Microapps

Contents

Integrate RSS

October 8, 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 **Depricated** 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.

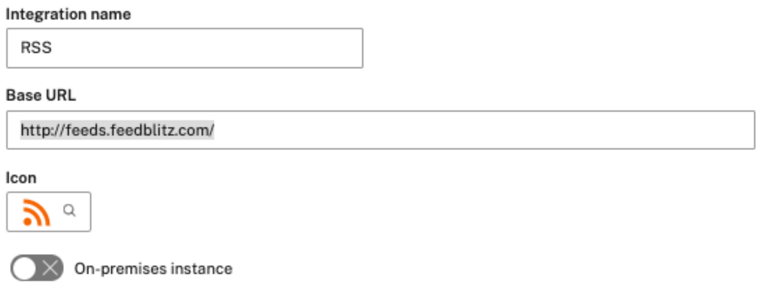
1. To replace the RSS endpoint, from the **Microapp Integrations** page, select the menu next to the RSS integration, and then **Edit**. The **Data Loading** screen opens. If you are in the configuration screen, select **Data Loading** from the left side navigation column.
2. Select the RSS endpoint.
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.

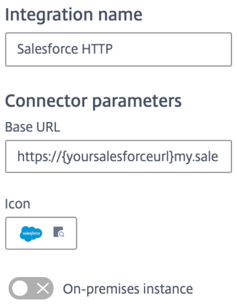
Add the Salesforce integration to Citrix Workspace Microapps

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.

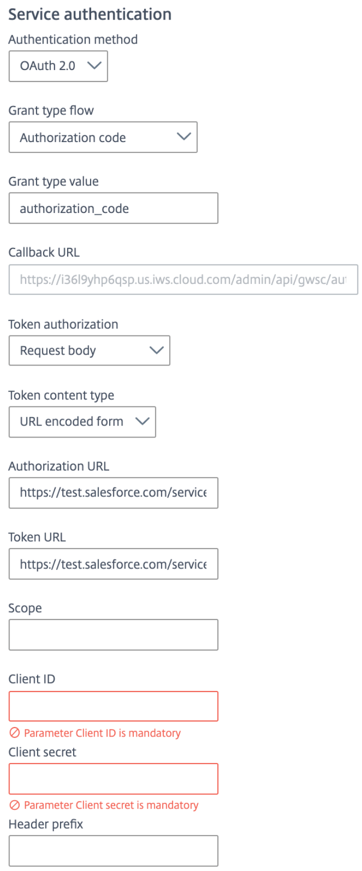
   b) Select **Request body** from the **Token authorization** menu.

   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:

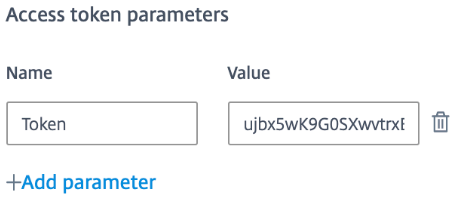
⊘ Parameter Client ID is mandatory

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

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.

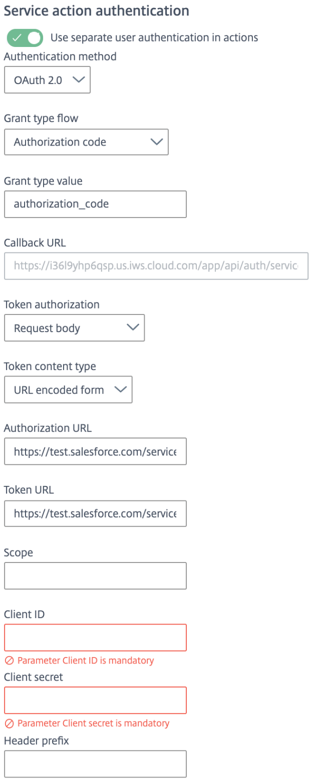
d) The **Authorization URL** and **Token URL** are prefilled. Endpoints require secure HTTP (HTTPS). Instead of using login.salesforce.com, you can also use the My Domain, community, or test.salesforce.com (sandbox) domains in these endpoints.

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:

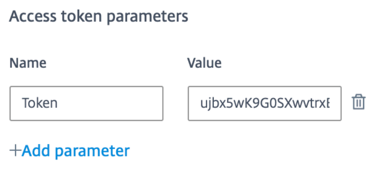
⊘ Parameter Client ID is mandatory

Client secret:

⊘ Parameter Client secret is mandatory

Header prefix:

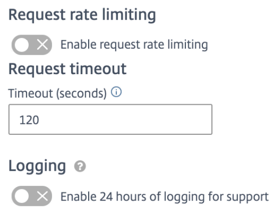
8. Again, select **Add Parameter** to include **Access token parameters**. Enter *Token* for **Name** and `{ yoursecuritytoken }` for **Value**. This parameter is required by the target application authorization server.



Access token parameters

Name	Value
Token	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

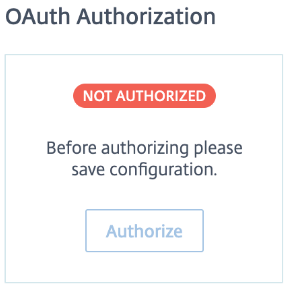
Logging

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

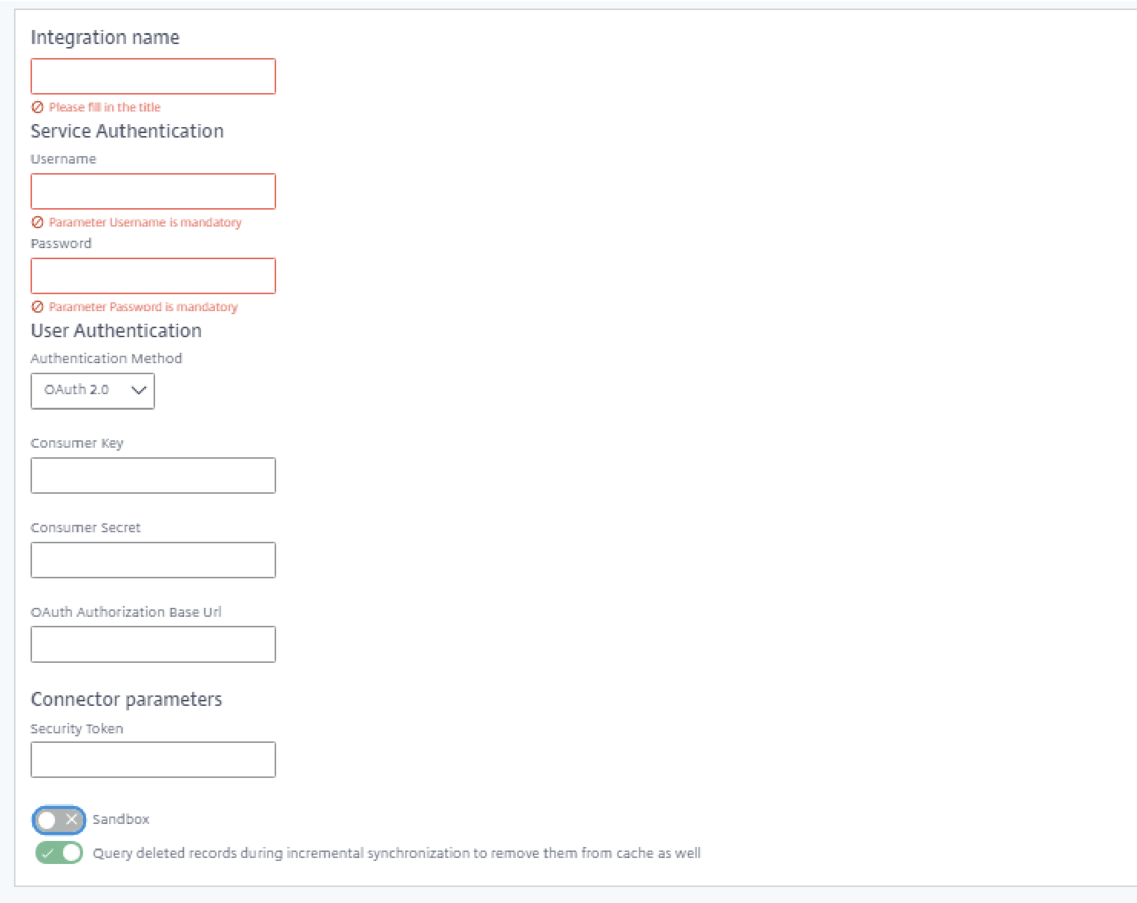
Add the Salesforce integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace.

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.

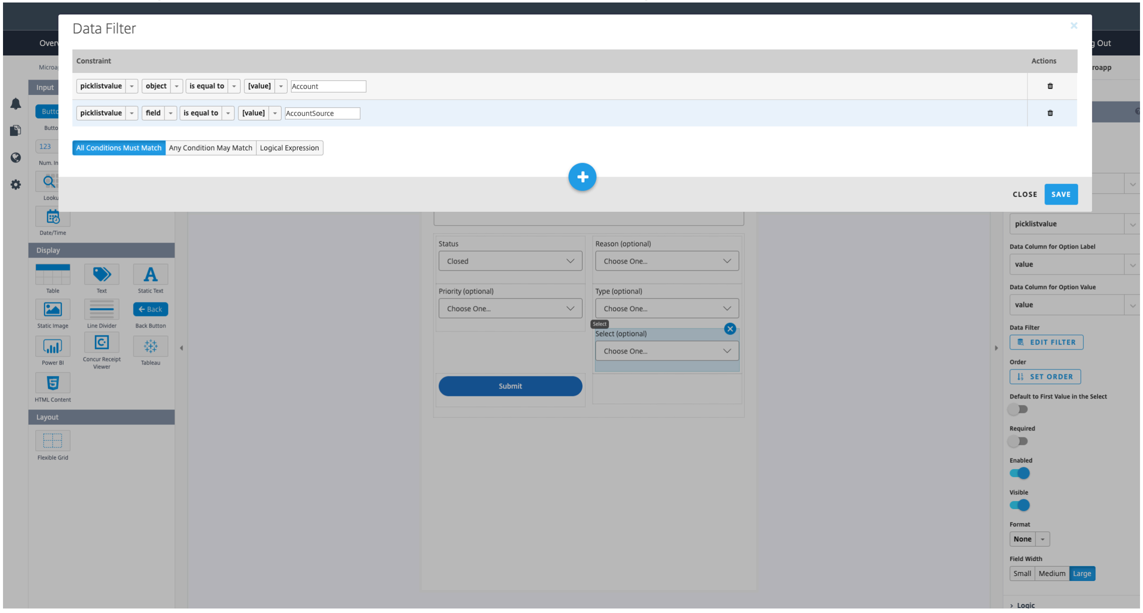
Add picklists’ values table

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.

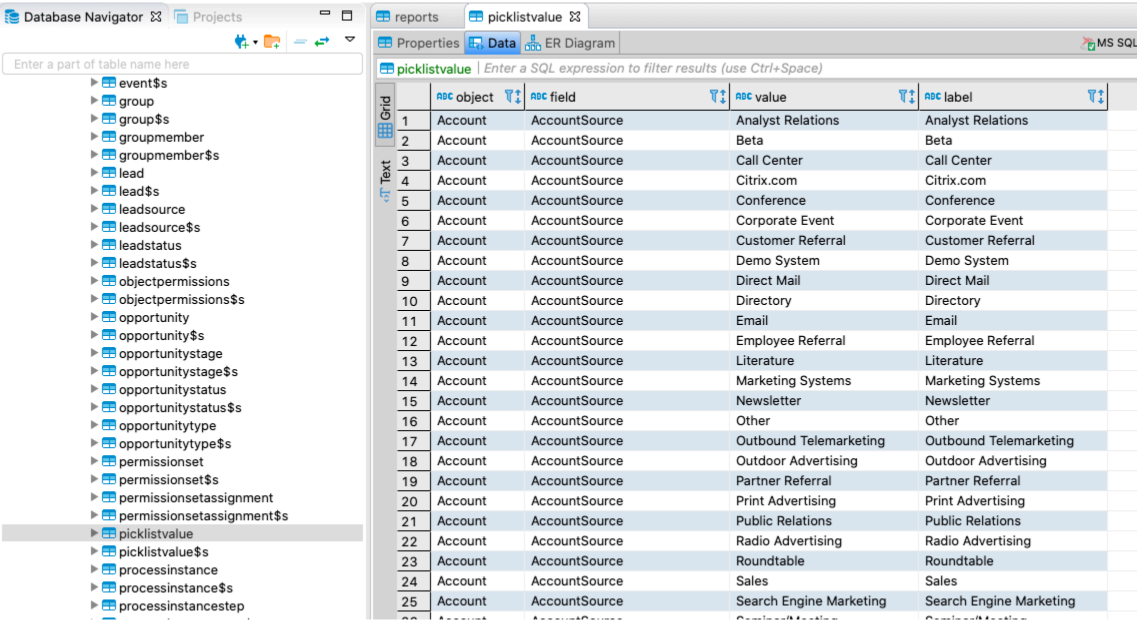
   The following screenshot shows an example of setting up the filter:



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

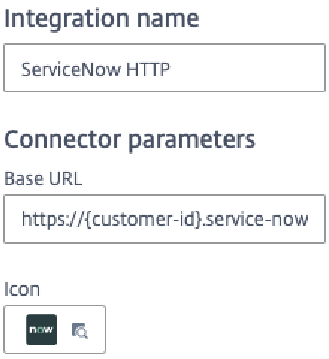
Add the ServiceNow integration to Citrix Workspace Microapps to connect to your application. This delivers out-of-the-box microapps with pre-configured notifications and actions which are ready to use within your Workspace. We provide two ServiceNow integration templates for your use. We recommend using 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

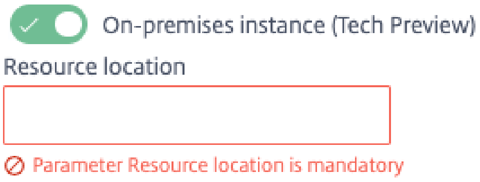
Connector parameters

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.

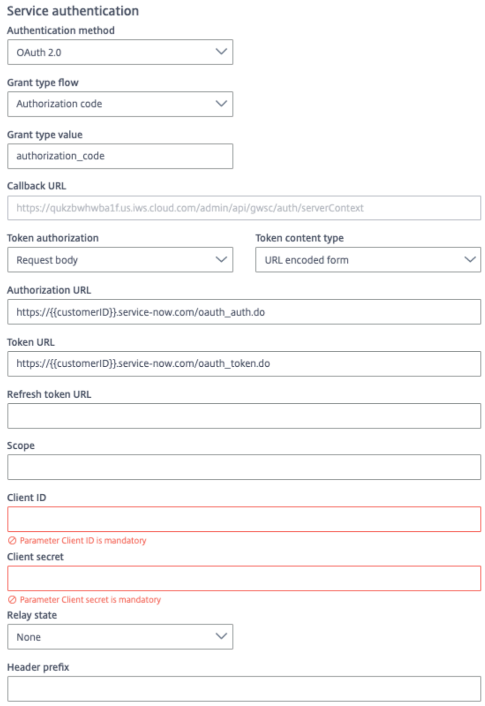
   c) Enter your **Token URL** or simply replace `{ customer-id }` in the example with your customer ID: `https://{ customer-id } .service-now.com/oauth_token.do`. This is the URL of the access authorization token.

   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

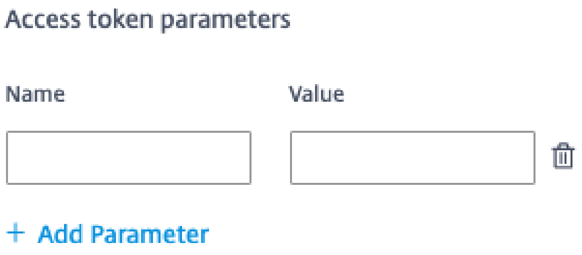
Client secret:

⊘ 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

+ 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.

   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.

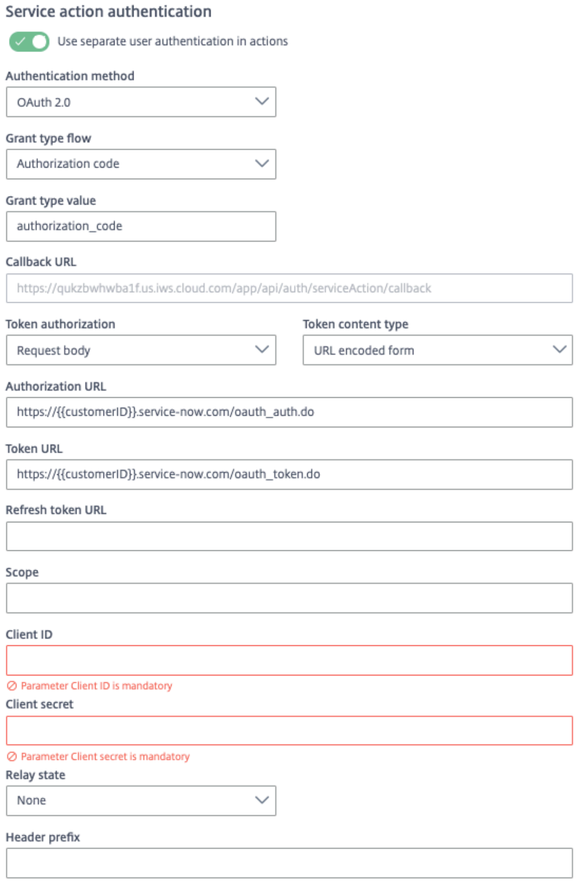
f) (Optional) Enter your **Scope** to define the scope of the access request. This string is defined by the authorization server when setting up your target integration application.

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

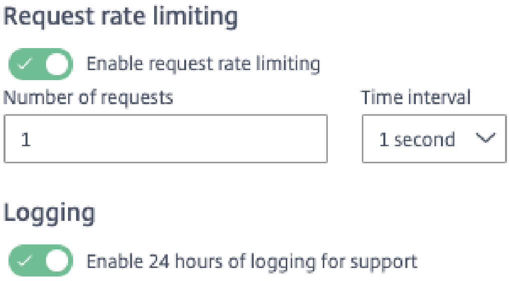
Client secret:

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

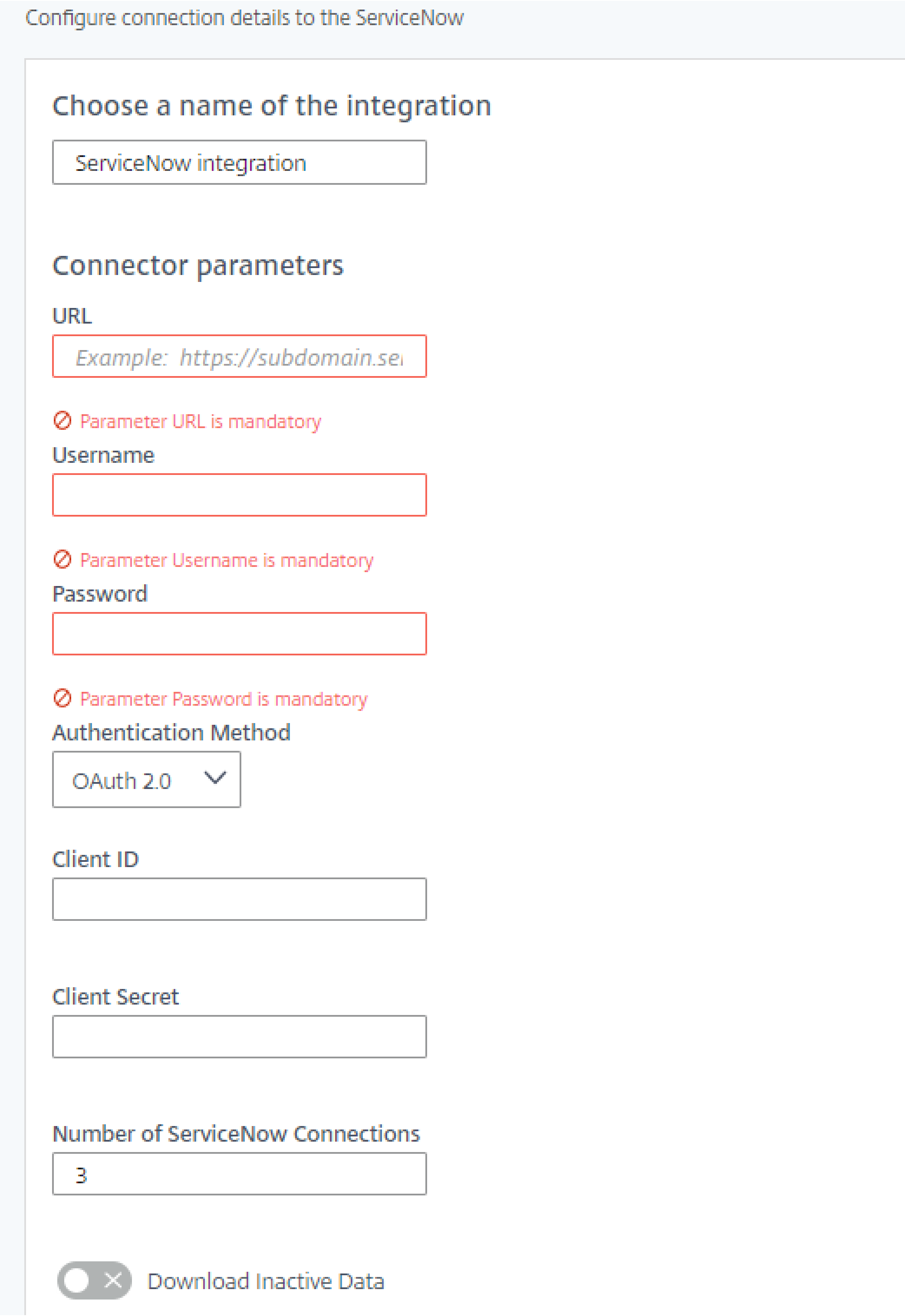
Follow these instructions to set up the legacy java-based ServiceNow integration.

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.

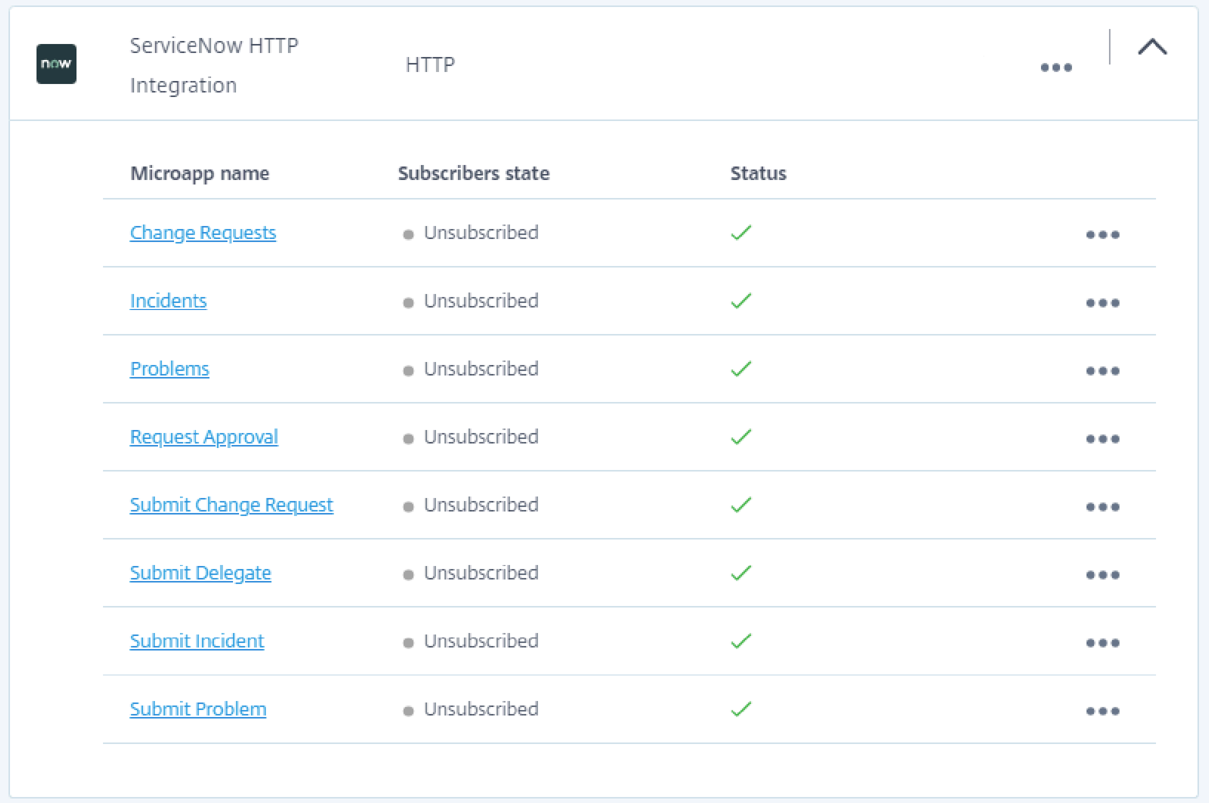
For more details of API endpoints and table entities, see ServiceNow HTTP connector specifications or ServiceNow connector specifications.

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:

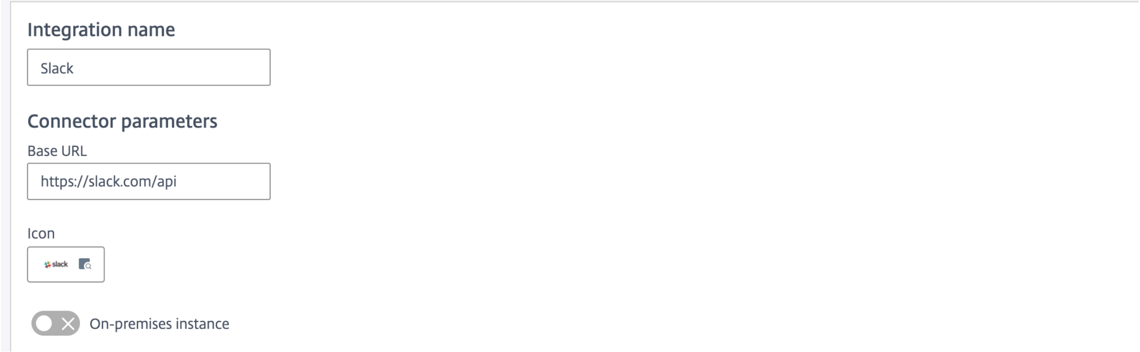
- Do not add a Bot to a noisy channel. Bots should be added to channels that are used for infrequent, time-critical communications within a select group, such as urgent sales issues for the Sales Group or IT security for General Employees.
- Multiple integrations can point to the same Slack app.
- Keep 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.



