORIZED

Before authorizing please save configuration.

Continue with the following procedures to finish the set-up process.

Replace Service Action variables

To enable Create Epic page functionality, you must manually modify the **Create Epic** and **Create Epic wo Assignee** service actions. Replace the `{ Epic-Name-customFieldId }` value of the `EpicFieldId` template variable with the id of a custom field that the Epic Name is stored in.

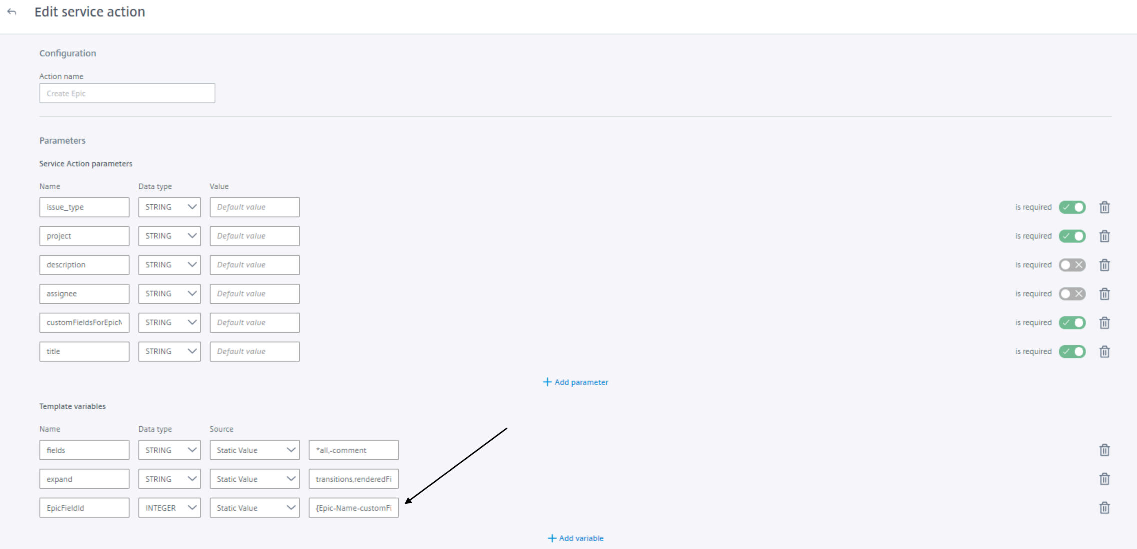
1. In the Jira admin portal, navigate to **Issues**. Select **Custom fields** from the left menu. Find the entry **Epic Name** and select the menu on the other side of the screen. Select **View field information**.

2. Copy and save the numeric value at the end of the URL.

3. Back in Microapps, select the menu next to the Jira integration, and then **Edit**. Select **Service Actions** from the left side navigation column.

4. Select the menu next to one of the service actions and select **Edit**, or select the name of the service action. Let’s start with the **Create Epic** service action.

5. In the **Edit Service Action** screen, under **Template variables** replace the `{ Epic-Name-customFieldId }` value of the `EpicFieldId` template variable with the custom field id that you collected earlier in Jira.



Edit service action

Configuration

Action name: Create Epic

Parameters

Service Action parameters

Name	Data type	Value	
issue_type	STRING	Default value	is required ✓
project	STRING	Default value	is required ✓
description	STRING	Default value	is required ✗
assignee	STRING	Default value	is required ✗
customFieldsForEpicN	STRING	Default value	is required ✓
title	STRING	Default value	is required ✓

Template variables

Name	Data type	Source	
fields	STRING	Static Value	*all,-comment
expand	STRING	Static Value	transitions,renderedFi
EpicFieldId	INTEGER	Static Value	{Epic-Name-customFi

6. Select **Save** to finish.

7. Now repeat this for the other service action: **Create Epic wo Assignee**.

Replace Data Loading variable

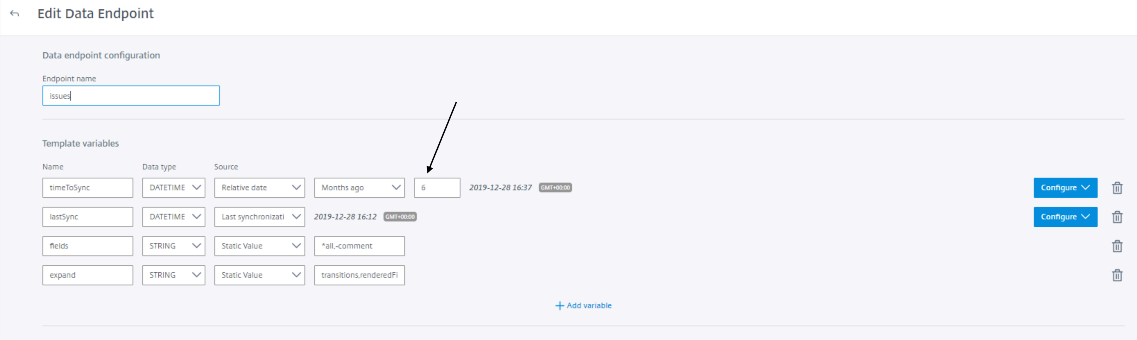
This integration synchronizes data for a six (6) month time period by default. We recommend that you modify this value based on your needs and usual age of your tickets. The filter is based on last updated, not created. To change this you must modify the `timeToSync` variable in the **Issues** data loading endpoint.

1. From the **Microapp Integrations** page, select the menu next to the Jira integration, and then **Edit**. The **Data Loading** screen opens. If you are in the configuration screen, select **Data Loading** from the left side navigation column.

2. Select the menu next to the **Issues** endpoint and then select **Edit**, or select the name of the endpoint: **Issues**.

3. In the **Edit Data Endpoint** screen, under **Template variables** replace the value for the `timeToSync` variable with the value that you want.

4. Select **Apply** at the bottom of the screen and confirm.



Edit Data Endpoint

Data endpoint configuration

Endpoint name: issues

Template variables

Name	Data type	Source			
timeToSync	DATETIME	Relative date	Months ago	6	2019-12-28 16:37 GMT+00:00
lastSync	DATETIME	Last synchronizati	2019-12-28 16:12 GMT+00:00		
fields	STRING	Static Value	*all,-comment		
expand	STRING	Static Value	transitions,renderedFi		

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Jira HTTP connector specifications.

Use Jira microapps

Our Jira HTTP integration comes with the following preconfigured out-of-the-box microapps:

Create Epic: Create a new Jira epic with details.

> **Note:**
>
> To enable Create Epic page functionality, you must modify the **Create Epic** and **Create Epic wo Assignee** service actions. See Replace Service Action variables.

Notification or Page	Use-case workflows
Create Epic page	Provides a form for creating a new Jira epic, including entering an epic name and selecting an issue type, project, and optionally an assignee, and also a field for adding a description.
Projects page	Provides a searchable form for selecting a project to assign the new epic to.

Create Ticket: Create a new Jira ticket with details.

Notification or Page	Use-case workflows
Create Ticket page	Provides a form for creating a new Jira ticket, including selecting an issue type, project, and optionally an assignee, and also a field for adding a description.
Projects page	Provides a searchable form for selecting a project to assign the new ticket to.

Tickets: View tickets, add comments, create subtasks, and change status and assignee.

Notification or Page	Use-case workflows
Comment Edited (Assigned Ticket) notification	When a comment is edited on an existing ticket that assigned to a user, they receive a notification of the edited comment in Workspace.
Comment Edited (Reported Ticket) notification	When a comment is edited on an existing ticket that a user reported, they receive a notification of the edited comment in Workspace.
New Comment (Assigned Ticket) notification	When a new comment is added to an existing ticket that assigned to a user, they receive a notification.
New Comment (Reported Ticket) notification	When a new comment is added to an existing ticket that a user reported, they receive a notification.
Ticket Assigned to You (Change) notification	When an existing ticket is assigned to a user, they receive a notification.
Ticket Assigned to You (New) notification	When a new ticket is assigned to a user, they receive a notification.
Ticket Assignee Change (Reported) notification	When a ticket is reassigned, the reporter of the ticket receives a notification.
Ticket Status Change (Assigned Ticket) notification	When the status of a ticket is changed, the assignee of the ticket receives a notification.
Ticket Status Change (Reported) notification	When the status of a ticket is changed, the reporter of the ticket receives a notification.

Notification or Page	Use-case workflows
Comment Detail page	Provides a read only view of a comment with details.
Create Sub-Task page	Provides a form for creating a subtask for a Jira ticket.
Ticket Detail page	Provides a detailed view of a Jira ticket with fields to add comments, and modify status, priority, and assignee directly from the page.
Tickets page	Allows users to search through Jira tickets with a search field, project selector, and status selector to quickly switch between All, My, Assigned, Reported, Watching, and Commented tickets.

Add the Legacy Jira integration

Follow these instructions in addition to the steps above to set up the legacy integration.

Prerequisites

For the Legacy integration, you need these values.

- URL
- Username
- Password - You must enter the **API token** as your **Password** when you add the integration to Citrix Workspace Microapps
- Authentication Method (either credentials or OAuth2)

For OAuth 2.0 you also need:

- Client ID
- Client Secret

Set duration to retrieve issues

For the Legacy Jira integration, set the duration that you want to retrieve issues in Jira. When you add the integration to Citrix Workspace Microapps, the amount of data the integration retrieves from Jira can be limited by applying a load filter.

1. Log in to Jira with your dedicated user account and password.

> **Note:**
>
> If you are using an existing account, the username is not the email of the account. To find the Username of an account, log in to your Jira instance, select the profile thumbnail, select **Profile**.

2. Enter the number of days of issues to retrieve.

   Default: 90 days.

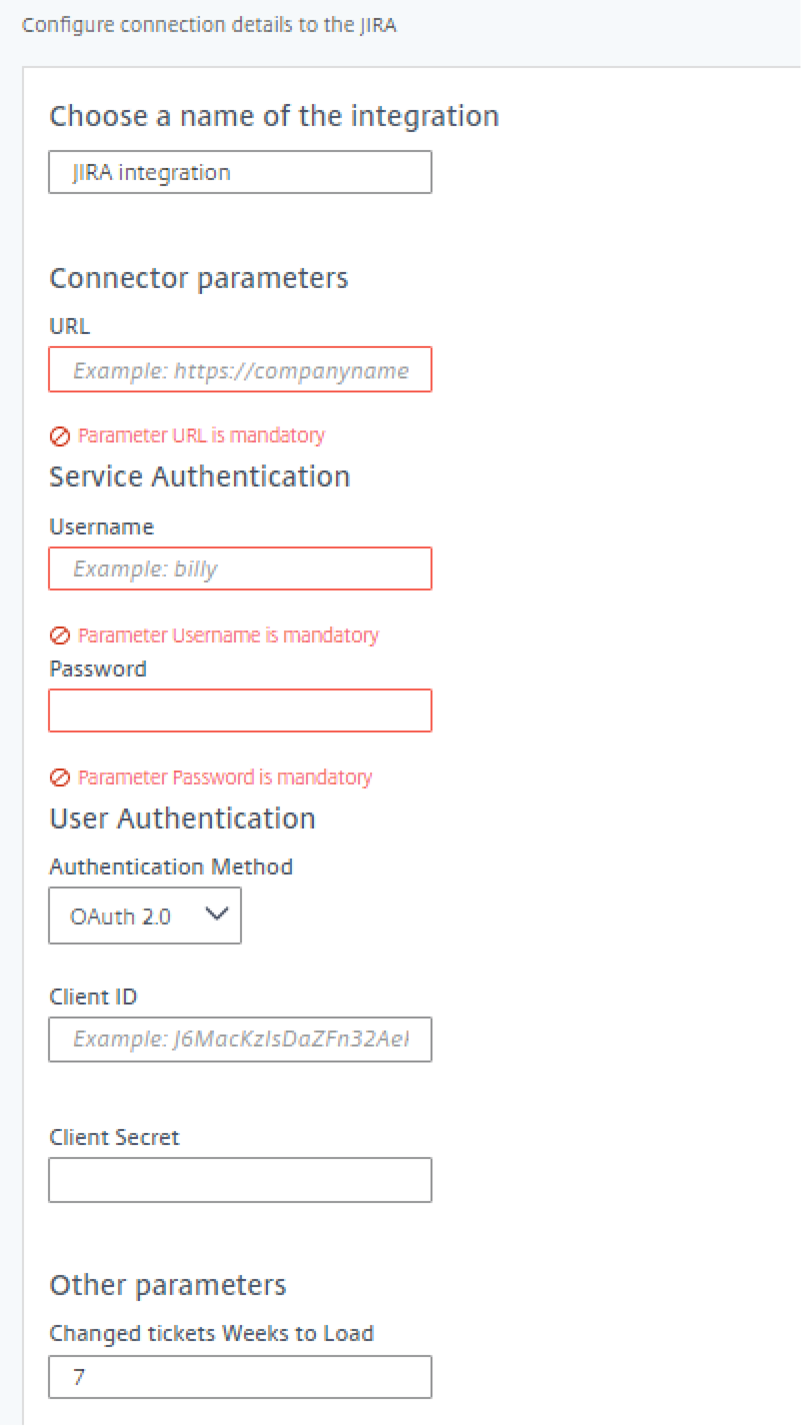
Add the Jira Legacy integrations

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Jira tile to add.
4. Enter a name for the integration.



Configure connection details to the JIRA

Choose a name of the integration

JIRA integration

Connector parameters

URL

Example: https://companyname

⊘ Parameter URL is mandatory

Service Authentication

Username

Example: billy

⊘ Parameter Username is mandatory

Password

⊘ Parameter Password is mandatory

User Authentication

Authentication Method

OAuth 2.0

Client ID

Example: J6MacKzIsDaZFn32AeI

Client Secret

Other parameters

Changed tickets Weeks to Load

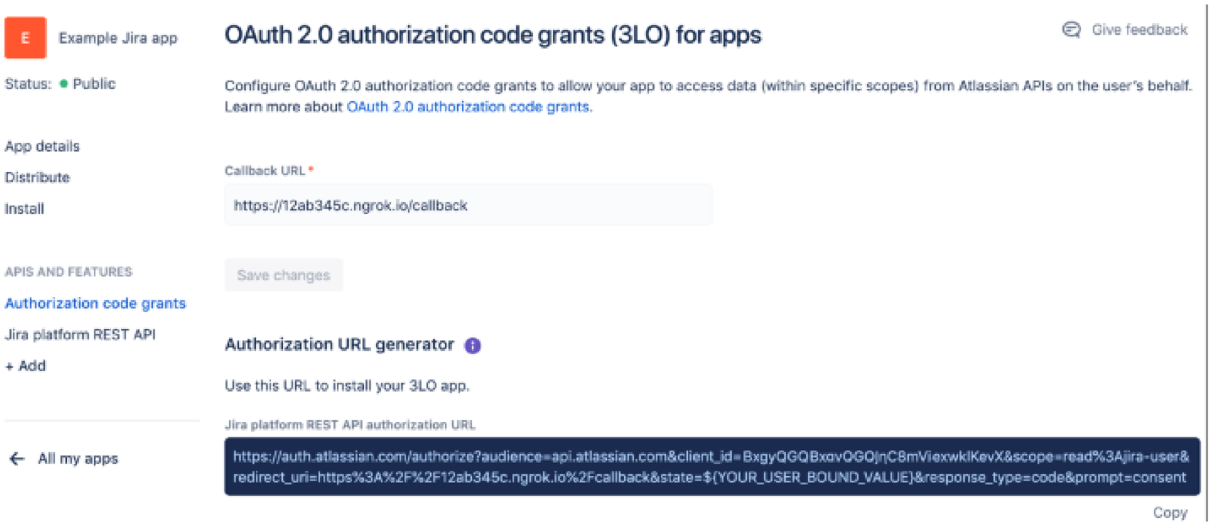
7

5. Enter the **Connector parameters** that you collected as prerequisites.

    - Enter your **URL**.
    - Enter your Service Authentication **Username** and **Password**.

> **Note:**
>
> Enter the **API token** you collected in a previous step as your **Password** when you add the integration to Citrix Workspace Microapps.

6. Select an **Authentication Method**. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access.
   - **Credentials** - Credentials The client's credentials are used.
   - **Oauth 2.0** - Enter the **OAuth Client ID** and **OAuth Client Secret** that you collected in the prerequisites procedure.
7. Select the number of **Changed Tickets Weeks To Load**.
8. Select **Add**.



OAuth 2.0 authorization code grants (3LO) for apps

Configure OAuth 2.0 authorization code grants to allow your app to access data (within specific scopes) from Atlassian APIs on the user's behalf. Learn more about OAuth 2.0 authorization code grants.

Callback URL *

https://12ab345c.ngrok.io/callback

Authorization URL generator

Use this URL to install your 3LO app.

Jira platform REST API authorization URL

```
https://auth.atlassian.com/authorize?audience=api.atlassian.com&client_id=BxgyQGQBxovQGOjnC8mViexwklKevX&scope=read%3Ajira-user&
redirect_uri=https%3A%2F%2F12ab345c.ngrok.io%2Fcallback&state=${YOUR_USER_BOUND_VALUE}&response_type=code&prompt=consent
```

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Jira connector specifications.

Legacy Jira microapps

Our Jira integration comes with the following preconfigured out-of-the-box microapps:

Create Ticket: Create a new Jira ticket with details.

Notification or Page	Use-case workflows
Create Ticket page	Provides a form for creating a new Jira ticket, including selecting an issue type, project, and optionally an assignee, and also a field for adding a description.

Tickets: View tickets, add comments, create subtasks, and change status and assignee.

Notification or Page	Use-case workflows
Comment Edited (Assigned Ticket) notification	When a comment is edited on an existing ticket that assigned to a user, they receive a notification of the edited comment in Workspace.
Comment Edited (Reported) notification	When a comment is edited on an existing ticket that a user reported, they receive a notification of the edited comment in Workspace.
New Comment (Assigned Ticket) notification	When a new comment is added to an existing ticket that assigned to a user, they receive a notification.
New Comment (Reported Ticket) notification	When a new comment is added to an existing ticket that a user reported, they receive a notification.
Ticket Assigned to You (Change) notification	When an existing ticket is assigned to a user, they receive a notification.
Ticket Assigned to You (New) notification	When a new ticket is assigned to a user, they receive a notification.
Ticket Assignee Change (Reported) notification	When a ticket is reassigned, the reporter of the ticket receives a notification.
Ticket Status Change (Assigned Ticket) notification	When the status of a ticket is changed, the assignee of the ticket receives a notification.
Ticket Status Change (Reported) notification	When the status of a ticket is changed, the reporter of the ticket receives a notification.
Add Comment page	Provides a form for adding a comment to a Jira ticket.

Notification or Page	Use-case workflows
Change Assignee page	Provides a form for changing the assignee of a Jira ticket.
Change Status page	Provides a form for changing the status of a Jira ticket.
Comment Detail page	Provides a read only view of a comment with details.
Create Sub-Task page	Provides a form for creating a subtask for a Jira ticket.
Ticket Detail page	Provides a read only view of a Jira ticket with details.
Tickets page	Allows users to search for Jira tickets that are assigned to them, reported by them, or that they have commented on.

Integrate Kronos Workforce Central

May 28, 2021

Deploy the Kronos Workforce Central integration template to access Kronos anywhere through Citrix Workspace. With this template integration:

- Managers can easily view and respond to potential workforce management activities.
- Employees can perform time management tasks and submit requests.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details about these microapps, see Use Kronos microapps.

Review prerequisites

After you set up this integration with Kronos Workforce Central, you need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `http://kronos-server.workspaceintelligent.com`
- **Kronos Username**: This is the name of the Kronos Superuser.

- **Kronos Password**: This is the Kronos Superuser password.
- **Kronos Date Format**: Enter the date format available in Kronos setup. See Verify date format in Kronos.

User account

The integration requires regular access to your Kronos Workforce Central instance. We recommend creating a dedicated Superuser account. This Superuser account must have full administrator privileges and permissions.

This Superuser account is created by an administrator of Kronos. For more information about the Kronos Partner Account, see https://www.kronos.com/kronos-partner-network/become-partner and https://www.kronos.com/2018/blogs/working-smarter-cafe/2020-ukg-kronos-community-superusers-announced.

API access

The Kronos Workforce Central APIs are enabled by default through web services for a partner account. This may require a separate agreement with the vendor in order to get SUPER USER credentials for setting up the integration.

Authentication is not enabled for this template. Kronos does not support O-AUTH 2.0 (that is, write-back for users is not supported by Kronos). Each service action (such as approve and refuse) performed using our Citrix Kronos Workforce Central integrations will be registered under the "SUPER USER" credential user authentication. As a workaround, our custom function in this integration can capture the name of the approver in the comments session.

Add the integration to Citrix Workspace Microapps

Add the Kronos Workforce Central integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions, which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Kronos Workforce Central tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**:

- Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.

5. Enter your super user credential for **Kronos Username**. We recommended this to be the provide SUPERUSER credentials.
6. Enter your **Kronos Password**.
7. Enter your **Kronos Date Format**. Ensure that the format that you select is used in Kronos as well. See Verify date format in Kronos.

   Service authentication and **Service action authentication** are not enabled.
8. Select the **Enable request rate limiting** toggle. Enter *320* for **Number of requests** and *1 second* for **Time interval**.
9. **Request timeout** is set to *120* by default.
10. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
11. Select **Save** to proceed.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

Verify date format in Kronos

Follow these steps to configure date format in Kronos. Our Kronos Workforce Central integration template supports these date formats: 'YYYY/MM/DD', 'MM/DD/YYYY', 'YYYY-MM-DD', 'MM-DD-YYYY', 'M/DD/YYYY', 'DD-MM-YYYY'.

Ensure the date format here is entered in the **Kronos Date Format** field as well when you add the integration.

1. Log in with your super user account to Kronos UI.
2. Follow the navigation path:**Setup > System Configuration > System Settings > Locale**
3. Get the display format for long date from the **site.local.LONG_DATE** field.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

> **Note:**
>
> We recommend running full sync once a day.
>
> The Kronos SoR does not support incremental synchronization. Using scripting, we have implemented custom sync function to cover notification use cases for better user experience.

For more details of API endpoints and table entities, see Kronos connector specifications.

> **Note:**
>
> The Kronos SoR does not support pagination. Thus we have limited the cache accordingly. This template integration stores only two months of data into the cache for most use cases.
>
> The Record Timestamp microapp can hold up to 2 days of data into cache.

Use Kronos microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

My Accrual Balance: View accrual balance for different days instantly.

Notification or Page	Use-case workflows
Accrual Balance Index page	Provides a view of a user's Accrual Balance for current date.
Accrual Balance Page	Provides a view of a user's accrual balance for any specific date using **View Accrual** button.

My Time Off History: Allows users to view their history of time off data available for current month.

Notification or Page	Use-case workflows
Time Off History page	Provides a personalized list of a user's time off requests and details for the current month.

Record Timestamp: Register their punch in and out date and time.

Notification or Page	Use-case workflows
Record Timestamp page	Provides a form with date and time to **Record Punch In-Time** and **Record Punch Out-Time**.

Request Time Off: Submit application for time off.

Notification or Page	Use-case workflows
Request Time Off page	Provides form to request time off and **Submit** the request.

Paid Time Off Request Approval: Receive notifications for all time off requests to managers and push notifications for all approved or refused time off requests back to the original requester.

Notification or Page	Use-case workflows
Paid Time off Approved notification	When a request for time off is approved, the requester receives a notification.
Paid Time off Refused notification	When a request for time off is refused, the requester receives a notification.
Paid Time off Request notification	When there is a new request for time off, the approving supervisor receives a notification with **Approve** and **Refuse** options.
Paid Time off Request page	Provides approving supervisor with request for time off details and **Approve** and **Refuse** options.

Time Log: Receive notifications for all Work Time requested. Approve or refuse for the single user or group of users.

Notification or Page	Use-case workflows
New Approval Request notification	When a Work Time request is submitted for approval, the approving supervisor receives a notification with **Approve** and **Refuse** options.
Time Log Approved notification	When a Work Time request is approved, the requester receives a notification.
Time Log Refused notification	When a Work Time request is refused, the requester receives a notification.
Time Log Request page	Provides a form for supervisors to view requested Work Time details along with **Approve** and **Refuse** options.

Integrate Microsoft Dynamics CRM

November 9, 2021

Integrate with Microsoft Dynamics CRM to monitor and manage leads, opportunities, and cases without requiring extra logins. Use the following process to enable the Microsoft Dynamics CRM Integration. Review the prerequisites, register the new application, get the key value, and delegate permissions.

> **Note**
>
> We provide two Microsoft Dynamics integration templates for your use. We recommend using the newer HTTP integration for most use-cases as it provides more power to configure the cached data structure. The **Microsoft Dynamics** template is the basis for the Microsoft Dynamics HTTP integration. For full details of the microapps available in each integration, see Use Microsoft Dynamics microapps.

For a comprehensive list of out-of-the-box Microsoft Dynamics CRM microapps, see Use Microsoft Dynamics CRM microapps.

Review prerequisites

Create a dedicated Office 365 account to configure the integration. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

After you set up this integration in MS Dynamics CRM, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- Authorization Sign-on URL
- Application (client) ID
- Secret key value
- Valid Microsoft Dynamics CRM account details

Ensure you meet the following prerequisites:

- If your internal server hosting Workspace is behind a firewall, allow access to host name www.dynamics.com with port 443, so Workspace can connect to the MS Dynamics CRM cloud.
- Authorization Sign-on URL (Citrix provided). Configure Citrix Gateway to support single sign-on for MS Dynamics so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.
- A dedicated Office 365 account that you use to configure the MS Dynamics CRM integration. This dedicated account must have full data access privileges in MS Dynamics CRM (System Admin-

istrator). For more information, see Create users in Dynamics 365 for Customer Engagement apps and assign security roles.

Register your application and callback URLs

Register your new application, add callback URLs, and collect the Application ID.

1. Log in to https://portal.azure.com.
2. Select **Azure Active Directory > App registrations > + New registration**.
3. Enter the **Name** that you want to use for the application, select your **Supported account types** according to your organizational needs. Select **Help me choose…** for guidance on this option.
4. Add a custom URL to your instance configuration to grant access to private data and enable OAuth authenticated user actions. Under **Redirect URI (optional)**, select **Web** and enter the two following callback URLs and the base URL for your instance of MS Dynamics. This is the same URL that you have to enter in the Microapps UI when setting up the integration. The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com:
   - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
   - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
   - Your Base URL, the consistent part of your web address that you use for this integration. For example: `https://app.{ yoursaasapp } .com`
5. Copy the **Application (client) ID** for use when you add the integration to Citrix Workspace Microapps. Copy the **Application ID URI** as well, and store for later use if needed.

Enable delegated permissions

With delegated permissions the app can be delegated permission to act as the signed-in user when making calls to the target resource.

1. From your registered app view, select **API permissions** and **+Add a permission**.
2. Select the **Dynamics CRM** tile, and under **What type of permissions does your application require?**, select the **DELEGATED PERMISSIONS** tile.
3. Select the check box for the permissions to add, specifically `user_impersonation`, and select **Add permissions** at the bottom.

Generate a secret key value

Generate a secret string that the application uses to prove its identity when requesting a token.

1. Select **Certificates and Secrets**. Select **+ New client secret**.
2. Enter a **Description** for the client secret.
3. Under **Expires**, select **Never**. Select **Add**.
4. Copy and save the client secret **Value** that was created. It is not visible after you leave this page.

Filter queries

Most Microsoft Dynamics CRM entities support filtering. The `$filter` parameter can be added to any endpoints. For more information, consult the Microsoft Dynamics CRM query Web API reference.

Examples:

```
Only active appointments:
//api/data/v9.0/appointments$filter(statecode eq 0 or statecode eq 3)

Only incidents from last 1 month:
//api/data/v9.0/incidents$filter Microsoft.Dynamics.CRM.LastXMonths(
    PropertyName='modifiedon',PropertyValue=1)
```

Add the Microsoft Dynamics integration to Citrix Workspace Microapps

Add the Microsoft Dynamics HTTP integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps to set up the Microsoft Dynamics HTTP integration. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

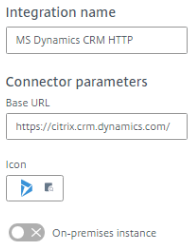
2. Choose the **Microsoft Dynamics** tile.

3. Enter an **Integration name** for the integration.

4. Enter **Connector parameters**.

    - Enter the instance **Base URL**. This is the domain for your MS Dynamics environment and the consistent part of your web address that you use for this integration.

- Select an **Icon** for the integration from the Icon Library, or leave this as the default MS Dynamics icon.
- Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.



Integration name

MS Dynamics CRM HTTP

Connector parameters

Base URL

https://citrix.crm.dynamics.com/

Icon

On-premises instance

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available and a Grant Type authorization code. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

    a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.

    b) Select **Request body** from the **Token authorization** menu.

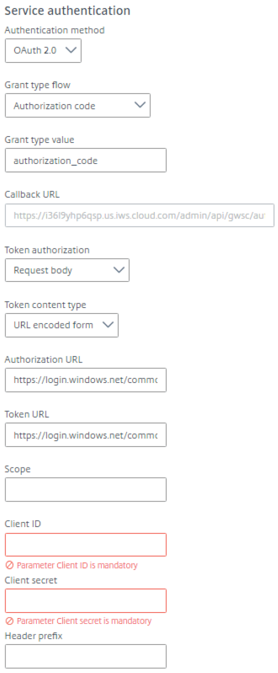
    c) The **Authorization URL** is prefilled. Enter your Authorization URL or simply keep the generally available provided URL.

    d) The **Token URL** is prefilled. Enter your Token URL or simply keep the generally available provided URL.

    e) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

    f) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

    g) (Optional) Enter your **Header prefix** if your bearer prefix is different from the default header.



Service authentication

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://login.windows.net/commc

Token URL: https://login.windows.net/commc

Scope:

Client ID:

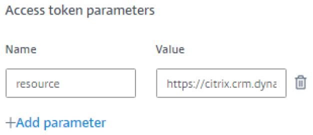
⊘ Parameter Client ID is mandatory

Client secret:

⊘ Parameter Client secret is mandatory

Header prefix:

6. When using OAuth 2.0, select **Add Parameter** to include **Access token parameters**. Enter *resource* for **Name** and `{ yourmsdynamicscrmurl }` for **Value**. This parameter is required by the target application authorization server.



Access token parameters

Name	Value
resource	https://citrix.crm.dyna

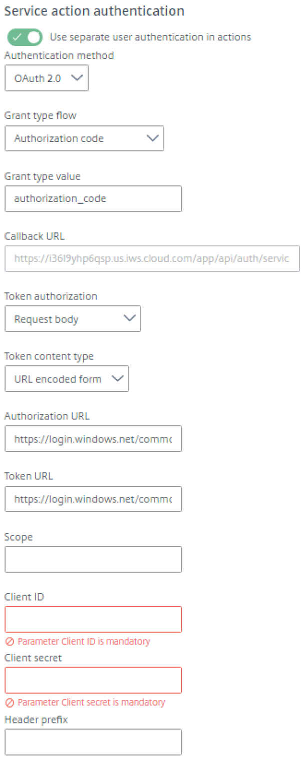
7. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.
   c) Select **Request body** from the **Token authorization** menu.
   d) The **Authorization URL** is prefilled.
   e) The **Token URL** is prefilled.
   f) Enter your **Client ID**. The client ID is the string representing client registration information

unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

h) (Optional) Enter your **Header prefix** if your bearer prefix is different from the default header.



Service action authentication

✓ Use separate user authentication in actions

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://login.windows.net/commc

Token URL: https://login.windows.net/commc

Scope:

Client ID:

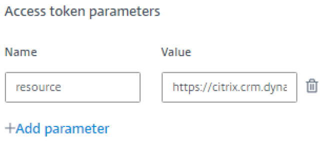
⊘ Parameter Client ID is mandatory

Client secret:

⊘ Parameter Client secret is mandatory

Header prefix:

8. Again, select **Add Parameter** to include **Access token parameters**. Enter *resource* for **Name** and `{ yourmsdynamicscrmurl }` for **Value**. This parameter is required by the target application authorization server.



Access token parameters

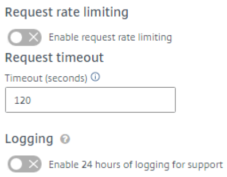
Name	Value
resource	https://citrix.crm.dyna

+Add parameter

9. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.

10. Leave **Request timeout** value as default.

11. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

Enable request rate limiting

Request timeout

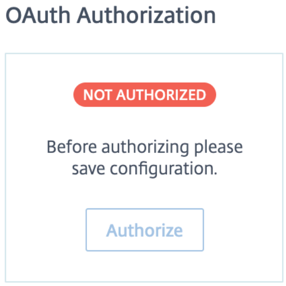
Timeout (seconds)

120

Logging

Enable 24 hours of logging for support

12. Select **Save** to proceed.
13. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Workday login screen.
    a) Enter your Service Account user name and password and select **Log in**.
    b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see MS Dynamics connector specifications.

Use MS Dynamics CRM microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our Microsoft Dynamics CRM integration comes with the following preconfigured out-of-the-box microapps:

Accounts: Search, view, and edit accounts.

Notification or Page	Use-case workflows
Account Assigned To You (Existing) notification	When the owner of an account is changed, the new owner receives a notification.

Notification or Page	Use-case workflows
Account Assigned To You (New) notification	When a new account is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Contact Detail page	Provides details of a contact.
Edit Account page	Provides a form for updating details of an account.
My Accounts page	Provides a table view of a user's accounts with search functionality and a link to details.

Appointments: Search, view, and edit appointments.

Notification or Page	Use-case workflows
Appointment Assigned To You (Existing) notification	When the owner of an appointment is changed, the new owner receives a notification.
Appointment Assigned To You (New) notification	When a new appointment is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Appointment Detail page	Provides details of an appointment.
Case Detail page	Provides details of a case.
Contact Detail page	Provides details of a contact.
Edit Appointment page	Provides a form for updating details of an appointment.
Lead Detail page	Provides details of a lead.
My Open Appointments page	Provides a table view of a user's appointments with search functionality and a link to details.
Opportunity Detail page	Provides details of an opportunity.

Cases: Search, view, and edit cases.

Notification or Page	Use-case workflows
Case Assigned To You (Existing) notification	When the owner of a case is changed, the new owner receives a notification.
Case Assigned To You (New) notification	When a new case is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Case Detail page	Provides details of a case.
Case Resolve page	Provides a form for resolving a case.
Contact Detail page	Provides details of a contact.
Edit Case page	Provides a form for updating details of a case.
My Open Cases page	Provides a table view of a user's open cases with search functionality and a link to details.

Contacts: Search, view, and edit contacts.

Notification or Page	Use-case workflows
Contact Assigned To You (Existing) notification	When the owner of a contact is changed, the new owner receives a notification.
Contact Assigned To You (New) notification	When a new contact is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Contact Detail page	Provides details of a contact.
Edit Contact page	Provides a form for updating details of a contact.
My Contacts page	Provides a table view of a user's contacts with search functionality and a link to details.

Create Account: Create an account.

Notification or Page	Use-case workflows
Create Account page	Provides a page for submitting a new account with details.

Create Appointment: Create an appointment.

Notification or Page	Use-case workflows
Create Appointment page	Provides a page for submitting a new appointment with details.

Create Case: Create a case.

Notification or Page	Use-case workflows
Create Case page	Provides a page for submitting a new case with details.

Create Contact: Create a contact.

Notification or Page	Use-case workflows
Create Contact page	Provides a page for submitting a new contact with details.

Create Lead: Create a lead.

Notification or Page	Use-case workflows
Create Lead page	Provides a page for submitting a new lead with details.

Create Opportunity: Create an opportunity.

Notification or Page	Use-case workflows
Create Opportunity page	Provides a page for submitting a new opportunity with details.

Create Phone Call: Create a phone call.

Notification or Page	Use-case workflows
Create Phone Call page	Provides a page for submitting a new phone call with details.

Create Task: Create a task.

Notification or Page	Use-case workflows
Create Task page	Provides a page for submitting a new task with details.

Leads: Search, view, and edit leads.

Notification or Page	Use-case workflows
Lead Assigned To You (Existing) notification	When the owner of a lead is changed, the new owner receives a notification.
Lead Assigned To You (New) notification	When a new lead is assigned to a user, they receive a notification.
Edit Lead page	Provides a form for updating details of a lead.
Lead Detail page	Provides details of a lead.
My Open Leads page	Provides a table view of a user’s open leads with search functionality and a link to details.

Opportunities: Search, view, and edit opportunities.

Notification or Page	Use-case workflows
Opportunity Assigned To You (Existing) notification	When the owner of an opportunity is changed, the new owner receives a notification.
Opportunity Assigned To You (New) notification	When a new opportunity is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Close as Lost page	Provides a form for closing an opportunity as lost with details.
Close as Won page	Provides a form for closing an opportunity as won with details.
Contact Detail page	Provides details of a contact.
Edit Opportunity page	Provides a form for updating details of an opportunity.
My Open Opportunities page	Provides a table view of a user’s open opportunities with search functionality and a link to details.
Opportunity Detail page	Provides details of an opportunity.

Phone Calls: Search, view, and edit phone calls.

Notification or Page	Use-case workflows
Phone Call Assigned To You (Existing) notification	When the owner of a phone call is changed, the new owner receives a notification.
Phone Call Assigned To You (New) notification	When a new phone call is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Case Detail page	Provides details of a case.
Contact Detail page	Provides details of a contact.
Edit Phone Call page	Provides a form for updating details of a phone call.
My Open Phone Calls page	Provides a table view of a user’s phone calls with search functionality and a link to details.

Notification or Page	Use-case workflows
Opportunity Detail page	Provides details of an opportunity.
Phone Call Detail page	Provides details of a phone call.

Tasks: Search, view, and edit tasks.

Notification or Page	Use-case workflows
Task Assigned To You (Existing) notification	When the owner of a task is changed, the new owner receives a notification.
Task Assigned To You (New) notification	When a new task is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Case Detail page	Provides details of a case.
Contact Detail page	Provides details of a contact.
Edit Task page	Provides a form for updating details of a task.
Lead Detail page	Provides details of a lead.
My Open Tasks page	Provides a table view of a user's open tasks with search functionality and a link to details.
Opportunity Detail page	Provides details of an opportunity.
Task Detail page	Provides details of a task.

Add the legacy integration

Follow these instructions to set up the legacy integration.

Add the legacy integration to Citrix Workspace Microapps

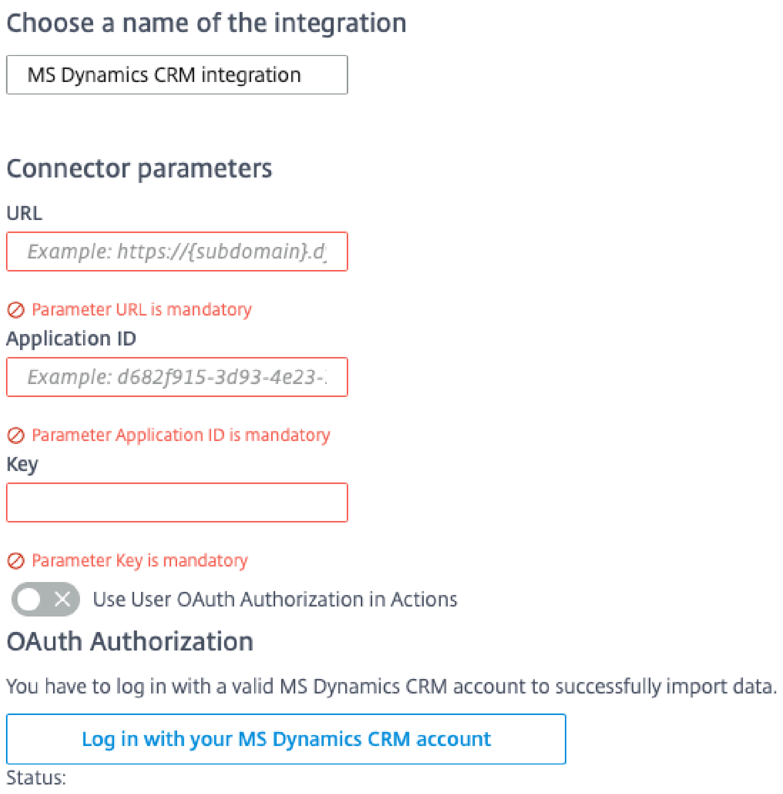
Add the Microsoft Dynamics CRM integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the overview page, select **Get Started**.

The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Microsoft Dynamics CRM tile to add.
4. Enter a name for the integration.



Choose a name of the integration

MS Dynamics CRM integration

Connector parameters

URL: Example: https://{subdomain}.d

Parameter URL is mandatory

Application ID: Example: d682f915-3d93-4e23-

Parameter Application ID is mandatory

Key:

Parameter Key is mandatory

Use User OAuth Authorization in Actions

OAuth Authorization

You have to log in with a valid MS Dynamics CRM account to successfully import data.

Status:

5. Enter the **Connector parameters** that you collected in the previous procedures.
   - Enter your **URL**.
   - Enter your **Application ID** and **Key**.
   - Toggle **Use User OAuth Authorization in Actions** if you require OAuth in microapp actions.
   - Select **Log in with your MS Dynamics CRM account** to enable OAuth Authorization. A sign-in page opens in a new tab. You are prompted to enter an account name, confirm access, and enter a password.
6. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Microsoft Dynamics connector specifications.

Use Microsoft Dynamics CRM legacy microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our MS Dynamics CRM integration comes with the following preconfigured out-of-the-box microapps:

Accounts: Search, view, and edit accounts.

Notification or Page	Use-case workflows
Account Assigned To You (Existing) notification	When the owner of an account is changed, the new owner receives a notification.
Account Assigned To You (New) notification	When a new account is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Contact Detail page	Provides details of a contact.
Edit Account page	Provides a form for updating details of an account.
My Accounts page	Provides a table view of a user’s accounts with search functionality and a link to details.

Appointments: Search, view, and edit appointments.

Notification or Page	Use-case workflows
Appointment Assigned To You (Existing) notification	When the owner of an appointment is changed, the new owner receives a notification.
Appointment Assigned To You (New) notification	When a new appointment is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.

Notification or Page	Use-case workflows
Appointment Detail page	Provides details of an appointment.
Case Detail page	Provides details of a case.
Contact Detail page	Provides details of a contact.
Edit Appointment page	Provides a form for updating details of an appointment.
Lead Detail page	Provides details of a lead.
My Open Appointments page	Provides a table view of a user's appointments with search functionality and a link to details.
Opportunity Detail page	Provides details of an opportunity.

Cases: Search, view, and edit cases.

Notification or Page	Use-case workflows
Case Assigned To You (Existing) notification	When the owner of a case is changed, the new owner receives a notification.
Case Assigned To You (New) notification	When a new case is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Case Detail page	Provides details of a case.
Case Resolve page	Provides a form for resolving a case.
Contact Detail page	Provides details of a contact.
Edit Case page	Provides a form for updating details of a case.
My Open Cases page	Provides a table view of a user's open cases with search functionality and a link to details.

Contacts: Search, view, and edit contacts.

Notification or Page	Use-case workflows
Contact Assigned To You (Existing) notification	When the owner of a contact is changed, the new owner receives a notification.

Notification or Page	Use-case workflows
Contact Assigned To You (New) notification	When a new contact is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Contact Detail page	Provides details of a contact.
Edit Contact page	Provides a form for updating details of a contact.
My Contacts page	Provides a table view of a user's contacts with search functionality and a link to details.

Create Account: Create an account.

Notification or Page	Use-case workflows
Create Account page	Provides a page for submitting a new account with details.

Create Appointment: Create an appointment.

Notification or Page	Use-case workflows
Create Appointment page	Provides a page for submitting a new appointment with details.

Create Case: Create a case.

Notification or Page	Use-case workflows
Create Case page	Provides a page for submitting a new case with details.

Create Contact: Create a contact.

Notification or Page	Use-case workflows
Create Contact page	Provides a page for submitting a new contact with details.

Create Lead: Create a lead.

Notification or Page	Use-case workflows
Create Lead page	Provides a page for submitting a new lead with details.

Create Opportunity: Create an opportunity.

Notification or Page	Use-case workflows
Create Opportunity page	Provides a page for submitting a new opportunity with details.

Create Phone Call: Create a phone call.

Notification or Page	Use-case workflows
Create Phone Call page	Provides a page for submitting a new phone call with details.

Create Task: Create a task.

Notification or Page	Use-case workflows
Create Task page	Provides a page for submitting a new task with details.

Leads: Search, view, and edit leads.

Notification or Page	Use-case workflows
Lead Assigned To You (Existing) notification	When the owner of a lead is changed, the new owner receives a notification.
Lead Assigned To You (New) notification	When a new lead is assigned to a user, they receive a notification.
Edit Lead page	Provides a form for updating details of a lead.
Lead Detail page	Provides details of a lead.
My Open Leads page	Provides a table view of a user's open leads with search functionality and a link to details.

Opportunities: Search, view, and edit opportunities.

Notification or Page	Use-case workflows
Opportunity Assigned To You (Existing) notification	When the owner of an opportunity is changed, the new owner receives a notification.
Opportunity Assigned To You (New) notification	When a new opportunity is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Close as Lost page	Provides a form for closing an opportunity as lost with details.
Close as Won page	Provides a form for closing an opportunity as won with details.
Contact Detail page	Provides details of a contact.
Edit Opportunity page	Provides a form for updating details of an opportunity.
My Open Opportunities page	Provides a table view of a user's open opportunities with search functionality and a link to details.
Opportunity Detail page	Provides details of an opportunity.

Phone Calls: Search, view, and edit phone calls.

Notification or Page	Use-case workflows
Phone Call Assigned To You (Existing) notification	When the owner of a phone call is changed, the new owner receives a notification.
Phone Call Assigned To You (New) notification	When a new phone call is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Case Detail page	Provides details of a case.
Contact Detail page	Provides details of a contact.
Edit Phone Call page	Provides a form for updating details of a phone call.
My Open Phone Calls page	Provides a table view of a user’s phone calls with search functionality and a link to details.
Opportunity Detail page	Provides details of an opportunity.
Phone Call Detail page	Provides details of a phone call.

Tasks: Search, view, and edit tasks.

Notification or Page	Use-case workflows
Task Assigned To You (Existing) notification	When the owner of a task is changed, the new owner receives a notification.
Task Assigned To You (New) notification	When a new task is assigned to a user, they receive a notification.
Account Detail page	Provides details of an account with contacts listed and a link to contact details.
Case Detail page	Provides details of a case.
Contact Detail page	Provides details of a contact.
Edit Task page	Provides a form for updating details of a task.
Lead Detail page	Provides details of a lead.
My Open Tasks page	Provides a table view of a user’s open tasks with search functionality and a link to details.
Opportunity Detail page	Provides details of an opportunity.
Task Detail page	Provides details of a task.

Integrate Microsoft Outlook

April 20, 2021

Deploy the Microsoft Outlook integration to schedule events and office hours, edit events and office hours, and receive a notification an hour before an event's start time.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapp for Microsoft Outlook, see Use Microsoft Outlook microapps.

Review prerequisites

After you set up this integration with Microsoft Outlook, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- BASE URL: `https://graph.microsoft.com/`
- AUTHORIZATION URL: `https://login.microsoftonline.com/{ tenantId } /oauth2/v2.0/authorize`
- TOKEN URL: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/token`
- CLIENT ID: The client ID is the string representing client registration information unique to the authorization server.
- SECRET: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for Microsoft Outlook so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service «https://docs.citrix.com/en-us/citrix-gateway-service/>.

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Microsoft Outlook API limitation form link: https://docs.microsoft.com/en-us/graph/throttling#microsoft-teams-service-limits
- Recommended plan: https://www.microsoft.com/en-in/microsoft-365/microsoft-teams/compare-microsoft-teams-options

Permissions

The integration requires regular access to your Microsoft Outlook instance, so we recommend creating a dedicated user account. You can view the permission/privileges at https://docs.microsoft.com/en-us/graph/permissions-reference.

This account must have the following permissions: **Global Administrator**. The Global Administrator role grants admin consent for application permissions and access API's in Microsoft Outlook.

Sign in here: https://account.microsoft.com/.

Configure OAuth server

Configure the OAuth server to read data through the Microsoft Outlook integration.

1. Log in with your service account to: https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps
2. Select **New registration**.
3. For **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multitenant)**.
4. Complete the required fields and enter the following authorized redirect URLs for this integration in the Redirect URL field: `https://{ yourmicroappserverurl } /admin/api/gwsc /auth/serverContext`
5. Click on **Register**.
6. Copy and save the **Application (client) ID** and **Directory (tenant) ID** shown on the screen. You use these details for Service Authentication while configuring the integration.
7. Click on **View Permissions** under Call API's and select **Add a permission** and choose **Microsoft Graph** tile.
8. Select **Application permissions** tile and add these listed scopes: `**User.Read.All Calendars.Read**`
9. Select **Grant admin consent for Citrix Systems** and select Yes.
10. Select **Certificates & secrets** from left panel and select **New client secret** and choose the expiration validity as never and click on add.
11. Copy and save the **Value** from the client secrets.

Configure OAuth client

Configure the OAuth client to write back data through the Microsoft Outlook integration.

1. Log in with your service account to: https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps
2. Select **New registration**.
3. For **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multitenant)**.
4. Complete the required fields and enter the following authorized redirect URLs for this integration in the Redirect URL field: `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
5. Click on **Register**
6. Copy and save the **Application (client) ID** and **Directory (tenant) ID** shown on the screen. You use these details for Service Authentication while configuring the integration.
7. Click on **View Permissions** under Call API's and select **Add a permission** and choose **Microsoft Graph** tile.
8. Select **Delegated permissions** tile and add these listed scopes: `**Calendars.ReadWrite**`
9. Select **Grant admin consent for Citrix Systems** and select Yes.
10. Select **Certificates & secrets** from left panel and select **New client secret** and choose the expiration validity as never and click on add.
11. Copy and save the **Value** from the client secrets.

Add the integration to Citrix Workspace Microapps

Add the Microsoft Outlook integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.

2. Choose the Microsoft Outlook tile.

3. Enter an **Integration name** for the integration.

4. Enter **Connector parameters**.

    - Enter the instance **Base URL:** `https://graph.microsoft.com/`
    - Select an Icon for the integration from the Icon Library, or leave this as the default icon.

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that

these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

a) Select **Client Credentials** from the **Grant type** menu.
b) Select **Request body** from the **Token authorization** menu.
c) The **Token URL** is prefilled: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/token`
d) Ensure the following is entered for Scope: `.default offline_access`
e) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configure the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.
f) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication** in Actions toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Request body** from the **Token authorization** menu.
   c) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the Callback URL, which you use when registering your application.
   d) The **Authorization URL** is prefilled: `https://login.microsoftonline.com/{ tenantId } /oauth2/v2.0/authorize`
   e) The **Token URL** is prefilled: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/token`
   f) Ensure the following is entered for Scope: `.default offline_access`
   g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configure the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.
   h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

7. Enable the **Enable request rate limiting** toggle. Enter 60 for **Number of requests** and 1 second for **Time interval**.
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
9. Select **Save** to proceed.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

> **Note:**
>
> The Citrix Microsoft Outlook integration uses Data Update After Action to pull in the most recent data for the logged in user via the **Refresh Table** button in the My Calendar service action. We recommend to use this approach as is. Please utilize the default full synchronization once every week for retaining an optimum amount of data for the user. Additionally, the integration doesn't support incremental synchronization and relies solely on Data Update After Action to pull in the most recent data. It is recommended to set the "Full Synchronization" interval as **Weekly** to remove the cancelled or deleted events from Microapps platform and subsequently from the user's calendar.
>
> **Refresh** button is used to sync cache with the most recent data, in lieu of full/incremental synchronization. Since this integration doesn't rely on full/incremental sync for latest data pull, pagination is not needed nor implemented. This also helps limit api calls.

For more details of API endpoints and table entities, see MS Outlook connector specifications.

> **Note:**
>
> Calendarview data endpoint is hardcoded with past start_date_time and end_date_time, since they are mandatory. However, the user will view the most recent data in their microapps using the **Refresh** button concept.

Use Microsoft Outlook microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note:**
>
> As the currently available 40 time zones are hardcoded in the **Create Event**, **My Office Hours** and **My Calendar** microapps, adding any other time zone would require the admin to add them manually.

Create Event: Microapp is used to schedule an Event/Meeting as per the user preference.

Notification or Page	Use-case workflows
Create Event page	Provides a form to schedule an event with the following details according to user preference: Event Title, Start Date/Time, End Date/Time, TimeZone, Recurrence (once, daily, weekly, monthly), Location, Description and Attendees or Guests for the meeting.

My Calendar: Microapp is used to view and edit upcoming Events/Meetings.

> **Note:**
>
> The **Event Reminder** notifications are triggered only to the event organizer one hour before event start_date_time. This notification only gets triggered for the events which are in the Microapp server cache at any given point. To ensure timely notifications, we recommend using the Refresh Table button on a frequent basis and also running the weekly Full sync as is, to avoid any incorrect/deleted event notifications.

Notification or Page	Use-case workflows
Event Reminder notification	Event owner receives a notification before an hour of the event start time.
Upcoming Event Detail page	Provides a read only view of an event with details, button to join meeting and edit button for event owner only.
Upcoming Events page	Allows users to search for events.
Event Detail page	Provides a read only view of event with details. Edit option is available for the event owner.
Edit Event page	Provides a form for editing an event.

My Office Hours: Microapp is used to create, view, and edit virtual office hours.

Notification or Page	Use-case workflows
Virtual Office Hours page	Allows users to view the office hours.
Create Office Hours page	Provides a form to schedule virtual office hours with the following details according to user preference: Start Date/Time, End Date/Time, TimeZone, Recurrence (daily, weekly, monthly), Description for the office hours.
Edit Office Hours page	Provides a form for editing office hours.

Integrate Microsoft Teams

October 26, 2021

Deploy the Microsoft Teams integration to schedule Teams Meetings, create a team from scratch or based on an existing team, add a new channel to an existing team, send a message to a specific channel and receive a notification for newly created channels.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapp for MS Teams, see Use Microsoft Teams microapps.

Review prerequisites

These prerequisites assume that the administrator is part of the MS Teams integration set up of the organization. This MS Teams admin account must have full read privileges for user information. After you set up this integration with Microsoft Teams, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- BASE URL: `https://graph.microsoft.com/`
- AUTHORIZATION URL: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/authorize`
- TOKEN URL: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/token`
- CLIENT ID: The client ID is the string representing client registration information unique to the authorization server.

- SECRET: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for MS Teams so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.

Permissions

The integration requires regular access to your MS Teams instance, so we recommend creating a dedicated user account. You can view the permission/privileges at https://docs.microsoft.com/en-us/graph/permissions-reference.

This service account must have either one of the following permission scope setups:

- **Global Administrator** or
- **Application Administrator** and **Teams Service Administrator**

Details of the roles:

- **Global Administrator** role grants admin consent for delegated permissions in Microsoft Teams and allows API access.
- **Application Administrator** role is grants admin consent for delegated permission.
- **Teams Service Administrator** role is required to access the channel API.

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Microsoft Teams API limitation form link: https://docs.microsoft.com/en-us/graph/throttling#microsoft-teams-service-limits

Create a new service account

Sign in here: https://portal.azure.com. For more information about getting started with Microsoft Teams, see https://support.microsoft.com/en-us/office/how-do-i-get-microsoft-teams-fc7f1634-abd3-4f26-a597-9df16e4ca65b.

Configure OAuth server

Configure the OAuth server to read data through the MS Teams integration.

1. Log in with your service account to: https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps.
2. Select **New registration**.
3. For **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multitenant)**.
4. Complete the required fields and enter the following authorized redirect URLs for this integration in the **Redirect URL** field:

   ```
   https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext
   ```

5. Select **Register**.
6. Copy and save the **Application (client) ID** and **Directory (tenant) ID** shown on the screen. You use these details for **Service Authentication** while configuring the integration.
7. Select **View Permissions** under **Call APIs**. Select **Add a permission** and choose the **Microsoft Graph** tile.
8. Select **Delegated permissions** tile and add the below listed scopes:

   ```
   Group.Read.All User.Read.All GroupMember.Read.All Channel.ReadBasic.All
   ```

9. Select **Grant admin consent for Citrix Systems**, and select **Yes**.
10. Select **Certificates & secrets** from the left panel, and select **New client secret**. Choose **never** for expiration validity, and select **Add**.
11. Copy and save the **Value** from the client secrets.

Configure OAuth client

Configure the OAuth client to write back data through the MS Teams integration.

1. Log in with your service account to: https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps.
2. Select **New registration**.
3. For **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multitenant)**.
4. Complete the required fields and enter the following authorized redirect URLs for this integration in the **Redirect URL** field:

   ```
   https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback
   ```

5. Select **Register**.

6. Copy and save the **Application (client) ID** and **Directory (tenant) ID** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.
7. Select **View Permissions** under **Call APIs**. Select **Add a permission** and choose the **Microsoft Graph** tile.
8. Select **Delegated permissions** tile and add the below listed scopes:

   `Channel.Create Group.ReadWrite.All ChannelMessage.Send Calendars.ReadWrite`
9. Select **Grant admin consent for Citrix Systems**, and select **Yes**.
10. Select **Certificates & secrets** from the left panel, and select **New client secret**. Choose **never** for expiration validity, and select **Add**.
11. Copy and save the **Value** from the client secrets.

Add the integration to Citrix Workspace Microapps

Add the Microsoft Teams integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Microsoft Teams tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**: `https://graph.microsoft.com/`
   - Select an Icon for the integration from the Icon Library, or leave this as the default icon.
5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
   a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization

server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.

b) Select **Request body** from the **Token authorization** menu.

c) The **Authorization URL** is prefilled: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/authorize`

d) The **Token URL** is prefilled: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/token`

e) Ensure the following is entered for Scope: *`https://graph.microsoft.com/default offline_access`*

f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configure the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.

   b) Select **Request body** from the **Token authorization** menu.

   c) The **Authorization URL** is prefilled: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/authorize`

   d) The **Token URL** is prefilled: `https://login.microsoftonline.com/{ tenant_id } /oauth2/v2.0/token`

   e) Ensure the following is entered for Scope: *`https://graph.microsoft.com/default offline_access`*

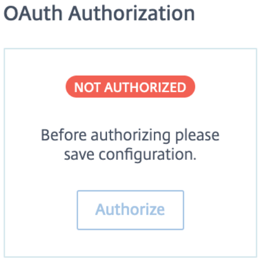
   f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configure the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.

   g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

7. Enable the **Enable request rate limiting** toggle. Enter 60 for **Number of requests** and 1 second for **Time interval**.

8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

9. Select **Save** to proceed.
10. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Microsoft login screen.
    a) Enter your Service Account username and password and select **Sign in**.
    b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

> **Note:**
>
> - It is recommended to set the Full Synchronization interval as **Daily** to regularly refresh data from MS Graph to the Microapps platform and receive timely notifications for any newly created channels.
> - As the currently available 40 time zones are hardcoded in the **Create Meeting** microapp, addition of any other time zone would require the admin to add them manually.
> - When a user creates a channel using **Add Channel** or **Create Team** microapp, the newly created channel is hidden by default in MS Teams.
> - We have currently hardcoded the template list in **Create Team** microapp. To add any other template type, the admin must add them manually.
> - To populate only Microsoft365 (Teams) related groups/channels, we use a filter used in the **Groups** endpoint:`filter=groupTypes/any(g:g+eq+'Unified'`. Note that + has been replaced by a blank space.
> - If users are getting additional Teams in the **Select Team** component in **Send Message** and **Add channel** microapps, use the beta endpoint from Microsoft at `https://graph.microsoft.com/beta/groups?$filter=grouptypes/any(g:g eq 'Unified')` and `resourceProvisioningOptions/any(p:p eq 'Team')` to filter only Teams(Groups) related to MS Teams.

The **Microapp Integrations** page opens with your added integration and its microapps. From here, you can add another integration, continue setting up your out-of-the-box Microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Microsoft Teams connector specifications.

Use MS Teams microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Add Channel: Add a new channel to an existing team.

Notification or Page	Use-case workflows
Add Channel page	Provides a form for adding channel to an existing team with the following details: Team (Teams drop-down), channel name and description.

Create Meeting: Schedule an MS Teams meeting as per user preference.

Notification or Page	Use-case workflows
Create Meeting page	Provides a form to schedule a meeting with the following details according to user preference: Meeting Title, Start Date/Time, End Date/Time, TimeZone, Recurrence (once, daily, weekly, monthly), Description and Attendees for the meeting.

Create Team: Create a team from scratch or based on an existing team as per user preference. Additionally, whenever a Channel is created for any team, the team owner will receive a notification.

Notification or Page	Use-case workflows
New channel has been added notification	When a new channel is added to a team, the team owner receives the notification.
Channel Details page	Provides a read only view of a newly created channel with Channel Details and Channel Members.

Notification or Page	Use-case workflows
Create Team/Channel page	Provides two buttons; **From Scratch** which navigates to the Create Team from Scratch page, and **From Existing Team** which navigates to the Create Team from Group page.
Create Team from Scratch page	Provides a form to Create a team from Scratch with the following details : Team Name, Team Description, Type of the team (Private / Public), Template (drop down with different Template options), Channel Name, Channel description, Add to favorite check box, Tab Name, Content URl, Member Settings and Discovery Settings.
Create Team from Group page	Provides a form to create a team from an existing team with the following details : Team (Team drop-down), Team Name, Type of the Team (Private / Public), Team Description and Parts to include from the original team.

Send Message: Send a message to a specific channel in any team.

Notification or Page	Use-case workflows
Send Message to a Channel page	Provides a form to send a message to a channel of an existing team with the following details: Team (Teams drop-down), Channel (Channel drop-down), and Message.

Integrate Oracle HCM

August 24, 2021

Deploy the Oracle HCM integration to deliver actionable notifications to item managers about their items, and view and edit items directly in Workspace.

For comprehensive details of the out-of-the-box microapps for Oracle HCM, see Use Oracle HCM microapps.

Review prerequisites

These prerequisites assume that the administrator is part of the Oracle HCM integration set up of the organization.

You need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: The base URL follows this model: `{ serverURL } /hcmRestApi/resources`. Find the REST serverURL in the welcome email that was provided to your Oracle Cloud service administrator. Replace the `{ serverURL }` value in our model with your serverURL.
- **Username**: We recommend using the username and password of the dedicated service account as specified below.
- **Password**: We recommend using the username and password of the dedicated service account as specified below.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Permissions

The integration requires regular access to your Oracle HCM instance. Create a dedicated service account in Oracle HCM for this implementation. The account requires access to this data only:

- employee directory
- absences
- time entry

Adding the integration to Citrix Workspace Microapps

Add the Oracle HCM integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Oracle HCM tile.
3. Enter a name for the Integration.
   - Enter the instance **Base URL**: `https://adc4-zrha-fa-ext.oracledemos.com/hcmRestApi/resources`.This base URL value might be different.
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.

4. Under **Service authentication**, select **Basic** from the **Authentication method** menu and complete the authentication details.
5. Enter your **Username** and **Password**.
6. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes. Leave all other toggles disabled.
7. Select **Save** to proceed.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Oracle HCM connector specifications.

Use Oracle HCM microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Employee Directory: View and search employee directory.

Notification or Page	Use-case workflows
Employee Search page	Search for an employee by name.
Employee Detail page	Details of selected employee after searching.

Enter My Time: Enter daily time for selected entry types.

Notification or Page	Use-case workflows
Time Entry page	Submit a time card (using deeplink) or time entry with start and end time for the selected entry type.

PTO: Create and submit absences.

Notification or Page	Use-case workflows
Create Absence page	Provides a form with absence parameters to create a new absence entry.

Integrate Power BI

July 2, 2021

Integrate with Power BI to track important reports and dashboards in Citrix Workspace.

Use the following process to enable the Power BI Integration. Ensure you meet the prerequisites and provide connection details. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

For a comprehensive list of out-of-the-box Power BI microapps, see Use Power BI microapps.

Review prerequisites

You must have a Power BI service account. For more information, go to https://powerbi.microsoft.com/en-us/landing/signin/. For any report to be shared through Citrix Microapps, the report must be shared with the Power BI service account.

Depending on your set-up, potentially two accounts are needed:

- An account that has permissions to create an app registration in Azure AD. This account might be problematic depending on which subscription/tenant it associates with first.
- A service account that is used to authenticate the integration.

These prerequisites assume you administer the Power BI instance of your organization to set up the integration. The service account must have the following API Permissions assigned with **Type: Delegated**:

Group	API/Permissions name	Type	Description
Azure Active Directory Graph (1)	User.Read	Delegated	Sign in and read user profile.
Power BI Service (18)	App.Read.All	Delegated	View all Power BI apps.
	Capacity.Read.All	Delegated	View all capacities.
	Dashboard.Read.All	Delegated	View all dashboards.
	Dataflow.Read.All	Delegated	View all dataflows.
	Dataset.Read.All	Delegated	View all datasets.
	Gateway.Read.All	Delegated	View all gateways.
	Report.Read.All	Delegated	View all reports.

Group	API/Permissions name	Type	Description
	StorageAccount.Read.A	Delegated	View all storage accounts.
	Workspace.Read.All	Delegated	View all workspaces
	Tenant.Read.All	Delegated	View all content in tenant

After you set up this integration in Power BI, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- Client ID
- Client Secret
- OAuth Authorization

Register your application

Navigate to https://dev.powerbi.com/apps, and register your application with Azure AD to allow your application to access the Power BI REST APIs and to set resource permissions for your application. Through this registration process, you create your Client ID and Client Secret.

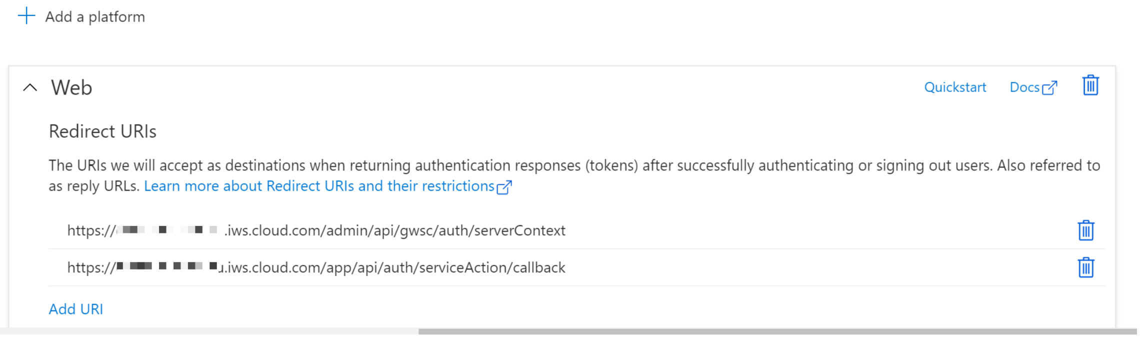
The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region(us/eu/ap-s)} .iws.cloud.com`.

You can only register one URL in the first field. Enter one URL in the **Server-side web application** registration page. Then go to *Azure AD App Registration* to add the second URL.

1. Enter a name for your application.
2. Select **Server-side web application**.
3. Enter your application's **Home Page URL**. This value must be the URL of the Citrix cloud tenant (the Microapps server URL). For example, `https://<customer_id>.us.iws.cloud.com`. You can find this Microapps server URL in the URL bar when logged in to Citrix Microapps.
4. Enter one of the following **Redirect URLs**:
   `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
   `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`

> **Note:**
>
> Two URLs are not permitted in this field. Go to *Azure AD App Registration* to add the second URL.

5. Select all read only APIs check boxes.
6. Select **Register**.
7. Navigate to **Azure App Registrations** > **Authentication** and enter the second callback URL.



Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating or signing out users. Also referred to as reply URLs. Learn more about Redirect URIs and their restrictions

https://[redacted].iws.cloud.com/admin/api/gwsc/auth/serverContext

https://[redacted].iws.cloud.com/app/api/auth/serviceAction/callback

Add the integration to Citrix Workspace Microapps

Add the Power BI integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Power BI tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**: `https://api.powerbi.com/v1.0/myorg`
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default ServiceNow icon.
   - Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.
5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always

use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.
b) Select **Request body** from the **Token authorization** menu.
c) The **Authorization URL** is prefilled: `https://login.microsoftonline.com/{ tenantID } /oauth2/authorize`
d) The **Token URL** is prefilled: `https://login.microsoftonline.com/{ tenantID } /oauth2/token`
e) Ensure the following is entered for **Scope**: *Tenant.Read.All*
f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.
g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.
h) Under **Access token parameters**, ensure that the following is entered for **Name**: *resource*, and for **Value**: `https://analysis.windows.net/powerbi/api`.

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

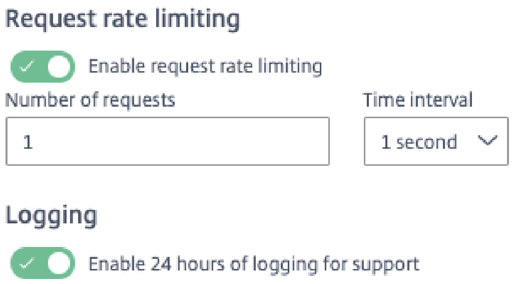
a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.
c) Select **Request body** from the **Token authorization** menu.
d) The **Authorization URL** is prefilled: `https://login.microsoftonline.com/{ tenantID } /oauth2/authorize`
e) The **Token URL** is prefilled: `https://login.microsoftonline.com/{ tenantID } /oauth2/token`
f) Ensure the following is entered for **Scope**: *Tenant.Read.All*
g) Enter your **Client ID**. The client ID is the string representing client registration information

unique to the authorization server. You collect this and the secret when you configured the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.

h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

i) Under **Access token parameters**, ensure that the following is entered for **Name**: *resource*, and for **Value**: `https://analysis.windows.net/powerbi/api`.

7. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

✓ Enable request rate limiting

Number of requests: 1

Time interval: 1 second

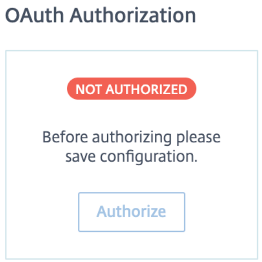
Logging

✓ Enable 24 hours of logging for support

9. Select **Save** to proceed.
10. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Power BI login screen.

    a) Enter your Service Account user name and password and select **Log in**.

    b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Power BI connector specifications.

Use Power BI microapps

Existing Web/SaaS integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our Power BI integration template comes with the following preconfigured out-of-the-box microapps:

Dashboards: View details of Power BI dashboards. The list of dashboards is personalized, so you only see the dashboards that are part of your Citrix Workspaces in Power BI.

Notification or Page	Use-case workflows
Dashboards page	Provides a list of available dashboards with a link to a page with details.
Dashboards Detail page	Provides a read only detailed view of a dashboard with a link to the target source of record for a more detailed view.

Reports: View details of Power BI reports. The lists of reports is personalized, so you only see the reports that are part of your Citrix Workspaces in Power BI.

Notification or Page	Use-case workflows
Report Detail page	Provides a read only detailed view of a report.
Reports page	Provides a list of available reports with a link to a page with details with a link to the target source of record for a more detailed view.

Integrate Qualtrics

April 20, 2021

Deploy the Qualtrics integration to receive notifications about surveys that require a response, view active surveys requiring attention, and also to allow the survey manager to access survey statistics. Qualtrics APIs and this integration require token based authentication. No service actions are supported. All user updates to surveys are executed through deep links.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For

> any issues, our team will also monitor our dedicated forum on a daily basis.

Review prerequisites

These are the values that you enter in Citrix Workspace Microapps:

- **Base URL**: `https://{ instance server location code } .qualtrics.com/API/v3/`
- **Token Parameter**: `X-API-Token`
- **Token**: The token permits API access. See Generate token.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with your configured microapp.

Create a new service account

The integration requires access to your Qualtrics instance. We recommend creating a dedicated user account for each functional group using Qualtrics, such as Sales, Customer Satisfaction, or Customer Success. As a result, the statistics gathered from Qualtrics surveys can be tailored for their functional group.

This account must have the following permissions: **Brand Admin role**. In a Qualtrics implementation with more than one admin, any projects that belong to another Brand Admin must be shared through collaboration with the integration brand admin account to be available in the integration. Alternately, an integration can be created for each Brand Admin in the enterprise.

Generate token

You need to generate a token for API access to Qualtrics.

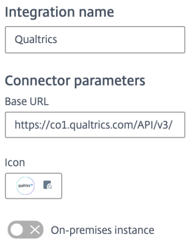
1. Navigate to PM:URL needed
2. Select **Account Setting** and then the **Qualtrics IDs** tab.
3. Under **API**, select **Generate Token**.
4. Copy and save the token for adding the integration in the next procedure.

Add the integration to Citrix Workspace Microapps

Add the Qualtrics integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions that are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration** and **Add a new integration from Citrix-provided templates**.
2. Choose the Qualtrics tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**:
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

Qualtrics

Connector parameters

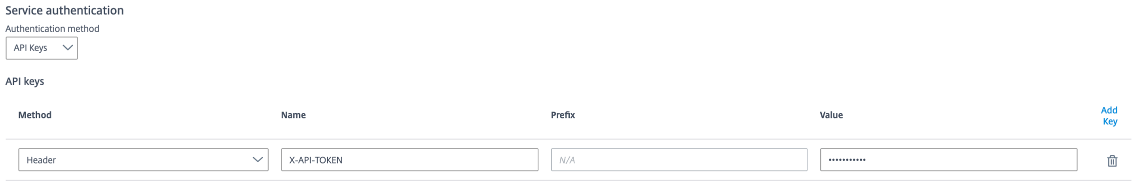
Base URL

https://co1.qualtrics.com/API/v3/

Icon

On-premises instance

5. Under **Service Authentication**, select **API Keys** from the **Authentication method** menu and enter the token value that you collected in the **Value** field.



Service authentication

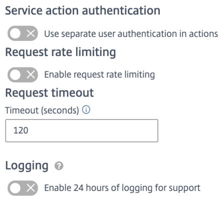
Authentication method

API Keys

API keys

Method	Name	Prefix	Value	Add Key
Header	X-API-TOKEN	N/A	•••••••••••	🗑

6. Leave the **Enable request rate limiting** toggle as disabled.
7. Leave **Request timeout** as default.
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
9. Select **Save** to proceed.



Service action authentication

Use separate user authentication in actions

Request rate limiting

Enable request rate limiting

Request timeout

Timeout (seconds)

120

Logging

Enable 24 hours of logging for support

The **Microapp Integrations** page opens with your added integration and its microapps. You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Qualtrics connector specifications.

Use Qualtrics microapps

My Surveys: Find all active surveys that are emailed to users. Notifications are sent when a survey is sent to a user and 24 hours before a survey expires.

Notification or Page	Use-case workflows
New Survey notification	When a new survey is created, a notification is sent to the user listed as the contact.
Survey about to expire notification	When a survey is due to expire in 24 hours, a notification is sent to the user listed as the contact.
My active Surveys page	Provides a personalized list of active surveys with a link to survey details.
Survey details page	Provides a detailed view of a survey with a link to button to **Respond to Survey in Qualtrics**.

Qualtrics Survey Statistics: For Brand Admins to manage surveys. Review statistics on active surveys. Notifications sent 24 hours before a survey expires with basic performance stats and the ability to deep link.

Notification or Page	Use-case workflows
Active Survey Stats notification	When a survey is due to expire in 24 hours, a notification is sent to subscribers.
Distribution Data page	Provides details of a survey, with a link to **Review Distributions**.
Open Surveys page	Provides a list of surveys, with a link to more details.
Results of a Survey	Provides results of a survey with a link to **Distribution Statistics**.

Integrate RSS

October 22, 2021

Deploy the RSS integration template to follow the Citrix Blogs channel. With this workflow, you remove the need to manually check the website for new content.

You can customize this template to follow the RSS feed of any chosen online channel. For more information, see Customize RSS template.

For a comprehensive list of out-of-the-box RSS microapps, see Use RSS microapps.

> **Note**
>
> We provide two RSS integration templates for your use. We recommend using the newer template in the **Integrations** category for most use-cases as it provides more power to configure the cached data structure. The second template is found in the **Deprecated** category.

Review prerequisites

The template provides a pre-filled URL to follow the Citrix Blogs channel: `http://feeds.feedblitz.com/`. Customize this URL if you want to use this microapp for another RSS feed.

