TEAMCENTER

Teamcenter OData API Framework

Teamcenter 2312

SIEMENS

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow’s idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: **support.sw.siemens.com**

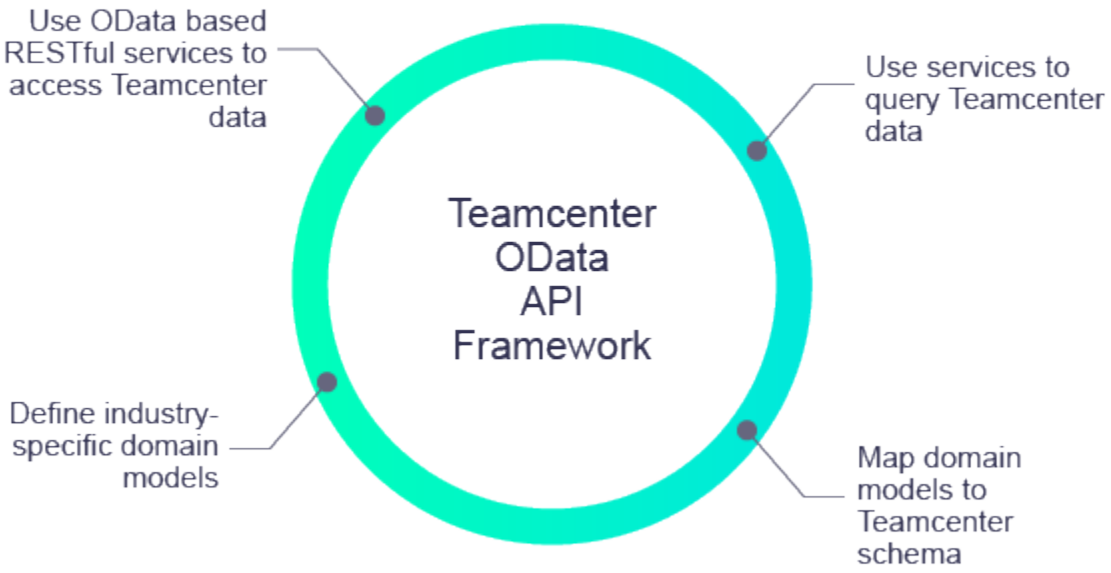
Send Feedback on Documentation: **support.sw.siemens.com/doc_feedback_form**

Contents

1. What is the Teamcenter OData API framework?

The Open Data Protocol (OData) API is an API service that allows developers, partners, users, and other stakeholders to access Teamcenter data using OData services.

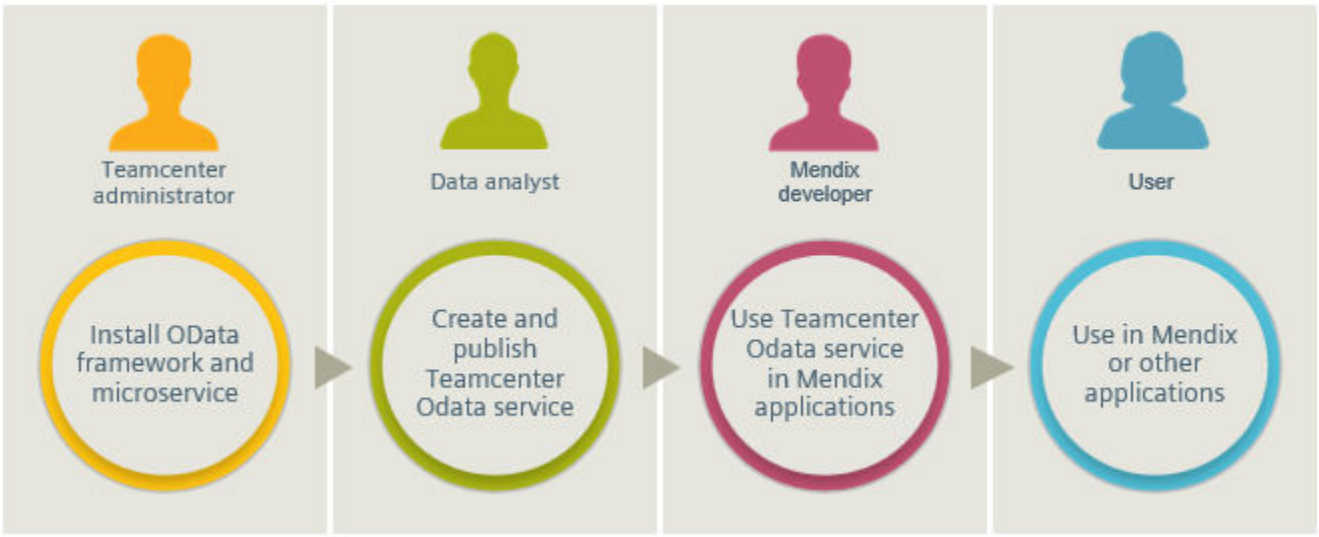
Using a low-code Mendix app called the OData Schema generator, you can easily create OData services to access Teamcenter data. Once you create the services, you can deploy them to Teamcenter or the Mendix Data Hub. You can access these services using any REST based-client or Mendix.



Where do I go from here?

Installer	**Install OData Support**
Administrator	**Set up OData Schema Generator**
Developer and Data Analyst	
Create and deploy OData services	See **the steps on how to create and deploy a service**.
Use OData services in Mendix	Refer to the **overview of using OData services in Mendix**.
Use OData services with a REST client	See **the sections on using the sample Postman collection**.

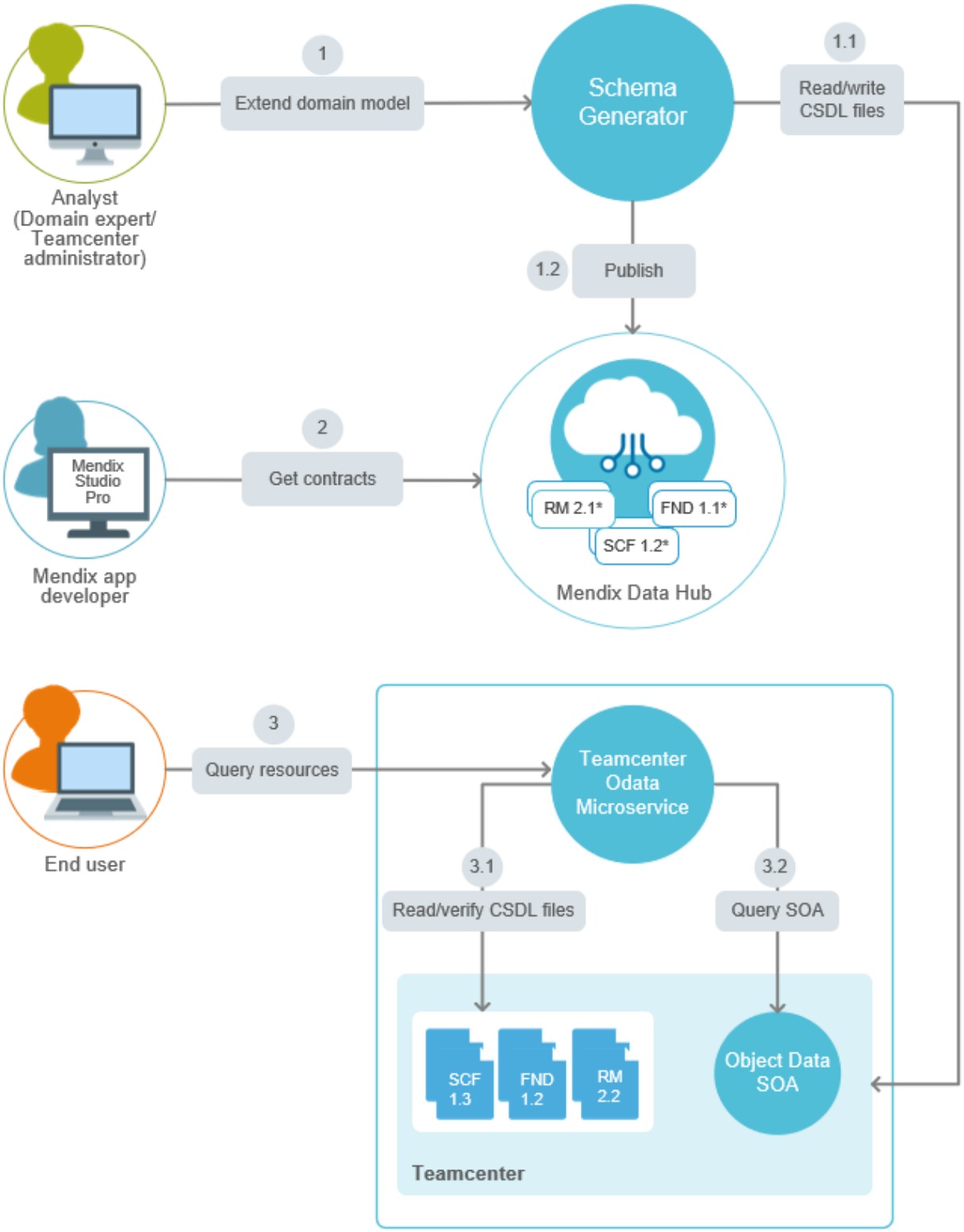
2. The process for using the Teamcenter OData framework



- The Teamcenter administrator installs OData support and the OData microservice.
- A data analyst uses the OData Schema Generator app to create and publish the Teamcenter OData Service.

  The service can be published on Teamcenter and the Mendix Data Hub.
- A Mendix developer uses the Teamcenter OData Service in Mendix apps.
- End users use Mendix apps to access Teamcenter content. The Teamcenter content is accessed using the Teamcenter OData Service.

3. Teamcenter OData framework architecture



OData Schema Generator

- This is a Mendix application to author OData-compliant services.
- This application creates an OData schema in the form of a Common Schema Definition Language (CSDL) XML file and mapping files in JSON format.
- The JSON mapping files are saved as datasets in Teamcenter.
- The OData Schema Generator publishes the service to Teamcenter and Mendix Data Hub.

OData microservice

- This microservice transforms OData requests to Teamcenter object model.
- It uses Teamcenter SOA and returns OData 4.0 compliant entities.
- By default, it fetches the latest published minor version of the OData schema from the Teamcenter server.

Mendix Data Hub

- The Mendix Data Hub is a place where assets such as data, APIs, and functions can be published. Mendix developers can use these data assets in their Mendix applications without the need to build new integrations.

4. Install OData support

You can install OData support using either Deployment Center or Teamcenter Environment Manager.

Deployment Center

In the **Applications** step, select the following components:

- **Teamcenter** > **OData Microservices**
- **Teamcenter** > **Object Data Services**

Teamcenter Environment Manager

In the **Features** panel, select the following features:

- **Extensions→Object Data Services**
- **Microservices→OData Microservice**

After this install, **run the utility to convert Teamcenter SOAs to JSON files**.