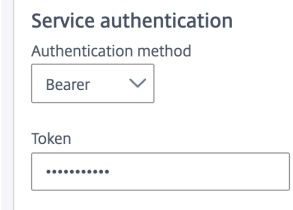
Integration name
Slack

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

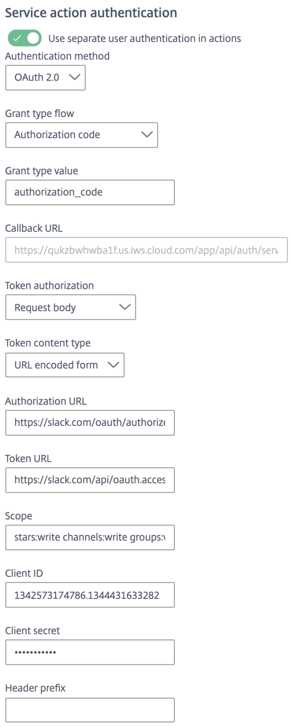
Bearer

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: •••••••••••

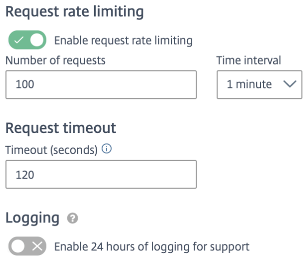
Header prefix:

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

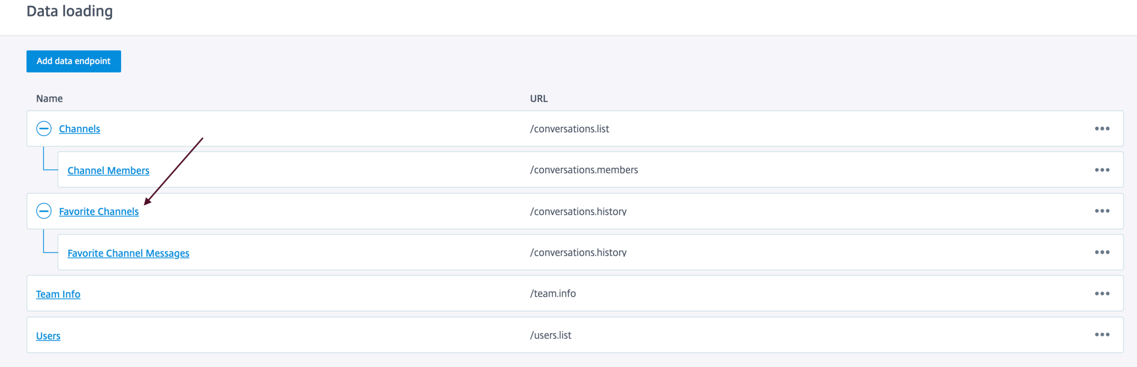
To complete this set up, you need to add the `channel` value with your channel ID collect in Create a favorites channel and collect Channel ID. Modify the **Favorite Channels** endpoint and both the **Favorite channel** and **Unfavorite** service actions.

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**.

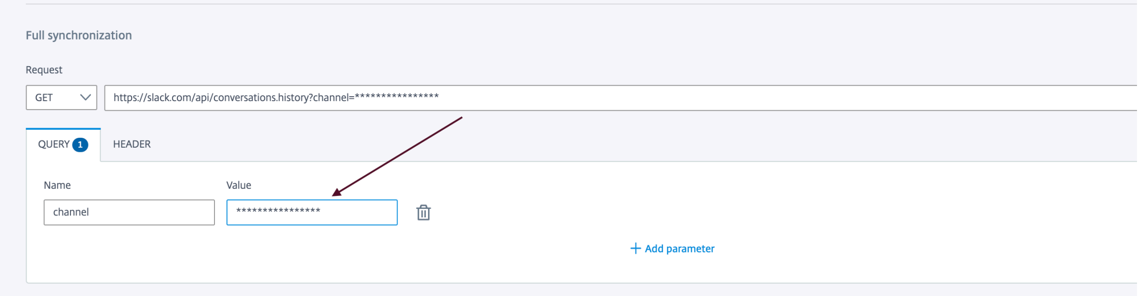


Data loading

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=***************

Name: channel

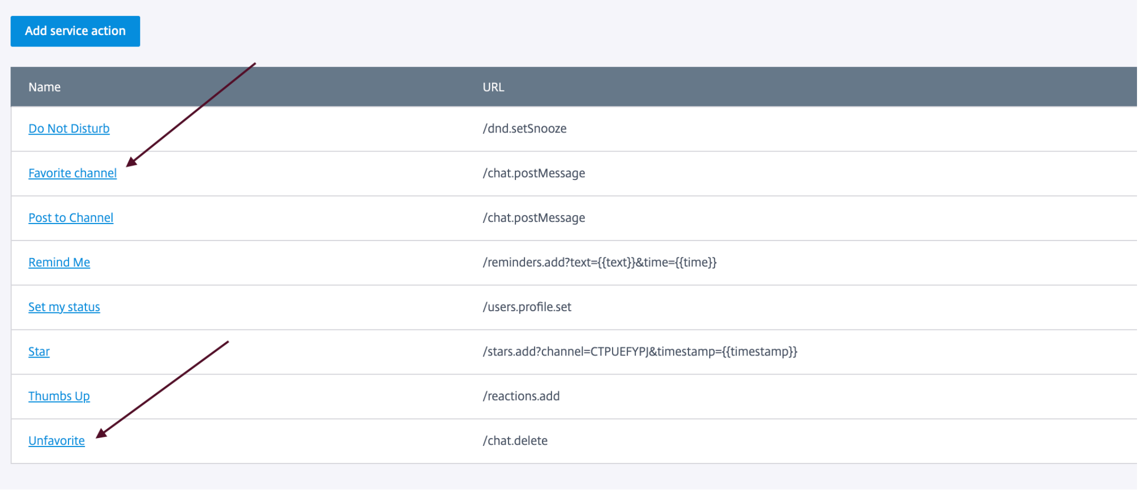
Value: ***************

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**.

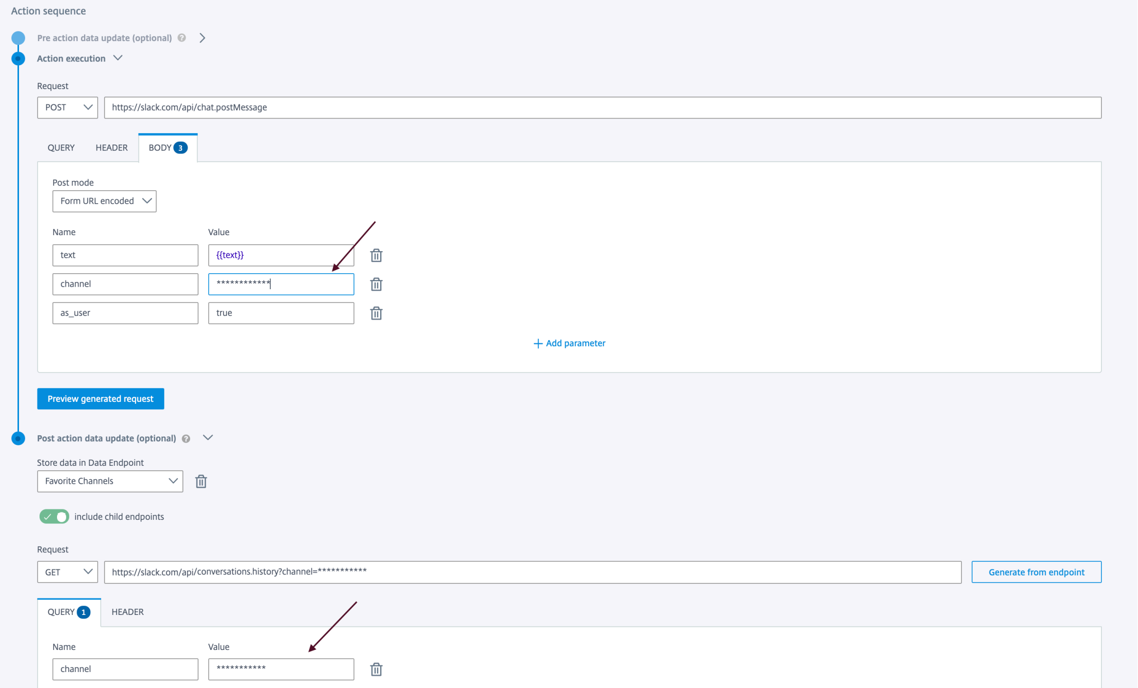


Name	URL
Do Not Disturb	/dnd.setSnooze
Favorite channel	/chat.postMessage
Post to Channel	/chat.postMessage
Remind Me	/reminders.add?text={{text}}&time={{time}}
Set my status	/users.profile.set
Star	/stars.add?channel=CTPUEFYPJ&timestamp={{timestamp}}
Thumbs Up	/reactions.add
Unfavorite	/chat.delete

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

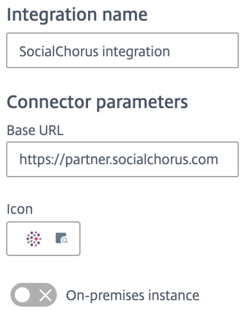
SocialChorus APIs are enabled via webservices for a paid account by default. This may require a separate agreement with the vendor. The number of API requests that can be made to specific resources is limited. We recommend reviewing SocialChorus API guidance: SocialChorus API Guidance.

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

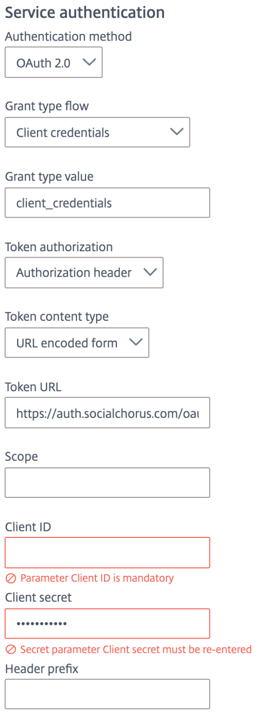
Base URL

https://partner.socialchorus.com

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) **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/oa

Scope:

Client ID:

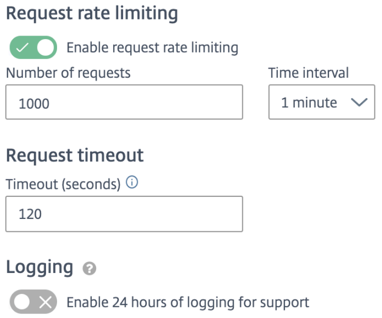
Parameter Client ID is mandatory

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 pre-configured 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:

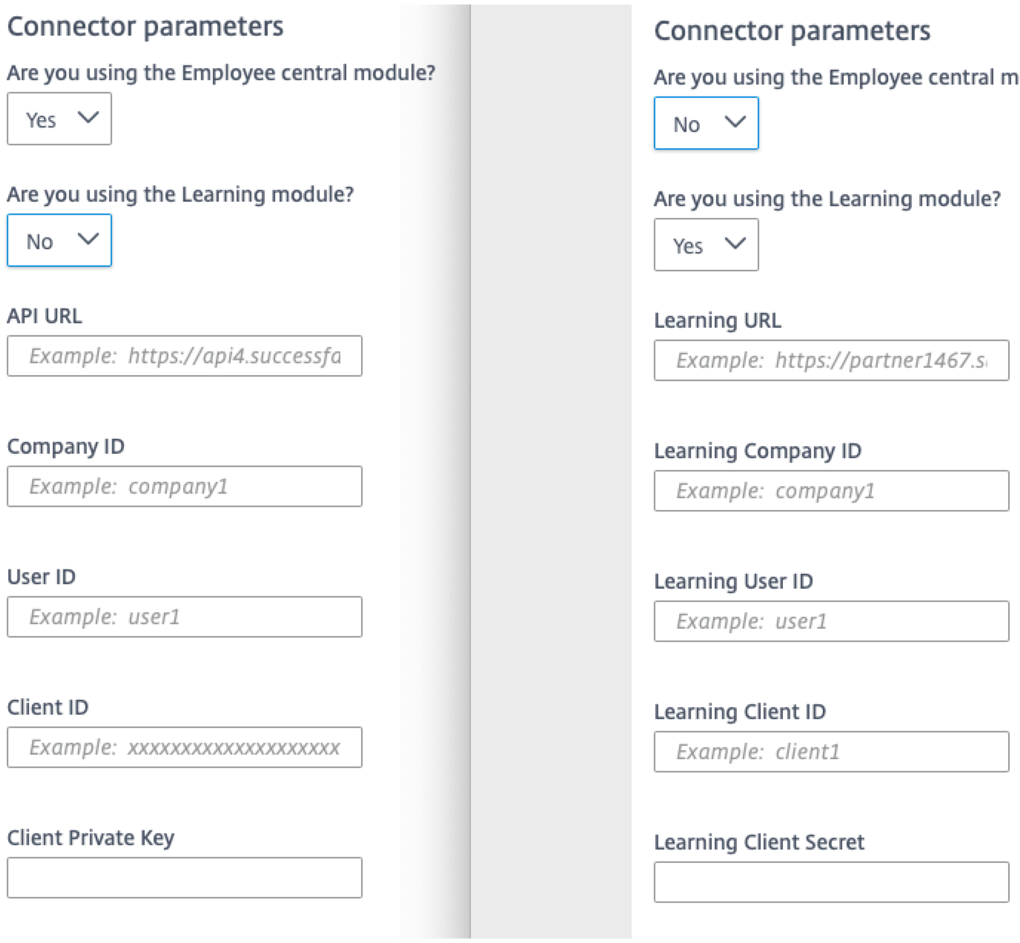
- the basic SuccessFactors HCM integration,
- the basic integration with Learning Module,
- just the Learning module.

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.

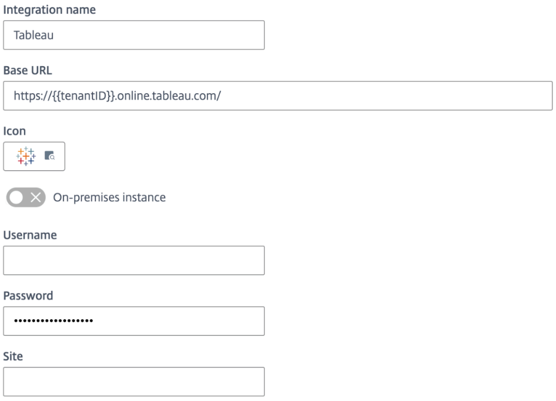
Add the integration to Microapps

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

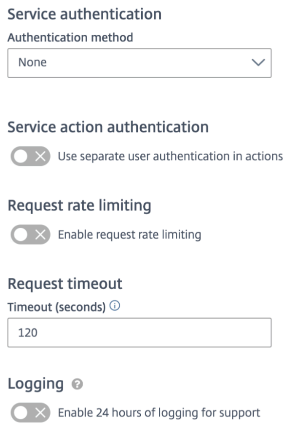
On-premises instance

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:

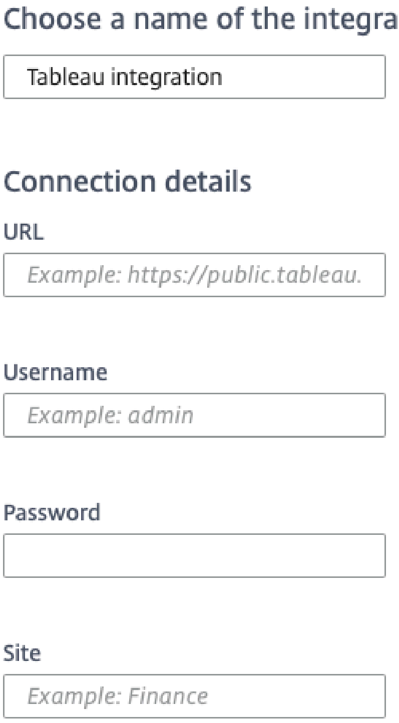
- URL
- Username
- Password
- Site
- Configure Tableau Server to recognize and trust requests by whitelisting its IP address.
- Only https connections are supported. Make sure the SSL certificate is trusted
- Configure Citrix Gateway to support single sign-on for Tableau so that once users log in they 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**.

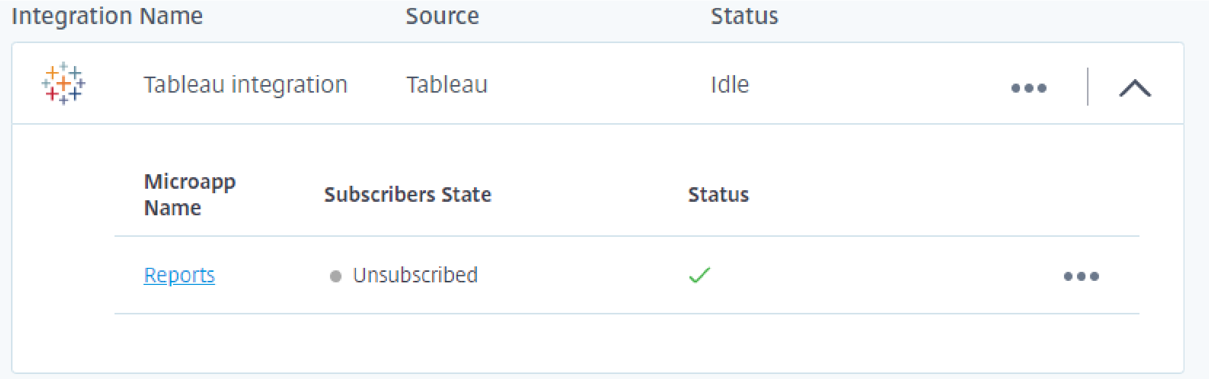
The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

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.

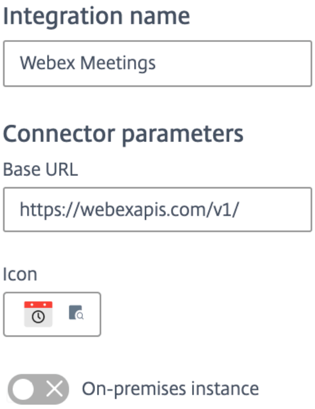
7. Select **Add Integration** after you complete all the required fields.
8. Copy and save the **ClientId** and **Secret** shown on the screen. You use these details for **Service Action Authentication** while configuring the integration.

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.

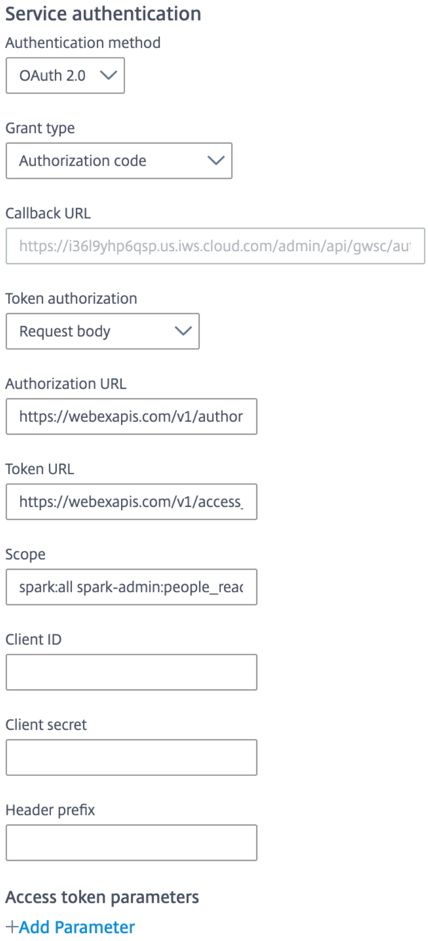
c) The **Authorization URL** is prefilled: `https://webexapis.com/v1/authorize`

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:

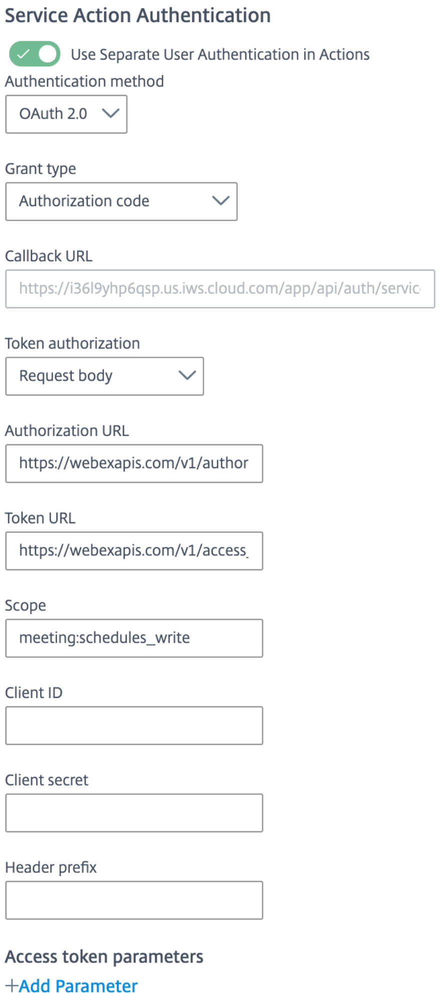
Header prefix:

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:

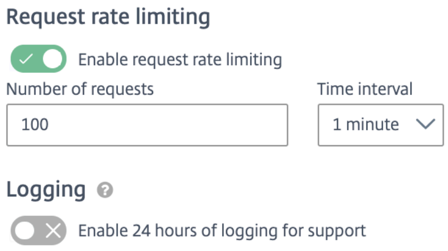
Client secret:

Header prefix:

Access token parameters

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

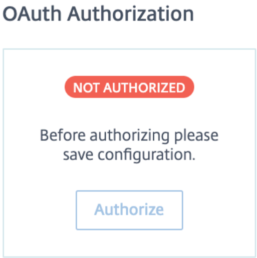
Enable 24 hours of logging for support

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**.

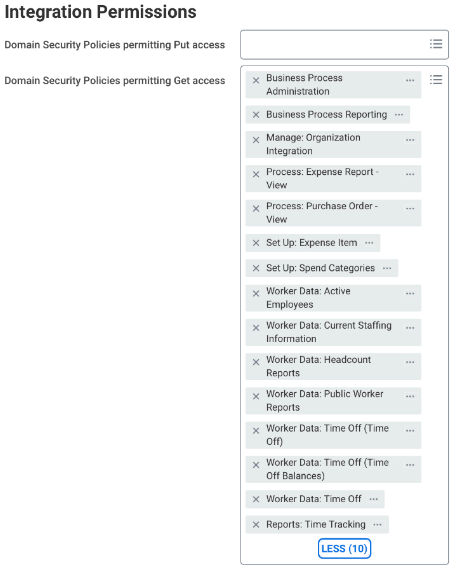
Manage security group permissions

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**.

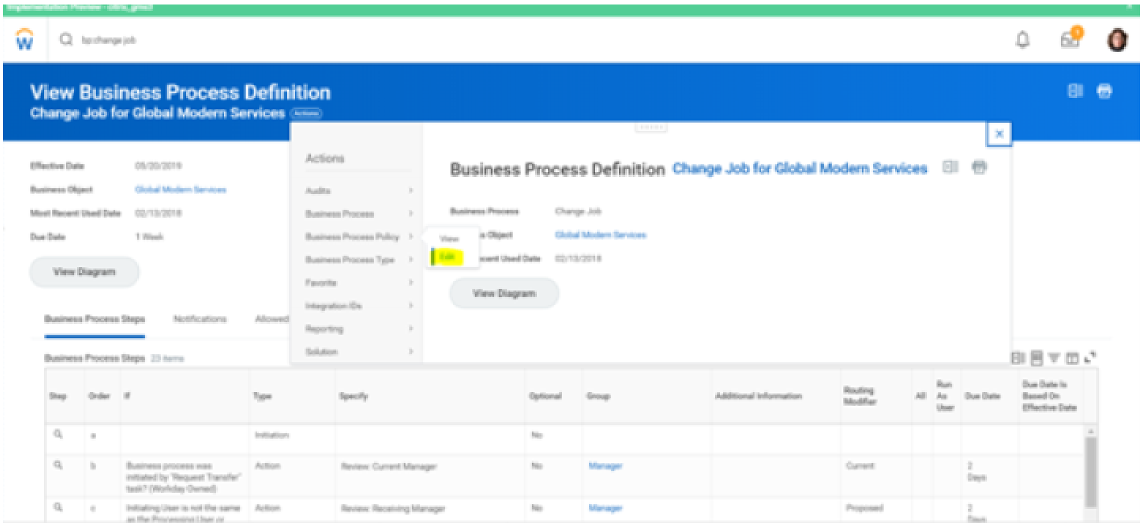
6. Search for *Activate Pending Security Policy Changes*.

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 Object	Global Modern Services
Most Recent Used Date	02/13/2018

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 }`.

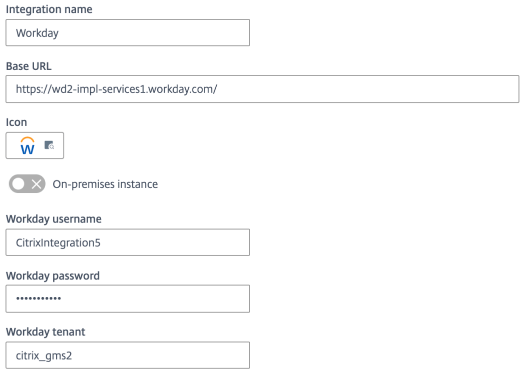
Add the Workday integration to Citrix Workspace Microapps

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

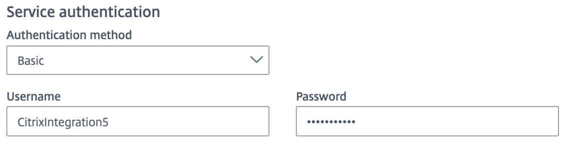
Workday username: CitrixIntegration5

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

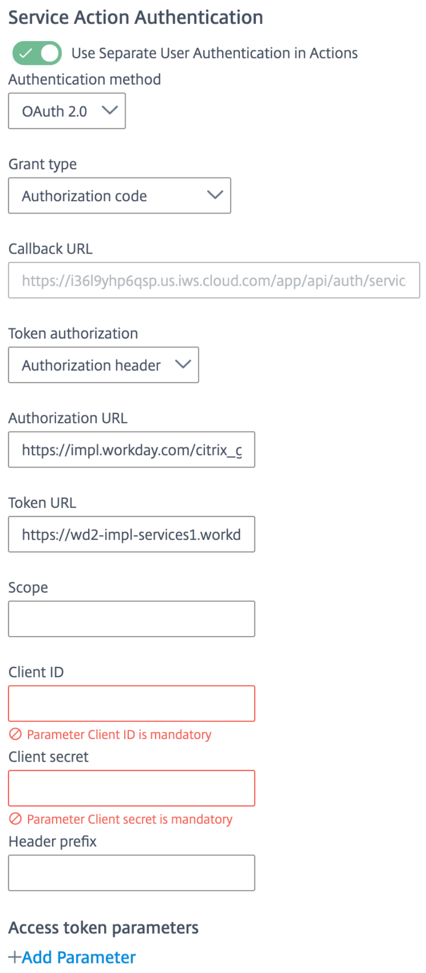
    c) Select **Authorization header** from the **Token authorization** menu.