Customize RSS template

To customize the RSS integration template for any channel that you want, you need to split the endpoint URL into its base and its data loading endpoint.

For example, if we take `http://feeds.bbci.co.uk/news/world/rss.xml`:

- The base URL is `http://feeds.bbci.co.uk/news/world/`. Replace the **Base URL** value with this value when you add the template integration.
- The data loading endpoint is `rss.xml`. Replace the RSS endpoint **Full synchronization** name with this value. This can be done before or after configuring the integration template, including the base URL.

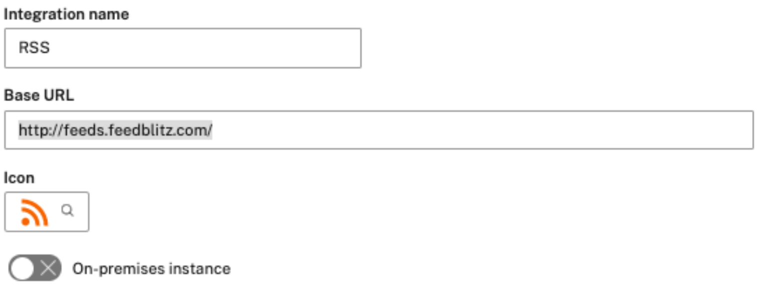
1. To replace the RSS endpoint, from the **Microapp Integrations** page, select the menu next to the RSS integration, and then **Edit**. The **Data Loading** screen opens. If you are in the configuration screen, select **Data Loading** from the left side navigation column.
2. Select the RSS endpoint.
3. Locate the field **Name** under the section **Full synchronization**, and replace the **X** with this data loading endpoint value.
4. Don't forget to select **Apply** at the bottom of the screen and confirm to save the endpoint change.

Add the integration to Citrix Workspace Microapps

Add the RSS integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the **RSS** tile.
3. Enter an **Integration name** for your integration. The template provides a pre-filled URL to follow the Citrix Blogs channel: `http://feeds.feedblitz.com/`. Customize this URL if you want to use this microapp for another RSS feed.
4. Enter the **Base URL**.



Integration name: RSS

Base URL: http://feeds.feedblitz.com/

Icon

On-premises instance

5. Leave all other fields disabled and **Request timeout** set as *120*.
6. Select **Save** to finish.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see RSS connector specifications.

Use RSS microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our RSS integration comes with the following preconfigured out-of-the-box microapps:

Feeds: Search for and view items.

Notification or Page	Use-case workflows
New Feed notification	When a user has a new RSS item, the user receives a notification.
Feed Details page	Provides a read only detailed view of an RSS item of interest for a user.
View all Feeds page	Provides a list of RSS items of interest for a user with a link to view details.

Integrate Salesforce

April 28, 2021

Integrate with Salesforce for anywhere access to leads, accounts, opportunities, cases, and contracts. Use the following process to enable the Salesforce HTTP integration. Ensure you meet the prerequisites then set up the Salesforce integration.

> **Note**
>
> We provide two Salesforce integration templates for your use. We recommend using the newer HTTP integration for most use-cases as it provides more power to configure the cached data structure. The **Salesforce** template is the basis for the Salesforce HTTP integration. For full details of the microapps available in each integration, see Use Salesforce microapps.

For a comprehensive list of out-of-the-box Salesforce HTTP microapps, see Use Salesforce microapps.

Review prerequisites

After you set up this integration in Salesforce, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- Username
- Password
- Security Token

For OAuth 2.0:

- Consumer Key
- Consumer Secret
- OAuth Authorization Base Url

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

The integration requires regular API access to your Salesforce instance, so we recommend creating a dedicated user account in Salesforce. Then use that account to configure the Salesforce integration. This account must:

- be given full data access privileges
- be API-enabled
- not allow two-factor authentication.

Using a dedicated account is useful for audit logs as it helps distinguish activities done through Workspace. This page contains tutorials for both Salesforce Classic and Salesforce Lightning Experience. Both tutorials assume that you are a System Administrator in Salesforce.

> **Note:**
>
> Salesforce "Contact" and "Group" Editions do not support any API. The "Professional" Edition does not include it automatically. However, support can be activated upon request.

Also the number of API requests are limited in Salesforce. If you plan to frequently synchronize an extensive amount of data, see Salesforce API Request Limits and Allocations.

- Configure Citrix Gateway to support single sign-on for Salesforce so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Salesforce single sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Add a new profile

Follow these steps:

1. Log in to www.salesforce.com
2. Go to **Setting** icon and select **Setup > Administration > Manage Users > Profiles > New Profile**.
3. Set **Existing Profile** to **System Administrator** to ensure that the user that you create for this profile has full data access privileges.
4. Enter a Profile Name, and select **Save**. We recommend naming the profile something like *Citrix Workspace Access* for easy reference when adding the profile as a new user in a following procedure.

   A Profile panel opens with your new profile.

Enable API access for the created profile

1. On the **Profile** panel, select **Edit**.
2. Scroll down to Administrative Permissions and select the **API Enabled** check box.
3. (Optional) To disable password expiration, select the **Password Never Expires** check box.

   > **Note:**
   >
   > Using this option is a potential security vulnerability.

4. Select **Save**.

Add callback URLs

Add a custom URL to your instance configuration to grant access to private data and enable OAuth authenticated user actions. The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com.

1. Log in to Salesforce as an admin.
2. Navigate to **Platform Tools > Apps > App Manager**.
3. Select **New Connected App**.
4. Under **Basic Information**, complete the following fields:
   - **Connected App Name**
   - **API Name**
   - **Contact Email**
5. Under **API**, select the **Enable OAuth Settings** check box.
6. In the **Callback URL** field, add the following authorized redirect URLs with your Microapp server URL:
   - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
   - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
7. Next to **Selected OAuth Scopes**, choose the following scopes under **Available OAuth Scopes**, and then select **Add** to move them to the **Selected OAuth Scopes** field:
   - **Access and manage your data (api)**
   - **Access your basic information (id, profile, email, address, phone)**
   - **Perform requests on your behalf any time (refresh_token, offline_access)**

8. Select **Save**.

(Optional) Restrict log-in IP ranges

If your organization sets IP ranges for User Profiles, you can control log-in access at the user level. Specify a range of allowed IP addresses on a user's profile. For more information, see Restrict Log-in IP Ranges in the Enhanced Profile User Interface.

If you restrict log-in IP ranges, you do not need to generate a security token in a following procedure.

Add a new user

Create a dedicated user account that is used to connect to Salesforce. Use the new profile that you added in the previous procedure, Add a new profile.

Follow these steps:

1. Go to **Setup > Administer > Manage Users > Users > New User**.
2. Complete the required fields in red.
3. Set **User License** to **Salesforce**.
4. Set **Profile** to the profile that you added in the previous procedure. In the example above, we recommended *Citrix Workspace Access* for easy reference when adding the profile.
5. Click **Save**.

Set up the new user

After you add a dedicated user account, you receive an email at the address you provided.

Follow these steps:

1. Find the email and click the link as instructed.
2. Log in to Salesforce.
3. Set a password and a password question.

Generate a security token

If you restricted log-in IP ranges for the dedicated user profile, you can skip this step. The security token is not required for accounts connecting to the Salesforce API from a white listed IP block.

Follow these steps:

1. Log in and select the account name.

2. Go to **My Settings > Personal > Reset My Security Token**.

3. Select **Reset Security Token**.

   The new security token is sent to the email address that you provided in the personal settings for this account. You also get a new security token whenever the password for this account is reset.

You can now complete adding the integration. Enter the **Username** and **Password** of the dedicated user account in the input fields of Salesforce service definition.

If you white listed the IP, you don't need to enter a Security Token. Otherwise, paste the Security Token that was sent to the email box of the dedicated account.

Filter queries

Most Salesforce entities support filtering. Choose between predefined queries or write your own custom queries using Salesforce SOQL language. For more information, see Salesforce Object Query Language documentation.

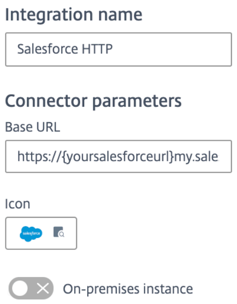
Add the Salesforce integration to Citrix Workspace Microapps

Add the Salesforce HTTP integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps to set up the Salesforce HTTP integration. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the **Salesforce** tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**. This is the domain for your Salesforce environment. `https://{ yoursalesforceurl } .my.salesforce.com`
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default Salesforce icon.
   - Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.



Integration name

Salesforce HTTP

Connector parameters

Base URL

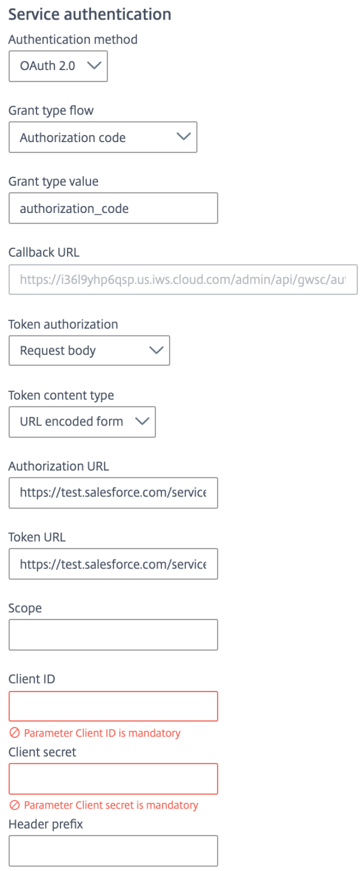
https://{yoursalesforceurl}my.sale

Icon

On-premises instance

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

   a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.
   b) Select **Request body** from the **Token authorization** menu.
   c) The **Authorization URL** and **Token URL** are prefilled. Endpoints require secure HTTP (HTTPS). Instead of using login.salesforce.com, you can also use the My Domain, community, or test.salesforce.com (sandbox) domains in these endpoints.
   d) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.
   e) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.
   f) (Optional) Enter your **Header prefix** if your bearer prefix is different from the default header.



Service authentication

Authentication method

OAuth 2.0

Grant type flow

Authorization code

Grant type value

authorization_code

Callback URL

https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/au

Token authorization

Request body

Token content type

URL encoded form

Authorization URL

https://test.salesforce.com/service

Token URL

https://test.salesforce.com/service

Scope

Client ID

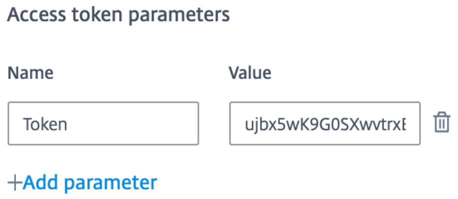
⊘ Parameter Client ID is mandatory

Client secret

⊘ Parameter Client secret is mandatory

Header prefix

6. Select **Add Parameter** to include **Access token parameters**. Enter *Token* for **Name** and { `yoursecuritytoken` } for **Value**. This parameter is required by the target application authorization server.



Access token parameters

Name	Value
Token	ujbx5wK9G0SXwvtrxE

+Add parameter

7. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.

   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.

c) Select **Request body** from the **Token authorization** menu.

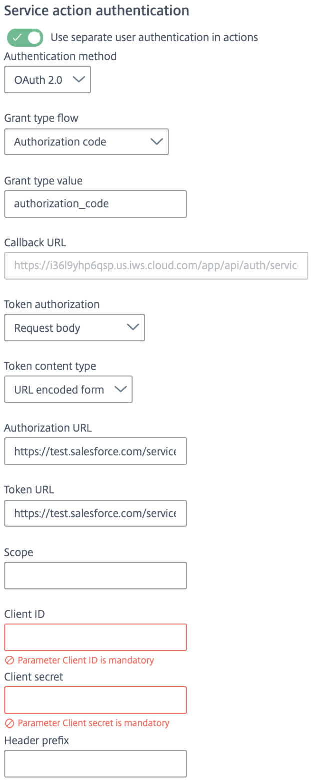
d) The **Authorization URL** and **Token URL** are prefilled. Endpoints require secure HTTP (HTTPS). Instead of using login.salesforce.com, you can also use the My Domain, community, or test.salesforce.com (sandbox) domains in these endpoints.

e) Leave **Refresh token URL** empty.

f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

h) (Optional) Enter your **Header prefix** if your bearer prefix is different from the default header.



Service action authentication

Use separate user authentication in actions

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://test.salesforce.com/service

Token URL: https://test.salesforce.com/service

Scope:

Client ID:

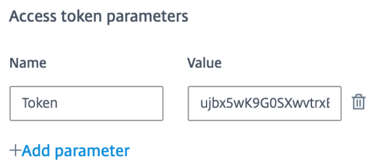
Parameter Client ID is mandatory

Client secret:

Parameter Client secret is mandatory

Header prefix:

8. Again, select **Add Parameter** to include **Access token parameters**. Enter *Token* for **Name** and `{ yoursecuritytoken }` for **Value**. This parameter is required by the target application authorization server.

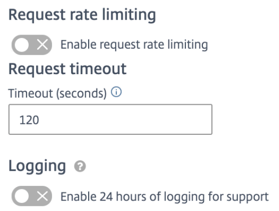


Access token parameters

Name: Token

Value: ujbx5wK9G0SXwvtrxE

9. Enter *120* in the **Request timeout** field.
10. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.
11. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

Enable request rate limiting

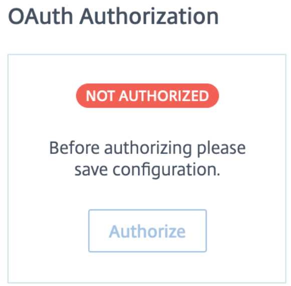
Request timeout

Timeout (seconds): 120

Logging

Enable 24 hours of logging for support

12. Select **Save** to proceed.
13. Now you are able to Authorize to Salesforce with your service account. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Salesforce login screen.
    a) Enter your Service Account user name and password and select **Log in**.
    b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Salesforce http connector specifications.

Use Salesforce microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note**
>
> To use the Convert Lead service action (see **Leads** microapp below), you need to develop a custom Apex code in your Salesforce environment before you add this service action in the builder. For more information, see Apex Developer Guide.

Our Salesforce integration comes with the following preconfigured out-of-the-box microapps:

Accounts: Search for, view, and edit accounts.

Notification or Page	Use-case workflows
Account Assigned To You (Existing) notification	When the owner of an account is changed, the new owner receives a notification.
Account Assigned To You (New) notification	When a new account is assigned to a user, they receive a notification.
Detail Account page	Provides a view of an account with details including contacts and a link to contact details.
Detail Contact page	Provides a read only view of a contact with details.
Edit Account page	Provides a form for submitting edits to an account.
Search Account page	Provides a personalized list of accounts.

Cases: Search for, view, and edit cases that are assigned to you.

Notification or Page	Use-case workflows
Case Assigned To You (Existing) notification	When the owner of a case is changed, the new owner receives a notification.
Case Assigned To You (New) notification	When a new case is assigned to a user, they receive a notification.
Detail Account page	Provides a view of an account with details including contacts and a link to contact details.
Detail Case page	Provides a view of a case and a button for opening the edit page.

Notification or Page	Use-case workflows
Detail Contact page	Provides a read only view of a contact with details.
Edit Case page	Provides a form for submitting edits to a case.
Search Case page	Provides a personalized list of cases that are assigned to a user.

Contacts: Search for, view, and edit contacts.

Notification or Page	Use-case workflows
Contact Assigned To You (Existing) notification	When the owner of a contact is changed, the new owner receives a notification.
Contact Assigned To You (New) notification	When a new contact is assigned to a user, they receive a notification.
Detail Contact page	Provides a view of a contact and a button for opening the edit page.
Edit Contact page	Provides a form for submitting edits to a contact.
Search Contact page	Provides a personalized list of contacts.

Contracts: Search for, view, and edit contracts.

Notification or Page	Use-case workflows
Contract Updated notification	When a detail of a contract is changed, the owner of the contract receives a notification.
Expiring Contract notification	When a contract passes a defined threshold before or after its end date (for example, 3 days by default), the owner receives a notification reminder.
New Contract For Activation notification	When a new pending contract activation approval request is assigned to a user, they receive a notification.
Detail Account page	Provides a view of an account with details including contacts and a link to contact details.

Notification or Page	Use-case workflows
Detail Contact page	Provides a read only view of a contact with details.
Detail Contract page	Provides a view of a contract and a button for opening the edit page and activating the contract.
Edit Contract page	Provides a form for submitting edits to a contract.
Search Contract page	Provides a personalized list of contracts pending activation.

Create Account: Create a new account.

Notification or Page	Use-case workflows
Create Account page	Provides a form for submitting a new account.

Create Case: Create a new case.

Notification or Page	Use-case workflows
Create Case page	Provides a form for submitting a new case.

Create Contact: Create a new contact.

Notification or Page	Use-case workflows
Create Contact page	Provides a form for submitting a new contact.

Create Contract: Create a new contract.

Notification or Page	Use-case workflows
Create Contract page	Provides a form for submitting a new contract.

Create Event: Create a new event.

Notification or Page	Use-case workflows
Create Event page	Provides a form for submitting a new event.

Create Lead: Create a new lead.

Notification or Page	Use-case workflows
Create Lead page	Provides a form for submitting a new lead.

Create Opportunity: Create a new opportunity.

Notification or Page	Use-case workflows
Create Opportunity page	Provides a form for submitting a new opportunity.

Create Task: Create a new task.

Notification or Page	Use-case workflows
Create Task page	Provides a form for submitting a new task.

Events: Search for, view, and edit events.

Notification or Page	Use-case workflows
Event Reminder notification	When an event passes a defined threshold before or after its activity date and time (for example, 1 hour by default), the owner receives a notification reminder.
Detail Account page	Provides a view of an account with details including contacts and a link to contact details.
Detail Contact page	Provides a read only view of a contact with details.
Detail Event page	Provides a view of an event and a button for opening the edit page.

Notification or Page	Use-case workflows
Edit Event page	Provides a form for submitting edits to an event.
Search Event page	Provides a personalized list of events.

Leads: Search for, view, edit, and convert leads.

Notification or Page	Use-case workflows
Lead Assigned To You (Existing) notification	When the owner of a lead is changed, the new owner receives a notification.
Lead Assigned To You (New) notification	When a new lead is assigned to a user, they receive a notification.
Lead Detail page	Provides a view of a lead and a button for opening detail page.
Edit Lead page	Provides a form for submitting edits to a lead.
Search Leads page	Provides a personalized list of leads.

Opportunities: Search for, view, and edit opportunities.

Notification or Page	Use-case workflows
Opportunity Assigned To You (Existing) notification	When the owner of an opportunity is changed, the new owner receives a notification.
Opportunity Assigned To You (New) notification	When a new opportunity is assigned to a user, they receive a notification.
Detail Account page	Provides a view of an account with details including contacts and a link to contact details.
Detail Contact page	Provides a read only view of a contact with details.
Detail Opportunity page	Provides a view of an opportunity and a button for opening the edit page.
Edit Opportunity page	Provides a form for submitting edits to an opportunity.
Search Opportunity page	Provides a personalized list of opportunities.

Tasks: Search for, view, and edit tasks.

Notification or Page	Use-case workflows
Task Reminder notification	When a task passes a defined threshold before or after its activity date and time (for example, 1 hour by default), the owner receives a notification reminder.
Detail Account page	Provides a view of an account with details including contacts and a link to contact details.
Detail Contact page	Provides a read only view of a contact with details.
Detail Task page	Provides a view of a task and a button for opening the edit page.
Edit Task page	Provides a form for submitting edits to a task.
Search Task page	Provides a personalized list of tasks.

Add the legacy integration

Follow these instructions to set up the legacy integration. These procedures are specific to the legacy integration.

Add the legacy integration to Citrix Workspace Microapps

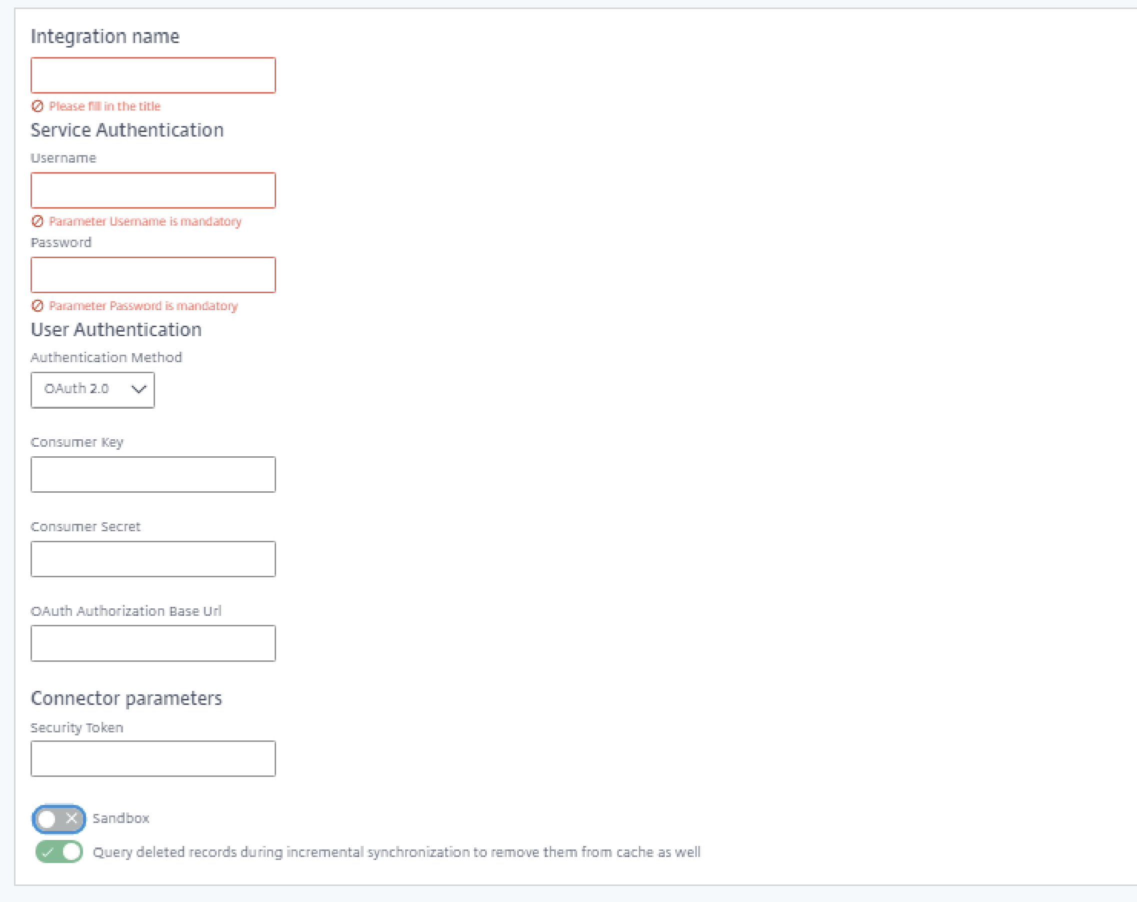
Add the Salesforce integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Salesforce tile.
4. Enter a name for the integration.



Integration name

⊘ Please fill in the title

Service Authentication

Username

⊘ Parameter Username is mandatory

Password

⊘ Parameter Password is mandatory

User Authentication

Authentication Method

OAuth 2.0

Consumer Key

Consumer Secret

OAuth Authorization Base Url

Connector parameters

Security Token

☒ Sandbox

☑ Query deleted records during incremental synchronization to remove them from cache as well

5. Enter the **Service Authentication** that you collected in the previous procedures.

    - Enter your **Username** and **Password** credentials for your target systems service authentication.

6. Select **OAuth 2.0** for **Authentication Method**.

    - Enter the **Consumer Key** and **Consumer Secret** that you collected in the prerequisites procedure.
    - Enter your **OAuth Authorization Base Url**. Allows you to configure a custom SSO login page for your Salesforce instance. Enter your domain. This is the same as the SFDC URL, where you normally log in, and requires secure HTTPS: `https://login.salesforce.com/`, or for sandbox environments: `https://test.salesforce.com/`.

7. Enter your **Connector parameters**.

    - Enter your **Security Token**.
    - Toggle **Sandbox** if you require your data to load into a sandbox environment.
    - Leave toggle **Query deleted records during incremental synchronization to remove them from cache as well** enabled. If the system throws the following error after your first incremental sync: *Unable to load deleted entities of type*, you can disable this toggle to

work around the issue.

8. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Salesforce connector specifications.

Use Salesforce legacy microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our Salesforce integration comes with the following preconfigured out-of-the-box microapps:

Accounts: Search for, view, and edit accounts.

Notification or Page	Use-case workflows
Account Assigned To You (Existing) notification	When the owner of an account is changed, the new owner receives a notification.
Account Assigned To You (New) notification	When a new account is assigned to a user, they receive a notification.
Account Detail page	Provides a view of an account with details including contacts and a link to contact details.
Contact Detail page	Provides a read only view of a contact with details.
Edit Account page	Provides a form for submitting edits to an account.
My Accounts page	Provides a personalized list of accounts.

Cases: Search for, view, and edit cases that are assigned to you.

Notification or Page	Use-case workflows
Case Assigned To You (Existing) notification	When the owner of a case is changed, the new owner receives a notification.
Case Assigned To You (New) notification	When a new case is assigned to a user, they receive a notification.
New Case Related To Your Account notification	When a new case is created that relates to a users account, they receive a notification.
Account Detail page	Provides a view of an account with details including contacts and a link to contact details.
Case Detail page	Provides a view of a case and a button for opening the edit page.
Contact Detail page	Provides a read only view of a contact with details.
Edit Case page	Provides a form for submitting edits to a case.
My Open Cases page	Provides a personalized list of cases that are assigned to a user.

Contacts: Search for, view, and edit contacts.

Notification or Page	Use-case workflows
Contact Assigned To You (Existing) notification	When the owner of a contact is changed, the new owner receives a notification.
Contact Assigned To You (New) notification	When a new contact is assigned to a user, they receive a notification.
Contact Detail page	Provides a view of a contact and a button for opening the edit page.
Edit Contact page	Provides a form for submitting edits to a contact.
My Contacts page	Provides a personalized list of contacts.

Contracts: Search for, view, and edit contracts.

Notification or Page	Use-case workflows
Contract Updated notification	When a detail of a contract is changed, the owner of the contract receives a notification.
Expiring Contract notification	When a contract passes a defined threshold before or after its end date (for example, 3 days by default), the owner receives a notification reminder.
New Contract For Activation notification	When a new pending contract activation approval request is assigned to a user, they receive a notification.
Account Detail page	Provides a view of an account with details including contacts and a link to contact details.
Contact Detail page	Provides a read only view of a contact with details.
Contract Detail page	Provides a view of a contract and a button for opening the edit page and activating the contract.
Edit Contract page	Provides a form for submitting edits to a contract.
My Contracts Pending Activation page	Provides a personalized list of contracts pending activation.

Create Account: Create a new account.

Notification or Page	Use-case workflows
Create Account page	Provides a form for submitting a new account.

Create Case: Create a new case.

Notification or Page	Use-case workflows
Create Case page	Provides a form for submitting a new case.

Create Contact: Create a new contact.

Notification or Page	Use-case workflows
Create Contact page	Provides a form for submitting a new contact.

Create Contract: Create a new contract.

Notification or Page	Use-case workflows
Create Contract page	Provides a form for submitting a new contract.

Create Event: Create a new event.

Notification or Page	Use-case workflows
Create Event page	Provides a form for submitting a new event.

Create Lead: Create a new lead.

Notification or Page	Use-case workflows
Create Lead page	Provides a form for submitting a new lead.

Create Opportunity: Create a new opportunity.

Notification or Page	Use-case workflows
Create Opportunity page	Provides a form for submitting a new opportunity.

Create Task: Create a new task.

Notification or Page	Use-case workflows
Create Task page	Provides a form for submitting a new task.

Events: Search for, view, and edit events.

Notification or Page	Use-case workflows
Event Reminder notification	When an event passes a defined threshold before or after its activity date and time (for example, 1 hour by default), the owner receives a notification reminder.
Account Detail page	Provides a view of an account with details including contacts and a link to contact details.
Contact Detail page	Provides a read only view of a contact with details.
Edit Event page	Provides a form for submitting edits to an event.
Event Detail page	Provides a view of an event and a button for opening the edit page.
My Events page	Provides a personalized list of events.

Leads: Search for, view, edit, and convert leads.

Notification or Page	Use-case workflows
Lead Assigned To You (Existing) notification	When the owner of a lead is changed, the new owner receives a notification.
Lead Assigned To You (New) notification	When a new lead is assigned to a user, they receive a notification.
Convert Lead page	Provides a form for converting a lead.
Edit Lead page	Provides a form for submitting edits to a lead.
My Active Leads page	Provides a personalized list of leads.

Opportunities: Search for, view, and edit opportunities.

Notification or Page	Use-case workflows
Opportunity Assigned To You (Existing) notification	When the owner of an opportunity is changed, the new owner receives a notification.
Opportunity Assigned To You (New) notification	When a new opportunity is assigned to a user, they receive a notification.

Notification or Page	Use-case workflows
Account Detail page	Provides a view of an account with details including contacts and a link to contact details.
Contact Detail page	Provides a read only view of a contact with details.
Edit Opportunity page	Provides a form for submitting edits to an opportunity.
My Open Opportunities page	Provides a personalized list of opportunities.
Opportunity Detail page	Provides a view of an opportunity and a button for opening the edit page.

Pending Account Approvals: Search for and approve or reject accounts.

Notification or Page	Use-case workflows
New Account For Approval notification	When a new account is submitted for an actor's approval, they receive a notification.
Approve Account page	Provides a form for approving or rejecting an account.
My Pending Account Approvals page	Provides a personalized list of pending account approvals and links to the approval page.

Pending Contact Approvals: Search for and approve or reject contacts.

Notification or Page	Use-case workflows
New Contact For Approval notification	When a new contact is submitted for an actor's approval, they receive a notification.
Approve Contact page	Provides a form for approving or rejecting a contact.
My Pending Contact Approvals	Provides a personalized list of pending contact approvals and links to the approval page.

Pending Contract Approvals: Search for and approve contracts.

Notification or Page	Use-case workflows
New Contract For Approval notification	When a new contract is submitted for an actor’s approval, they receive a notification.
Approve Contract page	Provides a form for approving or rejecting a contract.
My Pending Contract Approvals	Provides a personalized list of pending contract approvals and links to the approval page.

Tasks: Search for, view, and edit tasks.

Notification or Page	Use-case workflows
Task Reminder notification	When a task passes a defined threshold before or after its activity date and time (for example, 1 hour by default), the owner receives a notification reminder.
Account Detail page	Provides a view of an account with details including contacts and a link to contact details.
Contact Detail page	Provides a read only view of a contact with details.
Edit Task page	Provides a form for submitting edits to a task.
My Open Tasks page	Provides a personalized list of tasks.
Task Detail page	Provides a view of a task and a button for opening the edit page.

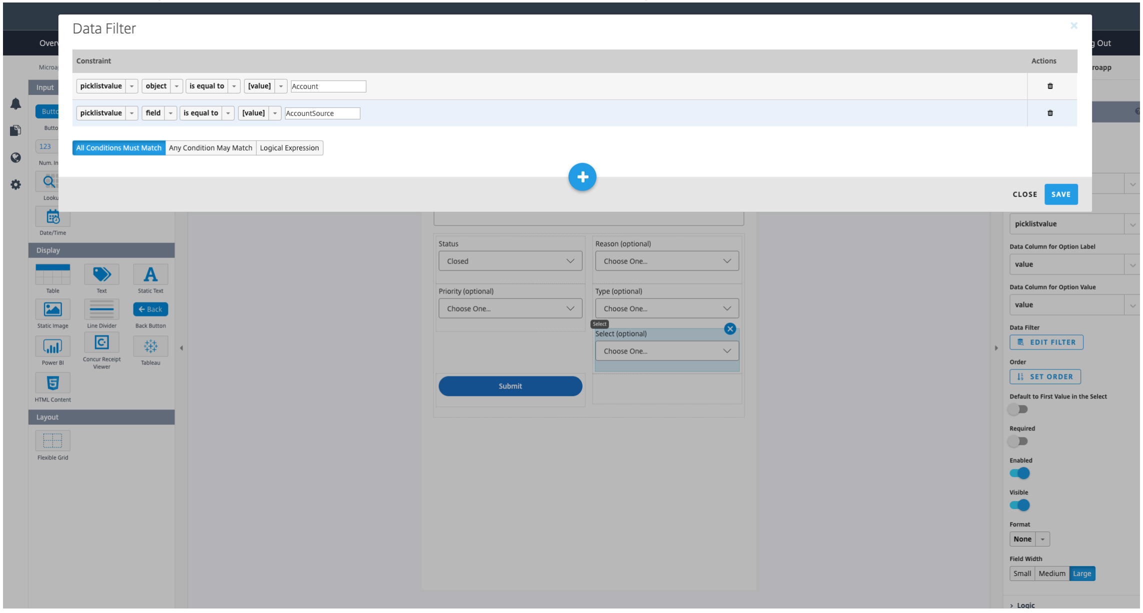
Add picklists’ values table

Due to the nature of the Salesforce schema, not all data is available as table entities. Use the picklist-value table to see every Salesforce object’s picklist and all of its options.

1. Open a Salesforce microapp and navigate to the page builder.
2. Select and drag a **Select** component to the field.
3. Under **Select Properties**, clear the **Map to Data Column** toggle. This allows you to view all Salesforce objects.
4. Select **picklistvalue** from the **Data Table** menu.
5. Select **EDIT FILTER** to open the data filter.

6. Select the **object** and the **field** that you need data from. Select **All Conditions Must Match** and **SAVE** to close the filter.

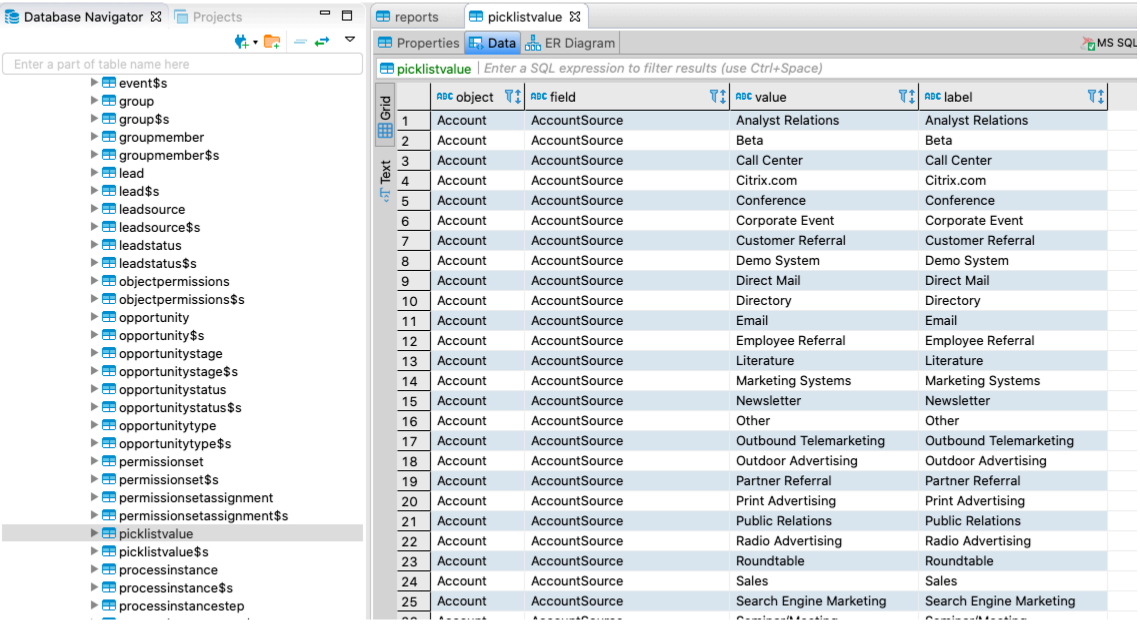
   The following screenshot shows an example of setting up the filter:



Data Filter

Constraint	Actions
picklistvalue ▾ object ▾ is equal to ▾ [value] ▾ Account	🗑
picklistvalue ▾ field ▾ is equal to ▾ [value] ▾ AccountSource	🗑

   This screenshot shows the database objects and fields that we are trying to access:



Database Navigator
Projects
Enter a part of table name here
reports
picklistvalue
Properties
Data
ER Diagram
MS SQL
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text

event$s
group
group$s
groupmember
groupmember$s
lead
lead$s
leadsource
leadsource$s
leadstatus
leadstatus$s
objectpermissions
objectpermissions$s
opportunity
opportunity$s
opportunitystage
opportunitystage$s
opportunitystatus
opportunitystatus$s
opportunitytype
opportunitytype$s
permissionset
permissionset$s
permissionsetassignment
permissionsetassignment$s
picklistvalue
picklistvalue$s
processinstance
processinstance$s
processinstancestep

picklistvalue

	object	field	value	label
1	Account	AccountSource	Analyst Relations	Analyst Relations
2	Account	AccountSource	Beta	Beta
3	Account	AccountSource	Call Center	Call Center
4	Account	AccountSource	Citrix.com	Citrix.com
5	Account	AccountSource	Conference	Conference
6	Account	AccountSource	Corporate Event	Corporate Event
7	Account	AccountSource	Customer Referral	Customer Referral
8	Account	AccountSource	Demo System	Demo System
9	Account	AccountSource	Direct Mail	Direct Mail
10	Account	AccountSource	Directory	Directory
11	Account	AccountSource	Email	Email
12	Account	AccountSource	Employee Referral	Employee Referral
13	Account	AccountSource	Literature	Literature
14	Account	AccountSource	Marketing Systems	Marketing Systems
15	Account	AccountSource	Newsletter	Newsletter
16	Account	AccountSource	Other	Other
17	Account	AccountSource	Outbound Telemarketing	Outbound Telemarketing
18	Account	AccountSource	Outdoor Advertising	Outdoor Advertising
19	Account	AccountSource	Partner Referral	Partner Referral
20	Account	AccountSource	Print Advertising	Print Advertising
21	Account	AccountSource	Public Relations	Public Relations
22	Account	AccountSource	Radio Advertising	Radio Advertising
23	Account	AccountSource	Roundtable	Roundtable
24	Account	AccountSource	Sales	Sales
25	Account	AccountSource	Search Engine Marketing	Search Engine Marketing

7. Finish setting up this component. For more information, see Page builder components.

Integrate ServiceNow

June 15, 2021

Integrate with ServiceNow to submit and monitor requests, and take action from any device, intranet, or messenger using Citrix Workspace.

> **Note**
>
> We provide two ServiceNow integration templates for your use. We recommend using the newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure. The set-up process for each integration is identical. For full details of the microapps available in each integration, see Use ServiceNow microapps.

Use the following process to enable the ServiceNow Integration. Ensure you meet the prerequisites, enable API access, and assign a role to the dedicated user. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

This integration enables you to:

- create a task in Workspace Experience. The solution adds an "opened_by" parameter to the API request based on the currently logged-in user. If you explicitly define the "opened_by" parameter in the service action parameters settings, it replaces the default value
- approve requests within the microapp. The solution adds the sentence Approval state set by the user_name to the comment field to identify who performed the approval
- create a new Service Catalog Request from the microapp. The solution adds the "requested_for" parameter to the API request based on the currently logged-in user

For a comprehensive list of out-of-the-box ServiceNow microapps, see Use ServiceNow microapps.

Review prerequisites

These prerequisites assume you administer the ServiceNow instance of your organization to set up the integration.

Workspace users need proper roles assigned to complete service actions. Proper roles depend on your ServiceNow configuration.

You must have these details to add the integration in Citrix Workspace Microapps:

- **Base URL**: This is your instance URL. You must enter your instance **Base URL** or simply replace `{ cloud-id }` in the example with your customer ID.
- **Authorization URL**: Replace `{ customer-id }` in the example with your customer ID, `https://{ customer-id } .service-now.com/oauth_auth.do` This is the authorization server URL provided when setting up the target application integration.
- **Token URL** Replace `{ customer-id }` in the example with your customer ID: `https://{ customer-id } .service-now.com/oauth_token.do`. This is the URL of the access authorization token.
- **Username**: This and Password are the credentials of the service account with access to the full table structure and all tables in ServiceNow.

- **Password**: This and Username are the credentials of the service account with access to the full table structure and all tables in ServiceNow.
- **Client ID**: You collect the Client ID by registering the OAuth client in your ServiceNow account. The Client ID and the Client Secret are the same for both Service authentication and Service action authentication.
- **Client Secret**: You collect the Client Secret by registering the OAuth client in your ServiceNow account. The Client ID and the Client Secret are the same for both Service authentication and Service action authentication.

> **Note**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

- Obtain a new oauth2 client_id and client_secret and define the scope of the client's application.

- Configure Citrix Gateway to support single sign-on for ServiceNow so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in ServiceNow Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

New HTTP integration privileges

Your ServiceNow admin account must have read access to all tables that we are fetching in the integration. See the list below:

- change_request
- incident
- problem
- sc_cat_item
- sys_user
- task
- cmn_location
- core_company
- sc_req_item
- sc_request
- sys_journal_field
- sys_user_delegate
- sys_user_group
- sys_user_has_role
- sys_user_role
- sys_user_role_contains

- sysapproval_approver
- sys_choice
- sc_item_option_mtom

> **Important**
>
> The ServiceNow admin account that administers the HTTP integration must have the timezone set to GMT. This is required to correct time handling in Workspace and in incremental data synchronization. If you see any time mismatch, first check these settings to resolve the issue.

ServiceNow roles

We recommend the following ServiceNow roles:

- approval_admin
- itil
- personalize_choices
- snc_read_only

Legacy integration privileges

This ServiceNow admin account must have full data access privileges. If you choose to use a separate ServiceNow account for the Microapps integration, you need to manually add read permissions on restricted tables, such as like sys_journal_field. Specifically, the administrator needs access to the following tables as they include information about the data structure of ServiceNow:

- sys_db_object
- sys_dictionary
- sys_choice

Enable API access for required tables

Most of the ServiceNow tables are enabled for access via web services by default. To confirm if a table you want to synchronize with Workspace is accessible via web services:

1. Log in to ServiceNow.
2. Select **System Definition > Tables**.
3. Select the **Information** icon next to the table name that you want to confirm. Select **Open record**. Select the **Application Access** tab and ensure that the "Allow access to this table via web services" check box is enabled.
4. Select the check box if necessary, and click **Update** to save your settings.

Add Callback URLs

Add a custom URL to your instance configuration to grant access to private data and enable OAuth authenticated user actions.

> **Note**
>
> This section of the URL `{ yourmicroappserverurl }` is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region(us/eu/ap-s)} .iws .cloud.com`.

1. Log in to ServiceNow as an admin.
2. Navigate to **System OAuth > Application Registry**, and select **New**.
3. Select **Create an OAuth API endpoint for external clients**.
4. Enter the following authorized redirect URLs for this integration in the **Redirect URL** field separated by a comma:

    - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
    - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`

    Ensure that **PKCE required** is not selected.
5. Click Submit.

Filter queries

Most ServiceNow entities support filtering. The sysparm_query URL parameter of the Table API GET method allows filtering. Choose predefined queries or write your own custom queries. For more information, consult the ServiceNow REST API reference and product documentation.

> **Note**
>
> If the query or any part of it is invalid, then the invalid part is ignored, as specified in the ServiceNow documentation.

Examples:

```
// Only Active objects:
active=true

// Updated in the last 2 days:
sys_updated_onONLast%20day@javascript:gs.daysAgoStart(1)@javascript:gs.daysAgoEnd(0)

```

```
// Updated in the last 3 hours:
sys_updated_onONLast%20hour@javascript:gs.hoursAgoStart(2)@javascript:gs.hoursAgoEnd(0)

// Updated in the last 4 months:
sys_updated_onONLast%20month@javascript:gs.monthsAgoStart(3)@javascript:gs.monthsAgoEnd(0)
```

Add the integration to Citrix Workspace Microapps

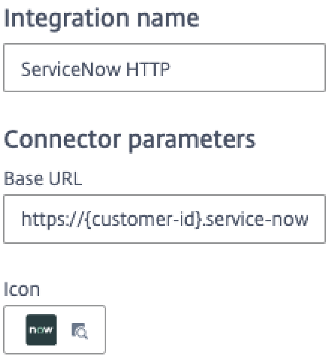
Add the ServiceNow integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace. We provide two ServiceNow integration templates for your use. We recommend using the newer HTTP integration for most use-cases.

Add the ServiceNow HTTP integration

Follow these steps to set up the ServiceNow HTTP integration. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the ServiceNow tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL** or simply replace `{ customer-id }` in the example with your customer ID.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default ServiceNow icon.



Integration name

ServiceNow HTTP

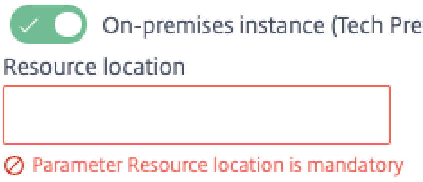
Connector parameters

Base URL

https://{customer-id}.service-now

Icon

- Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.



On-premises instance (Tech Preview)

Resource location

Parameter Resource location is mandatory

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

   a) Select **Resource Owner Password** from the **Grant type** menu. Provide the correct credentials to authorize resource server provision of an access token.

   b) Select **Request body** from the **Token authorization** menu.

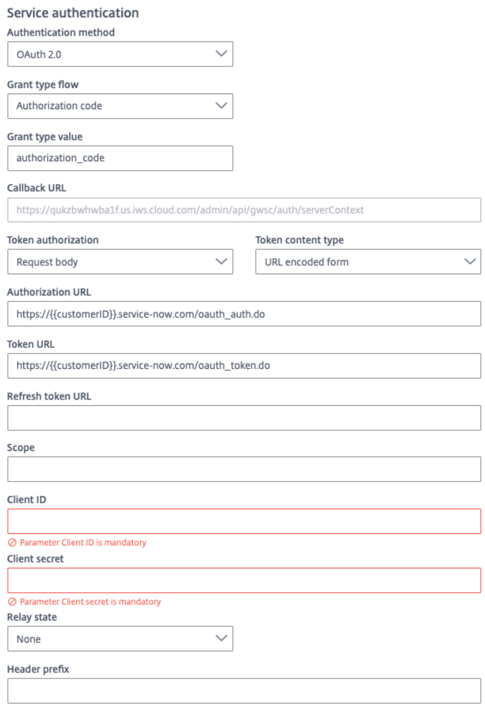
   c) Enter your **Token URL** or simply replace { `customer-id` } in the example with your customer ID: `https://{ customer-id } .service-now.com/oauth_token.do`. This is the URL of the access authorization token.

   d) Enter your **Username** and **Password**. These are the credentials of the service account with access to the full table structure and all tables in ServiceNow.

   e) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server.

   f) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

   g) Enter your **Header prefix**. (optional) Enter the header prefix if your bearer prefix is different from the default header.



Service authentication

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://qukzbwhwba1f.us.iws.cloud.com/admin/api/gwsc/auth/serverContext

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://{{customerID}}.service-now.com/oauth_auth.do

Token URL: https://{{customerID}}.service-now.com/oauth_token.do

Refresh token URL:

Scope:

Client ID:

⊘ Parameter Client ID is mandatory

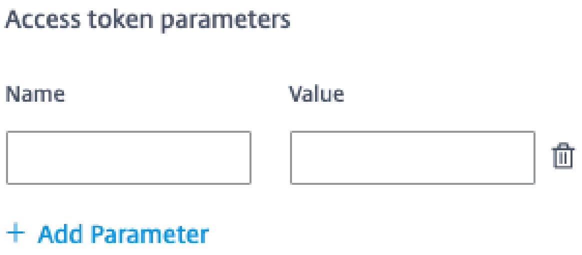
Client secret:

⊘ Parameter Client secret is mandatory

Relay state: None

Header prefix:

h) If you selected **OAuth 2.0** authentication method, you can select **+ Add Parameter** to include **Access token parameters**. Access token parameters define the access token parameters as required by the target application authorization server if necessary.



Access token parameters

Name:

Value:

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.

   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.

   c) Select **Request body** from the **Token authorization** menu.

   d) Enter your **Authorization URL** or simply replace `{ customer-id }` in the example with your customer ID, `https://{ customer-id } .service-now.com/oauth_auth.`

do This is the authorization server URL provided when setting up the target application integration.

e) Enter your **Token URL** or simply replace `{ customer-id }` in the example with your customer ID: `https://{ customer-id } .service-now.com/oauth_token.do`. This is the URL of the access authorization token.

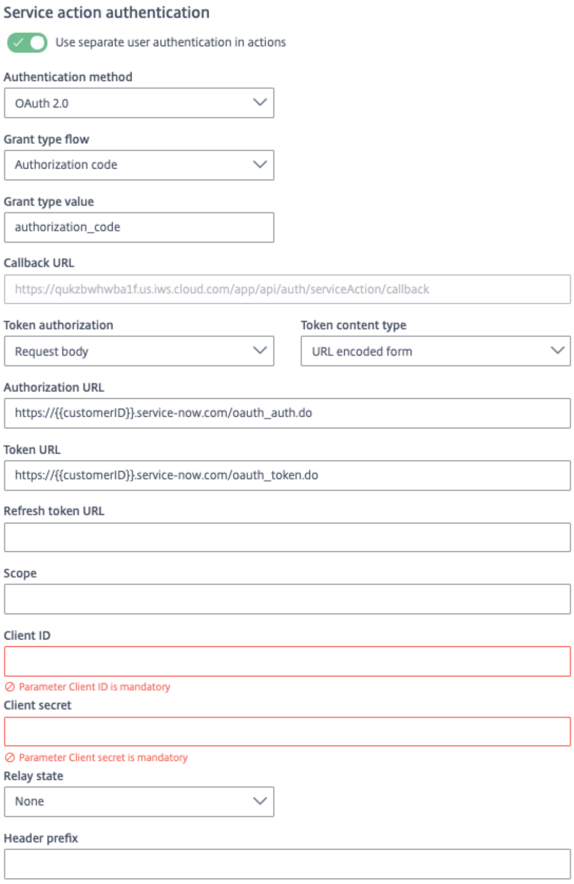
f) (Optional) Enter your **Scope** to define the scope of the access request. This string is defined by the authorization server when setting up your target integration application.

g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your ServiceNow account. You need to add the **Callback URL** you see on the integration configuration page.

h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.

i) (Optional) Enter your **Header prefix** if your bearer prefix is different from the default header.

j) If you selected **OAuth 2.0** authentication method, you can select **+ Add Parameter** to include **Access token parameters**. Access token parameters define the access token parameters as required by the target application authorization server if necessary.



Service action authentication

Use separate user authentication in actions

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://qukzbwhwba1f.us.iws.cloud.com/app/api/auth/serviceAction/callback

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://{{customerID}}.service-now.com/oauth_auth.do

Token URL: https://{{customerID}}.service-now.com/oauth_token.do

Refresh token URL:

Scope:

Client ID:

⊘ Parameter Client ID is mandatory

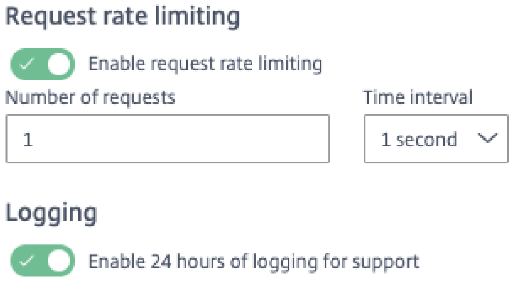
Client secret:

⊘ Parameter Client secret is mandatory

Relay state: None

Header prefix:

7. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

Enable request rate limiting

Number of requests: 1

Time interval: 1 second

Logging

Enable 24 hours of logging for support

9. Select **Save**.

Add the legacy integration

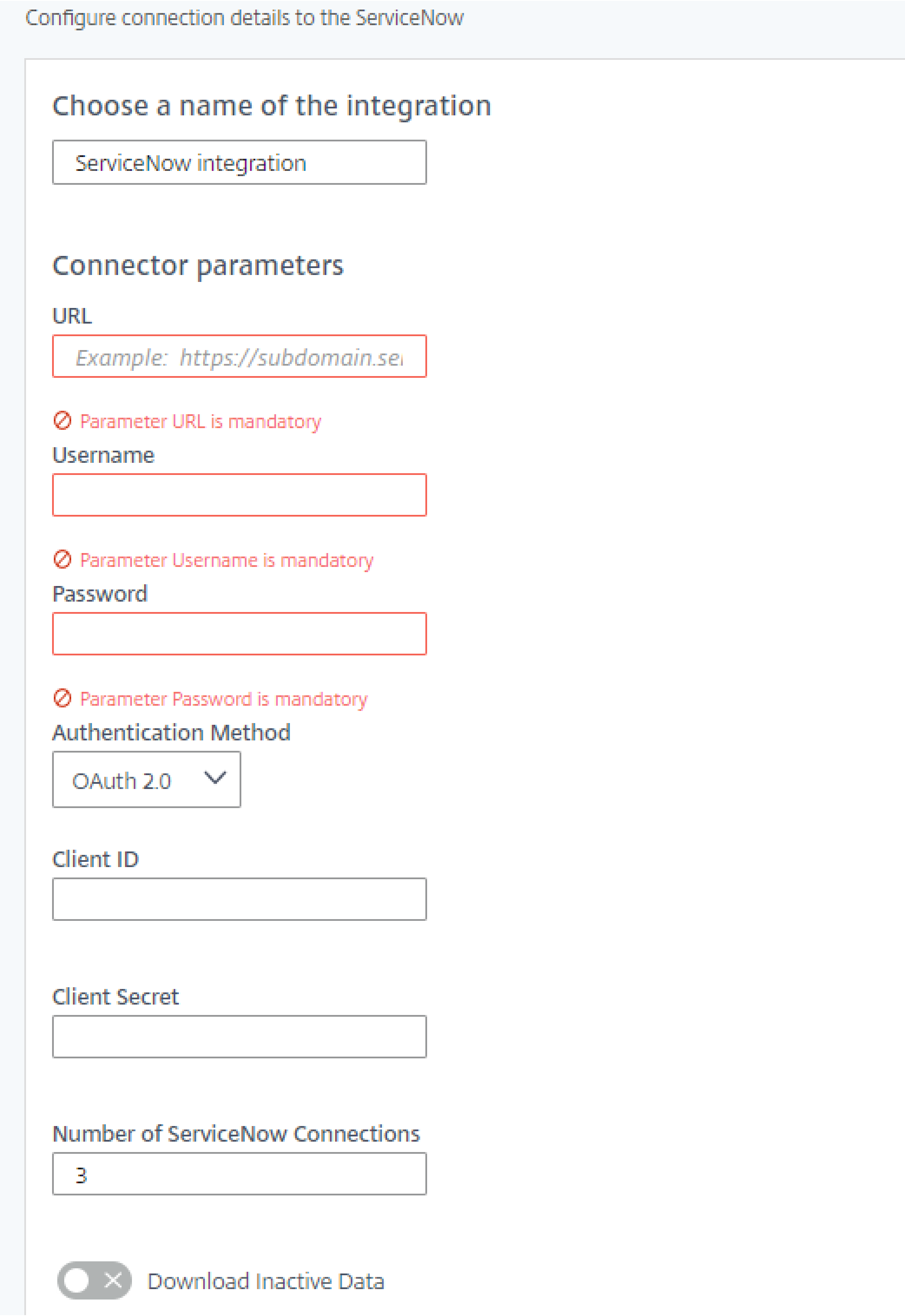
Follow these instructions to set up the legacy java-based ServiceNow integration.

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the ServiceNow tile.
4. Enter a name for the integration.



Configure connection details to the ServiceNow

Choose a name of the integration

ServiceNow integration

Connector parameters

URL

Example: https://subdomain.se

⊘ Parameter URL is mandatory

Username

⊘ Parameter Username is mandatory

Password

⊘ Parameter Password is mandatory

Authentication Method

OAuth 2.0

Client ID

Client Secret

Number of ServiceNow Connections

3

Download Inactive Data

5. Enter the **Connector parameters** that you collected as prerequisites.

    - Enter your **URL**.
    - Enter the **Username** and **Password**.

- Select an **Authentication Method**. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access.
- For Oauth 2.0, enter the **OAuth Client ID** and **OAuth Client Secret** that you collected in the prerequisites procedure.
- Enter a quantity for **Number of ServiceNow Connections**. This value determines the number of strings the data sync initiates.

  > **Note:**
  >
  > The default number of connections is three. Opening more connections reduces the time for data synchronization, but increases the load on the Microapps server and can influence its performance. If you require, we recommend no more than 10.

- Select the radio button to **Download Inactive Data** if you want to have a list of closed requests, for example, or other data that is set to `active = false`.

6. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

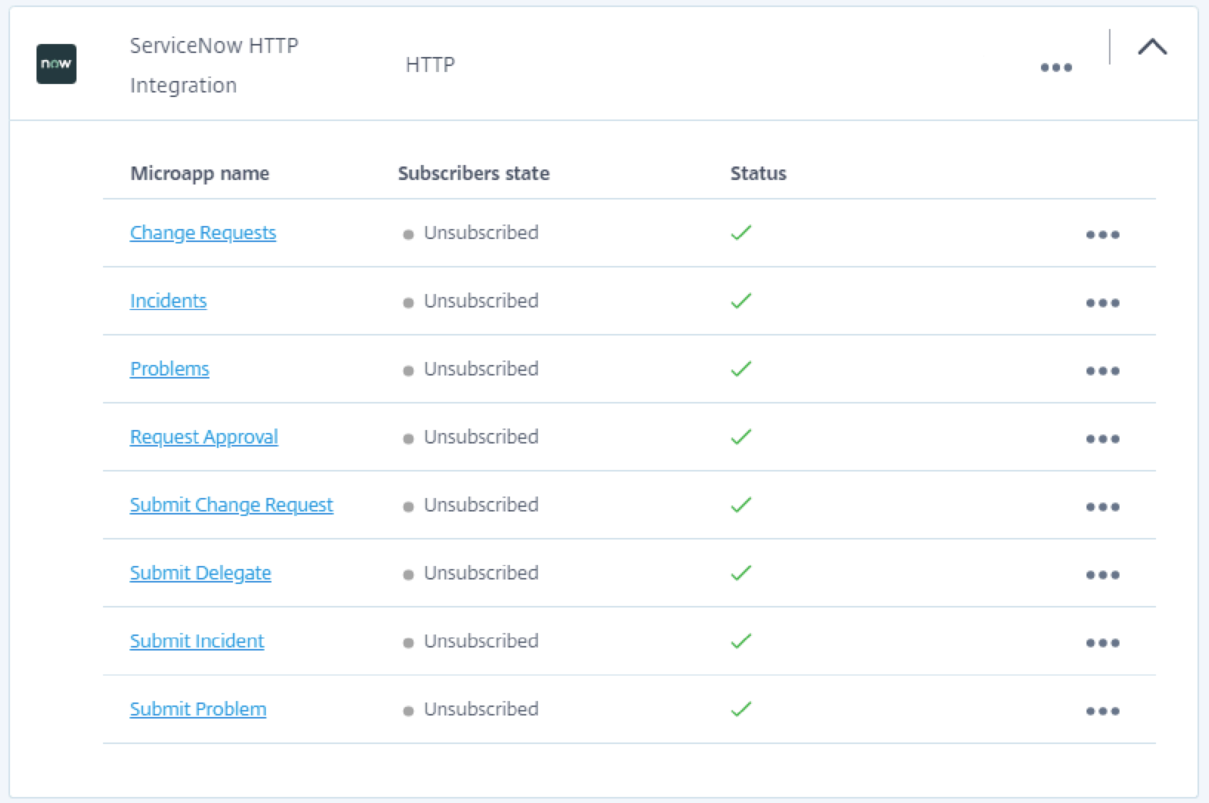
For more details of API endpoints and table entities, see ServiceNow HTTP connector specifications or ServiceNow connector specifications.

Use ServiceNow microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note**
>
> We provide two ServiceNow integration templates for your use. We recommend using the newer HTTP integration for most use-cases over the older java-based integration. The microapps that they contain differ slightly.

HTTP ServiceNow microapps



ServiceNow HTTP Integration HTTP

Microapp name	Subscribers state	Status
Change Requests	● Unsubscribed	✓
Incidents	● Unsubscribed	✓
Problems	● Unsubscribed	✓
Request Approval	● Unsubscribed	✓
Submit Change Request	● Unsubscribed	✓
Submit Delegate	● Unsubscribed	✓
Submit Incident	● Unsubscribed	✓
Submit Problem	● Unsubscribed	✓

Our HTTP ServiceNow integration comes with the following preconfigured out-of-the-box microapps:

Change Requests: Search for change requests, view their details, add comments, and update them.

Notification or Page	Use-case workflows
Change Request Assigned notification	When an existing change request is assigned to a user, they receive a notification.
Change Request Assignee Change (opened by) notification	When the assignee for a change request is changed, the user for whom the request was created receives a notification.
Change Request Assignee Change (requested by) notification	When the assignee for a change request is changed, the user who made the request receives a notification.
Change Request State Change (assigned to) notification	When the state of a change request is modified, the user who the request is assigned to receives a notification.

Notification or Page	Use-case workflows
Change Request State Change (opened by) notification	When the state of a change request is modified, the user who opened the request receives a notification.
Change Request State Change (requested by) notification	When the state of a change request is modified, the user for whom the request was created receives a notification.
New Change Request Assigned notification	When a new change request is assigned to a user, they receive a notification.
Change Request Detail page	Provides a read only view of a change request with details.
Comment Change Request page	Provides a form for commenting on a change request.
My Open Change Requests page	Allows users to search for open change requests that are assigned to them, requested by them, or opened by them.
Update Change Request page	Provides a form for updating a change request.

Incidents: Search incidents, view their details, add comments, and update them.

Notification or Page	Use-case workflows
Incident Assigned notification	When an existing incident is assigned to a user, they receive a notification.
Incident Assignee Change (caller) notification	When the assignee for an incident is changed, the user who reported the incident receives a notification.
Incident Assignee Change (opened by) notification	When the assignee for an incident is changed, the user who opened the incident receives a notification.
Incident State Change (assigned to) notification	When the state of an incident is modified, the user who the incident is assigned to receives a notification.
Incident State Change (caller) notification	When the state of an incident is modified, the user who reported the incident receives a notification.

Notification or Page	Use-case workflows
Incident State Change (opened by) notification	When the state of an incident is modified, the user who opened the incident receives a notification.
New Incident Assigned notification	When a new incident is assigned to a user, they receive a notification.
Comment Incident Form page	Provides a form for commenting on an incident.
Incident Detail page	Provides a read only view of an incident with details.
My Open Incidents page	Allows users to search for open incidents that are assigned to them, requested by them, or reported by them.
Update Incident page	Provides a form for updating an incident.

Problems: Search for problems, view their details, add comments, and update them.

Notification or Page	Use-case workflows
New Problem Assigned notification	When a new problem is assigned to a user, they receive a notification.
Problem Assigned notification	When an existing problem is assigned to a user, they receive a notification.
Problem Assignee Change (opened by) notification	When the assignee for a problem is changed, the user who opened the problem receives a notification.
Problem State Change (assigned to) notification	When the state of a problem is modified, the user who the problem is assigned to receives a notification.
Problem State Change (opened by) notification	When the state of a problem is modified, the user who opened the problem receives a notification.
Comment Problem page	Provides a form for commenting on a problem.
My Open Problems page	Allows users to search for open problems that are assigned to them or opened by them.

Notification or Page	Use-case workflows
Problem Detail page	Provides a read only view of a problem with details.
Update Problem page	Provides a form for updating a problem.

Request Approval: Search and view pending approvals, and approve or reject them.

Notification or Page	Use-case workflows
New Approve Request (Requested Item) notification	When an approval for a request or change request is assigned to a user, they receive an actionable notification that they can approve or reject.
New Approve Request (Problem) notification	When an approval for a problem is assigned to a user, they receive an actionable notification that they can approve or reject.
Pending Request Approval page	Allows users to search for pending approvals that are assigned to them.
Request Approval Detail page	Provides an actionable view of a pending approval with details that they can approve or reject.

Submit Change Request: Select items and submit a new change request.

Notification or Page	Use-case workflows
Submit Change Request page	Provides a form for submitting a change request.

Submit Delegate: Submit a new delegate.

Notification or Page	Use-case workflows
Submit Delegate page	Provides a form for submitting a new delegate.

Submit Incident: Submit a new incident.

Notification or Page	Use-case workflows
Submit Incident page	Provides a form for submitting a new incident.

Submit Problem: Submit a new problem.

Notification or Page	Use-case workflows
Submit Problem page	Provides a form for submitting a new problem.

Java-based ServiceNow microapps

Our java-based ServiceNow integration comes with the following preconfigured out-of-the-box microapps:

Approvals: Search and view pending approvals, and approve or reject them.

Notification or Page	Use-case workflows
New Approve Request notification	When a new request for approval is assigned to a user, they receive an actionable notification that they can approve or reject.
Approval Request Detail page	Provides an actionable view of a pending approval with details that they can approve or reject.
Pending Requests page	Allows users to search for pending approvals that are assigned to them.

Change Requests: Search for change requests, view their details, add comments, and update them.

Notification or Page	Use-case workflows
Change Request Assigned notification	When an existing change request is assigned to a user, they receive a notification.
Change Request Assignee Change notification	When the assignee for a change request is changed, the user who opened the request receives a notification.

Notification or Page	Use-case workflows
Change Request State Change notification	When the state of a change request is modified, the user who opened the request receives a notification.
New Change Request Assigned notification	When a new change request is assigned to a user, they receive a notification.
Change Request Detail page	Provides a read only view of a change request with details.
Comment Change Request Form page	Provides a form for commenting on a change request.
My Open Change Requests page	Allows users to search for open change requests that are assigned to them.
Update Change Request Form page	Provides a form for updating a change request.

Incidents: Search incidents, view their details, add comments, and update them.

Notification or Page	Use-case workflows
Incident Assigned notification	When an existing incident is assigned to a user, they receive a notification.
Incident Assignee Change notification	When the assignee for an incident is changed, the user who opened the incident receives a notification.
Incident State Change notification	When the state of an incident is modified, the user who opened the incident receives a notification.
New Incident Assigned notification	When a new incident is assigned to a user, they receive a notification.
Comment Incident Form page	Provides a form for commenting on an incident.
Incident Detail page	Provides a read only view of an incident with details.
My Open Incidents page	Allows users to search for open incidents that are assigned to them.
Update Incident Form page	Provides a form for updating an incident.

Problems: Search for problems, view their details, add comments, and update them.

Notification or Page	Use-case workflows
New Problem Assigned notification	When a new problem is assigned to a user, they receive a notification.
Problem Assigned notification	When the assignee of a problem is changed, the assignee receives a notification.
Problem Assignee Change notification	When the assignee for a problem is changed, the user who opened the problem receives a notification.
Problem State Change notification	When the state of a problem is modified, the user who opened the problem receives a notification.
Comment Problem Form page	Provides a form for commenting on a problem.
My Open Problems page	Allows users to search for open problems that are assigned to them.
Problem Detail page	Provides a read only view of a problem with details.
Update Problem Form page	Provides a form for updating a problem.

Submit Catalog Request: Select items and submit a new catalog request.

Notification or Page	Use-case workflows
Select Item page	Allows users to search the catalog and select available items.
Submit Catalog Request page	Provides a form for submitting a catalog request.

Submit Change Request: Submit a new change request.

Notification or Page	Use-case workflows
Submit Change Request page	Provides a form for submitting a change request.

Submit Delegate: Submit a new delegate.

Notification or Page	Use-case workflows
Submit Delegate page	Provides a form for submitting a new delegate.

Submit Incident: Submit a new incident.

Notification or Page	Use-case workflows
Submit Incident page	Provides a form for submitting a new incident.

Submit Problem: Submit a new problem.

Notification or Page	Use-case workflows
Submit Problem page	Provides a form for submitting a new problem.

Integrate Slack

April 20, 2021

Deploy the Slack integration to provide additional monitoring capabilities for critical channels that may not be traffic intensive but require the attention of its members. In order to tailor the channels available to a specific group or department, use multiple integrations.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapp for Slack, see Use Slack.

Review prerequisites

These are the values that you enter in Citrix Workspace Microapps:

- **Base URL**: `https://slack.com/api`
- **Authorization URL**: `https://slack.com/oauth/authorize`
- **Token URL**: `https://slack.com/api/oauth.access`

- **Channel ID**: You collect this when you create a new favorites channel in Slack. You need this to modify endpoints and service actions. See Create a favorites channel and collect Channel ID.
- **OAuth Access Token**: You enter this as the **Token** value when setting up the integration template. You collect this token, the Client ID, and the Client Secret when you Create Bot.
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server.
- **Client Secret**: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Create a new service account

The integration requires regular access to your Slack instance. We recommend creating a dedicated user account with full administrator privileges. Sign up here: https://slack.com/get-started#/create.

Enable APIs

The number of API requests that can be made to specific resources is limited. We therefore recommend the following:

- Slack API limitation per link: https://api.slack.com/docs/rate-limits#overview
- Slack API tiers: https://api.slack.com/docs/rate-limits
- Slack API plans: only one offered

Create the Bot

Bots are Slack apps that interact with users, with the ability to post, receive, and respond to messages from users. Create the Slack app and select scopes to add to the app.

1. Navigate to the Slack Management UI and create an app if you haven't already created one: https://api.slack.com/apps
2. Enter an **App Name**, and select the **Development Slack Workspace** where the app will be installed.
3. Select **Create App**.
4. Under **Basic Information**, copy the following **App Credentials** information:
   - **Client ID**
   - **Client Secret**

5. Navigate to **OAuth & Permissions** under **Features** on the left sidebar.
6. Under **Scopes/Bot Token Scopes**, select **Add an OAuth Scope** tile under the **Scopes** section, Ensure that you add scopes to the Bot Token, not your User Token. Add these scopes: `channels:history channels:join channels:read groups:history groups:read mpim:history mpim:read team:read users.profile:read users:read users:read.email`
7. Under **Redirect URLs**, for each of the following callbacks select **Add New Redirect URL**, enter the value, and select **Save URLs** when you're finished.
   - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
   - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`

Install the Slack app

Install the app to your Slack workspace to test your app and generate the tokens needed to interact with the Slack API.

1. Navigate to **Install App** under **Settings** on the left sidebar.
2. Select **Install App to Workspace**, ensure the app is permitted to **Perform acitons in channels & conversations**, and select **Allow**.
3. Copy the **Bot User OAuth Access Token**.

Create a favorites channel and collect Channel ID

Create a channel for tracking favorites. You need to collect the Channel ID from the URL of this channel for modifying the integration.

> **Note:**
>
> If using multiple Slack integrations, use a separate/dedicated favorite channel for each integration.

1. Create a new channel in Slack named *favorites*.
2. Select **Add all members of {SlackWorkspaceName}**
3. Copy the channel link. Save the ID at the end of the URL. This is your Channel ID you need to modify endpoints and service actions. See Modify endpoints and service actions.

Add the Bot to focus and favorite channels

Now add the Bot (Slack app) to any channels that you want to expose to the team that subscribes to the app and the favorite channel created above. Consider the following:

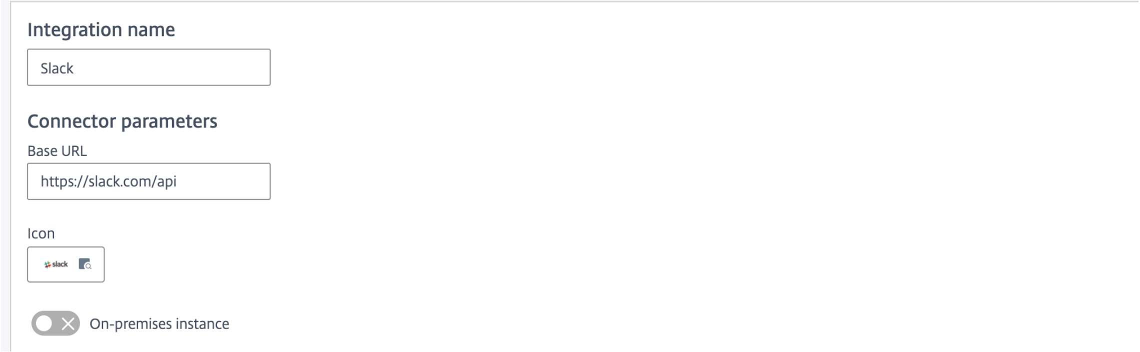
- Do not add a Bot to a noisy channel. Bots should be added to channels that are used for infrequent, time-critical communications within a select group, such as urgent sales issues for the Sales Group or IT security for General Employees.
- Multiple integrations can point to the same Slack app.
- Keep the channel list focused on a specific group.
- We only recommend adding the Bot to a public channel. Adding bots to a private channel may allow others to view membership of the private channel.

Add the integration to Citrix Workspace Microapps

Add the Slack integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions that are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration** and **Add a new integration from Citrix-provided templates**.
2. Choose the Slack tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**: `https://slack.com/api`
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

Slack

Connector parameters

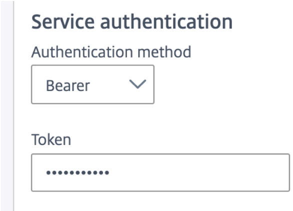
Base URL

https://slack.com/api

Icon

On-premises instance

5. Under **Service Authentication**, select **Bearer Token** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process.
6. Enter the **Token**. This value is the Bot User OAuth Access Token that you collect when you created the bot. See Create the Bot.



Service authentication

Authentication method

Bearer

Token

•••••••••••

7. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. This authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

    a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.

    b) Select **Authorization code** for **Grant type flow** menu.

    c) Enter **authorization_code** for **Grant type value**.

    Callback URL is prefilled.

    d) Select **Request Body** from the **Token Authorization** menu.

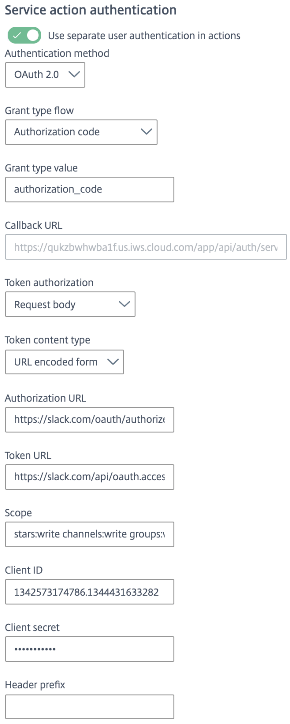
    e) Select **URL encoded form** from the **Token content type** menu.

    The **Authorization URL** is prefilled: `https://slack.com/oauth/authorize`. The **Token URL** is prefilled: `https://slack.com/api/oauth.access`.

    f) Ensure the following is entered for Scope: `channels:history channels:join channels:read groups:history groups:read mpim:history mpim:read team:read users.profile:read users:read users:read.email`

    g) Enter the **Client ID** that you obtained in Create the Bot.

    h) Enter the **Client Secret** that you obtained in Create the Bot.



Service action authentication

☑ Use separate user authentication in actions

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://qukzbwhwba1f.us.iws.cloud.com/app/api/auth/serv

Token authorization: Request body

Token content type: URL encoded form

Authorization URL: https://slack.com/oauth/authorize

Token URL: https://slack.com/api/oauth.acces

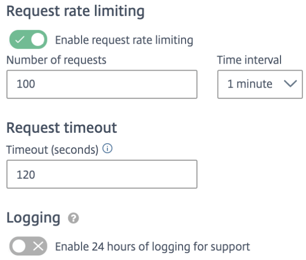
Scope: stars:write channels:write groups:\

Client ID: 1342573174786.1344431633282

Client secret: •••••••••••

Header prefix:

8. Enable the **Request Rate Limiting** toggle and enter *1* for **Number of requests** per second.
9. Enter *120* in the **Request timeout** field.
10. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
11. Select **Save**.



Request rate limiting

☑ Enable request rate limiting

Number of requests: 100

Time interval: 1 minute

Request timeout

Timeout (seconds): 120

Logging

☒ Enable 24 hours of logging for support

The **Microapp Integrations** page opens with your added integration and its microapps. Now modify the integration by adding the `channel` value as described in the next procedure.