5. Post-install setup: Run the utility to convert Teamcenter SOAs to JSON files

After installing **Object Data Services** and **OData Microservice**, you must convert the Teamcenter SOA template XML files to JSON format by running the *XmlToJsonSoaApi.pl* utility.

Prerequisites for running the XmlToJsonSoaApi.pl utility

1. Ensure that the Teamcenter SOA template files are located in the *TC-DATA\model* folder.

2. Ensure that the dependent Perl modules are installed.

   a. Download and extract the contents the following Perl modules:

Perl Module	Download Location
JSON::DWIW	**https://metacpan.org/pod/JSON::DWIW**
JSON::Syck	**https://metacpan.org/pod/JSON::Syck**
XML::XML2JSON	**https://metacpan.org/pod/XML::XML2JSON**
JSON	**https://metacpan.org/pod/JSON**

   b. Paste all the *.pm* files in the modules to the following locations:

JSON::DWIW	*TC-ROOT\perl\lib\JSON*
JSON::Syck	*TC-ROOT\perl\lib\JSON*
XML::XML2JSON	*TC-ROOT\perl\lib\XML*
JSON	*TC-ROOT\perl\lib*

Run the XmlToJsonSoaApi.pl utility

1. Using the Teamcenter command prompt, go to the directory *TC-ROOT%\bin\ods0objectdataservices*.

2. Run the following command:

```
tcperl XmlToJsonSoaApi.pl [source] [destination] {options}
```

   Where:

   source Specifies the directory where XML template files are located

   destination Specifies the directory where the JSON files are created.

IncludeInternal	(Optional) Converts the unpublished SOAs to JSON.
debug	(Optional) Enables the debug mode and prints in log file.

Example:

```
tcperl XmlToJsonSoaApi.pl C:\apps\tc\tc13\TD\model
C:\apps\tc\json_apis
```

The output is created in the destination folder. The output consists of:

serviceList.json	The list of SOAs whose corresponding JSON files are created.
logfile.txt	The log file created when you run the utility.
ActionMapping	The folder containing the JSON files sorted based on LibraryName/Year/Service/SOAName.json.

Copy the output to the to the resources directory of the ODataMappingApp

After running the utility and generating the output, copy the output to the resources directory of the ODataMappingApp as follows:

1. In Mendix Studio, open the ODataMappingApp.
2. Click **Project** > **Show Project Directory in Explorer**.

   This opens the project directory of the ODataMappingApp in Windows.
3. Open the **Resources** directory and copy the output files to this location.

   You must copy the following files:

   - *serviceList.json*
   - *ActionMapping*

6. Using the OData Schema Generator to create and maintain services

Setting up OData Schema Generator

Download the OData Schema Generator app into your Mendix project

You can get the schema generator app, *OData_Schema_Generator_App.mpk*, from the one of the following directories in the Teamcenter software kit:

TEAMCENTER_KIT\wntx64\additional_applications\ods0objectdataservices

OR

TEAMCENTER_KIT\lnx64\additional_applications\ods0objectdataservices

After you get the app, import it into your Mendix project.

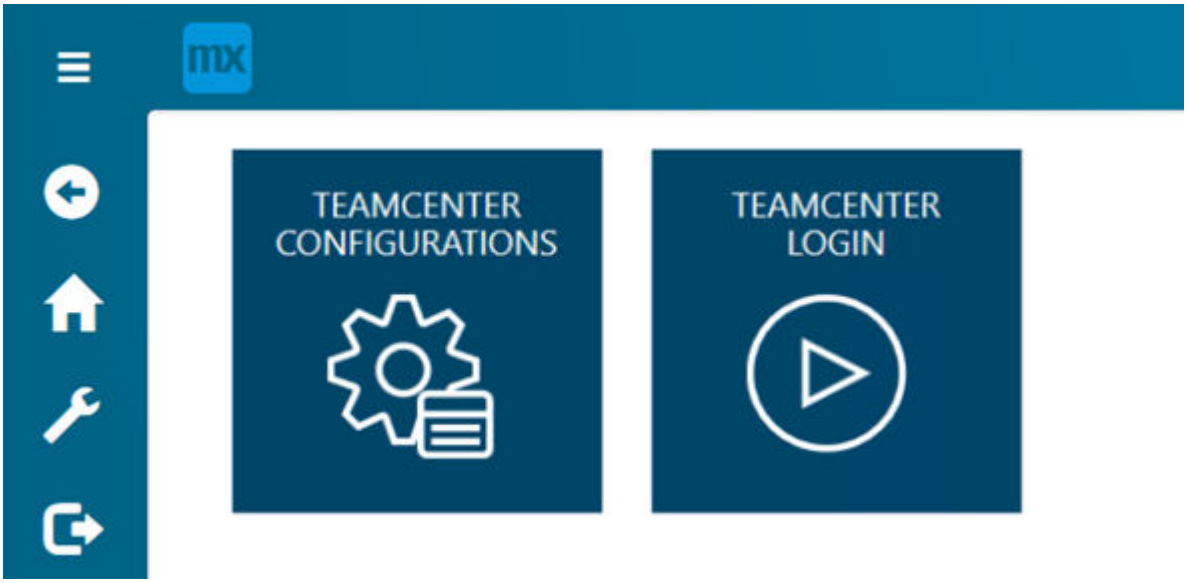
> Note:
>
> The OData Schema Generator app is available for preview and will be licensed in the future. Contact customer support to get the license details.

Configure the connection to Teamcenter

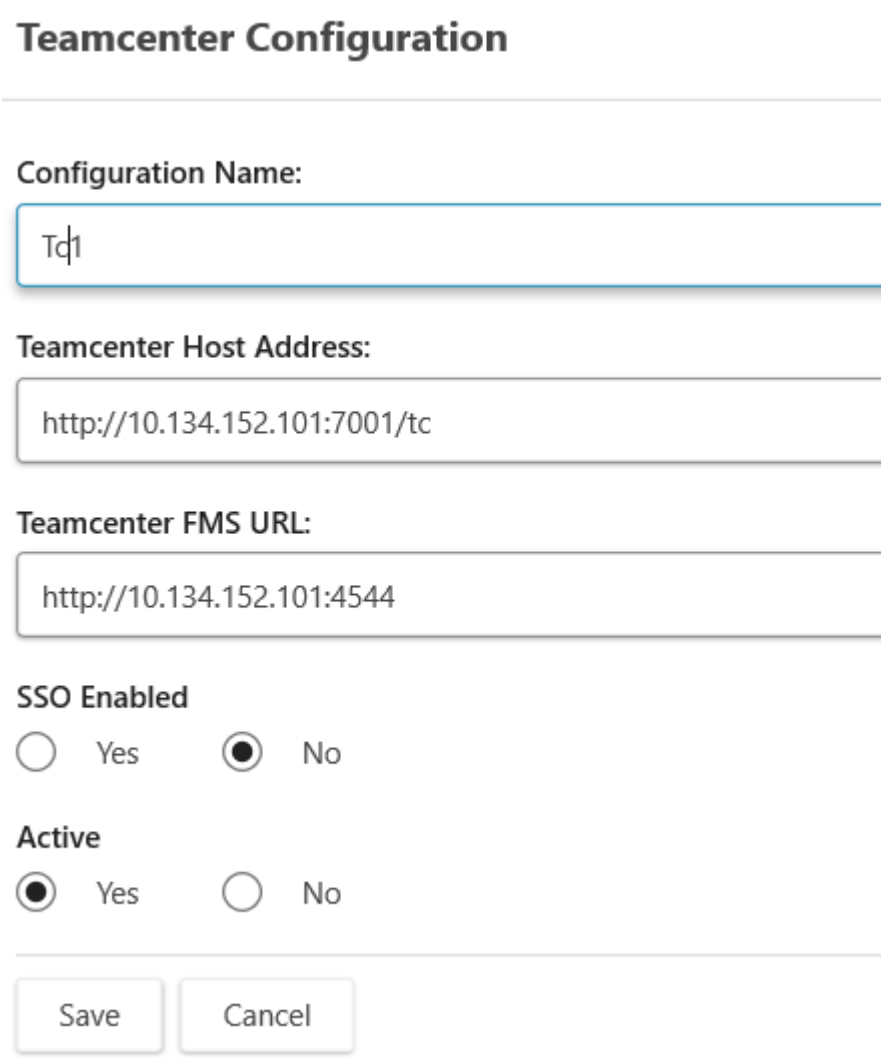
The OData Schema Generator must connect to Teamcenter. For this, you must specify the Teamcenter connection details as follows:

1. Run your Schema Generator app in Mendix Studio Pro 8.10.2.

2. Click the **TEAMCENTER CONFIGURATIONS** tile.



3. Click **New** and in the **Teamcenter Configuration** dialog box, specify the Teamcenter configuration information.

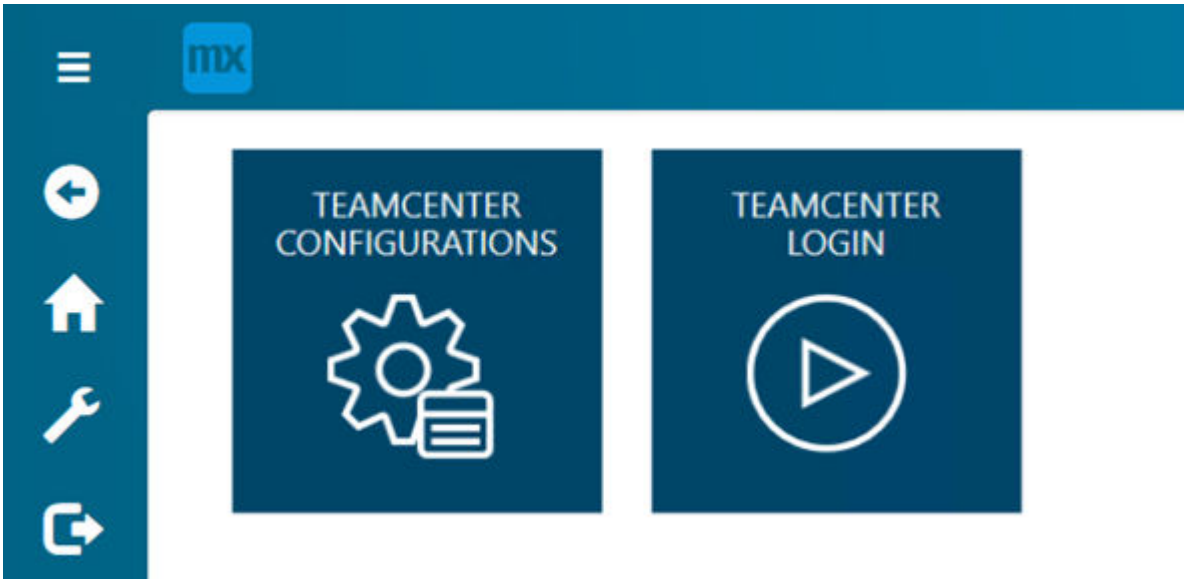


4. Click **Save**.

Log on to Teamcenter

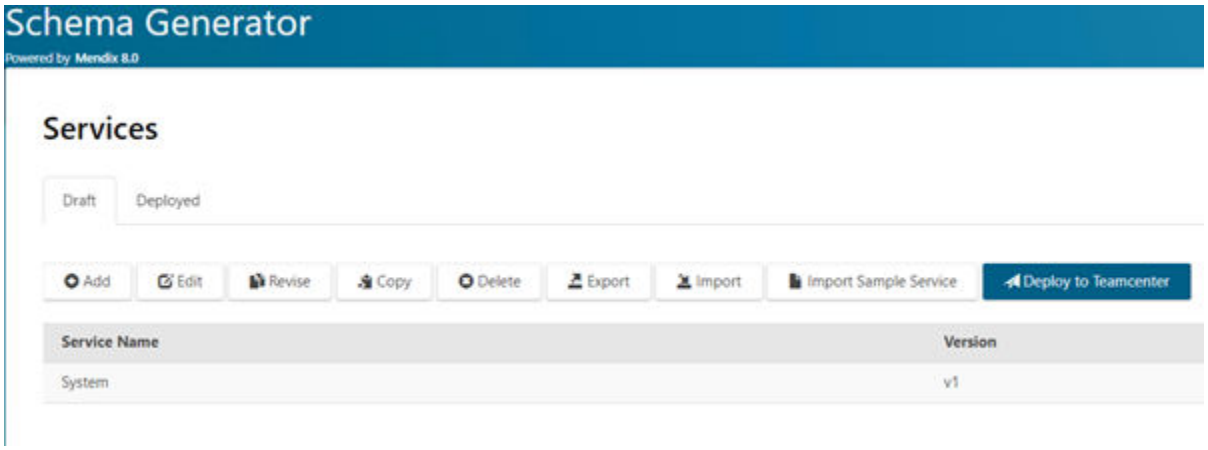
After you set up the Teamcenter configuration, log on to Teamcenter.

1. Click the **TEAMCENTER LOGIN** tile.



2. In the Teamcenter logon page, enter your Teamcenter credentials.

Once logon is successful, you see the **Services** page.



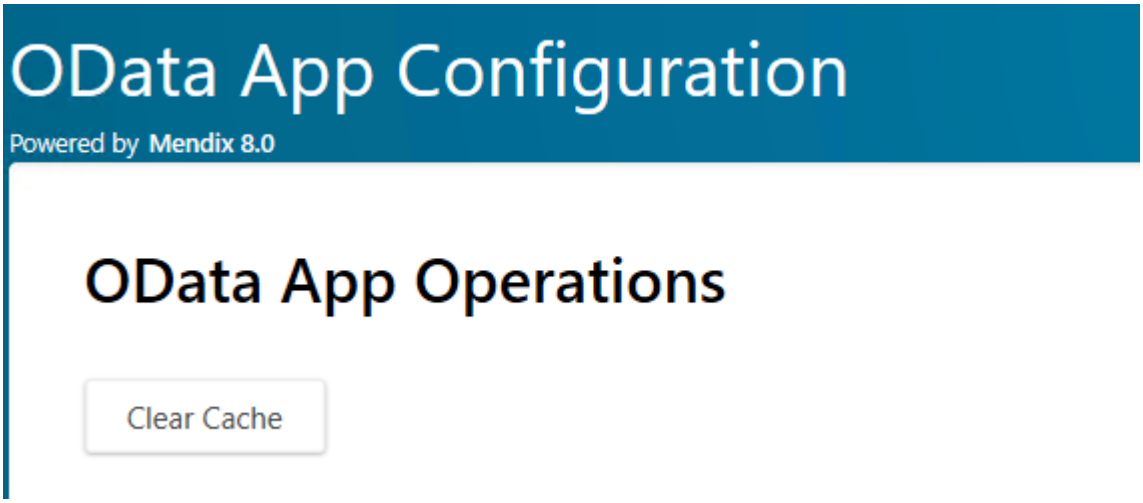
Update the OData Schema Generator with Teamcenter data model changes

The OData Schema Generator retrieves and caches the Teamcenter meta model information the first time you logon to Teamcenter from the Schema Generator. However, this cache is not deleted subsequently for performance reasons. If there are Teamcenter data model changes and the administrator wants the changes in the Schema Generator, you must delete the cache as follows:

1. Ensure all users are logged out of the OData Schema Generator app.

2. Click **OData App Configuration** to open the OData App Configuration page.

3. In the OData App Configuration page, under the **OData App Operations** section, click **Clear Cache** .



4. In the **Confirmation** dialog box, click **Proceed**.

5. After the cache is cleared, logout and logon the Schema Generator app to get the latest Teamcenter meta model information.

Configuring the connection to Mendix Data Hub

Update constants

Before you set up the connection to Mendix Data Hub, ensure that you update the following constants:

From the **Schema Generator** > **Resources** directory update:

- **DatahubLocation**: URL of the data hub

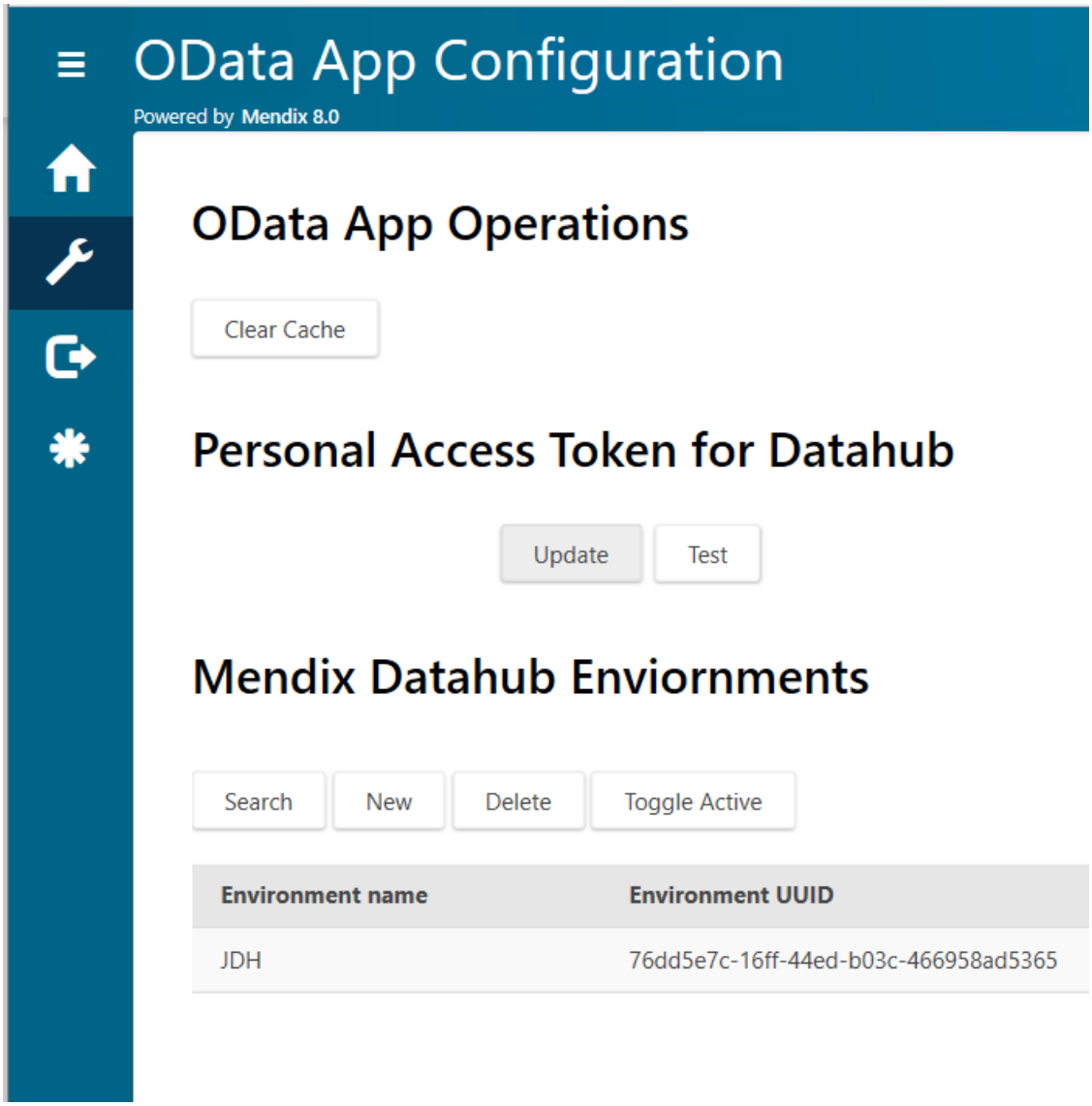
From the **OAuth** > **Config** directory update:

- **AuthenticationServer**: Server for logon
- **ClientID**: The OAuth ID
- **ClientSecret**: The OAuth secret
- **RedirectEndpoint**: The format *http://app-address/oauth/callback*

Register a new Teamcenter environment on Data Hub

1. Click **OData App Configuration** to open the OData App Configuration page.

2. In the OData App Configuration page, under the **Mendix Datahub Environments** section, click **New** .



3. In the Register Environment page, specify the Mendix Data Hub connection information as follows:

   a. **Use existing environment**: Select the **No** option to register a new Teamcenter environment.

   b. **Environment Name**: Specify a name for this environment.

   c. **Environment Type**: Choose whether this environment is a production, non-production, or sandbox environment.

   d. **Environment Location**: Type the URL of the Teamcenter server.

   e. **Set Active**: Set it to **Yes** to make the environment active.