    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:

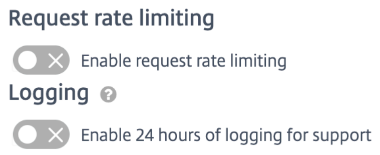
Parameter Client secret is mandatory

Header prefix:

Access token parameters

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 ✗

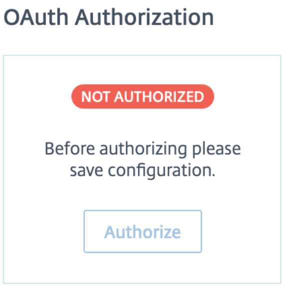
Logging

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

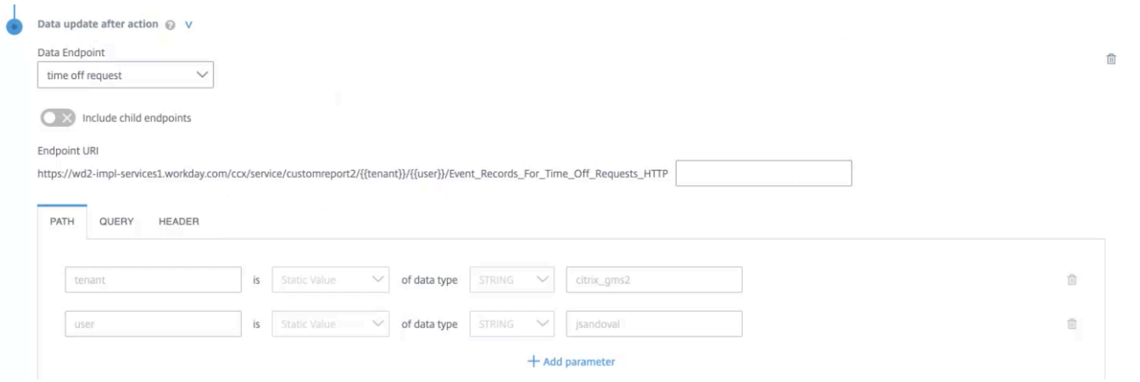
NOT AUTHORIZED

Before authorizing please save configuration.

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

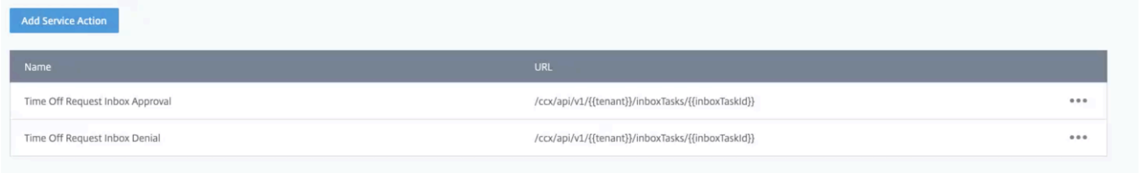
Endpoint URI

https://wd2-impl-services1.workday.com/ccx/service/customreport2/{{tenant}}/{{user}}/Event_Records_For_Time_Off_Requests_HTTP

tenant is Static Value of data type STRING citrix_gms2

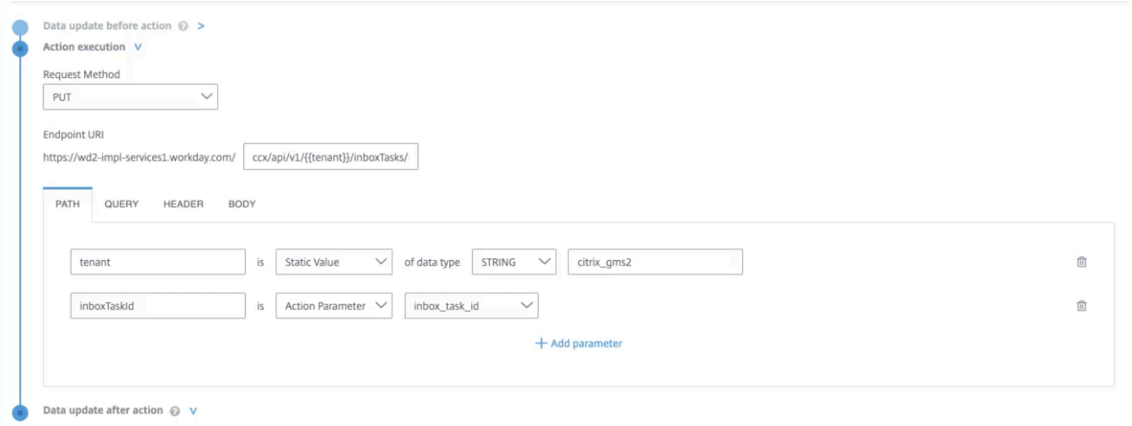
user is Static Value of data type STRING jsandoval

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

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

Data update after action

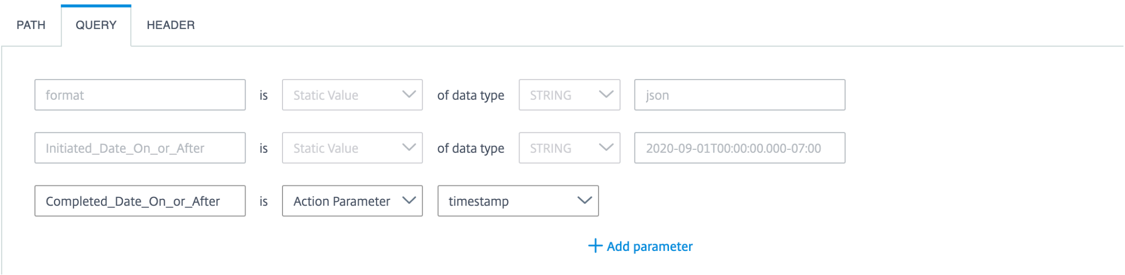
7. You need to delete and recreate **Data update after action**. Next to **Data update after action** label, select the **delete** icon to remove this section.

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

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`

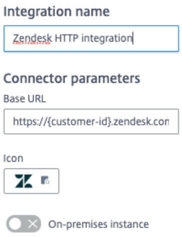
Add the integration to Citrix Workspace Microapps

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

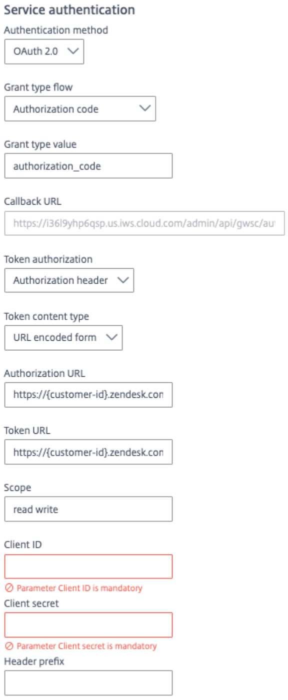
On-premises instance

    - 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

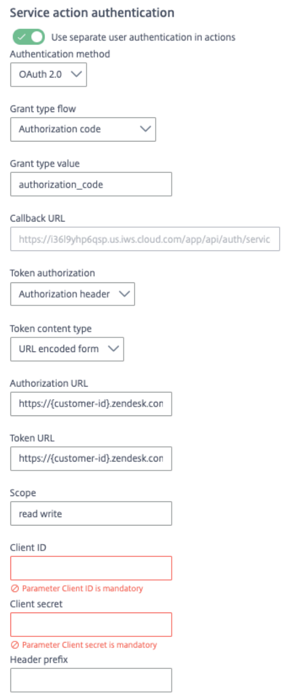
Header prefix:

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:

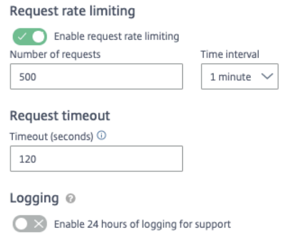
Parameter Client ID is mandatory

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

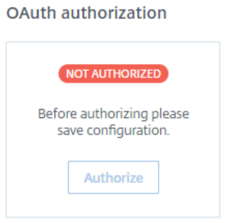
Logging

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

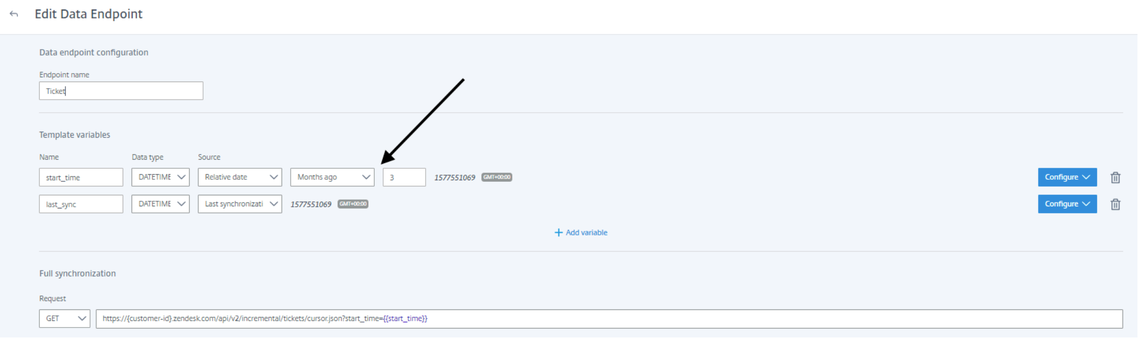
Before authorizing please save configuration.

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
last_sync	DATETIME	Last synchronizati	1577551069 GMT+00:00		

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.

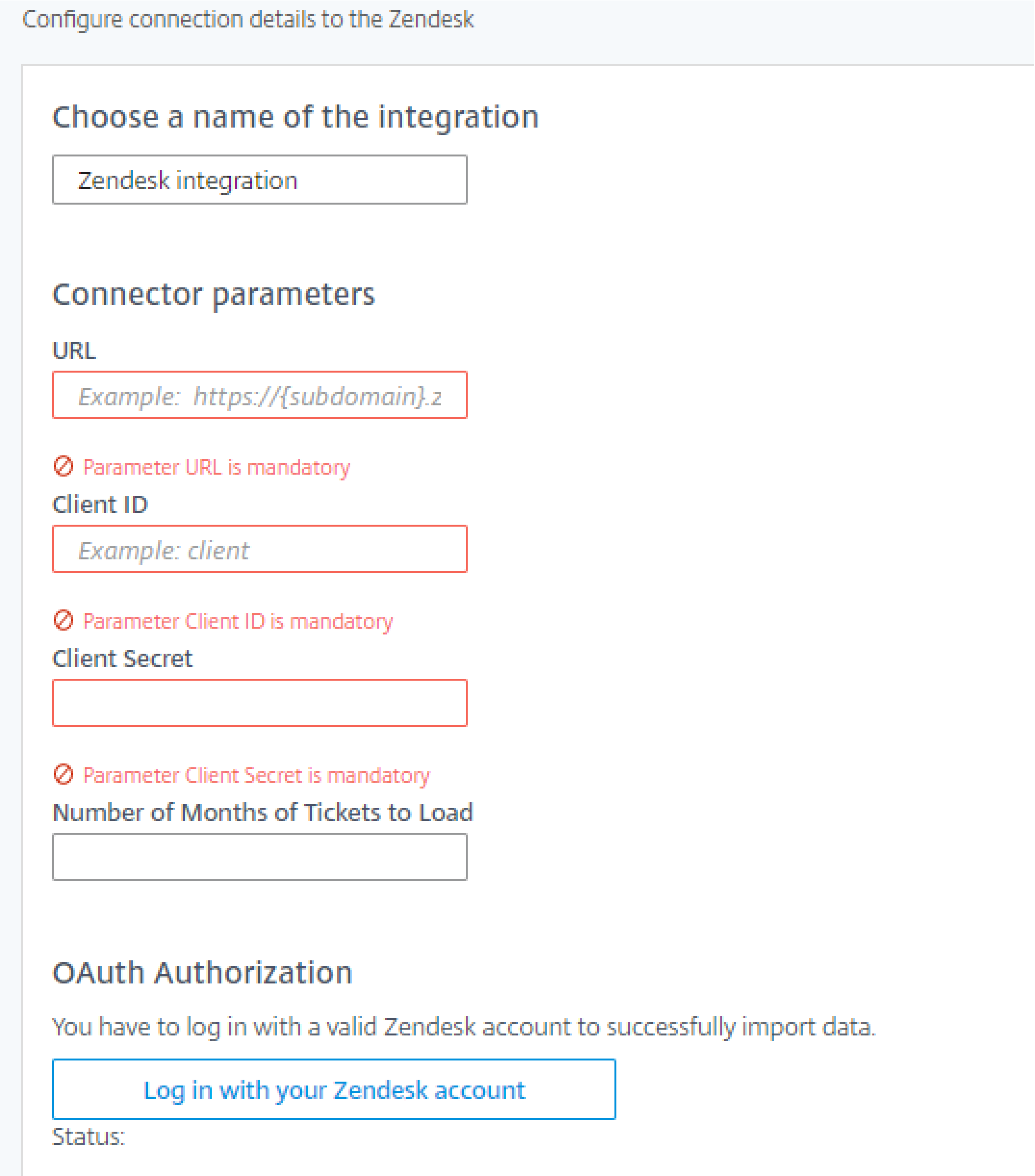
Notification or Page	Use-case workflows
New Ticket Assigned To You (changed) notification	When an existing ticket is assigned to a user, they receive a notification.
New Ticket Assigned To You (new) notification	When a new ticket is assigned to a user, they receive a notification.
Ticket Status Change 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.

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**.

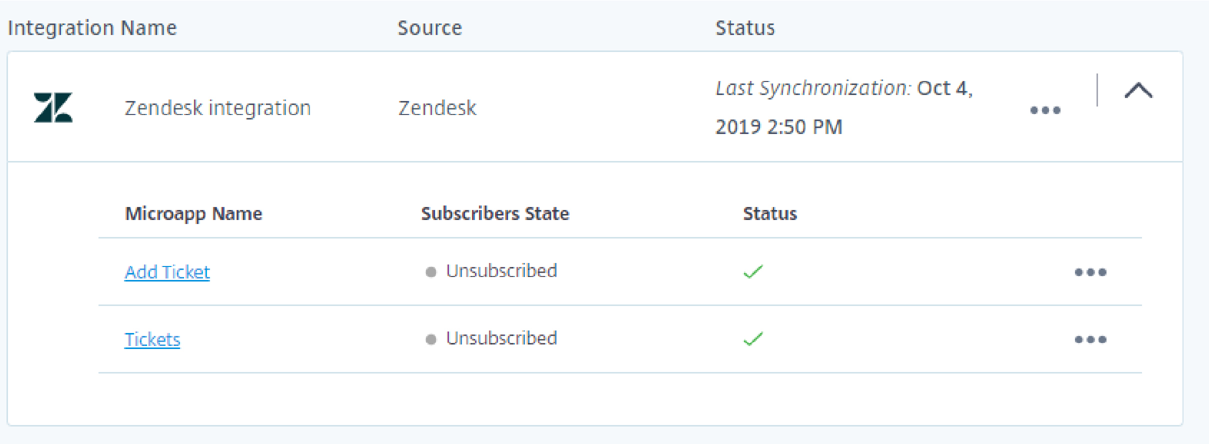
The **Microapp Integrations** page opens with your added integration and its microapps. From here you can add another integration, continue setting up your out-of-the-box microapps, or create a new microapp for this integration.

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**.

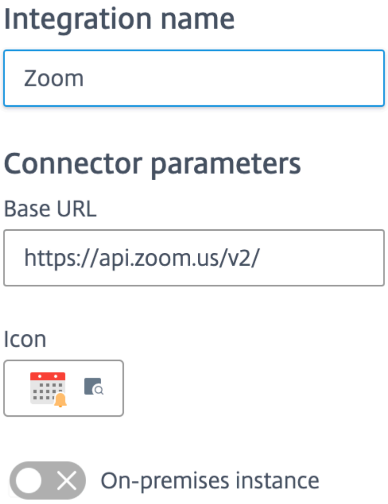
8. Under the Scopes section, select **Add Scopes** and select the scopes for **Meeting**, **Recording**, and **User**. Then select **Done**.
9. Select **Install**. A new tab opens to authorize the app then select **Authorize** and close the tab.
10. Copy and save the **ClientId** and **Secret** shown 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

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`

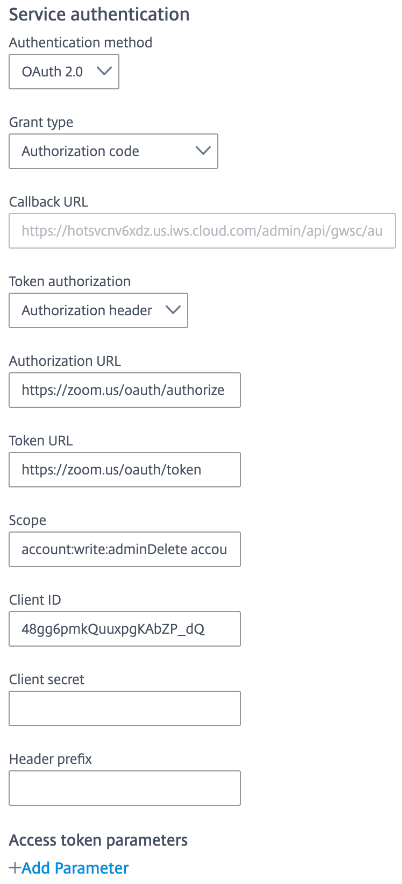
d) The **Token URL** is prefilled: `https://zoom.us/oauth/token`

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:

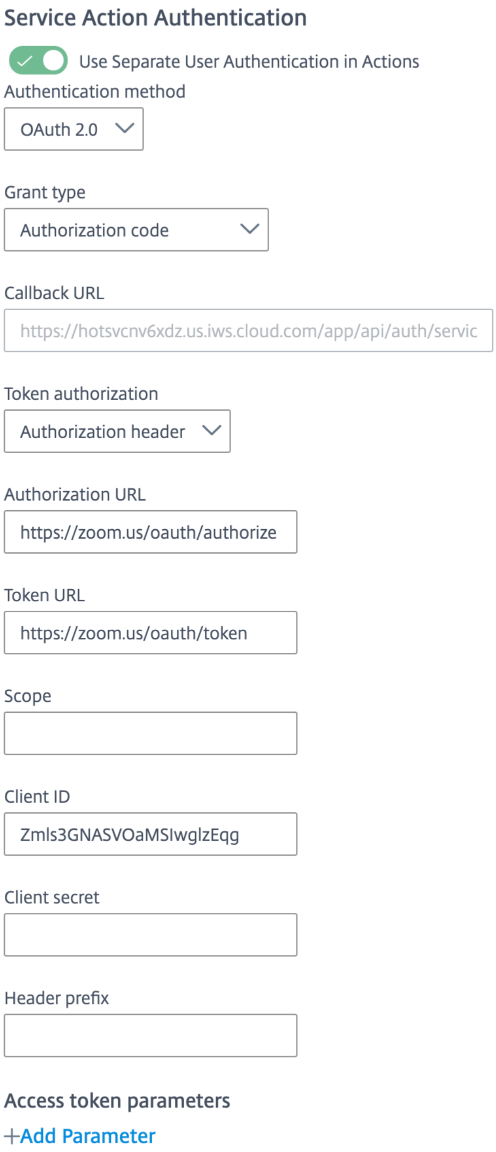
Header prefix:

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:

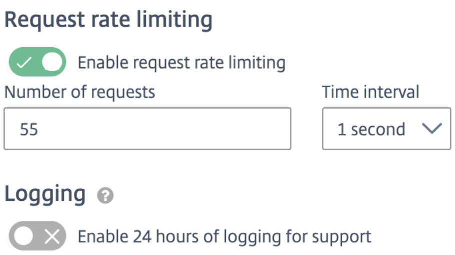
Client ID: Zmls3GNASVOaMSIwglzEqg

Client secret:

Header prefix:

Access token parameters

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

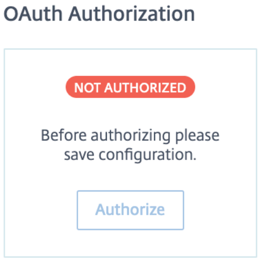
Time interval: 1 second

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

September 22, 2021

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 Virtual Apps and Desktops service connector specifications
- Covid-19 Self Certify connector specifications
- Adobe Sign connector specifications
- Ariba connector specifications
- Blackboard Learn connector specifications
- 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
- Integrate 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
- 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.

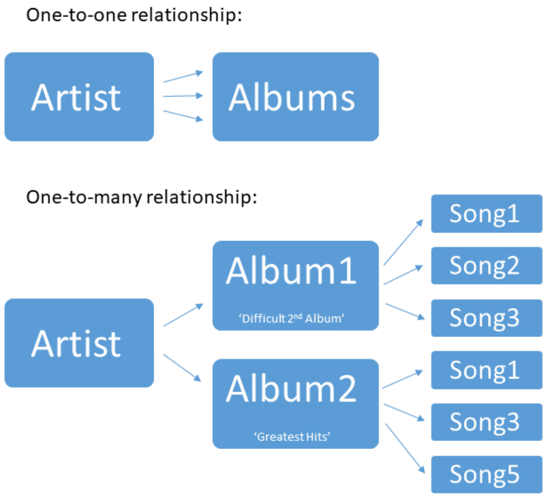
Data Loading and Service actions.

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:

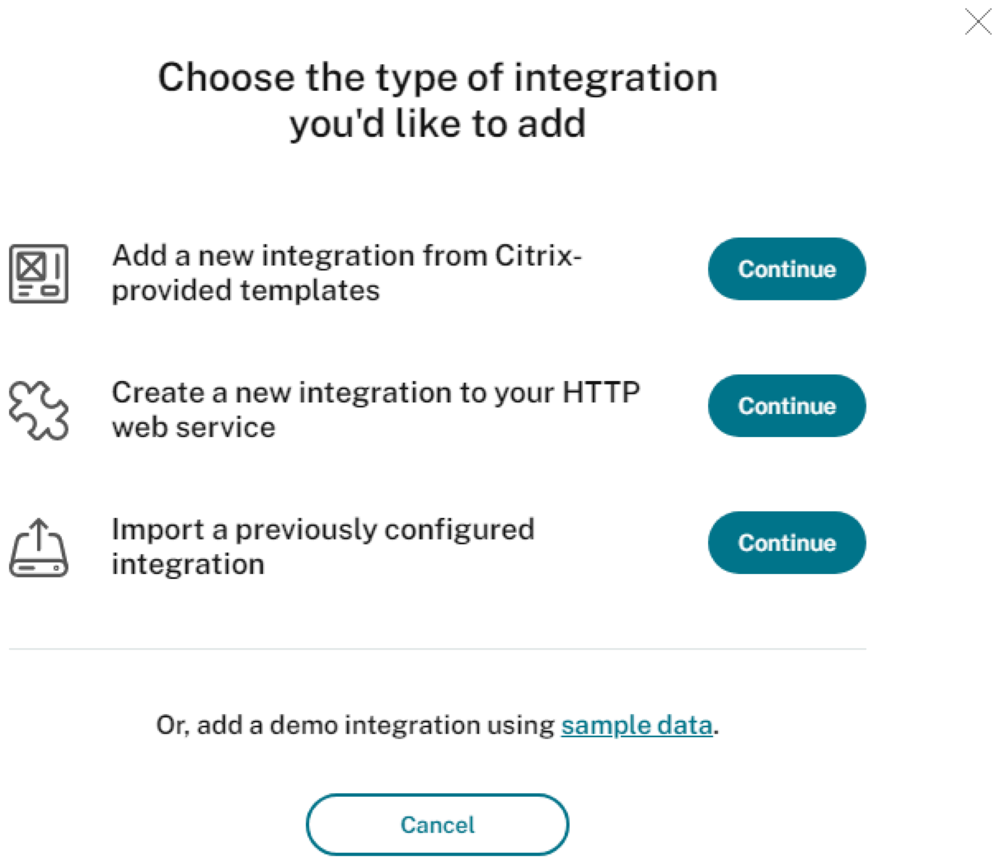
- Create HTTP integration
- Configure the integration

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:

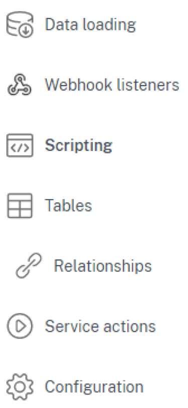
- Ready your script that you want to import via the Microapp administration interface.
- Scripts must be written in the javascript language edited in your preferred text editor / development tool.

- 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.