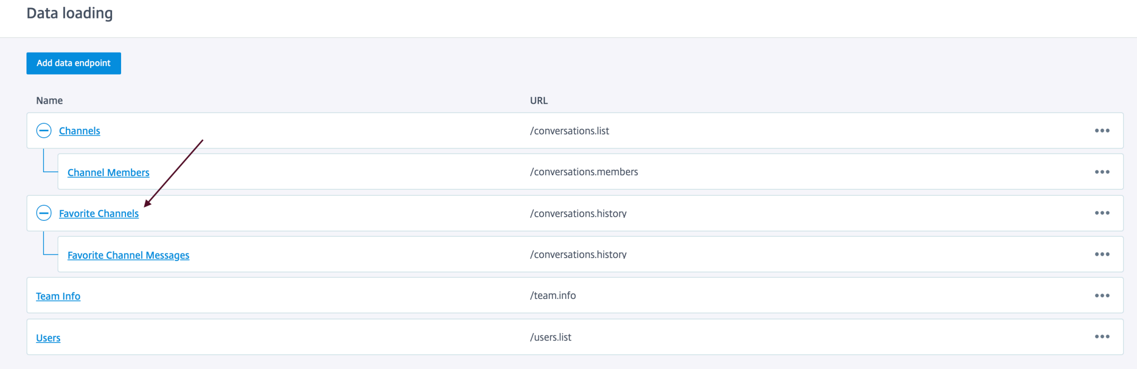
Modify endpoints and service actions

To complete this set up, you need to add the `channel` value with your channel ID collect in Create a favorites channel and collect Channel ID. Modify the **Favorite Channels** endpoint and both the **Favorite channel** and **Unfavorite** service actions.

Replace Data Loading endpoint

Manually add the `channel` value in the **Favorite Channels** endpoint with your Channel ID.

1. From the **Microapp Integrations** page, select the menu next to the Slack integration, and then **Edit**. The **Data Loading** screen opens. If you are in the configuration screen, select **Data Loading** from the left side navigation column.
2. Select the menu next to the **Favorite Channels** endpoint and then select **Edit**, or select the name of the endpoint: **Favorite Channels**.

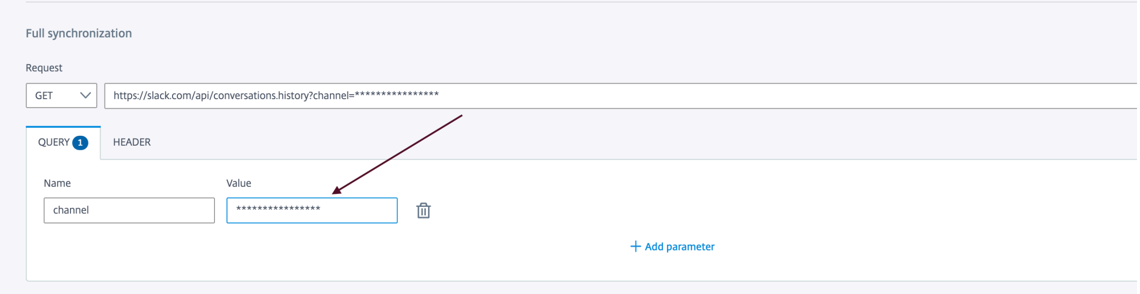


Data loading

Add data endpoint

Name	URL
Channels	/conversations.list
Channel Members	/conversations.members
Favorite Channels	/conversations.history
Favorite Channel Messages	/conversations.history
Team Info	/team.info
Users	/users.list

3. In the **Edit Data Endpoint** screen, under **Full synchronizations** enter the Channel ID in the value field for **channel**.
4. Select **Apply** at the bottom of the screen and confirm.



Full synchronization

Request

GET https://slack.com/api/conversations.history?channel=****************

QUERY 1 HEADER

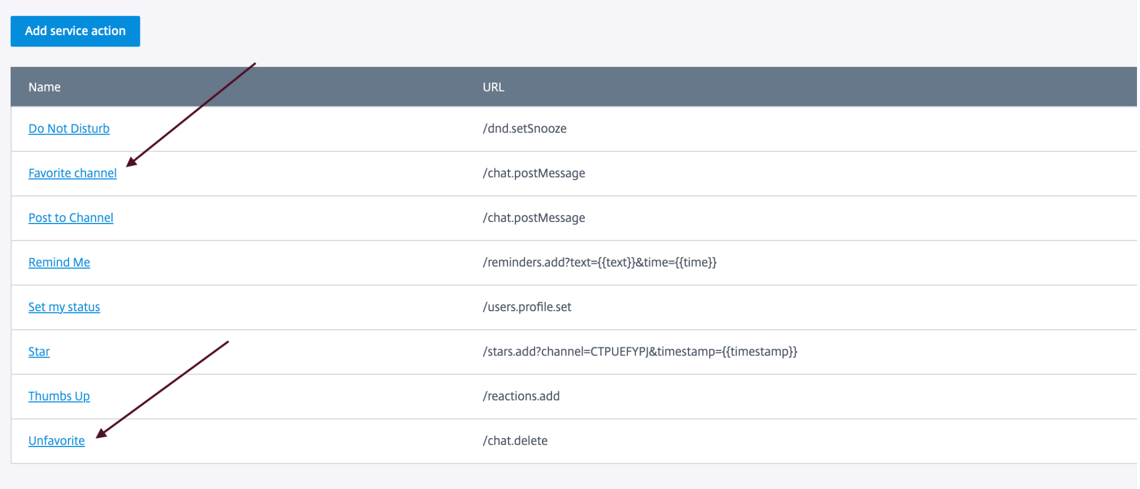
Name	Value
channel	****************

+ Add parameter

Replace Service Action variables

For the **Favorite channel** and **Unfavorite** service actions, you must manually add the `channel` value with your Channel ID twice for both service actions. Once under **Action execution** and once under **Post action data update (optional)**.

1. While editing the integration configuration, select **Service Actions** from the left side navigation column.
2. Select the menu next to one of the service actions that you need to edit and select **Edit**, or select the name of the service action that you need to edit. Let's start with the **Favorite channel**.

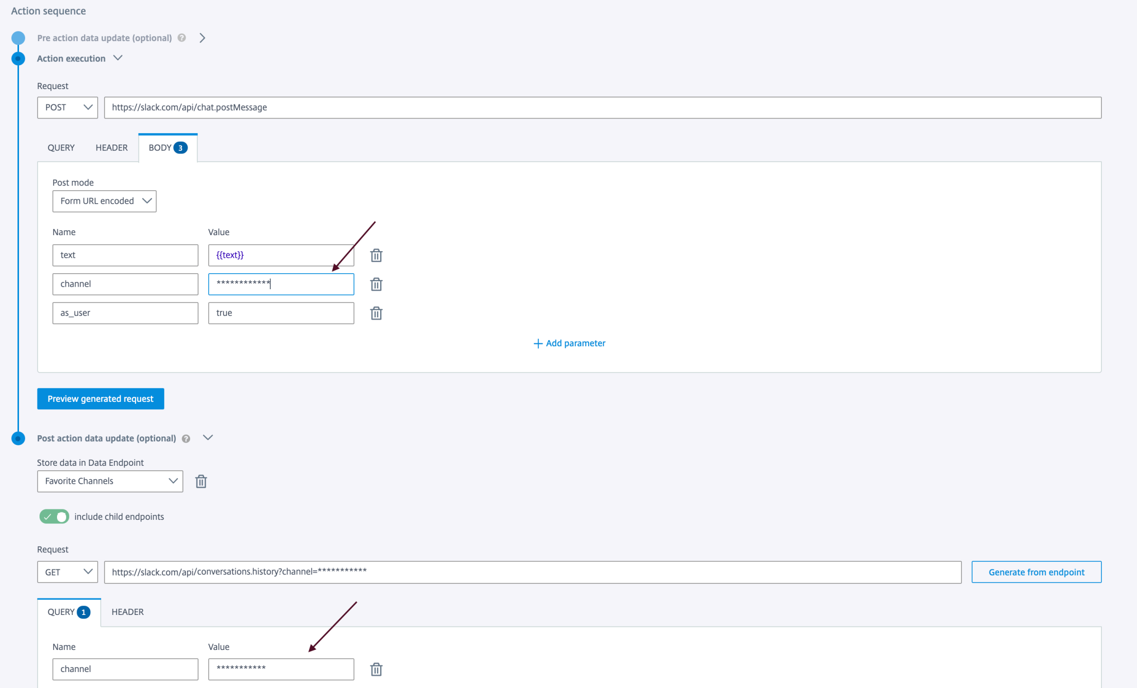


Name	URL
Do Not Disturb	/dnd.setSnooze
Favorite channel	/chat.postMessage
Post to Channel	/chat.postMessage
Remind Me	/reminders.add?text={{text}}&time={{time}}
Set my status	/users.profile.set
Star	/stars.add?channel=CTPUEFYPJ&timestamp={{timestamp}}
Thumbs Up	/reactions.add
Unfavorite	/chat.delete

3. In the **Edit Service Action** screen under **Action sequence** and then under **Action execution**, select **BODY**.

4. Enter the Channel ID in the value field for **channel**.

5. Under **Post action data update (optional)**, again enter the Channel ID in the value field for **channel**.



Request

POST https://slack.com/api/chat.postMessage

Post mode: Form URL encoded

Name	Value
text	{{text}}
channel	************
as_user	true

Store data in Data Endpoint: Favorite Channels

include child endpoints

Request

GET https://slack.com/api/conversations.history?channel=***********

Name	Value
channel	***********

6. Select **Save** to finish.

7. Now repeat this for the other service action: **Unfavorite**. Add the `channel` value with your Channel ID twice. Once under **Action execution** and once under **Post action data update (optional)**.

For more details of API endpoints and table entities, see Slack connector specifications.

Use Slack microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

My Favorite Channels: Receive notifications of activity in favorited channels.

Notification or Page	Use-case workflows
Channels List page	Provides a list of favorited channels, with an option to **Add Channels**.
Channel Detail page	View details of a favorited channel including past posts. Includes button options to **Post Message** and **Unfavorite Channel**.
Message Detail page	View message details from a favorited channel, with details of the sender. Includes button options to **Thumbs Up** and **View Replies In Slack**.
Post a Message page	Provides a form to compose and button to **Post** a message to a favorited channel.
Add Channels Detail page	View details of a channel including members. Includes a button option to **Add To Favorites**.
Add Channels List page	Provides a searchable list of channels, with an option to view details on the Channel Detail page.

Post to Slack: Post a message to the selected focused channel in slack.

Notification or Page	Use-case workflows
Post New Message page	Provides a form to compose and post a message.

Set My Slack Status: Set your slack status, create reminders, and enable **Do Not Disturb** for a set amount of time.

Notification or Page	Use-case workflows
Create a reminder page	Provides a form to set a reminder.

Notification or Page	Use-case workflows
Pause Notifications page	Provides a form to pause notifications for a set amount of time.
Set My Slack Status page	Provides a form to set a status, with options to **Set Reminder** and **Do Not Disturb**.

Integrate Smartsheet

May 27, 2021

Deploy the Smartsheet integration to manage sheets, discussions, update requests, and attachments.

For comprehensive details of the out-of-the-box microapps for Smartsheet, see Use Smartsheet microapps.

> **Note:**
>
> This Smartsheet integration template is released in **Citrix Labs** category. This allows the functionality to mature as a result of initial customer feedback. For Citrix Labs templates, there is no commitment to support and support is provided by the developer on a best-effort basis. Citrix Labs integration templates are shared for the purpose of testing/validation. We do not advise deploying them in production environments. Citrix Labs templates are listed in a separate section. We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

Review prerequisites

These prerequisites assume that the administrator will be a part of the SmartSheet integration set up of the organization. This Smartsheet admin account must have full read privileges for all users and sheets informations.

After you set up this integration with Smartsheet, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://api.smartsheet.com/`
- **Authorization URL**: `https://app.smartsheet.com/b/authorize`
- **Token URL**: `https://api.smartsheet.com/2.0/token`
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server. You collect this as **Application Key** when you configure the OAuth server.

- **Secret**: The client secret is a unique string issued when setting up the target application integration. You collect this as **Application Key** when you configure the OAuth server.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for Blackboard so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.

Create a service account

The integration requires regular access to your Smartsheet instance, so we recommend creating a dedicated user account. This account must have the following permissions for your Service Account: Full administrator privileges. You can view the permission/privileges using https://admin.smartsheet.com.

To create a service account, sign up here: https://app.smartsheet.com/b/signup. Ensure that the paid account is available to create a new service account.

In case of issues setting up the new service account, please connect to the respective sales support team or customer support team: https://www.smartsheet.com/contact/sales?fts=contact.

API access

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Recommended plan: Business
- Smartsheet API limitation form link: https://smartsheet-platform.github.io/api-docs/#rate-limiting

Smartsheet APIs are available in open source by default.

Configure OAuth server

Configure the OAuth server to read data through the Smartsheet integration.

1. Navigate to https://developers.smartsheet.com/register/ and enter your service account admin email.
2. Select **Register Developer Account**.
3. Check for an email from Smartsheet in the Service account admin inbox.

4. Select the link. You are navigated to https://app.smartsheet.com.
5. A dialog box will prompt you to create a new app. Select **Create New App** under the developer profile section.
6. Complete the required fields, including App name, Description, URL, and Contact/Support email.
7. Enter the following authorized redirect URLs for this app in the Redirect URL field: `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
8. Select the Publish app check box and select **Save**.
9. Copy and save the **ClientId** and **Secret** shown on the screen. You use this for **Service Authentication** while configuring the integration.

Configure OAuth client

Configure the OAuth client to write data back through the Smartsheet integration.

1. As in step 5 above, select **Create New App** under the developer profile section.
2. Complete the required fields, including App name, Description, URL, and Contact/Support email.
3. Enter the following authorized redirect URLs for this app in the Redirect URL field: `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serviceAction/callback`
4. Select the Publish app check box and select **Save**.
5. Copy and save the **ClientId** and **Secret** shown on the screen. You use this for **Service Action Authentication** while configuring the integration.
6. Select **Close**.

Add the integration to Citrix Workspace Microapps

Add the Smartsheet integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions that are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration** and **Add a new integration from Citrix-provided templates**.
2. Choose the SmartSheet tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**:
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization Header** from the **Token authorization** menu.
   c) The **Authorization URL** is prefilled: `https://app.smartsheet.com/b/authorize`
   d) The **Token URL** is prefilled: `https://api.smartsheet.com/2.0/token`
   e) Ensure the following is entered for Scope: *ADMIN_SHEETS ADMIN_USERS READ_SHEETS READ_USERS*
   f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this as **ClientId** when you configure the OAuth server.
   g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration. You collect this as **Secret** when you configure the OAuth server.
6. Under **Service Action Authentication**, enable the **Use Separate User Authentication** in Actions toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.
   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization Header** from the **Token authorization** menu.
   c) The **Authorization URL** is prefilled: `https://app.smartsheet.com/b/authorize`
   d) The **Token URL** is prefilled: `https://api.smartsheet.com/2.0/token`
   e) Ensure the following is entered for Scope: *ADMIN_SHEETS ADMIN_USERS SHARE_SHEETS WRITE_SHEETS CREATE_SHEETS ADMIN_WORKSPACES*
   f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this as **ClientID** when you configure the OAuth client.
   g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration. You collect this as **Secret** when you configure the OAuth client.
7. Enable the **Enable request rate limiting** toggle button. Enter *300* for **Number of requests** and *1 minute* for **Time interval**.
8. **Request timeout** is set to *120* by default.

9. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
10. Select **Save** to proceed.

You are now ready to set and run your first data synchronization.

Synchronization

Due to the API call limit, incremental synchronization is setup to retain only List org sheets, List org discussions, and Get all org sentupdaterequests. The remaining endpoints will be triggered as part of full synchronization.

We recommend setting the **Full Synchronization** interval as **Daily** and **Incremental Synchronization** interval as **Every 5 mins** to regularly refresh data from Smartsheet to the Microapps platform and receive timely notifications. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

> **Note:**
>
> The pagination limit is set to 100. Administrators can extend this limit based on APIs.

The default value for **Max pages to load variable** is set as shown below:

Endpoint Name	Value
List Org Sheets	50
List Groups	10
List Org Discussions	10
Get all org sentupdaterequests	10
List Sheets	50
Get Sheets	10
List Sheet Shares	10
List Groups	50

For more details of API endpoints and table entities, see Smartsheet connector specifications.

Use Smartsheet microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Access Sheets: View sheets, share a sheet to a licensed or non-licensed user or group, add a sheet as favorite, and allow users to view their individual sheet.

Notification or Page	Use-case workflows
View All Sheets page	Allows users to search for shareable and non-shareable sheets.
Shareable Sheet Detail page	Provides an actionable view of adding the sheet as a favorite, viewing the sheet, sharing the sheet with licensed/non-licensed users or groups.
Non-Shareable Sheet Detail page	Provides an actionable view of adding sheet as a favorite and viewing the sheet.
Shareable Group Sheet Detail page	Provides an actionable view of adding the sheet as a favorite, viewing the sheet, sharing the sheet with licensed/non-licensed users or groups.
Non-Shareable Group Sheet Detail page	Provides an actionable view of adding group sheet as a favorite and viewing the group sheet.

Create a Sheet: Create a new sheet with fields and options such as sheet name, enter column title, select column type, and select primary column.

Notification or Page	Use-case workflows
Create Sheet page	Provides a form to create a new sheet.

Discussion: Generate notifications to the discussion creator whenever there is a reply to theirs discussion thread.

Notification or Page	Use-case workflows
New comment added to your discussion notification	When a reply or comment is added to an existing discussion, the discussion creator receives a notification.
Discussion Detail page	Provides a form to reply to the discussion thread and view previous comments.

My Update Requests: View sent and received update requests with details such as sent to, sent by, subject and status. Additionally, when a user sends an update request to recipients, the recipients receive a notification. Once the update request is complete,d the sender receive a completed notification.

Notification or Page	Use-case workflows
Smartsheet Update Request Received notification	When a requester requests an update request, the recipient receives a notification.
Smartsheet Update Request Completed notification	When a recipient completes an update request, the requester receives a notification.
All Update Requests page	Allows users to search for sent and received update requests.
Sent Update Request Detail page	Provides an actionable view of sent update request with Delete update request and view sheet functionality.
Received Update Request Detail page	Provides a read only view of a received update request with view sheet functionality.
Completed Update Request Detail page	Provides a read only view of a completed update request with view sheet functionality.

Send Smartsheet as Attachment: Send Smartsheets as an attachment (PDF or Excel), with details such as To email, subject, and message.

Notification or Page	Use-case workflows
View All Sheets page	Allows users to search sheet they own.
Send as Email Detail page	Provides an actionable view to send a smartsheet as an attachment (PDF or Excel) with subject and/or message.

Share with Admin: Used by Non-Admin users to share their sheets to admins with view only access, and to unlock other features such as Access sheets, My Update request, Discussions, and receive the respective notification. Share the sheet with Admin to unlock additional Smartsheet actions and notifications in Workspace for you, including: update request actions/notifications, discussion notifications, and viewing your sheets.

> **Note:**
>
> In this microapp there is a page named **Share Sheet To Admin** which contains a Select component called **Admin Email** (this is not visible to the end user). This component is used to share user sheets with your organization's Admin account. This Admin account is the same Service Account that you have setup in the previous step. In case your organization has multiple Service/Admin accounts, please make sure to point this **Admin Email** Select component to the right account to ensure this microapp works correctly.

Notification or Page	Use-case workflows
View All Sheets page	Allows users to search through their sheets and share them with admin as needed.
Share Sheet to Admin Detail page	Provides an actionable view to share a sheet with admin with view-only access, and add a note to a sheet which is already shared.
Sheet Shared with Admin page	Provides user confirmation message when the sheet is shared with admin.

Start a Discussion: Start a discussion on sheet level.

Notification or Page	Use-case workflows
Start Discussion page	Provides an actionable view to initiate a discussion at a sheet level.

Integrate SocialChorus

April 20, 2021

Deploy the SocialChorus Integration to communicate important announcements from management and share the content, such as articles, links and notes, between employees through different channels. No images or media are displayed.

Users can view the past seven days of content that is posted in recommended channels. Using appropriate selections, users can view all featured content posted in various channels. Users can also view all content posted in their followed channels that have been posted in the past five days. All details relating to the content (such as title, summary, body, and published date) is shown on individual pages. Posted content can be viewed in a test instance with the click of a button.

A user assigned with the Program Manager role in SocialChorus marks channels as recommended or sets an article as featured using the SocialChorus Manage Channel UI. For example, these communication channels might be from senior management, or featured articles of interest to a user group or all users.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details about our SocialChorus microapps, see Use SocialChorus microapps.

Review prerequisites

You will need these artifacts to add the integration in Citrix Workspace Microapps:

- Base URL: `https://partner.socialchorus.com/`
- Token URL: `https://auth.socialchorus.com/oauth/token`
- Client ID: The Client ID is the string representing client registration information unique to the authorization server. Contact your Social Chorus account representative for the Client ID and Client Secret.
- Client Secret: The client secret is a unique string issued when setting up the target application integration.

This integration requires regular access to your SocialChorus instance, so we recommend creating a dedicated user account. This service account must have full administrator privileges and permissions.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Enable APIs

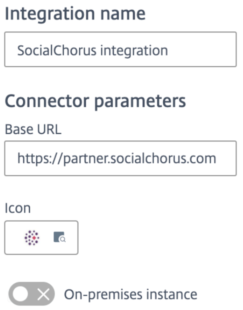
SocialChorus APIs are enabled via webservices for a paid account by default. This may require a separate agreement with the vendor. The number of API requests that can be made to specific resources is limited. We recommend reviewing SocialChorus API guidance: SocialChorus API Guidance.

Add the integration to Citrix Workspace Microapps

Add the SocialChorus integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the SocialChorus tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL:** `https://partner.socialchorus.com/`
    - Select an Icon for the integration from the Icon Library, or leave this as the default icon.

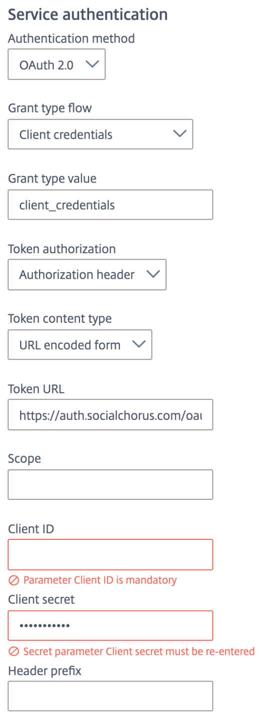


Integration name: SocialChorus integration

Connector parameters

Base URL: https://partner.socialchorus.com

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
    a) Select **Client Credentials** from the **Grant type flow** menu.
    b) **client_credentials** is entered for **Grant type value**.
    c) Select **Authorization header** from the **Token authorization** menu.
    d) Select **URL encoded form** from the **Token content type** menu.
    e) Enter the **Token URL**: `https://auth.socialchorus.com/oauth/token`
    f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configure the OAuth server.
    g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service authentication

Authentication method

OAuth 2.0

Grant type flow

Client credentials

Grant type value

client_credentials

Token authorization

Authorization header

Token content type

URL encoded form

Token URL

https://auth.socialchorus.com/oau

Scope

Client ID

Parameter Client ID is mandatory

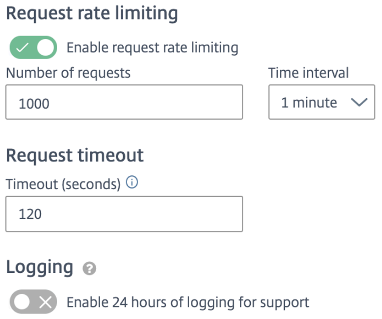
Client secret

•••••••••••

Secret parameter Client secret must be re-entered

Header prefix

6. Leave **Access token parameters** empty.
7. Enable the **Enable request rate limiting** toggle. Enter 1000 for **Number of requests** and 1 minute for **Time interval**.
8. In the **Request Timeout** field, enter *120*.
9. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
10. Select **Save** to proceed.



Request rate limiting

Enable request rate limiting

Number of requests

1000

Time interval

1 minute

Request timeout

Timeout (seconds)

120

Logging

Enable 24 hours of logging for support

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see SocialChorus connector specifications.

Use SocialChorus microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs. Our SocialChorus integration comes with the two following preconfigured out-of-the-box microapps. Both of these microapps retrieve content for the last 7 days:

Important Communications: Search and view important communications from recommended channels that are posted within the last 7 days.

Notification or Page	Use-case workflows
Change In Recommended Channel (Featured) notification	All subscribers receive a notification when there is a change in featured content tile, summary, body, or featured label under recommended channels. Notification expires 7 days from the created date.
Change In Recommended Channel (Non-Featured) notification	All subscribers receive a notification when there is a change in non-featured content tile, summary, or body under recommended channels. Notification expires 7 days from the created date.
New Recommended Article (Featured) notification	All subscribers receive a notification when a new featured article is posted in recommended type channels. Notification expires 7 days from the created date.
New Recommended Article (Non-Featured) notification	All subscribers receive a notification when a new non-featured article is posted in recommended type channels. Notification expires 7 days from the created date.
View Content page	Provides a complete list of articles posted in recommended channels within the last 7 days.
Content Detail page	Provides a form to view articles in detail with a **Read In The Blog** option to open the article in SocialChorus.

Latest Articles: Search and view content from subscribed channels that has been posted within the last 7 days.

Notification or Page	Use-case workflows
View Content page	Provides a complete list of articles posted in subscribed channels within the last 7 days.
Content Detail page	Provides a form to view articles in detail with a **Read In The Blog** option to open the article in SocialChorus.

Featured Content: Get recently featured communications from the subscribed channels with this microapp. Users can search, view images, and read content from Citrix Workspace.

Notification or Page	Use-case workflows
View Contents page	Provides a table with all the featured content where a user can view the content from all subscribed channels. Users can sort the table according to author and channel name.
Content Detail page	View all details pertaining to the selected content in this page including title, content summary, published date, and author. User can also read the content in SocialChorus instance by selecting **Read In The Blog**.

Integrate SolarWinds

June 3, 2021

Integrate with Solarwinds to submit and monitor tickets, service requests, and take action through Citrix Workspace. Higher tier agents have the ability to update tickets and service requests.

We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapp for SolarWinds, see Use SolarWinds microapps.

Review prerequisites

These prerequisites assume that the administrator is part of the SolarWinds integration set up of the organization.

You need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: `https://{ AccountName } .samanage.com`. See Find or change the account name.
- **API Key Value**: This value is used for **Value** when entering the **API Keys** parameters. See Collect API Token.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

User account

The integration requires regular access to your SolarWinds instance. We recommend creating a dedicated user account with the following permissions:

- **ROLE**: Administrator

Collect API Token

The SolarWinds administrator needs to collect an API Key token. You enter this in the field **Value** when entering the **API Keys** parameters.

1. Log in to SolarWinds Customer Portal using an account with account administrator access.
2. On the left side of the screen navigate to **Setup > Users & Groups > Users**.
3. Select the user name with the administrator role that will be used for the integration.
4. Next to **JSON Web Token**, select **Show Token**.
5. Copy and save the token value for later use as the API Key Value when configuring the integration, as shown in the next section.

Find or change the account name

To find or change your account name, follow these steps.

1. Log in to SolarWinds Customer Portal using an account with account administrator access.
2. In the user menu in the top-left corner of the screen, select **My Account**.
3. Copy or assign a value for **Account name**. This value will be used for the base URL: `https://{ AccountName } .samanage.com`

Add the integration to Citrix Workspace Microapps

Add the SolarWinds integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the SolarWindsn tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**:
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.
5. Under **Service Authentication**, select **API Keys** from the **Authentication method** menu. API Keys ensure that your integration meets the maximum security compliance.
   a) **Header** is selected for **API Keys Method** and *X-Samanage-Authorization* is entered for **Name**.
   b) For **Value**, enter the API Key Value that you collected earlier.
6. Leave the **Service Action Authentication** toggle as disabled.
7. Leave the **Request rate limiting** toggle as disabled.
8. Leave **Request timeout** empty.
9. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
10. Select **Save** to proceed.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see SolarWinds connector specifications.

Use SolarWinds microapps

Our SolarWinds integration template comes with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Create Ticket: Create a new incident.

Notification or Page	Use-case workflows
Create Ticket page	Provides a form to create a new ticket.

Delete Ticket: Delete an incident.

Notification or Page	Use-case workflows
List Tickets page	Provides a summary of all tickets requested by the user.
Ticket Details page	Provides a detailed page of a ticket and option to delete it.

My Assigned Tickets: View assigned tickets to update them and/or to change its status if needed.

Notification or Page	Use-case workflows
Assigned Ticket Created notification	When a new assigned ticket is created, the assignee receives a notification.
Assigned Ticket SLA Thresholds Hit notification	When an SLA threshold is hit, the assignee receives a notification with details.
Assigned Ticket Status Changed notification	When the status of a ticket changes, the assignee receives a notification.
My Assigned Tickets page	Provides a summary of all assigned tickets assigned by the user.
Ticket Details page	View details of a selected ticket with options to **Edit Ticket** the ticket and **Add comment**.
Ticket SLA Details page	View SLA details of a ticket.
Update Ticket / Change Status page	View and modify details of a ticket selected from the notification, with an **Update Ticket** option.

My Open Tickets: Allows the user to see his requested incidents and update them if needed.

Notification or Page	Use-case workflows
Ticket Created notification	When a new ticket is created, the requester receives a notification.
Ticket SLA Thresholds Hit notification	When an SLA threshold is hit, the requester receives a notification with details.
Ticket Status Changed notification	When the status of a ticket changes, the requester receives a notification.
My Open Tickets page	Provides a summary of all assigned tickets assigned by the user.
Ticket Details page	View details of a selected ticket with options to **Edit Ticket** and **Add comment**.
Ticket SLA Details page	View SLA details of a ticket.
Update/Close Ticket page	View and modify details of a ticket selected from the notification, with options to **Update Ticket** and **Resolve Ticket**.

My Open Service Requests: Allows the user to see his service requests and update them if needed.

Notification or Page	Use-case workflows
Service Request Created notification	When a new service request is created, the requester receives a notification.
Service Request SLA Thresholds Hit notification	When an SLA threshold is hit, the requester receives a notification with details.
Service Request Status Changed notification	When the status of a service request changes, the requester receives a notification.
My Service Request page	Provides a summary of all open service requests that are requested by the user.
Service Request Details page	View details of a selected service request with options to **Update Serviec Request**, **Open Service Request**, and **Add comment**.
Service Request SLA Details page	View SLA details of a service request.

Notification or Page	Use-case workflows
Update/Close Service Request page	View and modify details of a service request selected from the notification, with options to **Update Service Request** and **Resolve Service Request**.

Service Catalog Request: Search for a service catalog item by name or create common service requests quickly.

Notification or Page	Use-case workflows
Create Service Request	Provides a form to search for a service catalog item by name, and create a request for it, or use **Request Service** deeplink to create common service requests quickly.

Integrate SAP SuccessFactors

May 17, 2021

Integrate with SAP SuccessFactors for anywhere access to employee, skills, and course information.

> **Note:**
>
> We provide two SuccessFactors integration templates for your use. We recommend using the newer SuccessFactors EC HTTP integration for SAP SuccessFactors Employee Central use-cases. The HTTP integration provides more power to configure the cached data structure.

For a comprehensive list of out-of-the-box SuccessFactors microapps, see Use SuccessFactors microapps.

Review prerequisites

After you set up this integration in SAP SuccessFactors, you will need these artifacts to add the integration in Citrix Workspace Microapps based on the type of integration you need to enable. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

- The required configuration information to connect with SAP SuccessFactors depends on whether you use the Learning module.
- Create an Admin user in the Provisioning instance. Typically a SuccessFactors Certified Consultant performs all activities in Provisioning. Give the user a distinguishable name.
- Configure Citrix Gateway to support single sign-on for SuccessFactors so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.

> **Note:**
>
> Rate limits apply for SuccessFactors integrations to the number of requests per minute. This can impact testing instances. To avoid issues, set rate limits to 8 calls per second. For more information, consult your SuccessFactors consultant to find out the correct maximum request rate value.

For the SuccessFactors EC integration:

- **Base URL**: Your base URL follows this model: `https://{ tenant } .successfactors.{ region } /odata/v2`
- **Username**: Your unique user ID.
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server. See Collect your Company ID and Client ID.
- **Company ID**: The company ID is a short string of characters that identifies each SAP SuccessFactors system, like a username for your organization. See Collect your Company ID and Client ID.
- **Private Key**: This is the API key from registering the OAuth2 Client. See Register the OAuth2 Client.
- **OAuth URL**: This is the Application URL generated in the template, and follows this model: `https://{ tenant } .successfactors.{ region } /oauth`. You need this to Register the OAuth2 Client.

For the SuccessFactors HCM integration:

- API URL
- Company ID
- User ID (Username)
- Client ID (API Key)
- Client Private Key (Encrypted Private Key)

For the SuccessFactors Learning integration:

- Learning URL
- Learning Company ID
- Learning User ID
- Learning Client ID

- Learning Client Secret

Set up SuccessFactors HCM integration

Follow this process if you need to set up the basic SuccessFactors HCM integration or the basic integration with Learning Module. Using your admin user, you create a permission role, create a permission group, and assign the permission group to the permission role.

Create a permission role

To create a permission role, follow these steps:

1. Log in to SAP SuccessFactors Admin Center with your Admin user.
2. Search for and select **Manage Permission Roles**, and select **Create New**.
3. Enter a meaningful **Role Name** and select **Permission…**.
4. Scroll to **Manage Integration Tools**, click **Select All**, and select **Done**.
5. Select **Save Changes**.

Create a permission group

To create a permission group, follow these steps:

1. In the SAP SuccessFactors Admin Center, search for and select **Manage Permission Groups**, and select **Create New**.
2. Enter a meaningful **Group Name**.
3. Under **Choose Group Members: People Pool**, select category **Username**.
4. Enter the username of the dedicated user, select the check box next to the name, and select **Done**.
5. Select **Done** again.

Assign the new permission group to the permission role

To assign the new permission group to the permission role, follow these steps:

1. In the SAP SuccessFactors Admin Center, search for and select **Manage Permission Roles**, and select the previously created permission role.
2. Scroll down to **Grant this role to…** and select **Add…**.
3. Under **Grant this role to: Permission Group…**, click **Select…**.
4. Search for the previously created group, select the check box next to the name, and select **Done**.
5. Select **Done** again, and select **Save Changes**.

   You assigned the user permission group to the permission role.

Register the OAuth2 Client

To register the OAuth2 Client, follow these steps:

1. In the SAP SuccessFactors Admin Center, search for and select **Manage OAuth2 Client Applications**, and select **Register Client Application**.
2. Enter the following details:

   Application Name

   Application URL
3. Select **Generate X.509 Certificate**.
4. Enter a **Common Name (CN)**, and select **Generate**.
5. Select **Download** to download a copy of the **X.509 Certificate**. The Client Private Key is located within the certificate file as **Encrypted Private Key**. Copy and save this key. You use these details when configuring the integration.
6. Select **Register**.

   The new application is listed on the Manage Oauth2 Client Applications page.
7. Under **Actions**, select **View**.
8. Copy the **API Key** and store it for later use.

Set up SuccessFactors Learning integration

Follow this process if you need to set up the basic SuccessFactors integration with Learning Module or just the Learning module. Using your admin user, you collect the Company ID and Client ID, and generate a new Client Secret.

Collect your Company ID and Client ID

To collect your company ID and client ID, follow these steps:

1. Log in to SAP SuccessFactors Learning administration environment for your tenant.
2. Navigate to **System Admin > Configuration > OAuth Token Server**.
3. On the **Application Administration** screen, copy the **Company ID** and **Client ID** and store it for later use.

Generate a new client secret

To generate a new client secret, follow these steps:

1. On the **Application Administration** screen, select **Generate a new Client Secret** and confirm by selecting **OK**.

   The **Newly Generated Client Secret** populates below the Client ID.

2. Copy the client secret and store it for later use.

   The secret is not stored. When you leave the OAuth Token Server page, the secret disappears.

Add callback URLs

Add a custom URL to your instance configuration to grant access to private data and enable OAuth authenticated user actions. The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com.

Log in to SuccessFactors as an admin and add the following authorized redirect URLs for this integration:

- `https://{ yourmicroappserverurl } /admin/api/external-services/com.sapho.services.successfactors.SuccessFactorsService/auth/serverContext`
- `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`

Add the SuccessFactors EC integration

Follow these steps to set up the SuccessFactors EC integration. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the SuccessFactors EC tile under **Integrations**.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**: `https://{ tenant } .successfactors.{ region } /odata/v2`
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

   a) Select **SAML 2.0 Success Factors** from the **Grant type flow** menu.
   b) Leave **Scope** empty.
   c) Enter your **Username**.
   d) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. See Collect your Company ID and Client ID.
   e) Enter your **Company ID**. The company ID is a short string of characters that identifies each SAP SuccessFactors system, like a username for your organization. See Collect your Company ID and Client ID.
   f) Enter your **Private Key**. This is the API key from registering the OAuth2 Client. See Register the OAuth2 Client.
   g) Your **OAuth URL** is automatically generated. This is the Application URL generated in the template, and follows this model: `https://{ tenant } .successfactors.{ region } /oauth`. You need this to Register the OAuth2 Client.

6. Leave **Service Action Authentication** disabled.

7. The **Enable request rate limiting** toggle is enabled. Leave *1* for **Number of requests** and *1 second* for **Time interval**.
   1.. **Request timeout** is set to *120* by default.

8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

9. Select **Save** to proceed.

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

Use SuccessFactors EC microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs. Our SuccessFactors ECintegration comes with the following preconfigured out-of-the-box microapps:

Directory: Search for employees and preview their details including skill set.

Notification or Page	Use-case workflows
User page	Provides a searchable list of users.
Users Skills page	Provides a view of user details and their skill set.

Skills: Search for skills and preview employees with corresponding skill set.

Notification or Page	Use-case workflows
Skill Rating Changed notification	When a manager changes the rating of a skills of an employee, the employee receives a notification.
Skills page	Provides a searchable list of skills to connect to users.
User Rated Skills page	Provides a detailed view of rated skills. Rated skills are skills that employees and their managers rate in the Skills Profile portlet.
User Self Reported Skills page	Provides a detailed view of self-reported skills. Self-reported skills are manually added in the Skills Profile portlet.
Users page	Provides a view of a user's skill set.

Add the SuccessFactors integration to Citrix Workspace Microapps

Add the SuccessFactors integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Use the following process to enable the SuccessFactors Integration. Ensure you meet the prerequisites, and decide which integration you need to set up:

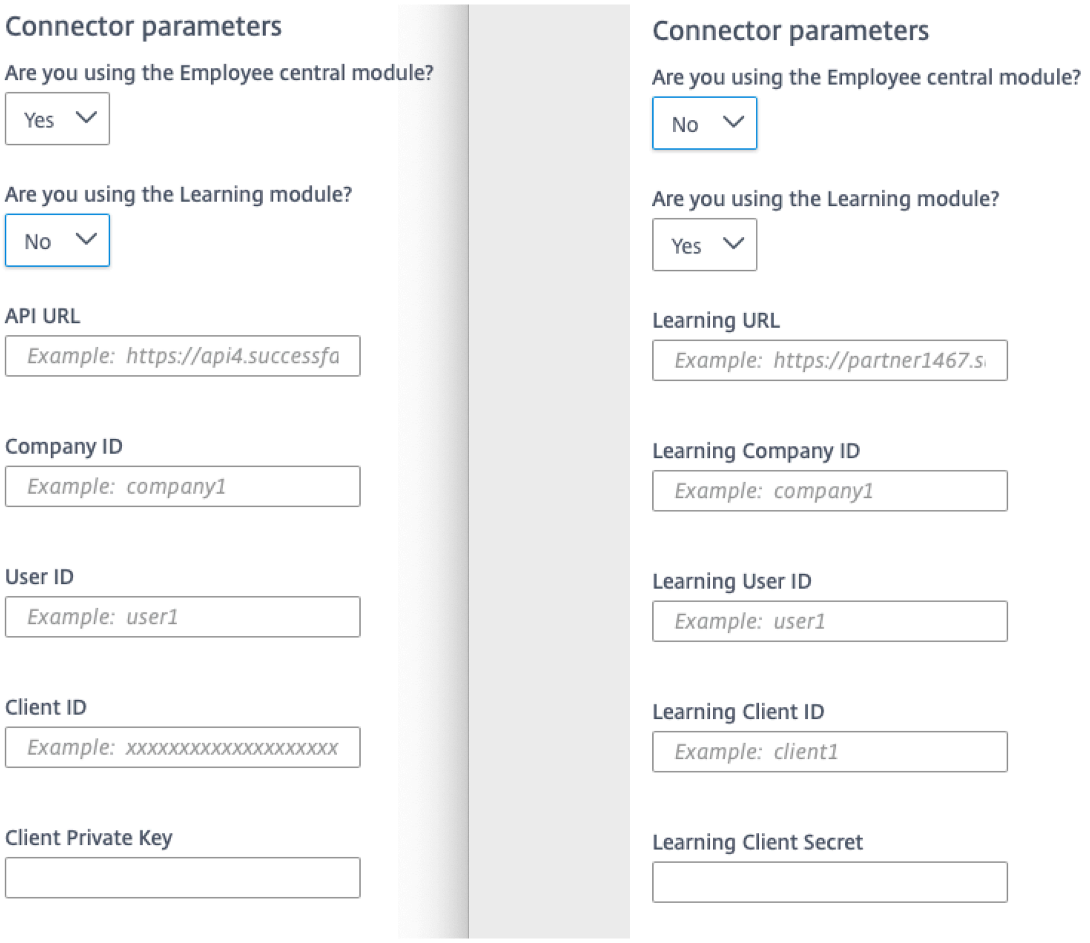
- the basic SuccessFactors HCM integration,
- the basic integration with Learning Module,
- just the Learning module.

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the SuccessFactors tile.
4. Enter a name for the integration.



Connector parameters

Are you using the Employee central module?

Yes

Are you using the Learning module?

No

API URL

Example: https://api4.successfa

Company ID

Example: company1

User ID

Example: user1

Client ID

Example: xxxxxxxxxxxxxxxxxxxx

Client Private Key

Connector parameters

Are you using the Employee central module?

No

Are you using the Learning module?

Yes

Learning URL

Example: https://partner1467.s

Learning Company ID

Example: company1

Learning User ID

Example: user1

Learning Client ID

Example: client1

Learning Client Secret

5. Enter the **Connector parameters** that you collected as prerequisites.
   - Select Yes/No from the **Are you using the Employee central module?**
     - Enter the **API URL**. For example, `https://api12preview.sapsf.eu/odata/v2v`.
     - Enter the **Company ID**.
     - Enter the **User ID**.
     - Enter the **Client ID**.
     - Enter the **Client Private Key**.
   - Select Yes/No from the **Are you using the Learning module?**
     - Enter the **Learning URL**.
     - Enter the **Learning Company ID**.
     - Enter the **Learning User ID**.
     - Enter the **Learning Client ID**.
     - Enter the **Learning Client Secret**.

6. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see SuccessFactors connector specifications.

Use SuccessFactors microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Our SuccessFactors integration comes with the following preconfigured out-of-the-box microapps:

Directory: Search, view, and edit employees with corresponding details.

Notification or Page	Use-case workflows
New Teammate notification	When a new teammate joins, all subscribers receive a notification highlighting the new teammate and their position.
Position Changed notification	When the title of an employee changes, all subscribers receive a notification highlighting the teammate and their new position.
My Detail page	Provides a form for viewing personal details and provides a link to manager subdetails.
My Team page	Provides a table view of an employee's teammates and links to user details.
User Detail page	Provides a form for viewing a user's details, and provides a link to their manager's and any direct reports' subdetails.
User SubDetail page	Provides a form for viewing a user's subdetails, and provides a link to their details.
Users page	Provides a table view of users with search functionality and a link to user details.

Learning: Search, view, share, and register available learning courses.

Notification or Page	Use-case workflows
Popular Course notification	When a learning course is defined as popular based on its rating, all subscribers receive a notification.
Courses page	Provides a list of available courses with a link to learning item details.
Learning Item Detail page	Provides a table view of learning items with a link to scheduled offering details and an option to share by email.
Scheduled Offering Detail page	Provides a detailed view of a scheduled offering with a list of instructors and an option to register for the offering.

Integrate Tableau

June 22, 2021

Integrate with Tableau to provide easy access to projects, workbooks, and views without requiring extra logins.

> **Note**
>
> We provide two Tableau integration templates for your use. We recommend using the newer HTTP integration for most use-cases as it provides more power to configure the cached data structure. For full details of the microapps available in each integration, see Use Tableau microapps.

These instructions describe how to set up the new HTTP template integration. If you need information about the legacy template, see Add the legacy integration.

Review prerequisites

After you set up this integration in Tableau, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: The base URL takes this form: `https://{ tenantID } .online.tableau.com /`

- **Username**: The Tableau user name of the site admin.
- **Password**: The password for the site admin.
- **Site**: In Tableau a collection of users, groups, and content. The Site ID is found in the URL after logging in to Tableau Online: `https://{ tenantID } .online.tableau.com/##/site/{ siteID } /home`
- Configure Tableau Server to recognize and trust requests by whitelisting its IP address.
- Only https connections are supported. Make sure the SSL certificate is trusted.
- Configure Citrix Gateway to support single sign-on for Tableau so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Tableau Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Set up the Tableau integration

1. Log in to Tableau with an admin account.
2. Enter connection information:
   - Name
   - URL
   - Username
   - Password
   - Site

   > **Note:**
   >
   > If you leave the **Site** field empty, you are connected to the "Default" Tableau Site. To find the names of the different Sites available in your Tableau instance, select the menu in the Tableau top navigation bar.

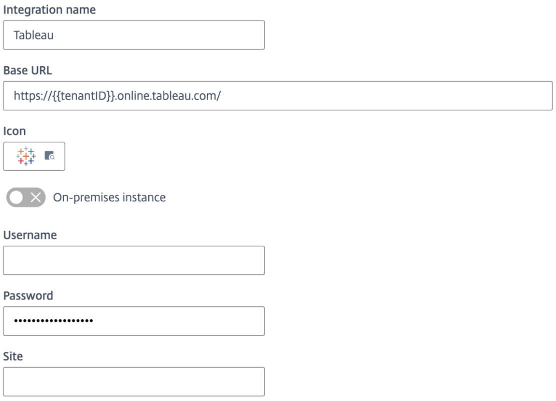
Add the integration to Microapps

Add the Tableau integration template to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the **Tableau** tile from the **Integrations** category of the catalog.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
   - Enter the instance **Base URL**. The base URL takes this form: `https://{ tenantID } .online.tableau.com/`
   - Select an **Icon** for the integration from the Icon Library, or leave this as the default Workday icon.



Integration name

Tableau

Base URL

https://{{tenantID}}.online.tableau.com/

Icon

On-premises instance

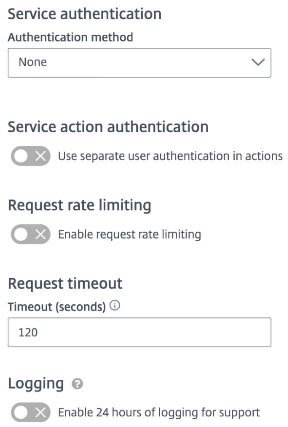
Username

Password

••••••••••••••••••

Site

5. Enter your **Username**.
6. Enter your **Password**.
7. Enter your **Site**.
8. Leave **Service authentication** and **Service action authentication** disabled. They are not used for this integration.
9. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.
10. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
11. The value *120* is prefilled in the **Request timeout** field.
12. Select **Save** to proceed.



Service authentication

Authentication method: None

Service action authentication

Use separate user authentication in actions

Request rate limiting

Enable request rate limiting

Request timeout

Timeout (seconds): 120

Logging

Enable 24 hours of logging for support

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, see Synchronize data.

For more details of API endpoints and table entities, see Tableau connector specifications.

Add the legacy integration

Add the Tableau integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

After you set up this integration in Tableau, you will need these artifacts to add the integration in Citrix Workspace Microapps:

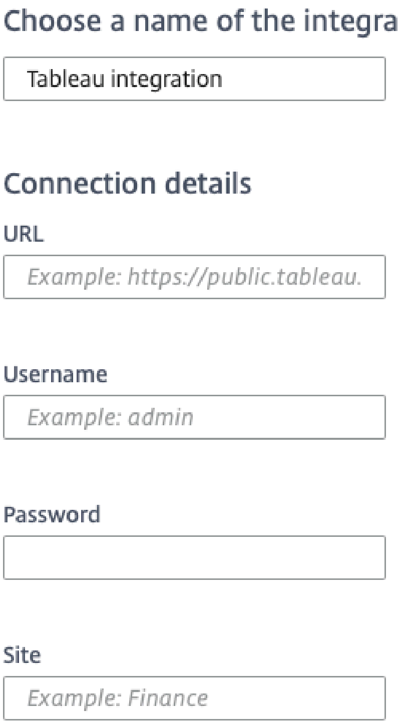
- URL
- Username
- Password
- Site
- Configure Tableau Server to recognize and trust requests by whitelisting its IP address.
- Only https connections are supported. Make sure the SSL certificate is trusted
- Configure Citrix Gateway to support single sign-on for Tableau so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Tableau Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Follow these steps:

1. From the overview page, select **Get Started**.

   The microapp Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Tableau tile.
4. Enter a name for the integration.



Choose a name of the integration

Tableau integration

Connection details

URL: Example: https://public.tableau.

Username: Example: admin

Password:

Site: Example: Finance

5. Enter the **Connector parameters** that you collected in the previous procedures.
   - Enter your **URL**.
   - Enter your **Username** and **Password**.
   - Enter your Tableau **Site** location.
6. Select **Add**.

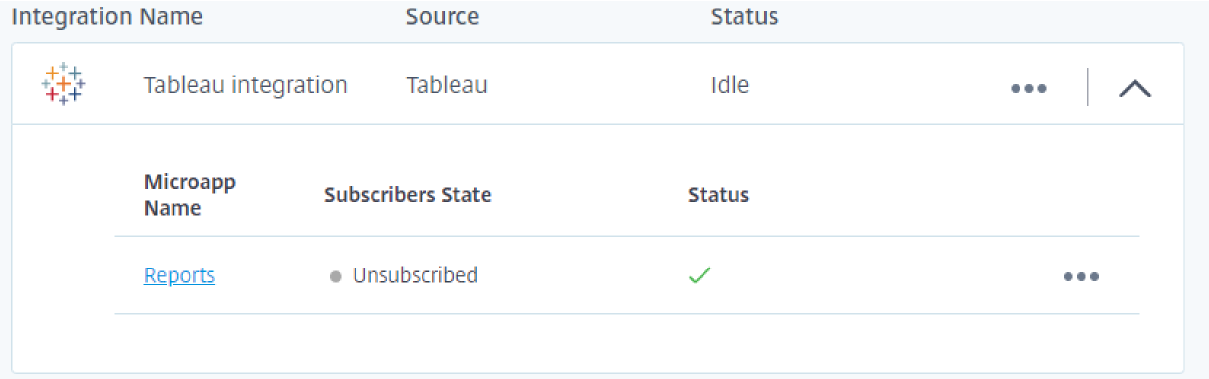
The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Tableau connector specifications.

Use Tableau microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.



Integration Name	Source	Status
Tableau integration	Tableau	Idle

Microapp Name	Subscribers State	Status
Reports	● Unsubscribed	✓

Our Tableau integration comes with the following preconfigured template microapp:

Reports: View details of Tableau reports.

Notification or Page	Use-case workflows
New Report Created notification	When a new report is created, users receive a notification.
Report Updated notification	When a report is updated, users receive a notification.
View Detail page	Provides a read only view of report details.
Views page	Provides a list of report views with a link to view details.

Integrate Webex

April 20, 2021

Deploy the Webex integration to schedule Webex Meetings from any device or intranet. Users can host one-time/recurring meetings, add invitees and co-hosts, and select from different timezones. The microapp also follows up with an email to the host and invitees with the corresponding meeting object for easy calendar integration.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For

> any issues, our team will also monitor our dedicated forum on a daily basis.

For a comprehensive list of out-of-the-box Webex microapps, see Use Webex microapps.

Review prerequisites

These prerequisites assume that administrator will be a part of the Webex integration set up of the organization. This Webex admin account must have full read privileges for user information.

After you set up this integration with Webex, you will need these artifacts to add the integration in Citrix Workspace Microapps, specifically the following list of parameters for setting up OAuth integration:

- BASE URL: `https://webexapis.com/v1/`
- AUTHORIZATION URL: `https://webexapis.com/v1/authorize`
- TOKEN URL: `https://webexapis.com/v1/access_token`
- CLIENT ID: The client ID is the string representing client registration information unique to the authorization server.
- SECRET: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for Webex so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service.

The integration requires regular access to your Webex instance, so we recommend creating a dedicated user account. This account must have the following permissions. You can view the permission/privileges using Webex Control Hub on .

- Permissions required for Service Account: Full administrator privileges

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Webex API limitation form link: https://developer.webex.com/docs/api/basics#rate-limiting
- Recommended plan: Webex Plus

Enable APIs

Webex APIs are enabled by default through web services for paid accounts.

Create a new service account

Sign up here: https://web.webex.com/. Refer to the below URL for new Service Accounts: https://help.webex.com/en-us/nkhozs6/Get-Started-with-Cisco-Webex-Control-Hub.

Configure OAuth server

Configure the OAuth server to read data through the Webex integration.

1. Log in with your service account to: https://developer.webex.com/docs/platform-introduction.
2. Select the user name present on the top-right.
3. Select **My Webex Apps** and select **Create a New App**.
4. Select **Create an Integration** under the Integration tile.
5. Complete the required fields and enter the following authorized redirect URLs for this integration in the **Redirect URL** field:
    - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
6. Under **Scopes** section, select the **spark:all** and **spark-admin:people_read** check boxs.
7. Select **Add Integration** after you complete all the required fields.
8. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Authentication** while configuring the integration.

Configure OAuth client

Configure the OAuth client for writing back data through the Webex integration.

1. Log in with your service account, as above: https://developer.webex.com/docs/platform-introduction.
2. Select the user name present on the top-right.
3. Select **My Webex Apps** and select **Create a New App**.
4. Select **Create an Integration** under the Integration tile.
5. Complete the required fields and enter the following authorized redirect URLs for this integration in the **Redirect URL** field:
    - `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`
6. Under **Scopes** section, select the **meeting:schedules_write** check box.

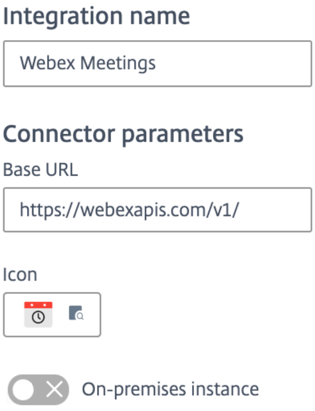
7. Select **Add Integration** after you complete all the required fields.
8. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

Add the integration to Citrix Workspace Microapps

Add the Webex Meeting integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Webex Meetings tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**: `https://webexapis.com/v1/`.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

Webex Meetings

Connector parameters

Base URL

https://webexapis.com/v1/

Icon

On-premises instance

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.
    a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization

server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.

b) Select **Request body** from the **Token authorization** menu.

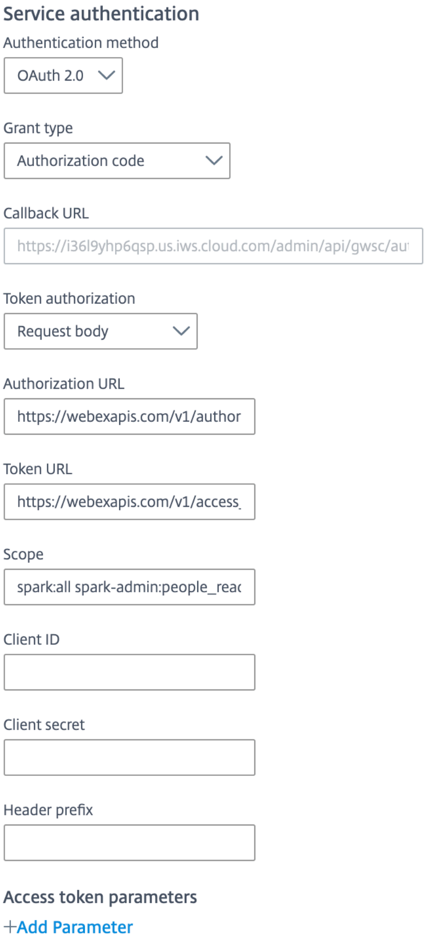
c) The **Authorization URL** is prefilled: `https://webexapis.com/v1/authorize`

d) The **Token URL** is prefilled: `https://webexapis.com/v1/access_token`

e) Ensure the following is entered for Scope: *spark:all spark-admin:people_read*

f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service authentication

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Request body

Authorization URL: https://webexapis.com/v1/author

Token URL: https://webexapis.com/v1/access_

Scope: spark:all spark-admin:people_reac

Client ID:

Client secret:

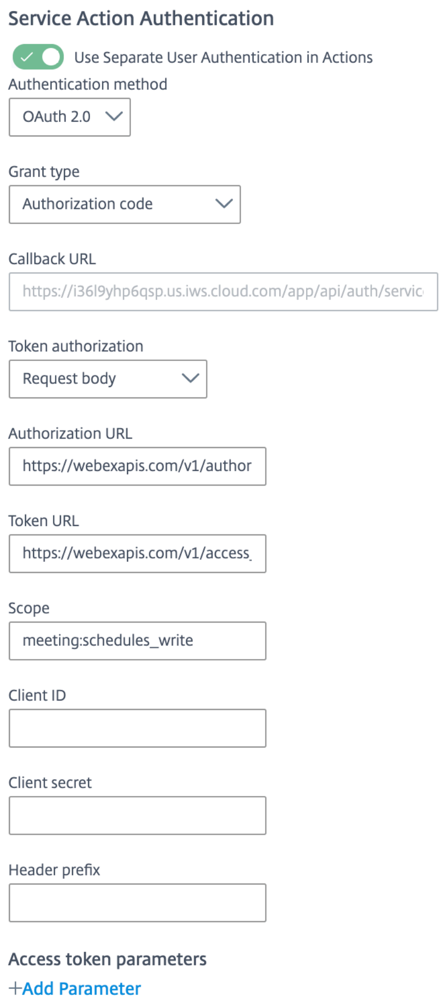
Header prefix:

Access token parameters

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete

the process.

a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application.
c) Select **Request body** from the **Token authorization** menu.
d) The **Authorization URL** is prefilled: `https://webexapis.com/v1/authorize`
e) The **Token URL** is prefilled: `https://webexapis.com/v1/access_token`
f) Ensure the following is entered for Scope: *meeting:schedules_write*
g) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.
h) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service Action Authentication

Use Separate User Authentication in Actions

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Request body

Authorization URL: https://webexapis.com/v1/author

Token URL: https://webexapis.com/v1/access_

Scope: meeting:schedules_write

Client ID

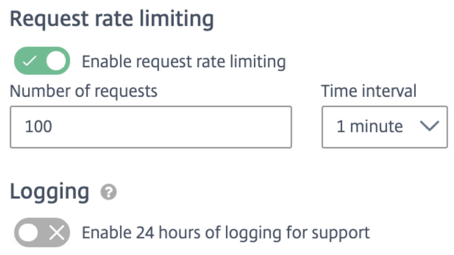
Client secret

Header prefix

Access token parameters

+Add Parameter

7. Enable the **Enable request rate limiting** toggle. Enter *100* for **Number of requests** and *1 minute* for **Time interval**.
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

Enable request rate limiting

Number of requests: 100

Time interval: 1 minute

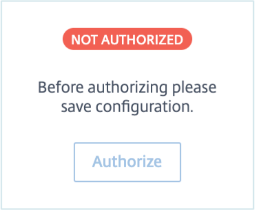
Logging

Enable 24 hours of logging for support

9. Select **Save** to proceed.
10. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Webex login screen.
    a) Enter your Service Account user name and password and select **Log in**.

b) Select **Accept**.

OAuth Authorization


NOT AUTHORIZED

Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Webex HTTP connector specifications.

Use Webex Meetings microapp

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Create a Meeting: Schedule a meeting with the option to select duration, time zones, invitees, and co-hosts.

Notification or Page	Use-case workflows
Create a Meeting page	Provides a form to schedule a meeting with the following details according to user preference: Meeting Title, Start and End Time, Time Zone, Recurrence (once, daily, weekly, monthly), Password, Meeting Attendees, and Co-host for the meeting.

Integrate Workday

June 14, 2021

Integrate with Workday to make it easy to submit requests, receive notifications about request status, and act on notifications. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps. Use the following process to enable the Workday Integration. For a comprehensive list of out-of-the-box Workday microapps, see Use Workday microapps.

> **Note**
>
> We provide two Workday integration templates for your use. We recommend using the newer HTTP integration for most use-cases as it provides more power to configure the cached data structure. For full details of the microapps available in each integration, see Use Workday microapps.

These instructions describe how to set up the new HTTP template integration. If you need information about the legacy template, see Add the legacy integration. A quick overview of the process:

1. Before setting up the integration in Microapps, ensure you meet the prerequisites and then complete the following procedures that are described in detail below:
   - Enable API access by registering an API Client and generating and collecting the Client ID and Client Secret.
   - Create custom reports.
   - Generate custom report endpoint path.
   - Filter PTO Types to restrict PTO microapps to specific time-off types.
   - Set up deep linking in Workspace.
   - Edit business process for time-off requests.
   - Manage security group permissions.
   - Identify your URL, Instance URL, and Tenant.
2. Set up the Workday integration. For more information, see Add the Workday integration to Citrix Workspace Microapps.
3. Configure Service action parameters. For more information, see Replace Data Loading and Service Action variables.

If you need help, have a look at our Troubleshoot common Workday integration errors article.

Review prerequisites

This set up process requires you to have an admin account and the following service account privileges for the Workday connector.

- Access to the creation of non-temporary custom reports.
- Access to HumanResources, Integrations, PerformanceManagement, ResourcesManagement, and custom reports APIs.

- For downloading milestone data, part of the deep link set-up process, you need the following security groups assigned to the admin:
  - HR Administrator
  - Information Administrator
- Configure Citrix Gateway to support single sign-on for Workday so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Workday Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

After you set up this integration in Workday, you need these artifacts to enter these account credentials when you set up the integration in Microapps:

- URL
- Username (Workday username)
- Password (Workday password)
- Workday tenant
- Client ID
- Client Secret
- Workday REST API Endpoint
- Token Endpoint
- Authorization Endpoint
- Custom report URLs collected

Register Workday API Client

Register an API client to generate a Client ID and Client Secret for each environment. If you have multiple environments, you must register an API Client for each individual environment. For the Workday integration, you must add two different callback URLs. This means you need to register two API clients; one for user actions and the other for synchronization. Perform this procedure twice. Once for **Service Authentication** and then for **Service Action Authentication**. They have different callback URLs. For the Legacy integration template, only perform the second operation to register the client.

Configure OAuth server

Configure the OAuth server to read data through the integration. You use these details for Service Authentication while configuring the integration. This is only performed for the HTTP integration.

1. Log in to Workday as an admin, and search for *Register API Client*. Complete the required fields:

   - Enter your **Client Name**.
   - Select **Authorization Code Grant** for **Client Grant Type**.
   - Select **Bearer** for **Access Token Type**.

  - Enter the **Redirection URI**: `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`. The callback depends on the target application, and can be found in your URL address bar when creating the integration. The section `{ yourmicroappserverurl }` is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region(us/eu/ap-s)} .iws.cloud.com`.
  - Enter the value *300* for **Refresh Token Timeout (in days)**.
  - Select the following **Scope (Functional Areas)**:
    - **Organizations and Roles**
    - **Staffing**
    - **Tenant Non-Configurable**

2. Select **OK**. The Client ID and Client Secret are generated. Collect and save for later use during the set-up process. You use these details for Service Authentication while configuring the integration.

3. Select **Done** to complete and exit.

Configure OAuth client

Configure the OAuth client for writing back data through the integration. You use these details for Service Action Authentication while configuring the integration.

1. Log in to Workday as an admin, and search for *Register API Client*. Complete the required fields:

   - Enter your **Client Name**.
   - Select **Authorization Code Grant** for **Client Grant Type**.
   - Select **Bearer** for **Access Token Type**.
   - Enter the **Redirection URI**: `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`. The callback depends on the target application, and can be found in your URL address bar when creating the integration. The section `{ yourmicroappserverurl }` is composed of a tenant part, a region part, and an environment part: `https://{ tenantID } .{ region(us/eu/ap-s)} .iws.cloud.com`.
   - Enter the value *300* for **Refresh Token Timeout (in days)**.
   - Select the following **Scope (Functional Areas)**:
     - **Organizations and Roles**
     - **Staffing**
     - **Tenant Non-Configurable**

2. Select **OK**. The Client ID and Client Secret are generated. Collect and save for later use during the set-up process. You use these details for Service Action Authentication while configuring the integration.

3. Select **Done** to complete and exit.

Test API calls

This article lists all Workday API calls and provides a detailed description of how to test whether your Workday instance has all endpoints ready for these calls. For more information, see Test Workday API calls.

Create custom reports

You create custom reports in Workday for each of the following reports. Download the attached spreadsheets and complete the details precisely as described in the spreadsheet. If any detail you enter differs from the provided report, the process does not work. For example, if you use an incorrect name, the report is generated, but no data is downloaded.

The following reports are the current reports. When we add new end-points to this integration, they are added here in the product documentation.

1. Open one of the following custom report spreadsheets:
   - All time offs and balance
     - Calculated field: cf lkp time off balance year
     - Calculated field: cf lkp time off plan default quantity
     - Calculated field: cf lkp time off plan id
     - Calculated field: cf lkp time off type id
     - Calculated field: cf time off balance year
     - Calculated field: cf yearend for reporting effective years
   - Absence requests
     - Calculated field: cf lkp time off event id
   - Event records for change job
     - Calculated field: cf esi worker
     - Calculated field: cf lkp worker id
   - Staffing activities
     - Calculated field: cf esi event record
     - Calculated field: cf lkp event record
   - Event records for time off requests
     - Calculated field: cf esi event records awaiting action
     - Calculated field: cf esi assigned to worker of event records of awaiting action

    - Calculated field: cf lrv assigned to worker email of event records awaiting action
    - Calculated field: cf lrv assigned to worker event records of awaiting action
    - Calculated field: cf lrv wid of assigned to worker email of event records awaiting action
    - Calculated field: cf lrv wid of event record awaiting action
  - Worker

    > **Note:**
    >
    > Do not add self-referencing objects to the Time off types per plan report.

2. In Workday, search for **Create Custom Report**.
3. Enter a **Report Name**. This name must be identical to the spreadsheet of the report that you want to create a URL for.
4. Select **Advanced** as **Report Type**.
5. Enter the **Data Source**. This value must be identical to the spreadsheet of the report that you want to create a URL for.
6. Do not select the **Optimized for Performance** check box.
7. Do not select the **Temporary Report** check box.
8. Select the **Enable As Web Service** check box.
9. Select **OK**.

   The custom report opens unpopulated except for the three fields you entered.
10. Complete the details exactly as described in the spreadsheet. Pay attention to the headings. The headings match the tabs in the Workday UI.
11. The field **Column Heading Override XML Alias** is auto-generated as you are populating the columns. Verify that the value in **Column Heading Override XML Alias** matches the spreadsheet instructions. This value often varies.
12. Select **OK**. The custom report is created.

Generate custom report endpoint path

You collect the custom report path by using the generated WSDL from the previous procedure. Use this when you enter the custom report URL for **Time off types per plan report path** and **Event records for milestone path** in the set-up procedure.

1. Open the generated WSDL link in your browser.
2. Scroll to the bottom and locate two URLs. The second one has the name *ReportREST*. This is the one we want to use.

3. Copy the path from this URL and use it when setting up the Workday integration. (Example path: </ccx/service/customreport2/company_tenant/user_name/report_name>.

Filter PTO Types

Restrict PTO microapps to specific time-off types, such as vacation, or sick leave. If you created the following custom reports in Workday, you need to create calculated fields in Workday and filter custom reports using the following procedures.

- Event records for time off requests
- Absence requests
- All time offs and balance

Create calculated fields for Time Off Requests microapp

You need to create two calculated fields. The first calculated field is intended to retrieve the Absence Type from the Time Off Event so it is available in the second calculated field. The second calculated field is used as the filter in the custom report.

1. Log in to your Workday instance and search for *create calculated field*.
2. Complete all fields:
    - **Field Name** Enter a name for the calculated field. For example, *SANDOVAL CF LKP Absence Type* from this format: {(report creator)(CF=calculated field)(LKP=lookup related value)(values this field returns)}.
    - **Business Object** Find and select **Time Off Event**. We want our calculated field to be part of this business object.
    - **Function** Find and select **Lookup Related Value**.
3. Select **OK** at bottom left.
4. Under the **Calculation** tab, complete these fields:
    - **Lookup field** Find and select **Time Off Event**.
    - **Return Value** Find and select **Time Off Types for Time Off Event**
5. Select **OK** and then **Done**.

You created a calculated field. Now, let's create the second field that retrieves the Time Off Event and will be used as the filter.

1. Again, from your Workday instance search for *create calculated field*.
2. Complete all fields:
    - **Field Name** Enter a name for the calculated field. For example, *SANDOVAL CF LKP Time Off Event for Business Process* from this format: {(report creator)(CF=calculated field)(LKP=lookup related value)(values this field returns)}.
    - **Business Object** Find and select **Action Event**. We want our calculated field to be part of this business object.

- **Function** Find and select **Lookup Related Value**.

3. Select **OK** at bottom left.
4. Under the **Calculation** tab, complete these fields:
    - **Lookup field** Find and select **Time Off Event**.
    - **Return Value** Find and select the first calculated field you created. In our example, **SANDOVAL CF LKP Absence Type**.
5. Select **OK** and then **Done**.

Filter a custom report using calculated field for Time Off Requests microapp

Add a filter to the custom report related to time off approval using the new calculated field that you just created to whitelist PTO Types, which will allow you to get all time off types you select.

1. From your Workday instance search for *Edit Custom Report* and select a custom report related to PTO approvals, in our case **Event Records for Time Off Requests**.
2. Select **Filter** tab and **+** to add new filters.
3. Complete these fields:
    - **And/Or** And
    - **Field** Find and select the second calculated field you created, in our example **SANDOVAL CF LKP Time Off Event for Business Process**.
    - **Operator** Find and select **any in the selection list**.
    - **Comparison Type** Find and select **Value specified in this filter**.
    - **Comparison Value** Find and select the PTO types that you want to whitelist, for example:
        - **Annual Leave (Days)**
        - **Annual Leave (Statutory)**
        - **Personal Leave (Days)**
        - **Sick (Days)**
        - **Time Off**
4. Select **OK** to save.

> **Note:**
>
> To block PTO Types, change field: ***Operator** to **none of the selection list**. This blocks the integration from getting any selected PTO types.

Filter a custom report using an existing field for Create PTO Requests and My PTO Requests microapps

Filter custom report: **All time offs and balance**.

> **Note:**
>
> We recommend using the same PTO types in all custom reports. This way if you decide to whitelist or block the outcome, the values in **Comparison Value** is the same for all custom reports).

1. From your Workday instance search for *Edit Custom Report* and select a custom report related to time off approvals, in our case **All time offs and balance**.
2. Select **Filter** tab and **+** to add new filters.
3. Complete these fields:
    - **And/Or** And
    - **Field** Find and select the custom report **Time Off Type**.
    - **Operator** Find and select **in the selection list**.
    - **Comparison Type** Find and select **Value specified in this filter**.
    - **Comparison Value** Find and select the PTO types that you want to whitelist, for example:
        - **Annual Leave (Days)**
        - **Annual Leave (Statutory)**
        - **Personal Leave (Days)**
        - **Sick (Days)**
        - **Time Off**
4. Select **OK** to save.

Filter custom report: **Absence requests**.

1. From your Workday instance search for *Edit Custom Report* and select a custom report related to time off approvals, in our case **Absence requests**.
2. Select **Subfilter** tab and select **+** to add new subfilters, similar to how you added filters.
3. Complete these fields:
    - **And/Or** And
    - **Field** Find and select the custom report **Time Off Type for Time Off Entry**.
    - **Operator** Find and select **in the selection list**.
    - **Comparison Type** Find and select **Value specified in this filter**.
    - **Comparison Value** Find and select the PTO types that you want to whitelist, for example:
        - **Paid Time Off (Days)**
        - **Sick (Days)**
        - **Sick (Hours)**
4. Select **OK** to save.

Set up deep link in Workspace

If you are referred to this article, contact your administrator and request that they set up deep linking for Citrix Workspace Microapps in Workday.

The following use-cases require the corresponding deep links. For example, if your domain is `impl.workday.com` and your tenant is `citrix_gms2v`, then the URL for *Create expense report* is: `https://impl.workday.com/citrix_gms2/d/task/2997$995.htmld`:

- **Create Expense Report** `https://your_domain/your_tenant/d/task/2997$995.htmld`
- **Create Change Job** `https://your_domain/your_tenant/d/task/2997$4819.htmld`
- **Approve/Deny Change Job (Change Job Requests)** `https://your_domain/your_tenant/d/unifiedinbox/initialinbox/2998$17139.htmld`

> **Note:**
>
> If you do not have this data stored, log in to your Workday account and copy them from the URL. These deep links must be set up with every newly added integration.

Edit business process for time-off requests

Depending on the Workday tenant that you are using, you have a business process established for creating time-off requests. Modify that business process as shown in the following steps.

1. In Workday, search for *bp: request time off*, and then select **Request Time Off for Global Modern Services**.
2. Go to **Actions > Business Process > Edit Definition**.
3. Do not change **Effective Date** unless required, and select **OK**.
4. Under **Business Process Steps**, select the **+** icon to add a new row.
5. Enter *b* for **Order**.
6. Enter *Approval* for **Type**.
7. Select *Manager* for **Group** from the menu.
8. Select **OK**, and then **Done**.

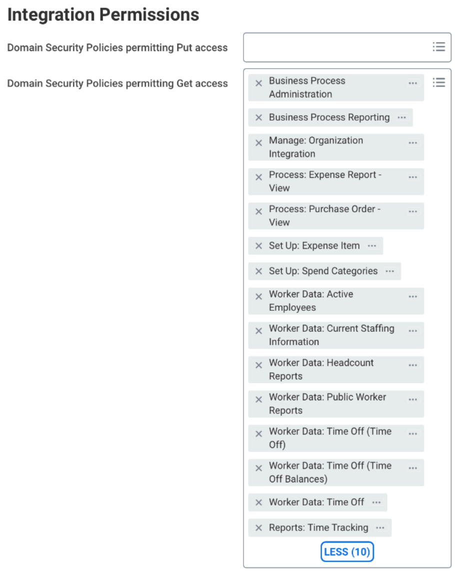
Manage security group permissions

To enable proper security permissions for your security group, you complete two procedures. First add permissions to the group and activate the settings. Then add the security group to view all list and activate the settings.

Add and activate integration permissions

1. In Workday, search for *view security group*.
2. Search for the security group that you need, select it, and select **OK**.
3. Go to **Actions > Security Group > Maintain Security Permissions**.