4. Click **Register**.

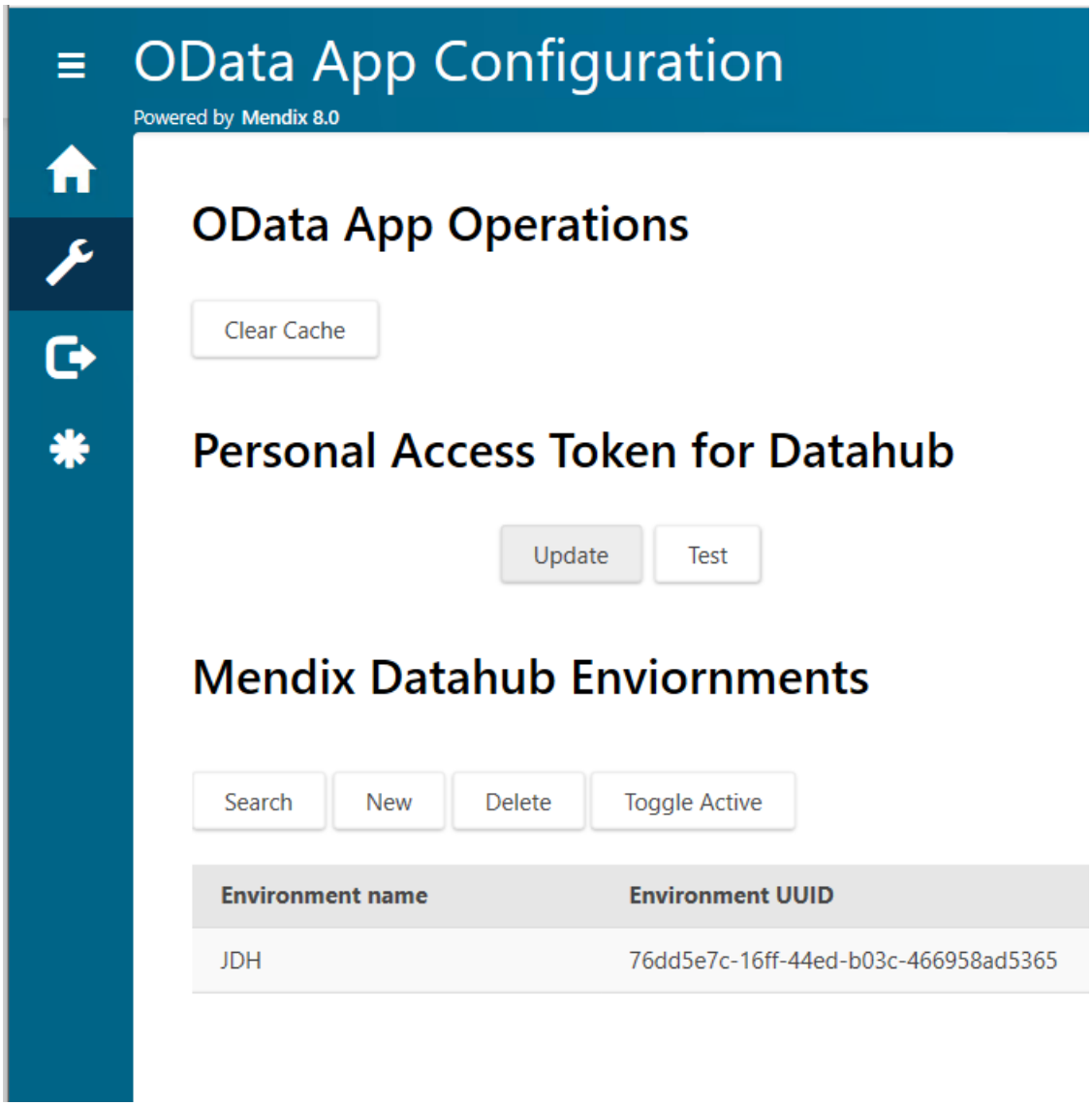
   If you have not logged on to Mendix Data Hub, a login dialog box appears. Log on to Mendix Data Hub.

   After logging on click **Register**.

Register an existing Teamcenter environment on Data Hub

1. Click **OData App Configuration** to open the OData App Configuration page.

2. In the OData App Configuration page, under the **Mendix Datahub Environments** section, click **New** .



3. In the Register Environment page, specify the Mendix Data Hub connection information as follows:

   a. **Use existing environment**: Select the **Yes** option to register an existing Teamcenter environment.

   b. **Environment Name**: Specify a name for this environment.

   c. **Environment Type**: Choose whether this environment is a production, non-production, or sandbox environment.

   d. **App UUID**: This is available on the **Teamcenter Environments for Datahub** page.

   e. **Environment UUID**: This is available on the **Teamcenter Environments for Datahub** page.

   f. **Set Active**: Set it to **Yes** to make the environment active.

4. Click **Register**.

   If you have not logged on to Mendix Data Hub, a login dialog box appears. Log on to Mendix Data Hub.

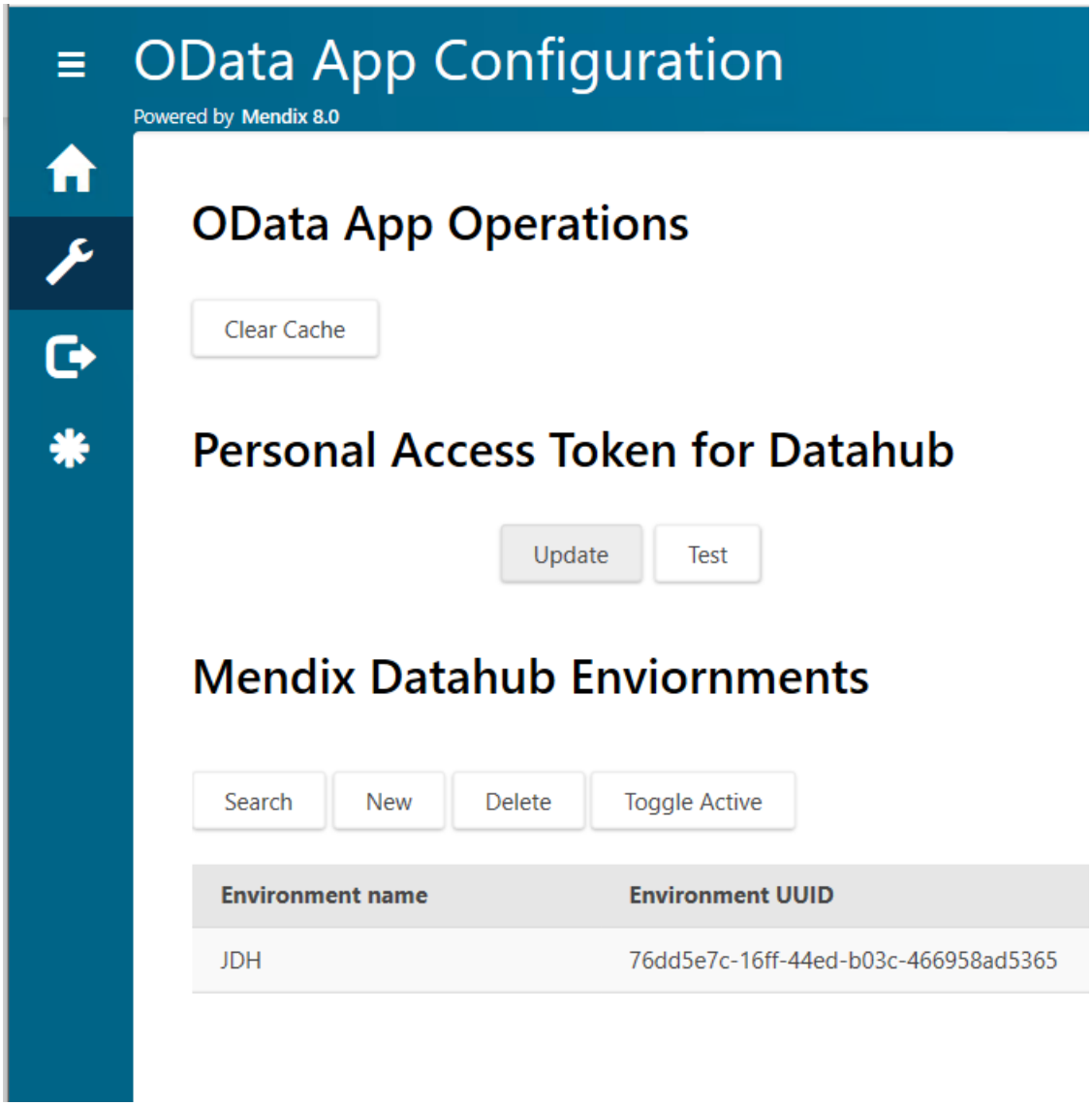
   After logging on, click **Register**.

Set up access to the Mendix Data Hub using Personal Access Token

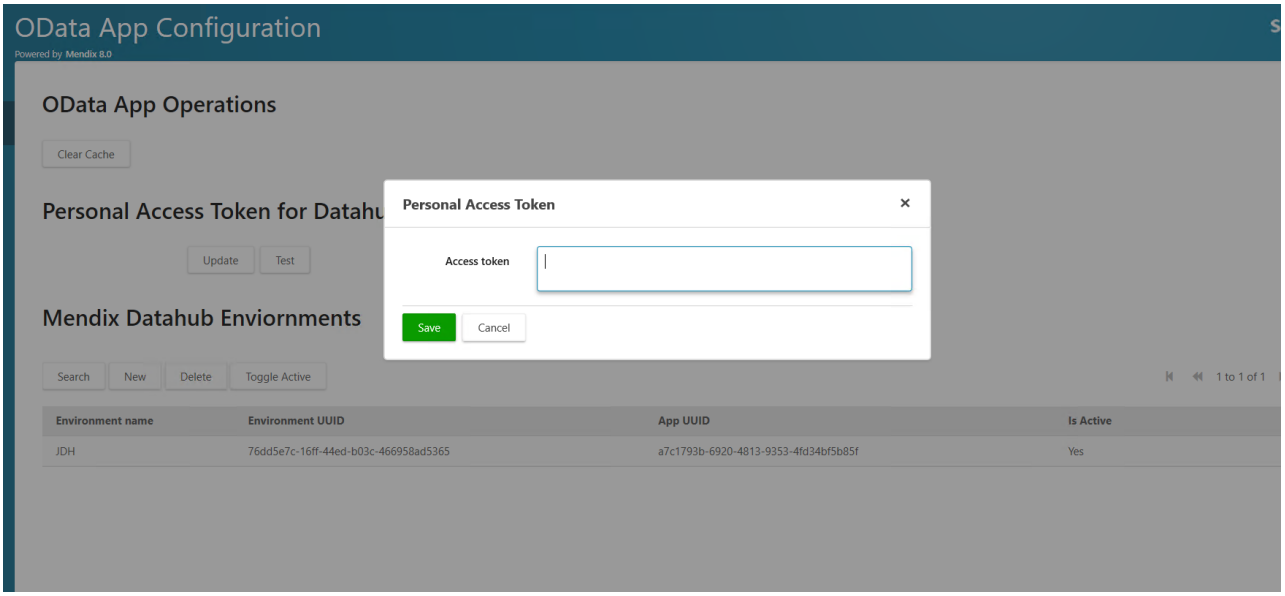
Prerequisites: You require the Personal Access Token (PAT) to set up access to the Mendix Data Hub. You can get the PAT from **https://warden.mendix.com/**.