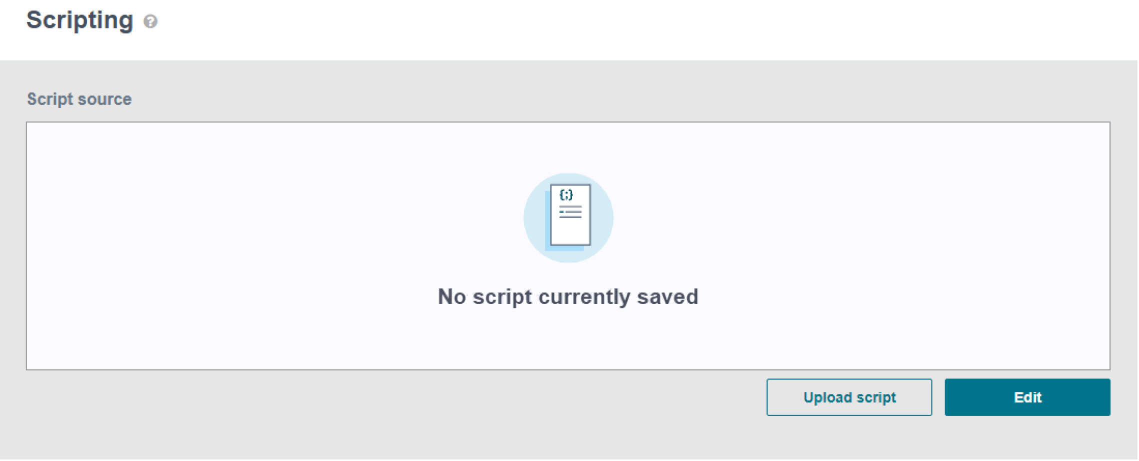
Import a script

To import your prepared script via the integrations tab in the Microapps admin interface.
Follow these steps:

1. Select **Scripting**.

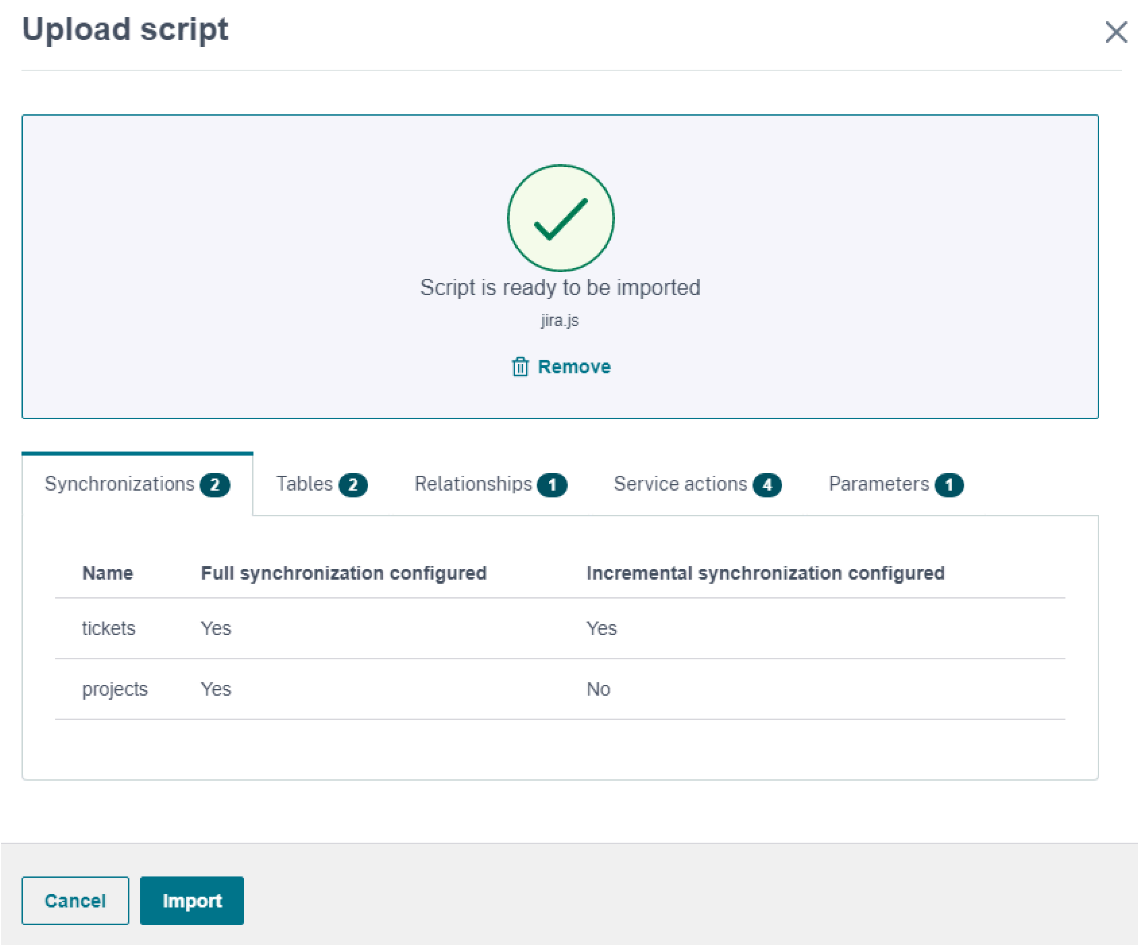


2. Select **Upload script**. Alternatively, you can input your script directly into the text area by selecting **Edit**.



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

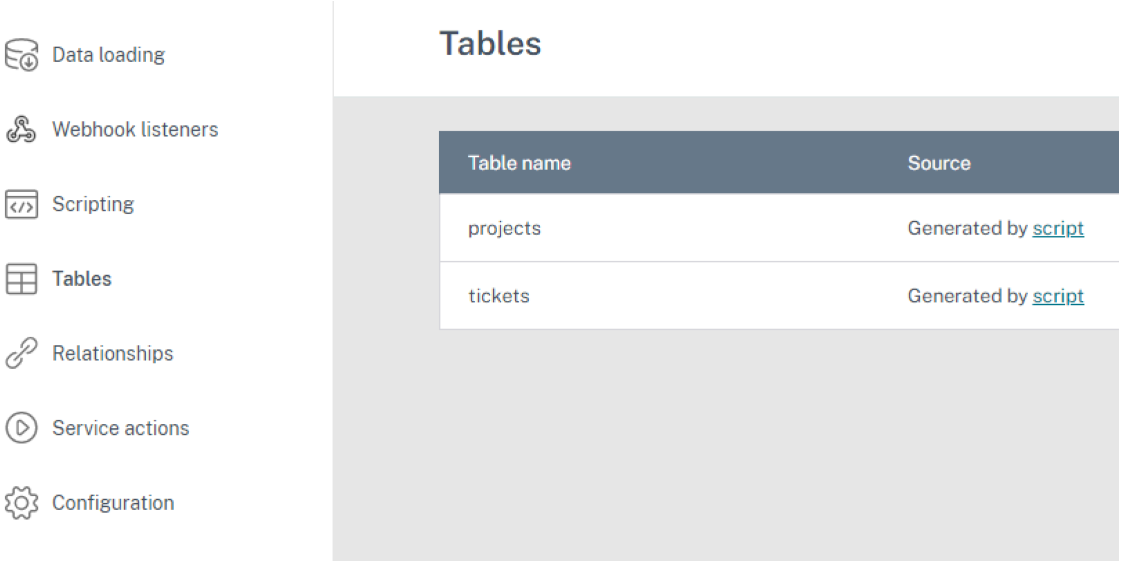
5. Select **import**.

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:

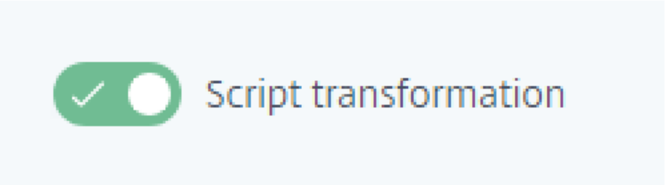
- Ready your script that you want to import via the **Data loading** or **Service actions** configuration pages.
- Scripts must be prepared in the javascript language edited in your preferred text editor / development tool.
- When ready, paste the script (or edit directly) via the script transformation 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.

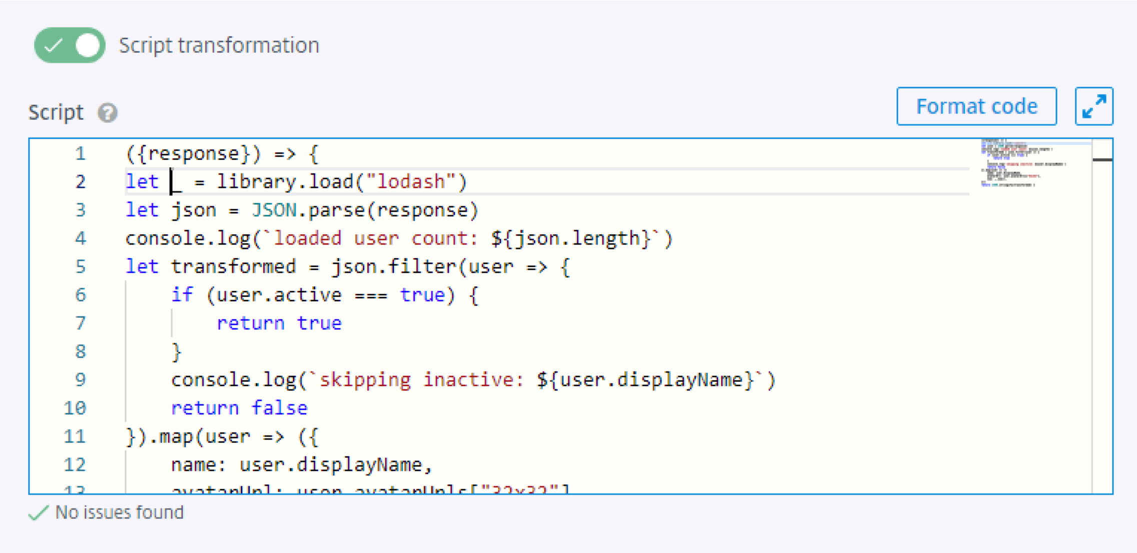
Enable script transformation

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

October 11, 2021

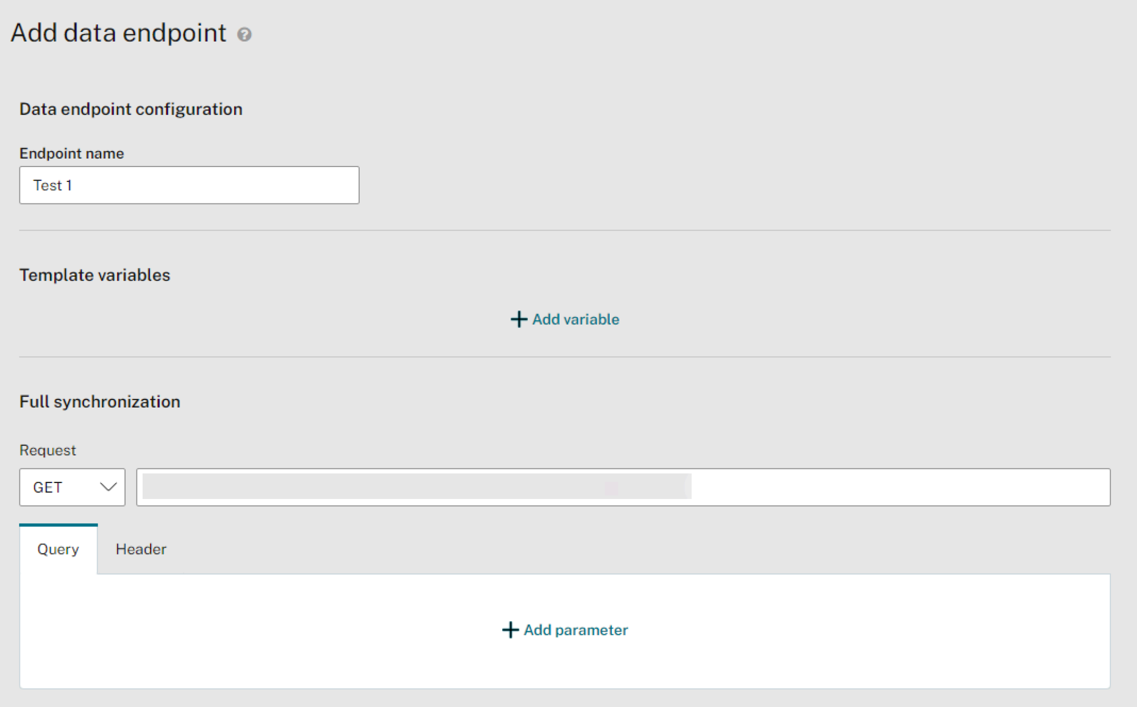
Now that you have added the HTTP integration, configure your integration. This endpoint data along with service action configuration forms the basis for creating actionable microapps.

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

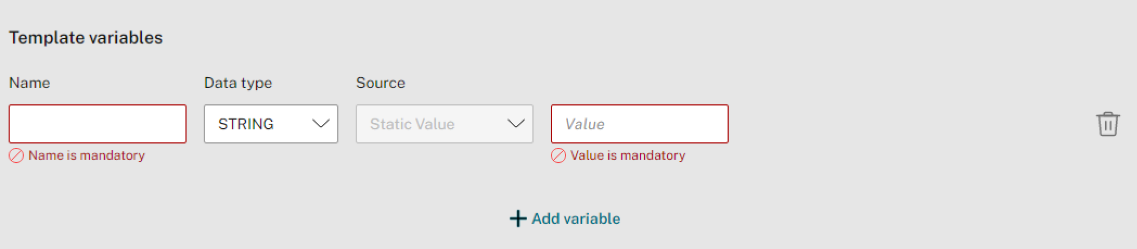
Request: GET [redacted]

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 determines 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.