4. Under the section **Integration Permissions**, search for the following list of permissions in the **Domain Security Policies permitting Get access** field, and add them all. Do not use colons *(:)* in your search:



Integration Permissions

Domain Security Policies permitting Put access

Domain Security Policies permitting Get access

- Business Process Administration
- Business Process Reporting
- Manage: Organization Integration
- Process: Expense Report - View
- Process: Purchase Order - View
- Set Up: Expense Item
- Set Up: Spend Categories
- Worker Data: Active Employees
- Worker Data: Current Staffing Information
- Worker Data: Headcount Reports
- Worker Data: Public Worker Reports
- Worker Data: Time Off (Time Off)
- Worker Data: Time Off (Time Off Balances)
- Worker Data: Time Off
- Reports: Time Tracking

   - Manage: Organization Integration
   - Worker Data: Public Worker Reports
   - Set Up: Spend Categories
   - Worker Data: Headcount Reports
   - Business Process Administration
   - Business Process Reporting
   - Set Up: Expense Item
   - Process: Purchase Order – View
   - Process: Expense Report – View
   - Worker Data: Active Employees
   - Worker Data: Current Staffing Information
   - Worker Data: Time Off (Time Off Balances)
   - Worker Data: Time Off (Time Off)
   - Worker Data: Time Off
   - Reports: Time Tracking

5. Select **OK**, review, and then select **Done**.

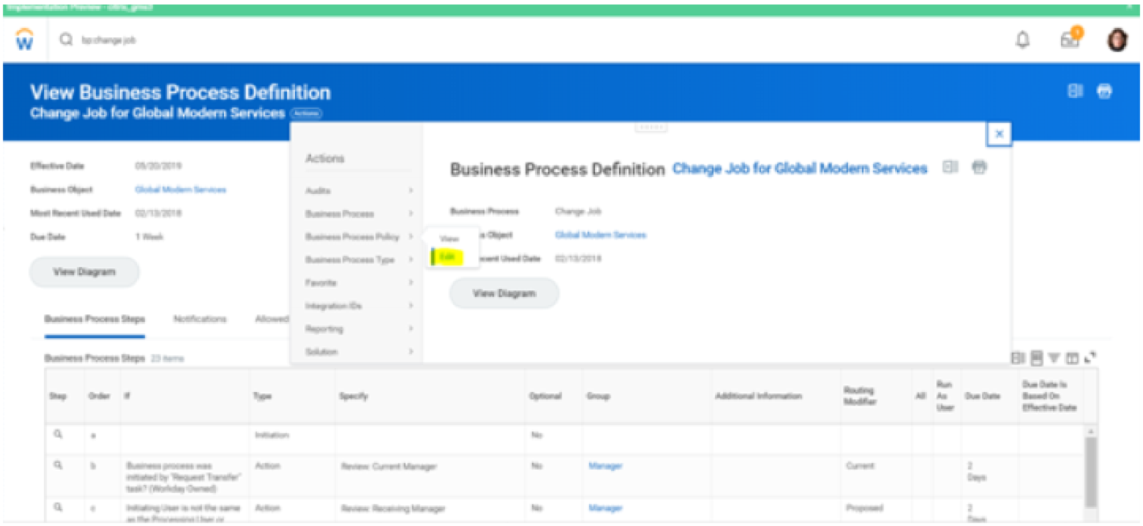
6. Search for *Activate Pending Security Policy Changes*.

7. In the **Comment** field, type *Activate*, and select **OK**.

8. Select the **Confirm** check box, and then select **OK**.
   Permissions have been updated and activated.

Add and activate security group to view all list

1. Search for *bp:change job* and select **Change Job for Global Modern Services**.
2. Go to **Actions > Business Process > Edit**.



Effective Date 05/20/2019

Business Object Global Modern Services

Most Recent Used Date 02/13/2018

Due Date 1 Week

Business Process Change Job

Business Process Steps 23 items

Step	Order	If	Type	Specify	Optional	Group	Additional Information	Routing Modifier	All	Run As User	Due Date	Due Date is Based On Effective Date
🔍	a		Initiation		No							
🔍	b	Business process was initiated by 'Request Transfer' task? (Workday Owned)	Action	Review Current Manager	No	Manager		Current			2 Days	
🔍	c	Initiating User is not the same as the Processing User or	Action	Review Receiving Manager	No	Manager		Proposed			2 Days	

3. Scroll to the section **View All**. Add your security group to the list of security groups that can view all.
4. Select **OK**, and then **Done**.
5. Search for *Activate Pending Security Policy Changes*.
6. In the **Comment** field, type *Activate*, and select **OK**.
7. Select the **Confirm** check box, and then select **OK**.
   Permissions have been updated and activated.

You can view your security group by searching for *View security group* and selecting your security group. There are now two tabs, **Domain Security Policy Permissions** and **Business Process Security Policy Permissions**.

Find your base URL

You need to enter the base URL (domain) for your Workday environment to enable API calls. The format is `https://{ domain } .workday.com`. For example, if the **Workday REST API Endpoint** is `https://wd2-impl-services1.workday.com/ccx/api/v1/citrix_gms`, your base URL is `wd2-impl-services1`. Follow the following procedure to identify your base URL.

1. Log in to Workday as an admin and search for *View API Clients*.
2. Look at the first field called **Workday REST API Endpoint**. The format is `https://{ domain } .workday.com/ccx/api/v1/{ tenant }`.

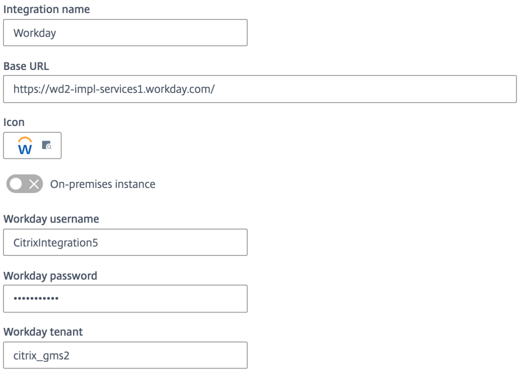
Add the Workday integration to Citrix Workspace Microapps

Add the Workday integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps to set up the Workday HTTP integration. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the **Workday** tile from the **Integrations** category of the catalog.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**. This is the domain for your Workday environment, for example `wd2-impl-services1`. For more information about identifying your base URL, see Find your base URL.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default Workday icon.
    - Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.



Integration name
Workday

Base URL
https://wd2-impl-services1.workday.com/

Icon

○ On-premises instance

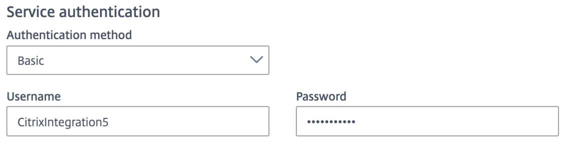
Workday username
CitrixIntegration5

Workday password
•••••••••••

Workday tenant
citrix_gms2

5. Enter your **Workday username**.
6. Enter your **Workday password**.
7. Enter your **Workday tenant**. For more information about identifying your tenant, see Find your base URL.

8. Under **Service authentication**, select **Basic** from the **Authentication method** menu and complete the authentication details.
9. Enter your **Username** and **Password**.



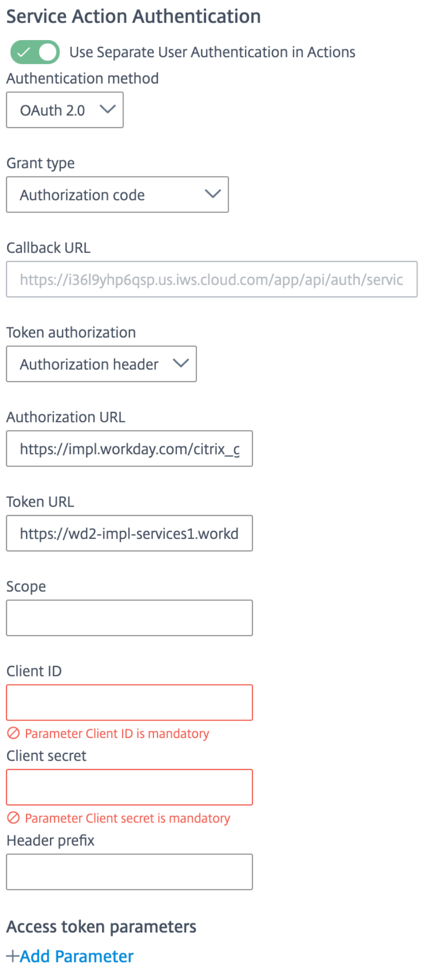
Service authentication

Authentication method: Basic

Username: CitrixIntegration5

Password: •••••••••••

10. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level.
    a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
    b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This will display the **Callback URL**, which you use when registering your application
    c) Select **Authorization header** from the **Token authorization** menu.
    d) Enter your **Authorization URL**. This is the format: `{ instance_url } /{ tenant } /authorize`. This is the authorization server URL provided when setting up the target application integration. For more information about identifying your base URL and tenant, see Find your base URL.
    e) Enter your **Token URL**. This is the format: `{ base_url } .workday.com/ccx/oauth2 /{ tenant } /token`. This is the URL of the access authorization token.
    f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Google account. You need to add the **Callback URL** you see on the integration configuration page.
    g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service Action Authentication

✓ Use Separate User Authentication in Actions

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Authorization header

Authorization URL: https://impl.workday.com/citrix_c

Token URL: https://wd2-impl-services1.workd

Scope:

Client ID:

⊘ Parameter Client ID is mandatory

Client secret:

⊘ Parameter Client secret is mandatory

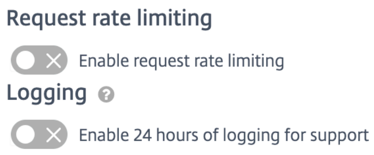
Header prefix:

Access token parameters

+Add Parameter

11. (Optional) If you want to activate rate limiting for this integration, enable the **Request rate limiting** toggle and set the **Number of requests** per **Time interval**.

12. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.



Request rate limiting

✕ Enable request rate limiting

Logging

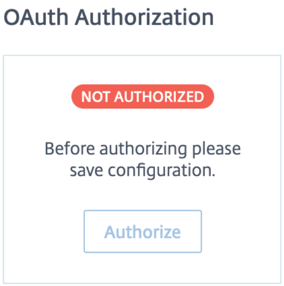
✕ Enable 24 hours of logging for support

13. Select **Save** to proceed.

14. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Workday login screen.

    a) Enter your Service Account user name and password and select **Log in**.

    b) Select **Accept**.



OAuth Authorization

NOT AUTHORIZED

Before authorizing please save configuration.

Authorize

Replace Data Loading and Service Action variables

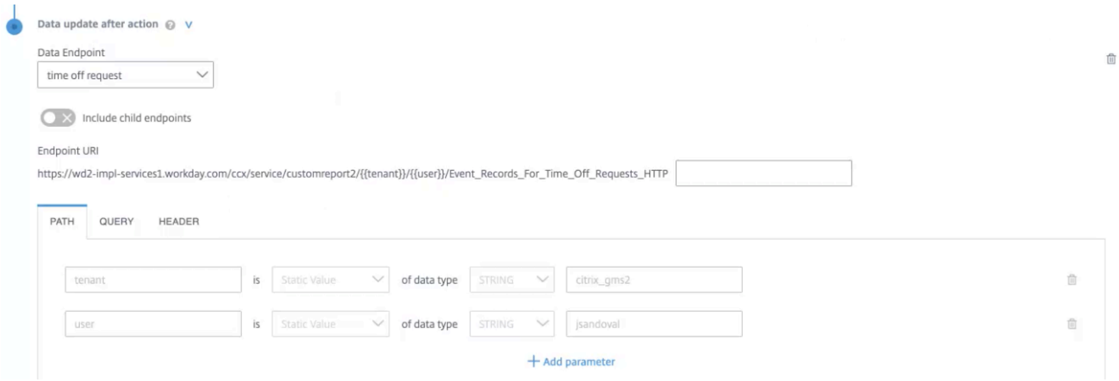
To complete this set up, you need to replace the `{ tenant }`and `{ user }` variables in the Workday integration configuration with your tenant and the user credentials that you use for authentication.

1. From the **Microapp Integrations** page, select the menu next to the Workday integration, and then **Edit**. The **Data Loading** screen opens. If not, select **Data Loading** from the left side navigation column.

2. For each data endpoint, you must manually add your details for: `{ tenant } /{ user }`. You do this six times. Select the menu next to the endpoint and **Edit**.

3. In the **Edit Data Endpoint** screen, under the **PATH** tab add your **tenant** and **user** to the empty fields.

4. Select **Apply** and confirm. Repeat for the other five data endpoints.



Data update after action

Data Endpoint: time off request

Include child endpoints

Endpoint URI

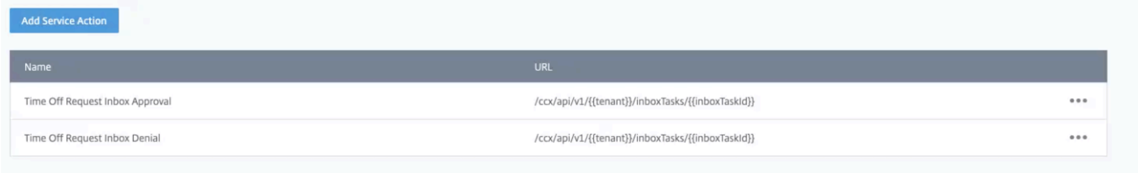
https://wd2-impl-services1.workday.com/ccx/service/customreport2/{{tenant}}/{{user}}/Event_Records_For_Time_Off_Requests_HTTP

PATH QUERY HEADER

tenant is Static Value of data type STRING citrix_gms2

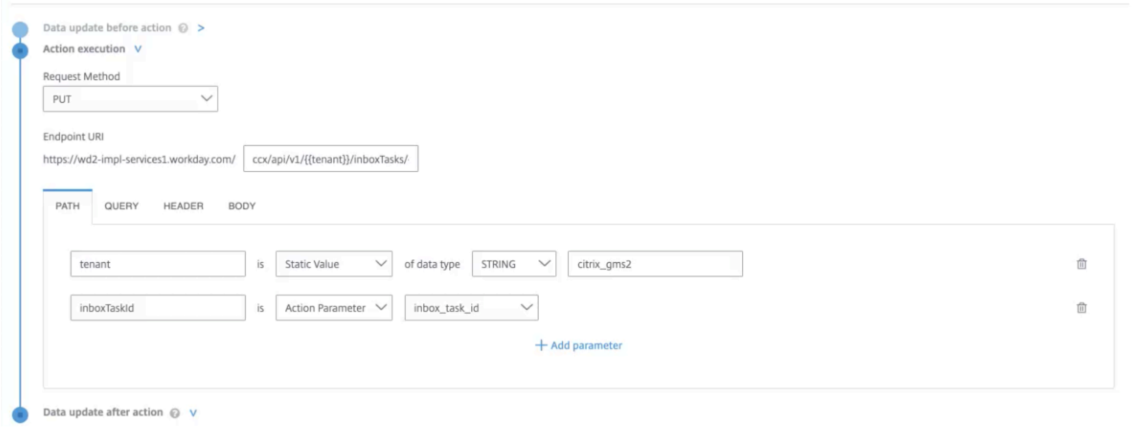
user is Static Value of data type STRING jsandoval

5. For each service action, you must manually add your details for: `{ tenant }`. You do this twice. While editing the integration, select **Service Actions** from the left side navigation column. Select the menu next to one of the service actions and **Edit**.



Name	URL	
Time Off Request Inbox Approval	/ccx/api/v1/{{tenant}}/inboxTasks/{{inboxTaskId}}	•••
Time Off Request Inbox Denial	/ccx/api/v1/{{tenant}}/inboxTasks/{{inboxTaskId}}	•••

6. In the **Edit Service Action** screen, under **Action execution** and the **PATH** tab replace the **tenant** value (**citrix_gms2**) with your tenant.



Data update before action >

Action execution v

Request Method: PUT

Endpoint URI: https://wd2-impl-services1.workday.com/ ccx/api/v1/{{tenant}}/inboxTasks/

PATH | QUERY | HEADER | BODY

tenant is Static Value of data type STRING citrix_gms2

inboxTaskId is Action Parameter inbox_task_id

+ Add parameter

Data update after action v

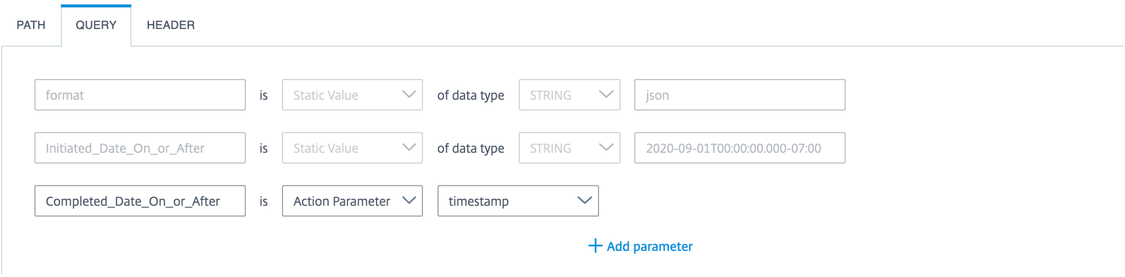
7. You need to delete and recreate **Data update after action**. Next to **Data update after action** label, select the **delete** icon to remove this section.

8. Select **+ Add data update**.

9. Select **time off request** from the **Data Endpoint** menu.

10. Select the **QUERY** tab, and select **+ Add parameter**.

11. Enter *Completed_Date_On_or_After* for **Parameter name**. From the **Choose parameter** menu, select **timestamp**.



PATH | QUERY | HEADER

format is Static Value of data type STRING json

Initiated_Date_On_or_After is Static Value of data type STRING 2020-09-01T00:00:00.000-07:00

Completed_Date_On_or_After is Action Parameter timestamp

+ Add parameter

12. Select **Save** to finish. Repeat for the other service action.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Workday HTTP connector specifications.

Use Workday microapps

Our Workday HTTP integration template comes with the following preconfigured out-of-the-box microapps.

Create Time Off Request: Submit a paid time-off (PTO) request.

Notification or Page	Use-case workflows
Request Time Off page	Provides a form for creating a paid time-off (PTO) request including choosing type, start and end dates, and optionally adding comments.

Change Job: View and approve change job requests.

Notification or Page	Use-case workflows
New Change Job Request for Approval notification	When a new change job approval request is submitted, approver receives a notification.
Change Job Approval page	Provides an actionable form with a detailed view of a change job request.
Change Job for Approval page	Provides a read only view of an approver's change job requests awaiting approval.

Create Change Job: Create a change job request.

Notification or Page	Use-case workflows
Create Change Job page	Provides a page for creating a change job. You must set up deep linking for Citrix Workspace Microapps in Workday.

Create Expense Report: Create an expense report.

Notification or Page	Use-case workflows
Create Expense Report page	Provides a form for creating an expense report. You must set up deep linking for Citrix Workspace Microapps in Workday.

My Time Off Request: View a personalized list of time-off requests.

Notification or Page	Use-case workflows
Time Off Request Status Updated notification	When the status of a PTO request changes, the owner of the PTO request receives a notification.
My Time Offs page	Provides a read only view of a user's active time-off requests including when submitted and its status.
Time Off Detail page	Provides a detailed view of all of a user's time-off requests.

Time Off Requests: View and approve paid time-off (PTO) requests.

Notification or Page	Use-case workflows
New Time Off for Approval notification	When a new time-off approval request is submitted, the approver receives a notification.
Time Off Requests Approval page	Provides an actionable form with a detailed view of a time-off request.
Time Off Requests for Approval page	Provides a read only view of an approver's time-off requests awaiting approval.

Add the legacy integration

Follow these instructions to set up the legacy integration.

Create custom reports

You create custom reports in Workday for each of the following reports. Download the attached spreadsheets and complete the details precisely as described in the spreadsheet. If any detail you enter differs from the provided report, the process does not work. For example, if you use an incorrect name, the report is generated, but no data is downloaded.

The following reports are the Legacy reports. When we add new end-points to this integration, they are added here in the product documentation.

1. Open one of the following custom report spreadsheets:
   - Time off request details
   - Event records for time off requests
   - Event records for change job
   - Event records for expenses
   - Event records for milestones
   - Staffing activities
   - Time off types per plan

   > **Note:**
   >
   > Do not add self-referencing objects to the Time off types per plan report.
2. Follow the process above.

Generate custom report URLs

To download data and generate notifications for the custom reports, you must generate the custom report URLs and download the WSDL. You need these URLs to complete the Microapps set-up procedure.

1. In Workday, open a custom report that you created.
2. Select **Actions** and select the name of the report to open settings.
3. Select **Actions > Web Service > View URLs** > WSDLs.
4. Enter any date in **Entered On** field.
5. Right-click on the WSDL link under Workday XML section and select **Copy URL**. (Example URL: https://wd2-impl-services1.workday.com/ccx/service/customreport2/company_tenant/user_name/report

Save the URL for use when you set up the microapp in the Add the integration to Citrix Workspace Microapps procedure.

Filter PTO Types

Restrict PTO microapps to specific time-off types, such as vacation, or sick leave. If you created the following custom reports in Workday, you need to create calculated fields in Workday and filter custom reports using the following procedures.

- Event Records for time off requests
- Time off request details
- Time off types per plan

Create calculated fields for PTO Requests microapp

You need to create two calculated fields. The first calculated field is intended to retrieve the Absence Type from the Time Off Event so it is available in the second calculated field. The second calculated field is used as the filter in the custom report.

1. Log in to your Workday instance and search for *create calculated field*.
2. Complete all fields:
    - **Field Name** Enter a name for the calculated field. For example, *SANDOVAL CF LRV Absence Type* from this format: {(report creator)(CF=calculated field)(LRV=lookup related value)(values this field returns)}.
    - **Business Object** Find and select **Time Off Event**. We want our calculated field to be part of this business object.
    - **Function** Find and select **Lookup Related Value**.
3. Select **OK** at bottom left.
4. Under the **Calculation** tab, complete these fields:
    - **Lookup field** Find and select **Time Off Event**.
    - **Return Value** Find and select **Time Off Types for Time Off Event**
5. Select **OK** and then **Done**.

You created a calculated field. Now, let's create the second field that retrieves the Time Off Event and will be used as the filter.

1. Again, from your Workday instance search for *create calculated field*.
2. Complete all fields:
    - **Field Name** Enter a name for the calculated field. For example, *SANDOVAL CF LRV Time Off Event for Business Process* from this format: {(report creator)(CF=calculated field)(LRV=lookup related value)(values this field returns)}.
    - **Business Object** Find and select **Action Event**. We want our calculated field to be part of this business object.
    - **Function** Find and select **Lookup Related Value**.
3. Select **OK** at bottom left.
4. Under the **Calculation** tab, complete these fields:

    - **Lookup field** Find and select **Time Off Event**.
    - **Return Value** Find and select the first calculated field you created. In our example, **SANDOVAL CF LRV Absence Type**.
5. Select **OK** and then **Done**.

Filter a custom report using calculated field for PTO Requests microapp

Add a filter to the custom report related to PTO approval using the new calculated field that you just created to whitelist PTO Types, which will allow you to get all PTO types you select.

1. From your Workday instance search for *Edit Custom Report* and select a custom report related to PTO approvals, in our case **Event Records for Time Off Requests**.
2. Select **Filter** tab and **+** to add new filters.
3. Complete these fields:
    - **And/Or** And
    - **Field** Find and select the second calculated field you created, in our example **SANDOVAL CF LRV Time Off Event for Business Process**.
    - **Operator** Find and select **any in the selection list**.
    - **Comparison Type** Find and select **Value specified in this filter**.
    - **Comparison Value** Find and select the PTO types that you want to whitelist, for example:
        - **Annual Leave (Days)**
        - **Annual Leave (Statutory)**
        - **Personal Leave (Days)**
        - **Sick (Days)**
        - **Time Off**
4. Select **OK** to save.

> **Note:**
>
> To block PTO Types, change field: ***Operator** to **none of the selection list**. This blocks the integration from getting any selected PTO types.

Filter a custom report using an existing field for Create PTO Requests and My PTO Requests microapps

Filter Time off types per plan custom report.

> **Note:**
>
> We recommend using the same PTO types in all custom reports. This way if you decide to whitelist or block the outcome, the values in **Comparison Value** is the same for all custom reports).

1. From your Workday instance search for *Edit Custom Report* and select a custom report related to PTO approvals, in our case **Time off types per plan**.
2. Select **Filter** tab and **+** to add new filters.
3. Complete these fields:
    - **And/Or** And
    - **Field** Find and select the custom report **Time Off Type**.
    - **Operator** Find and select **in the selection list**.
    - **Comparison Type** Find and select **Value specified in this filter**.
    - **Comparison Value** Find and select the PTO types that you want to whitelist, for example:
        - **Annual Leave (Days)**
        - **Annual Leave (Statutory)**
        - **Personal Leave (Days)**
        - **Sick (Days)**
        - **Time Off**
4. Select **OK** to save.

Filter Time off request details custom report.

1. From your Workday instance search for *Edit Custom Report* and select a custom report related to PTO approvals, in our case **Time off request details**.
2. Select **Subfilter** tab and select **+** to add new subfilters, similar to how you added filters.
3. Complete these fields:
    - **And/Or** And
    - **Field** Find and select the custom report **Time Off Type for Time Off Entry**.
    - **Operator** Find and select **in the selection list**.
    - **Comparison Type** Find and select **Value specified in this filter**.
    - **Comparison Value** Find and select the PTO types that you want to whitelist, for example:
        - **Paid Time Off (Days)**
        - **Sick (Days)**
        - **Sick (Hours)**
4. Select **OK** to save.

Set up deep link in Workspace

If you are referred to this article, contact your administrator and request that they set up deep linking for Citrix Workspace Microapps in Workday.

The following use-cases require the corresponding deep links. For example, if your domain is `impl.workday.com` and your tenant is `citrix_gms2v`, then the URL for *Create expense report* is: `https://impl.workday.com/citrix_gms2/d/task/2997$995.htmld`:

- **Create Expense Report** `https://your_domain/your_tenant/d/task/2997$995.htmld`

- **Create Milestone** `https://your_domain/your_tenant/d/task/2998$8704.htmld`
- **Create Change Job** `https://your_domain/your_tenant/d/task/2997$4819.htmld`
- **Approve/Deny Change Job (Change Job Requests)** `https://your_domain/your_tenant/d/unifiedinbox/initialinbox/2998$17139.htmld`

> **Note:**
>
> If you do not have this data stored, log in to your Workday account and copy them from the URL. These deep links must be set up with every newly added integration.

Add integration to Citrix Workspace Microapps

Follow these instructions to set up the legacy Workday integration.

Follow these steps:

1. From the overview page, select **Get Started**.

   The Manage Integrations page opens.

2. Select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
3. Choose the Workday tile.
4. Enter an **Integration name**.
5. Enter the **Connector parameters** that you collected as prerequisites.
   - Enter your **URL** to enable API calls. This is the domain for your Workday environment, for example `wd2-impl-services1`. For more information about identifying your base URL, see Find your base URL.
   - Enter your Workday instance for **Instance URL**. Find your instance domain by logging into your Workday environment, and copying the instance url. For example, `https://impl.workday.com`.
   - Enter your Workday **Tenant** location. Find an example of identifying the Workday tenant in Find your base URL.
   - Enter your **Client ID** and **Client Secret** collected in *Prerequisites* procedure.
   - Enter your **Username** and **Password**.
   - Enter the **Days to load** to set the day limit when loading data.
6. Toggle **Enable Time off module?** to **Yes** if you create time-off requests that Workday calculates and you want the data downloaded and notifications generated based on the data. You collected these URLs and paths in the procedure Generate custom report URLs and Generate custom report path.
   - Enter the custom report URL for **Time off request details report URL**.

- Enter the custom report path for **Time off types per plan report path**.
- Enter the custom report URL for **Event records for time off requests report URL**.

7. Toggle **Enable change job module?** to **Yes** if you create change jobs requests that Workday calculates and you want the data downloaded and notifications generated based on the data. You collected these URLs in the procedure Generate custom report URLs.

    - Enter the custom report URL for **Event records for change job report URL**.
    - Enter the custom report URL for **Staffing activities report URL**.

8. Toggle **Enable expense module?** to **Yes** if you create expense reports that Workday calculates and you want the data downloaded and notifications generated based on the data. You collected these URLs in the procedure Generate custom report URLs.

    - Enter the custom report URL for **Event records for expenses report URL**.

9. Toggle **Enable purchase orders module?** to **Yes** if you create purchase orders and you want the data downloaded and notifications generated based on the data. This uses a public API and does require a custom report.

10. Toggle **Download milestone items?** to **Yes** if you create milestone items and you want the data downloaded and notifications generated based on the data. You collected these paths in the procedure Generate custom report path.

    - Enter the custom report path for **Event records for milestone path**.

11. Select **Add**.

The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

Legacy Workday microapps

Our legacy Workday integration template comes with the following preconfigured out-of-the-box microapps:

Change Job Request: View and approve change job requests.

Notification or Page	Use-case workflows
New Change Job Request for Approval notification	When a new change job approval request is submitted, approver receives a notification.
Change Job Approval page	Provides an actionable form with a detailed view of a change job request.

Notification or Page	Use-case workflows
Change Job for Approval page	Provides a read only view of an approver's change job requests awaiting approval.

Create Change Job: Create a change job request.

Notification or Page	Use-case workflows
Create Change Job page	Provides a page for creating a change job. You must set up deep linking for Citrix Workspace Microapps in Workday.

Create Expense Report: Create an expense report.

Notification or Page	Use-case workflows
Create Expense Report page	Provides a form for creating an expense report. You must set up deep linking for Citrix Workspace Microapps in Workday.

Create Milestone: Create a milestone.

Notification or Page	Use-case workflows
Create Milestone page	Provides a form for creating a milestone. You must set up deep linking for Citrix Workspace Microapps in Workday.

Create PTO Request: Submit a paid time-off (PTO) request.

Notification or Page	Use-case workflows
Request PTO page	Provides a form for creating a paid time-off (PTO) request including choosing type, start and end dates, and optionally adding comments.

Expense Reports: View and approve expense reports.

Notification or Page	Use-case workflows
New Expense Report for Approval notification	When a new expense approval request is submitted, the approver receives a notification.
Expense Report Approval page	Provides an actionable form with a detailed view of an expense report.
Expense Reports for Approval page	Provides a read only view of an approver's expense report requests awaiting approval.

Milestones: View milestone details and receive milestone updates.

Notification or Page	Use-case workflows
Milestone Status Update notification	When a milestone is updated, a worker who the milestone is assigned to receives a notification.
Milestone Detail	Provides a detailed view of all of a user's milestones.
Milestones	Provides a searchable list of a user's milestones.

My Expenses: View a personalized list of expense reports with report details and details of individual expense items.

Notification or Page	Use-case workflows
Expense Status Update notification	When the status of an expense changes, the owner of the expense receives a notification.
Expense Report Detail page	Provides a detailed view of all of a user's expenses.
Expense Report Line Detail page	Provides drill-down view into one of the user's expenses.
My Expenses page	Provides a read only view of a user's expense.

My PTO Request: View a personalized list of time-off requests.

Notification or Page	Use-case workflows
PTO Request Status Updated notification	When the status of a PTO request changes, the owner of the PTO request receives a notification.
My PTO Requests page	Provides a read only view of a user's active time-off requests including when submitted and its status.
PTO Request Detail page	Provides a detailed view of all of a user's time-off requests.

Purchase Orders: View purchase orders with purchase order details.

Notification or Page	Use-case workflows
My Purchase Orders page	Provides a read only view of a user's active purchase orders.
Purchase Order Detail page	Provides a detailed view of all of a user's purchase orders.

PTO Balance: View a personalized list of remaining time-off days.

Notification or Page	Use-case workflows
PTO Balance page	Provides a read only view of a user's remaining time-off days.

PTO Requests: View and approve paid time-off (PTO) requests.

Notification or Page	Use-case workflows
New Time Off for Approval notification	When a new time-off approval request is submitted, the approver receives a notification.
Time Off Requests Approval page	Provides an actionable form with a detailed view of a time-off request.

Notification or Page	Use-case workflows
Time Off Requests for Approval page	Provides a read only view of an approver’s time-off requests awaiting approval.

Integrate Zendesk

February 19, 2021

Integrate with Zendesk to submit and monitor requests from any device, intranet, or messenger.

> **Note:**
>
> We provide two Zendesk integration templates for your use. We recommend using the newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure. For full details of the microapps available in each integration, see Use Zendesk microapps.

Use the following process to enable the Zendesk Integration. Ensure you meet the prerequisites, and get your Client ID and secret token. After you complete this process, your existing level of audit logging persists, including any actions carried out by the use of Citrix Microapps.

For a comprehensive list of out-of-the-box Zendesk microapps, see Use Zendesk microapps.

Review prerequisites

Create a dedicated Zendesk account and use it to set up the Zendesk integration. This account must have a role assigned such as Administrator with full data access privileges. After you set up this integration in Zendesk, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- **Base URL**: The base URL follows this model: `https://{ customer-id } .zendesk.com/`.
- **Customer ID**: Use the customer ID part of the URL, as modeled above, to replace `customer-id` during set up process.
- **Client ID**: The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Zendesk account.
- **Client Secret**: The client secret is a unique string issued when setting up the target application integration.
- **Account**: This is your service account username.
- **Password**: This is your service account password.

The following prerequisites should be met before you begin the integration process:

- Configure Citrix Gateway to support single sign-on for Zendesk so that once users log in they are automatically logged in again without having to enter their credentials a second time. Follow the instructions in Zendesk Single Sign-on Configuration. For more information about configuring SSO, see Citrix Gateway Service.

Zendesk has an option `Enable On-hold status` that allows users to assign a `Hold` status to tickets. Our microapp **Tickets** allows users to view Zendesk tickets with details, and the page **Update ticket** has a field **Status** where `Hold`is an option. If `Hold` is not allowed in your Zendesk instance, you need to remove the `Hold` item from the `Status` list. For more information, see Use Zendesk microapps.

Set up the Zendesk integration

1. Log in to www.zendesk.com with the dedicated user account.
2. Select **Admin** (the settings icon) on the left sidebar. Under **Channels** select **API**.
3. Select the **OAuth Clients** tab, and then select the **+** (plus) icon.
4. Enter a **Client Name** for your app.
5. (Optional) Add a **Description**, **Company**, and **Logo**.
6. Copy the auto-populated **Unique Identifier** value for later use.
7. Set **Redirect URLs** as described below.
8. Select **Save** and **OK**.
9. After the page refreshes, a new pre-populated Secret field appears. Copy the **Secret Token** that is generated for later use. This is not available after you leave this screen.
10. Select **Save**.

You are ready to complete the integration in Citrix Workspace Microapps.

Add callback URLs

Add a custom URL to your instance configuration to grant access to private data and enable OAuth authenticated user actions. The first callback that is listed does not change. The second callback depends on the target application, and can be found in your URL address bar when creating the integration. The section {yourmicroappserverurl} is composed of a tenant part, a region part, and an environment part: https://%7BtenantID%7D.%7Bregion(us/eu/ap-s)%7D.iws.cloud.com.

Log in to Zendesk as an admin and add the following authorized redirect URLs for this integration:

- `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
- `https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback`

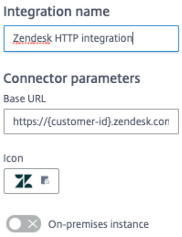
Add the integration to Citrix Workspace Microapps

Follow these steps to set up the Zendesk HTTP integration. We recommend using the newer HTTP integration for most use-cases. The authentication options are preselected. Ensure that these options are selected as you complete the process. We recommend using this newer HTTP integration for most use-cases. The HTTP integration provides more power to configure the cached data structure.

> **Note:**
>
> By default, this integration synchronizes data for a three (3) month time period. We recommend that you modify this value based on your needs and usual age of your tickets. The filter is based on last updated, not created. To change this you must modify the `start_time` variable in a data loading endpoint. See Replace Data Loading variable.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Zendesk tile under **Integrations**..
3. Enter a **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter your **Base URL** or simply replace `{ customer-id }` in the example with your customer ID.
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default Zendesk icon.

    Integration name

    Zendesk HTTP integration

    Connector parameters

    Base URL

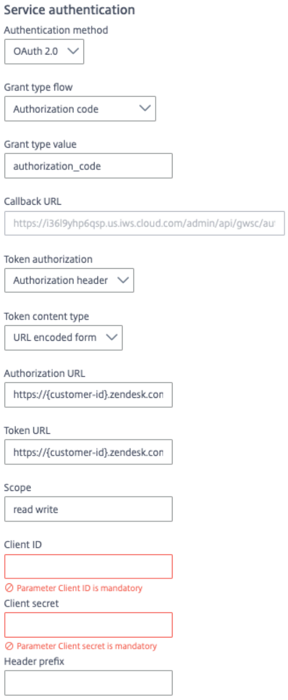
    https://{customer-id}.zendesk.con

    Icon

    - Enable the **On-premises instance** toggle if you are creating an on-premises connection. For more information, see On-premises instance.
5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

    a) Select **Authorization code** from the **Grant type flow** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This display the **Callback URL**, which you use when registering your application

b) Enter **authorization_code** in the **Grant type value** field.
c) Select **Authorization header** from the **Token authorization** menu.
d) Select **URL encoded form** from the **Token content type** menu.
e) Enter your **Authorization URL** or simply replace `{ customer-id }` in the example with your customer ID. This is the authorization server URL provided when setting up the target application integration.
f) Enter your **Token URL** or simply replace `{ customer-id }` in the example with your customer ID. This is the URL of the access authorization token.
g) Ensure *read write* for **Scope** is entered to define the scope of the access request.
h) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Zendesk account. You need to add the **Callback URL** you see on the integration configuration page.
i) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.
j) Enter your **Header prefix**. (optional) Enter the header prefix if your bearer prefix is different from the default header.



Service authentication

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Authorization header

Token content type: URL encoded form

Authorization URL: https://{customer-id}.zendesk.con

Token URL: https://{customer-id}.zendesk.con

Scope: read write

Client ID:

⊘ Parameter Client ID is mandatory

Client secret:

⊘ Parameter Client secret is mandatory

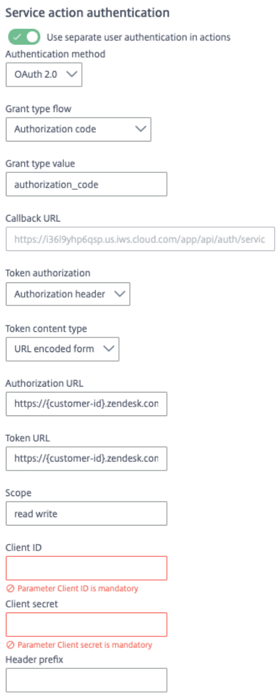
Header prefix:

a) If you selected **OAuth 2.0** authentication method, you can select **+ Add Parameter** to include **Access token parameters**. Access token parameters define the access token parameters as required by the target application authorization server if necessary.

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle Service action authentication authenticates at the service action level. Credentials

are the same as at the Service Authorization level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
b) Select **Authorization code** from the **Grant type flow** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This display the **Callback URL**, which you use when registering your application
c) Enter **authorization_code** in the **Grant type value** field.
d) Select **Authorization header** from the **Token authorization** menu.
e) Select **URL encoded form** from the **Token content type** menu.
f) Enter your **Authorization URL** or simply replace { `customer-id` } in the example with your customer ID. This is the authorization server URL provided when setting up the target application integration.
g) Enter your **Token URL** or simply replace { `customer-id` } in the example with your customer ID. This is the URL of the access authorization token.
h) Ensure *read write* for **Scope** is entered to define the scope of the access request.
i) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret by registering the OAuth client in your Zendesk account. You need to add the **Callback URL** you see on the integration configuration page.
j) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.
k) Enter your **Header prefix**. (optional) Enter the header prefix if your bearer prefix is different from the default header.



Service action authentication

Use separate user authentication in actions

Authentication method: OAuth 2.0

Grant type flow: Authorization code

Grant type value: authorization_code

Callback URL: https://i36l9yhp6qsp.us.iws.cloud.com/app/api/auth/servic

Token authorization: Authorization header

Token content type: URL encoded form

Authorization URL: https://{customer-id}.zendesk.con

Token URL: https://{customer-id}.zendesk.con

Scope: read write

Client ID

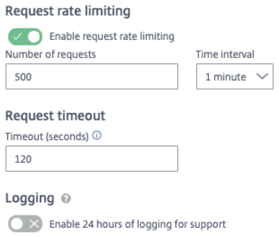
⊘ Parameter Client ID is mandatory

Client secret

⊘ Parameter Client secret is mandatory

Header prefix

7. The **Request rate limiting** toggle is enabled and the **Number of requests** per **Time interval** is set to *500* per minute.
8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.
9. The **Request timeout** field is set to 120 by default.



Request rate limiting

Enable request rate limiting

Number of requests: 500

Time interval: 1 minute

Request timeout

Timeout (seconds): 120

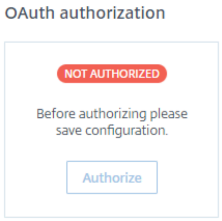
Logging

Enable 24 hours of logging for support

10. Select **Save** to proceed.
11. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Webex login screen.

    a) Enter your Service Account user name and password and select **Log in**.

    b) Select **Accept**.



OAuth authorization

NOT AUTHORIZED

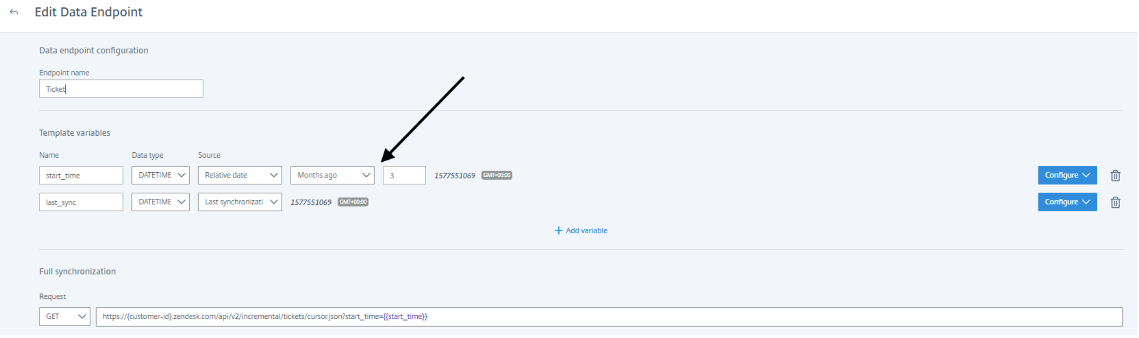
Before authorizing please save configuration.

Continue with the following procedures to finish the set-up process.

Replace Data Loading variable

By default during full synchronization this integration only loads tickets that are modified in the last three (3) months. If you need to change this, modify the `start_time` template variable parameter of the **Ticket** endpoint in this integration data loading setup. We recommend that you modify this value based on your needs and usual age of your tickets. The filter is based on last updated, not created.

1. From the **Microapp Integrations** page, select the menu next to the Zendesk integration, and then **Edit**. The **Data Loading** screen opens. If you are in the configuration screen, select **Data Loading** from the left side navigation column.
2. Select the menu next to the **Ticket** endpoint and then select **Edit**, or select the name of the endpoint: **Ticket**.
3. In the **Edit Data Endpoint** screen, under **Template variables** replace the value for the `start_time` variable with the value that you want changing the time parameter and numerical value, as required.
4. Select **Apply** at the bottom of the screen and confirm.



Edit Data Endpoint

Data endpoint configuration

Endpoint name: Ticket

Template variables

Name	Data type	Source				
start_time	DATETIME	Relative date	Months ago	3	1577551069 GMT+00:00	Configure
last_sync	DATETIME	Last synchronizati	1577551069 GMT+00:00			Configure

Full synchronization

Request: GET https://{customer-id}.zendesk.com/api/v2/incremental/tickets/cursor.json?start_time={{start_time}}

You are now ready to set and run your first data synchronization. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Zendesk connector specifications.

Use Zendesk microapps

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

> **Note:**
>
> We provide two Zendesk integration templates for your use. We recommend using the newer

> HTTP integration for most use-cases over the older java-based integration. The microapps that they contain differ slightly.

Use Zendesk microapps

Our Zendesk HTTP integration comes with the following preconfigured out-of-the-box microapps:

Add Ticket: Submit Zendesk tickets.

Notification or Page	Use-case workflows
Add Ticket page	Provides a form for submitting a ticket.

Tickets: View Zendesk tickets with details. If `Hold` is not allowed in your Zendesk instance, you need to remove the `Hold` item from the `Status` list in the **Update Ticket** page.

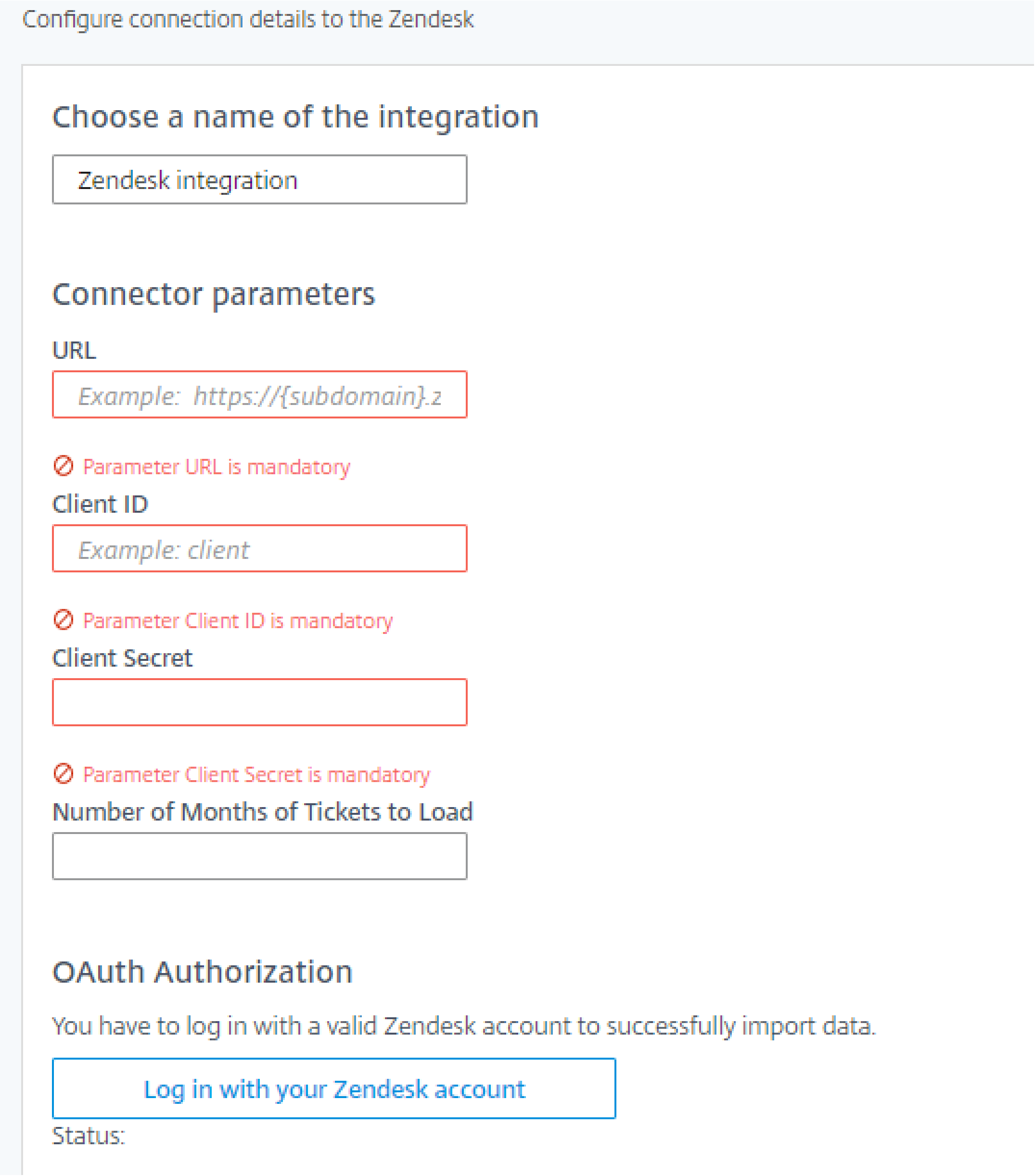
Notification or Page	Use-case workflows
New Ticket Assigned To You (changed) notification	When an existing ticket is assigned to a user, they receive a notification.
New Ticket Assigned To You (new) notification	When a new ticket is assigned to a user, they receive a notification.
Ticket Status Change notification	When the status of a ticket is changed, the submitter of the ticket receives a notification.
Ticket Was Updated notification	When a ticket is updated, the submitter receives a notification.
Add Comment page	Provides a page for adding a comment to a ticket.
Comment Detail page	Provides a read only view of a comment with its details.
My Tickets page	Provides a personalized list of tickets related to a user, and a link to ticket details.
Ticket Detail page	Provides a read only view of a ticket with details.
Update Ticket page	Provides a page for admins to modify tickets. Fields include Priority, Type, and Status.

Add the Zendesk Legacy integration

Follow these instructions to set up the legacy java-based Zendesk integration.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Under **Legacy**, choose the Zendesk tile.
3. Enter a name for the integration.



Configure connection details to the Zendesk

Choose a name of the integration

Zendesk integration

Connector parameters

URL

Example: https://{subdomain}.z

⊘ Parameter URL is mandatory

Client ID

Example: client

⊘ Parameter Client ID is mandatory

Client Secret

⊘ Parameter Client Secret is mandatory

Number of Months of Tickets to Load

OAuth Authorization

You have to log in with a valid Zendesk account to successfully import data.

Log in with your Zendesk account

Status:

4. Enter the **Connector parameters** that you collected as prerequisites.

  - Enter the instance **URL**.
  - Enter the **Client ID**. This value is the Unique Identifier you obtained when you registered your application with Zendesk.
  - Enter the **Client Secret**. This value is the Secret you copied when you registered your application with Zendesk.
  - Select a value for the **Number of Months of Tickets to Load**.

5. Select **Log in with your Zendesk account** to enable OAuth Authorization. A Zendesk sign-in page opens in a new tab. You are prompted to enter an account name, confirm access, and enter a password.
6. Select **Add**.

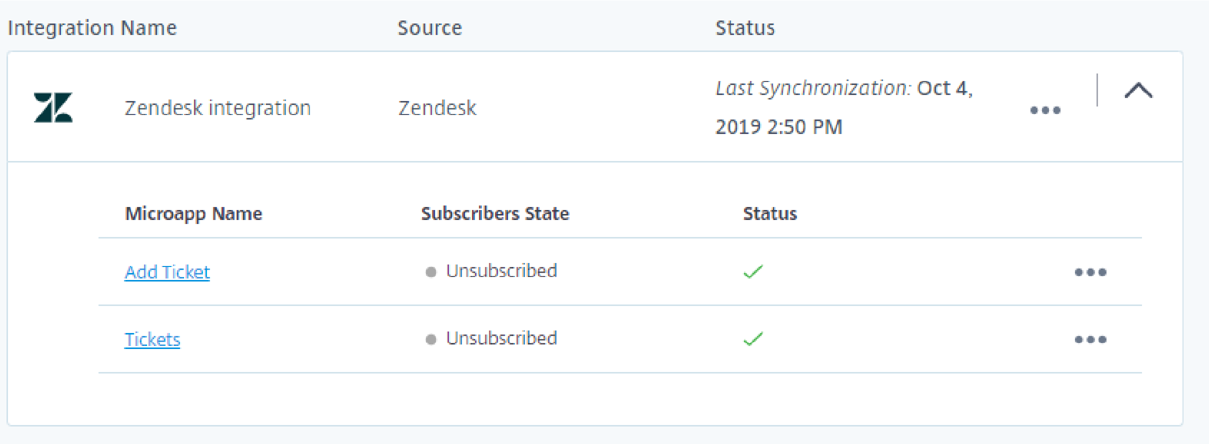
The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Zendesk connector specifications.

Legacy Zendesk microapps

Our java-based Zendesk integration comes with the following preconfigured out-of-the-box microapps:



Integration Name	Source	Status
Zendesk integration	Zendesk	*Last Synchronization:* Oct 4, 2019 2:50 PM

Microapp Name	Subscribers State	Status
Add Ticket	● Unsubscribed	✓
Tickets	● Unsubscribed	✓

Add Ticket: Submit Zendesk tickets.

Notification or Page	Use-case workflows
Submit Ticket page	Provides a form for submitting a ticket.

Tickets: View Zendesk tickets with details.

Notification or Page	Use-case workflows
New Ticket Assigned To You (changed) notification	When an existing ticket is assigned to a user, they receive a notification.
New Ticket Assigned To You (new) notification	When a new ticket is assigned to a user, they receive a notification.
Ticket Status Change notification	When the status of a ticket is changed, the submitter of the ticket receives a notification.
Ticket Was Updated notification	When a ticket is updated, the submitter receives a notification.
My Tickets page	Provides a personalized list of tickets related to a user, and a link to ticket details.
Ticket Detail page	Provides a read only view of a ticket with details.

Integrate Zoom

July 15, 2021

Deploy the Citrix integration for Zoom to schedule meetings from anywhere and from any device or intranet. With our integration for Zoom, users can:

- Create a one-time or recurring meeting, add co-organizers, and select different time zones. The microapp also provides invitation details of the meeting that the user scheduled.
- View, edit, and start created meetings.
- Schedule zoom office hours meetings.
- Receive their meeting recording notification after the meeting ends and play the recordings.

> **Note:**
>
> We want your feedback! Please provide feedback for this integration template as you use it. For any issues, our team will also monitor our dedicated forum on a daily basis.

For comprehensive details of the out-of-the-box microapp for Zoom, see Use microapps for Zoom.

Review prerequisites

These prerequisites assume that the administrator is part of the Zoom integration set up of the organization. This zoom admin account must have full read privileges for user information. After you set up this integration with Zoom, you will need these artifacts to add the integration in Citrix Workspace Microapps:

- Base URL: `https://api.zoom.us/v2/`
- Authorization URL: `https://zoom.us/oauth/authorize`
- Token URL: `https://zoom.us/oauth/token`
- Client ID: The client ID is the string representing client registration information unique to the authorization server.
- Secret: The client secret is a unique string issued when setting up the target application integration.

> **Note:**
>
> We recommend that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum-security compliance with your configured microapp.

Configure Citrix Gateway to support single sign-on for Zoom so that once users log in they are automatically logged in again without having to enter their credentials a second time. For more information about configuring SSO, see Citrix Gateway Service https://docs.citrix.com/en-us/citrix-gateway-service/.

The integration requires regular access to your Zoom instance, so we recommend creating a dedicated user account. You can view the permission/privileges at https://marketplace.zoom.us/docs/api-reference/other-references/privileges. This account must have the following permissions:

- Permissions required for Service Account: Full administrator privileges

The number of API requests that can be made to specific resources is limited, we therefore recommend the following:

- Zoom API limitation form link: https://marketplace.zoom.us/docs/api-reference/rate-limits
- Recommended plan: Business

Create a new service account

Sign in here: https://zoom.us/signin. Refer to the below URL for new Service Accounts: https://marketplace.zoom.us/docs/guides/getting-started

Configure OAuth server

Configure the OAuth server to read data through the Zoom integration.

1. Log in with your service account to: https://marketplace.zoom.us/.
2. Select the **Develop** drop-down menu on the top right.
3. Select **Build App**.
4. Select **Create** for OAuth app, choose app-type as **Account-Level App**.
5. Disable the toggle for publishing the app to marketplace and select **Create**.
6. Enter the following authorized redirect URLs for this integration in the **Redirect URL** field and also the **Whitelist URL** field. Then select **Continue**.
   - `https://{ yourmicroappserverurl } /admin/api/gwsc/auth/serverContext`
7. Complete the required fields then select **Continue**.
8. Under the Scopes section, select **Add Scopes** and select the scopes for **Meeting**, **Recording**, and **User**. Then select **Done**.
9. Select **Install**. A new tab opens to authorize the app then select **Authorize** and close the tab.
10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Authentication** while configuring the integration.

Configure OAuth client

Configure the OAuth client for writing back data through the Zoom integration.

1. Log in with your service account, as above: https://marketplace.zoom.us/.
2. Select the **Develop** drop-down menu on the top right.
3. Select **Build App**.
4. Select **Create** for OAuth app. Choose app-type as **User-Managed App**.
5. Disable the toggle for publishing the app to marketplace and select **Create**.
6. Enter the following authorized redirect URLs for this integration in the **Redirect URL** field and also the **Whitelist URL** field. Then select **Continue**.
   - `<https://{ yourmicroappserverurl } /app/api/auth/serviceAction/callback>`
7. Complete the required fields then select **Continue**.

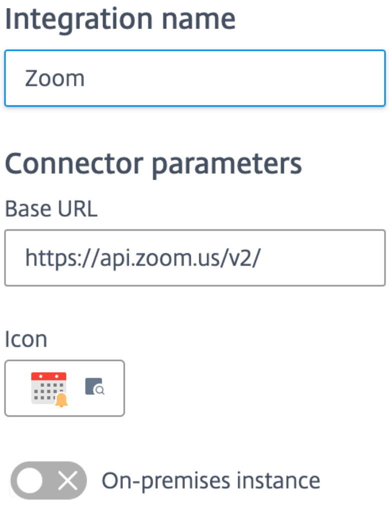
8. Under the Scopes section, select **Add Scopes** and select the scopes for **Meeting**, **Recording**, and **User**. Then select **Done**.
9. Select **Install**. A new tab opens to authorize the app then select **Authorize** and close the tab.
10. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

Add the integration

Add the Zoom Meeting integration to Citrix Workspace Microapps to connect to your application. The authentication options are preselected. Ensure that these options are selected as you complete the process. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

Follow these steps:

1. From the **Microapp Integrations** page, select **Add New Integration**, and **Add a new integration from Citrix-provided templates**.
2. Choose the Zoom Meetings tile.
3. Enter an **Integration name** for the integration.
4. Enter **Connector parameters**.
    - Enter the instance **Base URL**: `https://api.zoom.us/v2/`
    - Select an **Icon** for the integration from the Icon Library, or leave this as the default icon.



Integration name

Zoom

Connector parameters

Base URL

https://api.zoom.us/v2/

Icon

On-premises instance

5. Under **Service authentication**, select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details. The authentication options are preselected. Ensure that these options are selected as you complete the process. Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. It is recommended that you always

use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

a) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.

b) Select **Authorization header** from the **Token authorization** menu.

c) The **Authorization URL** is prefilled: `https://zoom.us/oauth/authorize`

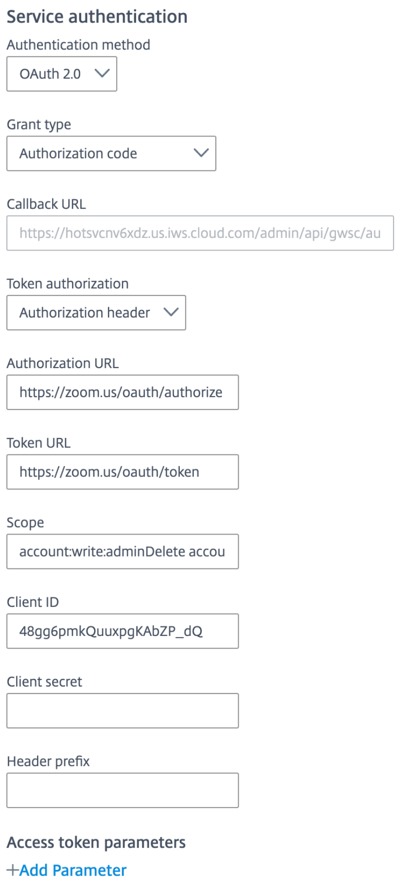
d) The **Token URL** is prefilled: `https://zoom.us/oauth/token`

e) Ensure the following is entered for Scope: *meeting:read:admin,recording:read:admin,user:read:admin meeting:read,meeting:write*

> **Note:**
>
> To make the Create a Meeting microapp active; meeting:read,meeting:write, meeting:read:admin
> To make the Upcoming Meetings (Current Week) microapp active; meeting:read,meeting:write, meeting:read:admin
> To make the My Office Hours microapp active; meeting:read,meeting:write, meeting:read:admin
> To make the Meeting Recordings microapp active; recording:read:admin

f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth server. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service authentication

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://hotsvcnv6xdz.us.iws.cloud.com/admin/api/gwsc/au

Token authorization: Authorization header

Authorization URL: https://zoom.us/oauth/authorize

Token URL: https://zoom.us/oauth/token

Scope: account:write:adminDelete accou

Client ID: 48gg6pmkQuuxpgKAbZP_dQ

Client secret:

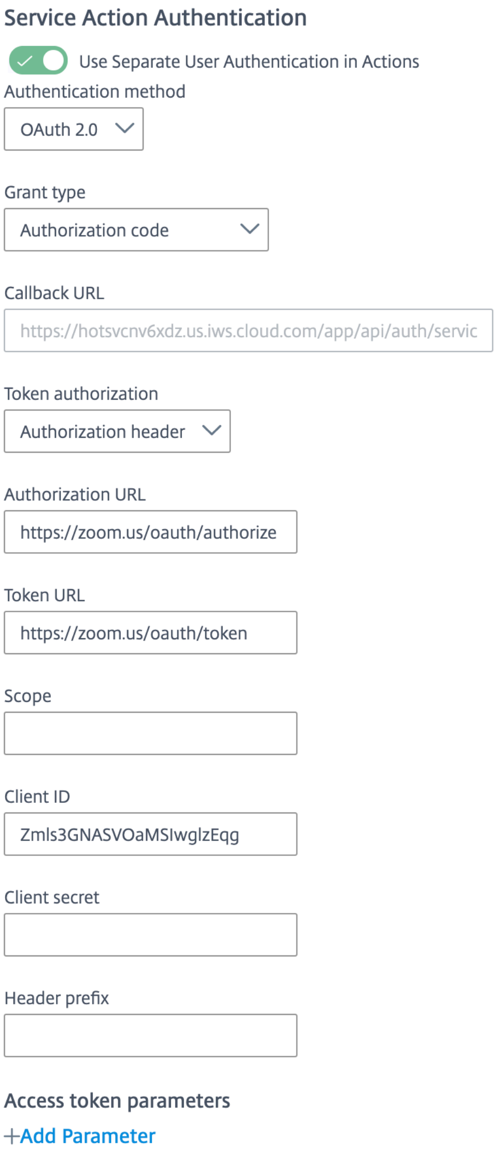
Header prefix:

Access token parameters

6. Under **Service Action Authentication**, enable the **Use Separate User Authentication in Actions** toggle. Service action authentication authenticates at the service action level. The authentication options are preselected. Ensure that these options are selected as you complete the process.

   a) Select **OAuth 2.0** from the **Authentication method** menu and complete the authentication details.
   b) Select **Authorization code** from the **Grant type** menu. This grants a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation. This displays the **Callback URL**, which you use when registering your application.
   c) Select **Authorization header** from the **Token authorization** menu.
   d) The **Authorization URL** is prefilled: `https://zoom.us/oauth/authorize`
   e) The **Token URL** is prefilled: `https://zoom.us/oauth/token`
   f) Enter your **Client ID**. The client ID is the string representing client registration information unique to the authorization server. You collect this and the secret when you configured the OAuth client. You need to add the **Callback URL** you see on the integration configuration page.

g) Enter your **Client secret**. The client secret is a unique string issued when setting up the target application integration.



Service Action Authentication

✓ Use Separate User Authentication in Actions

Authentication method: OAuth 2.0

Grant type: Authorization code

Callback URL: https://hotsvcnv6xdz.us.iws.cloud.com/app/api/auth/servic

Token authorization: Authorization header

Authorization URL: https://zoom.us/oauth/authorize

Token URL: https://zoom.us/oauth/token

Scope:

Client ID: Zmls3GNASVOaMSIwglzEqg

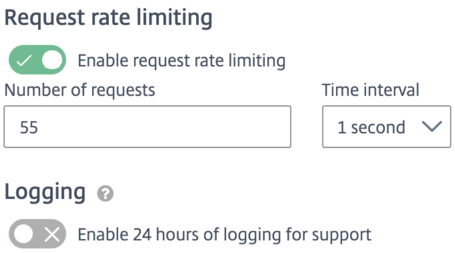
Client secret:

Header prefix:

Access token parameters

+Add Parameter

7. Enable the **Enable request rate limiting** toggle. Enter *55* for **Number of requests** and *1 second* for **Time interval**.



Request rate limiting

✓ Enable request rate limiting

Number of requests: 55

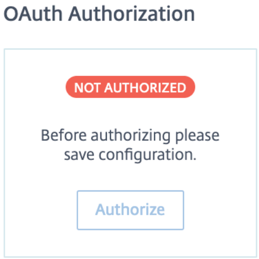
Time interval: 1 second

Logging

✕ Enable 24 hours of logging for support

8. (Optional) Enable **Logging** toggle to keep 24 hours of logging for support purposes.

9. Select **Save** to proceed.
10. Under **OAuth Authorization**, select **Authorize** to log in with your service account. A pop-up appears with a Zoom login screen.
    a) Enter your Service Account user name and password and select **Log in**.
    b) Select **Accept**.



Before authorizing please save configuration.

The **Microapp Integrations** page opens with your added integration and its microapps. From here, you can add another integration, continue setting up your out-of-the-box Microapps, or create a new microapp for this integration.

You are now ready to set and run your first data synchronization. As a large quantity of data can be pulled from your integrated application to the Microapps platform, we recommend you use the **Table** page to filter entities for your first data synchronization to speed up synchronization. For more information, see Verify needed entities. For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

For more details of API endpoints and table entities, see Zoom HTTP connector specifications. For more information about managing access and subscribers, see Assign subscribers. To remove an integration, from the **Microapp Integrations** page select the menu next to the integration that you want to uninstall. Select **Delete integration**, and confirm.

Use microapps for Zoom

Existing application integrations come with out-of-the-box microapps. Start with these microapps and customize them for your needs.

Create a Meeting: Schedule meetings according to your preference. User can choose the meeting title, duration, start date, co-organizers, and so forth.

Notification or Page	Use-case workflows
Create a Meeting page	Provides a form to schedule a meeting with the following details as per the user preference: Meeting Title, Start Time, Duration, Recurrence (once, daily, weekly, monthly), Password, Co-organizers, Dial-In Numbers for the meeting.
New Meetings One Time page	This page provides the success message for the created meetings with a **View Details** button for the One-Time Meetings.
New Meetings Recurring page	This page provides the success message for the created meetings with a **View Detail** button for the Recurring Meetings.
Invitations page	Provides the invitation details for the meeting instantly after the meeting is created, such as Meeting Topic, Password, Calendar Details, Occurrences Details, Join URL, Start Time, Meeting Id, Dial-In Numbers.

Upcoming Meetings (Current Week): View all upcoming meetings for the current week. User can edit and start the meeting.

Notification or Page	Use-case workflows
One Time Meeting Reminder notification	When the start time of the one-time meeting is less than one hour, the host of the meeting receives a notification.
Recurring Meeting Reminder notification	When the start time of the Recurring meeting is less than one hour, the host of the meeting receives a notification.
All Meeting page	Provides the information about all upcoming meetings of one-time and Recurring Meetings for the current week with search option to filter the meetings on the basis of meeting topic.

Notification or Page	Use-case workflows
Meeting Details One Time page	Provides the invitation details for the upcoming one-time meeting, such as Meeting Topic, Password, Calendar Details, Join URL, Start Time, Meeting Id, Dial-In Numbers with the option of Edit and Start Buttons.
Meeting Details Recurring page	Provides the invitation details for the upcoming Recurring meeting, such as Meeting Topic, Password, Calendar Details, Occurrences Details, Join URL, Start Time, Meeting Id, Dial-In Numbers with the option of Edit and Start Buttons.
Edit Meetings One Time page	Provides a form to edit a one-time meeting with the following details as per the user preference: Meeting Title, Start Time, Duration, Time Zone, Recurrence (Onetime, Daily, Weekly, Monthly), Password, Co-organizers, for the meeting.
Edit Recurring Meeting page	Provides a form to edit a Recurring meeting with the following details as per the user preference: Meeting Title, Start Time, Duration, Time Zone, Recurrence (Onetime, Daily, Weekly, and Monthly), Password, Co-organizers, for the meeting. With the option to save this occurrence and save all occurrences.
One Time Meeting Reminder page	Provides the invitation details for the upcoming one-time meeting, such as Meeting Topic, Password, Calendar Details, Join URL, Start Time, Meeting Id, Dial-In Numbers with the option of Start and Close Buttons.
Recurring Meeting Reminder page	Provides the invitation details for the upcoming Recurring meeting, such as Meeting Topic, Password, Calendar Details, Join URL, Start Time, Meeting Id, Dial-In Numbers with the option of Start and Close Buttons.

My Office Hours: Schedule Office Hours meeting according to preferences. User can choose the duration, start date, dial-in numbers, etc.

Notification or Page	Use-case workflows
Virtual Office Hours	Provides the scheduled Office Hours Meeting details in the table for the current month. This page also helps the user in creating the office hours if the user did not schedule any meetings for current month.
Setup Virtual Hours	Provides a form to schedule an Office Hours Meeting with the following details as per the user preference: Start Time, Date, Duration, Time Zone, Recurrence(Onetime, Daily, Weekly, Monthly), Password, Dial-In Numbers for the meeting.
New Meeting	Displays the success message once the meeting is created successfully with the "View Detail" button.
Invitations	Provides the invitation details for the meeting instantly after the meeting is created, such as Meeting Topic, Password, Calendar Details, Occurrences Details, Join URL, Start Time, Meeting Id, Dial-In Numbers.
Edit Office Hours	Helps the user to edit the scheduled Office Hours Meeting as per the user preference for the following fields, Such as Start Time, Duration, Date.

Meeting Recordings: View all the meeting recording for the last seven days. Also allows users to play recordings from any device.

Notification or Page	Use-case workflows
Meeting Recordings notification	When a new meeting recording is available, the host of the meeting receives a notification.
Recording Table page	Provides table to view all the meeting recordings of the host for the last seven days.

Notification or Page	Use-case workflows
Recording Details	Provides the detailed information of the recording, such as Meeting Topic, Date, Time, Download URL with the option of Play Recording Button.

Integration template connector specifications

March 31, 2022

Use connector specifications when you set up a template application integration to use the out-of-the-box microapps or build your own. Before you begin, make sure to review the best practices for configuring application integrations. For a comprehensive list of template integrations and their out-of-the-box microapps, see Set up template integrations.

Connector specification details include:

- API endpoints
- Service actions
- Entities and attributes

The following connector specifications are available for Citrix Microapps template integrations:

- Citrix Cloud Status Hub
- Citrix Podio connector specifications
- Citrix DaaS connector specifications
- Covid-19 Self Certify connector specifications
- Employee Survey App connector specifications
- Adobe Sign connector specifications
- SAP Ariba connector specifications
- Blackboard Learn connector specifications
- Cherwell connector specifications
- SAP Concur connector specifications
- DocuSign connector specifications
- Google Analytics connector specifications
- Google Calendar HTTP connector specifications
- Google Directory HTTP connector specifications
- Google Directory and Google Calendar Legacy connector specifications
- Google Meet connector specifications
- GoToMeeting connector specifications

- Jira HTTP connector specifications
- Jira connector specifications
- MS Dynamics HTTP connector specifications
- MS Dynamics connector specifications
- MS Outlook connector specifications
- MS Teams connector specifications
- Oracle HCM connector specifications
- Podio connector specifications
- Power BI connector specifications
- Power BI HTTP connector specifications
- RSS connector specifications
- Salesforce HTTP connector specifications
- Salesforce connector specifications
- ServiceNow HTTP connector specifications
- ServiceNow connector specifications
- SocialChorus connector specifications
- SAP SuccessFactors connector specifications
- Tableau connector specifications
- Webex connector specifications
- Workday connector specifications
- Workday HTTP connector specifications
- Zendesk connector specifications
- Zoom connector specifications

Export and import integrations and microapps

June 15, 2021

The Microapps service allows you an option to easily export and import integrations and microapps.

With **export** you can:

- Export an integration alone, with all microapps, or with selected microapps.
- Export microapps individually from an existing integration.

With **import** you can:

- Import an integration, with all microapps.
- Import microapps individually from an existing export file in addition to new versions of current microapps.

Benefits

Importing and exporting integrations and microapps can be used for the following scenarios:

- Backup and restore existing integrations and microapps.
- Reduce the time it takes to develop extra microapps with integrations.
- Test new configurations without affecting production integrations.
- Troubleshoot by allowing you to develop safe ways to test proposed solutions.
- Collaborate with other microapps developers within your organization or the broader Citrix Microapp Platform developer community.

Export Feature

The export feature packages the various settings and configurations into a file with a .mapp extension. This file can be imported into the Microapps admin console. There are two types of .mapp files. One for integrations and one for microapps.

> **Note**
>
> No sensitive data is contained in the export file including User IDs, passwords, OAUTH client IDs, and client secrets.

Template Integration .mapp configuration files contain the following:

- Synchronization schedule and configurations
- Tables
  - Edit Schema options
  - Attributes selected
  - Filters and filter queries
- Relationships
- Actions
- Configuration
  - Integration name
  - Connector parameters
    - Service URL
  - Service Authentication
    - User name
    - Password
  - User Authentication Method
  - Other parameters
  - On-premises Configuration
  - Logging

> **Note**
>
> Microapps are exported, but not with any subscribers previously configured. Subscribers must be reconfigured once the microapp is imported. For more information, see Assign subscribers.

HTTP Integration .mapp configuration files contain the following:

- Data Loading
  - Data endpoints (including chained child endpoints)
- Tables
- Relationships
- Service Actions
- Configuration
  - Integration name
  - Connector parameters
    - Base URL
  - Icon
  - On-premises instance
  - Service authentication
    - Authentication method
  - Service Action Authentication
    - Use Separate User Authentication in Actions
    - Authentication method
  - Logging

Microapp .mapp configuration files contain the following:

- Properties
  - Name
  - Description
  - Icon
  - Action
  - (Action page)
- Notifications
  - Name
    - Trigger
  - Toggles
  - Content
    - Action Buttons
  - Target Page
  - Settings
    - Conditions
  - Expiration conditions

- Pages
  - All Page properties and actions
  - All Page formatting
  - All page components and settings
  - All actions called
- Localization
  - All localization settings
- Metadata
  - Identification of the integration that was used to build your microapp.
  - A mapping structure of microapp components to the integration data cache layer must properly map to the new integration.
  - No subscriber settings are exported.

Export a Configuration

To export a configuration file, follow these steps:

1. Open the Microapps management console and locate the integration you want to export.
2. Click the ellipses menu for the integration and select **Export integration**.
3. Input the optional values for the **Vendor** and **Description** fields.
4. Select or deselect the microapps that you want to include in the export file.
5. Select **Export**.
6. Save the resulting .mapp file to a safe location.
   The .mapp configuration file for the integration is exported in the .mapp file format to your local machine.

Export a microapp

To export a microapp file, follow these steps:

1. Open the Microapps management console and locate the integration you want to export the microapp from.
2. Click the ellipses menu for the microapp you want to export and select **Export**.
3. Save the resulting .mapp file to a safe location.
   The .mapp configuration file for the integration is exported in the .mapp file format to your local machine.

Import feature

When importing you integration configurations and microapps consider the following before beginning your export/import workflow:

- What the state of the integration will be after importing.
- Depending on the type of integration exported and the settings that were configured, the integration configuration must be updated.
- After importing, the integration status can show a warning that **Authentication configuration needed**. You will need to configure authentication credentials again for the import to be successful.
- No Syncs, caching, or actions are possible until the Service credentials are updated.

OAuth

When exporting and importing integration and microapps that use OAuth, consider the following before beginning your export/import workflow:

- For integrations with OAuth configured for Service accounts or Service Actions, the integration is exported without client secrets.
- Doing so causes problems for any authentication schemes that use OAuth that can include the Service Authentication scheme and the Service Action Authentication scheme.
- No Syncs, or actions are possible until the Service credentials are updated.
- Reauthentication is required to obtain updated access tokens from the System of Record.

To fill in the OAUTH credentials, follow these steps:

1. From the Microapps admin console, locate the newly imported integration.
2. Click the ellipses menu for the integration and choose Edit.
3. Click Properties from the left
4. Fill in the missing passwords, secrets, and reauthenticate OAuth.

Importing microapps limitations

Microapps are created within integrations. The integration that is the parent to a microapp is called the **source integration**. When you import a microapp, you can import into the same source integration or another integration or **target integration**.
There are significant limitations that must be understood when importing microapps into target integrations.

Known impacts of importing microapps:

- Any existing notifications (aka feed cards) are deleted when the original microapp is deleted.
- New feed cards and push notifications are generated starting with the next sync (full or incremental) of the new integration.
- Microapps can only be imported within a target integration that is the same integration type (Template or HTTP integration) as the source integration.

> **Note**
>
> Even if the underlying data structure (aka schema) is equal for the source and target integrations, the microapp import feature is unable to match the microapp data structure to a different type of integration.

The target integration has a matching database structure to the source integration:

- If there are some cached tables missing in the target integration (the schema is different), the microapp is imported as misconfigured.
- To prevent misconfiguration, make sure the schema of the source and target integrations are equal.
- Navigate through the integration schemas to verify the tables required by the microapp are included in the schema.