1. Click **OData App Configuration** to open the OData App Configuration page.

2. On this page, under the **Personal Access Token for Datahub** section, click **Update** .



3. In the **Personal Access Token** dialog box, add the PAT and click **Save**.



4. Click **Test** to check if you can access the Mendix Data Hub with the token.

Creating and deploying services

Steps to create and deploy a service

You can create and deploy a service using two methods:

- Create a new service
- **Create a new service using existing services**

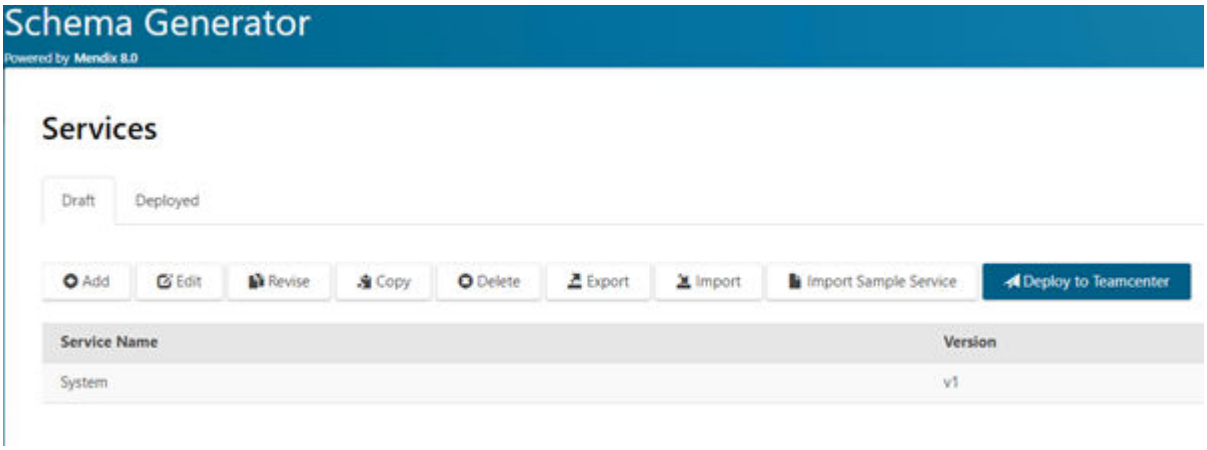
To create a new service, use the following steps:

- **Step 1: Specify the name of the service and create a namespace. You can add a new namespace or use an existing namespace.**
- **Step 2: Select the objects and the object properties you want in the service.**
- **Step 3: Specify operations that you want in the service.**
- **Step 4: Generate the service.**
- **Step 5: Deploy the service to Teamcenter.**
- **Step 6: Deploy the service to Mendix Data Hub.**

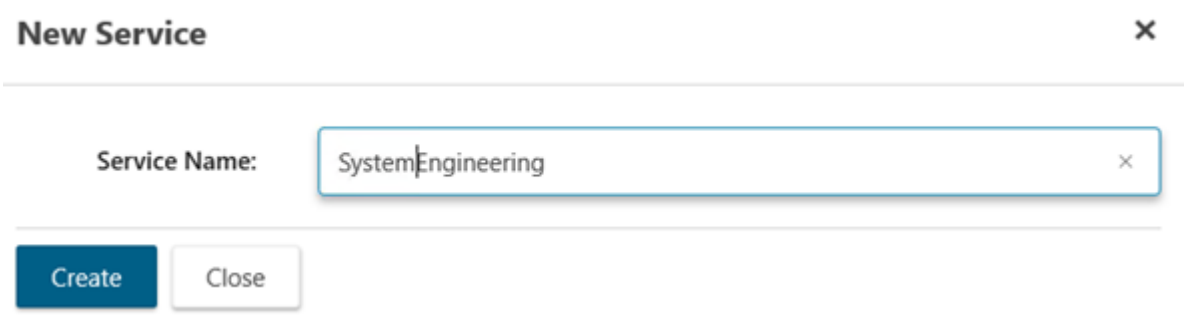
Step 1: Specify the name of the service, create a namespace, and add an existing service as a reference

Specify the name of the service

1. Log on to OData Schema Generator, and in the Services page, click **Add**.



2. In the **New Service** dialog box, specify the name of the new service and click **Create**.



Create a namespace

A namespace is a collection of objects and its properties. You can create a namespace to represent different parts of your service. For example, you can create a namespace named *Requirements* for your *Systems Engineering* service.

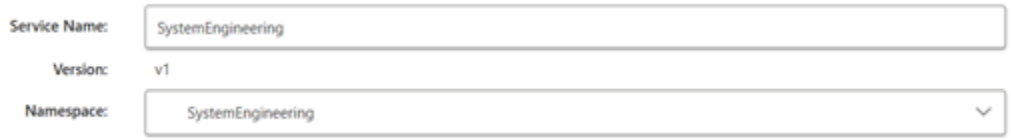
You can create a namespace by:

- Selecting an existing namespace
- Creating a new namespace

- Referencing an existing service and selecting the object and properties of the selected service to create a new namespace.

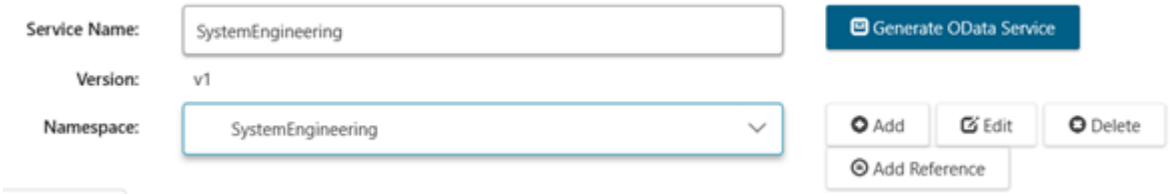
Selecting an existing namespace

- From the **Namespace** list, select the namespace.

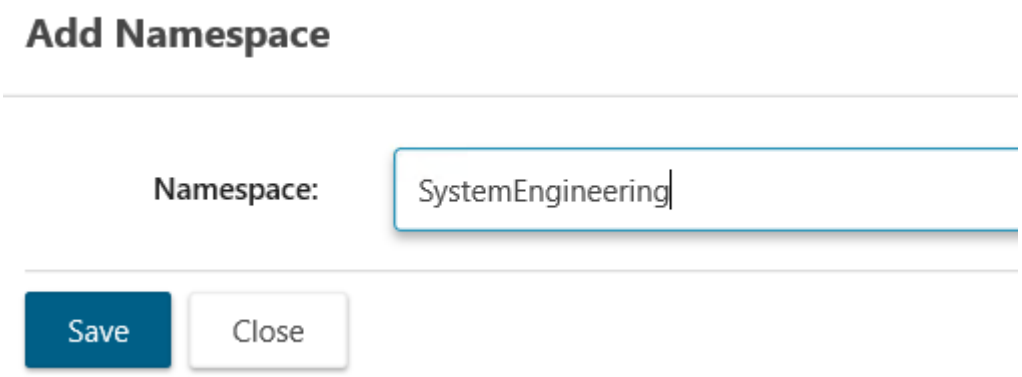


Create a new namespace

1. Click **Add** next to the **Namespace** list.



2. In the **Add Namespace** dialog box, specify the name of the namespace and click **Save**.



Add an existing service as a reference

1. Click **Add Reference** next to the **Reference Schema** list.

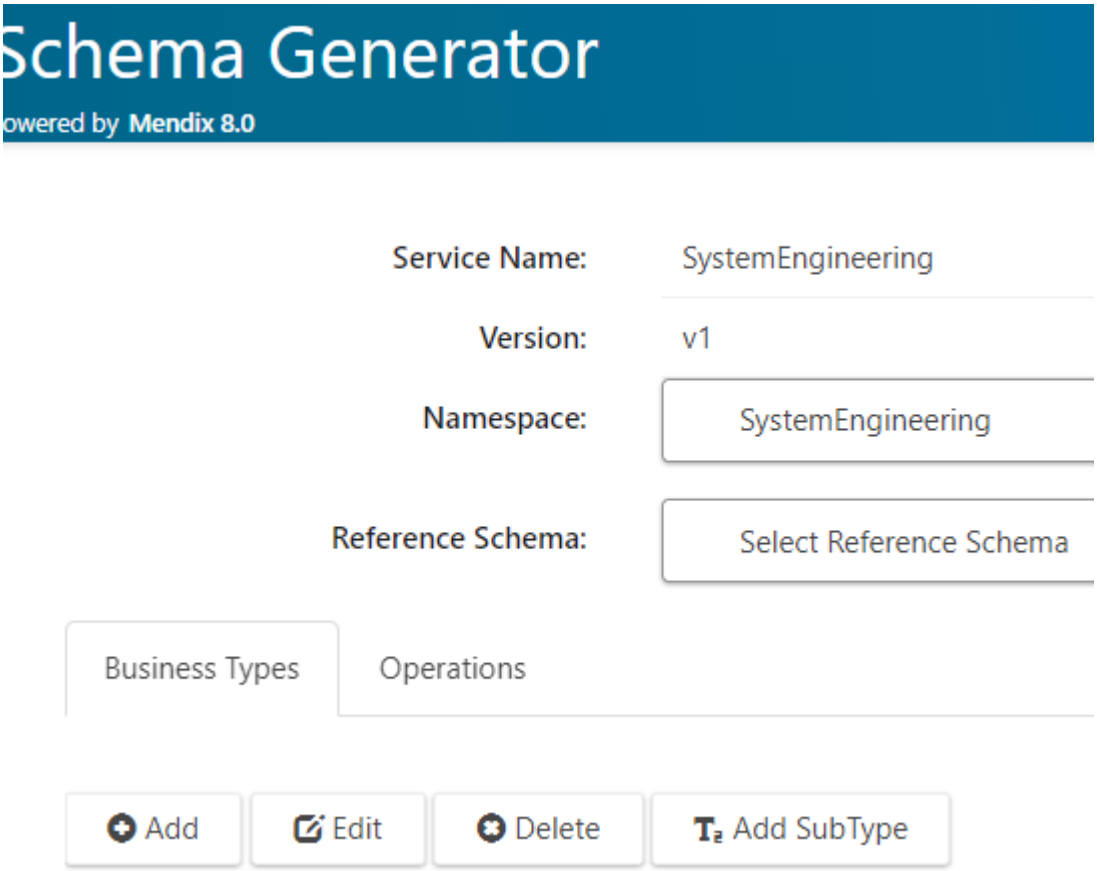
2. In the **Add Reference** dialog box, select a service from the **Deployed** tab and click **Add**.

After performing these steps, you must **select object and its properties for the service**.