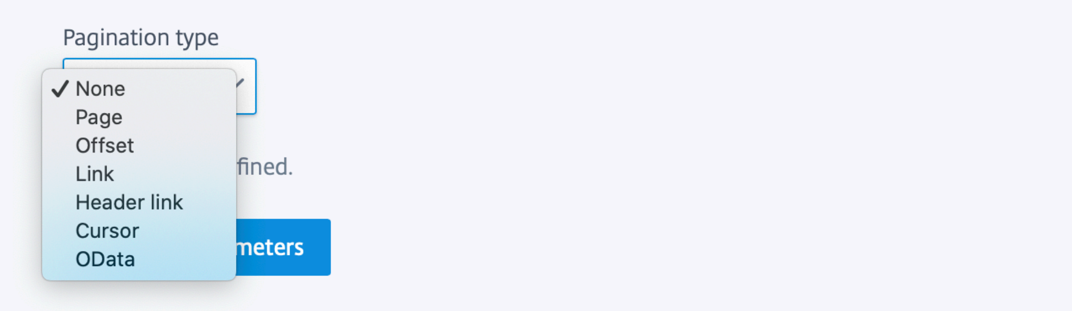
  Example:

  ```
  Link: <https://api.github.com/search/code?q=addClass+user%3
      Amozilla&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 is dependent 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**

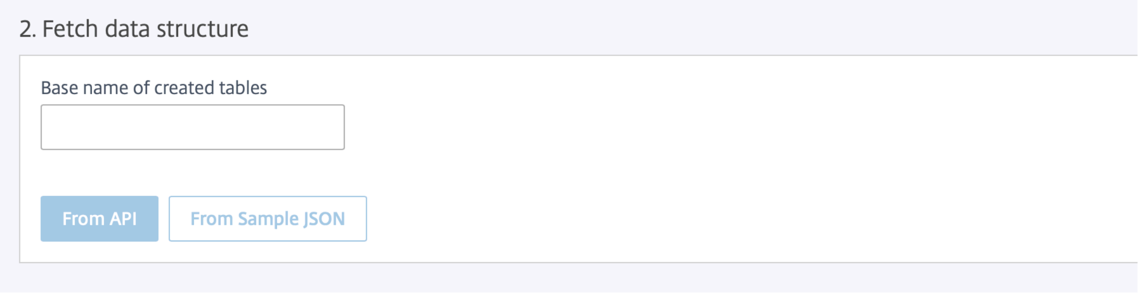
   With **Incremental Synchronization** toggled you can set up Synchronization. That is, to download only updated records since the last data synchronization in more frequent intervals. To do this, configure how often you would like your API call to run. Enter at least one server time parameter. Leaving 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.

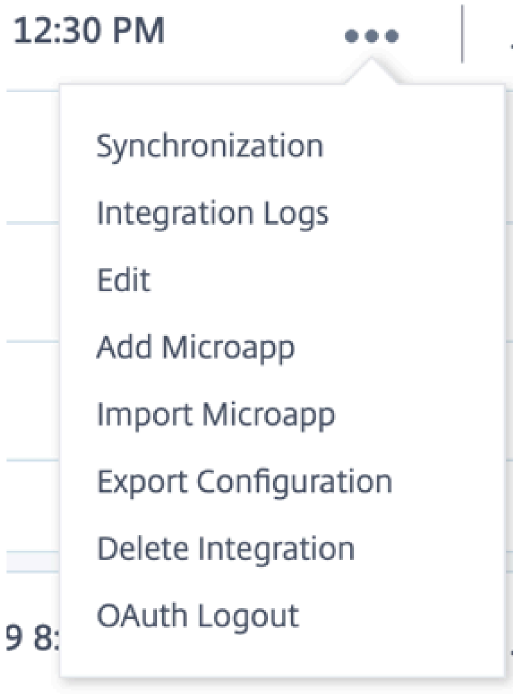
> **Note:**
>
> After setting up and adding your endpoint, your table structure is locked. If you need to restructure you must create and configure a new data loading endpoint.

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.

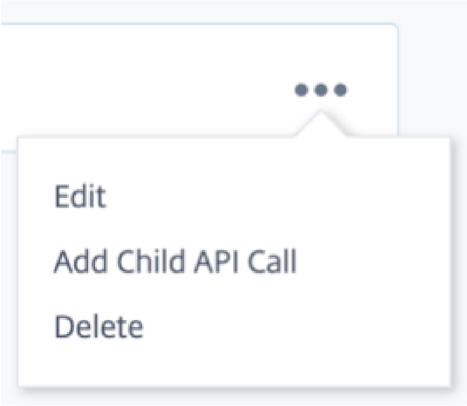
Follow these steps:

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 above. As your application integration can be any number of bespoke combinations we will take a generic approach to explaining how service actions work.

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.

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.

Data update before action execution

(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.

To set up 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.

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.

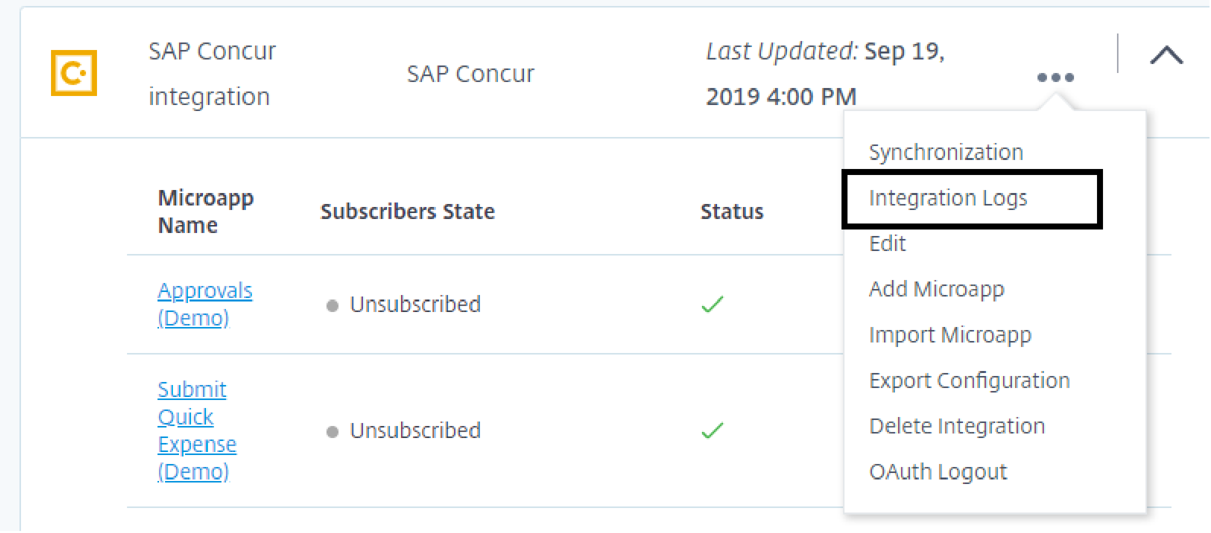
        > **Important:**
        >
        > Not all APIs support incremental synchronization.
        >
        > When you define **daily** or **weekly** synchronization, synchronization occurs randomly within the timeslot you select. For example, selecting 00-04 daily full synchronize will run a full synchronize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

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:

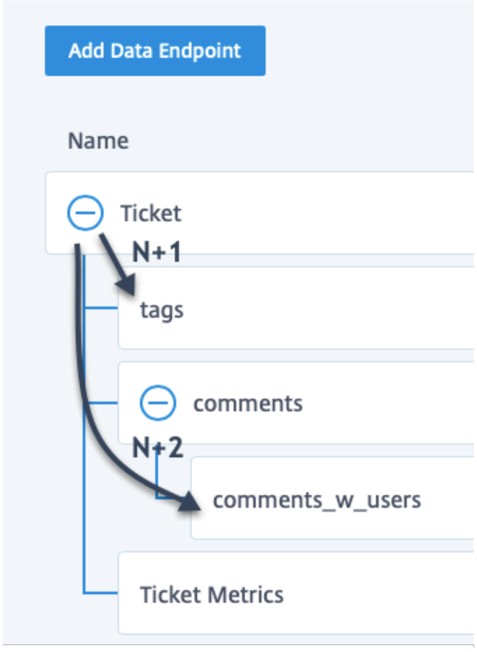
- You have defined your end-to-end use case with the understanding of what must be run in your microapp and what information your end user views and actions.
- You have created the endpoint to return the table data you need from your target System of Record.

> **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`).

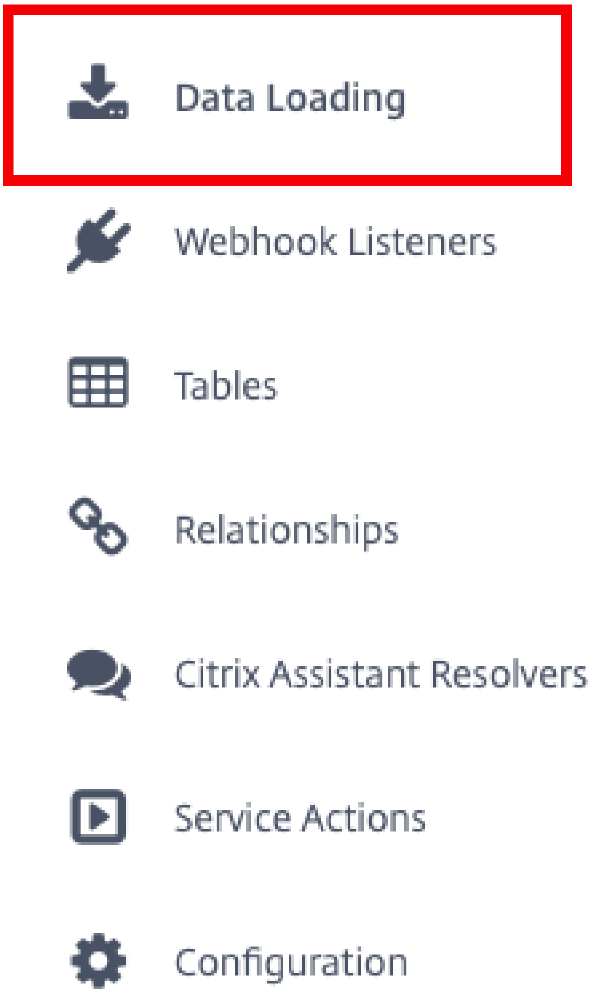
Configuring in this manner results in three levels of data being contained in one large database table. This new table can be used to build the page as per the use case defined when you started to build your microapps. This method can be used for as many levels are required.

API Child Call and Table Merge example

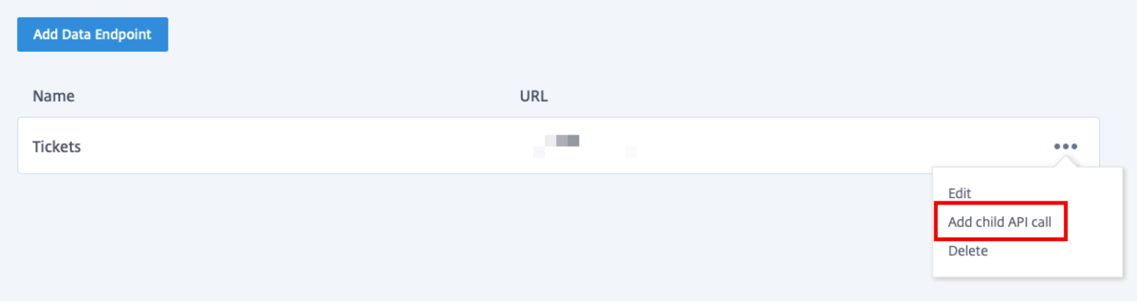
The following example illustrates the general workflow of creating a table structure to reach data beyond an n+1 relationship. Each individual use must be built based on the individual use case you want to build for your microapp. Ensure you are familiar with your target integration System of Record 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:

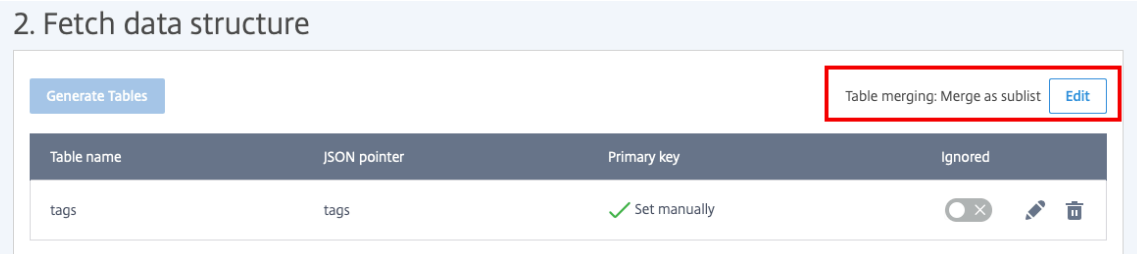


When finished, you can view your data structure on the main Data Loading page.

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

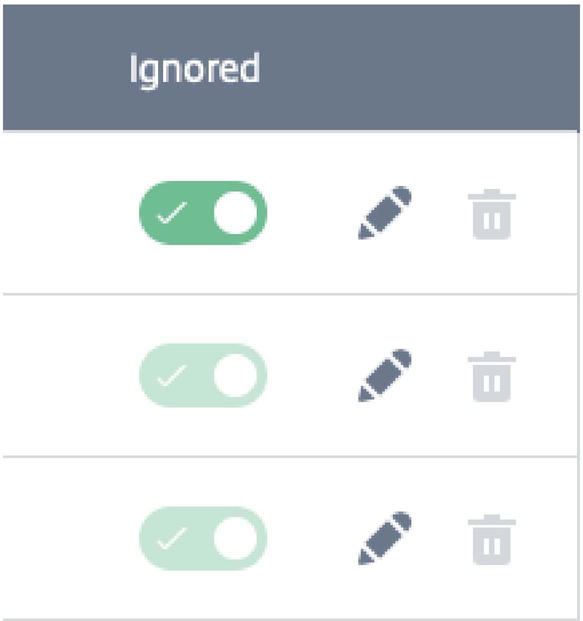
Table name	JSON pointer	Primary key	Ignored
tags	tags	✓ Set manually	○

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

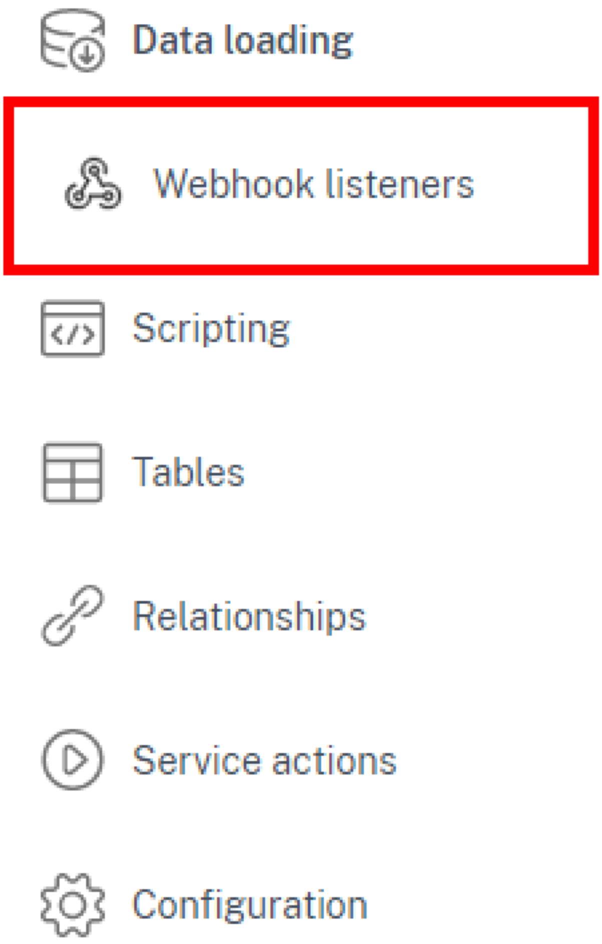
July 29, 2021

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

September 8, 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.

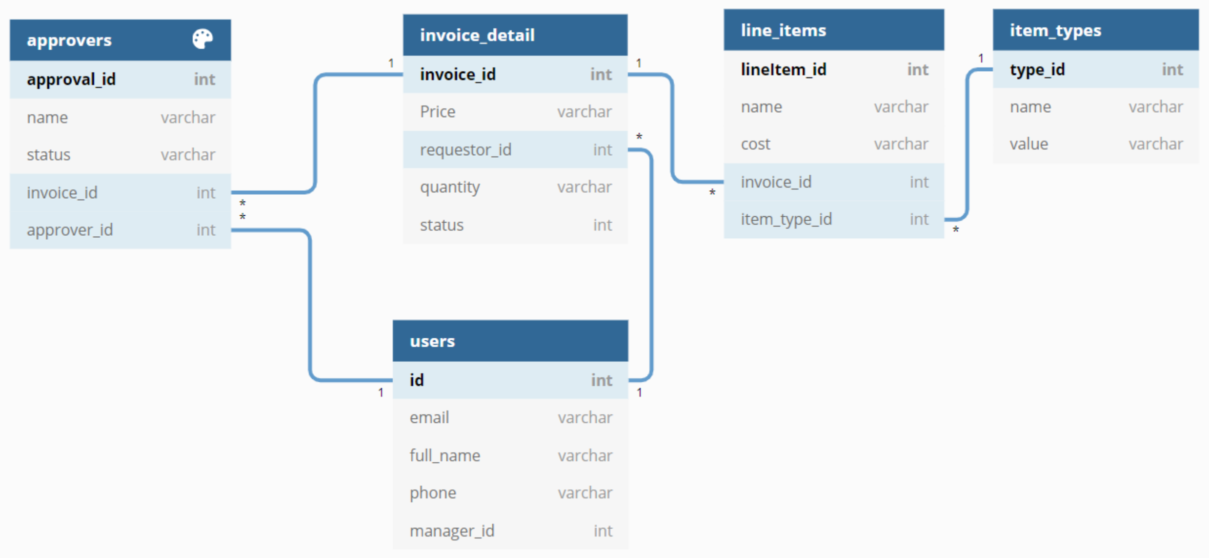
> **Note:**
>
> You can also export and import a new version of the microapp from the same menu. For more information on these capabilities, see Export and import integrations and microapps.

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 page

The Embed page 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.

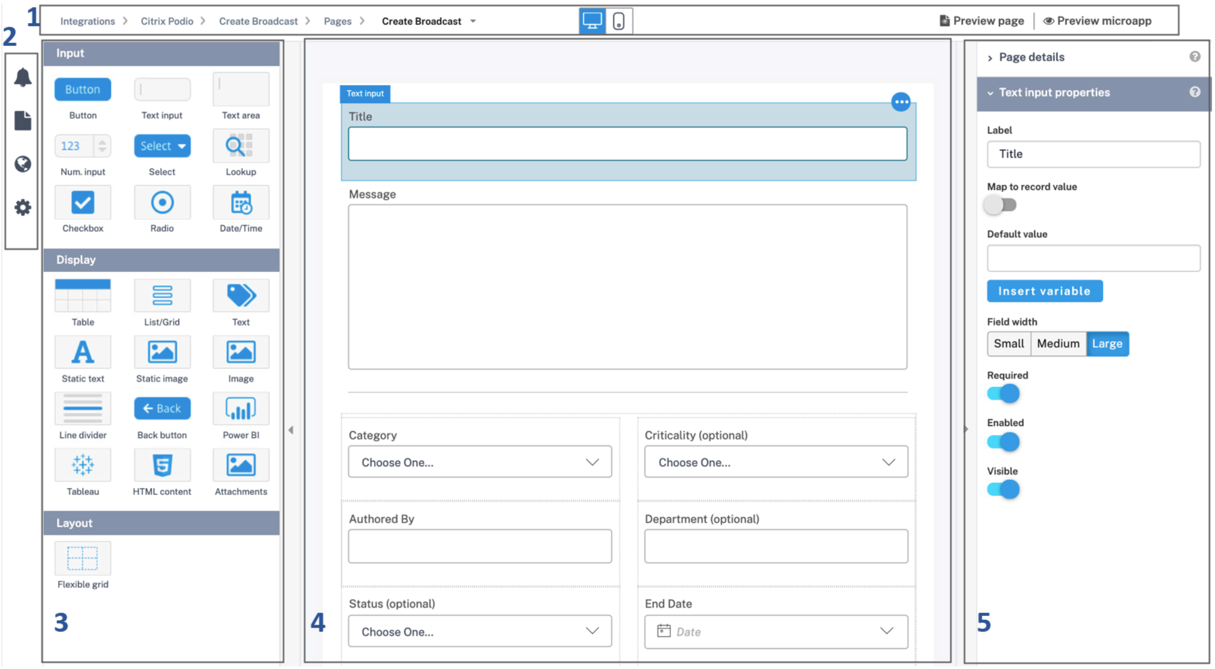
Page builder

October 5, 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 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.

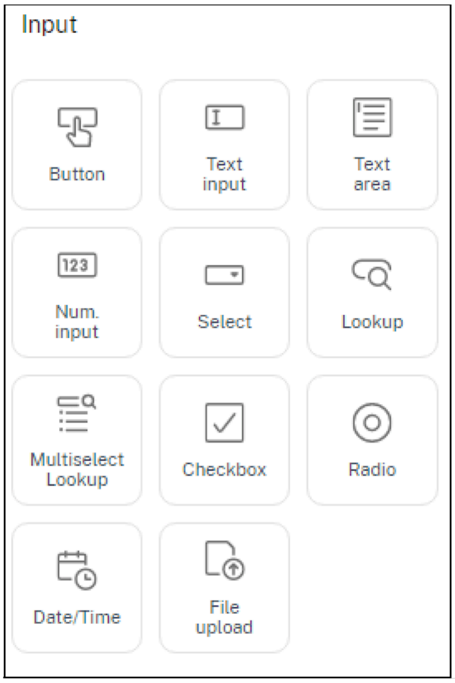
- **Detail** - Page template that provides static details and is connected to a particular record from the cache.
- **Form** - Create a page that provides static details in addition to the ability to input user data into your page.