Microapp template schema

To view the schema of a template integration, follow these steps:

1. Log in to the Microapps admin console and locate the integration you want to view.
2. Click the ellipses menu and choose Edit.
3. Choose Tables from the left menu and click the button to edit schema.
4. Review the tables and compare the source and target schemas. This ensures that the identical tables and entities are being synced to the microapps data cache.

Microapp status after import

When microapps are imported, the following conditions occur:

- The microapp has no subscribers. Subscribers must be recreated manually.
- There will be no notifications created against this microapp until all subscribers are set and the next synchronization takes place.
- Notifications are generated automatically based on the notification trigger preferences (typically after the next synchronization).

Import Configuration Steps

To import a configuration, follow these steps:

1. Open the Microapps management console and click **Add Integration** at the top of the management console.
2. Select the type of integration you would like to add.
3. Select **Continue** button next to the option to Import a previously configured integration.

4. Drag your integration .mapp file or choose **browse** to select the file from a specific location.
5. If the wrong file was selected, you can choose to remove it by clicking the remove link. Otherwise, click **Import**.
6. The integration is displayed along side all the other integrations in the admin console.

Next steps:

- Add missing credentials to the new integration.
- Add subscribers to the new microapps.
- Delete the original integration on the target environment.

Import a microapp into an existing integration

> **Note**
>
> Microapps contain references to the data structure of the integration that was used to create them. Therefore, microapps must only be imported within a compatible target integration.

To import a new microapp into an existing target integration:

1. Open the Microapps admin console and locate the target integration.
2. Select the ellipses menu for the target integration and choose **Import microapp**.
3. Drag your integration .mapp file or choose **browse** to select the file from a specific location.
4. If the wrong file was selected, you can choose to remove it by clicking the remove link. Otherwise, click **Import**.
5. The microapp is displayed alongside all the other microapps for the integration.

Next steps:

- Add subscribers to the new microapps.

Import a new microapp version

You can update a microapp to a newer version from the microapp option (ellipsis) menu.

1. Select **Import new version** on your desired microapp in the Microapp Integration screen.
2. Drag your new microapp and select **Import**.

   (Optional) Select **Delete existing feed cards** if you want to completely remove the old version of your microapp from the system. If you do not select this option, your old microapp remains on the system marked with and end-of-life (EOL) flag. Your newer version is set as the active microapp. It is recommended you do **not** delete your old microapps to keep your created feed cards working correctly.
3. Click **Import**

Your new microapp is imported.

Next steps:

- Add subscribers to the new microapps.
- End of Life (EOL): You can set a microapp for end of life manually. The EOL toggle is found by clicking to edit the microapp and choosing Properties.

Upgrade an integration

To upgrade an integration, follow these steps:

1. Open the Microapps management console and click **Upgrade integration** at the top of the management console.
2. Select the type of integration you would like to add.
3. Drag your integration .mapp file or choose **browse** to select the file from a specific location.
4. If the wrong file was selected, you can choose to remove it by clicking the remove link. Otherwise, click **Upgrade**.
5. The integration is displayed along side all the other integrations in the admin console.

Upgrade integration considerations

- Only HTTP integrations are supported.
- Accepted data structures include new tables, new columns in existing tables, and new relationships. No modifications are allowed for:
  - Tables (removal of a table, or changing table names, primary keys).
  - Columns (removal of a column, or changing column names, data types, primary keys, unique constraint, nullable).
  - Relationships (no removal or change at all is possible).
- If parts of the old structure are no longer needed, you can keep the data structures empty, or use scripting to define values.
- No removal of target service actions is allowed. The validation applies to a service action universally unique identifier (UUID) and its definition, including parameters and so on.
  - If changes to service actions are required you must configure those service actions as new ones and update each microapp to call the updated service action.

When an integration upgrade succeeds, the following are fully replaced:

- All data end points and webhook definitions.
- Service actions (equal to keeping the old actions while adding the newly configured ones).
- All scripts prepared as part of HTTP integration scripting.

After upgrading:

- A full sync is required to cache the newly included tables and columns. Until successfully synced, the app may not work correctly (due to missing data).
- Only integration entities, relationships, data endpoints, scripts, and service actions are imported and available for the integration upgrade (no properties, authorization, and so on).

Build a custom application integration

July 30, 2020

Integrations extend Citrix Workspace and their microapps provide users with a cutting-edge experience and user interface. Deliver relevant, actionable notifications, combined with intuitive microapp workflows, to make the most important use-cases of business systems and applications directly accessible from a user's Workspace.

Save users time by reducing context switching and eliminating the need to learn how to use various applications for one-off interactions. This improves the user experience because they can focus on their primary responsibilities.

Use the low-code editor to make working with microapps an easy process:

1. Plan the integration by selecting a business app, identifying use-cases, and determining which APIs need to be used.
2. Create the integration by adding the base URL, setting up authentication, and configuring the integration.
3. Create a microapp and add notifications and pages to it.

> **Note:**
>
> If you need a test instance of Citrix Workspace to get started, visit the Citrix Workspace developer portal.

Now let's dig into some details of the journey.

Plan the integration. Select a target business application for integration, identify integration use-cases, and identify APIs.

There are countless applications that can be integrated into Citrix Workspace. Select a target application that holds information of interest to Citrix Workspace users. Of particular interest are applications that are regularly used for quick tasks and are not intuitively accessible to users. Actionable applications enabling users to directly interact from within Citrix Workspace have much more value than applications that simply enable notification of users. For example, approving, creating, adding.

When you're done you have the target application's **Base URL**, the consistent part of your web address that you use for this integration. For example:

`https://app.{ yoursaasapp } .com/api/1.0/workspaces/{ YOUR_WORKSPACE_ID }`

Next, identify key use-cases for the selected target business application that we want to integrate into Citrix Workspace. For example:

- Approve PTO
- Create PTO
- Find pending approvals
- Mark task complete
- Notify user of created or changed assigned tasks

Once use-cases are known, the next step is to identify the APIs that will allow us to extract relevant information from the target system or inject back into it. This step might well involve back-and-forth iterations to the use-case identification because the target system might not provide suitable APIs to implement a use-case. For example:

- API endpoint to approve PTO: PUT `https://my.api.example/pending_pto_approvals/{ id }`
- API endpoint to book PTO: POST `https://my.api.example/pto/`
- API endpoint to get pending approvals: GET `https://my.api.example/pending_pto_approvals/`
- API endpoint to mark task complete: GET `https://app.asana.com/workspaces/{ your.workspace.id } /projects`

Create the integration. Add the base URL, set up authentication, and configure the integration.

You use the target application's **Base URL** you collected in the planning phase.

Select a service authentication type. HTTP integration supports Basic, NTLM, Bearer, and OAuth 2.0 authentication methods.

Now configure your integration. Use the endpoint data you collect in the planning phase. This endpoint data along with service action configuration forms the basis for creating actionable microapps.

Create a microapp. Add notifications and pages.

Build your own microapps to deliver the best end-user experience that meets your needs and streamlines daily workflows. Add a blank microapp to an application integration and then create *pages* or *event notifications* or both.

- **Notifications** are event-driven microapps that automatically notify users when something requires their attention, for example as a card in the Workspace activity feed. For example *New Expense Report for Approval* and *New Course Available for Registration*.

- **Pages** are user-initiated microapps that are available as actions in Workspace and make it easy to do initiate actions. For example, *Request PTO*, *Submit a Help Desk Ticket*, and *Search the Directory*.

That's it. Now let's get started.

Where to go next

Review the next steps in building a custom application integration:

- Plan the integration
- Create HTTP integration
- Configure the integration
- Create microapps

HTTP integration concepts

May 28, 2021

This section covers basic concepts that are used when creating your HTTP integration and is meant as a reference for improving understanding of how the various components work with each other. In depth training on all these concepts and how they relate to Microapps can be found at the Citrix Training Portal. Be aware that a Citrix login is required to access the Citrix Training Portal.

URLs and URIs

A **URL** (Universal Resource Location) is a set of schemes that have specific instruction on how to access a resource over the internet.

The **URL** is basically the address of some service or resource on a network. Every resource that is accessible over HTTP is identified by a URL. These addresses tell our browsers how and where to look for certain resources.

URI (Universal Resource Identifier) sometimes referred to as the Endpoint.
It is similar to the URL we saw earlier but has one key piece added. The name of the resource we want to interact with. This string of characters uniquely identifies a particular resource on the network.
The URI is the combination of the entire base URL (from protocol to directory) with the addition of the actual resource at the end.

The **URL**s and **URI**s of your target application are required when setting up your initial **HTTP integration**.

For more information, see HTTP integration.

HTTP methods

HTTP methods are verbs that represent the actions a client can invoke against the data or resource on the server. Methods are used to run actions against server resources. You can find information on how HTTP integration uses these methods at API request methods

HTTP methods are involved when setting up your Data Loading and Service actions to load and alter your data for your required microapps integrations.

Constructing HTTP requests and responses

HTTP methods include GET, PUT, POST, DELETE, and so on. HTTP requests tells the server what the client wants to do once connected to the resource. For example, the client can view data or GET, create data or PUT, update data or POST, or DELETE data.

The Path contains the location of the resource requested or the URI. This comprises the server host name and resource location on the server of the specific resource requested, aka the URI.

The protocol defines the language of communication the two systems uses to speak, like HTTP/1.1, for instance.

HTTP requests are composed of the following basic structure:

- **Headers**

  The request-header fields allow the client to pass additional information about the request, and about the client itself, to the server.

- **Body**

  The last part of a request is the body, this contains any data to be sent to the server. Not all requests need a body. Only if we are sending data to the server do we need this attribute, like for the POST and PUT methods.

- **Response**

  After receiving and interpreting a request message, a server responds with an HTTP response message. An HTTP response is the data sent back to the client from the server. It provides to the client a representation of the resource requested.

HTTP requests and responses are involved when setting up your HTTP integration, Data Loading and Service actions, and Webhook listeners to load and alter your data for your required microapps integrations.

Pagination

Pagination methods are configured when setting up your Data loading, Webhook listeners and Service Actions. Each pagination method required is dependant on your target application integration.

To learn more about pagination types as used in HTTP integration, see the pagination section in Data Loading.

Validating APIs

There are various third party platforms (for example, Postman) that enable a good sandbox environment to experiment with your APIs. Plenty of information on using these tools is available via the specific program's platform and documentation.

HTTP integration and databases

This section describes basic database concepts used when configuring HTTP integration with your target application integration System of Record (SoR).

Basic Database Structure

Database tables are composed of a set of data elements using a model of named vertical **columns** and horizontal **rows**. Each intersection of a column and row is know as a cell or entity. A database table has a defined number of columns and can have any number of rows. Each row is a record, and represents one instance of an entity. A specific choice of columns which uniquely identify rows is called the **primary key**.

Primary key

A primary key is an attribute or column in a table that contains a unique identifier used to uniquely identify each row or record in the table.
All primary key values must be globally unique within the column and cannot contain a null value. Primary keys reduce data redundancy and help form relationships between data in primary and foreign tables.

The Primary key is configured when setting up your Data Loading and Service actions.

Foreign key

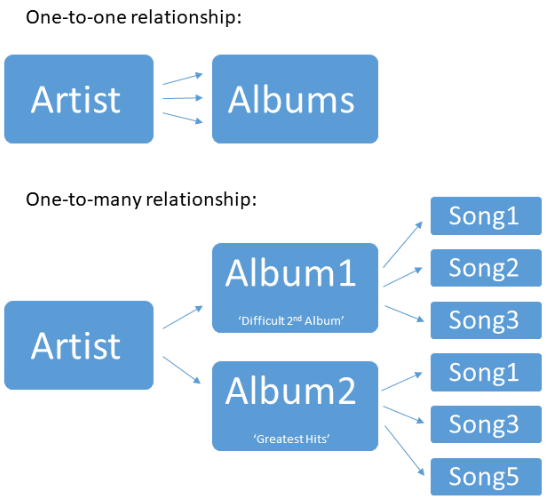
A Foreign key is a column of a table that points to the primary key of another (foreign) table.
Foreign Keys act as a crossreference link between tables and are the basis of how you build **relationships** in your integration data structures. Foreign keys Must Match the primary key in another table or be a null value.

Data Loading and Service actions.

Relationships

Database relationships are associations between tables that are created using join statements to retrieve data from your target application integration.

- **One to One (1:1)**
  A one-to-one table relationship links two tables where the **Primary Key** in the child table is also a **Foreign Key** referencing the primary key in the parent table. Essentially this means that the child table shares the primary key of the parent.
- **One to Many (1:N)**
  A one-to-many relationship in HTTP integration links two tables where a **foreign key** in a child tables links to the **primary key** in the parent table.



Relationships are a central concept when editing your tables to create your microapps and are used when creating Custom relationships.
You can also read more about establish complex relationships using HTTP integration in Create integration data structures in depth.

Data types

Data types are used when constructing your data structure when configuring **Service actions**.

- **string:** An alphanumeric sequence of letters and numbers.
- **integer:** A whole number — can be positive or negative.
- **boolean:** True or false value.
- **object:** Key-value pairs in JSON format.

- **array:** A list of values.

Data types are defined and configured when setting up your Service actions.

SQL queries

Queries are sent to the Microapps Data Cache to return and display the values in microapps pages. Workspace users see data in their feed and page Data is pulled from the Microapps data cache Using SQL queries.

More information on how to show and monitor SQL in your microapps can be found in Page details.

Additional learning

Additional resource for learning about Microapps and Workspace can be found at the Citrix Training Portal. Be aware that a Citrix login is required to access the Citrix Training Portal.

Plan the integration

October 12, 2021

Select a target business application for integration, identify integration use-cases, and identify APIs.

There are countless applications that can be integrated into Citrix Workspace. Select a target application that holds information of interest to Citrix Workspace users. Of particular interest are applications that are regularly used for quick tasks and are not intuitively accessible to users. Actionable applications enabling users to directly interact from within Citrix Workspace have much more value than applications that simply enable notification of users. For example, approving, creating, adding.

Select a target business application for integration

Select a target application that holds information of interest to Citrix Workspace users.

Of particular interest are applications that are regularly used for quick tasks and are not intuitively accessible to users. Also, applications that enable users to directly interact (for example, approve items) from within Citrix Workspace have much more value than applications that simply enable notification of users.

If the target system uses JSON REST and any common authentication mechanism (OAuth 2.0, NTLM, Basic Auth, Bearer Auth), chances are good that the system can be integrated with Citrix Workspace seamlessly. To be able to use HTTP integration with your target integration system of records (SoR), ensure your SoR meets the following prerequisites:

- Your target integration application SOR uses REST API that returns data in the JSON format.
- Your product supports the use of a service account that can access the data of all users and write back on their behalf in the service actions, possibly two separate accounts.
- Your product supports fetching all instances of an object from a single endpoint. For example, all Jira tickets can be fetched via GET /search, whereas O365 requires fetching emails user by user.
- The SOR is populated with representative data (data table autogeneration is done by fetching results and discerning its structure, if the data in the SOR is missing nested JSON fields then tables are not created for them).
- Your product supports one of the following Authorization formats: None, basic, OAuth 2.0, NTLM, or bearer/token authentication. When OAuth 2.0 is available, always use this method as the default to ensure maximum security compliance.
- Your product supports one of the following forms of pagination: none, page, offset, link, header link, cursor, OData.

Identify integration use-cases and identify APIs

Next, we identify key use-cases for the selected target business application that we want to integrate into Citrix Workspace. This activity is a creative process and needs account for:

- The potential time savings that can be achieved by integrating the use-case.
- The effort required to implement the use-case.

Once your use-cases are known, the next step is to identify the APIs that allow us to extract relevant information from the target system or inject back into it. This step might well involve back-and-forth iterations to the use-case identification because the target system might not provide suitable APIs to implement a use-case.

The most common API standard today is RESTful APIs, which provide responses formatted using JSON. Nearly all modern enterprise SaaS applications implement APIs like that.

Where to go next

Now that you have your integration planned, create and then configure the integration:

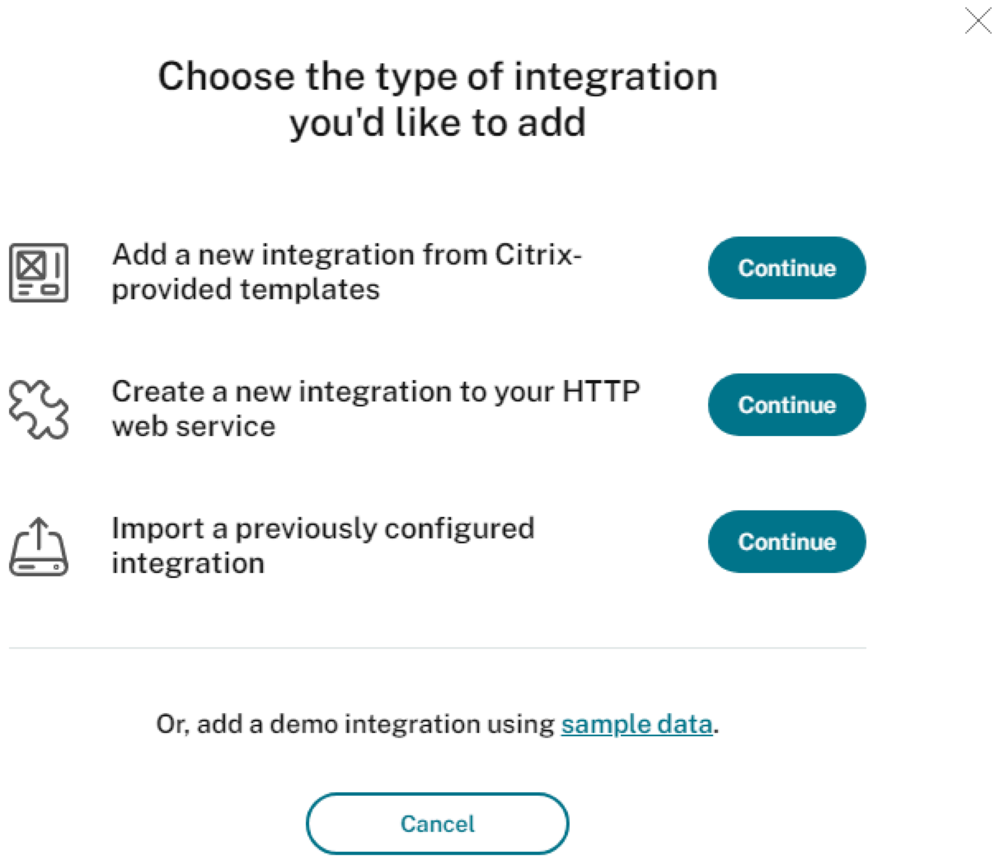
- Create HTTP integration
- Configure the integration

Create HTTP integration

August 24, 2021

Now that you have identified your APIs, let's add an HTTP integration to Microapps service.

1. From the **Microapp Integrations** page, select **Add Integration**.
2. Choose the **Create a new integration to your HTTP web service** to add configuration details.



Choose the type of integration you'd like to add

Add a new integration from Citrix-provided templates

Create a new integration to your HTTP web service

Import a previously configured integration

Or, add a demo integration using sample data.

3. Give your integration a **Name** and enter the Base URL that you collected. The Base URL is the consistent part of your web address that you will use for this integration. For example, `https://app.asana.com/api/1.0/workspaces/${ YOUR_WORKSPACE_ID }`. Replace ${YOUR_WORKSPACE_ID} with your workspace ID (ex. 419224638481718).

   You can add only one Base URL per integration. If you require more Base URLs you must create another integration.

   > **Note:**
   >
   > While HTTP and HTTPS are both permitted as Base URLs for SaaS integrations, HTTP is

> considered a much less secure connection method and it is unlikely that you use it for your target integration application. On-premises integrations do not permit HTTP base URLs.

4. Select an **Icon** to show with your integration. Choose one from the predefined set of icons, or add a custom icon. For details about adding custom icons, see Add custom icons.
5. (Optional) To connect to an on-premises System of Record (SoR), enable the **On-premises instance** toggle. For more information, see On-premises instance.
6. Select your **Service authentication** method and **Service Action Authentication** as required. For more information, see Set up Service Authentication.
7. (Optional) To enable rate limiting for your integration, select the **Request rate limiting** toggle. For more information, see Request rate limiting.
8. Select **Add** at the top-right to save these integration configurations. You now continue configuring the integration. For more information, see Configure the integration.

Add custom icons

You can add custom icons to better identify your integrations. When you publish your HTTP integration to a broader audience, the icons files are uploaded to the Azure CDN storage, and are accessible publicly.

The icon file must conform to these parameters:

- The file is in the png format, with a transparent background.
- The file's resolution must be 128x128 pixels exactly.
- Maximum file size is 80 KB.

> **Note**
>
> Custom icons are for your overview of integrations only. You cannot propagate them to Workspace notifications.

To add an icon, choose **Add an icon**, and select the file that you want to upload.

When you export an integration and then import it to another instance, the icon is added to the list of custom icons at the target instance.

To remove an icon, select an icon from the icons popup, and click **Remove icon**. When you remove an icon, the icon isn't deleted. The integration contains a link to the icon, but you can't select the icon again.

On-premises instance

Microapps service allows you to connect your on-premises System of Record (SoR). On-premises integrations do not permit HTTP base URLs. To create an on-premises connection, first connect using the Connector Appliance then follow this procedure to collect and add the resource identifier id. For more information, see Citrix Cloud Connector Appliance.

1. Go to Citrix Cloud and sign in with your credentials.
2. After signing in to Citrix Cloud, select **Resource Locations** from the top left menu.
3. Find the resource location you want to use and select the **ID** icon below the resource name to reveal the ID of your resource location.
4. Copy the resource location ID.
5. Paste the location ID into the On-premises instance **resource location** field in the **Add HTTP Integration** screen.
6. (Optional) disable **SSL certificate validation** if you require your integration to accept unsigned certificates.

Your on-premises integration is configured.

Set up Service Authentication

When configuring your HTTP Integration service authentication, you must set up your service account with your target application (System of Record). You must also possess both read and write privileges in your target application if you are using the service account to write data to your application. After you have gathered all the necessary information on your target application (login, passwords, security credentials and so on) you can begin the service integration process.

Select your authentication method from the following:

- **None** - No security credentials needed.
- **Basic** - Use your user name and password of the target application for authentication.
- **NTLM** - Configure your HTTP integration to use a suite of Microsoft protocols to connect via New Technology LAN Manager (NTLM) authentication server to authenticate NTLM users via Microsoft Windows credentials.
- **Bearer** - Configure the target integration's authentication scheme to use bearer tokens generated by the server in response to a log-in request.
- **OAuth 2.0** - Use the OAuth 2.0 security protocol to generate request/authorization tokens for delegated access. OAuth 2.0 implementation varies from system to system but the general workflow for OAuth 2.0 works as described below.
- **API Keys** - use the API Keys method to authenticate a user, developer, or calling program to an API.

> **Note:**
>
> It is recommended that you always use OAuth 2.0 as your service authentication method where available. OAuth 2.0 ensures that your integration meets the maximum security compliance with your configured microapp.

Follow these steps:

1. Enter **Service Authentication** parameters for the integration.
2. (Optional) For Authorization Code grant type, select **Log in with your service account** and wait for the login to complete.
3. (Optional) Select the **Service Action Authentication** radio button, and enter authentication parameters at the service action level.

   > **Important:**
   >
   > If you are using delegated permissions, you might not have full access. In this case, use **Service Action Authentication** to authenticate at the service action level. In this situation, you can use basic authentication at the service level, but you must use OAuth 2.0 at the service action level for security reasons.

4. Select **Add**.

OAuth 2.0 Authentication

OAuth 2.0 enables applications to gain specific access to HTTP service user accounts on third-party applications. It works by delegating authentication to the service that contains the user account, and then authorizes third-party applications to access that user account.

OAuth Callback URLs

Callback URLs for authentication follow this pattern:

```
https://{
 customer_id }
 .{
 customer_geo }
 .iws.cloud.com/admin/api/gwsc/auth/serverContext
https://{
 customer_id }
 .{
 customer_geo }
 .iws.cloud.com/app/api/auth/serviceAction/callback
```

The second part of this URL is used only when defining per user authenticated actions. The customer and geographic identifiers are variable and unique to each customer.

OAuth 2.0 Grant Types

HTTP integration allows you to select from four grant types. When setting up Oath 2.0, select your grant type from the menu. When configuring OAuth 2.0 we recommended that you use the Authorization Code as this is the most secure grant flow. Use Client Credential and Resource Owner Password grant types if you require them for additional service action authentication methods:

- **Authorization Code** - Grant a temporary code that the client exchanges for an access token. The code is obtained from the authorization server where you can see the information the client is requesting. Only this grant type enables secure user impersonation.
- **Client Credentials** - Grant type is used to obtain an access token outside of the context of a user. This is used by clients to access their own resources rather than access a user's resources.
- **Resource Owner Password** - Provide the correct credentials to authorize resource server provision of an access token.
- **Implicit Flow** - Implicit grant type is present only for Service action authentication and only in developer mode. You can set response type either to **token** or **id_token**. Automatic access token refresh is not provided. You must provide consent again when the access token expires.

Grant Type Inputs

Depending on the grant type defined above you are provided with the following options to complete to enable OAuth 2.0 authentication:

- **Scope** - Define the scope of the access request, this is a string that is defined by the authorization server when setting up your target integration application.
- **Client ID** - Define the string representing client registration information unique to the authorization server.
- **Client Secret** - Define the unique string issued when setting up the target application integration.
- **Username** - Define the user name of your target application account.
- **Password** - Define the password of your target application account.
- **Authorization URL** - Define the authorization server url provided when setting up the target application integration.
- **Token URL** - Define the URL of the access authorization token.
- **Refresh token URL** - (Optional) Define the refresh token URL of the access authorization token. If not set, the Token URL is used.
- **Access Token Parameters** - Define the access token parameters as required by the target application authorization server if necessary.
- **Log in with your service account** - Log into the service account of the authorization server of your target application.
- **Header Prefix** - (Optional) enter the header prefix if your bearer prefix is different from the default header.

- **Relay state** allows you to configure authentication that enables users to access certain microapps without needing reenter their credentials.

Extra resources regarding OAuth 2.0 can be found at OAuth 2.0's request for questions page.

Relay state

Relay state can only be used if the following conditions are met:

- You use Okta as your identity provider.
- The target SoR supports Okta as its identity provider.
- Relay state is enabled and configured correctly with the correct Okta URL.

After a successful Okta setup, enter the **SingleSignOnService** URL provided by Okta into the Relay state Okta URL field for your integration. For example: `https://{ your Okta } .okta.com/app/{ SoR ID } /{ ID } /sso/saml`.

Relay state works only with user actions and not with full/incremental sync and only passes end user credentials. Some SoRs require end users to confirm a consent page and configuring RelayState does not remove this requirement.

Additional information on configuring Okta can be found in Okta's official documentation. For example you can view how to configure Okta in Salesforce in How to Configure SAML 2.0 for Salesforce.

Troubleshooting OAuth 2.0

If you are having problems connecting your target application to the microapp platform check the following possible solutions errors against your own configuration:

- **invalid_request** - Your authorization request may miss a required parameter, contain an unsupported parameter value (or other grant type), repeat a parameter, include multiple credentials, utilize more than one mechanism to authenticate a client.
- **invalid_client** - Your client authentication failed for the following reasons: unknown client, no client authentication included, or unsupported authentication method. The authorization server may return an HTTP 401 (Unauthorized) status code to indicate which HTTP authentication schemes are supported.
- **invalid_grant** - The authorization grant or refresh token may be invalid, expired, revoked, does not match the redirection URI used in the authorization request was issued to another client.
- **unauthorized_client** - The authenticated client is not authorized to use this authorization grant type.
- **unsupported_grant_type** - The authorization grant type is not supported by the authorization server.
- **invalid_scope** - The requested scope is invalid, unknown, malformed, or exceeds the scope granted by the resource owner.

If you are still having problems configuring OAuth 2.0, check whether you have entered the correct URL for Token and Authorization URL, as these are both unique. Also recheck that your other inputs are correct, such as Scope and so on. If problems persist, check settings on the integrated application server side.

Request rate limiting (optional)

Select the **Request rate limiting** to enable rate limiting for your integration. When toggled, you can define the number of requests and the time interval (1 second or 1 minute) extracted from your target application. Configure rate limiting based on the best practices/rate limits as defined in your target application's documentation.

Where to go next

Now that you have created the HTTP integration, configure the integration:

- Configure the integration

HTTP integration scripting

July 29, 2021

HTTP integration scripting allows you to programmatically script some of your HTTP integration features:

- **Data loading** - your script can define one or more synchronization functions that fetch data from a System of Record (SoR), transform it and store it in the cache.
- **Service actions** - your script can define one or more functions that write data to the SoR, fetch information about a created or updated record and store it in the cache.
- **Webhooks** - your script can define one or more functions that react to data configured and pushed by your application system of record.

Each script defines multiple synchronizations and action functions. HTTP integration then invokes these functions during synchronization execution or when your microapp user invokes an action.

Additional detailed developer resources regarding microapp scripting can be found at the Citrix Developer Portal.

Before you begin

Using the scripting functionality infers that you are familiar with your target application SoR. Use scripting when all other integration methods have been exhausted in configuring your integration.

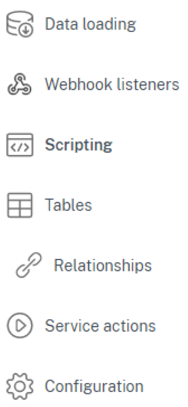
When using scripting for HTTP integrations you must follow this general process:

- Ready your script that you want to import via the Microapp administration interface.
- Scripts must be written in the javascript language edited in your preferred text editor / development tool.
- When ready, import the script via the integrations tab in the Microapps admin interface or optionally you can enter your script directly into the text editor provided in the scripting feature.
- When imported, test the script.

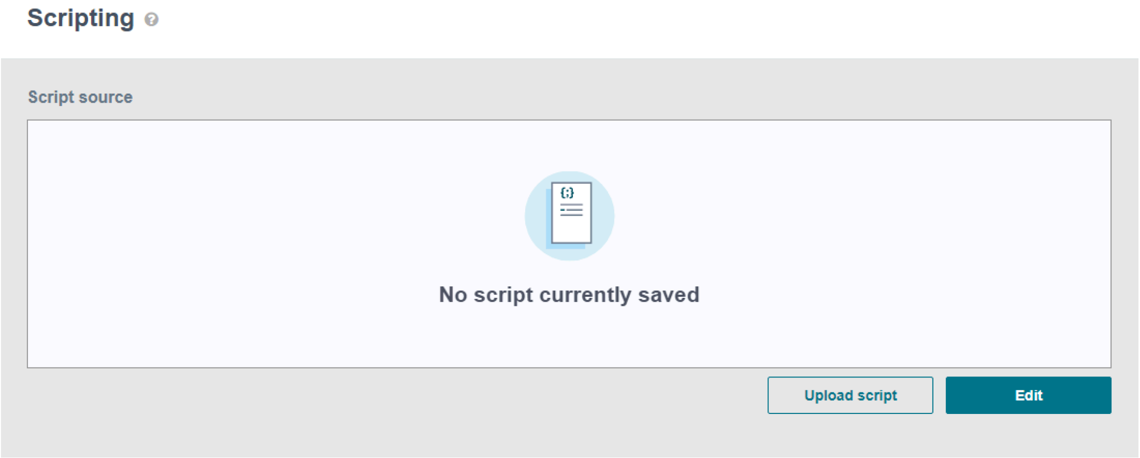
Import a script

To import your prepared script via the integrations tab in the Microapps admin interface.
Follow these steps:

1. Select **Scripting**.



2. Select **Upload script**. Alternatively, you can input your script directly into the text area by selecting **Edit**.



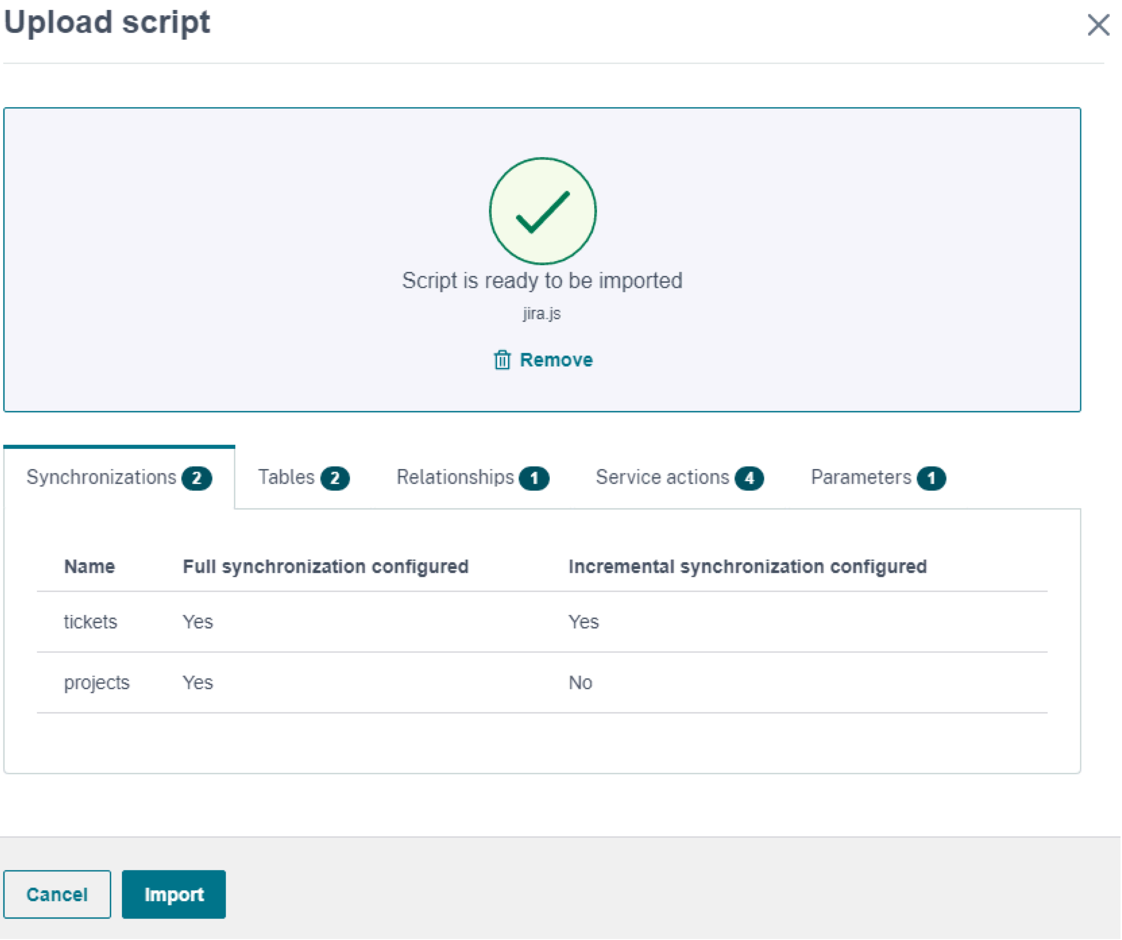
Scripting

Script source

No script currently saved

A blade opens.

3. Drag your script onto the import pop-up.
4. The script is parsed and validated.



Upload script

Script is ready to be imported

jira.js

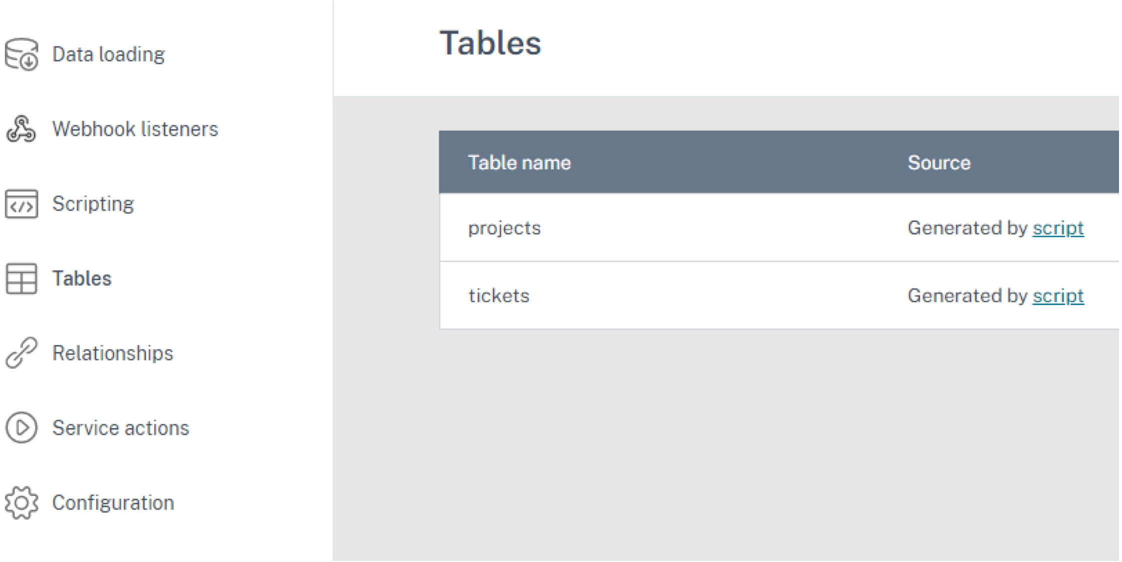
Name	Full synchronization configured	Incremental synchronization configured
tickets	Yes	Yes
projects	Yes	No

5. Select **import**.

6. Your script is imported.

> **Note:**
>
> You can now edit the script directly in the Scripting text editor or update the script by importing the script file again.

- You can view the scripted synchronization as data endpoints in the **Data loading**, **Service actions**, and **Webhook** screens.
- You can view the table defined by the script in the tables page.



Tables

Table name	Source
projects	Generated by script
tickets	Generated by script

- You can view the script output in the log.
- You can see the requests made by the script in the sync log.

To view and monitor your script as it runs go to the logs screen.

FAQ

Authentication - Scripting uses the same client as configured for your System of Record (SoR) integration therefore matches all the same authentication defined in your HTTP integration.

All configuration settings as configured in **Data loading** and **Service actions** for your target SoR are propagated into the scripts.

Scripts, once loaded, are included in microapps import / export (so can be imported exported to the bundle repository).

Custom Integration Parameters

Scripting also supports custom integration parameters for when configuring your HTTP integrations. For example, your integration uses specific application IDs to reference a specific application in a user workspace. This ID is specific to the user and must be set for each integration.

Custom parameter or secrets are defined by Configuration Parameter definition, consisting of:

- name (String, no spaces, no special characters)
- label (String)
- description (String)
- type (String, same as column type)
- default value (populated during import)
- required (boolean)
- secret (boolean) (secrets are never recorded in the Microapps cache or logs).

Other resources

Learn about developing scripts at Citrix Developer Portal.

Get started with developing your own scripts at Getting started with Microapps scripting.

See examples of Microapps scripts at Microapps script SDK.

Get the latest SDK release at Microapps script SDK releases.

Script transformation

October 12, 2021

Script transformation allows you to enable inline script transformation for **Data loading** endpoints and **Service actions**. Scripts can be configured to receive a response object obtained from the HTTP response and transform it to another response object depending on your target integration System of Record (SoR). Scripts cannot perform any requests or store any data. Scripting transformation is only intended to transform the response so that it can be parsed by the HTTP integration JSON parser.

- Each HTTP endpoint has its own editable script.
- When testing an HTTP integration with the **Test Service** button, the script is run along with the test request.
- Scripts run with request transformation are limited to 10000 statements.
- Script execution is also limited by the time period of one minute. If the script does not finish in this period, it is terminated.

Before you begin

Using the script transformation infers that you are familiar with your target application SoR. When using script transformation for HTTP integrations follow this general process:

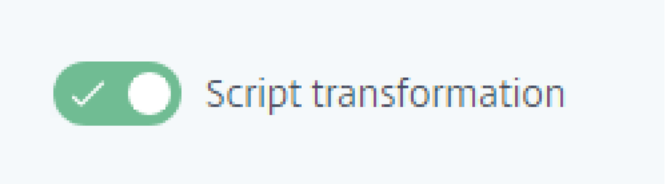
- Ready your script that you want to import via the **Data loading** or **Service actions** configuration pages.
- Scripts must be prepared in the javascript language edited in your preferred text editor / development tool.
- When ready, paste the script (or edit directly) via the script transformation box in the Microapps admin interface.
- When imported, test the script.

> **Note:**
>
> When using script transformation, the standard script output is logged and indexed by Citrix for debugging purposes. You must ensure that no sensitive information is logged when using print, console.log, and so on during configuration.

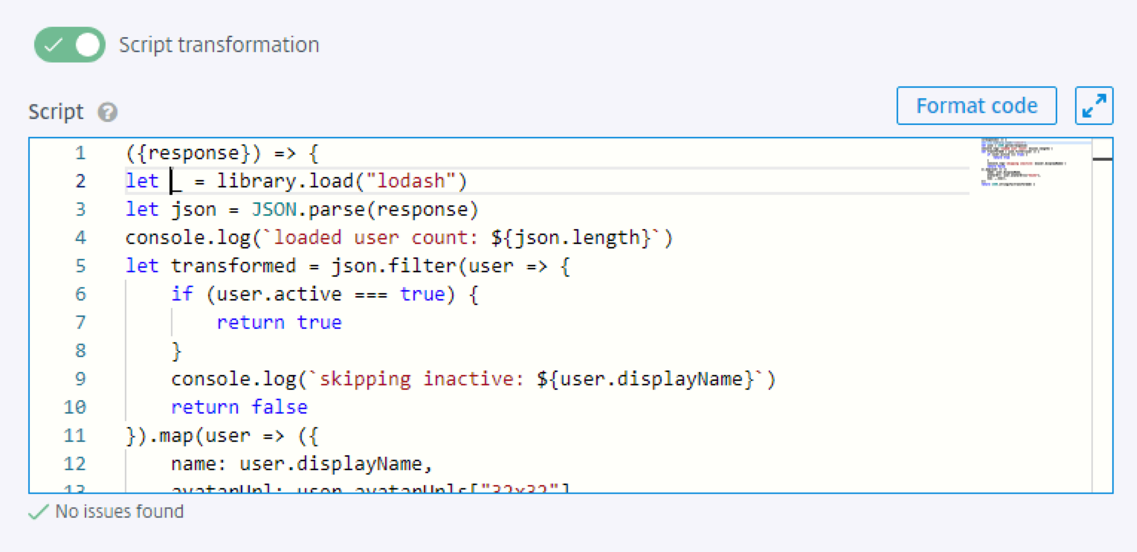
Enable script transformation

To enable script transformation when configuring either your **Data loading** or **Service Actions** follow these steps:

1. Select the **Script transformation** button:

2. Enter your prepared script:



```
({response}) => {
let _ = library.load("lodash")
let json = JSON.parse(response)
console.log(`loaded user count: ${json.length}`)
let transformed = json.filter(user => {
    if (user.active === true) {
        return true
    }
    console.log(`skipping inactive: ${user.displayName}`)
    return false
}).map(user => ({
    name: user.displayName,
    avatarUrl: user.avatarUrls["32x32"]
```

✓ No issues found

3. Select **Test with parameters** followed by **Test service** to see the original response, script output, transformed response, and any script errors if applicable.

> **Note:**
>
> If you change the transformation script, you must regenerate the table as well

When your script is working correctly with your **Data loading** or **Service actions** select **Add**.

Example scripts

```
({
 response }
 ) => {

_ = library.load("lodash")
let json = JSON.parse(response)
console.log(`loaded user count: ${
 json.length }
 `)
let transformed = json.filter(user => {

    if (user.active === true) {

        return true
     }

    console.log(`skipping inactive: ${
 user.displayName }
 `)
```

```
    return false
 }
 ).map(user => ({

    name: user.displayName,
    avatarUrl: user.avatarUrls["32x32"],
    now: _.now(),
 }
 ))
return JSON.stringify(transformed)  }
```

> **Note:**
>
> You can return the string as in the preceding example or alternatively as the js object (array) that would return the transformed variable.

Example: Renaming JSON array name `before.json after.json`:

```
({

response
 }
 ) => {

_ = library.load("lodash")

function rename(obj, key, newKey) {

    if (_.includes(_.keys(obj), key)) {

        obj[newKey] = _.clone(obj[key], true)
        delete obj[key]
     }

    return obj
 }


let json = JSON.parse(response)
let transformed = rename(json, 'tickets', 'new_tickets')
return JSON.stringify(transformed)  }
```

Configure the integration

November 8, 2021

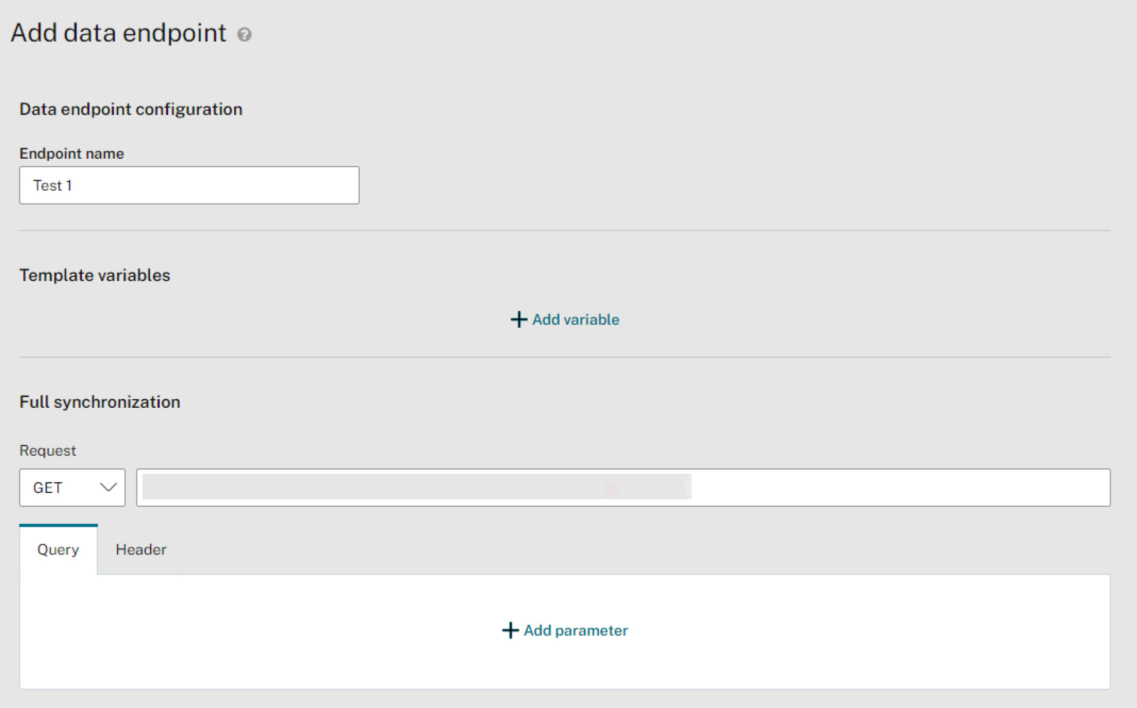
Now that you have added the HTTP integration, configure your integration. This endpoint data along with service action configuration forms the basis for creating actionable microapps.

Add Data Endpoints

Configure the data endpoints to read relevant data into the cache. Any data that we want to show to the user (or want to trigger events or actions with) must be cached.

To add a data endpoint, follow these steps:

1. Select **Data Loading**.



Add data endpoint

Data endpoint configuration

Endpoint name

Test 1

Template variables

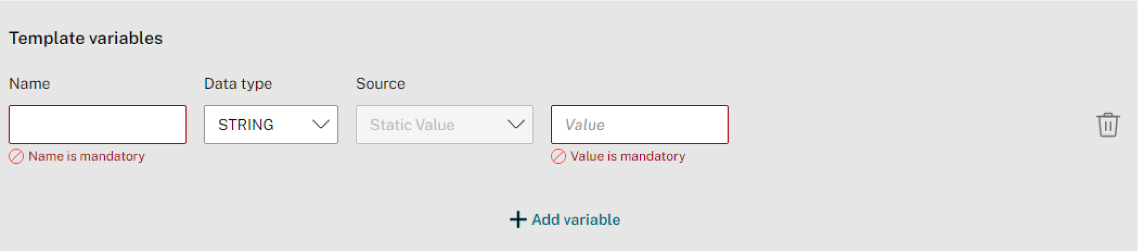
Full synchronization

Request

GET

2. Under **Data Endpoint configuration**, enter **Endpoint Name**.
3. (Optional) Add **Template variables** if necessary. This field provides a dynamic value that is used inside HTTP request definitions. Template variables enable you to override or change all parameters of the original endpoint definition in any of the following:
   - Pre action or post action update
   - Action invocation
   - Incremental sync

For example, you might want to use the dynamic value of the template variable during incremental synchronization.



Template variables

Name	Data type	Source	
Name is mandatory	STRING	Static Value	Value Value is mandatory

a) Select **Add variable**.

b) Enter a name for the new variable.

c) Select a Data Type and **Source**. The data type selected determines the options for source.

   i. Under **DATETIME** data type, for **Static Value**, enter a **Value**.
   ii. Under **DATETIME** data type, for **Relative date**, choose a time period.

d) Select the **Configure** menu, and enter **Format Type** details for **Datetime** if necessary.

e) Select **Save**.

The new variable that you created is now available to be used in HTTP requests. When you enter a variable between mustache tags and the variable does not exist, a pop-up lets you add this variable by selecting **Create variable**.

> **Note:**
>
> Template variable values are percent-encoded following the standard HTTP encoding rules.
> If you do not want this to occur or know that the values are already percent-encoded, you must use the triple-stash mustache tags instead of normal double mustaches to prevent double encoding. For example, change {{example}} to {{{example}}}.

After configuring template variables, you must define the request method, pagination type, and test the service for both **Full synchronization** and **Incremental synchronization** sections as described below.

4. Configure Request method and URL.

5. (Optional) Select **+ ADD PARAMETERS**, and configure **QUERY**, **HEADER**, or request body parameters if necessary.

> **Note:**
>
> The **data type** you select determines the formatting of the attributes. The formatting de-

> termines the fields in microapps. Use mustache tags to reference parameter names. For example, {{parameterName}}.

6. Select **Pagination Type**. The pagination type you must select depends on your target application integration's API standard. Consult your target application integration's API documentation to see what pagination method your application integration uses.

   Our HTTP Integration lets you select from the following standard pagination methods. **Page**, **Offset**, and **Cursor** methods contain a field **Page size value** that defines the number of records per page to pull.

   - **None** - No pagination defined.
   - **Page** - Set the limit of returns per page.

     Example: `https://example.com?limit=100&page=3`

   - **Offset** - Supply two parameters, offset and limit. Offset defines the number of records to skip, limit the number of records to display per page.

     Example: `https://example.com?limit=100&offset=300`

   - **Link** - Define the pagination method to define the next page link in the body.

     Example:

     ```
     {

      “data”  [ … ]
      “next” : https://example.com?lpage=3
      }
     ```

   - **Header Link** - Similar to link pagination, but define the pagination based on the URL page header.

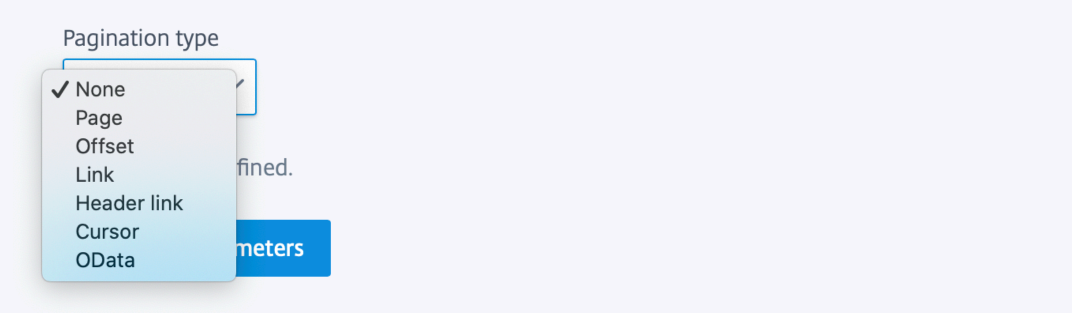
     Example:

     ```
     Link: <https://api.github.com/search/code?q=addClass+user%3Amozilla&page=15>; rel="next",
     <https://api.github.com/search/code?q=addClass+user%3Amozilla&page=34>; rel="last"
     ```

   - **Cursor** - Cursor pagination uses a unique identifier for a specific record that is used as a pointer to the next record to query to return the page of results.

     Example: `https://example.com?paginationToken=BFLMPSVZ`

   - **OData** - Select your OData version to perform pagination to OData standards.



7. Select **Test with parameters** to check that your endpoint is correctly configured. Select a **Number of pages to test** and select **Test with parameters**. Select **Done** to close the blade.
8. (Optional) Toggle **Pagination boundary** if necessary to define conditions for your returned records. This is present only for **page** and **offset** pagination types and depends on the target SoR requirements.
9. Set the **Max pages to load** variable (default 10000, with a maximum of 1000000). Use this variable to limit the volume of records returned from your SoR when required. **Max pages to load** can be configured separately for each endpoint.
10. Select **Test with parameters** to check that your endpoint is correctly configured.

    If the test is successful, continue with the next step. If you receive an error message, troubleshoot based on the error message you receive.

Script transformation

For more information about configuring script transformation, see Script transformation.

Fetch data structure

You can now create your tables in the **Fetch data structure** section.

1. Select **Generate Tables**.
2. Select either **From API** or **From JSON**:

    - **From API** - fetch the data automatically from the defined endpoint.
    - **From JSON** - Use From Sample JSON to paste the API if necessary, for example you have the response but cannot call the API now.
      (Optional) define the **root path** if an alternate root is required from the defined endpoint. The root path must be defined in JSON pointer notation.

3. Select **Generate Tables**.

4. To set your primary key, select the **Edit Attributes** pencil icon and toggle the **Primary Key** for the attribute to be used as the primary key (for example, **id**). Importantly, the primary key column cannot contain null values. Do not change data type to TIMESTAMP.

   > **Note:**
   >
   > When creating a customized integration and microapp, you must always allocate a primary key to enable correct incremental loading rather than a complete overwrite when synchronizing your data.

5. (Optional) Select **Add Table** and configure extra table properties as required by your desired target application and click **Save**. You can then reload the table structure from your API endpoint.
6. **Incremental Synchronization**

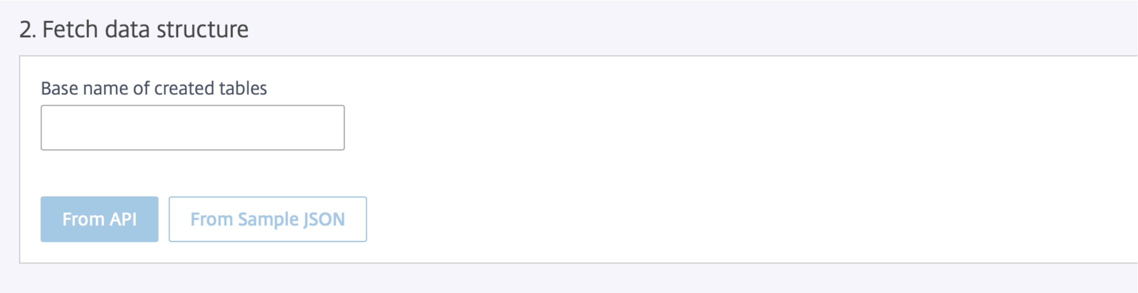
   With **Incremental Synchronization** toggled you can set up Synchronization. That is, to download only updated records since the last data synchronization in more frequent intervals. To do this, configure how often you would like your API call to run. Enter at least one server time parameter. Leaving without setting a synchronization schedule sets the synchronization to manual.

   Custom parameters are only required if the target application requires them. Consult your target application integration documentation when needed. If creating a custom string for synchronization you must enter in square brackets. For example [updated >=] 'YYY-MM-DD HHmm'.

   For more information on synchronization, see **Set data synchronization** below.

7. Input your base name of created tables in the **Fetch data structure** section and select either:

   - **From API** - fetch the data automatically from the defined endpoint.
   - **From Sample JSON** - Use From Sample JSON when to paste the API if necessary, for example you have the response but cannot call the API now.



2. Fetch data structure

Base name of created tables

8. Select **Add**.

Your Data Endpoint is mapped, you can now set up your service actions.

> **Note:**
>
> After setting up and adding your endpoint, your table structure is locked. If you need to restruc-

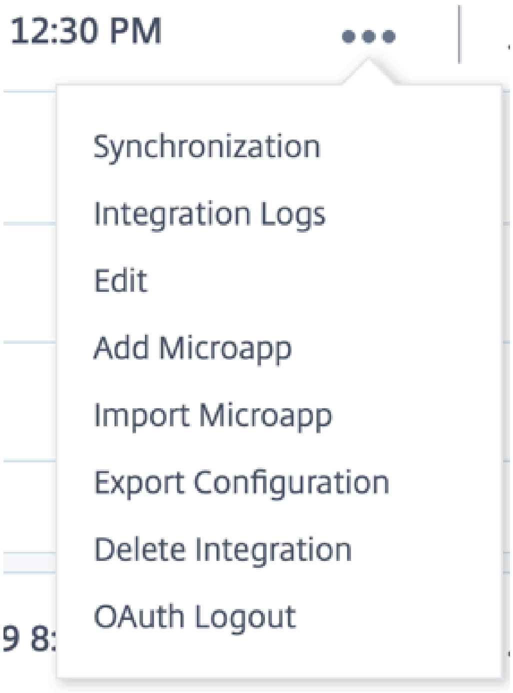
> ture you must create and configure a new data loading endpoint.

Add additional API calls

When configuring data endpoints for your application integration, you can add extra child endpoints to the original parent endpoint to enable call chaining. Once you have set up your data endpoints you can add further associated endpoints.

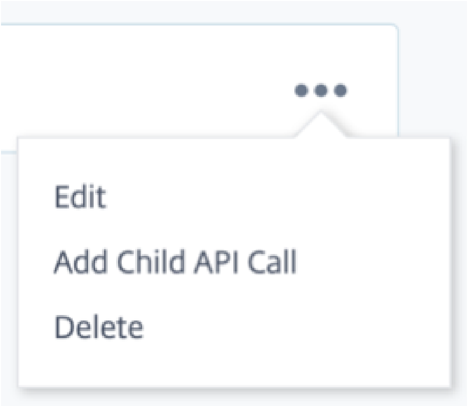
Follow these steps:

1. Select **Edit** on the menu of integration you want to add a child call to.



   The integration page opens.

2. Select **Add Child API Call** from the Data Endpoint menu you want to add:



3. Select the parent table and define the endpoint as you did in the steps in Data Loading section above.

When defining the request method you can either set the path to a static or column value.
Your API call chain is now associated with the parent API call. Your defined parent/child endpoints are now visualized in the Data Endpoint page.

> **Note:**
>
> When creating your integrations it is recommended to load your data from only one endpoint rather than multiple endpoints. Where possible favor batch calls over individual endpoint calls.

Merge tables

When configuring child API calls you can optionally merge child tables with the parent table by selecting one the following options:

Select **Do not merge** if you do not want to merge the parent and child tables.

Select **Merge as detail** to fetch all tasks and requests from the system of record along with every request detail from the request and merge them, for example, if the parent and child tables are:

```
/request-list
{

 “id” : 123,
 “Title” : “Car” ,
 “Role” : “Order” ,
 “Category” : “Sales”
 }
```

And:

```
/request-detail/123
{

 “id” : 123,
 “Title” : “Car” ,
 “Desc” : “Cabriolet” ,
 “Date” : “2020-01-01”
 }
```

The following table is returned after **Merge as detail** is selected:

Id	Title	Role	Category	Description	Date
123	Car	Order	Sales	Cabriolet	2020-01-01

Select **Merge as sublist** to append each child table onto the parent table individually. Using the example above, merging as a sublist results in the following:

Id	Title	Role	Category	id	Title	Desc	Date
123	Car	Order	Sales	123	Car	Cabriolet	2020-01-01

Configuring tables

You can reconfigure table primary keys without setting up a new configuration. To do this, delete the individual table entries in the table screen, resynchronize the table, and select the new primary key.

Supported time formats

HTTP integration supports the following time formats for your system of record data:

- ISO date format
- OData format
- `"yyyy-M-dd HH:mm:ss.SSS"`,
- `"M/d/yy h:mm a"`,
- `"M/d/yyyy h:mm:ss a"`,
- `"dd/MM/yy HH:mm"`,
- `"MMM d, yyyy h:mm:ss a"`,
- `"dd-MMM-yyyy HH:mm:ss"`,
- `"MMMM d, yyyy h:mm:ss a"`,
- `"dd MMMM yyyy HH:mm:ss"`,
- `"EEEE, MMMM d, yyyy h:mm:ss a"`,
- `"EEEE, d MMMM yyyy HH:mm:ss 'o''clock'"`,
- `"h:mm a"`,
- `"HH:mm"`,
- `"h:mm:ss a"`,
- `"HH:mm:ss"`

Add service actions

When you have configured your HTTP integration you can then configure your service actions. With service actions you configure the writeback actions on your application integration's system of records. You configure service actions in a similar manner to the data endpoints. As your application integration can be any number of bespoke combinations we will take a generic approach to explaining how service actions work. You can also optionally configure your service actions to check your target System of Record (SoR) by setting pre and post action execution data updates.

When configuring service action parameters and template variables, the following characters are not supported:

- `Whitespace ! " ## % & ' ( ) * + , . / ; < = > @ [ \ ] ^ { | } ~ true`, `false`, `else`, `null`, `undefined`, `this`.

To add service actions, follow these steps:

1. Select the integration you created under **Application Name**.
2. Select **Service Actions** and **Add New Service Action**.
3. Give it an **Action Name** (such as Get JIRA Ticket Info).
4. (Optional) Select **add parameter** in the **Parameter** section and define your desired parameter **Name**, **Data type** and **Value**.

Define your **Action execution** in the **Action sequence** section:

1. Enter the **Endpoint URI** path: (/rest/api/2/issue/{{issueKey}}).
2. Configure your **Request Method** based on your application integration's API requirements.

   > **Note:**
   >
   > Use mustache tags to reference parameter names. For example, {{parameterName}}.

3. Select **ADD** to save the service action. You can now add extra service actions as required.

Action sequence

You can configure your service action sequence to return data through the action execution and pre and post-action data updates. Pre and post-action data updates are optional and are intended to provide robust checks on the data your microapps is accessing and to ensure your data always accurate when configuring actionable microapps.
(Optional) Configure data update before action execution to ensure that your data is fully synchronized for your microapp end users when they action it. For example, you want to make sure the amount shown on the actionable microapp is the correct amount to approve and that it has not been updated in the time between its creation and its approval.

> **Important:**
>
> This Data update before action execution capability only works with the **Text** component. This means that other data changes before action execution are not displayed to end users. Likewise, if a value is entered for **Page Logic** of the **Text** component, then the check is not run. If Workspace users make a concurrent modification, there is no warning.

Data update before action execution

To set up a data update before execution, follow these steps:

1. Select the existing **Data endpoint** that you want to fetch the updated record from.
2. (Optional) Enable **Include child endpoints** only if the child data endpoints are required to fetch the full detail of the updated record.
3. (Optional) Extend the original **Endpoint URI** if it allows the update to fetch a single record. For example, if the data endpoint URI `https://domain/api/items` is updated to `https://domain/api/items/itemId`. The new endpoint URI must return the same data structure as the original one, otherwise data parsing will fail.

> **Note:**
>
> If the endpoint configuration is changed after this initial setup, changes are not propagated here automatically.
>
> 1. (Optional) Extend the original request parameters with **Add additional parameter** if it enables filtering of a single record.

1. Select **Test with parameters** to check that your endpoint is correctly configured. Select a **Number of pages to test** and select **Test with parameters**. Select **Done** to close the blade.

   If the test is successful, continue with the next step. If you receive an error message, troubleshoot based on the error message you receive.

When finished with your configuration, select **save**.

Data update after action execution

(Optional) To ensure that your data is fully synchronized after action execution, you can configure a data update to fetch fresh data from your target application system of records.

To set up data update after execution, follow these steps:

1. Select the existing **Data endpoint** that you want to fetch the updated record from.

2. (Optional) Enable **Include child endpoints** only if the child data endpoints are required to fetch the full detail of the updated record.

3. (Optional) Extend the original **Endpoint URI** if it allows the update to fetch a single record. For example, if the data endpoint URI `https://domain/api/items` is updated to `https://domain/api/items/itemId`. The new endpoint URI must return the same data structure as the original one, otherwise data parsing will fail.
4. (Optional) Extend the original request parameters with **Add additional parameter** if it enables filtering of a single record.
5. Select **Test with parameters** to check that your endpoint is correctly configured. Select a **Number of pages to test** and select **Test with parameters**. Select **Done** to close the blade.

   If the test is successful, continue with the next step. If you receive an error message, troubleshoot based on the error message you receive.

When finished with your configuration, select **save**.

Verify needed entities

Use **Table** to verify your current list of tables stored in the cache and filters that are applied to those tables.

You are now ready to set and run your first data synchronization unless you need to create custom relationships. For more information, see Set data synchronization.

API request method

You can now configure your Request Method based on your application integration API requirements with the following components:

- **GET** - Retrieve resources from the application integration SOR without modifying.
- **POST** - Create a new resource in the application integration SOR.
- **PUT** - Update existing resources in the application integration.
- **PATCH** - Make a partial update to a resource.
- **DELETE** - Delete a resource.

With following configurable API parameters:

- **Header** - Define parameters included in the request header, related to authorization.
- **Path** - Define parameters within the path of the endpoint, before the query string.
- **Query** - Define parameters in the query string of the endpoint.
- **Body** - Define parameters included in the request body.

Script transformation

For more information about configuring script transformation, see Script transformation.

Create custom relationship

Use theRelationships page to create a custom connection between tables in your integration. You might use this if you have multiple Base URLs and multiple integrations are required, or if you want to create a custom relationship in the same integration. This is an advanced use-case and we recommend you familiarize yourself with creating microapps on a single integration before you start mapping multiple integrations.

1. From the Manage Microapps page, select the menu next to the integration that you want to verify entities for.
2. Select **Edit** and then **Relationships**.
3. Select **Add New Relationship**.

   The Add Relationship Page Opens.
4. You can map your primary table to your foreign table.
5. Enter the alias you want to have.

You can now map and add extra reference columns based on the primary keys you have set up on each of your integrations.

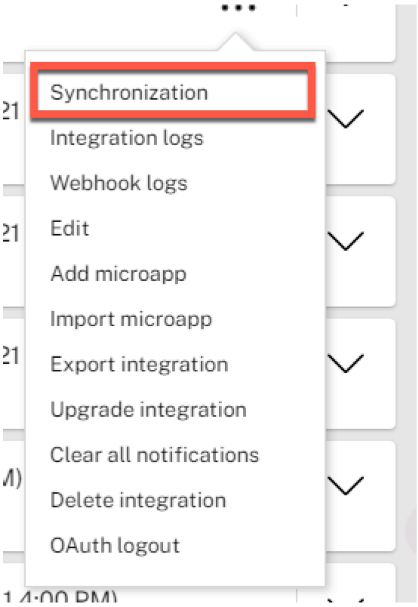
> **Important:**
>
> If you delete a table, all relationships are deleted as well.

Set data synchronization

Pull data from your integrated applications to the Microapps platform so that a comparison can be made to the cache. As a best practice, full synchronization is performed every 24 hours and incremental syncs can be configured to pull every five minutes.

For complete information about synchronization rules, synchronization that does not meet its schedule and veto rules, see Synchronize data.

1. From the Manage Microapps page, select the menu next to the integration for which you want to set synchronization.
2. Select **Synchronization**.



3. Set **Full** and **Incremental** data synchronization values.

    - **Full** Drops the local cache and pulls all data from the source system.

        > **Important:**
        >
        > Running full synchronization can take a long time. We recommend running full synchronization at night or generally during off hours. You can cancel a data synchronization that is in progress at any time by selecting the *X* icon.

    - **Incremental** Pulls only changed (new and updated) records. Does not load deleted data.

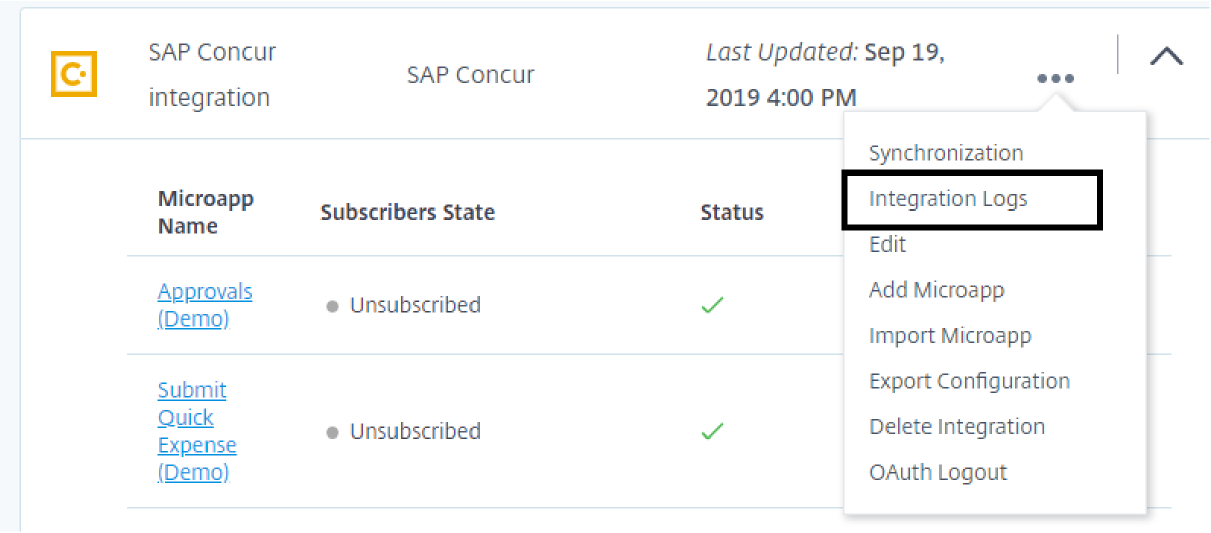
        > **Important:**
        >
        > Not all APIs support incremental synchronization.
        >
        > When you define **daily** or **weekly** synchronization, synchronization occurs randomly within the timeslot you select. For example, selecting 00-04 daily full synchronize will run a full synchronize at a randomly selected time in that period.

4. Select **Save**.

> **Note:**
>
> You can also select the arrow icons to run the integrations on demand if necessary.

Show integration logs

Use **Integration Log** to view a history of changes categorized by severity. Use this for troubleshooting issues with your integration. For example, if you see that the synchronization failed, check the integration logs to see why. Or if the expected cards are not showing, check the integration logs to see if the synchronization occurred.



SAP Concur integration

SAP Concur

Last Updated: Sep 19, 2019 4:00 PM

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
Submit Quick Expense (Demo)	● Unsubscribed	✓

1. From the Manage Microapps page, select the menu next to the integration that you want to view integration logs for.
2. Select **Integration Logs**.
3. Review the entries, and select the menu to filter by Errors if necessary.

Export integration configurations

You can export your integration configurations. All credentials are discarded. This includes passwords and client details. Only the configuration that is stored in the Microapps server is exported. For example, the export keeps your user name but not your Password, and also the export keeps your OAuth configuration but not the client secret.

1. From the Manage Microapps page, select the menu next to the integration that you want to export.
2. Select **Export Configuration**.

   The service.mapp file downloads.

Where to go next

Now that you have created and configured your custom integration, build your own microapps to deliver the best end-user experience that meets your needs and streamlines daily workflows. For more information, see Create microapps.

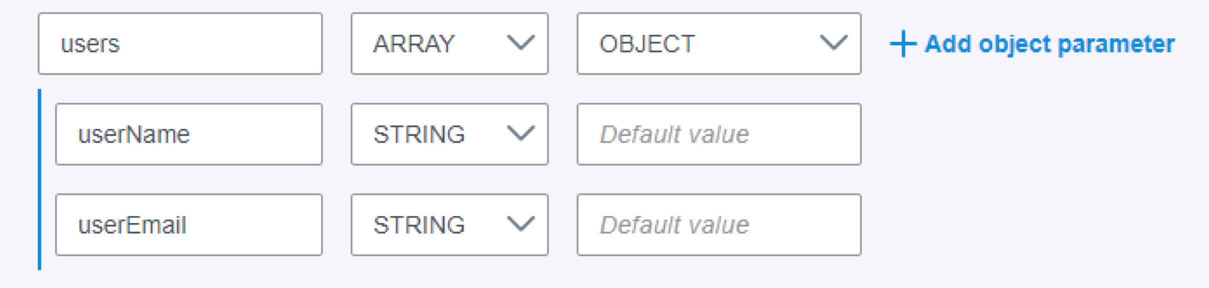
Array data type

November 4, 2021

Use the Array data type to configure multiple returns in your service action's configured JSON body. This is useful when developing your microapps to use the **multiselect lookup** component in **page builder**.

To configure an array data type, follow these steps:

- Enter the name for your **array**.
- Select the **array** data type and choose the subtype for items in the array. Depending on your multi select lookup requirements, the array subtype can be a primitive data type or an object with multiple parameters. Primitive data types result in a list of values, and multiple parameters result in a list of objects with values.
- Use **Add object parameter** and add as many parameters as you require. For example, in the following image, every object in the users array can contain `userName` and `userEmail` parameters:

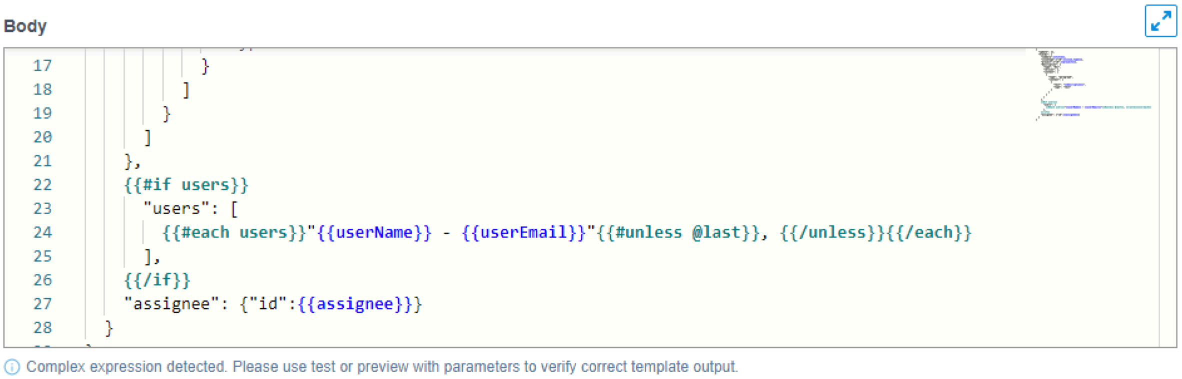


users	ARRAY	OBJECT	+ Add object parameter
userName	STRING	Default value	
userEmail	STRING	Default value	

Configure JSON body

When configuring the JSON body of your array, use mustache tags to reference parameter and template variable names. For example, {{parameterName}}.

For extra configuration options, the JSON body supports handlebars formatting including helpers, data variables as defined in the handlebars format in addition to context changing:



Body

```
                }
              ]
            }
          ]
        },
        {{#if users}}
          "users": [
            {{#each users}}"{{userName}} - {{userEmail}}"{{#unless @last}}, {{/unless}}{{/each}}
          ],
        {{/if}}
        "assignee": {"id":{{assignee}}}
      }
```

ⓘ Complex expression detected. Please use test or preview with parameters to verify correct template output.

You can also test your configured JSON template by using the **test with parameters** feature. This provides feedback on whether your JSON body is configured correctly and valid JSON is generated during action execution.

Mustache tags are either replaced with actual values and enquoted automatically if needed or replaced with **null** during JSON body template processing. It is also possible to enforce value enquoting with the **enquote** Handlebar helper.