Step 2: Select the objects and the object properties you want in the service

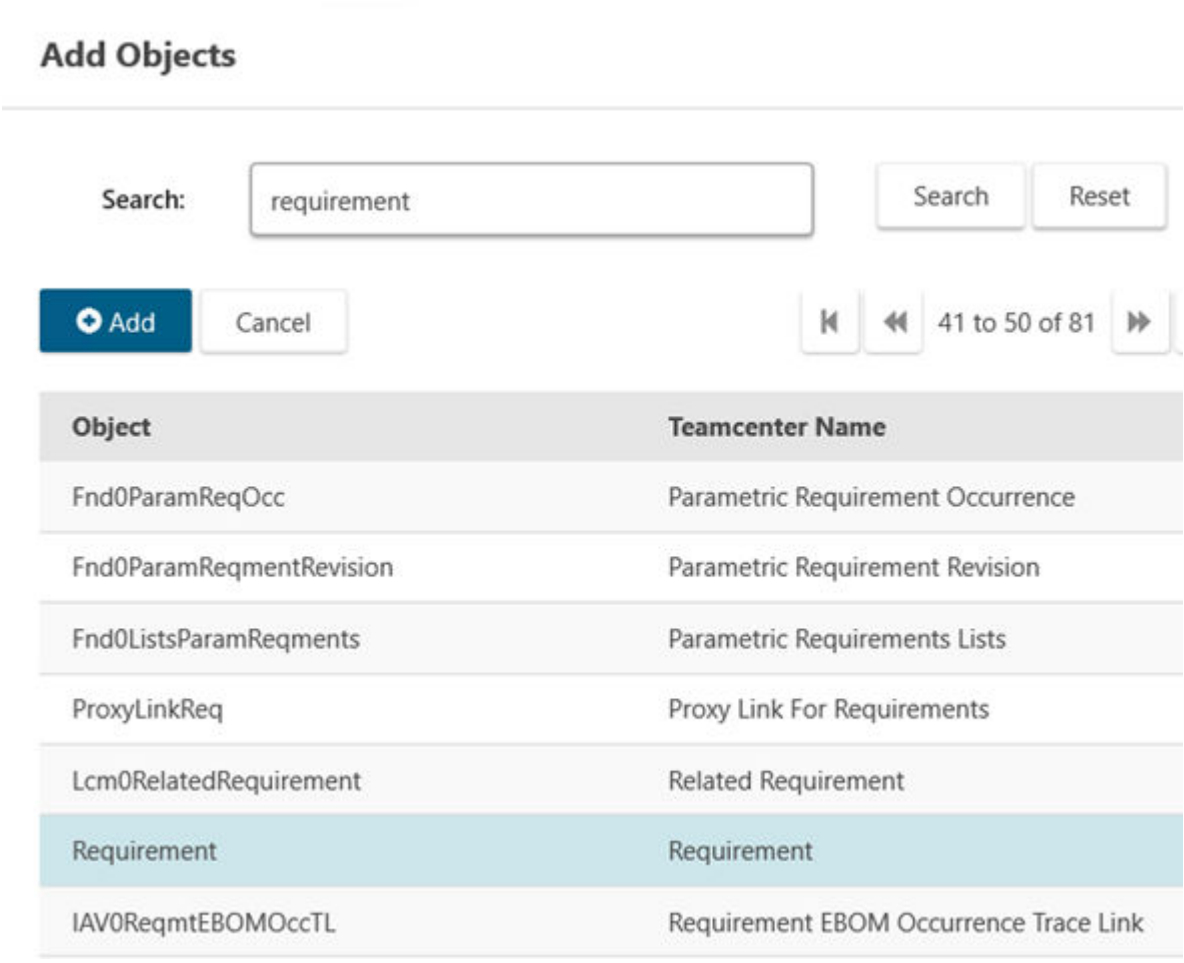
After you have **specified the service name and the namespace**, add object and properties to the service as follows:

1. In the **Business Types** tab, click **Add**.



2. In the **Add Objects** dialog box, search for and select an object.

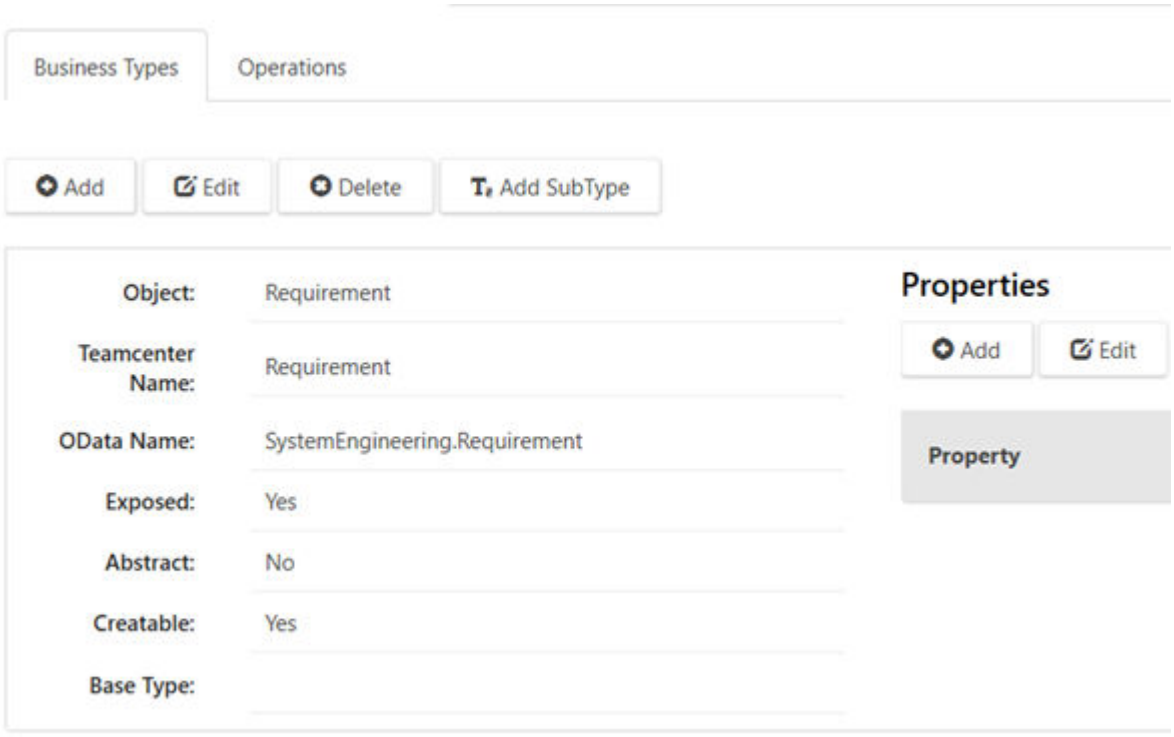
   After selecting the object, click **Add**.



You can add additional objects. Once you have finished adding objects, close the **Add Objects** dialog box.

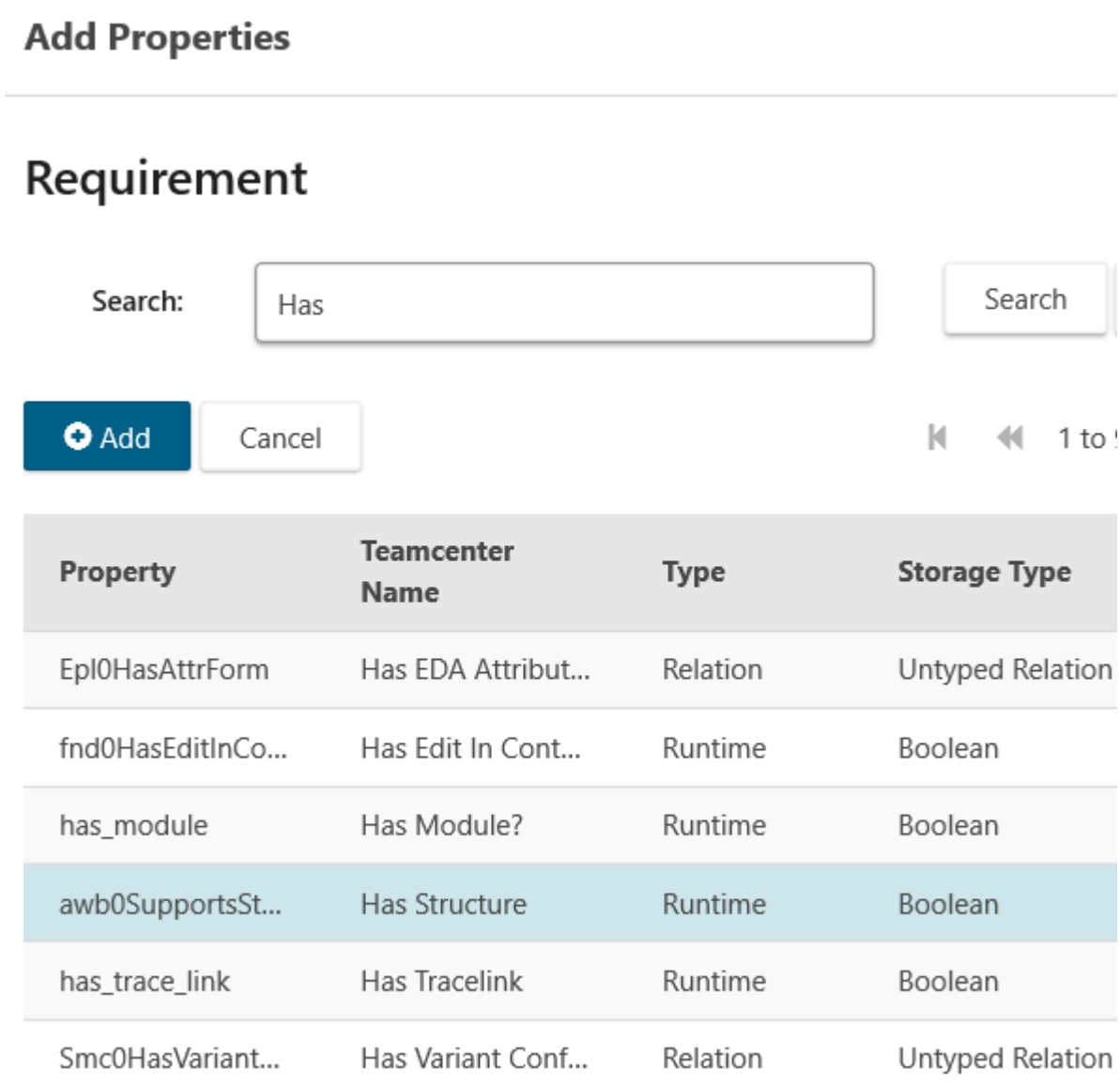
3. After adding objects, you must add properties for those objects.

   In the **Properties** section, click **Add**.

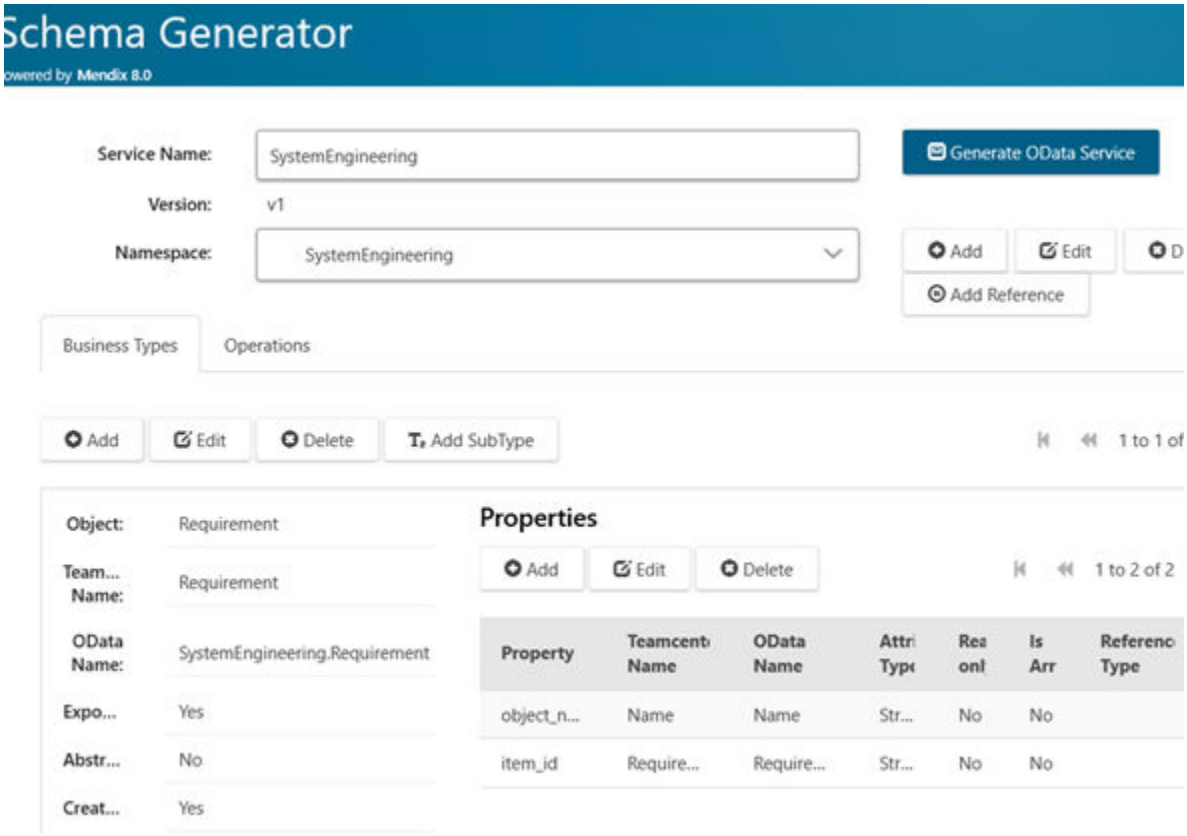


4. In the **Add Properties** dialog box, search for and select a property.

   Click **Add**.