- **Table** - Create a page listing multiple records based on the data 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.

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 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. 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 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, etc. 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, etc. 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.

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 new 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 required. 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 required. 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.

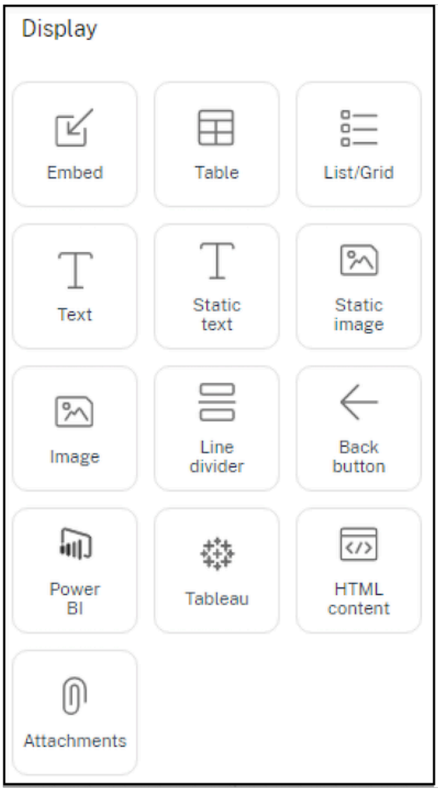
Considerations

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, etc. 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**.

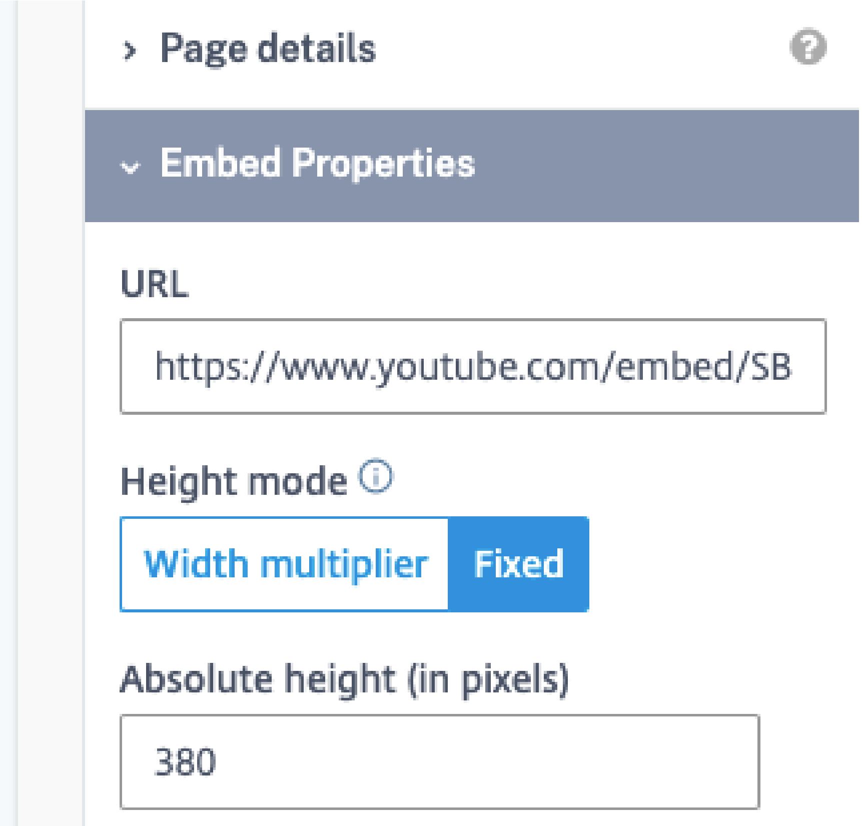
> **Note**
>
> To succesfully embed content from some websites you must use the embed code generated directly from the website (as opposed to a direct URL or share link).
> For example, when generating an embed link on Google Maps via the **Share** option, use the link generated from **Embed a map** 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

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're 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 doesn't 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 additional 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 doesn't 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 don't want end users to have the ability to change.
- **Visible** toggle - Used in case you need additional 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 don't 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:

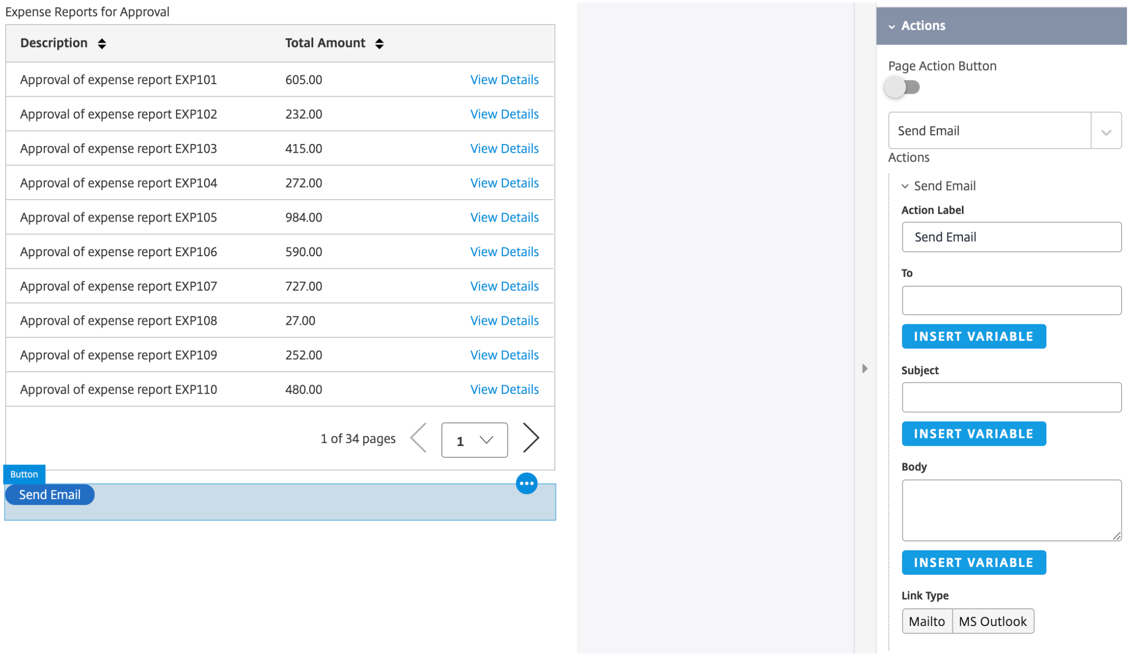
- (Send Email) Workspace users can enter email recipients in a Workspace field for a given action.
- (Go To URL) User's input from a Workspace form can be used as a part of a URL template opened in Workspace.

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

Page Action Button: off

Send Email

Action Label: Send Email

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 need to 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 will be 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

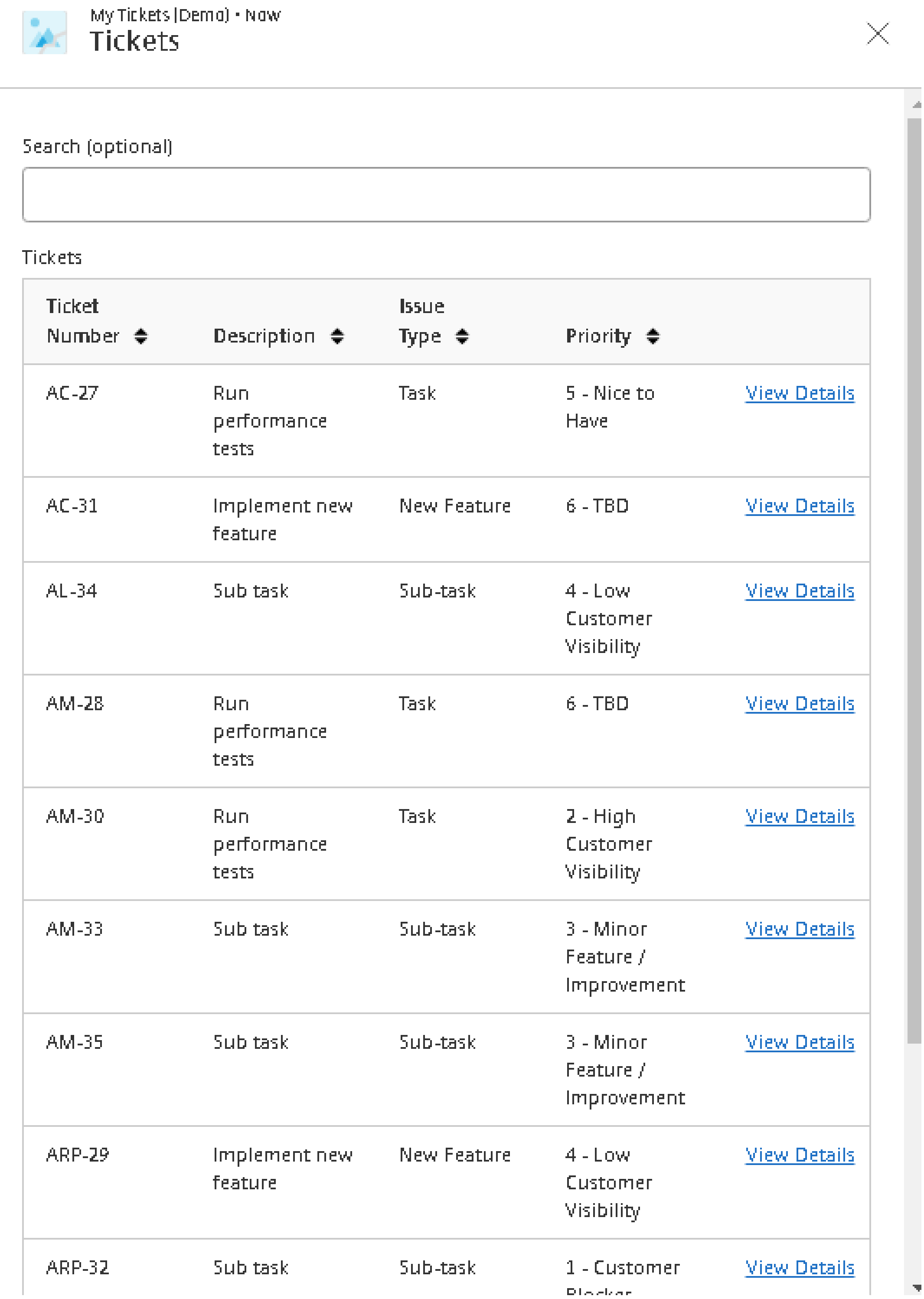
April 28, 2021

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

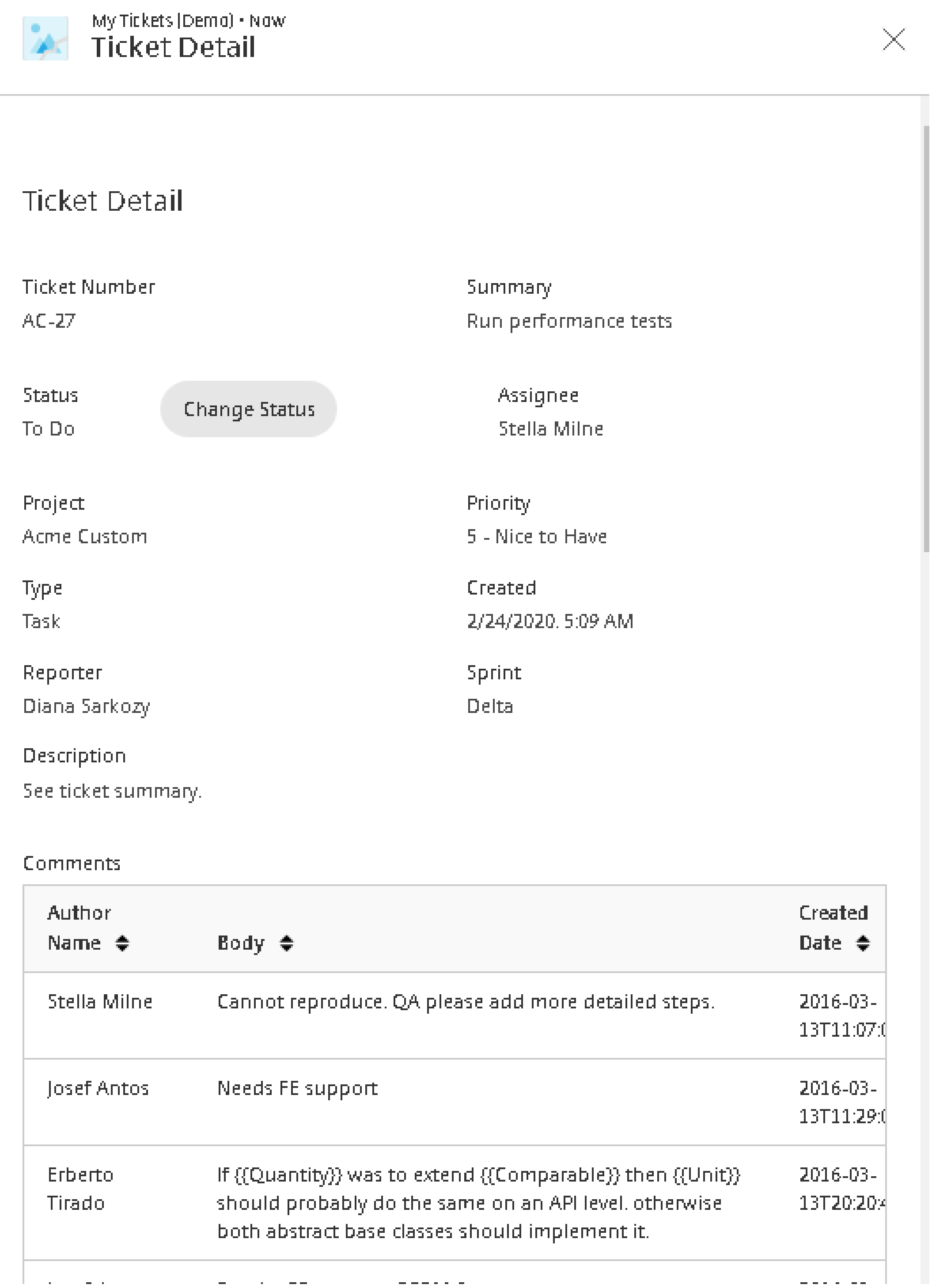
April 28, 2021

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

Change Status

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:

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.

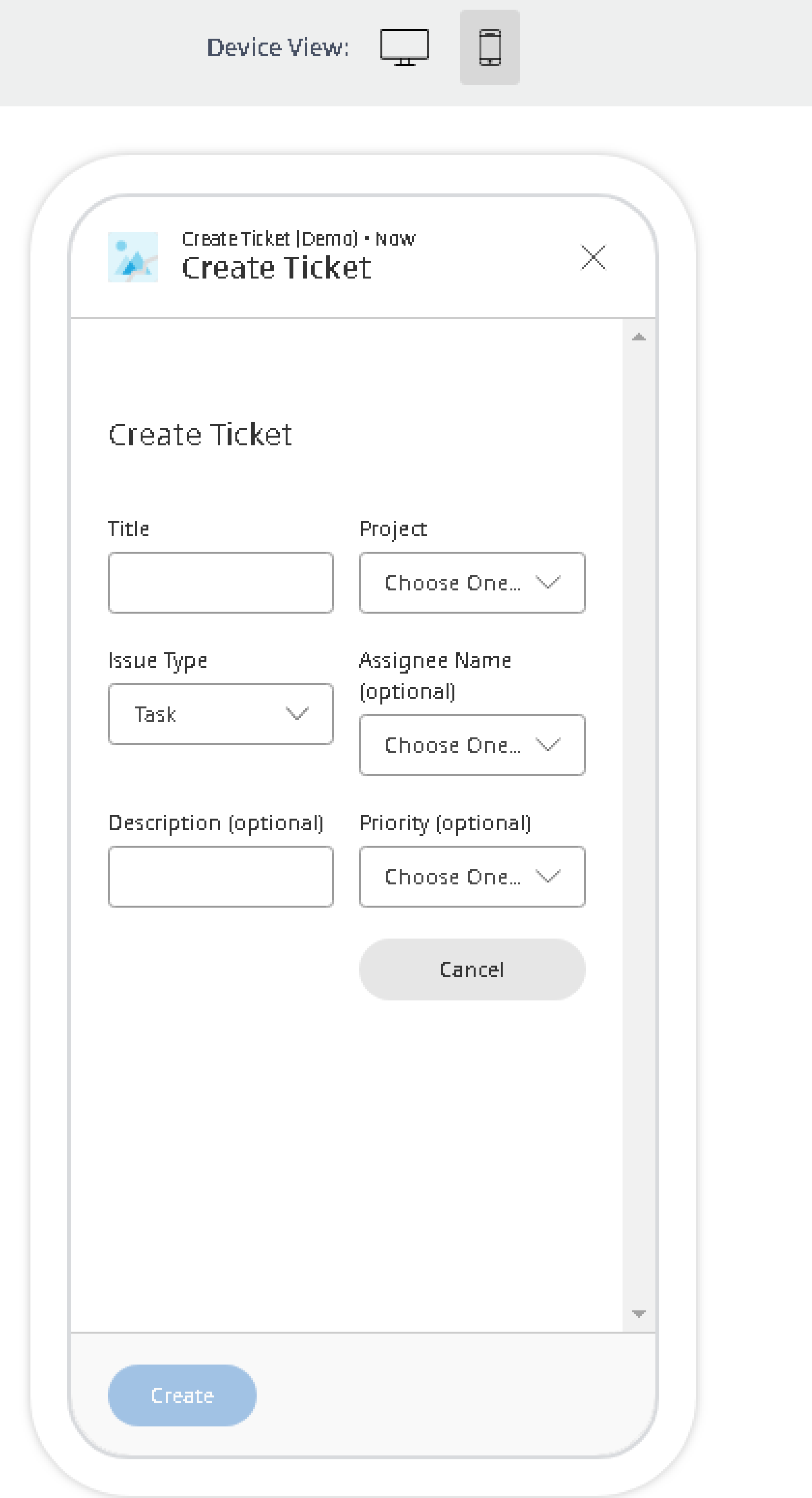
> **Note:**
>
> Create functionality is limited based on API write-back access.

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



Device View:

Create Ticket (Demo) • Now

Create Ticket

Create Ticket

Title

Project: Choose One…

Issue Type: Task

Assignee Name (optional): Choose One…

Description (optional)

Priority (optional): Choose One…

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**.

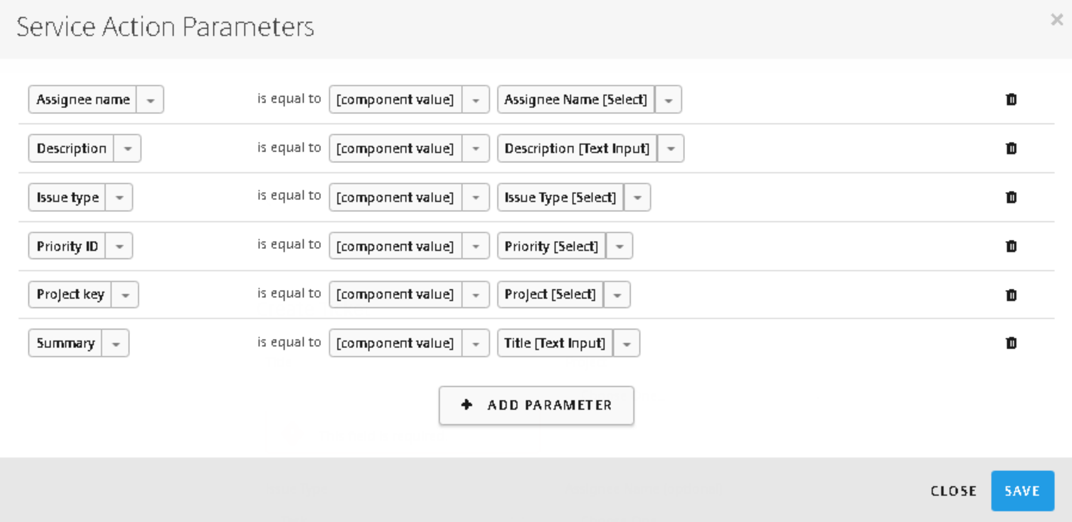
    - Select the **Text Input Properties** tab. In the **Label** field, enter *Description*.
    - Disable the **Map to Record Value** toggle.
    - For this field, do not activate the **Required** toggle to automatically add an **(optional)** tag to the field.

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:



Service Action Parameters

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

June 9, 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.

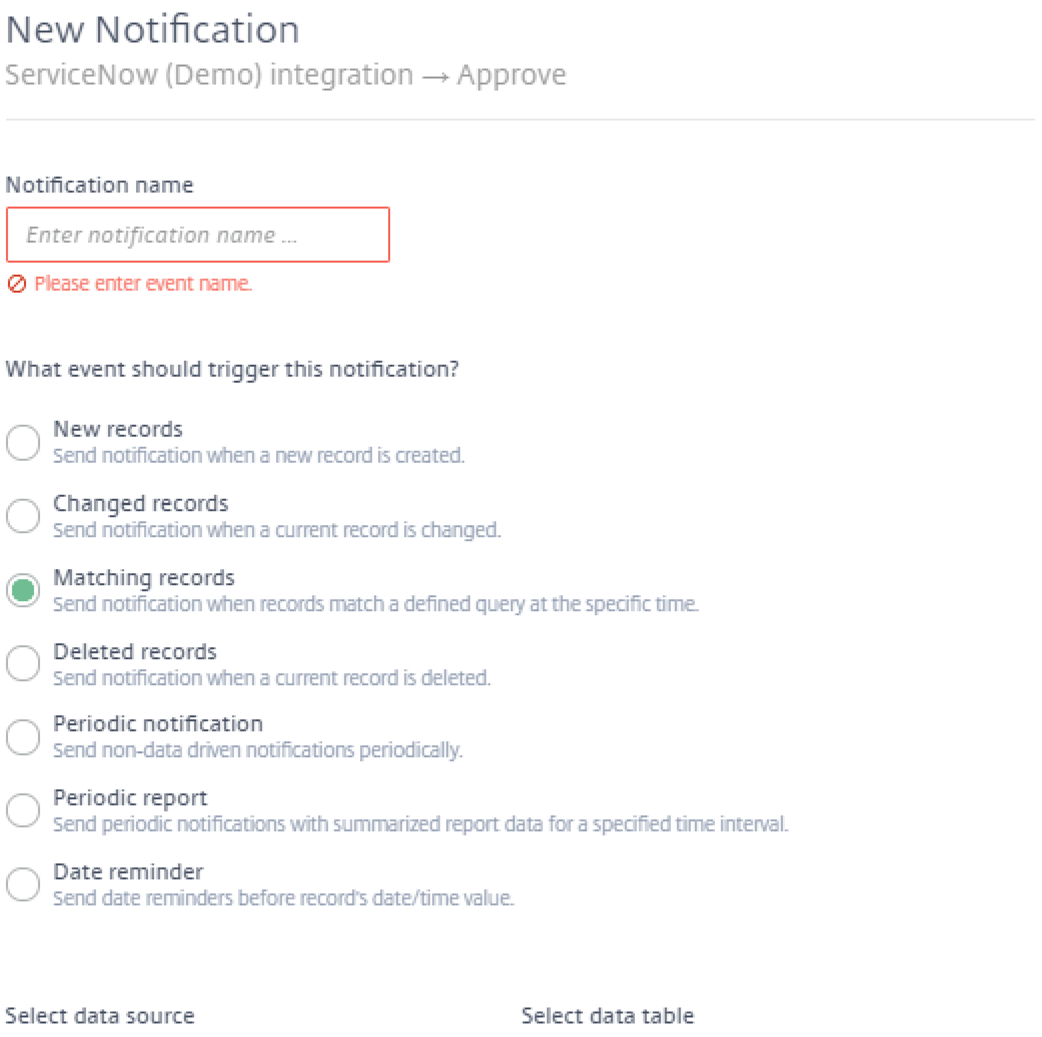
Create a notification

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

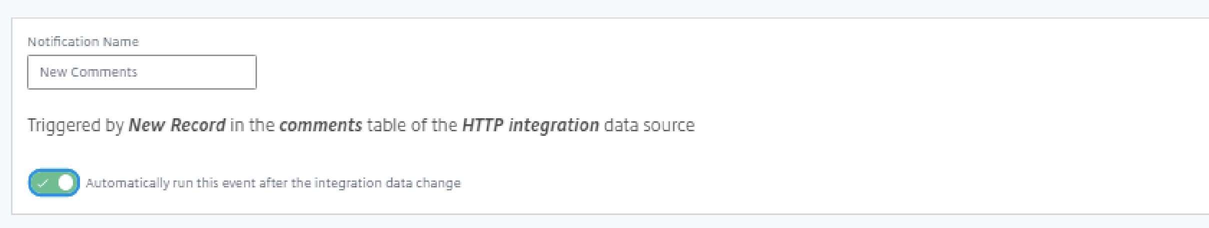
4. Confirm your **Data source** and select the **Data table** from which you want to track changes. Select **Add**. The new notification is added to the Notifications list and is ready to be configured.

   > **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

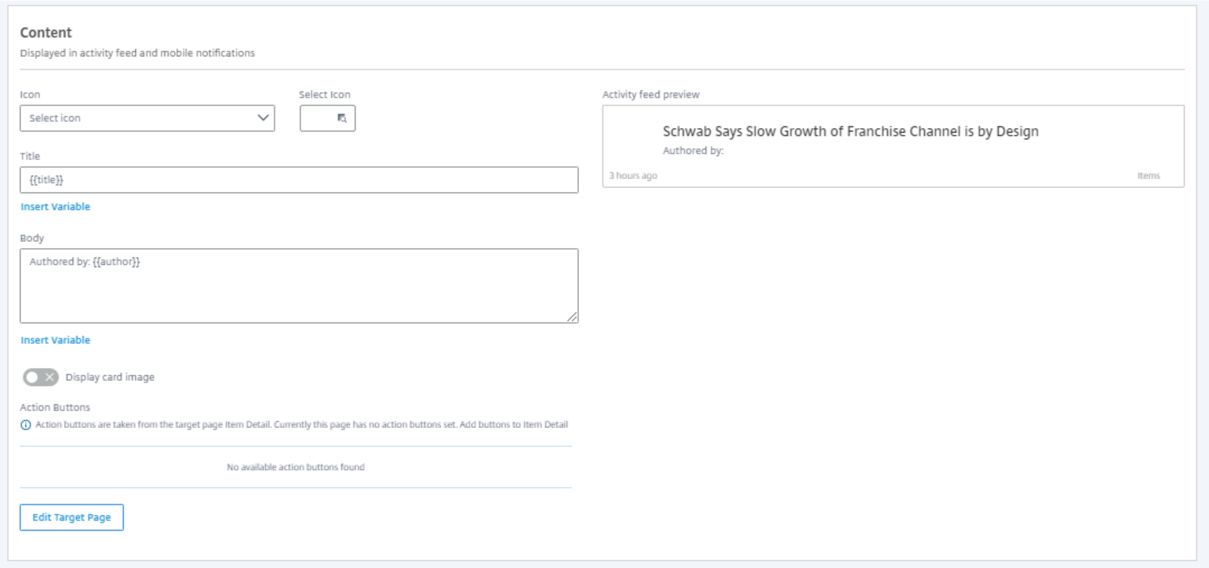
New Comments

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

Select Icon

Title

{{title}}

Body

Authored by: {{author}}

☐ Display card image

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

Activity feed preview