Examples:

```
     *     INPUT_JSON       -> ENQUOTED_JSON          -> HANDLEBARS_OUTPUT
     *
     *     // applied enquote helper
     *     {{string}}       -> {{enquote string}}      -> "value"
     *     {{stringNull}}   -> {{enquote stringNull}}  -> null
     *     {{integer}}      -> {{enquote integer}}     -> 42
     *     {{integerNull}}  -> {{enquote integerNull}} -> null
     *
     *     // not applied enquote helper, inside quotes
     *     "{{string}}"     -> "{{string}}"            -> "value"
     *     "{{stringNull}}" -> "{{stringNull}}"        -> ""
     *
     *     // applied enquote with escaping, because of invalid character
     *     {{string!}}      -> {{enquote [string!]}}   -> "value"
     *
```

Array examples:

```
{
   "tickets": [
      {{#each tickets}}
            {
              "id": {{id}},
              "status": {{enquote status}},
              "timestamp": {{enquote created}},
              "tag": {{@root.tag}}
           }
     {{/each}}
   ]
}
```

```
"comments": "{{#each comment}}{{this}}{{#unless @last}}|{{/unless}}{{/each}}"
```

```
"comment": {{enquote comment[1].date}}
```

Create integration data structures in depth

September 30, 2020

When creating microapps you may find the requirement to access tables in your target System of Record that are separated by more than two levels from the parent table. Owing to limitations currently found in the HTTP integration, a solution around this is possible.

This article provides information on how to access tables in your target System of Record when this use case arises. This solution is not straight forward, but if you follow the description below, you can create deeper data structures.

Use case

You want to build a microapp that allows a user to approve a request on ServiceNow. To use this microapp, the user must be able to:

- receive and open a notification
- receive a page with a list of items to approve
- see each item's details
- view who sent the request
- approve these requests

The details needed for building an action or page for each of these steps are stored in tables retrieved via endpoints. However, the table with the data for the approver (the table with data containing the item list) is further than two tables apart from another data locations.

Solution prerequisites

To create this workaround you need to use a combination of child API call-chaining and table merging described in Configure the integration.

Prerequisites:

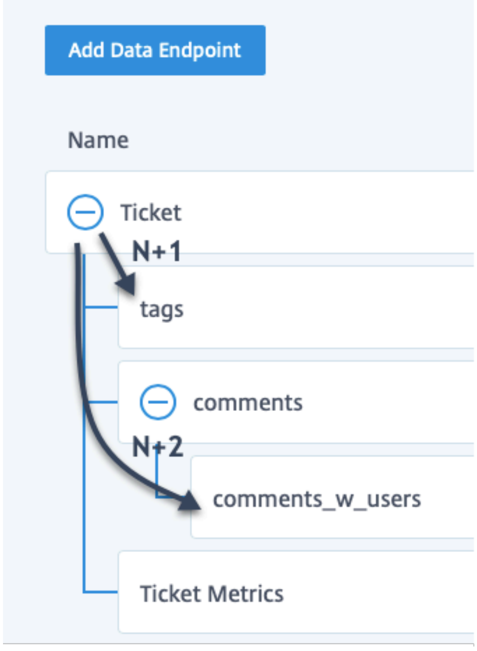
- You have defined your end-to-end use case with the understanding of what must be run in your microapp and what information your end user views and actions.
- You have created the endpoint to return the table data you need from your target System of Record.

> **Note:**
>
> Configured tables and primary keys cannot be edited after initial set-up.
>
> - You have familiarized yourself with the add additional API calls and merge tables features in HTTP integration.

View and build your data structure

When building your microapps, the conventional model supported by the Microapps Platform is for between tables separated only by one step away (N+1 model).

You can see this by checking your integration configuration set up during HTTP integration. For example, you can see that **Ticket** is one step away from **tags**, but neither is directly connected to **comments_w_users**.



Some relationships are created automatically during endpoint configuration, and you see them in the table reference of the integration. However, for this specific use case, you must create some manual definitions to create the relationships between tables.

Data Structure Merge Strategy

When designing the data structure to build your microapps in this scenario, consider the following important points:

- Choose the parent API call depending on the data structure you must achieve to build your microapp. Consider how to use incremental sync for your data set and the API that will return only the updated data structure. This API must be set as the parent.
- Where possible, configure only one-to-many rather than many-to-one. Many-to-one configurations result in repetitive API calls and will impact data sync efficiency.
- Consider the source of the notification you require and how it is configured so that your user will receive only one notification in cases where table merging is configured and data can be duplicated.
- The parent API must always be the most volatile object.

Use the following merge table methods for the specific cases:

- **One-to-one** - Use **Merge as detail**. This results in only one record stored in the database that contains all attributes from the parent and child APIs. The child values are used when the attribute is present in both parent and child API call.
- **One-to-many** - Use **Merge as sublist**. All parent attributes are stored with every child record.
- **Many-to-one** - Usually many-to-one is not a scenario for Child API calls. You must consider the

most suitable method, whether to use table merging or manual setup of the entity relationship (no merging applied). If no merging is applied, only the first child is stored, other children are ignored due to duplicated primary key detection.

Define manual relationships

To define relationships manually, there must be a common column in both tables to use to build a relationship. You can check this in the tables and relationships section of the data integration. If two separated tables have a column in common, you can manually create a relationship between them. If there is no common column then you must create relationships in the example shown in the following procedure.

Advanced use case

If you cannot create data structures beyond n+1 using the common column relationship you can create a flattened data structure using a combination of API child calls and table merging. The general 'advanced use case' follows the basic principle of:

1. Set up your integration.
2. Edit you table structure.
3. Create your API call chains from your primary table to the table you want to combine to.
4. Merge tables via table merging in a top-down method (for example, parent to child).
5. When your large table is created, return to the parent table and set ignore for all table entities.

For example building a microapp with `request-list>item list>item details>approver`, the microapp must be able to show the request and detail for the approver - but is not able due to current limitation of only n+1 relationships. You can use the table merging feature to fix this problem.

While building your data endpoint, propagate the table structure from the parent data endpoint (`request-list`) to the child endpoint (`approver`) within the item list.

You can then set to merge everything from the parent data endpoint to this child API using a table merging strategy. The result is that everything that was in the parent table, displays in the data structure of the child API (`item-list`).

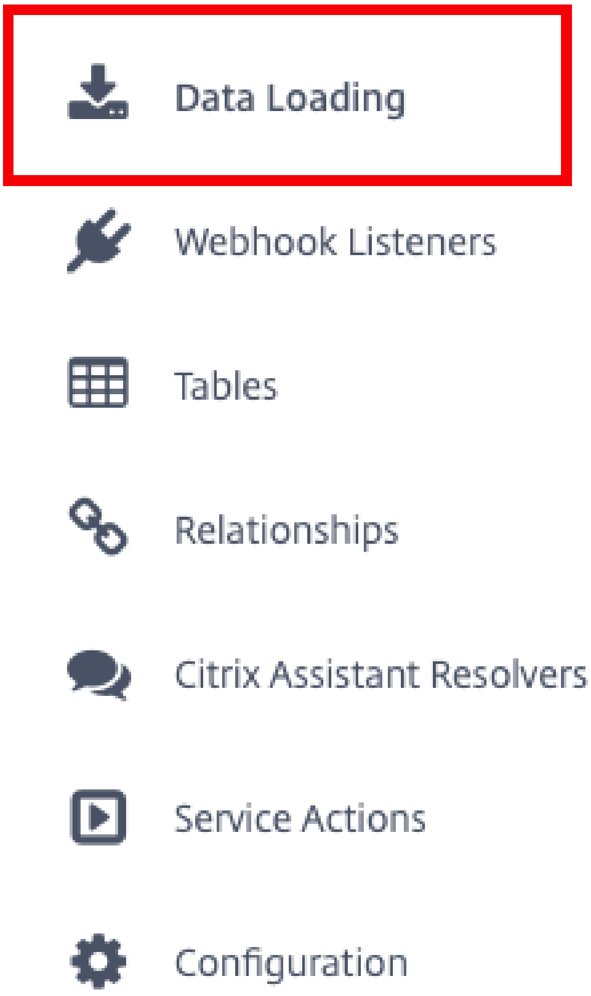
Configuring in this manner results in three levels of data being contained in one large database table. This new table can be used to build the page as per the use case defined when you started to build your microapps. This method can be used for as many levels are required.

API Child Call and Table Merge example

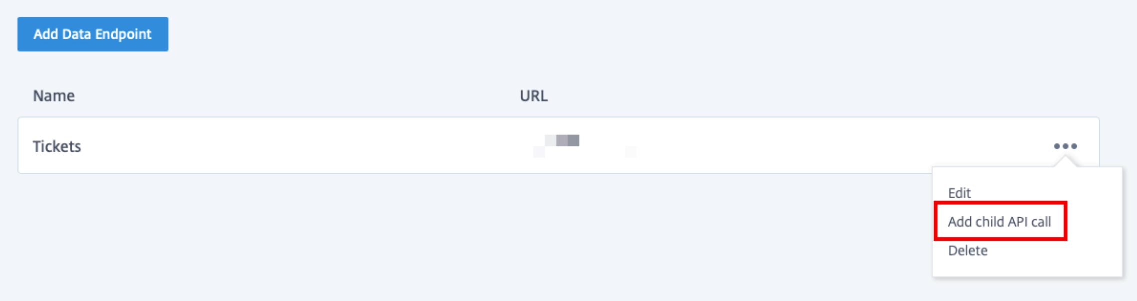
The following example illustrates the general workflow of creating a table structure to reach data beyond an n+1 relationship. Each individual use must be built based on the individual use case you want to build for your microapp. Ensure you are familiar with your target integration System of Record and you have a good understand of the outcome of your structure when using this method.

Create API call chain

1. Navigate to the **Data Loading** page for your integration:



2. Add as many child API calls from your root endpoint to the destination child endpoint as required:

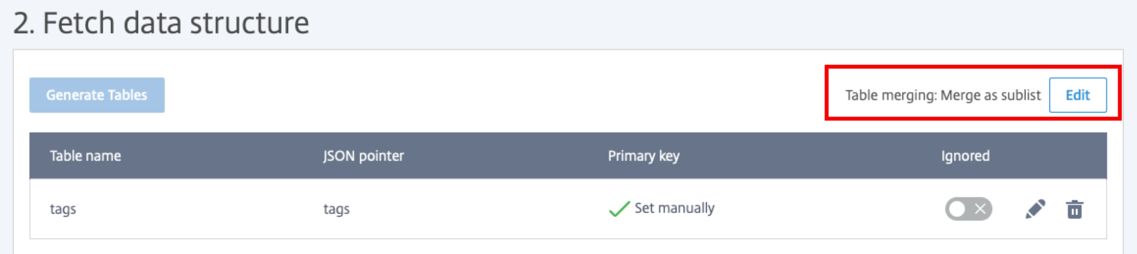


When finished, you can view your data structure on the main Data Loading page.

Merge the parent to children API calls

Now merge the root / parent table to the child endpoints in sequence until you reach the destination table:

1. Select **edit** from the ellipsis menu for your integration.
2. Select **edit** from the ellipsis menu for the child endpoint of your root integration.
3. Navigate to the bottom of the **Edit Data Endpoint** page and select **Edit** to select configure table merging:



2. Fetch data structure

Table merging: Merge as sublist

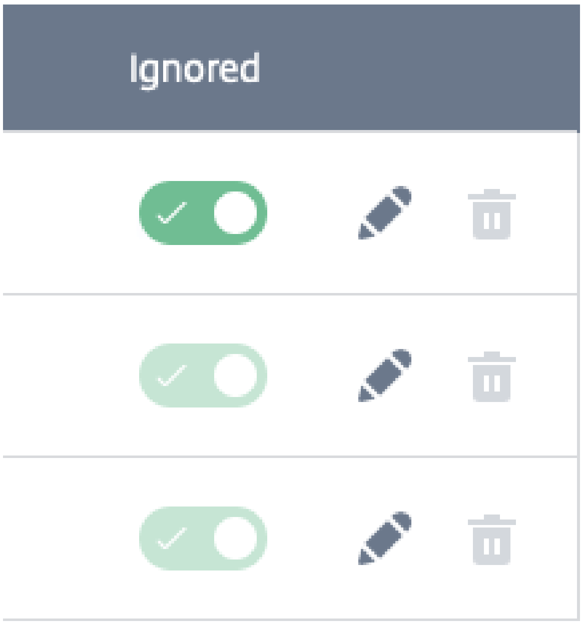
Table name	JSON pointer	Primary key	Ignored
tags	tags	✓ Set manually	○

Repeat this process as many times as needed for each child table in the sequence until you reach the destination table that will enable you to build your microapp.

Ignore repeated API calls

When you have finished the merge 'chain' return back to the root endpoint. Follow these steps:

1. Select **edit**.
2. Set all tables to the **Ignored** status:



This stops the table from loading twice into the cache and therefore improve performance. You can now use your chained / merged table to build your microapp.

Important Considerations

Always consider the following when building your data using this method:

- All parent and child API calls have their own data structure.
- These structures are different sets of data.
- If the data structure is merged (from parent to a child), all attributes show up in the child data structure.
- If the full chain is kept, the data is stored “twice” - ensure that the data structure in the parent call is deleted completely as every attribute appears in the child data structure.
- Don’t leave the parent API call with the data structure as is - delete it where possible.

Configure Webhook listeners

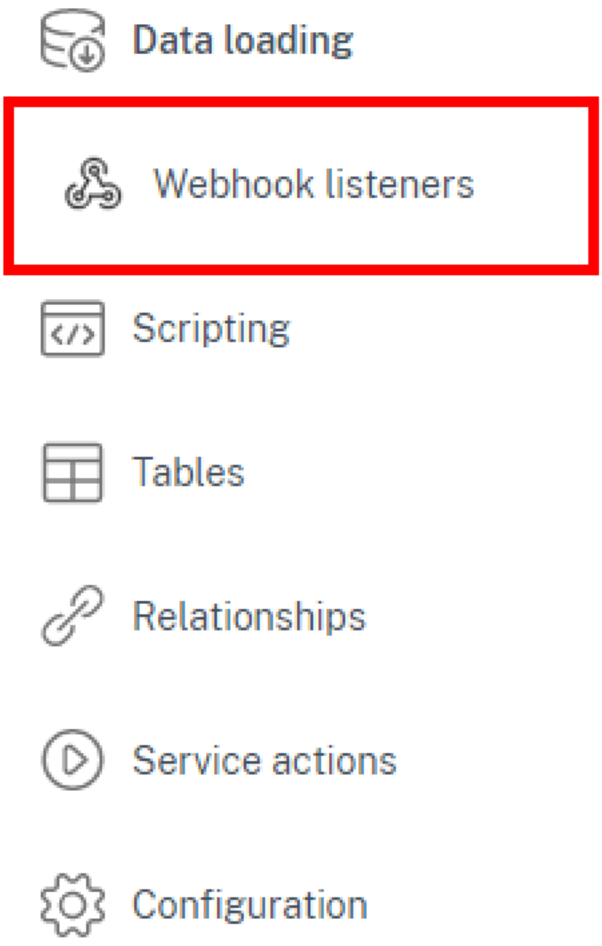
July 29, 2021

Configure webhook listeners (also known as HTTP push API) to enable your microapps to provide near real-time data to your end users. Configuring a webhook allows your apps to deliver data to other applications at a much quicker rate than synchronization from the Microapp platform side. The maximum size of a webhook request body that can be handled by MA server is 64 kb.

Adding webhook listeners requires you to be familiar with your target application System of Record and have the necessary tools and administration privileges set up to configure your webhooks in those locations.

Configure your webhooks after you have set up your integration via **Data Loading** and follow these steps:

1. Click **Webhook Listeners** on the left hand bar of the HTTP Integration screen:



2. Enter your desired webhook name.
3. Select **Copy** to copy the webhook URL for use in your target System of Record administration interface.

Authorization method

You can select either the **Token** authorization method or **None** when configuring your authorization method. To configure the **token** method follow these steps:

1. Select **Token** in the **Authorization** method menu.
2. Select **Generate token** and then select **Copy** to add the token to your clipboard for use in your target System of Record administration interface.
3. Select **read token from** to choose from:
   - Custom header
   - Query parameter
   - Authorization header
4. Define either the **Name** or the **Prefix** depending on your selected read method.

The token is now set up.

Request methods

When configuring your Webhook listeners, use the following definitions to build your calls:

- **PUT** is used to update existing resource.
- **POST** creates new subordinate resources, therefore, POST methods are used to create a new resource in the collection of resources.
  Both PUT and POST deletes data from child tables using the primary key info from of root table. It then replaces an existing record or inserts a new one.
- **DELETE** is used to delete resources.
  DELETE has two endpoints:
  - `{ serviceUuid } /{ webhookListenerUuid } /{ recordId }` where `recordId` is the value of the primary key of the record in the root table to be deleted. Records in the child table are deleted accordingly.
  - `{ serviceUuid } /{ webhookListenerUuid } /?id1=1&id2=1` where `id1` and `id2` represent the values of the composite primary key of the record in the root table to be deleted. Records in the child table are deleted accordingly.
- **PATCH** requests are used to make partial updates on a resource.

Define data structure

You can define your data structure in a similar method described when you **Fetch data structure** during **Data Loading** configuration. For more information see Configure the integration.

To define your webhook data structure follow these steps:

1. Set your desired data retention period. All entries that are older than this date are deleted. Each saved entry contains its date and time of modification. That is `lastModified`. This date and time is used to decide which entries to delete.
2. Select **Generate Tables**.

   The **Generate Tables** screen opens.

   Paste your JSON sample request from your target application System of Record here.
3. Set your base name of created tables.
4. (Optional) set the **root path** if necessary.
5. Select **Generate**.

With this process complete alongside the configuration measures completed in your target application System of Record administration, select **Add**.

Your webhook is now configured.

Scripting support

Webhook listeners support custom scripts configured in the HTTP integration scripting feature.

You can find additional information on developing and implementing your own scripts at the Citrix Developer Portal.

Show Webhook logs

Use **Webhook logs** to view a history of requests and errors from all webhook endpoints. You can filter by webhook name and state, such as success, error, or all. For the purposes of performance, only the last 10 webhook log entries are kept for review in the webhook logs screen.

1. From the Manage Microapps page, select the menu next to the integration that you want to view integration logs for.
2. Select **Webhook logs**.
3. Review the entries, and select from the menu to filter as required.

Create microapps

October 22, 2021

A key component of creating microapps is to plan your workflow with an understanding of the target app's database schema. With this, you can identify APIs to build your integration, which you need to build a custom integration.

There are countless applications that can be integrated into Citrix Workspace. Select a target application that holds information of interest to Citrix Workspace users. Of particular interest are applications that are regularly used for quick tasks and are not intuitively accessible to users. Actionable applications enabling users to directly interact from within Citrix Workspace have much more value than applications that simply enable notification of users. Examples include approving, creating, and adding items.

Next, identify key use-cases for the selected target business application that you want to integrate into Citrix Workspace. For example:

- Create PTO/Vacation request
- Approve PTO/Vacation request
- Find pending approvals
- Mark task complete
- Notify user of created or changed assigned tasks
- Approve invoice

Once use-cases are known, the next step is to identify the APIs that will allow us to extract relevant information from the target system or inject back into it.

Below we show a scenario of designing a workflow using an invoice approval use-case. For full details, see Sample scenario workflow design.

Important considerations

Review the following considerations and limitations before designing your workflow:

- You can access data on a page only one relation away. This means that when a page is built you can address only directly related data.
- All personalizations are made from the context of a user email. This means that if you are creating notifications or creating personalized pages, the user email can be a maximum of one level of relationship away.
- You can only set **Go to Page** action links for notifications to a page that is based on the same table as the notification.
- A record based detail page (that is a page using a recordID) cannot be set as an action page. Make sure that no component is mapped to a record value. For example, a Detail page should not be set as an action page, unless it's a pure input form, such as Add a Task.
- One change generates one notification. If the recipient is in the related table the relation must be 1:1. 1:N relations are not supported.
- The primary table is the table upon which you build the notification event.
- We expect that there is at most one recipient. If the recipient is not in the primary table, make sure that there is only one matching record in the non-primary table for each record of the primary table. For example, if you create a notification over the table `pto_approval`, but there

can be two different approvers who can approve that request in the table `pto_approver` (so called one-to-many relationship) and you want to notify them both, it is not possible. The notification engine picks up only one approver randomly. If there are many such cases, the notification event is not evaluated at all.

- The expected relationship between the primary table and other tables is 1:1. This means that for each record in the primary table there will be at most one record in the other table. Thus, the notification event can never produce more notification messages than the number of rows of the primary table. If this condition is violated and you have more than one record for some records of the primary table, duplication in the underlying data will appear. The notification engine will pick randomly only the first record and ignore the duplicates.
- If any value in the primary key column is missing or has an invalid type, the record is skipped during synchronization and a log warning is generated.
- There are some hard limits to protect the infrastructure and also the admin users from defining wrong notification events. The maximum ratio of the number of rows returned by the notification query compared to the number of rows of the primary table is 1.6:1. For example, let's say a primary table contains 130 rows, but the notification query returns 416 rows for some reason. This would be 3.2 times more than expected and exceed this limit. In this case, the notification event is not evaluated at all and a warning is printed to the log instead.
- The database structure must be narrow because of server limitations. This must be taken into account when you are designing your endpoints and creating a database structure.
- Notification messages are generated after all the conditions are evaluated. The maximum number of notification messages produced by one notification run is 100,000.
- The maximum number of records a notification query can return is 1,000,000.
- If you configured custom fields that contain highly sensitive data, such as credentials, API keys or secrets, the data is not protected. Such data appear in debug logs and elsewhere.

> **Note:**
>
> All administrators with access to Citrix Workspace microapps have access to data that is in the cache. Administrators do not have access to credentials for data sources.

Building microapps basics

Microapps are made up of *pages* or *event notifications*, and usually both.

- **Notifications** are event-driven microapps that automatically notify users when something requires their attention, for example as a card in the Workspace activity feed. Such microapps include *New Expense Report for Approval* and *New Course Available for Registration*. The following list shows available event trigger types:
  - New records - Sends a notification when a new record is created in the source of record (SoR).

  - Changed records - Sends a notification when an existing record is changed in the SoR.
  - Matching record - Sends a notification when records match a defined query at the specific time in the SoR.
  - Delete records - Sends a notification when a current record is deleted in the SoR.
  - Periodic notification - (user action) Sends non-data driven notifications periodically.
  - Periodic report - Sends periodic notifications with summarized report data (grouping) for a specified time interval.
  - Date reminder - Sends a notification at the specified time before or after the records date column value.

- **Pages** are user-initiated microapps that are available as actions in Workspace and make it easy to do initiate actions. For example, *Request PTO*, *Submit a Help Desk Ticket*, and *Search the Directory*. The following list shows page type templates:

  - Detail - Create a page to show static details from an individual record from your SoR.
  - Form - Create an editable page to provide static details in addition to the ability to input user data into your page.
  - Table - Create a page based on the multiple data tables loading from your target application SoR.
  - Static content - Create a page to show static, non-actionable information such as headlines, error messages, or reminders.

Add a new microapp

This procedure is the same for any blank microapp that you want to create.

Follow these steps:

1. From the **Microapp Integrations** page or in the integration view (opened by selecting the integration), select the menu next to the integration that you want to add the microapp to.
2. Select **Add Microapp**.
3. Select **Blank Template** to build your own microapp based on your business needs.
   After you add the blank microapp, it appears under the related integration on the **Microapp Integrations** page.
4. Return to the Manage Microapps page and select **Blank Microapp** from the list under the integration.
   The **Properties** page opens.
5. Give it an appropriate name and description.
6. Select **Microapp Icon** and choose an appropriate icon from the menu. There are **App Icons**, **Action and Notification Icons**, and **Microapps and Data** icons from which you can select.

Clone a microapp

You can also clone an existing microapp to create a new microapp. This microapp exists in the same integration. You must give the new microapp a unique name as no two microapps in the same integration can have the same name.

1. From the **Microapp Integrations** page or in the integration view (opened by selecting the integration), select the menu next to the microapp that you want to copy, and then select **Clone**.
2. Enter a **New microapp name** in the field, and select **Clone**.

   The new microapp is added to the list of microapps.

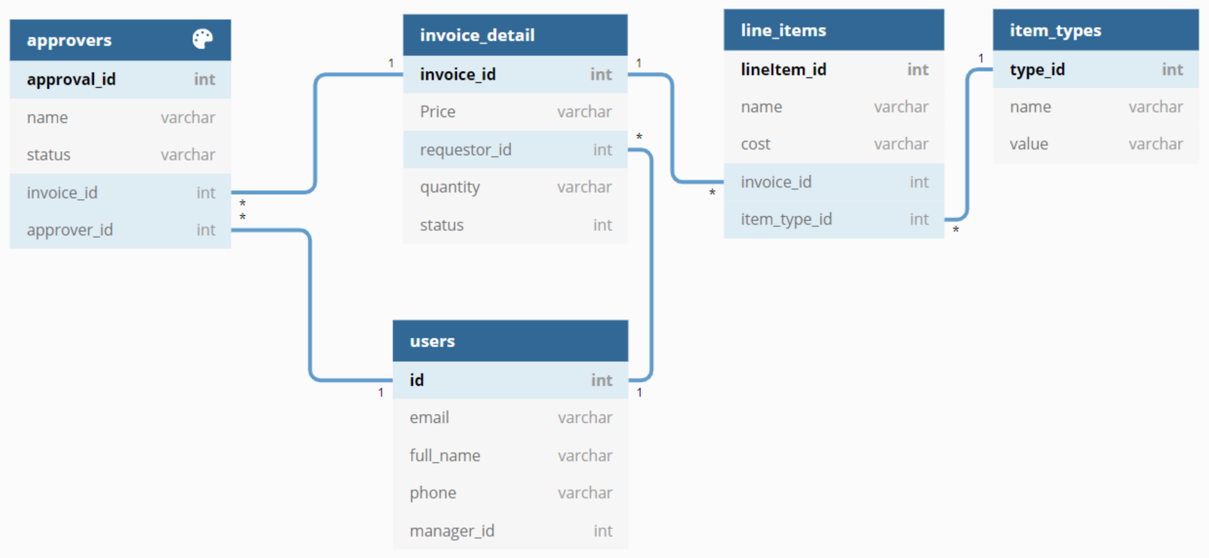
> **Note:**
>
> You can also export and import a new version of the microapp from the same menu. For more information on these capabilities, see Export and import integrations and microapps.

Sample scenario workflow design

You have an invoice approval system, and you need to accommodate the following use-cases into our workflow:

- Approvers must be notified when there are new approvals.
- Requestors must be notified when their request is approved or denied.
- Approvers need information about invoices, including status, total price, requestor details (name/email/phone), and a list of line item details (name/price/quantity).
- Requestors need information about invoices, including total price, list of approvers details (names/emails/phones), and a list of line item details (name/price/quantity).

Now let’s have a look at our database, and their table relationships:



approvers	
approval_id	**int**
name	varchar
status	varchar
invoice_id	int
approver_id	int

invoice_detail	
invoice_id	**int**
Price	varchar
requestor_id	int
quantity	varchar
status	int

line_items	
lineItem_id	**int**
name	varchar
cost	varchar
invoice_id	int
item_type_id	int

item_types	
type_id	**int**
name	varchar
value	varchar

users	
id	**int**
email	varchar
full_name	varchar
phone	varchar
manager_id	int

Design your microapp

From this you know that you need four notifications and five pages.

You need to build four notifications, two for approvers and two for requestors. The approver and requestor email is in the user table which has a direct relation with `approvers` and `invoice-detail` tables.

You need to build five pages, one for each of these: approvers invoice list, approvers invoice detail, requestors invoice list, requestors invoice detail, and approvers and requestors line item detail.

Build your notifications

Start by creating the notifications. All of the considerations and limitations apply to notifications. Notifications must be sent by the user. The user email needs to be in the table or a maximum of one level of relationship away.

Build the notifications for approvers on the `approvers` table:

- notification for new record in the table
- notification for status change

Build the notifications for requestors on the `invoice_detail` table:

- notification for new record in the table
- notification for status change

Design limitations (approvers)

Data for the approver feed card can only be taken from these tables:

- `approvers` (primary table)
- `invoice-detail` (invoice_id relation)
- `users` (only approver_id relation)

This means that you cannot get any data about the requestor to the feed card because personalizations are made from the context of a user email. For example, if you want to have the requestor name you must change the database schema and add the requestor name to the `invoice_detail` table.

Design limitations (requestors)

Data for the requestor feed card can only be taken from these tables:

- `invoice-detail` (primary table)
- `users` (only requestor_id relation)

This means that you cannot get any data from `line_items` and `approvers` because the relation is 1:N. For example, feed card text *Your request was approved by manager@company.com* is not possible. If you need this information, you must change the database schema and add this information to `invoice_detail`.

Conclusions

From this you can determine that there must be two `invoice_detail` pages:

- Invoice Detail Approver that you build on the `approvers` table
- Invoice Detail Requestor that you build on the table `invoice_detail`table

You can recognize now that you have a limitation here with the **Invoice Detail Approver** page. You can add all data from the `approvers` (primary table), `invoice-detail` (`invoice_id` relation), and `users` (only `approver_id` relation) tables. However, you have the same problem as with the notification. There is missing requestor information and the `line_items` table is too far away, that is two levels of relation.

Workaround

There is a workaround to get data from a table two levels of relation away.

Option 1 Use GotoPage. You can add a third button, such as *See details* and move your users from this page to the **Invoice Detail Requestor** page. You built that page on the `invoice-detail` table, so the `requestor` and `line_items` tables are only one level away.

Option 2 Use the unbound table component: Unselect the **Use Records Related to Detail Page** toggle and you can select `line_items`. This creates a table with all items. You need to add a filter to select only items for the particular invoice. `line_items` invoice_id = `approvers` invoice_id. You can use a similar approach for the `requestor`. As the table is over the `line_items`, you can also add data from a table which has 1:1 or N:1 relation with `line_items`.

Build your pages

You need to build five pages, one for each of these tables:

- approvers invoice list
- approvers invoice detail
- requestors invoice list
- requestors invoice detail
- approvers and requestors line item detail

If you need to allow for detailed permissions based on user assignments, use a separate microapp. For example, your workflow might require only certain users to access a create page. For a complete overview of the page builder UI and its components, see Page builder components. The cookbook below has detailed steps that leverage useful components to build *detail* and *list* type pages.

Microapps cookbook

Follow these examples of common types of notifications and pages you can build with a list of ingredients (components) provided.

Notifications

Build a notification to push new or changed items from your workflow to users. Select from the event trigger types, shown below, then customize the event in the builder. For more information, see Build event notifications.

Event triggers:

- New records - Sends a notification when a new record is created in the source of record (SoR).
- Changed records - Sends a notification when an existing record is changed in the SoR.
- Matching record - Sends a notification when records match a defined query at the specific time in the SoR.
- Delete records - Sends a notification when a current record is deleted in the SoR.
- Periodic notification - (user action) Sends non-data driven notifications periodically.
- Periodic report - Sends periodic notifications with summarized report data (grouping) for a specified time interval.
- Date reminder - Sends a notification at the specified time before or after the records date column value.

List page

Build a list page to show all items available in your workflow. Start by using the following components. For a look at the finished page and step-by-step details to reproduce it, see Build a list page.

Components:

- **Table** - Add a table by defining table source, filters, and defining columns. Page link actions can be added. Personalized queries must be set to limit data exposure.
- **Text Input** - Define text source by specifying the data table, column, and value to load to the page entered by the user. Component can be marked as optional. Field width can be modified. Validation rules can be configured based on a minimum or maximum length or text pattern to identify user input.

Detail page

Build a list page to view details of one item that is available in your workflow. Start by using the following components. For a look at the finished page and complete steps to reproduce it, see Build a detail page.

Components:

- **Text** - Define text source and formatting to load from the cache to the page.
- **Back Button** - Allows users to go back to previous page.
- **Static Text** - Define static text to appear the page.
- **Flexible Grid** - Gives you more control over the positioning of components on your pages. Helpful when you're designing pages intended for devices with larger screens. Set the label and the total number of cells you want in your grid.
- **Table** - Add a table by defining table source, filters, and defining columns. Page link actions can be added. Personalized queries based on users' emails may be set to limit data exposure.

Create page

Build a create page to add items into your workflow. Start by using the following components. For a look at the finished page and complete steps to reproduce it, see Build a create page.

Components:

- **Static Text** - Define static text to appear the page.
- **Flexible Grid** - Gives you more control over the positioning of components on your pages. Helpful when you're designing pages intended for devices with larger screens. Consists of **Grid Items**. Set the label and the total number of cells you want in your grid.
- **Text Input** - Define text source by specifying the data table, column, and value to load to the page entered by the user. Component can be marked as optional. Field width can be modified. Validation rules can be configured based on a minimum or maximum length or text pattern to identify user input.
- **Select** - Allows users to choose from a set list of values. Populated by data from the source system or you can enter the list of values manually. Actions can be added.
- **Lookup** - Allows users to search though a large quantity of values and allows users to select a value by searching for something else.
- **Button** - Add a clickable component on the page with actions and logic.

Embed

The **Embed** page template renders an iframe for embedding custom webpages. The size of the iframe is adjusted automatically to the available space on the page. The Embed page has no components, and is not bound to a database table, similarly to the Static Content page. This also means that the

URL for the Embed page cannot make use of template variables as there is no connected database table. For information on the embed component see Embed component.

Other resources

Check out this overview of Citrix Workspace Intelligence and the Microapps service at Video: Microapp Overview.

Learn about creating custom integrations and microapps at Video: Microapp Custom Integrations.

Find out more about getting a test instance at Citrix Workspace Developer Portal.

Here's a quick guide to setting up an RSS microapp: Get notifications when there is a Citrix security bulletin.

Visit the Microapps Discussions Forum.

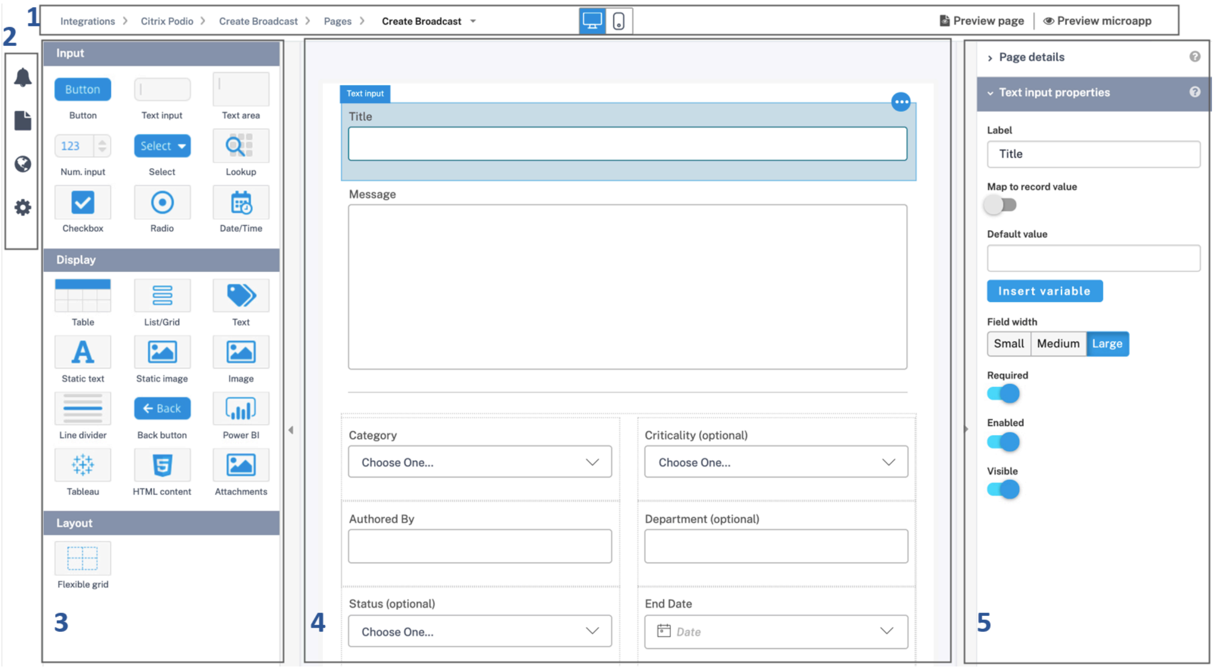
Page builder

November 4, 2021

Familiarize yourself with the Page Builder and its components to enable you to create action pages. The different components and sections of the page builder are described in the following sections. You add and customize extra fields and buttons depending on your own requirements.

Page builder walkthrough

The screenshot below shows the complete page builder with sections called out. Descriptions follow below referencing the numbered sections:



1. The top bar has selectable breadcrumbs on the left. Selecting the page name (in bold with an open menu icon on the right) allows you to quickly jump between pages. In the middle you can select either standard monitor or mobile view. On the right you have preview options. **Preview page** presents a view of this page you're viewing in the builder. **Preview microapp** delivers a mock workflow of the microapp actions where you can open all pages and view notifications and their actions.

2. The left-hand side has quick navigation options to screens in the microapp. From top to bottom, you can jump to a list of all **Notifications** or **Pages** in this microapp, the **Localization** screen, and the **Properties** screen where you can modify the name, description, and icon. For more information, see

   - Build a notification
   - Localize microapps
   - Add a new microapp

3. Components are in the left pane. Select and drag them to the builder canvas in the middle section of the screen. See Page builder components. Components are divided into Input, Display, and Layout.

   - Input components create actionable sections on your page including buttons, text input, and radio buttons.
   - Display components deliver information to your end users of microapps including tables, static text, and images.
   - Layout component provides the grid component for setting the layout of your page.

4. The builder canvas is the middle section. You can move the components around here to arrange

them as you require. Select the component here to enable component properties, actions, and other tabs that are visible in the right pane.

5. Customize the components and add page details in the right pane. The Page Details tab lets you configure the page you are creating by entering name, setting filters for information, and adding logic to page components. Also, use the informational debugging feature. This tab remains the same for the page and this tab does not depend on the selected component. Other tabs differ depending on the component that is selected. Available tabs include:

    - Properties: Each page builder component has its own specific properties menu with various options to choose from depending on the component.
    - Input validation rules: Some components enable their own specific validation rules.
    - Actions: Different actions are available depending on the components. Actions allow the microapp recipients (Workspace user) to respond with actionable input.

    Other tabs that are unique for one component are fully described with the relevant component below under the Display components and Layout component sections.

For a complete list and description of available component property fields, toggles, and selectable elements, see Component properties.

Page templates

When creating a page, you can select from the following basic page layouts depending on the information you want. Each template is intended only to speed up your activity to produce the page you want.

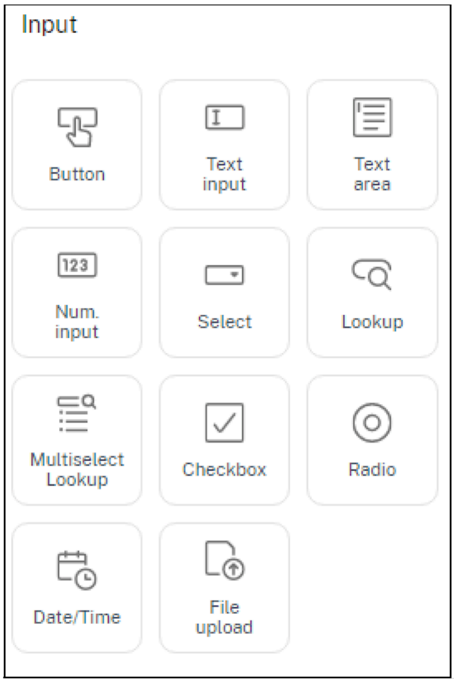
- **Detail** - Page template that provides static details and is connected to a particular record from the cache.
- **Form** - Create a page that provides static details in addition to the ability to input user data into your page.
- **Table** - Create a page listing multiple records based on the data tables loading from your target application integration.
- **Static content** - Set up page components that provide static, non-actionable, information such as headlines, error messages, reminders.
- **Embed** - Create a page that renders an iframe for embedding custom webpages in your microapps.

Page builder components

The Page Builder lets you choose from various page components that let you customize and configure your microapp output, information, and display. Use these components described below to build a page microapp based on your expectations and needs. For example, if you want to show a list of users, you use the Table component to build it. The different template pages have different component features available by default. The following lists cover all available options that are available.

Input components

Input section provides components that create actionable sections on your page including buttons, text input, radio buttons and so on.



Button

Add a clickable component on the page with actions and logic. Button size and style can be adjusted. There are actions as options to Run Service Action, Go to URL, and Run Notification Trigger. For example, using actions the button can direct users to another page or submit an entry. For more about button actions, see Actions.

Text input

Define the text source of the displayed data by specifying the data table, column, and value that a user sees on the page. Component can be marked as optional. Field width can be modified. Validation rules can be configured based on a minimum or maximum length or text pattern to identify user input. For an example of this component in use, see Build a create page and Build a list page.

Text area

Define the text source of displayed data by specifying the data table, column, and value that a user sees on the page. Component can be marked as optional. Validation rules can be configured based on a minimum or maximum length or text pattern to identify user input.

Num. input

Define source of displayed data by specifying the data table and column that a user sees on the page. You can define the format such as time, date, and so on Component can be marked as optional and field width can be modified. Validation rules can be configured based on a minimum or maximum length or text pattern to identify user input. **Precision** defines the number of total digits. **Scale** defines the number of digits to the right of the decimal point.

Select

Allows users to choose from a set list of values (limit is 100 values). Populated by data from the source system or you can enter the list of values manually. Component can be marked as optional and field width can be modified. You can define the format such as time, date, and so on For an example of this component in use, see Build a create page.

Lookup

Allows users to search through a large quantity of values and allows users to select a value by searching for something else. You must specify where the data is being pooled from. Component can be marked as optional. For an example of this component in use, see Build a create page.

> **Note**
>
> Configuring the Lookup component with the 'Contains' strategy can result in long load times for a search term. We recommend you test with a large sample of data to check whether there is an impact on performance and user experience. Alternatively you can use the 'Starts with' strategy to improve performance, but only records that start with the search term are returned.

Multiselect lookup

Multiselect lookup works in a similar manner to the **Lookup** component and allows users to search through, and select a large quantity of values by searching for an alternative value.

For example users can search and add multiple users or user emails when scheduling a meeting, or add multiple labels when creating a Jira ticket. To specify where your data is pooled for use by the multi select lookup component you must configure the **array** data type during service action configuration in your HTTP integration.

Example use cases for multi select:

- **Webex** - Select multiple users to send a meeting invite.
- **Smartsheet** - Select multiple users to share a sheet.
- **Jira** - Select multiple labels to assign to a single Jira ticket.
- **Office 365 Calendar** - Select multiple attendees when scheduling an event.
- **Office 365 E-mail** - Select multiple email recipients.

Configuration

Configuration is similar to Lookup but has two more configuration options: "Max items count" and "Can create new items". Apart from the standard page builder configuration options, define the following to configure Multiselect lookup:

Map to record value - defines the data source for preselected values and is available only on the detail page (a record id is required). Unlike Select, Multiselect lookup is not restricted to a single value.

Data source for options - defines the data source for options of the drop-down list.

Search term matching strategy - selects whether the search term matches the start of the database value or anywhere inside the database value.

Max items count - sets the maximum number of items selected during a search.

Can create new items - enables users to create items that are not yet available in the lookup selection (for example, add new labels to a Jira ticket).

Integrity Check

The integrity check works the same as lookup but the maximum number of items selected must be non-negative. The data source table must exist and table columns for value, display value, and additional data must be correctly mapped to a data table with a foreign key.

For more information on configuring your arrays to return multiple entries, see Data types.

Checkbox

Add a selectable component on the page by defining source (data table and column), and default (either selected or disabled). Component can be marked as optional.

Radio

Add a set of options where only one can be selected. Populated by data from the source system or you can enter the list of values manually. Component can be marked as optional.

Date/Time

Define either date, time, or date and time to display in the microapp, and default time (the time displayed before selection) to load to the page. Component can be marked as optional and field width can be modified.

File upload component

Allows uploading raw files to the SoR during submit or update actions. To enable this action, create a special service action that supports file upload.

This component enables user workflows for uploading files to SoR as attachments to existing records and as attachments when creating records. For example:

- **Jira**: Upload log files and screenshots and attach them to an issue ticket.
- **Salesforce**: Upload a discovery agreement document, and attach it to an account or opportunity.
- **SAP Concur**: Attach a receipt when managing expense reports.
- **ServiceNow**: Upload an attachment as part of a submission workflow.

Create service action

To enable file uploads, scripting must be used to configure a service action. There is a **FILES** parameter type to support this capability. For a general overview, see HTTP integration scripting.

Before you begin

- Ready your script that you want to import via the Microapp administration interface.
- Scripts must be written in the javascript language edited in your preferred text editor / development tool.
- When ready, import the script via the integrations tab in the Microapps admin interface or optionally you can enter your script directly into the text editor provided in the scripting feature.
- When imported, test the script.

The following is an example of a service action which uploads multiple files at once to JIRA. For more scripts, see this Script repository.

~~~~
```javascript
function addAttachmentsSingleRequest({
 client, actionParameters }
 ) {

    console.log(`attaching file(s) to issue ${
 actionParameters.issueKey }
 `);
    const formData = new FormData();
    const url = `/rest/api/2/issue/${
 actionParameters.issueKey }
 /attachments`;
    actionParameters.attachments.forEach(file => {

        formData.append("file", file);
     }
 );
    const response = client.fetchSync(url, {

        method: 'POST',
        headers: {

            "Content-Type": "multipart/form-data",
            "X-Atlassian-Token": "nocheck"
         }
 ,
        body: formData
     }
 );
    if (response.ok) {

        console.log('Attachment(s) posted');
     }
  else {

        const errorMessage = `Request failed(${
 response.status }
 : ${
 response.statusText }
 )`
        console.error(errorMessage)
        throw new Error(errorMessage)
     }

~~~~