Add additional properties as required. Once you are finished adding properties, close the **Add Properties** dialog box.



> Note:
>
> For file operations such as file upload, download, and creation, you must select a *dataset* as the object type.

After adding the object and its properties, you can:

- Edit and delete the object and its properties.
- Create a subtype of the object.
- You can also **add additional namespaces** and add object and properties under those namespaces.
- **Specify operations that you want in the service.**
- **Generate the service**.

Step 3: Specify the operations that you want in the service

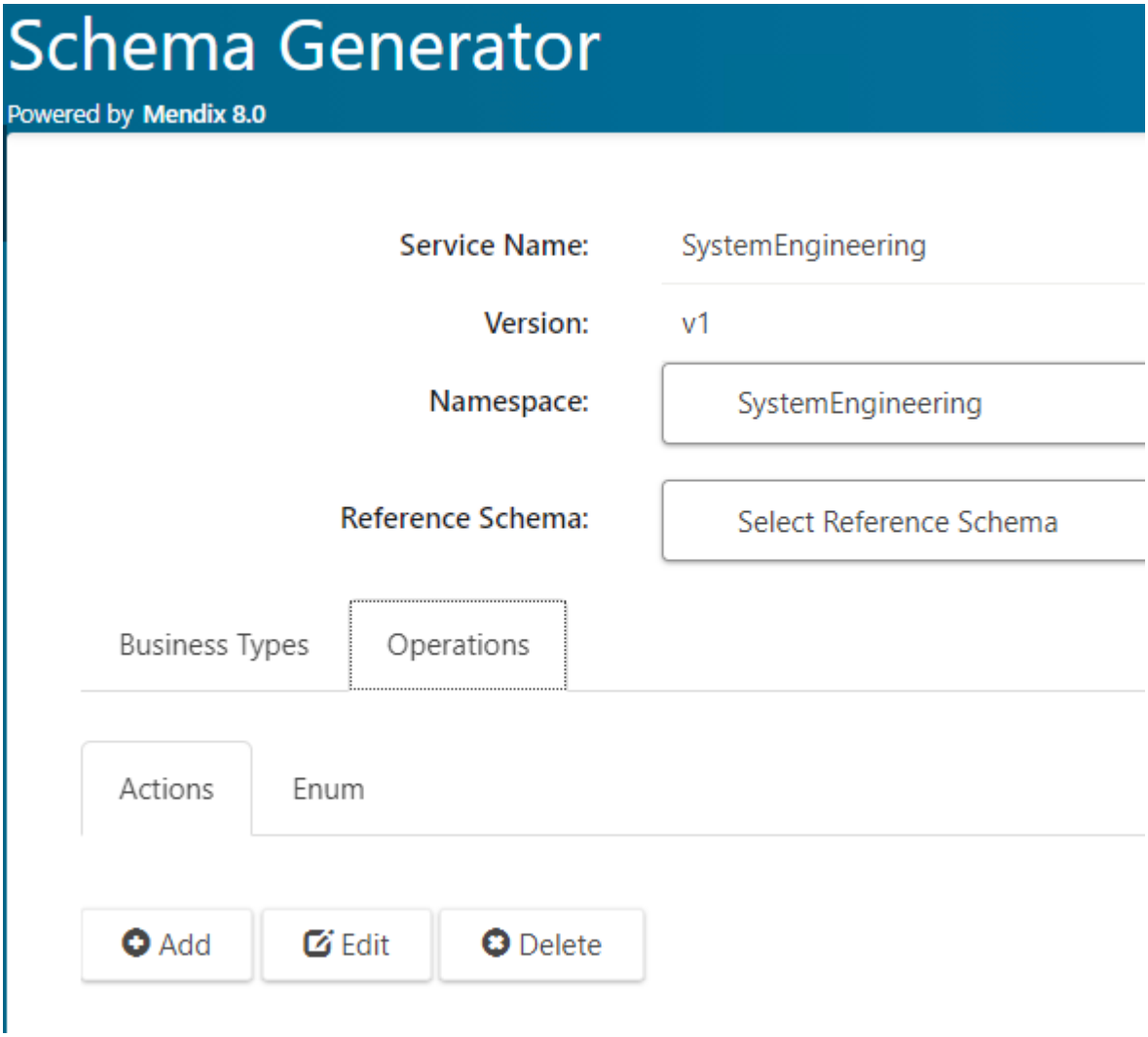
You can add Teamcenter SOA operations as actions in your OData service. These actions are unbounded, that is, not associated with a business entity. The process to create actions is as follows:

- **Select the Teamcenter SOA operation you want to use in the OData action.**
- **Add the request and return parameters to the action.**
- **Map the entity Parameter to OData business model entities.**
- **If the Teamcenter SOA operation input has multiple values, you can create an enum for these input values.**

You can add actions as follows:

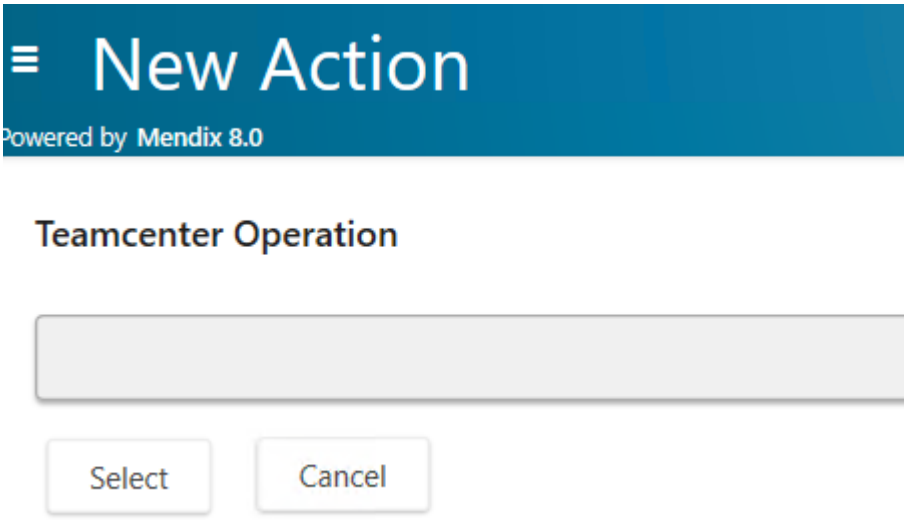
1. **Select the Teamcenter SOA operation you want to use in the OData action**

   a. In the Schema Generator page, click the **Operations** tab and then click the **Actions** tab.

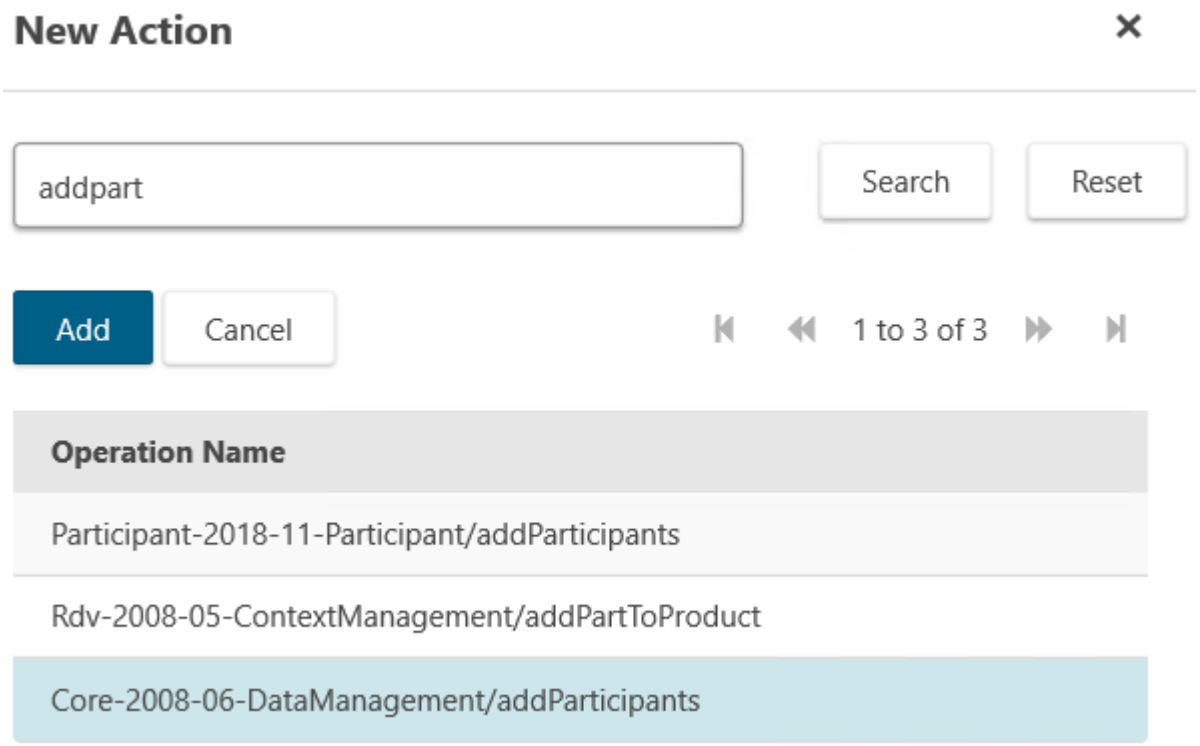


   b. Click **Add**.

c. In the New Action page, click **Select**.