Schwab Says Slow Growth of Franchise Channel is by Design

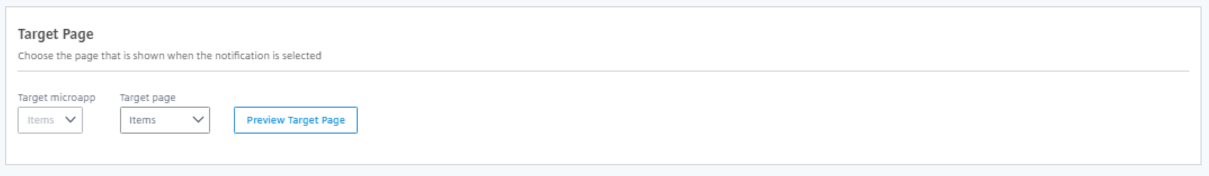
Authored by:

3 hours ago

Items

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

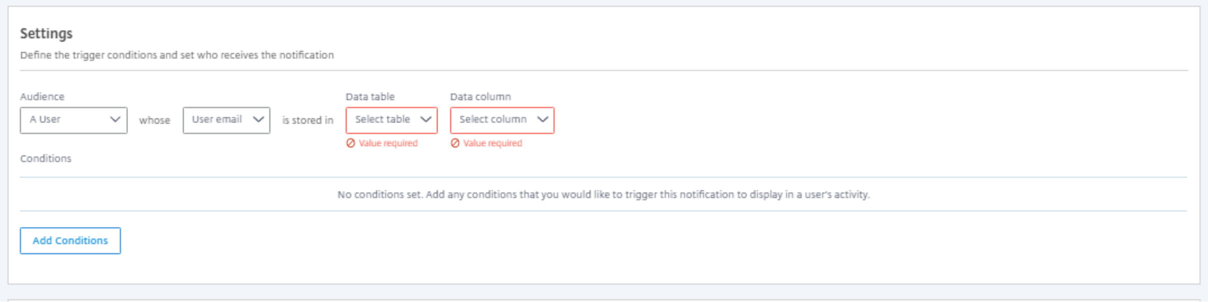
Items

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)** will result in displaying **1 AND 2 AND 3 OR 4 AND 5** as the redundant brackets are removed from the definition.

Increase notification threshold

To better sustain performance, Microapps service limits the maximum number of notification cards that are generated per user per notification job. By default, this is 50 and any notification cards generated above this amount are lost.

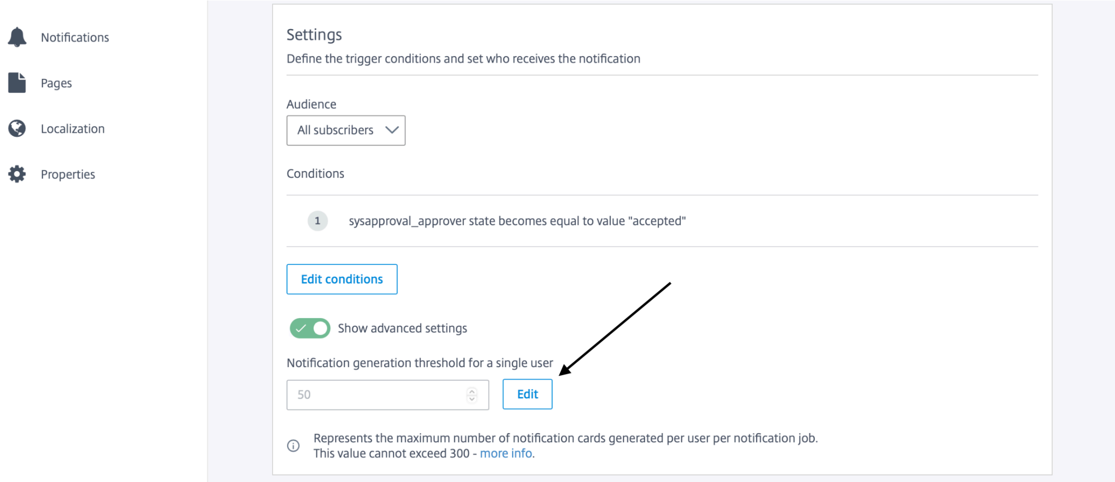
Using advanced settings options, you can increase this value. However, a large number of notifications can flood Workspace users' Activity Feed. Doing this can dilute the value of generating the notification cards and increases the chance that they are not read at all. Consider your users and their Workspace 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 need to 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

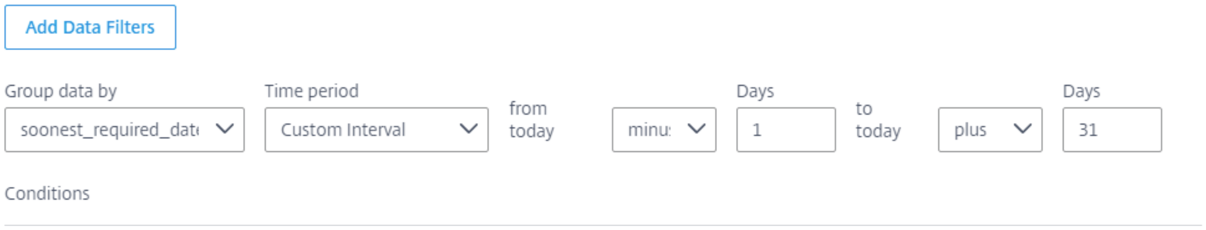
Represents the maximum number of notification cards generated per user per notification job.
This value cannot exceed 300 - more info.

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 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_date

Time period: Custom Interval

from today: minus, Days: 1

to today: plus, Days: 31

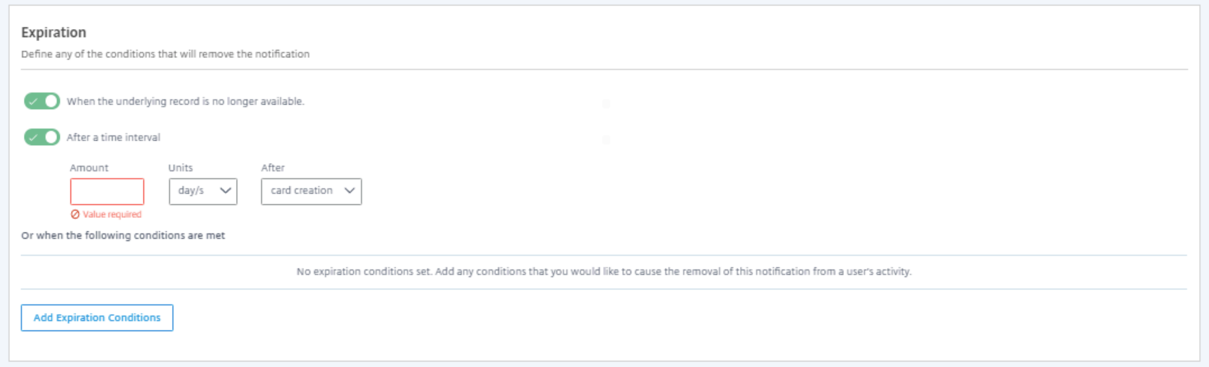
Conditions

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: (empty) — Value required

Units: day/s

After: 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.

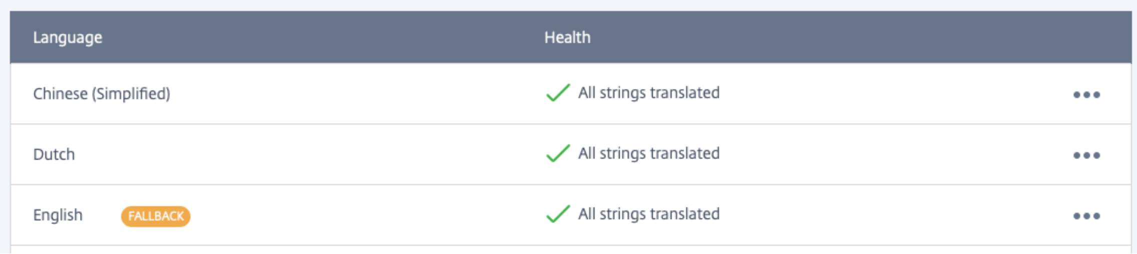
Export files for translation

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

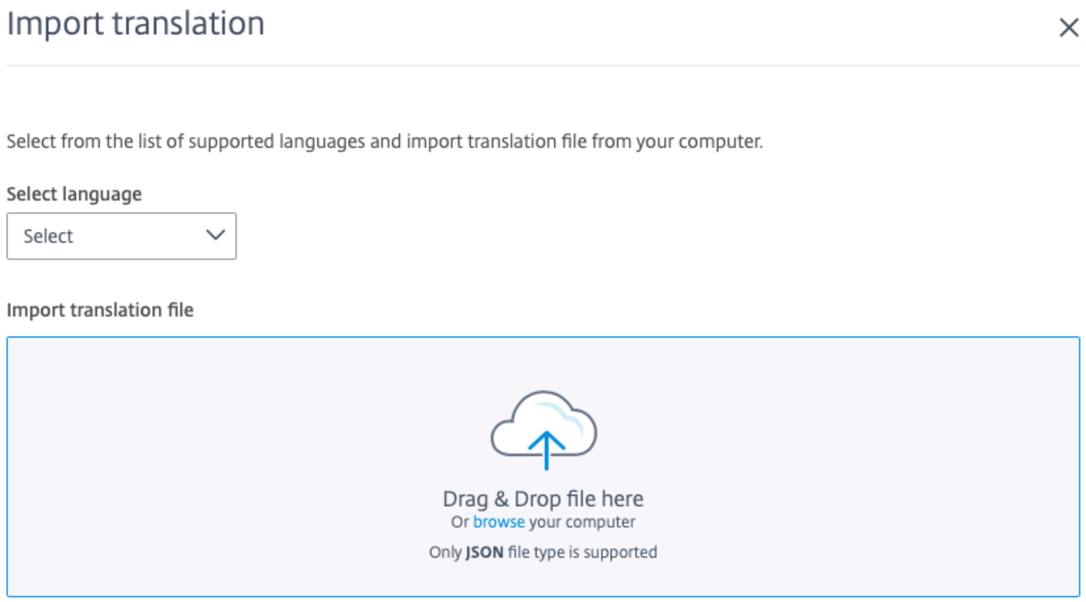
When you have prepared your localized JSON files, import them back into the microapp platform.

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

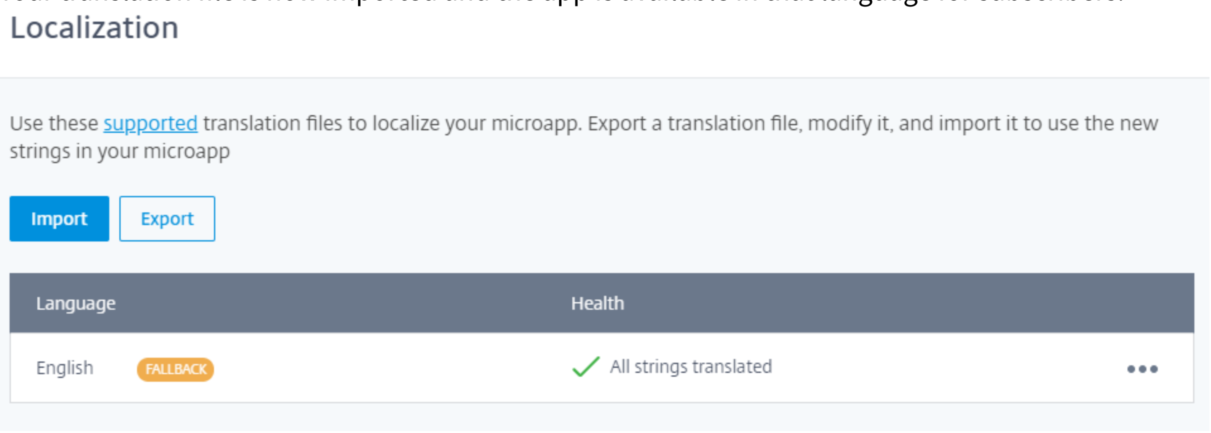
Select from the list of supported languages and import translation file from your computer.

Select language: Select

Import translation file

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.

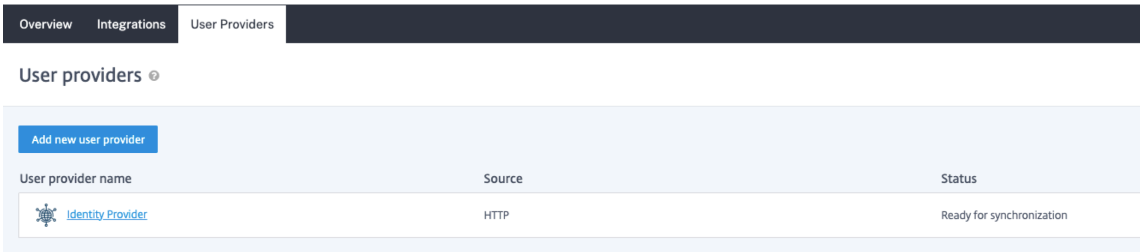
Microapps admins can configure user providers to collect user and user group data from your SoR and use this data to manage microapp subscriptions in all integrations. To configure user providers, your application SoR must provide APIs that expose users/user groups, for example ServiceNow, Salesforce, 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.

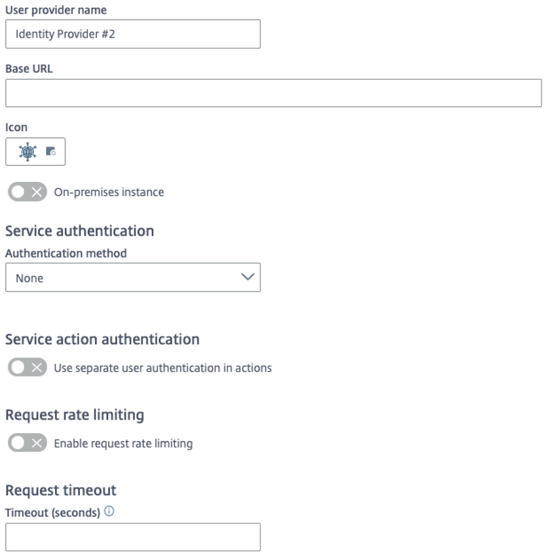
Follow these steps:

1. From the **Microapp Integrations** page, select the **User Providers** tab in the top bar.



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.

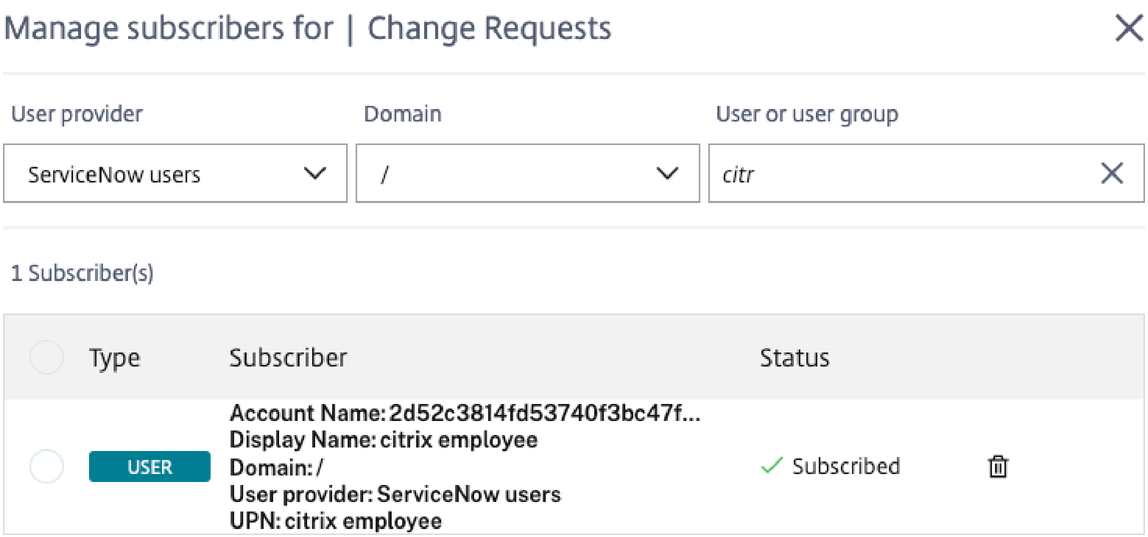
After you added the user provider (in the previous procedure), the page opens with a view of your user providers. Follow these steps to add the script.

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	Domain	User or user group
ServiceNow users	/	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

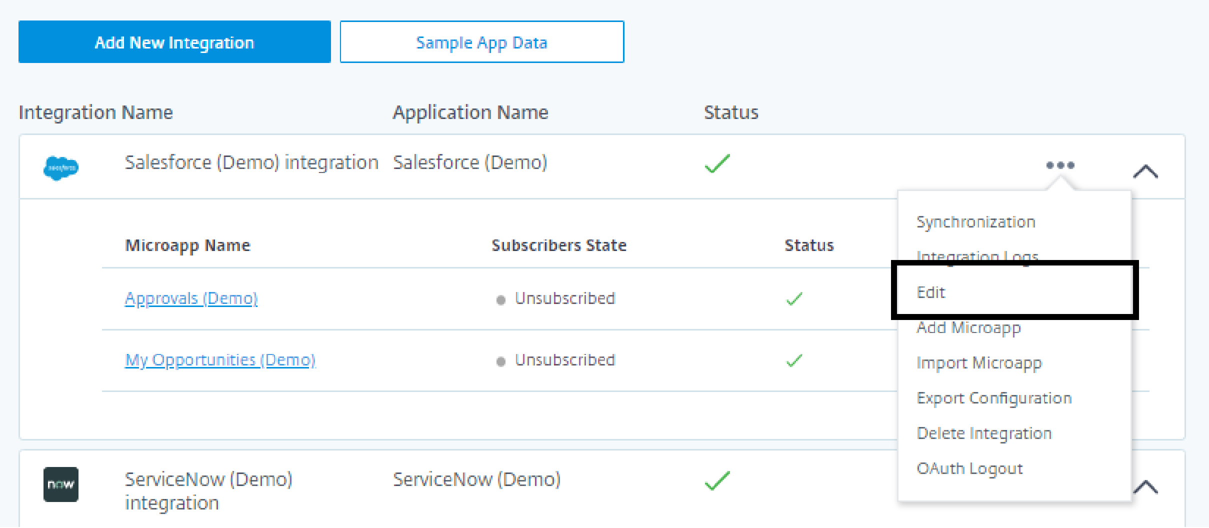
You can add a new microapp to suit your business needs. The template integration comes with a robust database connection. Ensure the required table entities are already available, then use the Microapps builder to create a microapp from scratch. Add new pages and notifications and then populate them.

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**.

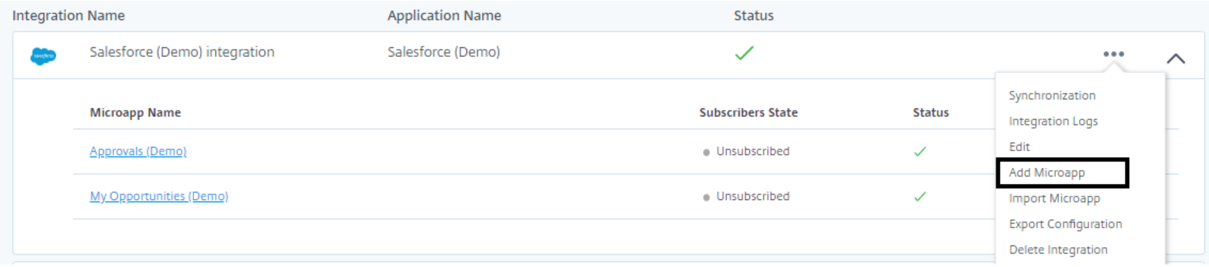
   The **Tables** page opens with an overview of how the database is divided into database tables.
3. To find your required table, select **Edit Schema**, and filter for the required entity and confirm that it exists.

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.

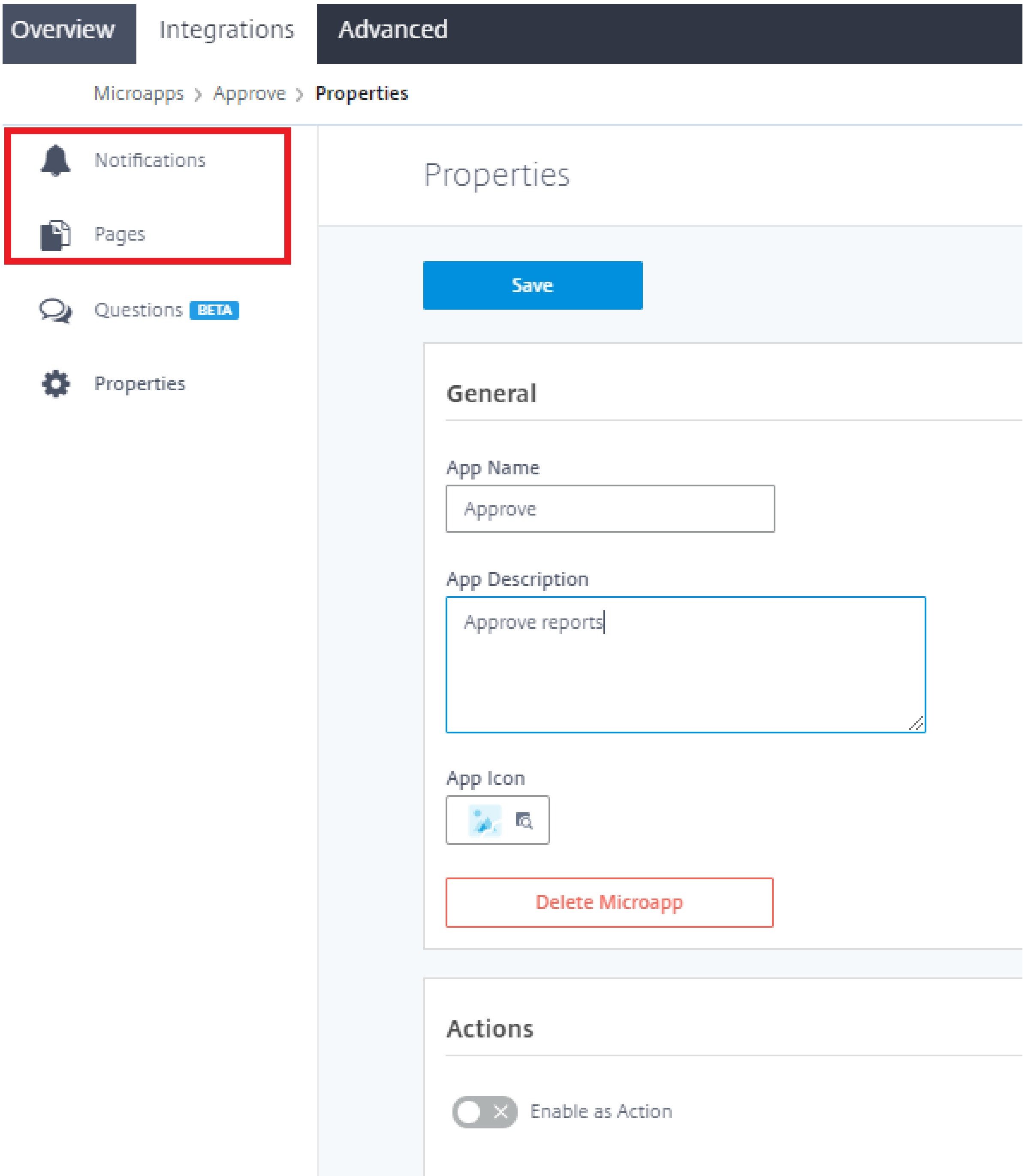
   After you add the blank microapp, it appears under the related integration on the **Integrations** page.
4. Return to the **Integrations** page and select **Blank Microapp** from the list under the integration.

   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.



App Name: Approve

App Description: Approve reports

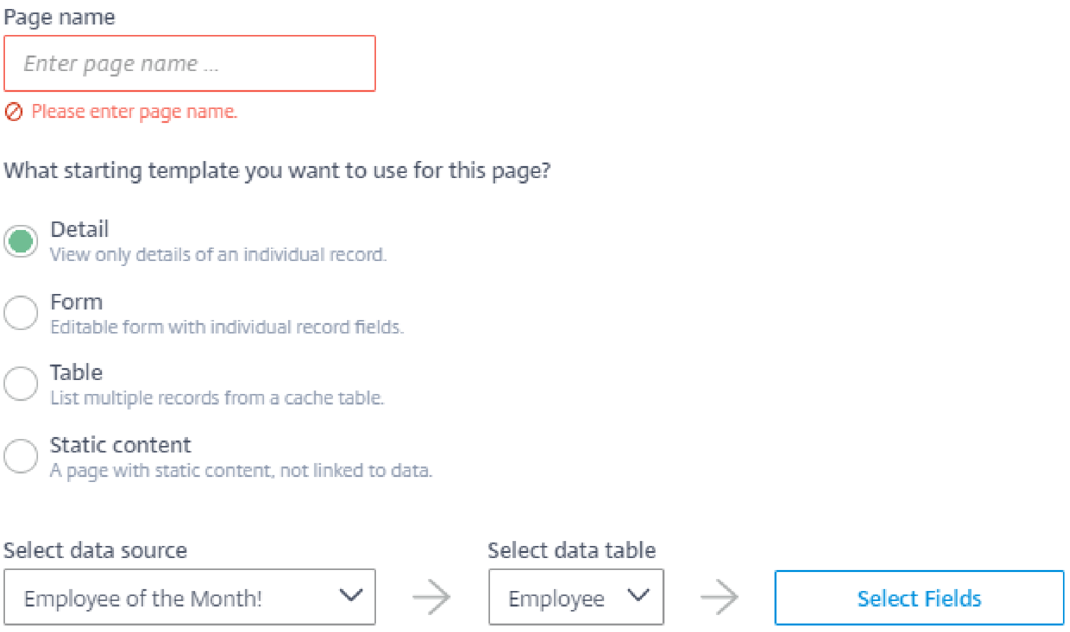
Add Action Pages