````
 }

<!--NeedCopy--> ```
````

Import the script

To import your prepared script, follow these steps:

1. From the integration configuration screen of the integration, select **Scripting** from the left-hand navigation.
2. Select **Upload script**. Alternatively, you can input your script directly into the text area by selecting **Edit**.
3. Drag your script onto the import pop-up.
4. The script is parsed and validated.
5. Select **Import**.
6. Your script is imported.

More information about scripting

- To get started developing scripts, see Citrix Developer Portal.
- For an end-to-end process of setting up a custom integration using scripting, see Getting started with Microapps scripting.
- For examples of Microapps scripts, see Microapps script SDK.

Configure the File upload component

Now return to the page in the microapp for this integration that you want to create for uploading files. After dropping the **File upload** component in the builder, complete the **File upload properties**.

1. Modify the value for **Label** if desired. By default, this is **File upload**. No label is required. This value appears above the drop frame in Citrix Workspace.
2. Modify the value for **Max files count** if necessary. By default, this is **1**. This value appears next to **File upload limit** under the drop frame in Citrix Workspace. When multiple files are selected, metadata upload and content upload for each file occurs independently. This capability is limited by the SoR.
3. Modify the value for **Max file size MB** if necessary. By default, this is **5**. This value appears next to **Total size limit** under the drop frame in Citrix Workspace. File size is limited by the SoR. We recommend configuring the maximum file size to what is actually needed to prevent a waste of resources.
4. Enter a value for **File extensions (e.g. .jpeg, .png, .pdf)**. No value is required. This value appears next to **Accepted file formats** under the drop frame in Citrix Workspace. This field is not case sensitive and does not accept special characters.

5. Enable the **Required** toggle if adding a file is required to submit this page. A **This field is required** label is shown above the drop frame in Citrix Workspace.
6. Disable the **Enabled** toggle if you want the drop frame to appear dimmed and unavailable in Citrix Workspace.
7. Disable the **Visible** toggle if you want to hide the drop frame in Citrix Workspace.

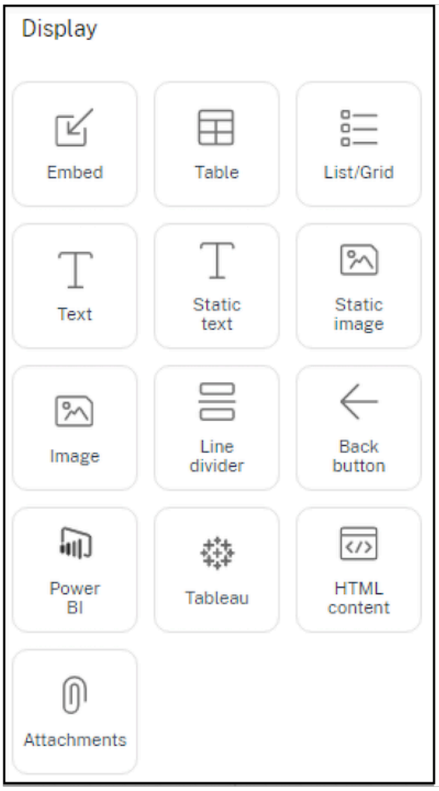
Considerations

Users should consider the following limitations when uploading content:

- The file name must end with one of the allowed suffixes. All characters are allowed except for null byte `\0`.
- Microapps service does not provide file storage. Files are uploaded directly to the SoR. During the upload process Microapps service first transfers the file from a users computer to secure temporary storage and then immediately streams the file to the SoR. Processed files on the temporary storage are from that point not accessible for any purpose to anyone and are deleted after a short period of time.
- Users can remove a file that they have previously selected and pre-uploaded before confirming the upload.
- The file type (mime type) is detected by the browser based on the file's extension. Microapps service does not use the file type for any checks.

Display components

Display section provides components that deliver information to your end users of microapps including tables, static text, and images and so on. Wherever you can select user email as a variable to extract data when building an action page, you can use an Active Directory user principal name (UPN) attribute. This includes page and component filters, constraints, and service action parameters.



Table

Add a table to display a list of records by defining table source, filters, and defining columns. Page link actions can be added which direct the user to another page. Alignment can be modified.

A **Columns** tab is enabled in the right pane after selecting this component. Add as many columns as required with these settings: **Column title**, **Data type**, **Data table**, **Data column**, **Format**, and **Conditional format**. See Component properties for explanations of these fields.

Personalized queries must be set to limit data exposure. A table without a personalized query exposes all records that match a table filter regardless of the data relation to subscribers. Your potentially sensitive data may be exposed to all microapp users unless you limit data exposure. Control data access for pages using the **Data filter** feature. Select the table in the component builder pane. Select **Table Properties** and then **Data filter**. Select **+Add Condition** and add the constraints as required. For an example of this component in use, see Build a detail page.

List/Grid

Displays a list of data to users in Citrix Workspace. Select from preconfigured layouts to surface the right information in a way that suits your data the best. You can define items in the list manually or pull them from the cache.

Under the **List properties** tab, select either:

- **Data table** to pull data from the cache. The **List data source**, **List items data source**, and **Layout & Style** tabs are available. Property descriptions for **List data source** and **List items data source** are available in Component properties.
- **Specify manually** to define items in the list. **List items** and **Layout & Style** tabs are available. For **List items**, define what items you want in your list. Property descriptions are available in Component properties.

The **Layout & Style** tab is available for both types of list properties. The **Layout** menu provides a selection of preconfigured layouts. Choose the layout that best suits your needs from the following options:

- **List - accordion**: Displays expandable text only content units.
- **List - basic**: Displays text only lists.
- **List - bulleted**: Displays unordered, bulleted text only lists.
- **List - ordered**: Displays ordered, numbered text only lists.
- **List - thumbnail**: Displays lists with thumbnails and styles.
- **Grid - hero image card side**: Displays cards with a large image on the side.
- **Grid - hero image card top**: Displays cards with a large image on the top.
- **Grid - image**: Displays multi-columned image units.
- **Grid - thumbnail side**: Displays multi-columned lists with thumbnails.
- **Grid - thumbnail top**: Displays center-aligned content units.

Text only layouts allow you to toggle **Show title** and **Show description**. You can disable one of these, but not both options. Thumbnail layouts open a **Show image** toggle and other formatting options for images. Other property descriptions are available in Component properties.

Text

Define text source from tables and formatting to load to the page. Data Source, Table, Column are all selected for the text to populate. A text format such as time, date, and so on can be given or a conditional one based on parameters. You can add actions to go to a Page, URL, Send Email, or Call Phone. For an example of this component in use, see Build a detail page.

The Text component is designed to display a single database value. An Integrity check is run and alerts users if a page relies on a record ID. For detail or form type page that is set as an action page, a page data filter is required to call on a unique record. We show this message to alert you of this issue. Your microapp might work as is, but the component can display the wrong data if the unique record is not passed over to the page.

Static text

Define static text to appear on the page. Actions can be added. Alignment can be modified. You can make a distinction if the text is a header. Font style of bold and italics are available. Font size can be changed from Normal, Light, Small, and Small Light. You can add actions to go to a Page, URL, Send Email, or Call Phone. For an example of this component in use, see Build a create page and Build a detail page.

Static image

Display a static image referenced from a predefined static URL source. Image size and alignment can be specified. No actions can be added.

Image

Display an image using its URL stored in the cache (data table and column) and formatting to display on the page. Add text to display if image cannot load. Image formatting such as alignment, size, and shape can be configured.

Line divider

Use the divider to separate unrelated and group related information. No properties can be modified.

Back button

Allows users to go back to previous page. A variable can be specified; that is data table, data column, and format. For an example of this component in use, see Build a detail page.

Power BI

Displays a BI report in Workspace. Define source and formatting. When a user views this Power BI component in Citrix Workspace, they are first prompted with a Power BI login.

A **Power BI component properties** tab is enabled in the right pane after selecting this component. When this component is first added, no service actions are selected and you cannot edit the component. You must set up service actions for user authorization. After setting up service actions, set up authorization in the component. For complete steps, see Configure Power BI component service actions.

Concur receipt viewer

Displays a receipt image. Define source with data table and data column, and formatting. The Concur receipt viewer component is only visible in Concur integration template.

Tableau

Displays a Tableau report. Define source with data table and column.

HTML content

Displays HTML content from pulled sources (for example RSS feeds) to display HTML correctly. Define source with data table and column. Basic text elements are supported. The allowed elements are:

- "p", "div", "h1", "h2", "h3", "h4", "h5", "h6", "ul", "ol", "li", "blockquote", "b", "i", "font", "s", "u", "o", "sup", "sub", "ins", "del", "strong", "strike", "tt", "code", "big", "small", "br", "span", and "em".
- "a" element only with attributes "target" and "href".
- "img" element only with attributes "height", "width", "src".
- Also, the "style" attribute is allowed on any element.

Attachments

Lists attachments from data source and allows end users to download attachments. Images and PDFs can be previewed directly. Define source by specifying the data table, data column for URL, and data column for name.

- If data mime type (media type) is not configured, then attachment preview icon appears as *?*.
- If file size is not configured, then attachment preview shows *0B* as the size.

Select **Attachment URL security** option:

- **Inherited**: Attachments must be housed on the same domain that the integration accesses. If not, the attachment does not display and cannot be downloaded. For example, this is a known issue for Google integrations.
- **Public**: Public attachments from other domains can be displayed if they do not need an authentication method.

Embed

Allows webpages to be embedded in your microapps using the builder and then displayed in Citrix Workspace. Supported content includes: public YouTube videos, Google maps, Podio/Google/Microsoft forms, most public responsive webpages, and any content designed by its provider to be embedded into a website and displayed in an iframe.

Some external content is not designed with embedding in mind or it is strictly prohibited. For example

- Webpages that explicitly forbid embedding by sending the `X-Frame-Options "SAMEORIGIN";` header. A notable example is `https://google.com`.
- Webpages that redirect to a different domain than what is specified in the component configuration in the page builder.
- Webpages that require authentication can run into problems when embedded. Administrators must test carefully.
- Webpages that contain cookie consent (for example, for GDPR requirements) do not have their selections cached in the embed component. If a user accesses the embedded site again, they must reconfirm cookie consent each time unless the site uses the `SameSite: none; secure =true` cookie attributes.

Configure the Embed component

After dropping the **Embed** component in the builder, complete the **Embed Properties**.

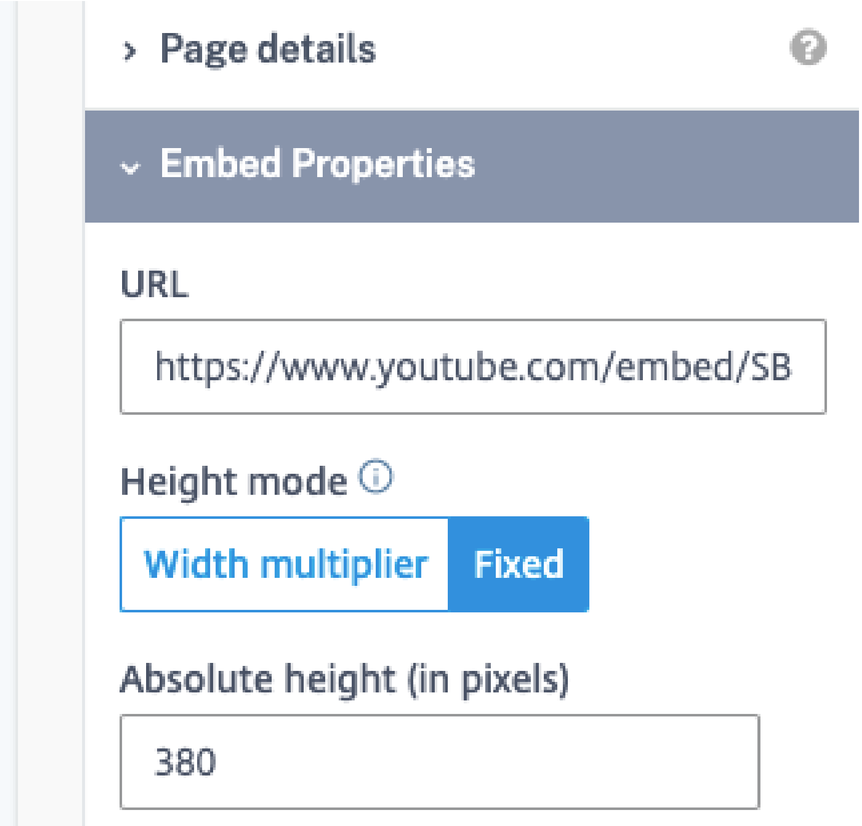
> **Note**
>
> To successfully embed content from some websites you must use the embed code generated directly from the website (as opposed to a direct URL or share link).
> For example, when generating an embed link on Google Maps via the **Share** option, use the link generated from **Embed a map** rather than **Send a link**.

1. Enter the **URL** of the content that you want to embed.

2. Select one of the **Height mode** options to choose how the embedded content size is determined.

   - **Width multiplier**: Enables a **Multiplier (in percents)** field. Enter a percentage value in relation to the width of the embedded content. For example, if you enter *50*, the displayed content is half the height of its width.
   - **Fixed**: Enables an **Absolute height (in pixels)** field. Enter a pixel count to determine the height of the embedded content.

Embedded content previews are not available in the builder. Select **Preview microapp** to see how the embedded content looks.



URL

https://www.youtube.com/embed/SB

Height mode

Absolute height (in pixels)

380

Considerations

Consider the following when embedding content:

- Use embeddable versions of webpages when possible. These pages have already been optimized for embedding by the content provider. For example, a YouTube video with id `XXX` can be accessed at `https://www.youtube.com/watch?v=XXX` and `https://www.youtube.com/embed/XXX`. The first URL leads to the full site and the second to the embeddable video.
- When specifying a URL, you can use template variables to pass a parameter over to the remote URL. For the same YouTube example and assuming you have a data table called `video_id` containing video IDs, enter `https://www.youtube.com/embed/{ video_id }` into the `URL` field. The string `{ video_id }` will be substituted by the row value from your table.
- The embedded page is displayed in a sandboxed iframe with the following attributes: `allow-scripts allow-same-origin allow-popups allow-popups-to-escape-sandbox allow-forms allow-pointer-lock allow-downloads`. These flags ensure complete isolation of the third party content from Citrix Workspace, but can cause some functionality on the remote page to be broken.
- The Embed component cannot display documents which require custom browser plug-ins. For this reason, PDFs cannot be shown within the Embed component.

Layout component

Layout section provides the grid component for setting the layout of your page.

- **Flexible grid** - Gives you more control over the positioning of components on your pages as allows for an easier "snap-in" of the components. This option is helpful when you are designing pages intended for devices with larger screens.

  A **Flexible grid properties** tab opens in the right pane after selecting this component. Set the number of columns and rows that you want in your grid and customize alignment. For an example of this component in use, see Build a create page and Build a detail page.

Page details

Configure the page you are creating by entering name, setting filters for information, and adding logic to page components. Also, use informational debugging features:

- **Page name** - Set the name of the page.
- **Data filter** - Use to set constraints on the action data.
- **Show SQL** - Use this to display the SQL for debugging purposes.
- **Logic** - Use this to add and display the component logic. Find details of **Add logic** in the following section.

Add logic

Select the **Add logic** button under the **Page Details** tab to open the logic configuration. This enables you to configure the logic for your particular page component. Specify the behavior or appearance of the components on this page. Create conditions using standard logic arguments to achieve various outcomes depending on the desired behavior of your component. You can add multiple, stacked actions to any single button to create multiple action effects with a single click. When each service action runs successfully, the system moves through the chain of actions until all are completed. Available actions include:

- **Evaluate condition** - Set and edit condition via a logical argument.
- **Set component value** - Set the component value for the button to commit an action.
- **Show component** - Configure conditions to show the component.
- **Hide component** - Configure conditions to hide the component.
- **Enable component** - Configure conditions to enable the component.
- **Disable component** - Configure conditions to disable the component.
- **Set component to required** - Configure conditions to require the component.
- **Set component to not required** - Configure conditions to exclude the component.

Component properties

Each page builder component has its own specific properties menu with various options to choose from depending on the component including:

- **Label** - Customize the label of the button, text, image, and so on.
- **Placeholder text** – Explain to user how to use this component. For example, list attributes that a user can search with.
- **Content** – Enter static text to show end user.
- **Alignment** – Set alignment of displayed text or image.
- **Text type** – Set text size of displayed text.
- **Font style** – Set font style of displayed text.
- **Font size** – Set font size of displayed text.
- **Field width** – Set how wide a component displays.
- **Format** – Define how data is formatted when displaying to users. For example, date, time, decimal, percent, and so forth.
- **Default value** – Used for the **Input components** to set default value manually or by using variables.
- **Style** - Used for the **Button component** to switch between different predefined colors and formats.
- **Use records related to the page** – Use for the **Table component** to filter records related to the record displayed on the page. For example, you have a page displaying data about an account and you want to display the table with a list of contacts related to this particular account.
- **Map to record value** – Toggle to display data for a particular record from the cache for the **Input components**. For example, enable this toggle when you create an Edit type page because you want to display actual data to end users before these values are changed. On the other hand, if you are creating a Create Record type page, do not enable Map to record value because this page

is not tied to any existing record. Thus, it does not make sense to map your **Input components** to any record.

- **Select type** – Select source of options for the **Select component** and **Radio component**.
    - If **Enter values manually** is selected, then **Value** and **Label** must be completed. Value is then used in the service action and Label is what the end user sees in Citrix Workspace.
    - If **Select from database** is selected, then you must complete these fields: **Data source for options**, **Data table for options**, **Data column for option label**, and **Data column for option value**. Data column for option value is then used in the service action and Data column for option label is what the end user sees in Citrix Workspace.
    - A combination of **Select type** and **Map to record value** can be used for different use cases. For example, Select from database together with enabled Map to record value is usually used for Edit type pages. In this scenario, Data table and Data column is used to display current data, Data source for options, Data table for options, Data column for option label, and Data column for option value is used to display all other available options from the cache which the end user can use while editing a record.
- **Data source** - Select the data source for the displayed element.
    - Use **Column value** if you want to display one particular column from the cache.
    - Use **Template** if you want to display a string of various attributes from the cache. For example, account address is split in the cache into 3 columns, but you want to display them all together in one component.
- **Data table** - Select the data table for the displayed element.
- **Data column** - Select the data column for the displayed element.
- **Conditional format** - Configure conditional formatting for the element.
- **Insert variables** - Add variable to the page element to automatically display application integration data.
- **Data filter** - Set constraints of displayed data. Select **Set filter**, then **Add**, and add the condition constraints as required.
- **Data order/Order** – Set order of displayed data. Select **Set order**, then **Add rule**, and the rule constraints as required.
- **Display additional data column** toggle – Used for the **Lookup component** to define an extra column to be shown to users while performing search.
- **Search term matching strategy** – Used for the **Lookup component** to define search strategy.
    - **Starts with** - This search method is the fastest as it does not overload the cache. It searches first characters of values in the defined cache attribute. This search method is satisfactory for most use cases.

  - **Contains** – Depending on the size of your data collection, this search method can be very slow. It searches through all characters of values in the defined cache attribute.

- **Field width** - Used for **Input components** to define the width of the input field.
- **Required** toggle - Used for **Input components** so that the end user is prompted to enter data before the button with configured service action is actionable.
- **Enabled** toggle - Used for **Input components** to display data for a particular record that you do not want end users to have the ability to change.
- **Visible** toggle - Used in case you need extra data for Page logic, Go To Page and so forth, but you don't want this information to be shown to the end user.
- **Hide if empty** – Used for some **Display components** when you do not want to show this component at all to end user if no data is available.
- **Alt. text/Alt attribute** – Used for **Image** component. Enter the text that shows if there is a problem displaying the image.
- **URL prefix** – Used for **Image** component to configure static prefix for a URL while the rest of the URL is taken from **Data table** and **Data column** fields.
- **Image size** – Select the size of the displayed image. For example, **Thumbnail** or **Full width**.
- **Image shape** – Select the shape of the image. For example, **Circle** or **Rounded corners**.
- **Layout** - Select from preconfigured layouts for the **List component**.

Input validation rules

Add rules for some input components to restrict format of data user can type in. Each component has its own specific validation rules available. There is always a minimum and maximum option to set.

This tab is available for these components: **Text input**, **Date/Time**, **Text area**, and **Num. input**. See each component description for more details.

Actions

The actions menu is available for the **Button** component to allow the microapp recipients (Workspace user) to respond with actionable input. Different actions are present for different components.

Enable Page action button

Enable the **Page action button** toggle to display the button component in the footer of the page blade in Citrix Workspace. A different subset of actions is available for the Button component based on this toggle. Consider the following:

- If the button is displayed in the blade footer, the blade is automatically closed after the user selects the footer button in Citrix Workspace.
- If you want to add actions such as Go To Page or Go to URL, you don't want to show buttons in the blade footer so that Citrix Workspace can navigate end users to the next screen. Such actions are not available if you enable this toggle.

Add action

Define the action that the button runs from the configured service actions that you configured in the integration. Actions include:

- **Run service action** - Define the action that the button runs from configured service actions that are set up in the integration. See Configure Service action parameters for an example.
- **Send email** - Sends an email based on pre-configured attributes. See Add a Send Email action for an example.
- **Add G Suite event** - Creates Google Calendar events based on pre-configured attributes.
- **Back** - Navigates user to the previous page.
- **Go to page** - Navigates user to a pre-configured microapp page. See Add a Go to Page action for an example.
- **Go to URL** - Navigates user to a pre-configured URL. See Add a Go to URL action for an example.
- **Run notification trigger** - Runs a pre-configured notification event. See Add a Run notification trigger action for an example.

Add a Go to Page action

Define an action to send users to a pre-configured microapp page.

1. After adding a **Button** component to the builder and giving it a name under the **Button properties** tab, select the **Actions** tab.
2. Disable the **Page action button** toggle. In the **Add action** field, select **Go to Page**.
3. Select **Go to page** under **Actions**. **Action label** field, **App** selector, and **Page** selector open.
4. Under **App**, select the microapp that you want to choose the page from.
5. Under **Page**, select the page that you want the button to open.
6. (Optional) Under **Target page record** select **Set conditions** if you want to filter data in the configured page. For example, a user is viewing a page with account data. The user selects a button labeled *Opportunities* and, based on the target page record conditions, is navigated to the page with a list of all related opportunities with expected value higher than a certain value.
7. (Optional) Under **Populate target page** select **Edit fields** if you want to pre-populate fields in the target page. For example, a user is viewing a Jira ticket and wants to create a new one in the same project. The user selects a button labeled *New* and is navigated to page where the Project

field is pre-populated with the value from the previous page but all other attributes must be entered manually.

Add a Run notification trigger action

Define an action to trigger a notification to run an evaluation of notification events and send notification messages to the target audience. This action can be displayed in the page body or in the blade footer.

1. After adding a **Button** component to the builder and giving it a name under the **Button properties** tab, select the **Actions** tab.
2. Leave the **Page action button** toggle enabled to show the button in the footer of the page. Disable the toggle to show the button in the body of the page.
3. In the **Add action** field, select **Run notification trigger**.
4. Select **Run notification trigger** under **Actions**.
5. Under **Events**, select the event trigger that you want to run. You can select more than one event for this button.

Use component values as parameters

You can use component values as parameters in **Send Email** and **Go To URL** actions. This feature allows:

- (Send Email) Workspace users can enter email recipients in a Workspace field for a given action.
- (Go To URL) User's input from a Workspace form can be used as a part of a URL template opened in Workspace.

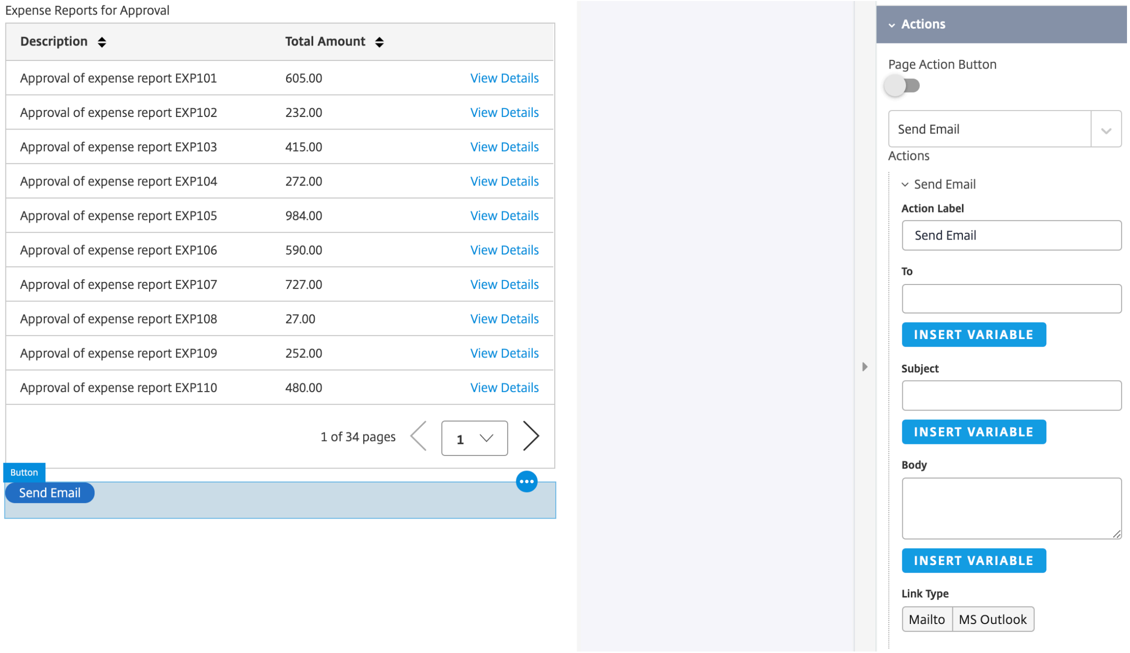
Follow the steps below according to your use-case.

Add a Send Email action

1. After adding a **Button** component to the builder and naming under **Button properties**, select the **Actions** tab.

2. Disable the **Page action button** toggle. In the **Add action** field, select **Send Email**.

3. Under **To**, select **INSERT VARIABLE**.



Expense Reports for Approval

Description	Total Amount	
Approval of expense report EXP101	605.00	View Details
Approval of expense report EXP102	232.00	View Details
Approval of expense report EXP103	415.00	View Details
Approval of expense report EXP104	272.00	View Details
Approval of expense report EXP105	984.00	View Details
Approval of expense report EXP106	590.00	View Details
Approval of expense report EXP107	727.00	View Details
Approval of expense report EXP108	27.00	View Details
Approval of expense report EXP109	252.00	View Details
Approval of expense report EXP110	480.00	View Details

1 of 34 pages 1

Actions

Page Action Button

Send Email

Actions

Send Email

Action Label: Send Email

To

Subject

Body

Link Type: Mailto | MS Outlook

4. On the **Insert Variable** screen, from the **Type** menu select **Component value**.

5. Under **Component**, select your required input component that a user enters on their Workspace form. The **To** field populates with the component ID.

6. Add a **Subject** and **Body** for the message as required.
   In Workspace, users can enter an email address in the field enabled by this component. When they submit the email address, their email opens with a prepopulated message that can be modified if necessary and sent.

Add a Go to URL action

1. After adding a **Button** component to the builder and naming under **Button properties**, select the **Actions** tab.
2. Leave the **Page action button** toggle enabled. In the **Add action** field, select **Go To URL**.
3. In the **URL Template** field, enter the URL of target site that you want to open.
4. Under **URL Template** field, select **INSERT VARIABLE**.
5. On the **Insert Variable** screen, from the **Type** menu select **Component value**.
6. Under **Component**, select your required component. The **URL Template** field populates with the component ID added to the URL you entered.
   In Workspace, users can enter a value in the field enabled by this component. When they submit the query, the selected component value is used as a variable in the URL that is opened in their browser.

> **Note:**
>
> Select the info icon next to a template field to view detailed annotation about used components.

Configure Power BI component service actions

The out-of-the-box microapps that come with the Power BI template have the components configured as needed. If you want to make changes or add other microapps, follow these steps as a model. Important considerations include:

- To display a report the first time, users must log in to Power BI.
- There is a limit to the number of embed tokens a Power BI master account can generate. You can purchase more capacity. For more information, see https://docs.microsoft.com/en-us/power-bi/developer/embedded/embedded-faq#technical.
- AS Azure or AS OnPrem live connection reports may experience a delay after rebinding. For more information, see: https://docs.microsoft.com/en-us/rest/api/power-bi/reports/RebindReport.

The Power BI component setup is needed to authorize the logged in user before they can view a dashboard, report, or tile. To set this up, you need to configure a service action to generate a token for the user. Configure this for Dashboards, Reports, and Tiles. Authorization works as a regular Service Action. For example, if you have a separated OAuth 2.0 authentication method for Service Actions setup, the user is asked to log in to Power BI and only then the component will show the desired content.

1. In the **Edit** screen for an integration, or from the menu, select **Service Actions** from the left side navigation column.
2. Configure these new service actions as required using these Power BI endpoints:
   - Dashboards `https://docs.microsoft.com/en-us/rest/api/power-bi/embedtoken/dashboards_generatetokeningroup`
   - Reports `https://docs.microsoft.com/en-us/rest/api/power-bi/embedtoken/reports_generatetokeningroup`
   - Tiles `https://docs.microsoft.com/en-us/rest/api/power-bi/embedtoken/tiles_generatetokeningroup`
3. Select **Add service action**. This action needs groupId and dashboardId, reportId, and tileId parameters. Use the model below as an example:
4. Under **Action execution** select the **BODY** tab. Select **JSON** from the **Content type** list.

   ```
   {
   “accessLevel”: “View”
   }
   ```

   Now set up authorization in the component using this newly configured service action. Follow the general example below:

5. In the microapp, for example a microapp where you are pulling data for dashboards, select any page where you have the Power BI component.
6. Select the **Power BI** component, and then the **Power BI Authorization** tab on the right-hand side.
7. Select **Edit parameters**, and complete the fields as you see below:
8. Select **Save** to finish.

Build a list page

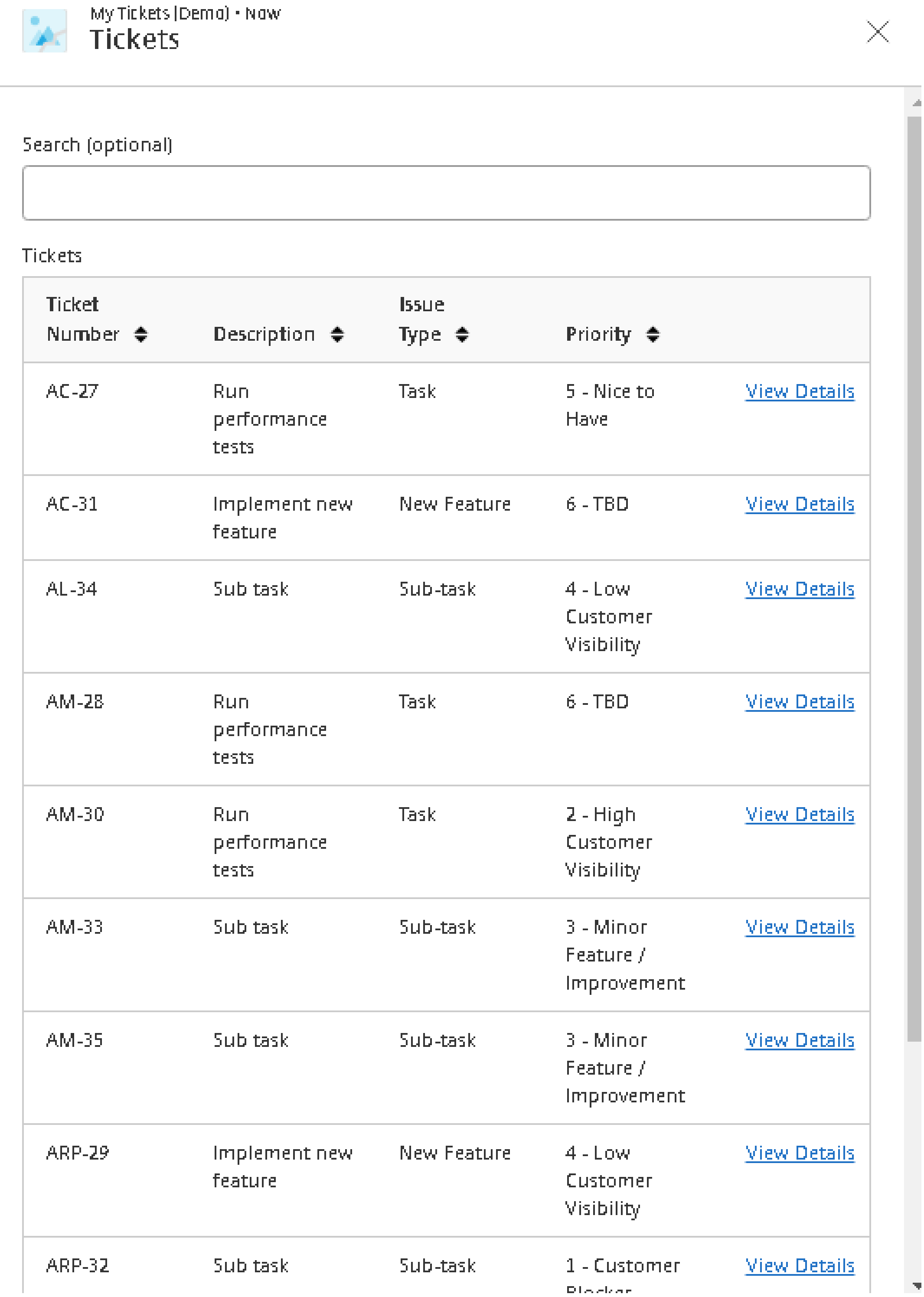
April 28, 2021

Build a list page to show all records available in your system of record. This can be defined, for example, as all issues belonging to a particular user (personalized) or all in a particular project. This article assumes that you have already created your microapp for this workflow. For step-by-step details, see Add a new microapp.

To add a list page for your microapp, select from the starting templates then customize the page in the builder. For this *List page*, start with a **Table** template which has the **Table** builder components already available. Then add a **Text Input** component to search the table. This page uses the following builder components:

- **Table** - Add a table by defining table source, filters, and defining columns. Page link actions can be added. Personalized queries must be set to limit data exposure.
- **Text Input** - Use this component as a search input. This means that you do not define text source or default.

The following image shows an example list page showing ticket details with a link to a detail page that we built with the components listed above:



My Tickets (Demo) • Now

Tickets

Search (optional)

Tickets

Ticket Number	Description	Issue Type	Priority	
AC-27	Run performance tests	Task	5 - Nice to Have	View Details
AC-31	Implement new feature	New Feature	6 - TBD	View Details
AL-34	Sub task	Sub-task	4 - Low Customer Visibility	View Details
AM-28	Run performance tests	Task	6 - TBD	View Details
AM-30	Run performance tests	Task	2 - High Customer Visibility	View Details
AM-33	Sub task	Sub-task	3 - Minor Feature / Improvement	View Details
AM-35	Sub task	Sub-task	3 - Minor Feature / Improvement	View Details
ARP-29	Implement new feature	New Feature	4 - Low Customer Visibility	View Details
ARP-32	Sub task	Sub-task	1 - Customer Blocker	View Details

> **Important:**
>
> A table without a personalized query exposes all records that match a table filter regardless of the data relation to subscribers. Your potentially sensitive data may be exposed to all microapp

> users unless you limit data exposure. Control data access for pages using the **Data filter** feature.
>
> In a page, select the table in the component builder pane. Select **Table Properties** and select **Data filter**. Then select **+Add Condition** and add the constraints as required.

Follow these steps:

1. Select the microapp that you want to add a page to. Select **Pages**, and **Add New Page**.
2. Give the *Page* a name, select the **Table** template.
3. Confirm your **Data source** and select the **Data table** you want the records in the table to be from. Select **Select Fields** to choose fields that populate your page. Select **Add**.

   The new page is added to the **Pages** list and is ready to be customized. The builder page populates with the fields we selected. Now let's customize the page.
4. To add a search box, drag a **Text Input** component to the builder and place it above the table component.
   - Select the **Text Input Properties** tab. In the **Label** field, enter **Search**.
   - Disable the **Map to Data Column** toggle.
   - Do not set a **Default Value**.
   - For this field, do not activate the **Required** toggle.
5. Select the **Table** component that is already available in the builder.
   - Select the **Table Properties** tab. In the **Label** field, enter *Tickets*.
   - Select the **Data Table** that you want the table to show. You must add the columns that you want to display in the table. See the screenshot above for a model of what we want to add.
   - Under **Data Filter**, select **SET FILTER** to map to the columns where the search is performed.
     a) Select **Add**.
     b) Select a value for **Select column**.
     c) Select *contains* for **Action**. This retrieves more results for partial or unfinished string inputs.
     d) Select *component value* for **Value type**.
     e) Select *Search Text input* for **Component**.
     f) Enable **Only with value** toggle.
     g) (Optional) You can add additional filters. For example, filter for active records. Also, you can combine the filter rules either by selecting **ALL**, **ONE OFF**, or by writing logical expressions.
     h) Select **Save**.
   - Select **Columns**, and select the first item you see listed. Under **Column Title** give the column a name. For example, we'll name the column *Ticket Number*, but the data column we map to is labeled *issue_key*. Select a value for **Data Table**, which is pre-populated, and

Data Column to map to the correct column.
   - Select **+**, and repeat for the following columns that we label: *Description*, *Issue Type*, and *Priority*.
   - Select the **Actions** tab, and select the *Details* page we already created. This places a **View Details** link on the right-hand side, and builds a link to a page populated with details of the given issue.

   You finished building the list page. As a final step, let's select this page as the action page for the microapp, and make it visible in the list of actions for this integration.

6. Select **Properties**. This is the cog on the left when you are in the builder.
7. Under **Actions**, select the **Enable as Action** toggle and select this page in the **Action page** menu.

   This list page is now ready.

Build a detail page

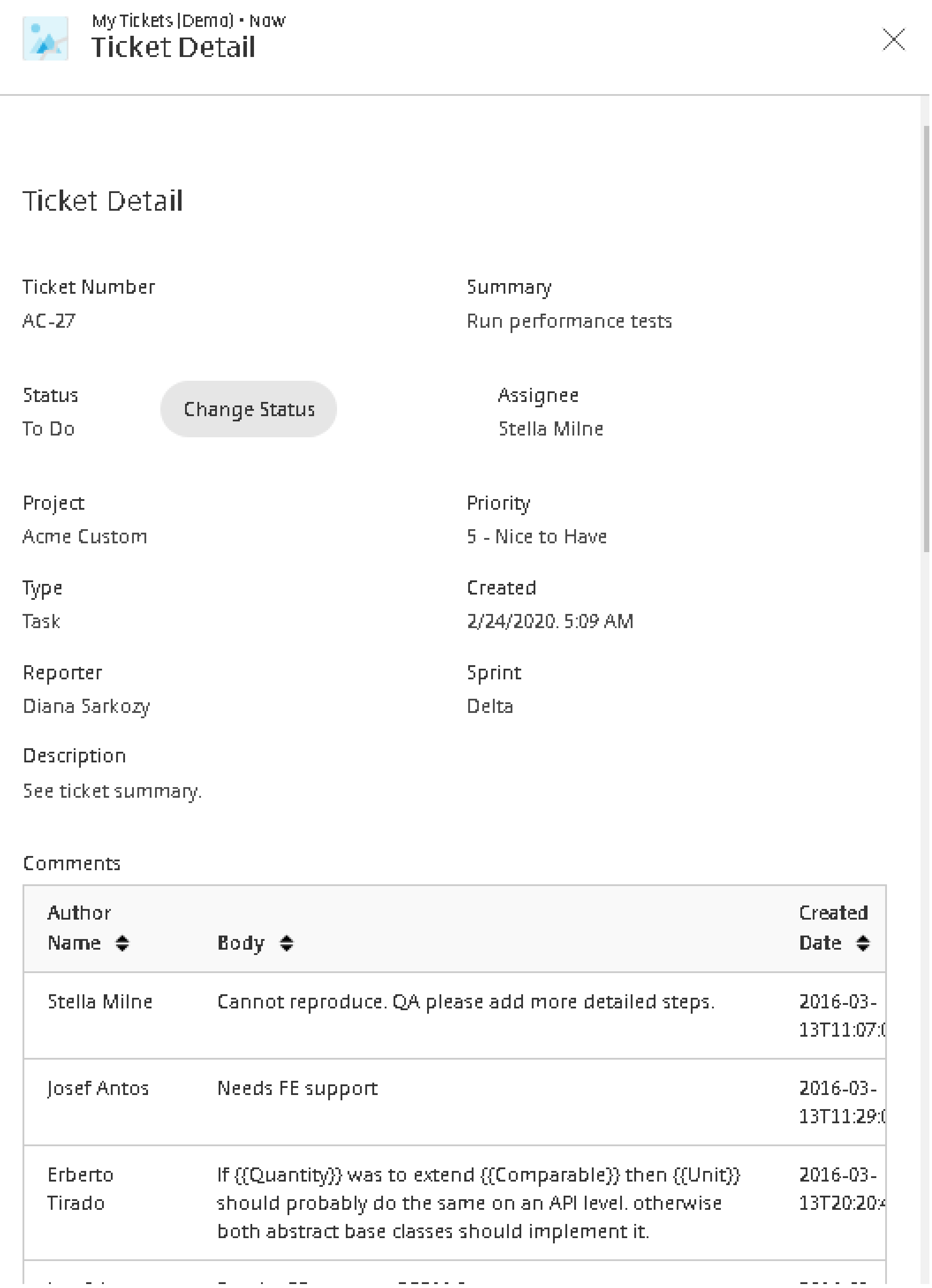
April 28, 2021

Build a detail page to view details of one record that is available in your system of record. Remember, you can design and customize these pages for your needs. This article assumes that you have already created your microapp for this workflow. For step-by-step details, see Add a new microapp.

To add a detail page for your microapp, select from the starting templates then customize the page in the builder. For this *Detail page*, start with a **Detail** template which pre-populates the builder with **Text** components showing the fields we selected. Use the following components to build this sample detail page.

- **Text** - Define text source and formatting to load from the cache to the page.
- **Back Button** - Allows users to go back to previous page.
- **Static Text** - Define static text to appear on the page.
- **Flexible Grid** - Gives you more control over the positioning of components on your pages. Helpful when you're designing pages intended for devices with larger screens. Set the label and the total number of cells you want in your grid.
- **Table** - Add a table by defining table source, filters, and defining columns. Page link actions can be added. Personalized queries based on users' emails may be set to limit data exposure.

The following image shows an example detail page showing ticket details that we built with the components listed above:



My Tickets (Demo) • Now

Ticket Detail

Ticket Detail

Ticket Number
AC-27

Summary
Run performance tests

Status
To Do

Assignee
Stella Milne

Project
Acme Custom

Priority
5 - Nice to Have

Type
Task

Created
2/24/2020, 5:09 AM

Reporter
Diana Sarkozy

Sprint
Delta

Description
See ticket summary.

Comments

Author Name ⬍	Body ⬍	Created Date ⬍
Stella Milne	Cannot reproduce. QA please add more detailed steps.	2016-03-13T11:07:(
Josef Antos	Needs FE support	2016-03-13T11:29:(
Erberto Tirado	If {{Quantity}} was to extend {{Comparable}} then {{Unit}} should probably do the same on an API level. otherwise both abstract base classes should implement it.	2016-03-13T20:20:4

Follow these steps:

1. Select the microapp that you want to add a page to. Select **Pages**, and **Add New Page**.
2. Give the *Page* a name and select the **Detail** template.
3. Confirm your **Data source** and select the **Data table** you want the records in the table to be

from. Select **Select Fields** to choose fields that populate your page. Select **Add**.

The new page is added to the **Pages** list and is ready to be customized. The builder page populates with the fields we selected. Now let's customize the page.

4. Select and drag the **Back Button** element to the top of the builder panel.
5. Select and drag the **Static Text** element to the top of the builder panel under the back button.
   - Select **Static Text Properties**, and in the **Content** field, enter *Ticket Detail*.
   - Under **Text Type**, select **Header**.
6. Select and drag the **Flexible Grid** element to the builder panel. Use the **Grid Items** to place our existing **Text** components.
   - We need to add new cells. Select the **Flexible Grid Properties** tab, and under **Total Number of Cells** change the value to *8*.
   - Select and drag the existing **Text** components to the location in the **Flexible Grid** where you want to place them.
7. Next, select and drag a **Table** component to the builder. Place it at the bottom.
   - Select the **Table Properties** tab. In the **Label** field, enter *Comments*.
   - Activate the **Use Records Related to Detail Page** toggle.
   - Select the **Data Table** that you want the table to show. In this case **Comments**
     You must add the columns that you want to display in the table. See the screenshot above for a model of what we want to add.
   - Select **Columns**, and select the first item you see listed. Under **Column Title** give the column a name. For example, let's name the column *Author Name*. Select a value for **Data Table**, which is pre-populated, and select the **Data Column** to map to the correct column.
   - Select **+**, and repeat for the following columns that we will label: *Body* and *Created Date*. For the date column, select **Format** to specify the time format used. You can, for example, build an action to a comment detail page, if necessary.
   - Select **Set Filter** if you need to filter data in your table based on certain conditions.
   - Select **Set Order** to view your table items in a desired order.

   This detail page is now ready.

Build a create page

April 28, 2021

Build a create page to add records into your system of record. This article assumes that you have already created your microapp for this workflow. For step-by-step details, see Add a new microapp.

We recommend housing this page in a separate microapp for these reasons. Keep these considerations in mind when designing your workflow:

- You can have only one action per microapp. Meaning, you cannot have a search page and create page in the same microapp if you want them both as actions.
- To allow for different user permission settings, if needed.

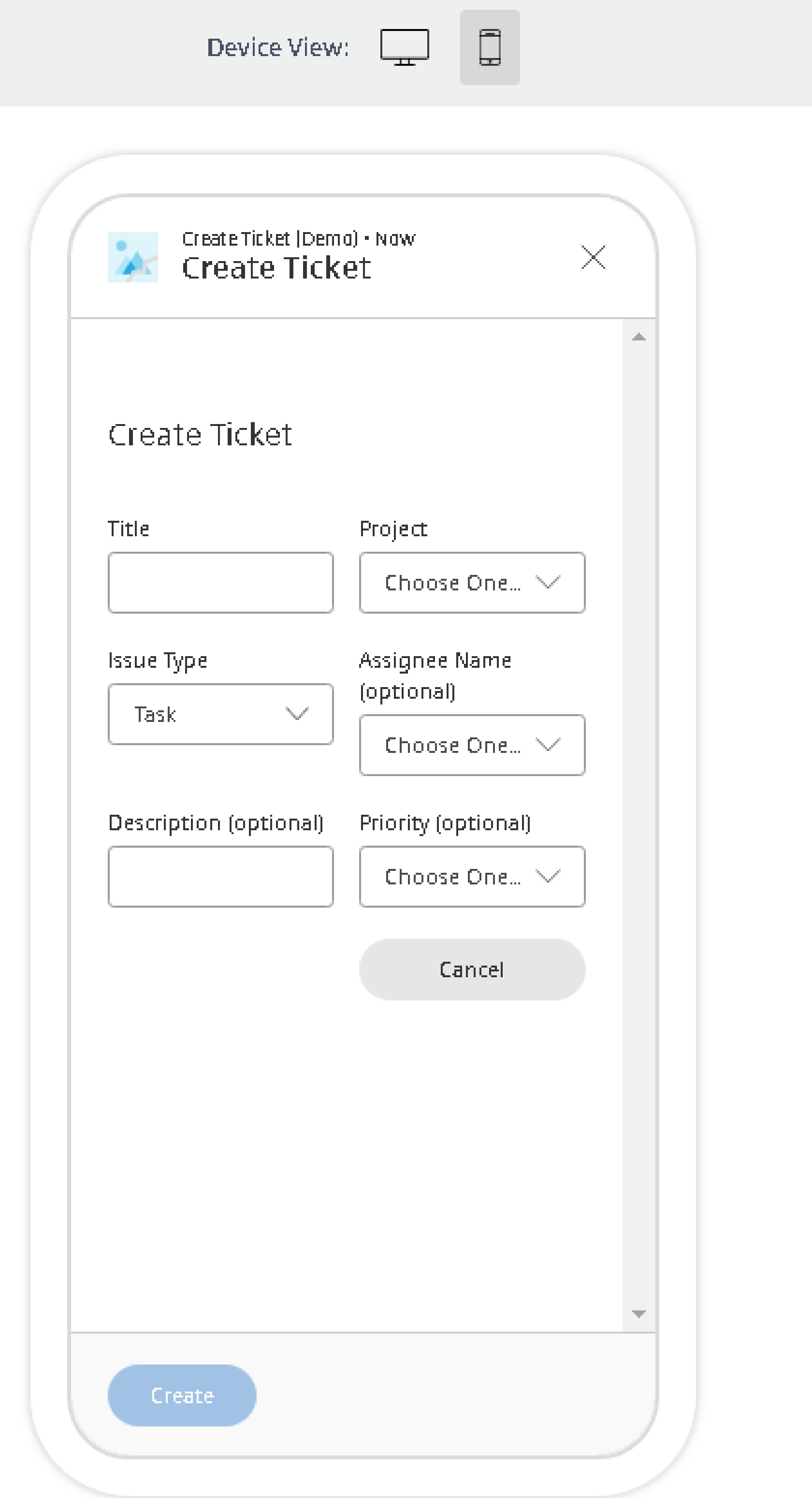
> **Note:**
>
> Create functionality is limited based on API write-back access.

For this *Create page*, start with a **Form** template, and then use the following builder components:

- **Static Text** - Define static text to appear the page.
- **Flexible Grid** - Gives you more control over the positioning of components on your pages. Helpful when you're designing pages intended for devices with larger screens. Consists of **Grid Items**. Set the label and the total number of cells you want in your grid.
- **Text Input** - Define text source by specifying the data table, column, and value to load to the page entered by the user. Component can be marked as optional. Field width can be modified. Validation rules can be configured based on a minimum or maximum length or text pattern to identify user input. If this component is not mapped to record value, users use the field to input text.
- **Select** - Allows users to choose from a set list of values. Populated by data from the source system or you can enter the list of values manually. Actions can be added.
- **Lookup** - Allows users to search though a large quantity of values and allows users to select a value by searching for something else.
- **Button** - Add a clickable component on the page with actions and logic.

The following image shows an example create page showing details mapped to the data columns listed below that we built with the components listed above. For this page, and microapp, we need to map to the following data columns:

- project
- issue type
- priority
- assignee name



Follow these steps:

1. After you have added the microapp specifically for this create action, select that microapp. For

step-by-step details, see Add a new microapp. Select **Pages**, and **Add New Page**.

2. Give the *Page* a name, and select the **Form** template.
3. Confirm your **Data source** and select the **Data table** that you want to access. Select **Add**.

   The new page is added to the **Pages** list and is ready to be customized. The builder page populates with the fields we selected. Now let's customize the page.

4. Select and drag the **Static Text** element to the top of the builder panel.

   - Select **Static Text Properties**, and in the **Content** field, enter *Create Ticket*.
   - Under **Text Type**, select **Header**.

5. Select and drag the **Flexible Grid** element to the builder panel. Use the **Grid Items** to place our other components. You need to add new cells. Select the **Flexible Grid Properties** tab, and under **Total Number of Cells** change the value to *8*.
6. To add a *Title* field, drag a **Text Input** component to the top-left **Grid Item**.

   - Select the **Text Input Properties** tab. In the **Label** field, enter *Title*.
   - Disable the **Map to Record Value** toggle.
   - Activate the **Required** toggle.

7. To add a *Project* drop-down selector, drag a **Select** component to the top-right **Grid Item**.

   - Select the **Select Properties** tab. In the **Label** field, enter *Projects*.
   - Under **Select Type**, select **Select from Database**.
   - Disable the **Map to Record Value** toggle.
   - Select **Data Table**, **Data Column** from the menus. In our case, *project* and *id*.
   - Activate the **Required** toggle.

8. To add an *Issue type* drop-down selector, drag a **Select** component to the middle-left **Grid Item**.

   - Select the **Select Properties** tab. In the **Label** field, enter *Issue Type*.
   - Under **Select Type**, select **Select from Database**.
   - Disable the **Map to Record Value** toggle.
   - Select **Data Table**, **Data Column** from the menus. In our case, *issue_type* and *name*.
   - Activate the **Required** toggle.

9. To add an *Assignee name* searchable field, drag a **Lookup** component to the middle-right **Grid Item**. Use this component because it allows users to search easily though a large quantity of values.

   - Select the **Lookup Properties** tab. In the **Label** field, enter *Assignee Name*.
   - Under **Select Type**, select **Select from Database**.
   - Select **Data Table to Search**, **Data Column to Search**, and **Data Column to Use as Value** from the menus. In our case, *user* and *display_name*.
   - Disable the **Display Additional Data Column** toggle.

10. To add a *Description* field, drag a **Text Input** component to the lower-middle-left **Grid Item**.

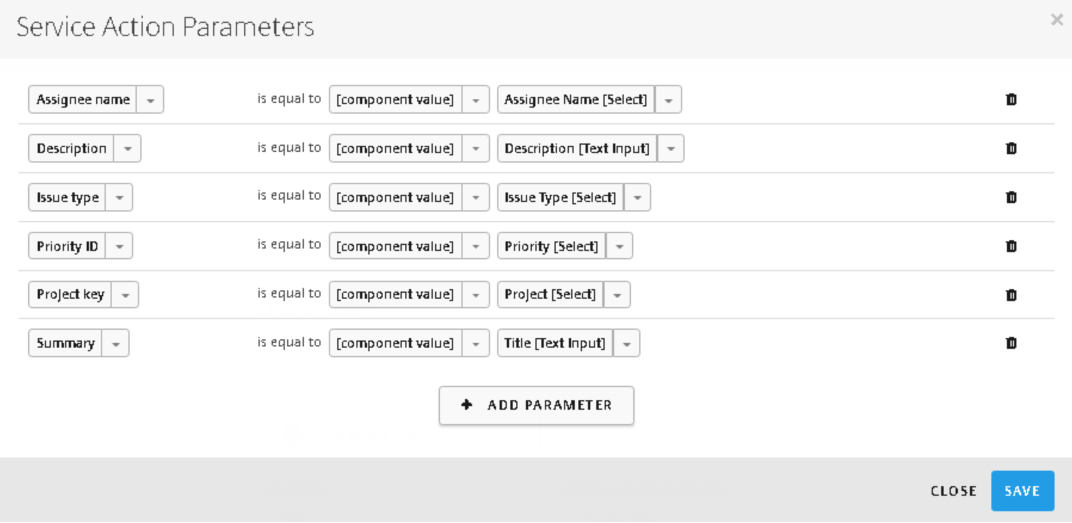
    - Select the **Text Input Properties** tab. In the **Label** field, enter *Description*.
    - Disable the **Map to Record Value** toggle.
    - For this field, do not activate the **Required** toggle to automatically add an **(optional)** tag to the field.

11. To add a *Priority* drop-down selector, drag a **Select** component to the lower-middle-right **Grid Item**.

    - Select the **Select Properties** tab. In the **Label** field, enter *Priority*.
    - Under **Select Type**, select **Select from Database**.
    - Disable the **Map to Record Value** toggle.
    - Select **Data Table**, **Data Column** from the menus. In our case, *priority* and *name*.
    - For this field, do not activate the **Required** toggle. This automatically adds an **(optional)** tag to the field.

12. To add a *Create* button, drag a **Button** component to the bottom-left **Grid Item**.

    - Select the **Button Properties** tab. In the **Label** field, enter *Create*.
    - Leave the **Style** option as **Primary** to make it a blue option button.
    - Select the **Actions** tab. Ensure the **Page Action Button** toggle is enabled. This displays the button in the footer of the blade and closes the Workspace blade after the action is completed.
    - Click the **Add Action** drop-down and select **Run Service Action**.
    - Click the **Run Service Action** text. Click the **Data** drop-down, and select the integration you want to connect to. Click the **Action** drop-down, and select the action you want to take, in this case **Create Issue**.
    - Select **EDIT PARAMETERS**, and complete all required parameters based on the fields you created for the page. You can model yours after this example:



Assignee name	is equal to	[component value]	Assignee Name [Select]
Description	is equal to	[component value]	Description [Text Input]
Issue type	is equal to	[component value]	Issue Type [Select]
Priority ID	is equal to	[component value]	Priority [Select]
Project key	is equal to	[component value]	Project [Select]
Summary	is equal to	[component value]	Title [Text Input]

13. To add a *Cancel* button, drag a **Button** component to the bottom-right **Grid Item**. This allows users to reload the page without submitting changes.

    - Select the **Button Properties** tab. In the **Label** field, enter *Cancel*.
    - Select the **Style** option as **Secondary** to make it a gray option button.
    - Select the **Actions** tab. Click the **Add Action** drop-down and select **Go to page**. Select this microapp for **App** and the name of this page you are creating for **Page** to make this page refresh itself when you select cancel.

    You finished building the create page. As a final step, let's select this page as the action page for the microapp, and make it visible in the list of actions for this integration.

14. Select **Properties**. This is the cog on the left when you are in the builder.

15. Under **Actions**, select the **Enable as Action** toggle and select this page in the **Action page** menu.

    This create page is now ready.

Build event notifications

November 2, 2021

Create triggers for events to be sent to the client application, such as new PTO request or notification that a record changed. Select from the template types, then customize the event in the builder. This article assumes that you have already created your microapp for this workflow. For step-by-step details, see Add a new microapp.

Follow these steps to build an event notification:

- Create the conditions to send the notification/action and the target subscribers.
- Configure the notification card for subscribers.
- Configure action items.
- Set the expiration condition and time period.

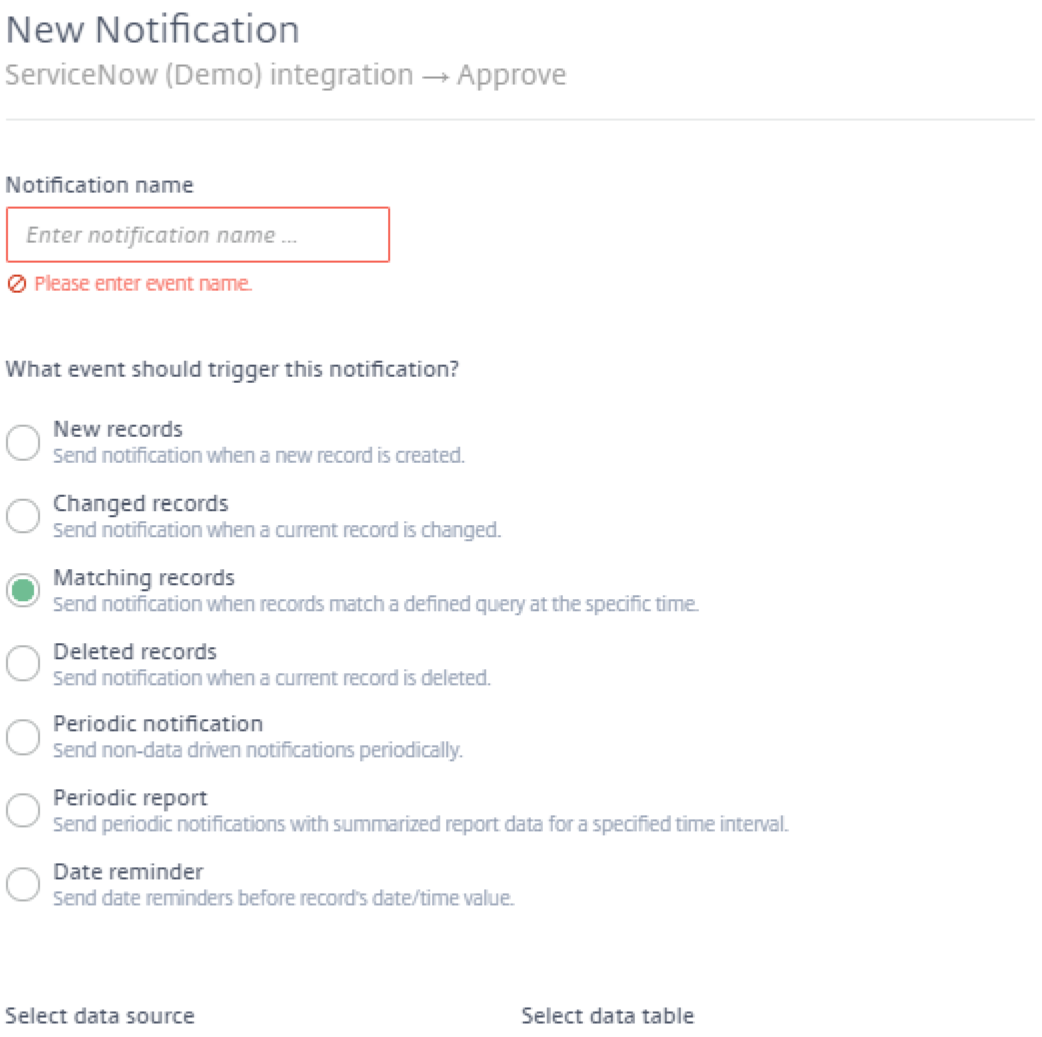
> **Note:**
>
> When editing the settings of an existing notification, stop all synchronization for that particular integration before trying to save.

Create a notification

When setting a **Periodic Notification**, **Periodic Report** or **Date Reminder** the following behavior applies:

- When scheduling a **time interval**, the interval is set to run upon completion of the previous run. For example, a notification is set to 5 minutes, the notification runs at 10.00, completes at 10.02, and then runs again at 10.07.
- When scheduling a **daily** notification, the notification runs at a random time selected within the time frame. For example, a notification is set to run at 14.00, the run begins randomly between 14.00 and 14.05.

1. Select the microapp that you want to add an event to. Select **Add Notification** at the top-right of the page.
2. Enter a **Notification name** for the notification event.
3. Select your desired trigger and notification type from the following. The set-up steps differ slightly depending on the specific event trigger type you select:
    - **New records** - Sends a notification when a new record is created in the source of record (SoR).
    - **Changed records** - Sends a notification when an existing record is changed in the SoR.
    - **Matching record** - Sends a notification when records match a defined query at the specific time in the SoR.
    - **Delete records** - Sends a notification when a current record is deleted in the SoR.
    - **Periodic notification** - (user action) Sends non-data driven notifications periodically.
    - **Periodic report** - Sends periodic notifications with summarized report data (grouping) for a specified time interval.
    - **Date reminder** - Sends a notification at the specified time before or after the records date column value.



New Notification

ServiceNow (Demo) integration → Approve

Notification name

Enter notification name ...

⊘ Please enter event name.

What event should trigger this notification?

○ New records
Send notification when a new record is created.

○ Changed records
Send notification when a current record is changed.

● Matching records
Send notification when records match a defined query at the specific time.

○ Deleted records
Send notification when a current record is deleted.

○ Periodic notification
Send non-data driven notifications periodically.

○ Periodic report
Send periodic notifications with summarized report data for a specified time interval.

○ Date reminder
Send date reminders before record's date/time value.

Select data source

Select data table

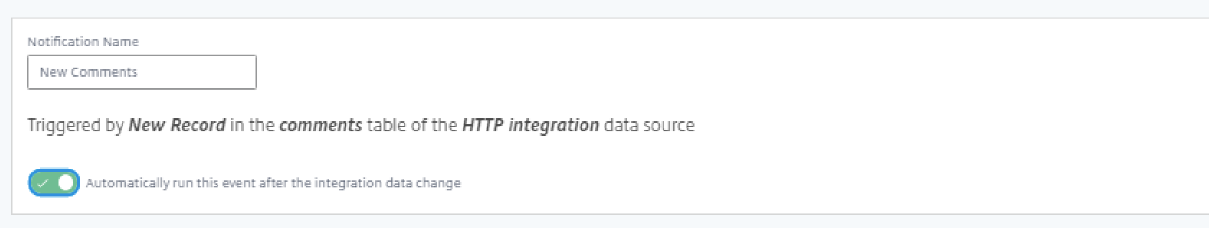
4. Confirm your **Data source** and select the **Data table** from which you want to track changes. Select **Add**. The new notification is added to the Notifications list and is ready to be configured.

   > **Note:**
   >
   > As Citrix Workspace Microapps supports cross-integration microapps, the **Select data source** list shows all available integration data sources.

The **Edit Notification** screen opens. Follow the steps below to configure the notification. When your notification is configured correctly, you must scroll to the top of the page and select **Save**.

Notification name

Notification Name lets you input the notification name and optionally select whether the notification event is run immediately after synchronization of your target application integration.



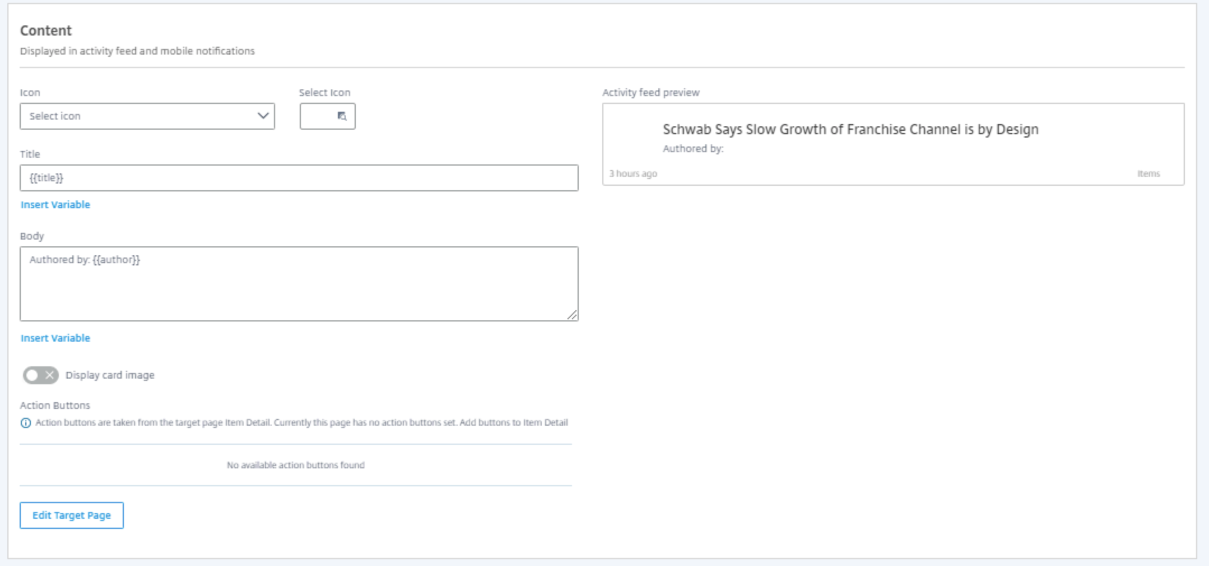
Notification Name: New Comments

Triggered by ***New Record*** in the ***comments*** table of the ***HTTP integration*** data source

Automatically run this event after the integration data change

Content

Content lets you configure the information displayed on your notification. Here you can configure the notification icon, notification title, and body content in addition to the display card image. You can optionally enter variables to incorporate elements generated from your target application integration.



Content

Displayed in activity feed and mobile notifications

Icon: Select icon

Title: {{title}}

Body: Authored by: {{author}}

Activity feed preview

Schwab Says Slow Growth of Franchise Channel is by Design

Authored by:

3 hours ago

Items

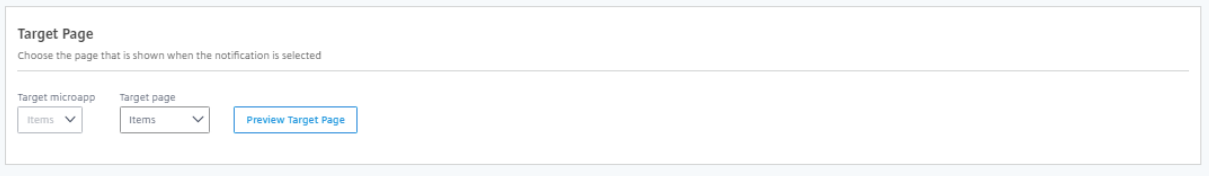
Action Buttons

Action buttons are taken from the target page Item Detail. Currently this page has no action buttons set. Add buttons to Item Detail

No available action buttons found

Target Page

Target Page lets you choose the page that is shown when the notification is selected. You can select the target microapp, target page, and optionally preview the page to see what your microapp recipients receive.



Target Page

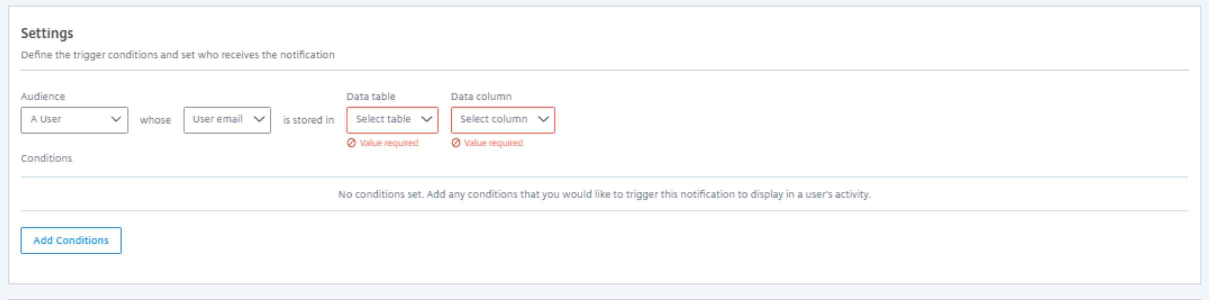
Choose the page that is shown when the notification is selected

Target microapp: Items

Target page: Items

Settings

Settings let you define the trigger conditions of your notification and the notification recipients. Select **Audience** to define the recipients of your notification from your integration and **Add conditions** to set what conditions trigger the notification for your users. After at least one condition is set, there is an option to **Edit conditions**.



Settings

Define the trigger conditions and set who receives the notification

Audience: A User whose User email is stored in Data table: Select table (Value required) Data column: Select column (Value required)

Conditions

No conditions set. Add any conditions that you would like to trigger this notification to display in a user's activity.

> **Note:**
>
> Complex boolean expressions are simplified when parsed internally after definition and are stripped of redundant formatting if applicable.
> For example, defining **1 AND (2 AND 3) OR (4 AND 5)** results in displaying **1 AND 2 AND 3 OR 4 AND 5** as the redundant brackets are removed from the definition.

Increase notification threshold

For better performance, the Microapps service limits the maximum number of notification cards that are generated per user per notification job. By default, this is set to 50, and any notification cards generated above this amount are lost.

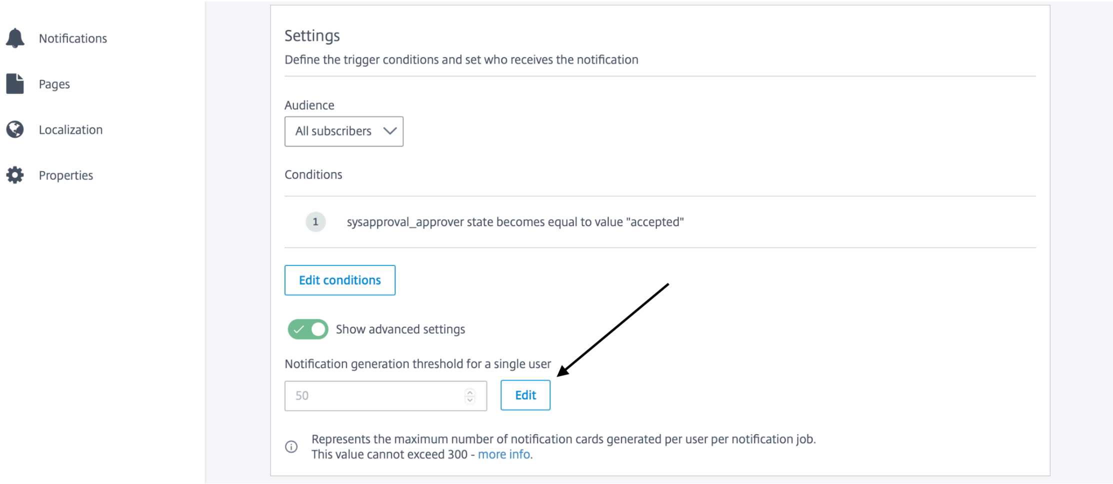
Using advanced settings options, you can increase this value. However, a large number of notifications can flood Workspace users' Activity Feed. Doing this can dilute the value of generating the notification cards and increases the chance that they are not read at all. Consider your users and their Workspace experience before increasing this value. The maximum value permitted is 300.

For example, you have an integration with a synchronization time set to every 10 minutes. Each subscribed user receives as many notifications as changes that occur during this time period, up to the threshold; be it the default 50 or up to 300 if you modify this value. Any additional changes that occur over this value are not generated as notifications for users' Activity Feed. After this period between synchronization elapses (10 minutes in our example), a new synchronization runs and captures the next 50/300 changes, according to this setting.

1. Under **Settings**, select **All subscribers** from the **Audience** menu. Advanced settings are only visible when the audience is set to all subscribers.

2. Enable the **Show advanced settings** toggle.

3. Under **Notification generation threshold for a single user**, select **Edit**. The default value of *50* is prefilled.
4. Enter a new value. Do not exceed the maximum: *300*.
5. Select **Save** at the top of the page to save all changes. The field will be disabled, and you must select **Edit** to enable the field again.



Settings

Define the trigger conditions and set who receives the notification

Audience: All subscribers

Conditions

1 sysapproval_approver state becomes equal to value "accepted"

Show advanced settings

Notification generation threshold for a single user: 50

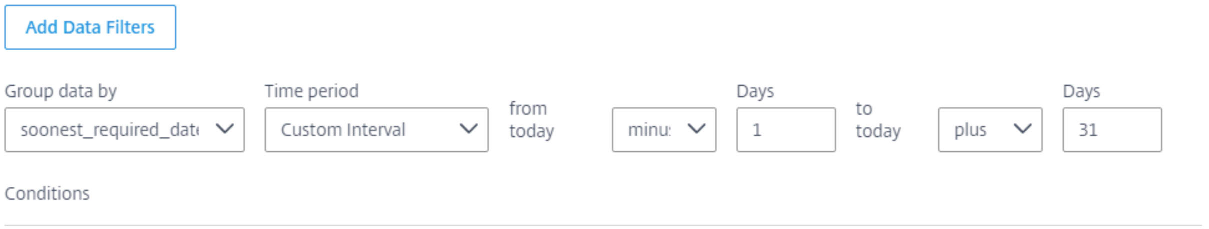
Represents the maximum number of notification cards generated per user per notification job. This value cannot exceed 300 - more info.

Grouping - Periodic report notifications

For **Periodic report** notifications, there is a grouping feature under settings. Use this feature to collect multiple events into one notification. For example:

- A user receives 10 work tickets during a given period (such as a day), but you want users to receive just one notification.
- A user clears many approvals during a work day, but you want them to receive a notification of any remaining approvals at the end of the day.

Choose a data column value for **Group data by**, and select a **Time period**. The **Time period** field defines the period of time from when you set the event to run in the **Run frequency** field above. For example, **Today** means that the group of notifications is sent the same day the event trigger is run. If you select **Custom Interval**, detailed **from** and **to** fields open.



Group data by: soonest_required_dat‹ Time period: Custom Interval from today minu: Days: 1 to today plus Days: 31

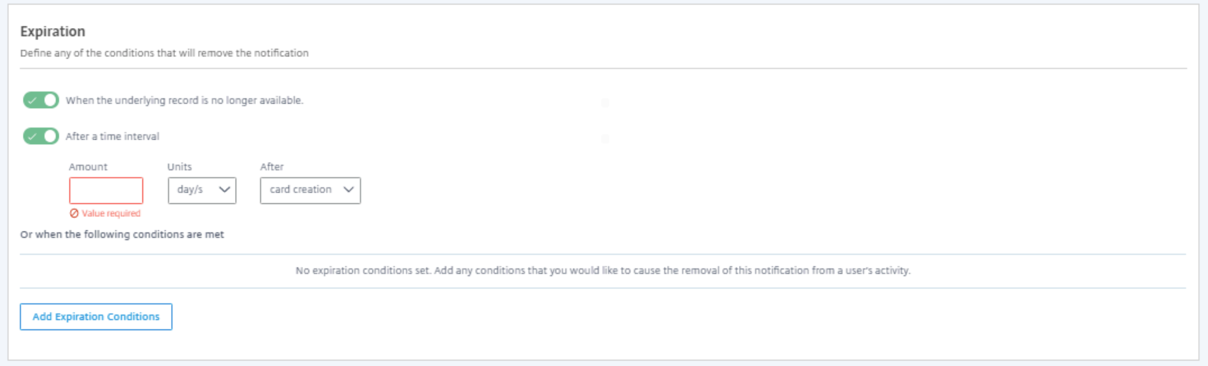
Conditions

Send a reminder - Date reminder notifications

For **Date reminder** notifications, the **Run frequency** field near the top of the page defines when the synchronization is run. The **Send a reminder** field under **Settings** defines how long before or after the event the notification is sent.

Expiration

Expiration lets you define any of the conditions to remove the notification. You can set to expire the notification when the record in your integration is no longer available. You can expire the notification after a defined interval. You can also configure integration trigger conditions to expire the notification when there is a change in data in your target integration.



Expiration

Define any of the conditions that will remove the notification

● When the underlying record is no longer available.

● After a time interval

Amount	Units	After
⊘ Value required	day/s	card creation

Or when the following conditions are met

No expiration conditions set. Add any conditions that you would like to cause the removal of this notification from a user's activity.

When your notification is configured correctly, scroll to the top of the page and select **Save**.

Run event

Select **Run Event** in the top bar of the notification builder to manually trigger this event notification to run. Select **Show Event Log** to view a history of changes categorized by severity. You can also **Run all** events from the top bar of the Notifications overview screen of the microapp.

Clear all notifications

To remove all notifications from a microapp, select **Clear all notifications** on the individual notification's edit menu on the Notifications overview screen of the microapp. This feature deletes your notifications when you need to reorganize or regenerate your notifications (for example, when testing) when using a newer data structure.

You can also remove all notifications in all microapps in an integration. From the **Microapp integrations** overview page, select the menu next to the integration for which you want to delete all notifications. Select **Clear all notifications**, and confirm.

Localize microapps

May 26, 2020

Citrix Workspace Microapps allows you to export and import translated JSON files for the purposes of localization. With microapp localization options you can export these files, edit them with the required localized language and import the localized microapp file back into the microapp platform for use by your microapp users.

Localization currently only supports a defined set of languages:

- English (default, fall back language for microapp)
- Chinese (simplified)
- Dutch
- French
- German
- Japanese
- Spanish
- Brazilian Portuguese
- Italian

Extra language support will be added in future updates. Once a microapp is localized to your desired language, the language is displayed based of the end user's browser locale.

Using the localization feature for your microapps involves the following:

1. Export your desired microapp configuration file.
2. Edit the file and translate the entities to the required language.
3. Import the translated file back into the microapp platform.

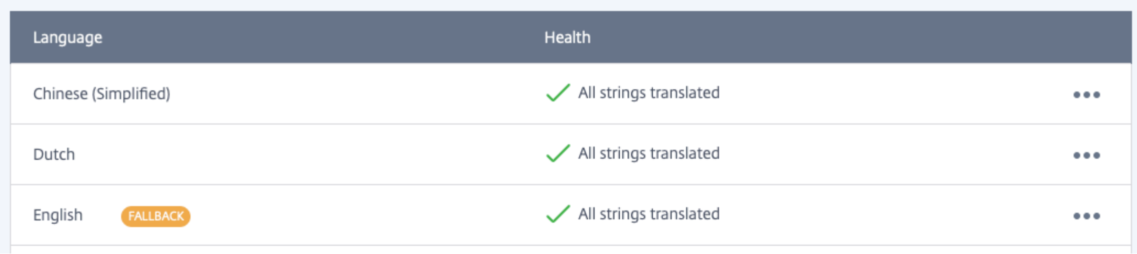
Export files for translation

To export a localization file, open the microapp edit screen by selecting the relevant microapp's hamburger button.

Follow these steps:

1. Select **Localization** in the left column

   The **Localization** page opens that displays all the currently localized languages.



Language	Health	
Chinese (Simplified)	✓ All strings translated	•••
Dutch	✓ All strings translated	•••
English FALLBACK	✓ All strings translated	•••

2. Select **Export**.
3. Select the languages you want to export for localization, and whether you want to export only missing translation strings.
4. Select **Export**

   The JSON files download to your local machine.

Working with localization files

You can then open and edit the desired localization JSON file with your preferred text editor and once ready, save the file in the JSON format ready for import back to the microapp admin console.

Import localization

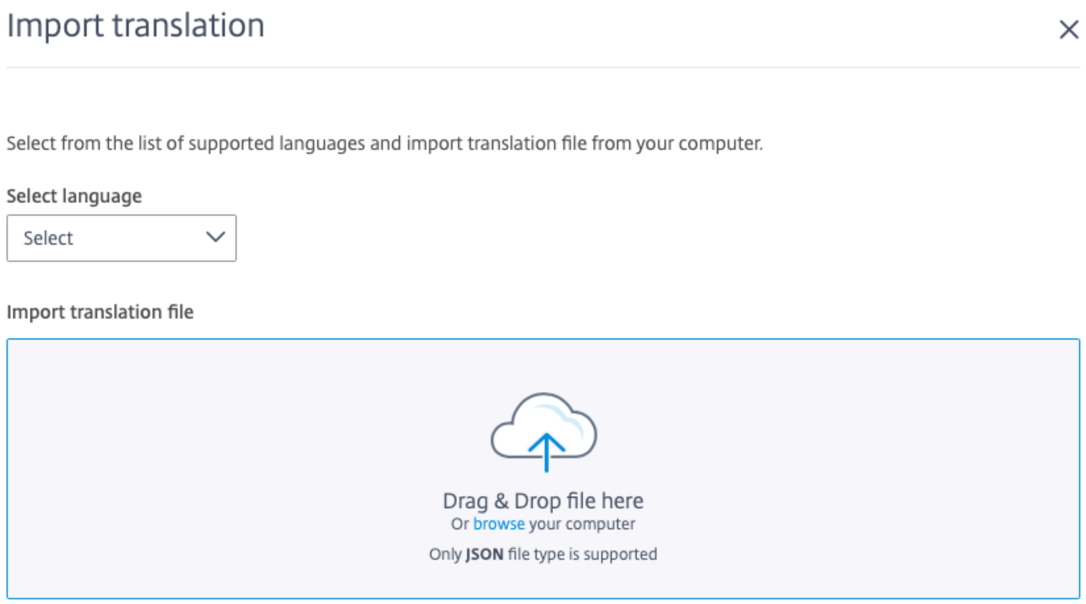
When you have prepared your localized JSON files, import them back into the microapp platform.

Follow these steps:

1. Select **import**.

   The import translation file blade opens.

2. Select your required localization language from the available languages.
3. Drag the translated JSON file:



Import translation

Select from the list of supported languages and import translation file from your computer.

Select language

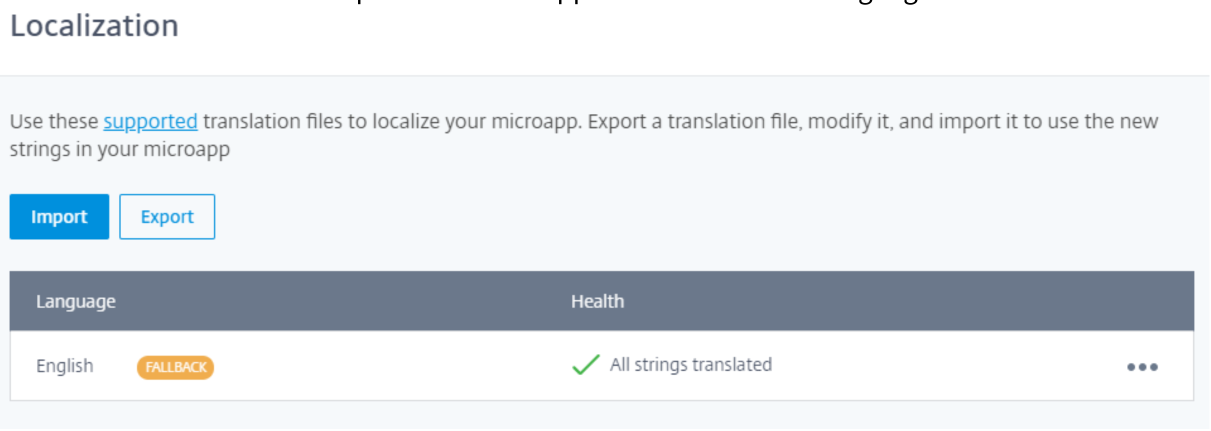
Select

Import translation file

Drag & Drop file here
Or browse your computer
Only **JSON** file type is supported

4. Select import.

Your translation file is now imported and the app is available in that language for subscribers:



Localization

Use these supported translation files to localize your microapp. Export a translation file, modify it, and import it to use the new strings in your microapp

Language	Health
English FALLBACK	✓ All strings translated •••

Configure User providers

October 13, 2021

Configure user providers increases administration efficiency by removing the need to replicate and synchronize user groups created and maintained in your System of Record’s (SoR) identity providers configuration settings.

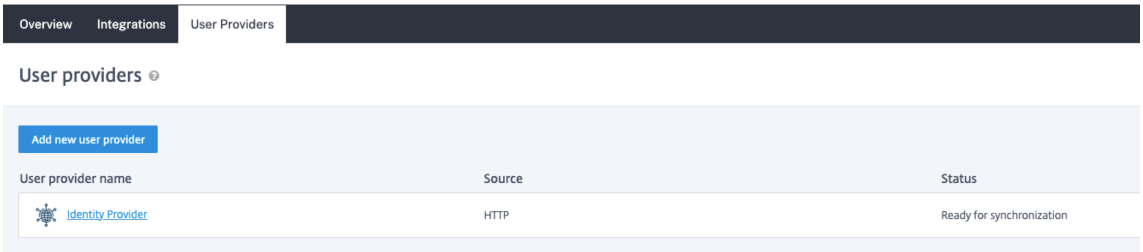
Microapps admins can configure user providers to collect user and user group data from your SoR and use this data to manage microapp subscriptions in all integrations. To configure user providers, your application SoR must provide APIs that expose users/user groups, for example ServiceNow, Salesforce, Jira, and so on.

Create user provider

You can either create a new user provider manually or import a user provider that is already configured.

Follow these steps:

1. From the **Microapp Integrations** page, select the **User Providers** tab in the top bar.



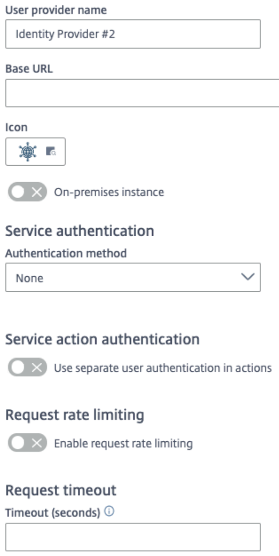
Overview | Integrations | User Providers

User providers

Add new user provider

User provider name	Source	Status
Identity Provider	HTTP	Ready for synchronization

2. Choose a user provider type. Select **Create a new user provider from your HTTP web service** to configure this user provider manually. You can also **Import a previously configured user provider**. For more information on importing, see Import.
3. Enter a **User provider name** for the connection.
4. Enter the user provider **Base URL**.
5. Select an **Icon** for the user provider from the Icon Library, or leave this as the default icon.
6. Select the **Service authentication** method, and complete all required details based on the authentication method that you use. For more information on authentication methods, see Set up Service Authentication.
7. (Optional) Enable the **Enable request rate limiting** toggle if needed and enter a **Number of requests** and a **Time interval**.
8. (Optional) Enter a value for **Request timeout** if needed.
9. Select **Add** to finish creating the user provider.



User provider name

Identity Provider #2

Base URL

Icon

On-premises instance

Service authentication

Authentication method

None

Service action authentication

Use separate user authentication in actions

Request rate limiting

Enable request rate limiting

Request timeout

Timeout (seconds)

Now you import a script to finish.

Import script

Using the **User provider script** capability in Microapps, you need to upload a script to finish the user provider setup. We've provided script requirements and a JavaScript model below. See Prepare your script. For general information about scripting, see HTTP integration scripting.

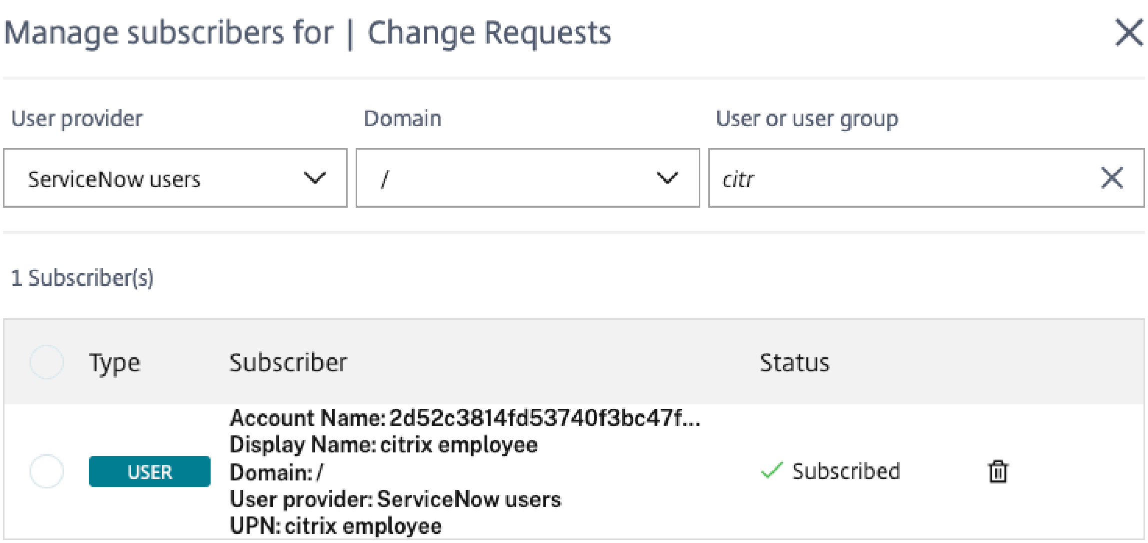
After you added the user provider (in the previous procedure), the page opens with a view of your user providers. Follow these steps to add the script.

1. Select **Add script** under **Status**.
2. Select **Import script**.
3. Drag your script onto the import pop-up, or browse for the file. The script must be a .js file no larger than 1 MB.

   The script is parsed and validated.
4. Select **Import** to finish.

   You can see scripting details under **Script handlers** in the **User provider script** view of the user provider. To see requests made by the script, select the menu next to the user provider entry and select **Sync log**.

Your new user provider is now available when assigning subscribers. For more information, see Manage subscribers.



Manage subscribers for | Change Requests

User provider: ServiceNow users

Domain: /

User or user group: citr

1 Subscriber(s)

	Type	Subscriber	Status	
○	USER	Account Name: 2d52c3814fd53740f3bc47f... Display Name: citrix employee Domain: / User provider: ServiceNow users UPN: citrix employee	✓ Subscribed	🗑

Prepare your script

When preparing your script, consider the following requirements:

- The import script must start by loading the built-in library microapp-user-groups. This library defines the objects that must be stored in the database: `const { User, Group, UserGroupMapping } = library.load("microapp-user-groups");`
- Objects have the following structure/properties:
  - `User(accountName, displayName, email, domain, userPrincipalName, userId)` Email addresses must be unique within user provider
  - `Group(accountName, displayName, domain, userPrincipalName, groupId, parentGroupId)` Group hierarchy is also defined using parentGroupId
  - `UserGroupMapping(userId, groupId)` Maps users to groups
- All properties are of data type `STRING`.
- User.email has to match the email of a user logged in to Citrix Workspace.

Model script

Use the following JavaScript code as a model.

> **Note**
>
> The following model is written specifically for the ServiceNow SoR. This script is not compatible with other services.

```
   const {
 User, Group, UserGroupMapping  }
 = library.load("microapp-user-groups");

   function fullSync(params) {

     fullSyncUsers(params);
     fullSyncGroups(params);
     fullSyncUserGroupMapping(params);
    }


   function fullSyncUsers({
 client, dataStore  }
) {

     let offset = 0;
     do {

       const response = client.fetchSync(
         `/api/now/table/sys_user?sysparm_fields=sys_domain_path%2
            Cname%2C%20sys_id%2Cuser_name%2Cemail&sysparm_query=
            emailISNOTEMPTY^active%3Dtrue&sysparm_limit=100&
            sysparm_offset=${
offset }
`
       );
       if (!response.ok) {

         console.log("Error status:", response.status, response.
            statusText);
         console.log("Error body:", response.textSync());
         throw new Error("Network response was not ok");
        }

       console.log("fetch done");

       const users = response.jsonSync().result;
       console.log("users");

       users.map((user) =>
         console.log(
           user.user_name,
           user.name,
      user.email,
      user.sys_domain_path,
      user.name,
      user.sys_id
     )
    );
    dataStore.save(
     User.tableModel,
     users.map(
      (user) =>
       new User(
        user.user_name,
        user.name,
        user.email,
        user.sys_domain_path,
        user.user_name,
        user.sys_id
       )
     )
    );

    offset = offset + 100;
    console.log(`offset: ${
offset }
`);
   }
 while (offset < 300);
  }


  function fullSyncGroups({
 client, dataStore  }
) {

   let offset = 0;
   do {

    const response = client.fetchSync(
     `/api/now/table/sys_user_group?sysparm_query=active%3Dtrue&
      sysparm_limit=100&sysparm_offset=${
offset }
`
    );
    if (!response.ok) {

         console.log("Error status:", response.status, response.
            statusText);
         console.log("Error body:", response.textSync());
         throw new Error("Network response was not ok");
        }


       const groups = response.jsonSync().result;
       groups.map((group) =>
         console.log(
           group.name,
           group.name,
           "/",
           group.name,
           group.sys_id,
           group.parent.value
         )
       );
       dataStore.save(
         Group.tableModel,
         groups.map(
           (group) =>
             new Group(
               group.name,
               group.name,
               "/",
               group.name,
               group.sys_id,
               group.parent.value
             )
         )
       );
       offset = offset + 100;
       console.log(`offset: ${
offset }
`);
      }
 while (offset < 400);
    }


   function fullSyncUserGroupMapping({
 client, dataStore  }
) {

     let offset = 0;
     do {

       const response = client.fetchSync(
         `/api/now/table/sys_user_grmember?&sysparm_limit=100&sysparm_offset=${
offset }
`
       );
       if (!response.ok) {

         console.log("Error status:", response.status, response.statusText);
         console.log("Error body:", response.textSync());
         throw new Error("Network response was not ok");
        }


       const mappings = response.jsonSync().result;
       mappings.map((mapping) =>
         console.log(mapping.user.value, mapping.group.value)
       );
       dataStore.save(
         UserGroupMapping.tableModel,
         mappings.map(
           (mapping) =>
             new UserGroupMapping(mapping.user.value, mapping.group.value)
         )
       );
       offset = offset + 100;
       console.log(`offset: ${
offset }
`);
      }
 while (offset < 400);
    }


   integration.define({

     synchronizations: [
       {

         name: "snowUserGroups", // Logical name
```

```
          fullSyncFunction: fullSync,
         }
 ,
      ],
      model: {

        tables: [User.tableModel, Group.tableModel, UserGroupMapping.
           tableModel],
       }
 ,
     }
 );
<!--NeedCopy-->
```

Synchronize data

April 26, 2021

As an administrator, you have detailed control on the synchronization schedules that you set. However, you must pay attention to how you set the synchronization times to avoid jobs missing their schedule. As a measure to help prevent this from occurring, we have randomized timetables for the same time periods.

There are three types of jobs that are relevant:

- **Full synchronization** - Optimized for huge data volumes which may take a lot of time to complete.
- **Incremental synchronization** - Optimized for small but frequent updates.
- **Notification jobs** - Evaluation of notification events and sending notification messages to the target audience. Notification jobs run after each full synchronization, incremental synchronization, after service actions, and also independently.

Synchronization rules

To get started, for any given integration one full synchronization must finish successfully before any incremental synchronization can run.

Only one type of job can run at any time for any given integration. For example, while a full synchronization is running, there cannot be an incremental synchronization running nor any notification job running. It is the same situation for incremental synchronization and notification jobs.

However, several notification jobs can run at the same time. The maximum number of jobs for all integrations combined is three per instance of Microapps service.

It might happen that the schedules for full synchronization and incremental synchronization overlap. It is not possible to predict which succeeds and which fails. There are no rules governing this situation. In this case, we rely on randomization and the limited throughput of three jobs per instance which decreases the odds that a full synchronization and incremental synchronization start at the same time and collide.

Synchronization that does not meet its schedule

If a job doesn't run on schedule, it is marked as misfired and the system attempts to schedule the misfired job as soon as possible. Reasons you might miss the schedule:

- There are already three other jobs currently running on this instance.
- A job takes longer to complete than what is set in the repeat interval. For example, you set incremental synchronization for every 15 minutes, but the job takes 20 minutes to complete for some reason.

> **Note**
>
> If any value in the primary key column is missing or has an invalid type, the record is skipped during synchronization and a log warning is generated.

Veto rules in detail

Every time a job starts, the veto rules that can cancel the job are checked. Veto rules are different for different types of jobs.

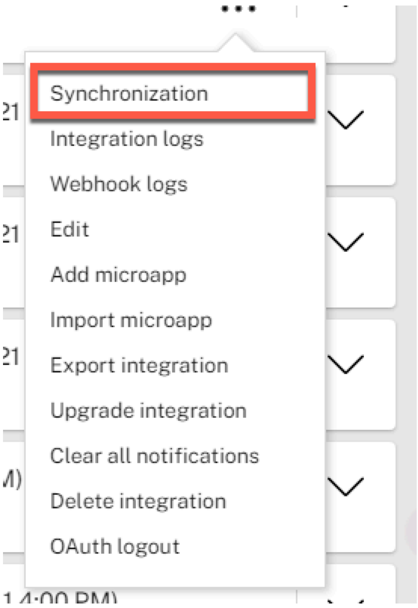
- For full synchronization, a job is vetoed if another notification/synchronization job is already running for the same data integration. In this case, the job is retriggered in 5 seconds.
- For incremental synchronization, if there's never been a successful full synchronization for the integration the scheduler starts a full sync instead as a one-time job. Also, as with the above, a job is vetoed if another notification/sync job is already running for the same data integration and the job is retriggered in 5 seconds.
- For notification events, a job is vetoed if there's never been a successful full synchronization for the integration. Notification jobs run concurrently. This means that several notification jobs can run at the same time. However there is only one changelog table for each primary table for optimization reasons. Therefore, there can only be one notification job updating the changelog table at a time. As a result, while one notification job updates the changelog table, other notification jobs wait. When this is complete, the other notification jobs can run.

Set data synchronization

Pull data from your integrated applications to the Microapps platform so that a comparison can be made to the cache. As a best practice, full synchronization is performed every 24 hours and incremental syncs can be configured to pull every five minutes.

Scheduled synchronization jobs run at the interval defined after the last successful run. For example, if the interval is set to **5 minutes** the job starts at 10.05, runs (for example for 15 minutes) and once successful pauses for an interval of five minutes and starts again. Therefore the job starts at 10.05, runs successfully until 10.20, and then starts again at 10.25.

1. From the Manage Microapps page, select the menu next to the integration for which you want to set synchronization.
2. Select **Synchronization**.



3. Set **Full** and **Incremental** data synchronization values.

   - **Full** Drops the local cache and pulls all data from the source system.

     > **Important:**
     >
     > Running full synchronization can take a long time. We recommend running full synchronization at night or generally during off hours. You can cancel a data synchronization that is in progress at any time by selecting the *X* icon.

   - **Incremental** Pulls only changed (new and updated) records. Does not load deleted data.

     > **Important:**
     >
     > Not all APIs support incremental synchronization.
     >
     > When you define **daily** or **weekly** synchronization, synchronization occurs randomly within the timeslot that you select. For example, selecting 00-04 daily full synchronize will run a full synchronize at a randomly selected time in that period.

4. Select **Save**.

> **Note:**
>
> You can also select the arrow icons to run the integrations on demand if necessary.

Customization scenarios

December 12, 2019

After you set up your integration, customize your microapps. The following table shows four key use-case scenarios and the needed activities. You can review an overview of the following use-case scenarios, or follow the link to the appropriate scenario.

Create a microapp for a template integration	Customize an existing microapp for a template integration	Create a microapp for a custom integration that you built using the HTTP Connector	Import a microapp
Verify needed table entities and add new entities if necessary	Verify needed table entities and add new entities if necessary	Add entities, as needed	Verify needed table entities and add new entities if necessary
Add a blank microapp	Already exists	Add a blank microapp	Import a microapp
Create pages or notifications or both	Create pages or notifications or both, or open existing pages or notifications	Create pages or notifications or both	Create pages or notifications or both, or open existing pages or notifications
Customize the pages and notifications using the builder, and customize notification cards	Customize the pages and notifications using the builder, and customize notification cards	Customize the pages and notifications using the builder, and customize notification cards	Customize the pages and notifications using the builder, and customize notification cards
Manage access and subscriptions	Manage access and subscriptions	Manage access and subscriptions	Manage access and subscriptions
Create a microapp for a template integration	Customize an existing microapp for a template integration	Create a microapp for a custom integration that you built using the HTTP Connector	Import a microapp

Create a microapp for a template integration

Add a microapp to suit your business needs. The template integration comes with a robust database connection. Ensure the required table entities are already available, then use the Microapps builder to create a microapp from scratch. Add new pages and notifications and then populate them.

Follow these steps:

- Verify needed table entities and add new entities if necessary
- Add the microapp
- Add pages or notifications or both
- Customize the pages and notifications using the builder
- Manage access
- Manage subscriptions

For full scenario details, see Create a microapp for a template integration.

Customize an existing microapp for a template integration

As with creating a microapp, you can add new pages and notifications. In this case, you can also edit existing notifications and pages using the Microapps builder.

Follow these steps:

- Verify needed table entities and add new entities if necessary
- Add pages or notifications or both, or open existing pages or notifications
- Customize the pages and notifications using the builder
- Manage access
- Manage subscriptions

For full scenario details, see Customize an existing microapp for a template integration.

Create a microapp for a custom integration that you built using the HTTP Connector

In this case, you have to manually add database connections for custom integrations, then add an app and use the builder to create a microapp from scratch.

Follow these steps:

- Add entities, as needed
- Add the microapp
- Add pages or notifications or both
- Customize the pages and notifications using the builder
- Manage access
- Manage subscriptions

For full scenario details, see Create a microapp for a custom integration that you built using the HTTP Connector.

Import a microapp

Import a microapp that you created in another instance. Then edit an existing microapp or add pages and notifications.

Follow these steps:

- Upload the .mapp file containing the microapp to the Application Integration
- Verify needed table entities and add new entities if necessary
- Add pages or notifications or both, or open existing pages or notifications
- Customize the pages and notifications using the builder
- Manage access
- Manage subscriptions

For full scenario details, see Import a microapp.

Create a new microapp for a template integration

April 28, 2021

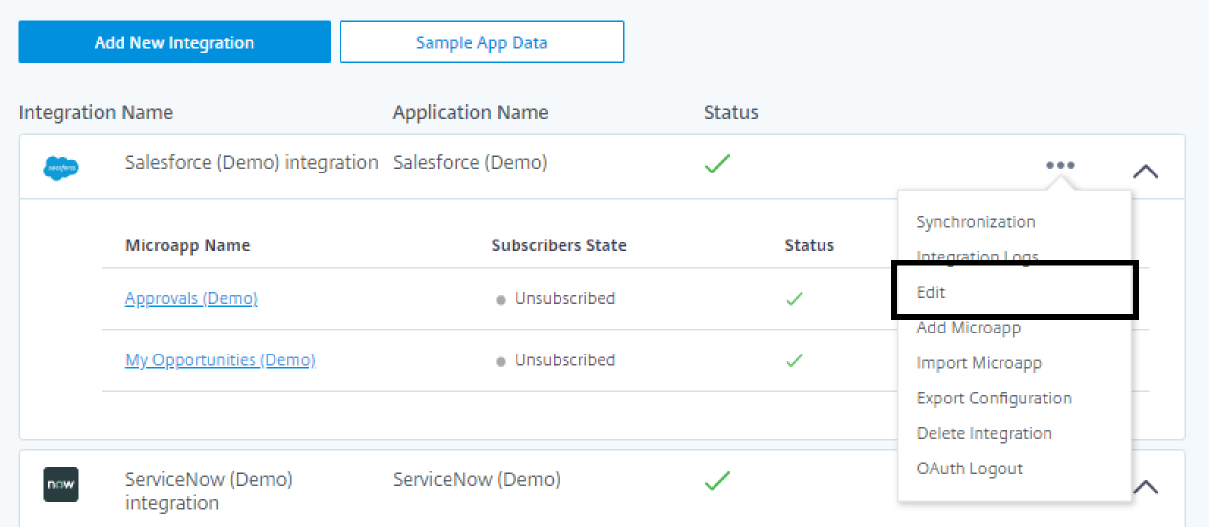
You can add a new microapp to suit your business needs. The template integration comes with a robust database connection. Ensure the required table entities are already available, then use the Microapps builder to create a microapp from scratch. Add new pages and notifications and then populate them.

Follow these steps:

1. Verify needed table entities. Add new entities if necessary.
2. Add the microapp.
3. Add pages or notifications or both.
4. Customize the pages and notifications using the builder.
5. Manage access.
6. Manage subscriptions.

Verify needed table entities and add new entities if necessary

Check an existing integration to ensure that the tables that you require exist already in Microapps. If you find a required table missing, you must add it. For more information, see Map database table entities.



Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓
ServiceNow (Demo) integration	ServiceNow (Demo)	✓

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
My Opportunities (Demo)	● Unsubscribed	✓

1. From the **Integrations** page, select the menu next to the integration to which you want to add a microapp.
2. Select **Edit**.

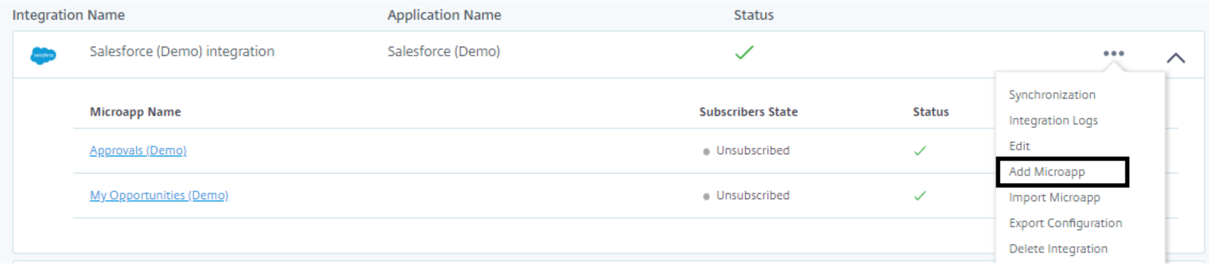
   The **Tables** page opens with an overview of how the database is divided into database tables.
3. To find your required table, select **Edit Schema**, and filter for the required entity and confirm that it exists.

You are ready to add a microapp.

Add a microapp

Select from out-of-the-box microapps or create a microapp from scratch. Once you add a new microapp, it appears under the related integration on the **Integrations** page.

Before you begin, make sure you verified or added required table entities.



Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
My Opportunities (Demo)	● Unsubscribed	✓

1. From the **Integrations** page, select the menu next to the integration to which you want to add a microapp.
2. Select **Add Microapp**.
3. Choose one of the out-of-the-box microapps or select **Blank Template** to build your own microapp based on your business needs.

After you add the blank microapp, it appears under the related integration on the **Integrations** page.

4. Return to the **Integrations** page and select **Blank Microapp** from the list under the integration.

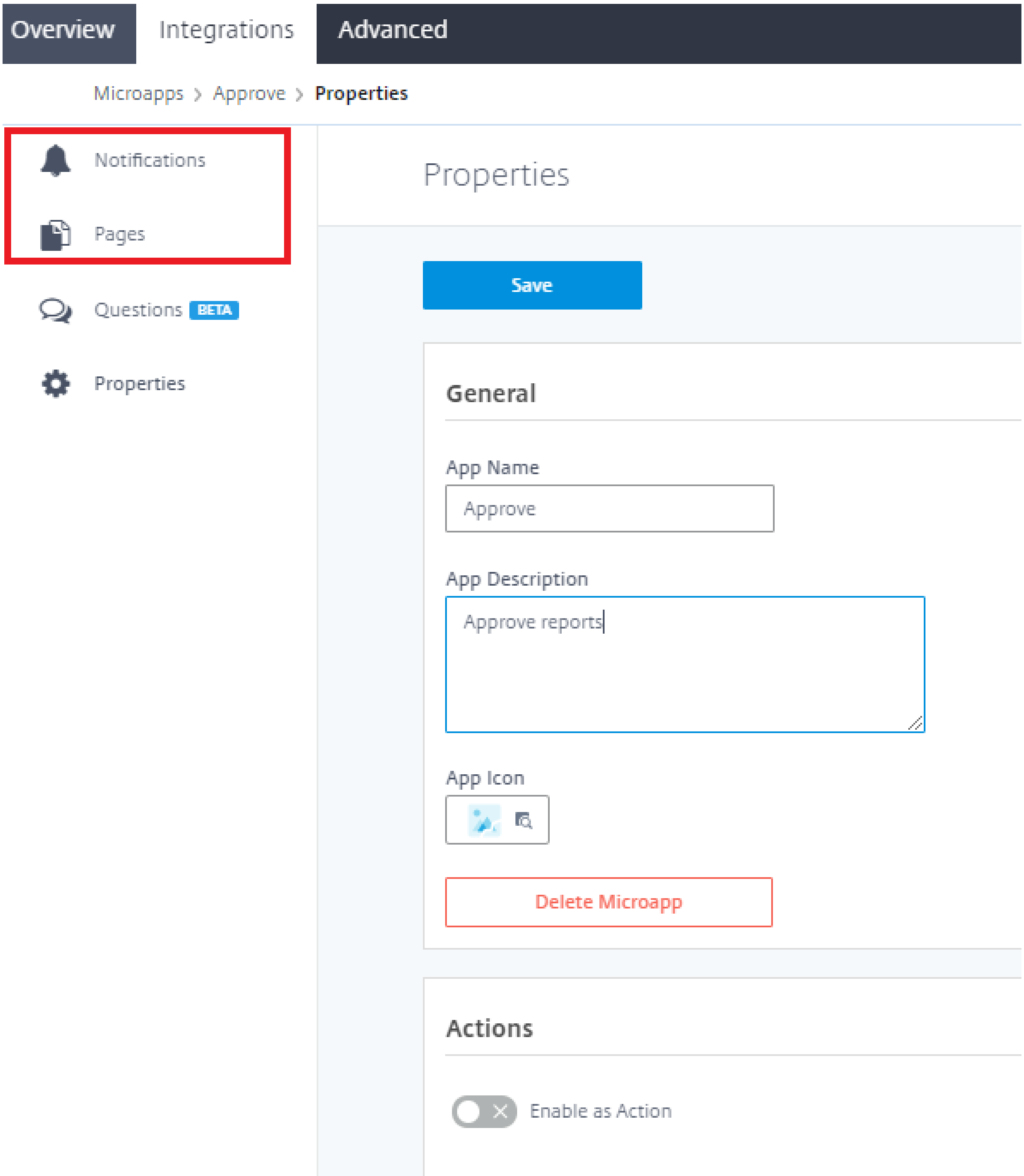
   The **Properties** page opens.

5. Give it an appropriate name and description.
6. Select **Microapp Icon** and choose an appropriate icon from menu. There are **App Icons**, **Action and Notification Icons**, and **Microapps and Data** icons from which you can select.

You are ready to add a Page or Notification.

Add new pages and notifications

After you have your microapp ready and database entities prepared, you need to create Action Pages or Notifications or both.



Properties

General

App Name: Approve

App Description: Approve reports

App Icon

Actions

Enable as Action

Add Action Pages

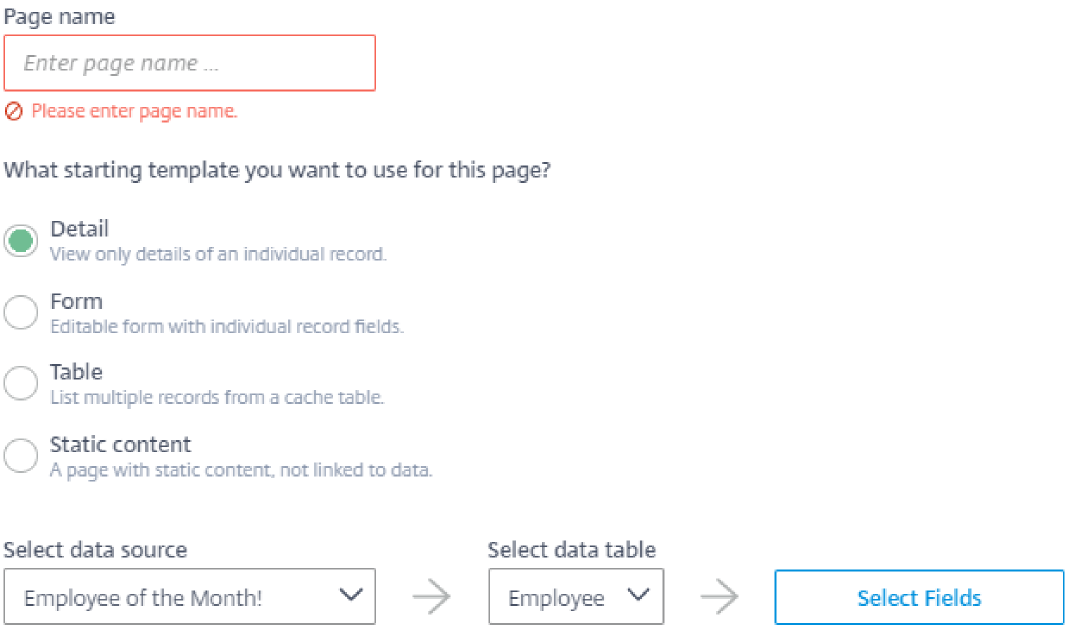
Add an action page for this microapp. Select from the starting templates, then customize the page in the builder.

Follow these steps:

1. Select the microapp to which you want to add a page.

2. Select **Pages**, and **Add New Page**.
3. Give the Page a name.
4. Select a starting template for the page:
   - Detail
   - Form
   - Table
   - Static content

   The following screenshot shows what the **New Page** screen looks like:



Page name

Enter page name ...

⊘ Please enter page name.

What starting template you want to use for this page?

● Detail
View only details of an individual record.

○ Form
Editable form with individual record fields.

○ Table
List multiple records from a cache table.

○ Static content
A page with static content, not linked to data.

Select data source: Employee of the Month! → Select data table: Employee →

5. To set the table columns that you want your page to be prepopulated with, click **Select Fields** and select related field names.
6. Select **Set Columns**, and **Add**.

The new page is added to the **Pages** list and is ready to be customized.

Add Event Notifications

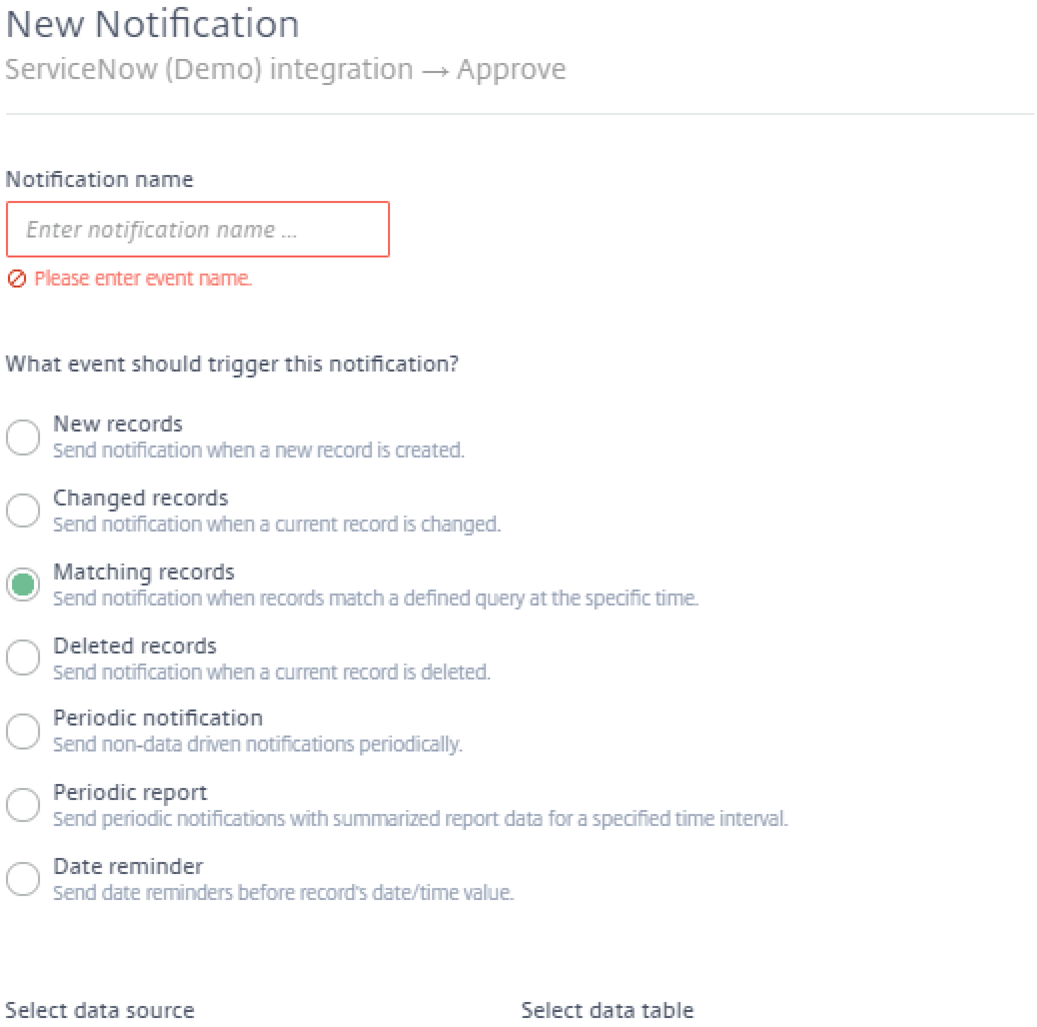
Create triggers for events to be sent to the client application, such as new PTO request or notification that a record changed. Select from the template types, then customize the event in the builder.

Follow these steps:

1. Select the microapp to which you want to add an event.
2. Select **Create New Notification** at the bottom of the page.

3. Give the Notification a name.
4. Select your desired trigger and notification type from the following:
    - New Records - Send notification when a new record is created.
    - Changed records
    - Matching record
    - Delete records
    - Periodic report
    - Periodic notification (user action)
    - Date reminder

    The following screenshot shows what the **New Notification** screen looks like:



New Notification

ServiceNow (Demo) integration → Approve

Notification name

Enter notification name ...

Please enter event name.

What event should trigger this notification?

○ New records
Send notification when a new record is created.

○ Changed records
Send notification when a current record is changed.

● Matching records
Send notification when records match a defined query at the specific time.

○ Deleted records
Send notification when a current record is deleted.

○ Periodic notification
Send non-data driven notifications periodically.

○ Periodic report
Send periodic notifications with summarized report data for a specified time interval.

○ Date reminder
Send date reminders before record's date/time value.

Select data source

Select data table

5. Verify the data source and select your data table.
6. Select Add.

The new notification is added to the **Notifications** list and is ready to be configured.

Customize the pages and notifications using the builder

For more information about customizing pages and notifications, see Page builder components and Build event notifications.

Manage subscriptions

Manage microapp subscribers to enable the microapps for specific users and user groups within your organization. For more information, see Assign subscribers.

Customize an existing microapp for a template integration

April 28, 2021

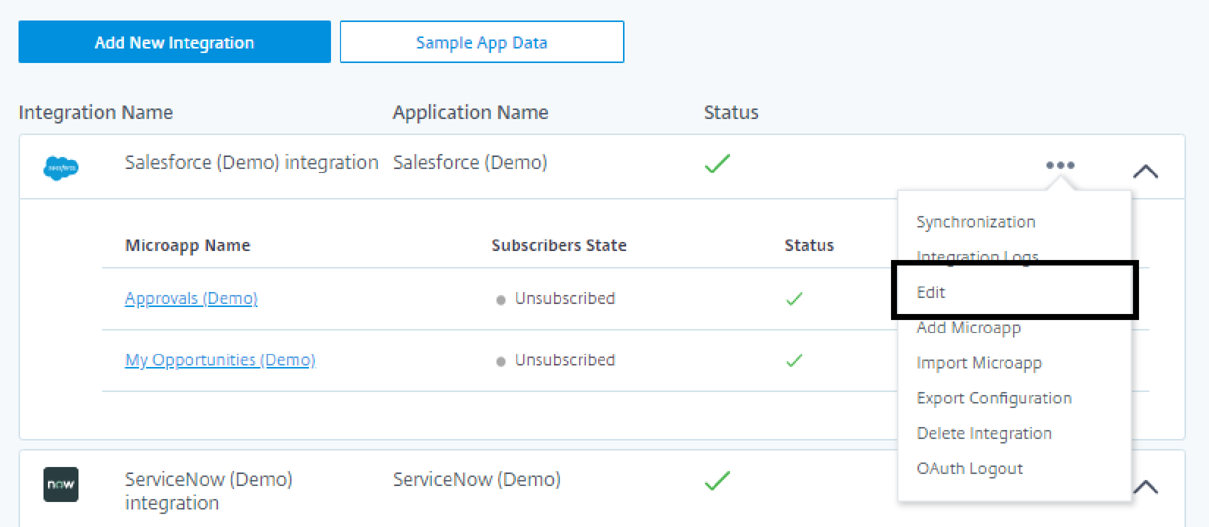
As with creating a new microapp, you can add new pages and notifications. In this case, you can also edit existing notifications and pages using the Microapps builder.

Follow these steps:

1. Verify needed table entities. Add new entities if necessary.
2. Add pages or notifications or both or open existing pages or notifications.
3. Customize the pages and notifications using the builder.
4. Manage access.
5. Manage subscriptions.

Verify needed table entities and add new entities if necessary

Check an existing integration to ensure that the tables that you require exist already in Microapps. If you find a required table missing, you must add it. For more information, see Map database table entities.



Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓
ServiceNow (Demo) integration	ServiceNow (Demo)	✓

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
My Opportunities (Demo)	● Unsubscribed	✓

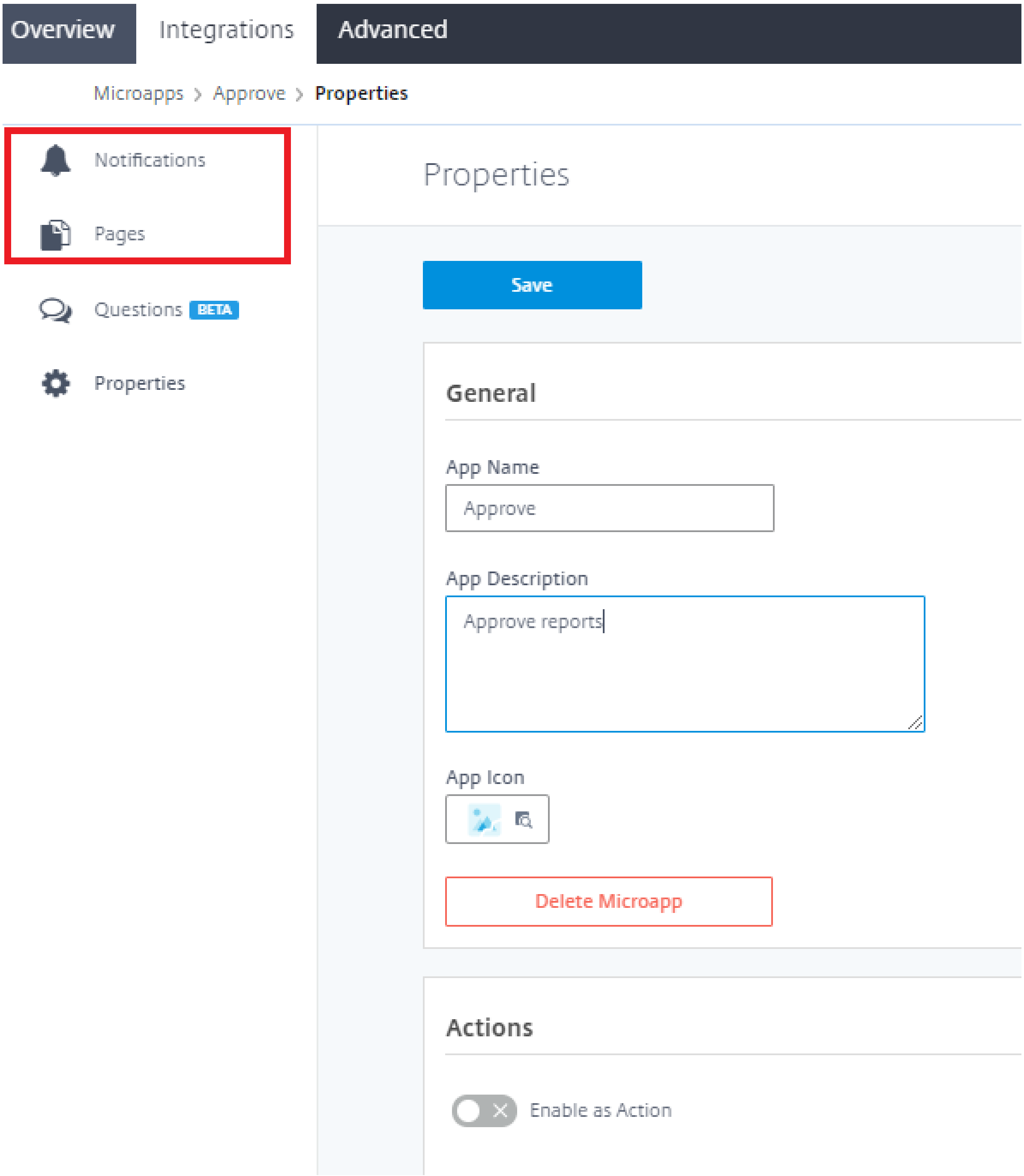
1. From the **Integrations** page, select the menu next to the integration to which you want to add a microapp.
2. Select **Edit**.

   The **Tables** page opens with an overview of how the database is divided into database tables.
3. To find your required table, select **Edit Schema**, and filter for the required entity and confirm that it exists.

You are ready to customize your microapp.

Add new pages and notifications

After you verify needed entities, you need to create Action Pages or Notifications or both.



App Name: Approve

App Description: Approve reports

○ Enable as Action

Add Action Pages

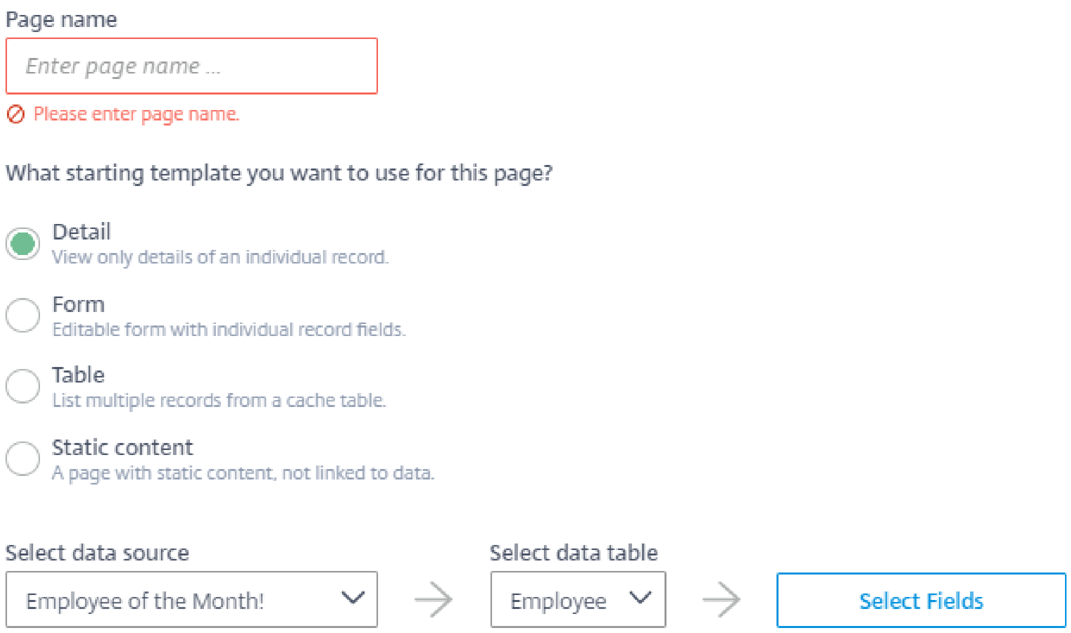
Add an action page for this microapp. Select from the starting templates, then customize the page in the builder.

Follow these steps:

1. Select the microapp to which you want to add a page.

2. Select **Pages**, and **Add New Page**.
3. Give the Page a name.
4. Select a starting template for the page:
    - Detail
    - Form
    - Table
    - Static content

    The following screenshot shows what the **New Page** screen looks like:



Page name

Enter page name ...

⊘ Please enter page name.

What starting template you want to use for this page?

● Detail
View only details of an individual record.

○ Form
Editable form with individual record fields.

○ Table
List multiple records from a cache table.

○ Static content
A page with static content, not linked to data.

Select data source: Employee of the Month!

Select data table: Employee

5. To set the table columns that you want your page to be prepopulated with, click **Select Fields** and select related field names.
6. Select **Set Columns**, and **Add**.

The new page is added to the **Pages** list and is ready to be customized.

Add Event Notifications

Create triggers for events to be sent to the client application, such as new PTO request or notification that a record changed. Select from the template types, then customize the event in the builder.

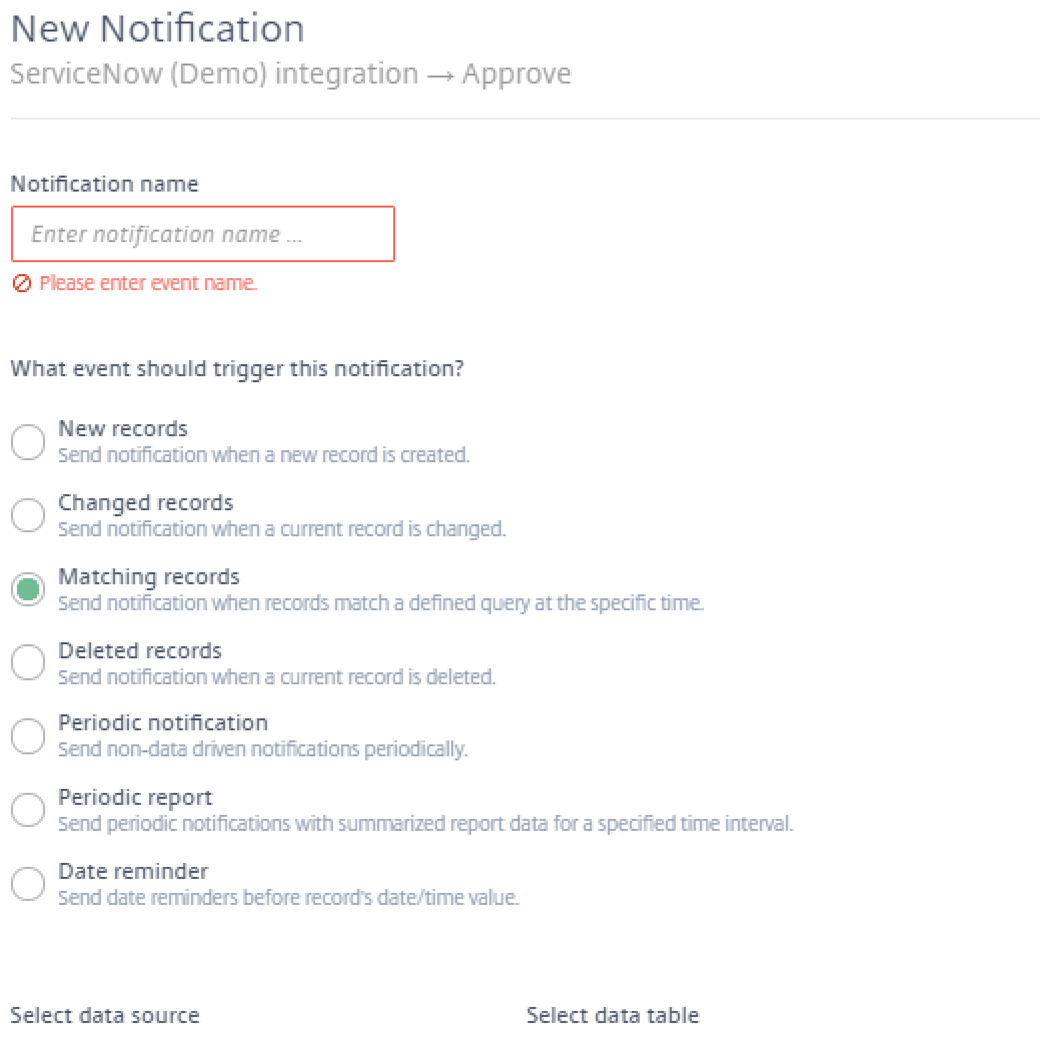
Follow these steps:

1. Select the microapp to which you want to add an event.
2. Select **Create New Notification** at the bottom of the page.

3. Give the Notification a name.
4. Select your desired trigger and notification type from the following:

   - New Records - Send notification. when a new record is created.
   - Changed records
   - Matching record
   - Delete records
   - Periodic report
   - Periodic notification (user action)
   - Date reminder

   The following screenshot shows what the **New Notification** screen looks like:



New Notification

ServiceNow (Demo) integration → Approve

Notification name

Enter notification name ...

⊘ Please enter event name.

What event should trigger this notification?

○ New records
Send notification when a new record is created.

○ Changed records
Send notification when a current record is changed.

● Matching records
Send notification when records match a defined query at the specific time.

○ Deleted records
Send notification when a current record is deleted.

○ Periodic notification
Send non-data driven notifications periodically.

○ Periodic report
Send periodic notifications with summarized report data for a specified time interval.

○ Date reminder
Send date reminders before record's date/time value.

Select data source

Select data table

5. Verify the data source and select your data table.
6. Select Add.

The new notification is added to the **Notifications** list and is ready to be configured.

Customize the pages and notifications

For more information about customizing pages and notifications, see Page builder components and Build event notifications.

Manage subscriptions

Manage microapp subscribers to enable the microapps for specific users and user groups within your organization. For more information, see Assign subscribers.

Create a new microapp for a custom integration that you built using the HTTP Connector

April 28, 2021

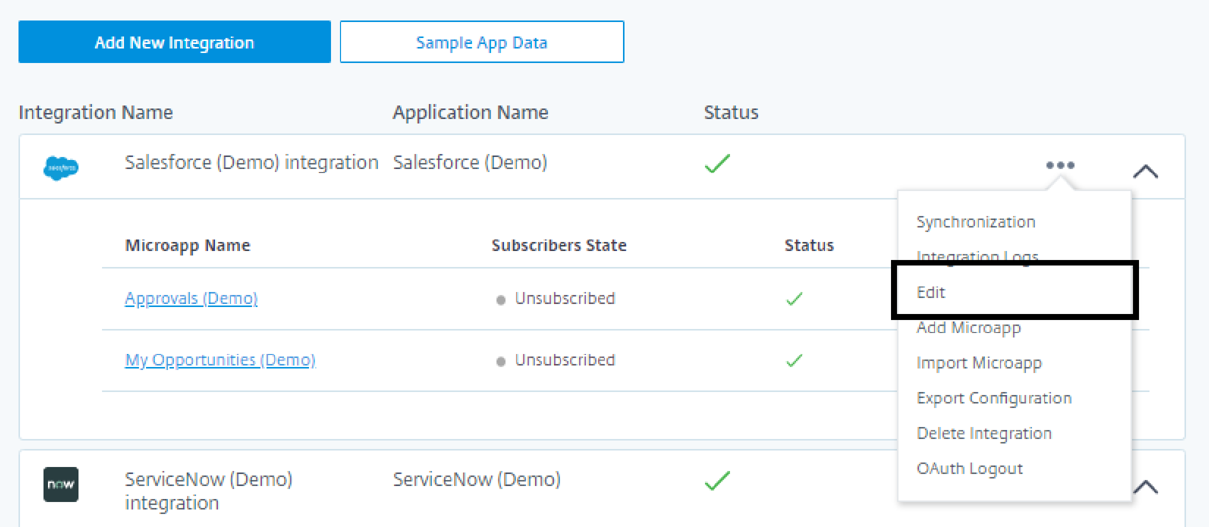
In this case, you have to manually add database connections for custom integrations, then add a microapp and use the Microapps builder to create a microapp from scratch.

Follow these steps:

1. Add entities, as needed.
2. Add the microapp.
3. Add pages or notifications or both.
4. Customize the pages and notifications using the builder.
5. Manage access.
6. Manage subscriptions.

Add entities

Since you added this integration, you need to add all required entities. For more information, see Map database table entities.



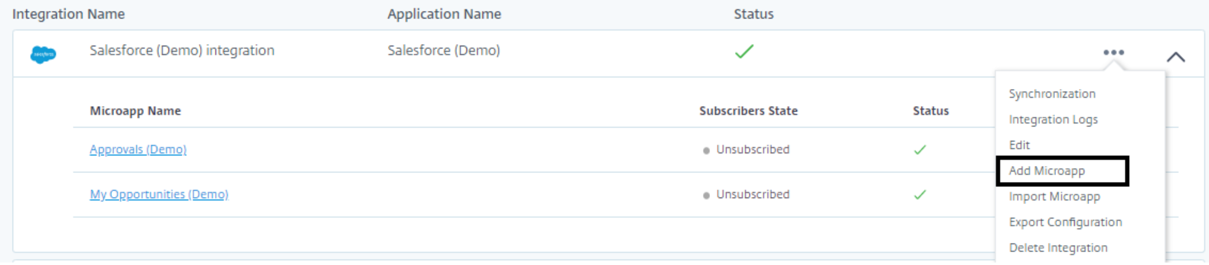
Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓
ServiceNow (Demo) integration	ServiceNow (Demo)	✓

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
My Opportunities (Demo)	● Unsubscribed	✓

Add a microapp

Select from out-of-the-box microapps or create a microapp from scratch. Once you add a new microapp, it appears under the related integration on the **Integrations** page.

Before you begin, make sure you verified or added required table entities.



Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
My Opportunities (Demo)	● Unsubscribed	✓

1. From the **Integrations** page, select the menu next to the integration to which you want to add a microapp.
2. Select **Add Microapp**.
3. Choose one of the out-of-the-box microapps or select **Blank Template** to build your own microapp based on your business needs.

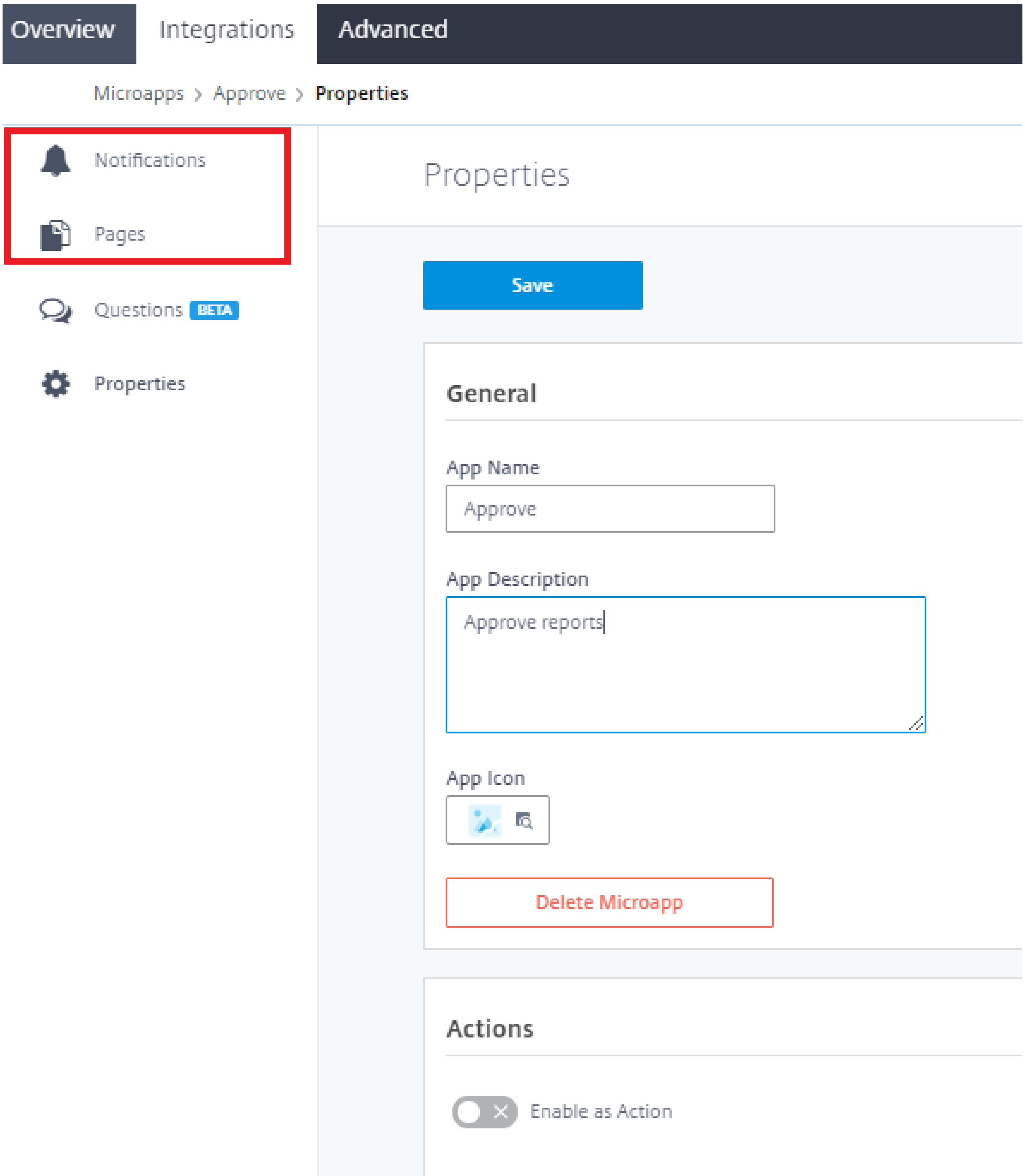
   After you add the blank microapp, it appears under the related integration on the **Integrations** page.
4. Return to the **Integrations** page and select **Blank Microapp** from the list under the integration.

   The **Properties** page opens.
5. Give it an appropriate name and description.
6. Select **Microapp Icon** and choose an appropriate icon from menu. There are **App Icons**, **Action and Notification Icons**, and **Microapps and Data** icons from which you can select.

You are ready to add a Page or Notification.

Add new pages and notifications

After you have your microapp ready and database entities prepared, you need to create Action Pages or Notifications or both.



App Name: Approve

App Description: Approve reports

Add Action Pages

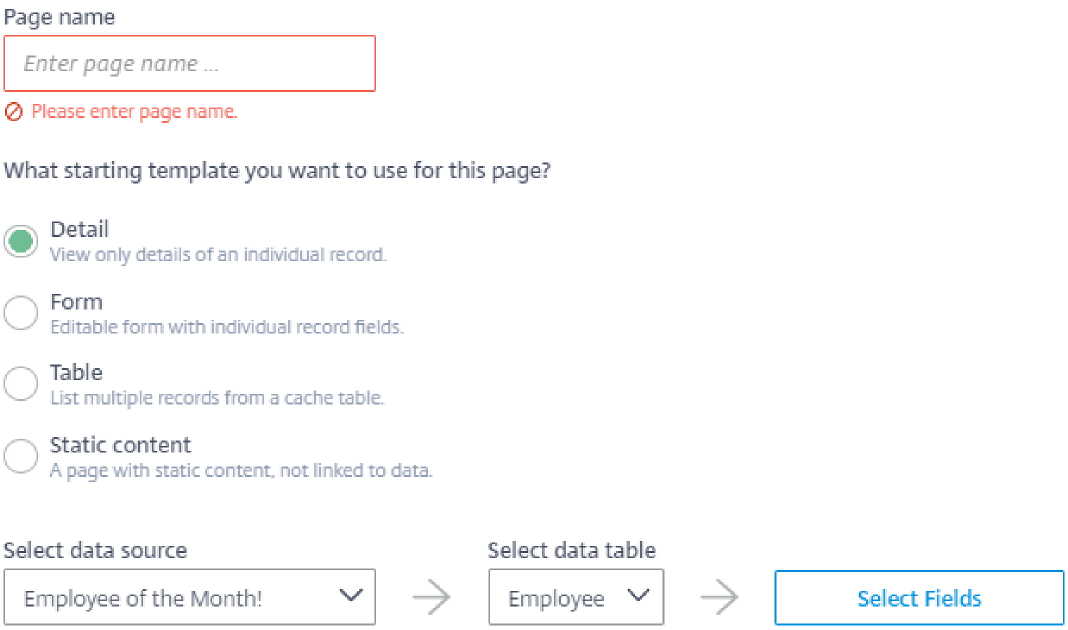
Add an action page for this microapp. Select from the starting templates, then customize the page in the builder.

Follow these steps:

1. Select the microapp to which you want to add a page.
2. Select **Pages**, and **Add New Page**.
3. Give the Page a name.
4. Select a starting template for the page:

    - Detail
    - Form
    - Table
    - Static content

    The following screenshot shows what the **New Page** screen looks like:

    Page name

    Enter page name ...

    Please enter page name.

    What starting template you want to use for this page?

    ● Detail
    View only details of an individual record.

    ○ Form
    Editable form with individual record fields.

    ○ Table
    List multiple records from a cache table.

    ○ Static content
    A page with static content, not linked to data.

    Select data source: Employee of the Month!

    Select data table: Employee

5. To set the table columns that you want your page to be prepopulated with, click **Select Fields** and select related field names.
6. Select **Set Columns**, and **Add**.

The new page is added to the **Pages** list and is ready to be customized.

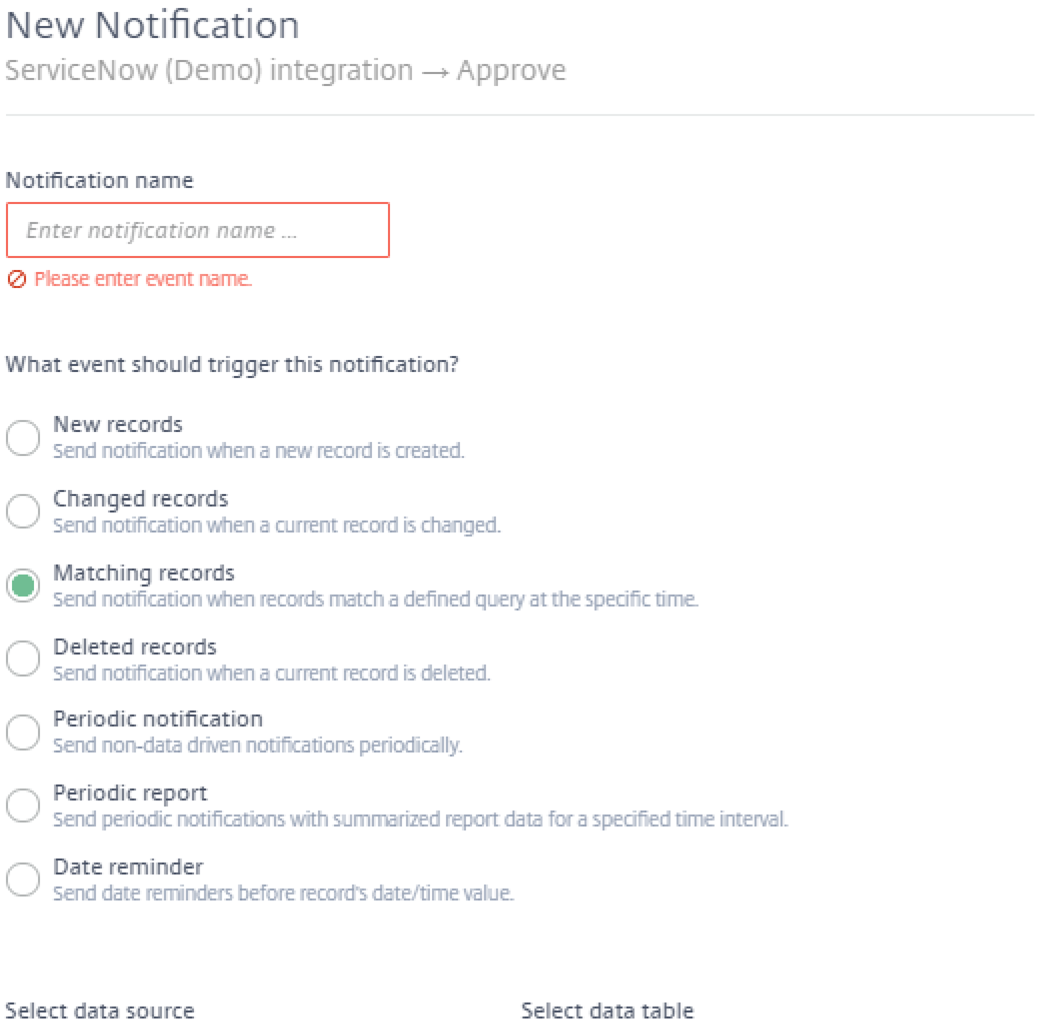
Add Event Notifications

Create triggers for events to be sent to the client application, such as new PTO request or notification that a record changed. Select from the template types, then customize the event in the builder.

Follow these steps:

1. Select the microapp to which you want to add an event.
2. Select **Create New Notification** at the bottom of the page.
3. Give the Notification a name.
4. Select your desired trigger and notification type from the following:
   - New Records - Send notification. when a new record is created.
   - Changed records
   - Matching record
   - Delete records
   - Periodic report
   - Periodic notification (user action)
   - Date reminder

   The following screenshot shows what the **New Notification** screen looks like:



New Notification

ServiceNow (Demo) integration → Approve

Notification name

Enter notification name ...

⊘ Please enter event name.

What event should trigger this notification?

○ New records
Send notification when a new record is created.

○ Changed records
Send notification when a current record is changed.

● Matching records
Send notification when records match a defined query at the specific time.

○ Deleted records
Send notification when a current record is deleted.

○ Periodic notification
Send non-data driven notifications periodically.

○ Periodic report
Send periodic notifications with summarized report data for a specified time interval.

○ Date reminder
Send date reminders before record's date/time value.

Select data source

Select data table

5. Verify the data source and select your data table.
6. Select Add.

The new notification is added to the **Notifications** list and is ready to be configured.

Customize the pages and notifications using the builder

For more information about customizing pages and notifications, see Page builder components and Build event notifications.

Manage subscriptions

Manage microapp subscribers to enable the microapps for specific users and user groups within your organization. For more information, see Assign subscribers.

Import a microapp

April 28, 2021

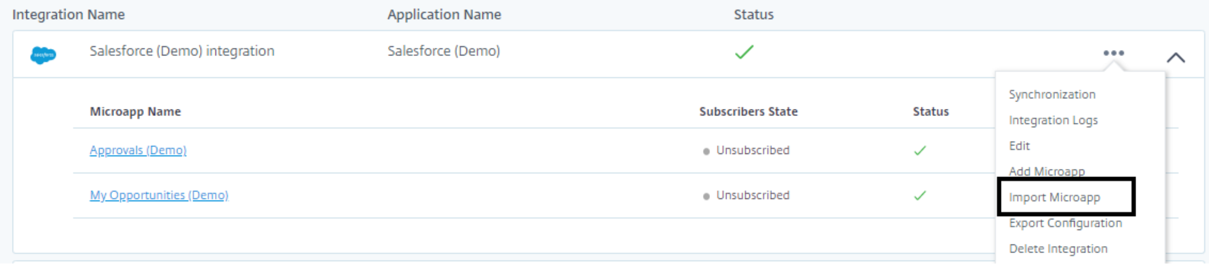
Import a microapp that you created in another instance. You can then edit or add pages or notifications or both.

Follow these steps:

1. Upload the .mapp file containing the microapp to the Application Integration.
2. Verify needed table entities. Add new entities if necessary.
3. Add pages or notifications or both or open existing pages or notifications.
4. Customize the pages and notifications using the builder.
5. Manage access.

Upload .mapp file containing the microapp to the Application Integration

After you upload the new microapp, it appears under the related integration on the **Integrations** page.



Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
My Opportunities (Demo)	● Unsubscribed	✓

1. From the **Integrations** page, select the menu next to the integration to which you want to add a microapp.
2. Select **Import**, choose a microapp file that you have available in .mapp format, and drag it to the import panel.

After you add the imported microapp, it appears under the related integration on the **Integrations** page.

1. To modify details for the microapp, return to the **Integrations** page and select the newly imported microapp from the list under the integration.

   The **Properties** page opens.

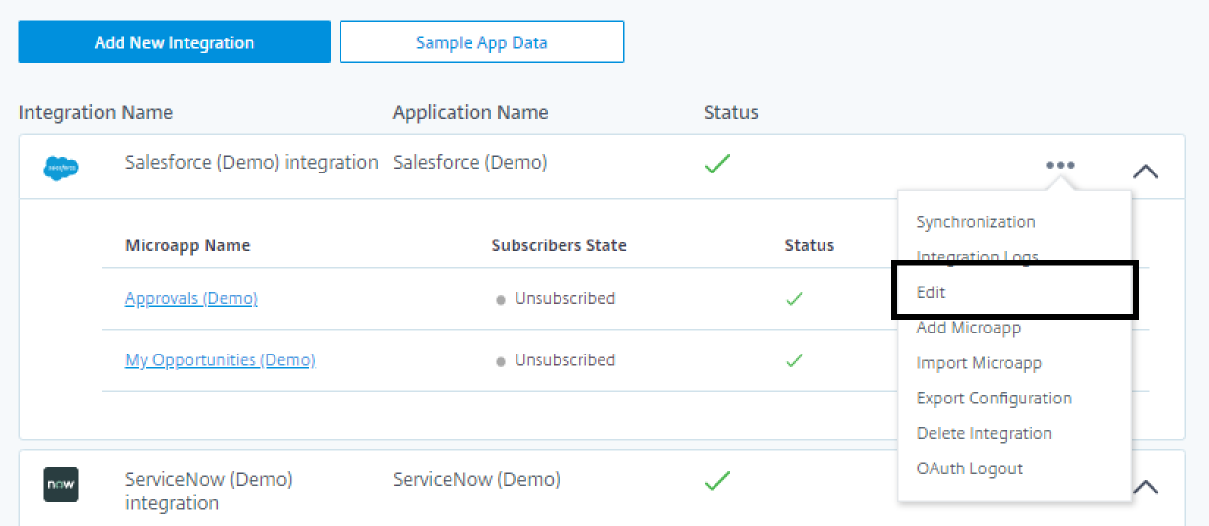
2. Change the microapp's name and description if needed.

3. If needed, select **Microapp Icon** and choose an appropriate icon from menu. There are **App Icons**, **Action and Notification Icons**, and **Microapps and Data** icons from which you can select.

You are ready to verify and add required table entities.

Verify needed table entities and add new entities if necessary

Check an existing integration to ensure that the tables that you require exist already in Microapps. If you find a required table missing, you must add it. For more information, see Map database table entities.



Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓
ServiceNow (Demo) integration	ServiceNow (Demo)	✓

Microapp Name	Subscribers State	Status
Approvals (Demo)	● Unsubscribed	✓
My Opportunities (Demo)	● Unsubscribed	✓

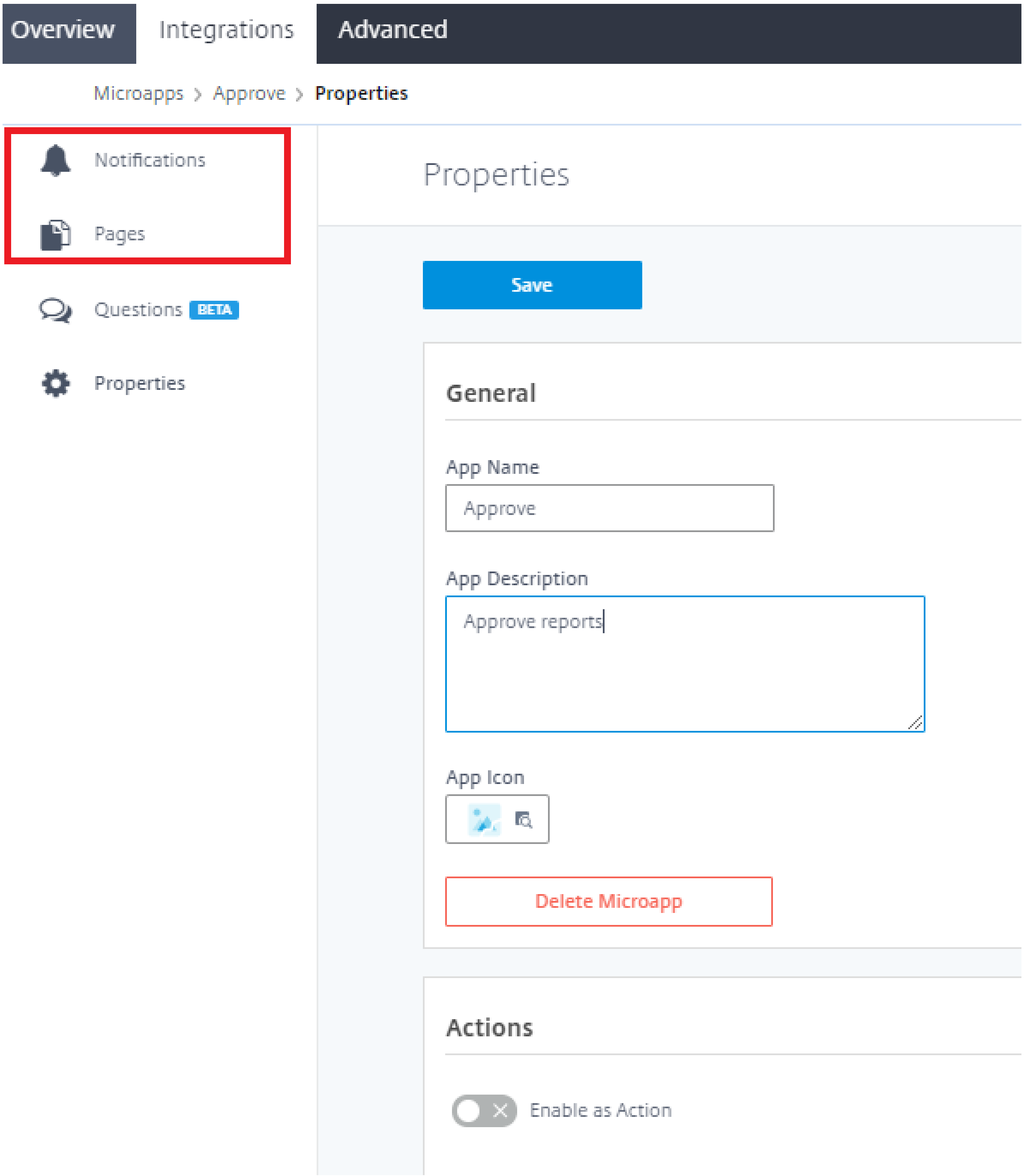
1. From the **Integrations** page, select the menu next to the integration to which you want to add a microapp.
2. Select **Edit**.

   The **Tables** page opens with an overview of how the database is divided into database tables.
3. To find your required table, select **Edit Schema**, and filter for the required entity and confirm that it exists.

You are ready to customize your microapp.

Add new pages and notifications

After you verify needed entities, you need to create Action Pages or Notifications or both.



App Name: Approve

App Description: Approve reports

Add Action Pages

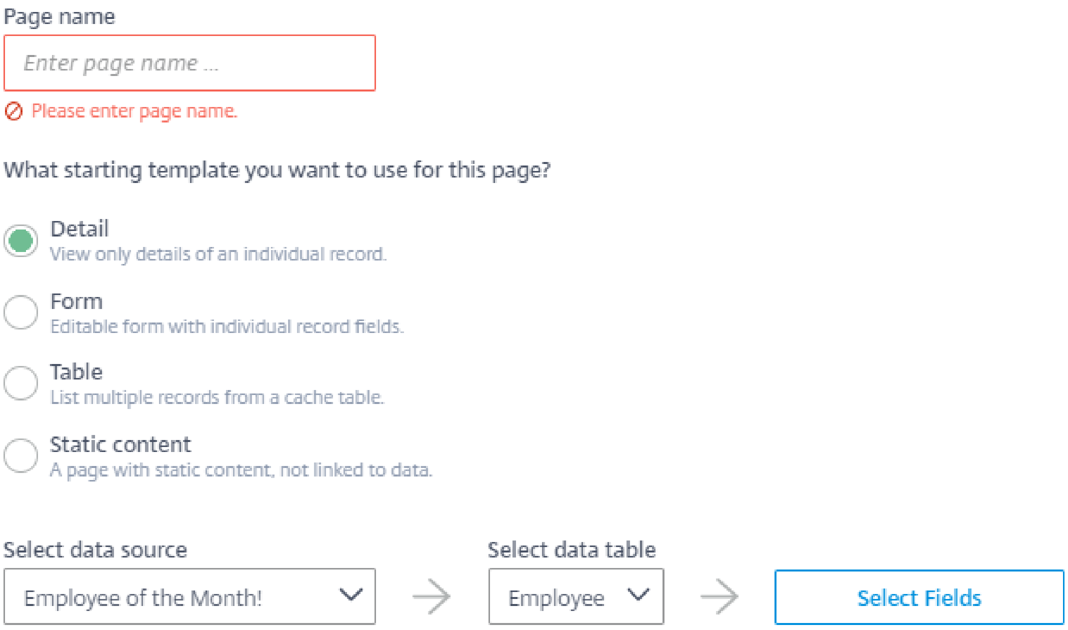
Add an action page for this microapp. Select from the starting templates, then customize the page in the builder.

Follow these steps:

1. Select the microapp to which you want to add a page.

2. Select **Pages**, and **Add New Page**.
3. Give the Page a name.
4. Select a starting template for the page:
   - Detail
   - Form
   - Table
   - Static content

   The following screenshot shows what the **New Page** screen looks like:

   Page name

   Enter page name ...

   ⊘ Please enter page name.

   What starting template you want to use for this page?

   ● Detail
   View only details of an individual record.

   ○ Form
   Editable form with individual record fields.

   ○ Table
   List multiple records from a cache table.

   ○ Static content
   A page with static content, not linked to data.

   Select data source: Employee of the Month! → Select data table: Employee → Select Fields

5. To set the table columns that you want your page to be prepopulated with, click **Select Fields** and select related field names.
6. Select **Set Columns**, and **Add**.

The new page is added to the **Pages** list and is ready to be customized.

Add Event Notifications

Create triggers for events to be sent to the client application, such as new PTO request or notification that a record changed. Select from the template types, then customize the event in the builder.

Follow these steps:

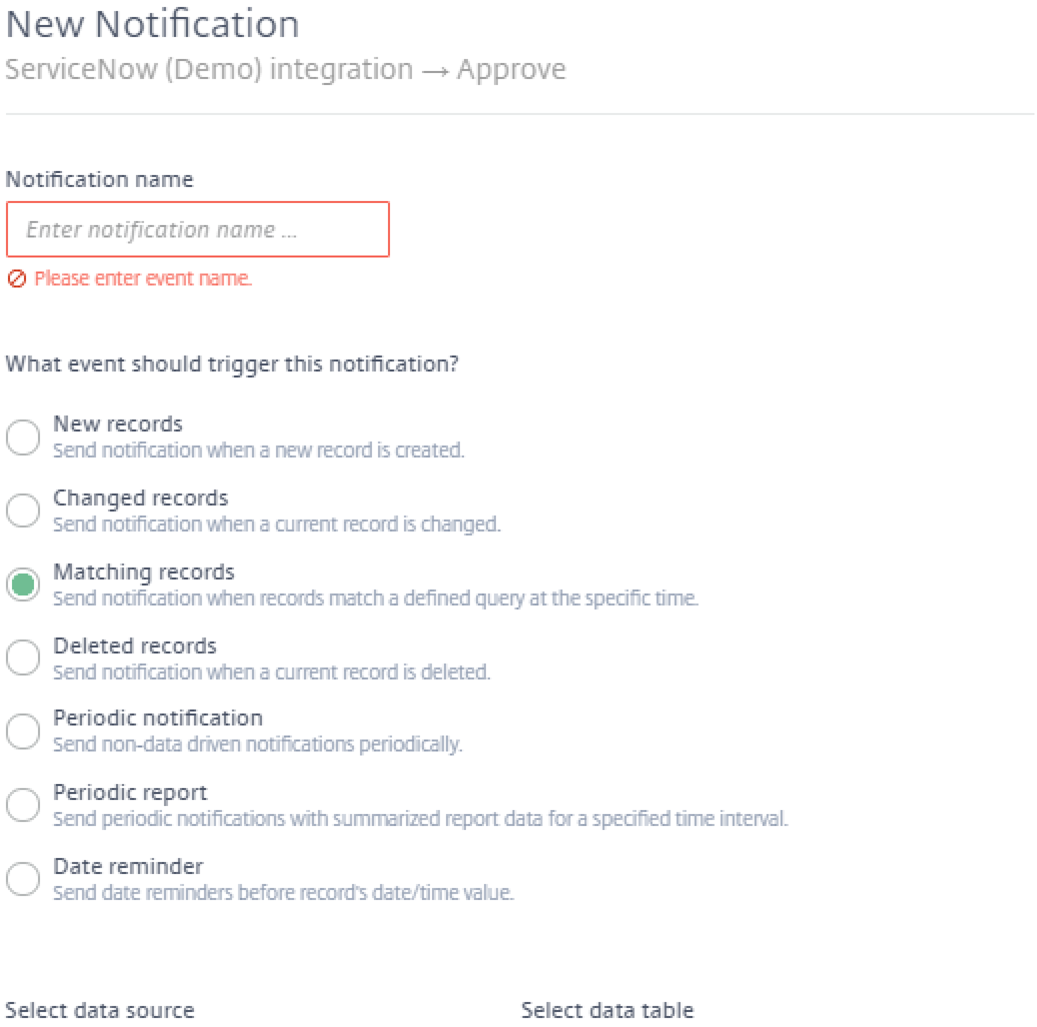
1. Select the microapp to which you want to add an event.
2. Select **Create New Notification** at the bottom of the page.

3. Give the Notification a name.

4. Select your desired trigger and notification type from the following:

   - New Records - Send notification. when a new record is created.
   - Changed records
   - Matching record
   - Delete records
   - Periodic report
   - Periodic notification (user action)
   - Date reminder

   The following screenshot shows what the **New Notification** screen looks like:

   New Notification

   ServiceNow (Demo) integration → Approve

   Notification name

   Enter notification name ...

   Please enter event name.

   What event should trigger this notification?

   ○ New records
   Send notification when a new record is created.

   ○ Changed records
   Send notification when a current record is changed.

   ● Matching records
   Send notification when records match a defined query at the specific time.

   ○ Deleted records
   Send notification when a current record is deleted.

   ○ Periodic notification
   Send non-data driven notifications periodically.

   ○ Periodic report
   Send periodic notifications with summarized report data for a specified time interval.

   ○ Date reminder
   Send date reminders before record's date/time value.

   Select data source

   Select data table

5. Verify the data source and select your data table.

6. Select Add.

The new notification is added to the **Notifications** list and is ready to be configured.

Customize the pages and notifications using the builder

For more information about customizing pages and notifications, see Page builder components and Build event notifications.

Manage subscriptions

Manage microapp subscribers to enable the microapps for specific users and user groups within your organization. For more information, see Assign subscribers.

Assign subscribers

April 28, 2021

Manage your microapps' subscribers to enable the microapp pages and notifications for specific users and user groups within your organization. Subscriptions are managed as individual users or as groups. Subscriptions are managed at the microapp level and assigned to each microapp individually.

Enable administrator access

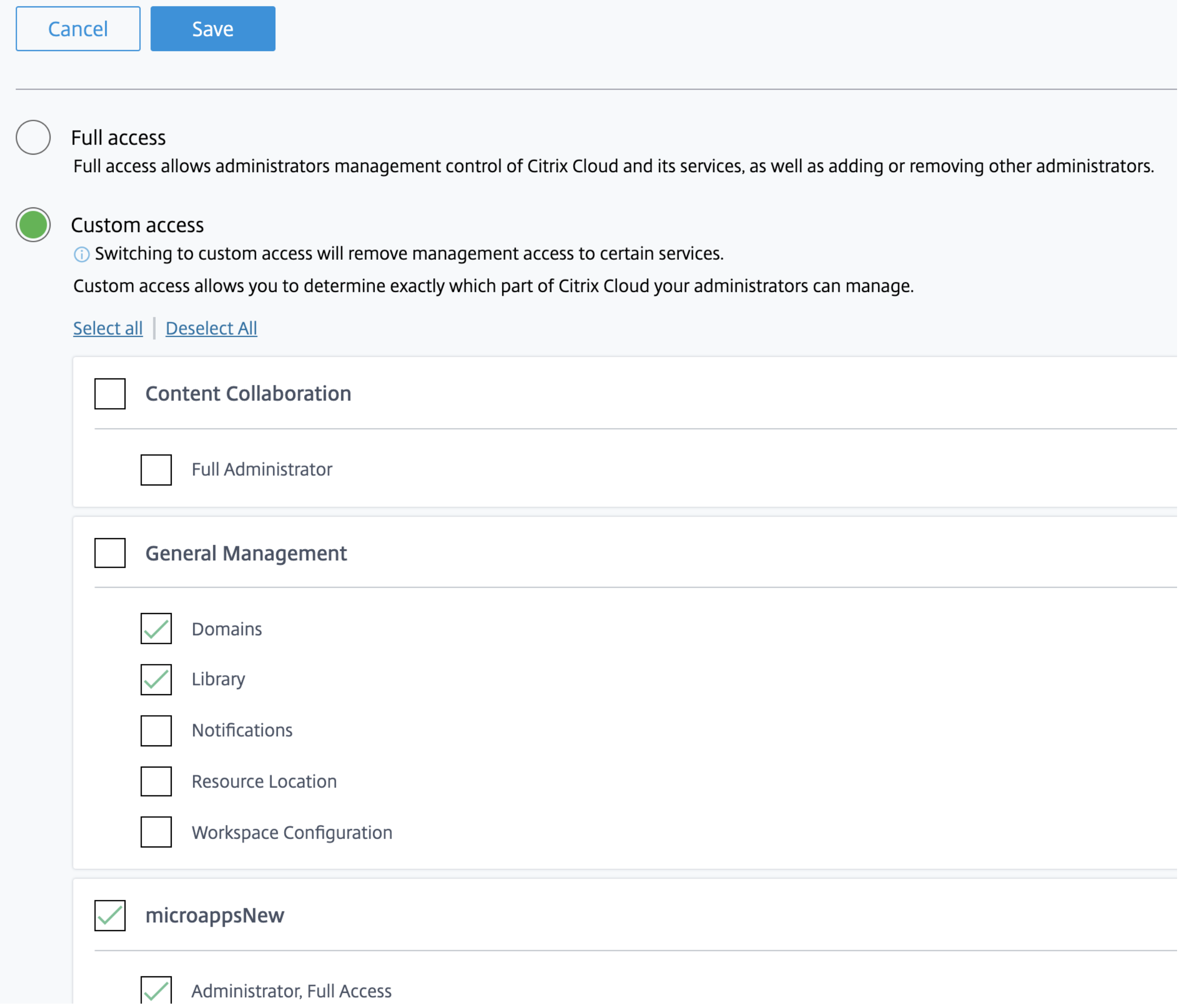
Before you begin, grant correct administrator access to add subscribers to your microapps. Use this delegated permissions process to enable administrators to add subscribers.

1. After signing in to Citrix Cloud, select **Identity and Access Management** from the menu, and select **Administrators**.

   The console shows all the current administrators in the account.

2. Locate the administrator that you want to manage, select the menu (ellipsis) button, and select **Edit Access**.
3. Select **Custom access**.
4. Ensure the following check boxes are selected and then select **Save**:

   - Under **General Management**, select **Domains** and **Library**.
   - Under **microappsNew**, select **Administrator, Full Access**.



○ Full access

Full access allows administrators management control of Citrix Cloud and its services, as well as adding or removing other administrators.

● Custom access

ⓘ Switching to custom access will remove management access to certain services.

Custom access allows you to determine exactly which part of Citrix Cloud your administrators can manage.

Select all | Deselect All

□ Content Collaboration

□ Full Administrator

□ General Management

☑ Domains

☑ Library

□ Notifications

□ Resource Location

□ Workspace Configuration

☑ microappsNew

☑ Administrator, Full Access

Repeat this procedure for all administrators who need to add subscribers. For more information about managing administrators including adding new administrators, see Add administrators to a Citrix Cloud account.

> **Note:**
>
> Granting domains and library admin access allows administrators to assign resources. For more information, see Assign users and groups to service offerings using Library.

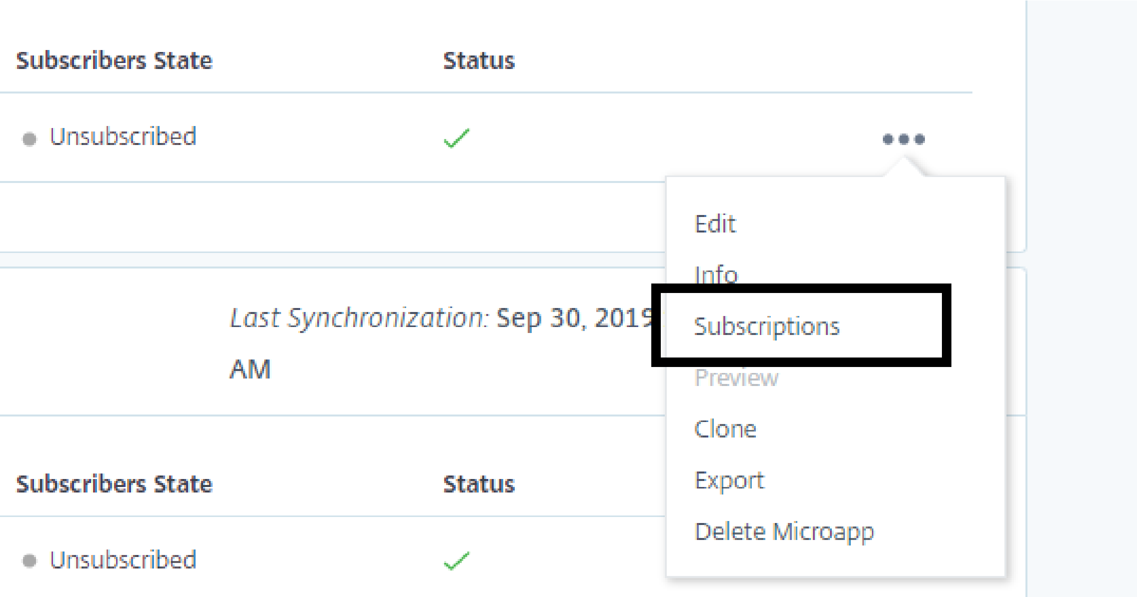
Manage subscribers

Now add specific users and user groups within your organization. Remember, subscriptions are managed at the microapp level and assigned to each microapp individually. To subscribe users to a subset of functionalities it is better to separate applications into multiple microapps.

Follow these steps:

1. From the Integrations page, select the microapp that you want to add subscribers to, and select

Subscriptions.

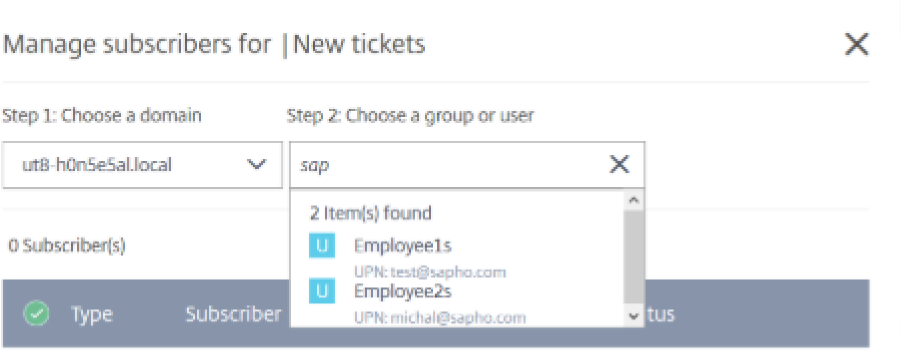


Subscribers State	Status
● Unsubscribed	✓

Last Synchronization: Sep 30, 2019 AM

Subscribers State	Status
● Unsubscribed	✓

2. Under **Step 1: Choose a domain**, select the domain that you would like to use for this microapp.
3. Under **Step 2: Choose a group or user**, use the search to find the groups or users that you want to subscribe to the microapp. Select one or more groups or users.



Manage subscribers for | New tickets

Step 1: Choose a domain: ut8-h0n5e5al.local

Step 2: Choose a group or user: sap

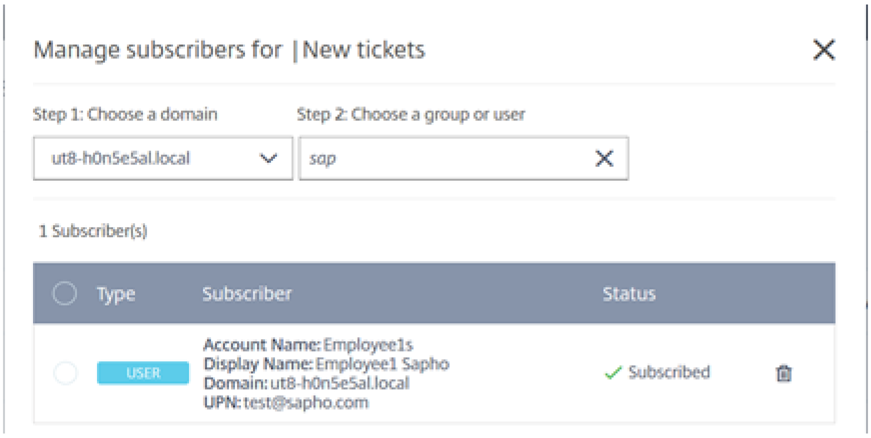
2 Item(s) found

U Employee1s UPN: test@sapho.com

U Employee2s UPN: michal@sapho.com

0 Subscriber(s)

Subscribers are shown in the following subscribers list. You can check to make sure that their status shows as **Subscribed**.



Manage subscribers for | New tickets

Step 1: Choose a domain: ut8-h0n5e5al.local

Step 2: Choose a group or user: sap

1 Subscriber(s)

Type	Subscriber	Status
USER	Account Name: Employee1s Display Name: Employee1 Sapho Domain: ut8-h0n5e5al.local UPN: test@sapho.com	✓ Subscribed

> **Note:**
>
> After you unsubscribe users or groups from a microapp, there is a delay of approximately five minutes until changes take effect in Citrix Workspace. During this time, users can still access these affected microapps in Citrix Workspace.

Where to go next

To learn more about defining identity providers and accounts, see Identity and Access Management.

Optimize workflows

March 31, 2022

Simplify valuable workflows with Citrix Workspace, harnessing microapp technology with out-of-the-box templates available today. These use cases give employees a consistent and modern experience independent of the legacy systems they leverage, providing a simplified and effective way to perform important departmental workflows.

Workspace Tools Starter Pack

With Citrix Workspace, companies can provide a consistent work experience on any device, enabling employees to quickly find the IT resources that they need, when they need them. Leveraging a new portfolio of employee engagement and self-service microapps within the workspace, organizations can reduce time spent by employees on IT tasks, improve overall employee NPS for IT services, and consistently communicate and share relevant information with employees.

This starter pack is our guide for IT admins leveraging Citrix Workspace to improve employee experience and offers easy tools provided by Citrix to engage employees, monitor usage and feedback and measure the ROI of investing in a best class digital experience as you roll out Workspace to employees.

Integration options include Citrix Cloud admin status, Citrix DaaS self-service apps, employee broadcast, FAQ and surveys to measure engagement, get feedback and understand employee satisfaction with Citrix Workspace.

Recommended apps and integrations:

- Citrix Cloud admin status
- Citrix DaaS self-service
- Employee Broadcast
- Employee FAQ
- Employee Experience Survey

To find out more, see Workspace Tools Starter Pack.

IT Self-service

IT Self-service workflows enable employees to quickly find the IT resources that they need, when they need them. Leveraging this new portfolio of IT Self-service microapps within the workspace, organizations can reduce time spent by employees on IT tasks, improve overall employee NPS for IT services, and reduce IT case volume.

This use case is available through the Microapps service via our out-of-the-box template integrations with:

- ServiceNow integration: Submit Incident microapp and Incidents microapp
- Zendesk integration: Add Ticket microapp and Tickets microapp
- Jira: Create Ticket microapp and Tickets microapp

To find out more, see IT Self-service.

HR Self-service

It is more essential than ever that businesses rethink their people strategies, placing new emphasis on delivering a best-in-class employee experience that differentiates and elevates the business. Using this new portfolio of HR self-service microapps within the workspace, organizations can improve process efficiency, time savings and reduce HR case volume.

This use case is available through the Microapps service via our out-of-the-box template integrations with:

- Workday integration: Create PTO Request microapp and PTO Balance microapp
- SAP SuccessFactors: Directory microapp and Learning microapp
- Kronos Workforce Central: Request Time Off microapp and My Accrual Balance microapp

To find out more, see HR Self-service.

Sales Productivity

Your Sales teams are critical to your organization. Empower them to spend more time driving business, and less time searching for information and inputting notes across multiple systems. Using the new Sales Productivity microapps within the workspace, organizations can accelerate time-to-close through greater account insights, increase visibility of sales exceptions and process delays, and reduce time spent on administrative tasks. Simplify workflows like lead creation, opportunity conversion, and task management.

This use case is available through the Microapps service via our out-of-the-box template integrations with:

- Salesforce integration: Create Lead, Create Contact, Create Contract, Create Opportunity, Create Task, Contracts, and Opportunities microapps
- MS Dynamics CRM integration: Create Lead, Create Contact, Create Opportunity, Create Task, and Opportunities microapps

To find out more, see Sales Productivity.

Employee Well-being

Deliver a workspace that integrates well-being into the way people like to work. There's no doubt that employees can benefit from well-being tools that help them manage the stress and complexities of the workday. The challenge is getting those tools to employees without adding yet another item to their to-do list. Teams can use Citrix Workspace technology to improve the overall employee experience by delivering well-being tools and resources within an intelligent feed.

This use case is available through the Microapps service via our out-of-the-box template integrations with Citrix Podio. Available microapps include:

- Broadcast microapps – Customize and send a dynamic message to employees' intelligent feeds.
- FAQ microapp – Compile a list of FAQs or table of information, communicated and expandable within employees' intelligent feeds.

To find out more, see Employee Well-being.

Video resources

Check out these videos for demos of these workflows:

IT Self-service microapp Demo

HR Self-service microapp Demo

Sales Productivity microapp Demo

Employee Well-being Demo