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

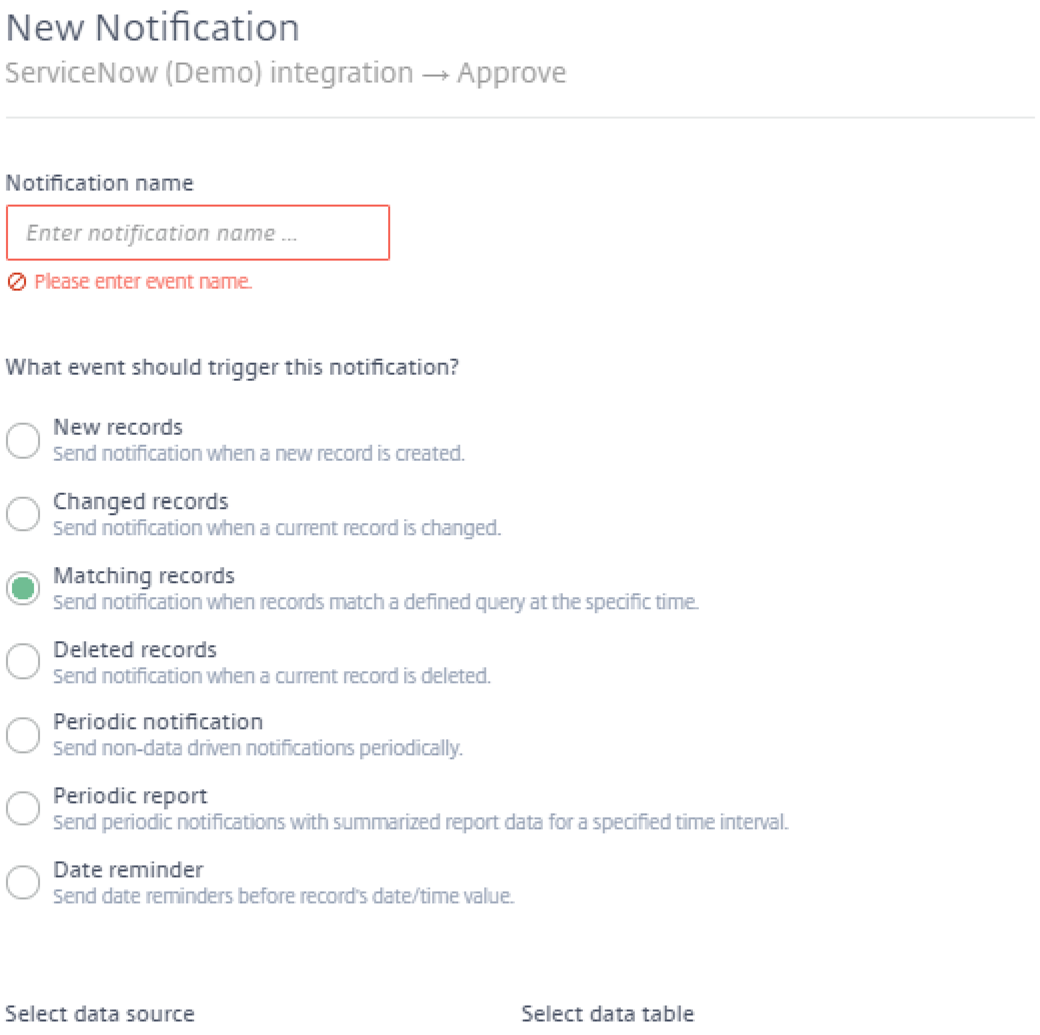
Create triggers for events to be sent to the client application, such as new PTO request or notification that a record changed. Select from the template types, then customize the event in the builder.

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.

Customize an existing microapp for a template integration

April 28, 2021

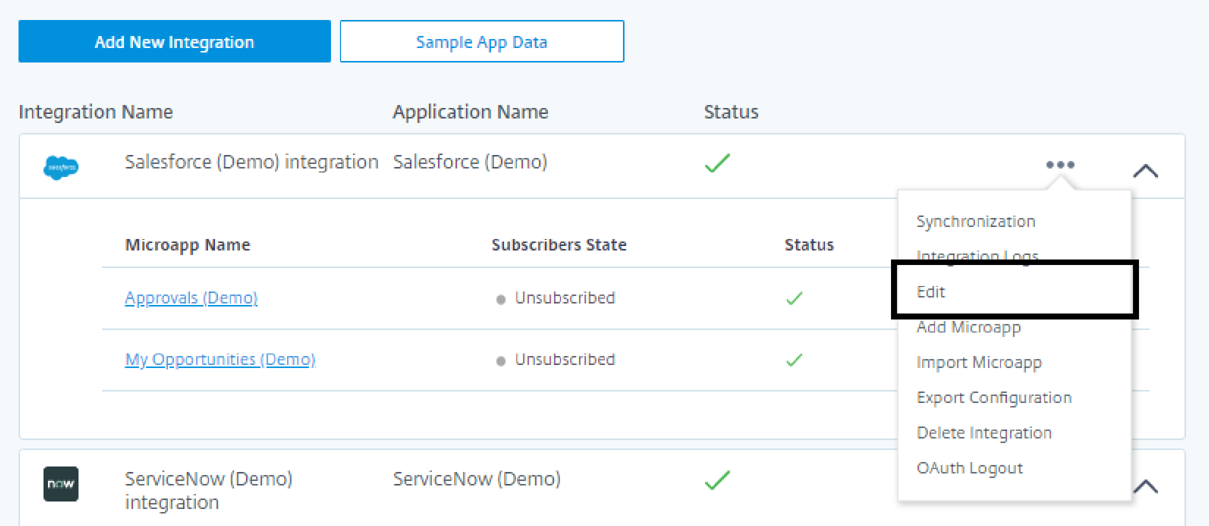
As with creating a new microapp, you can add new pages and notifications. In this case, you can also edit existing notifications and pages using the Microapps builder.

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**.

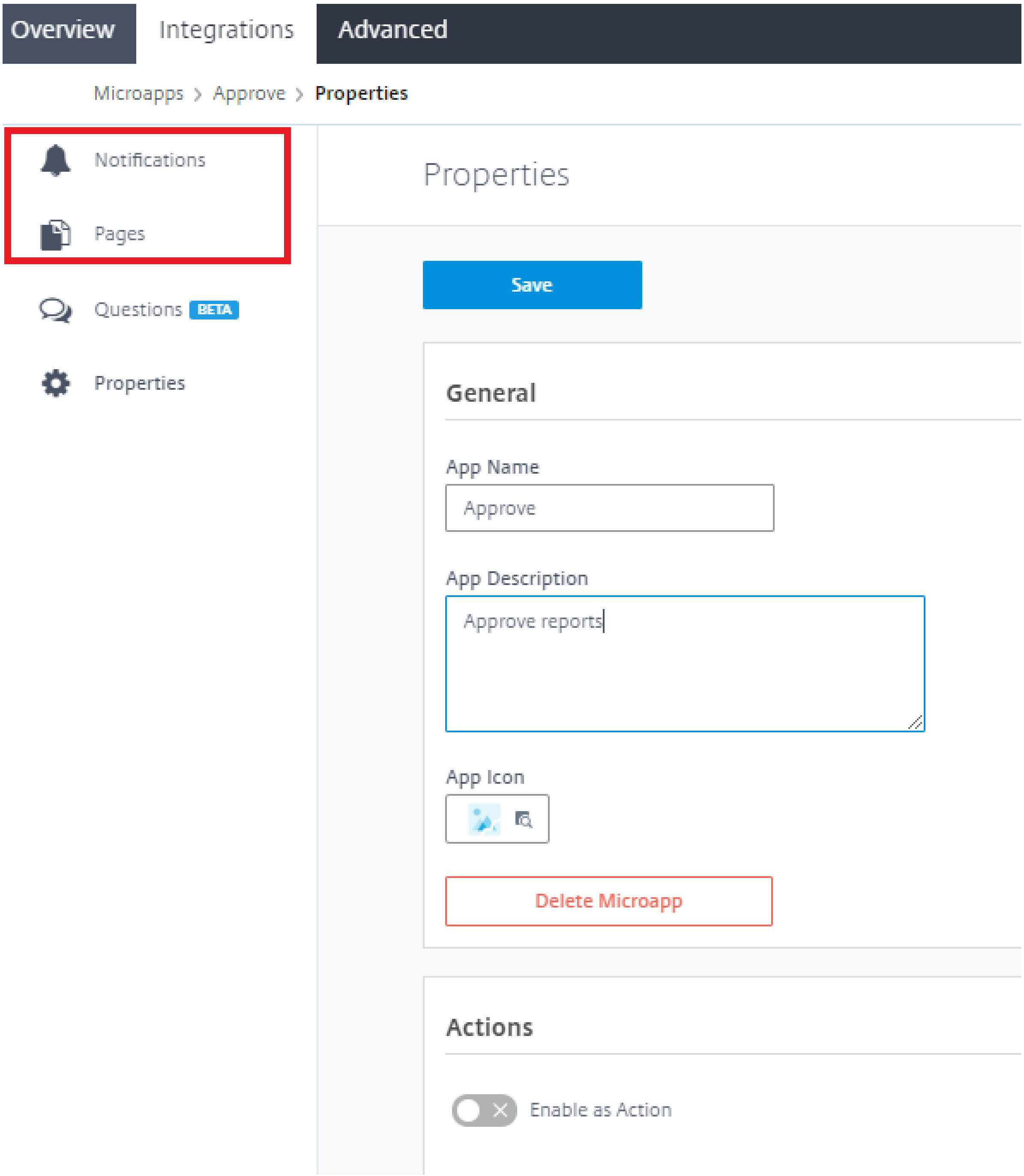
   The **Tables** page opens with an overview of how the database is divided into database tables.

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

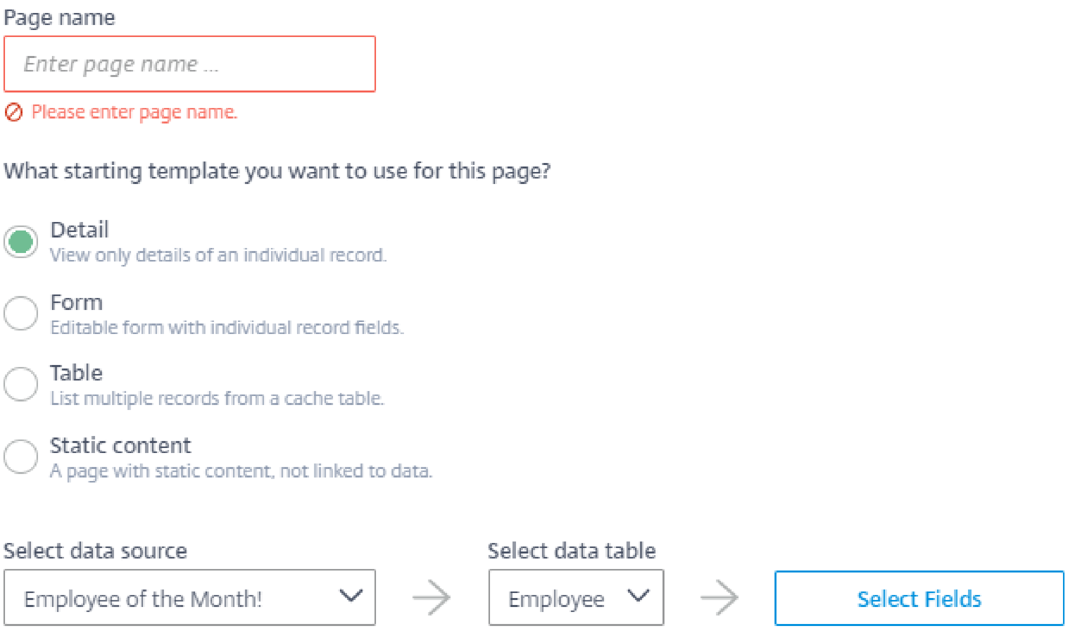
Add Action Pages

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.

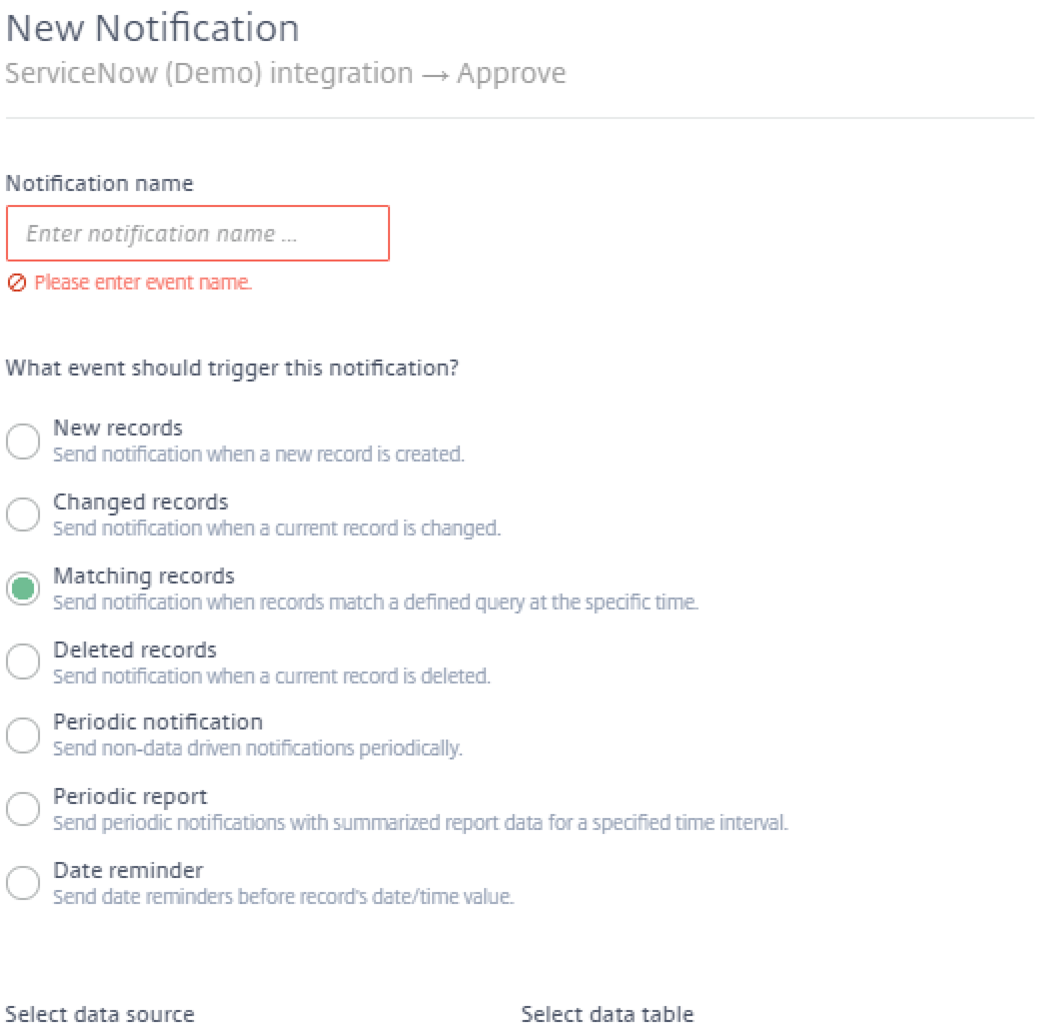
Add Event Notifications

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

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

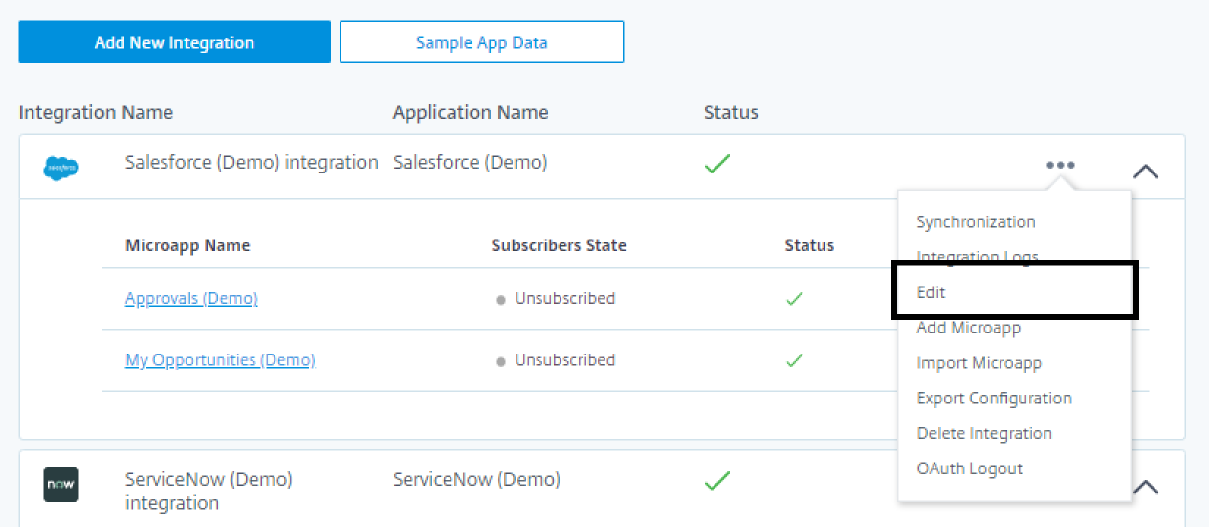
In this case, you have to manually add database connections for custom integrations, then add a microapp and use the Microapps builder to create a microapp from scratch.

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.



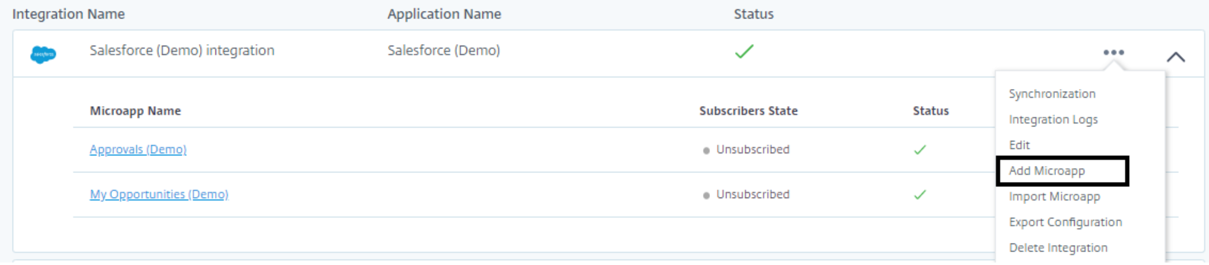
Integration Name	Application Name	Status
Salesforce (Demo) integration	Salesforce (Demo)	✓
ServiceNow (Demo) integration	ServiceNow (Demo)	✓

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.

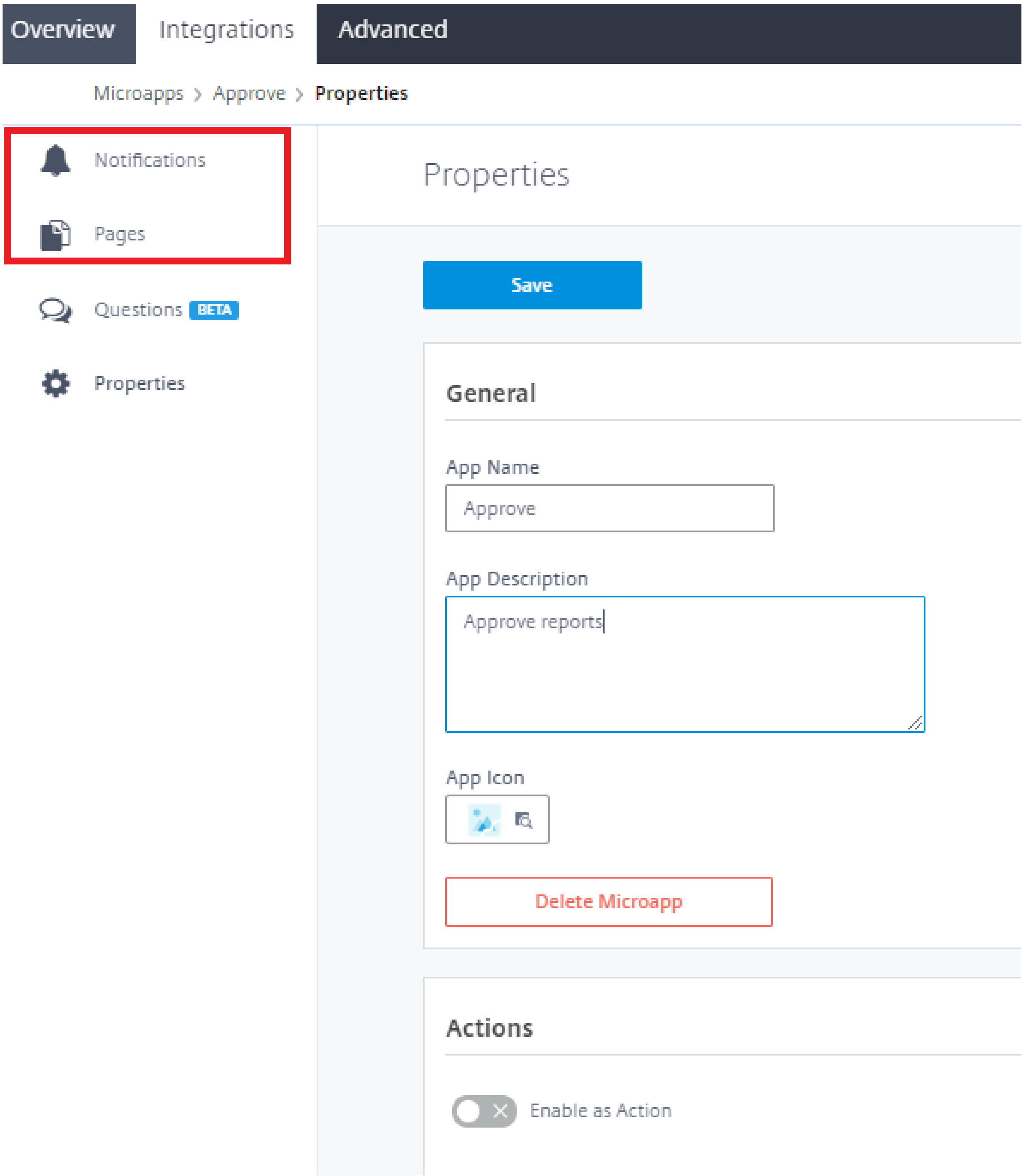
   After you add the blank microapp, it appears under the related integration on the **Integrations** page.
4. Return to the **Integrations** page and select **Blank Microapp** from the list under the integration.

   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

App Icon

Add Action Pages

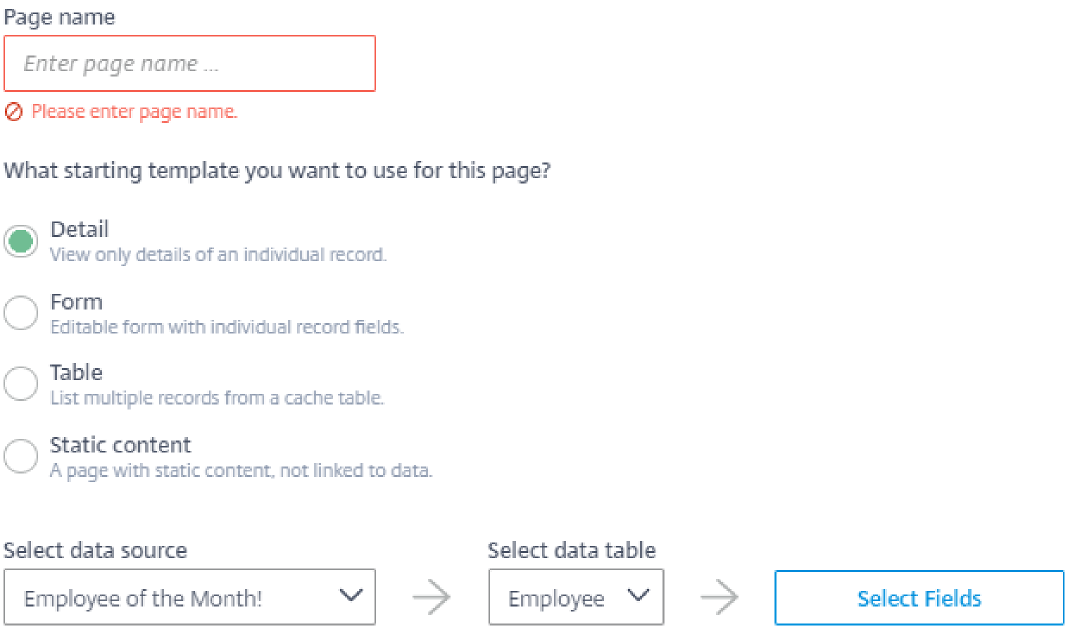
Add an action page for this microapp. Select from the starting templates, then customize the page in the builder.

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.

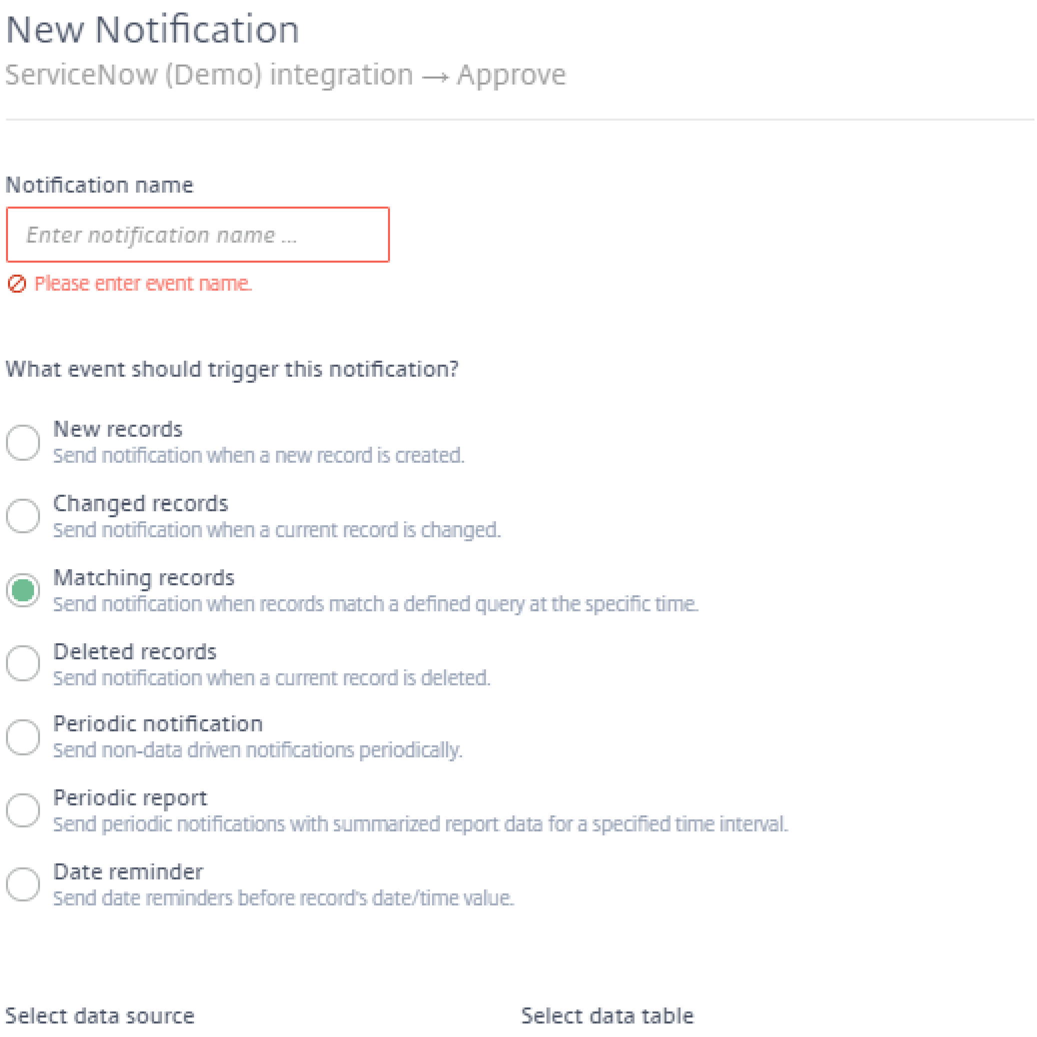
Add Event Notifications

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

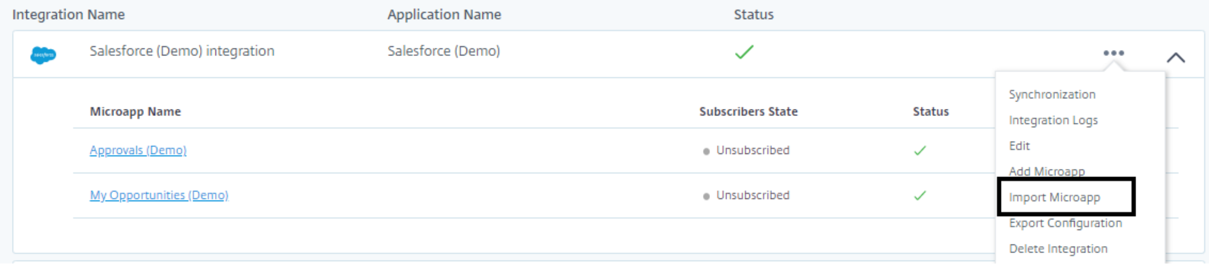
Import a microapp that you created in another instance. You can then edit or add pages or notifications or both.

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.

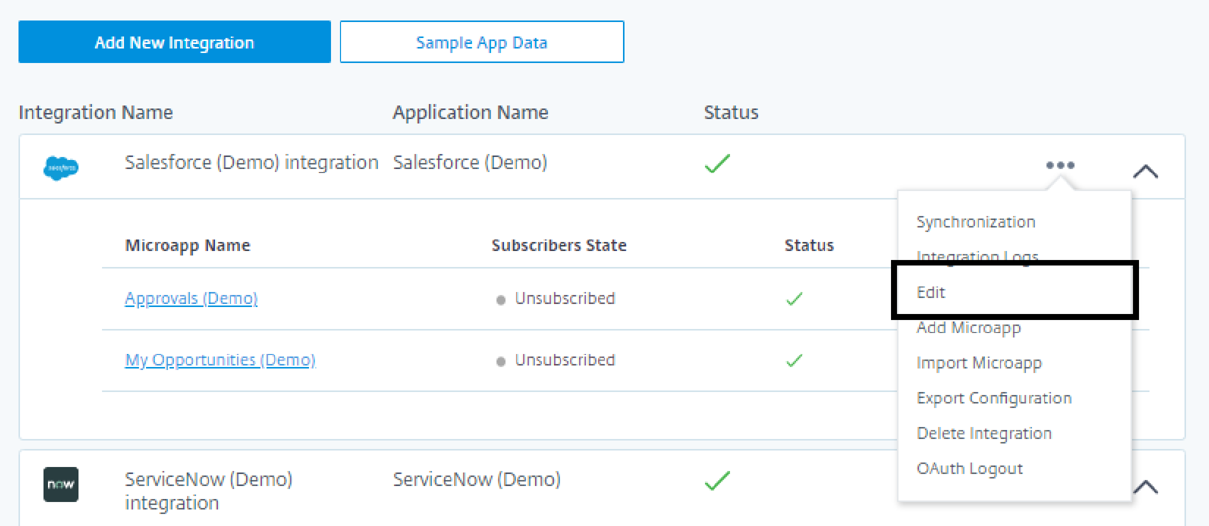
2. Change the microapp's name and description if needed.

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	✓

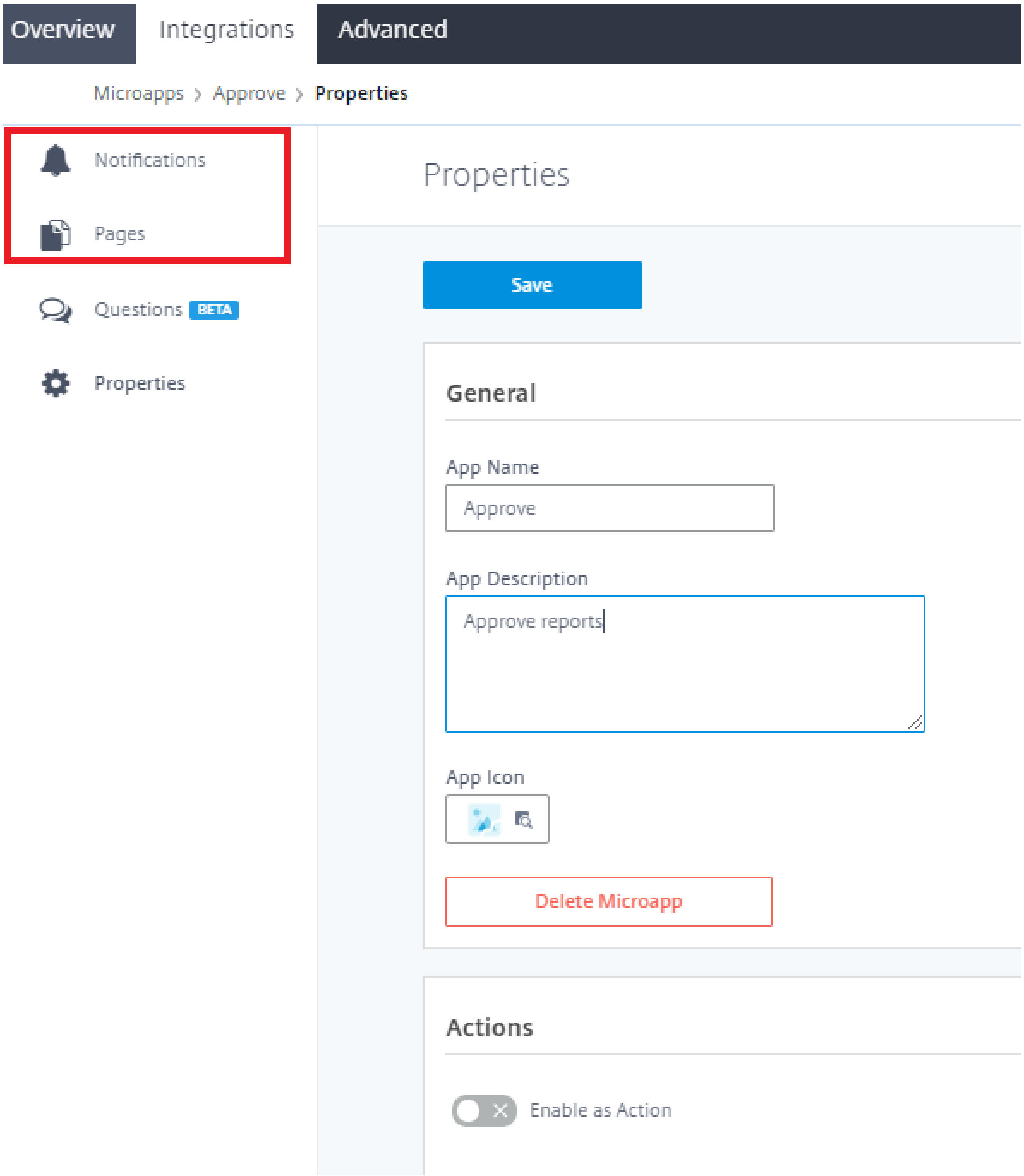
1. From the **Integrations** page, select the menu next to the integration to which you want to add a microapp.
2. Select **Edit**.

   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.

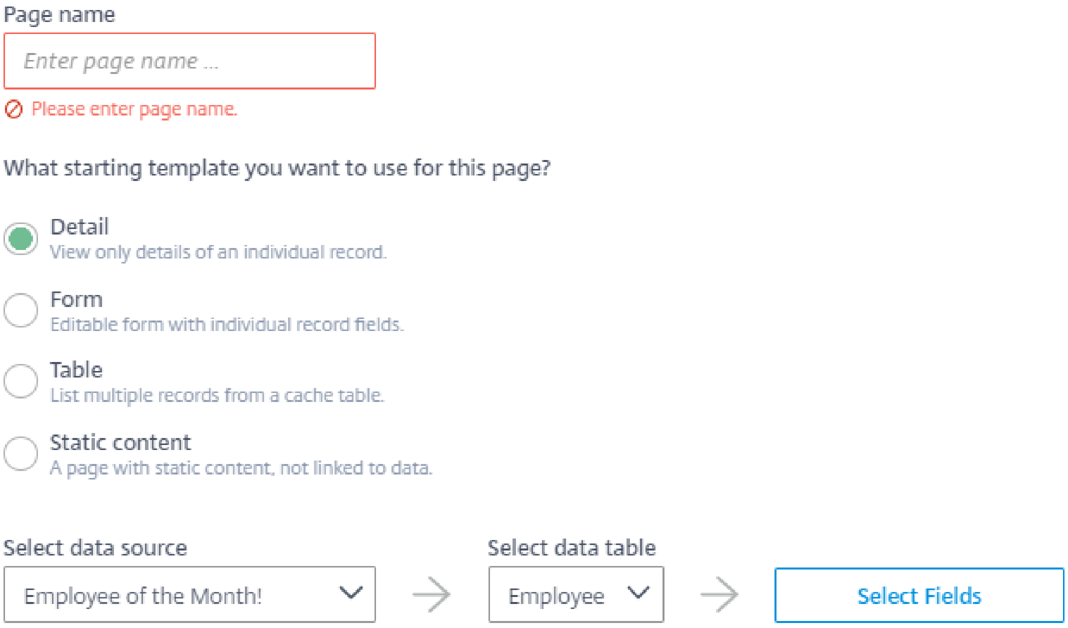
2. Select **Pages**, and **Add New Page**.

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

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:

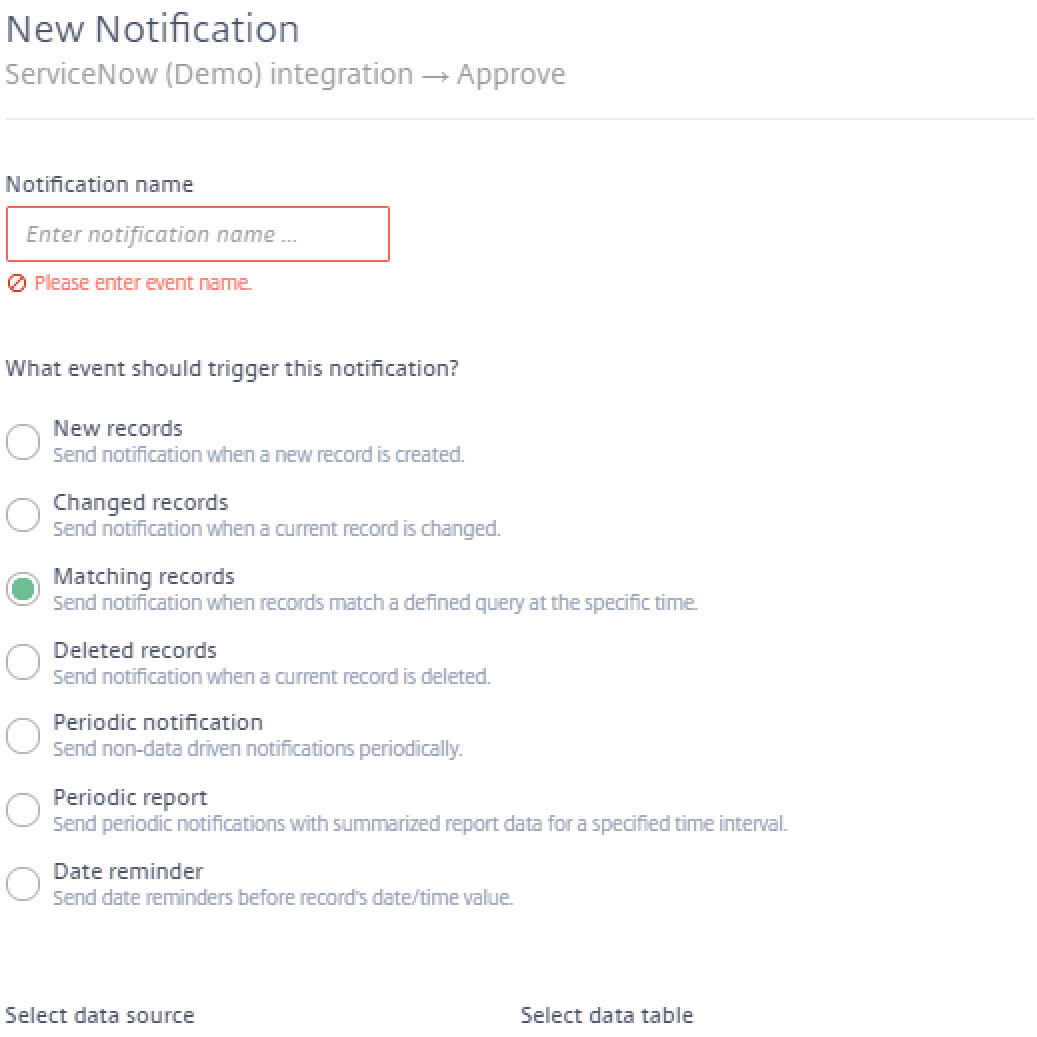
1. Select the microapp to which you want to add an event.

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

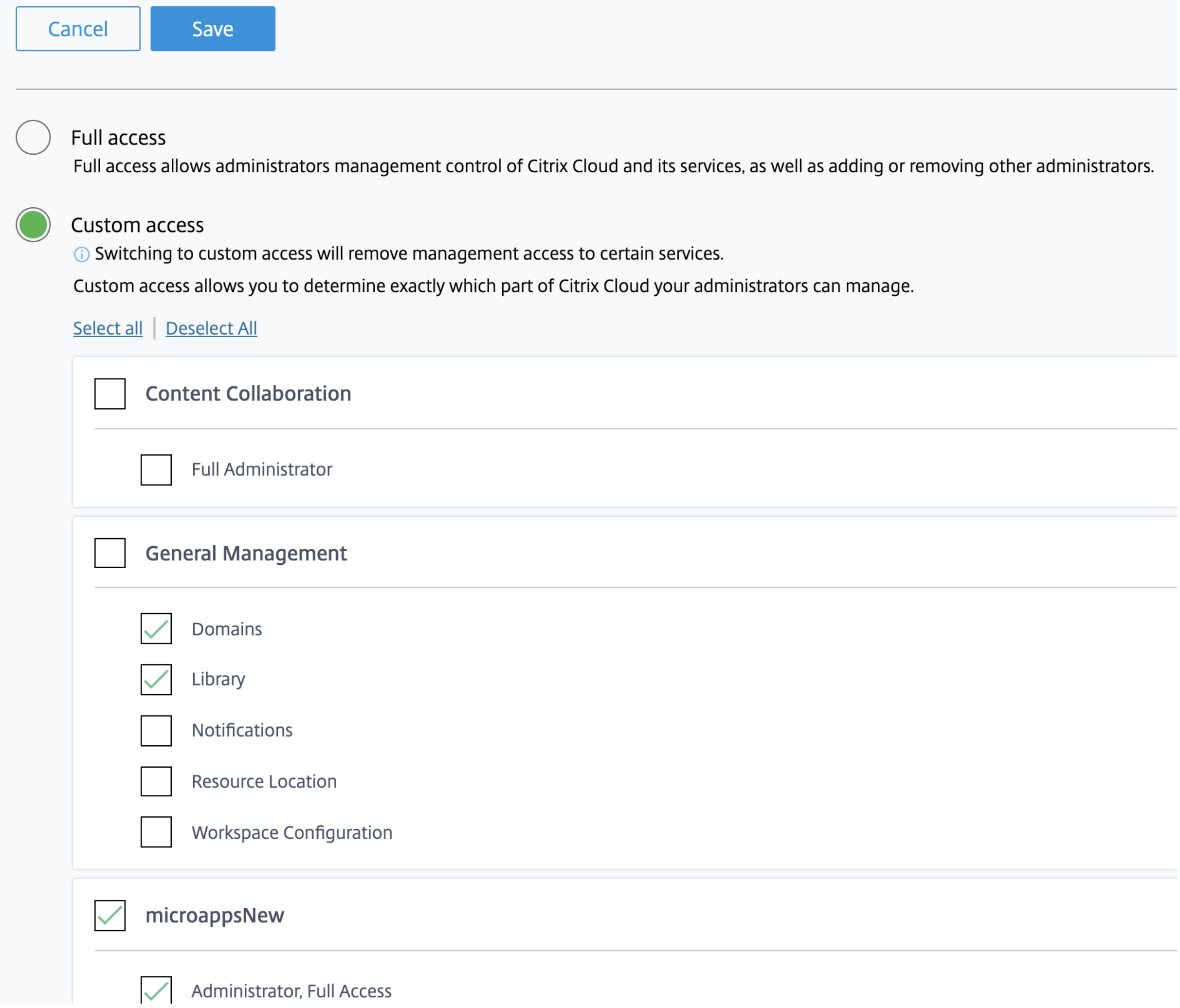
Before you begin, grant correct administrator access to add subscribers to your microapps. Use this delegated permissions process to enable administrators to add subscribers.

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.

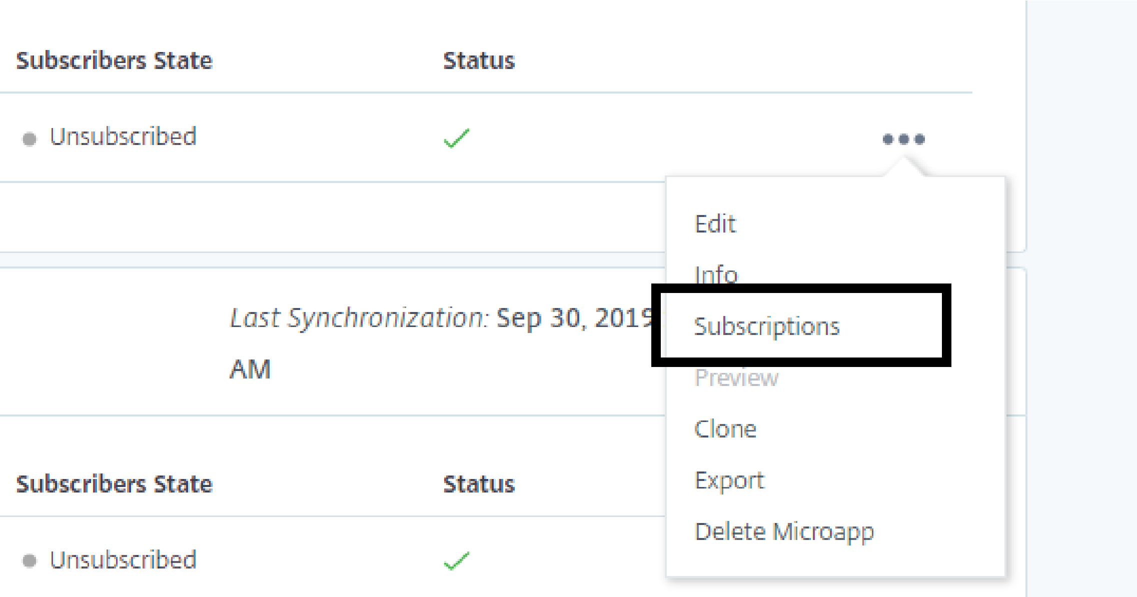
Manage subscribers

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.

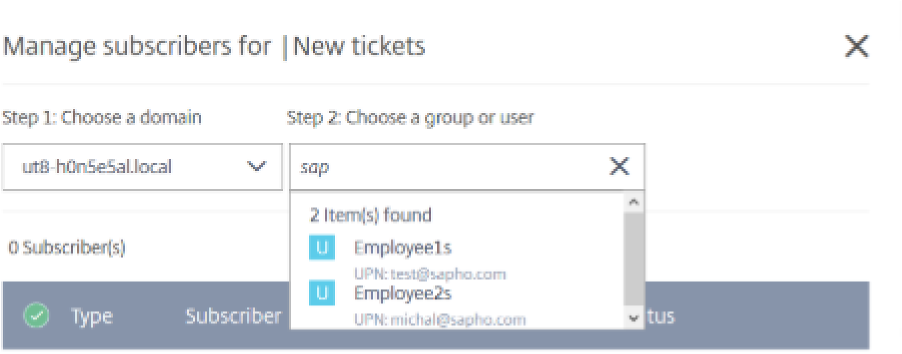


Subscribers State	Status
● Unsubscribed	✓

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: utB-h0n5e5al.local

Step 2: Choose a group or user: sap

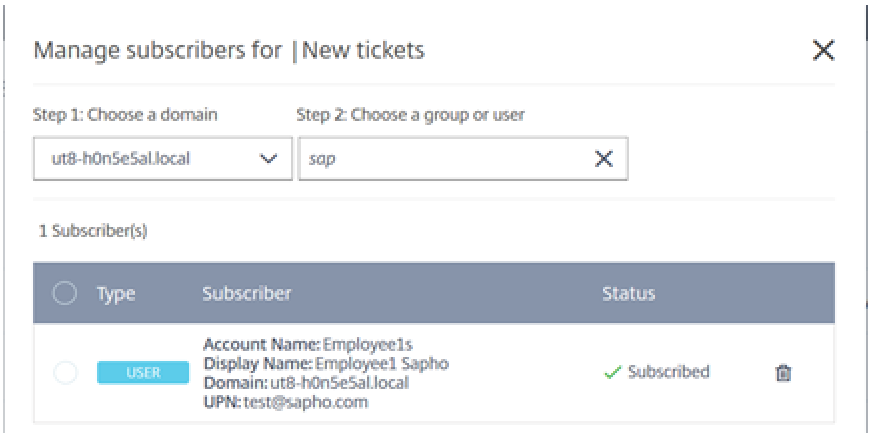
2 Item(s) found

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 with Citrix Workspace

February 9, 2021

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.

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

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

Locations
Corporate Headquarters | 851 Cypress Creek Road Fort Lauderdale, FL 33309, United States
Silicon Valley | 4988 Great America Parkway Santa Clara, CA 95054, United States