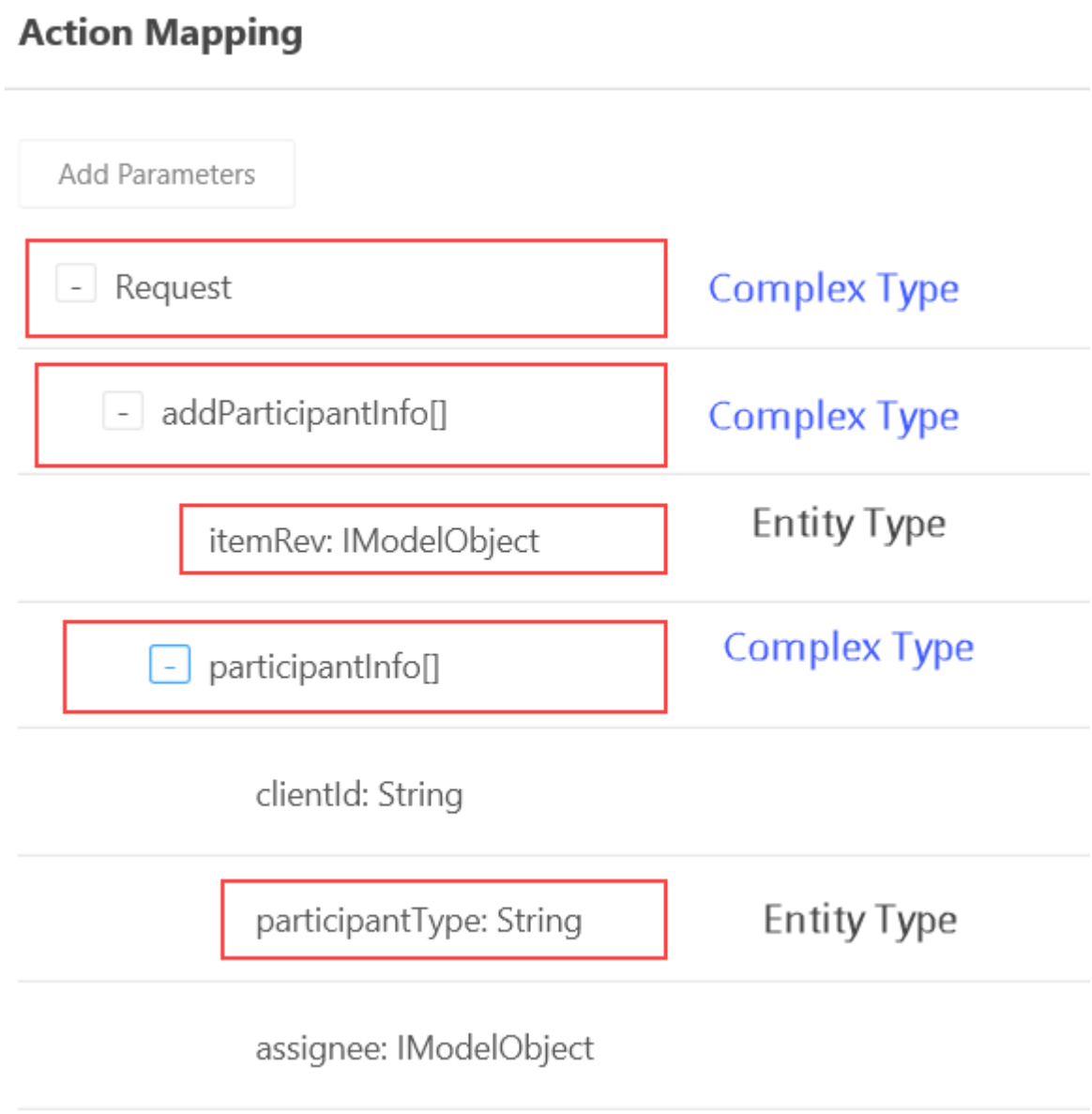
d. In the **New Action** dialog box, search for the Teamcenter SOA operation, select the operation, and click **Add**.



e. On the New Action page, update the name of the OData action if required.

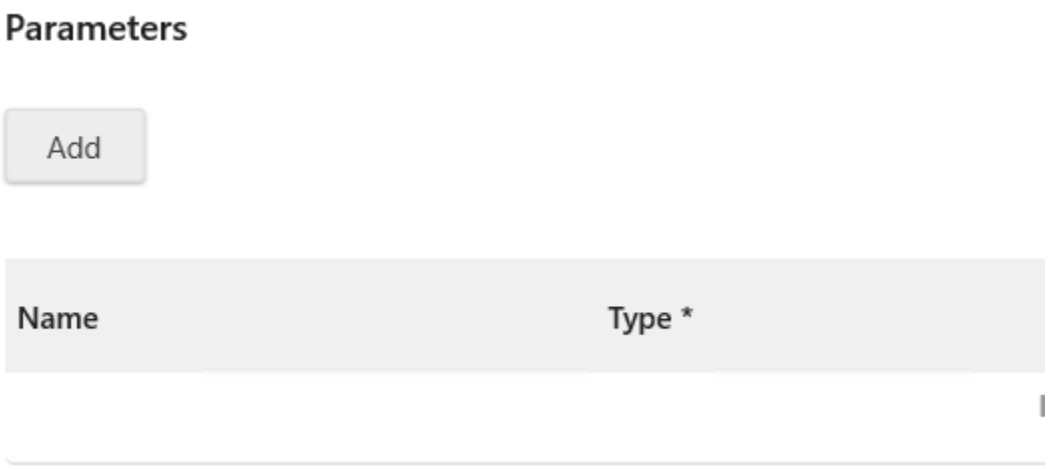
2. **Add the Request and Return parameters to the action**

After selecting the Teamcenter SOA operation you want to expose as an OData action, you must select the parameters to be added to the action. That is, you must specify the inputs and outputs of the actions corresponding to the inputs and outputs available for the Teamcenter SOA operation. You can specify the input in the **Request** type and the output in the **Return** type. When you select a leaf node, a parameter of the type primitive or entity is created. When you select a non-leaf node, a parameter of the type complex is created.

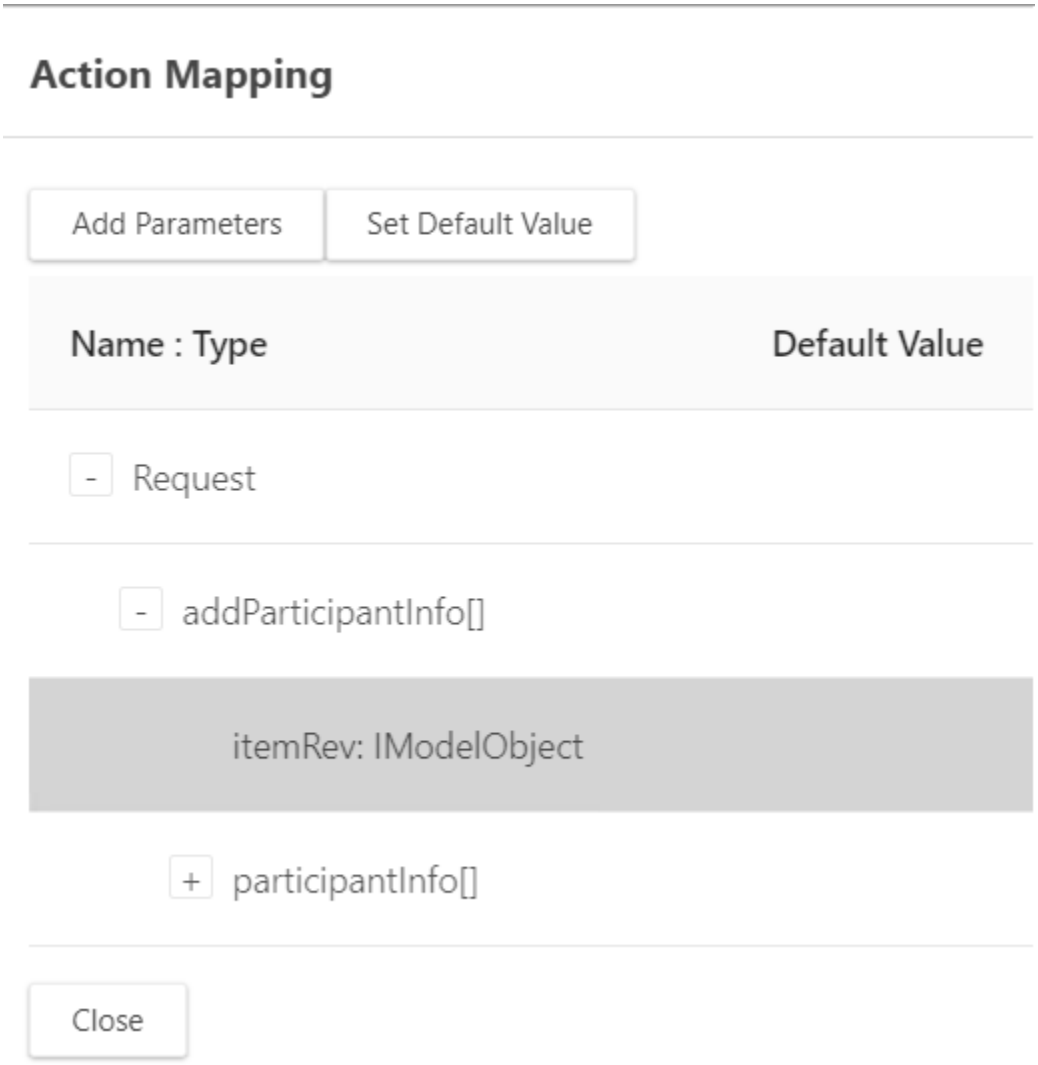


You can add parameters as follows:

a. A. In the **Parameters** section, click **Add**.

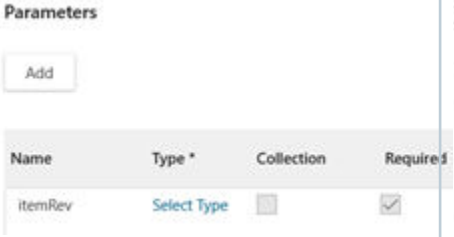


B. In the **Action Mapping** dialog box, select the request type, and click **Add Parameters** or double-click the parameter.



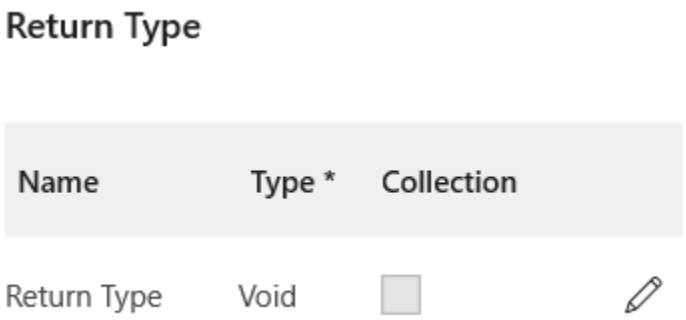
If you select a primitive type, you can also specify the default value for that type by clicking **Set Default Value**.

You can either select a leaf node or an intermediate node from the request tree.A parameter of the type based on your selection is created.

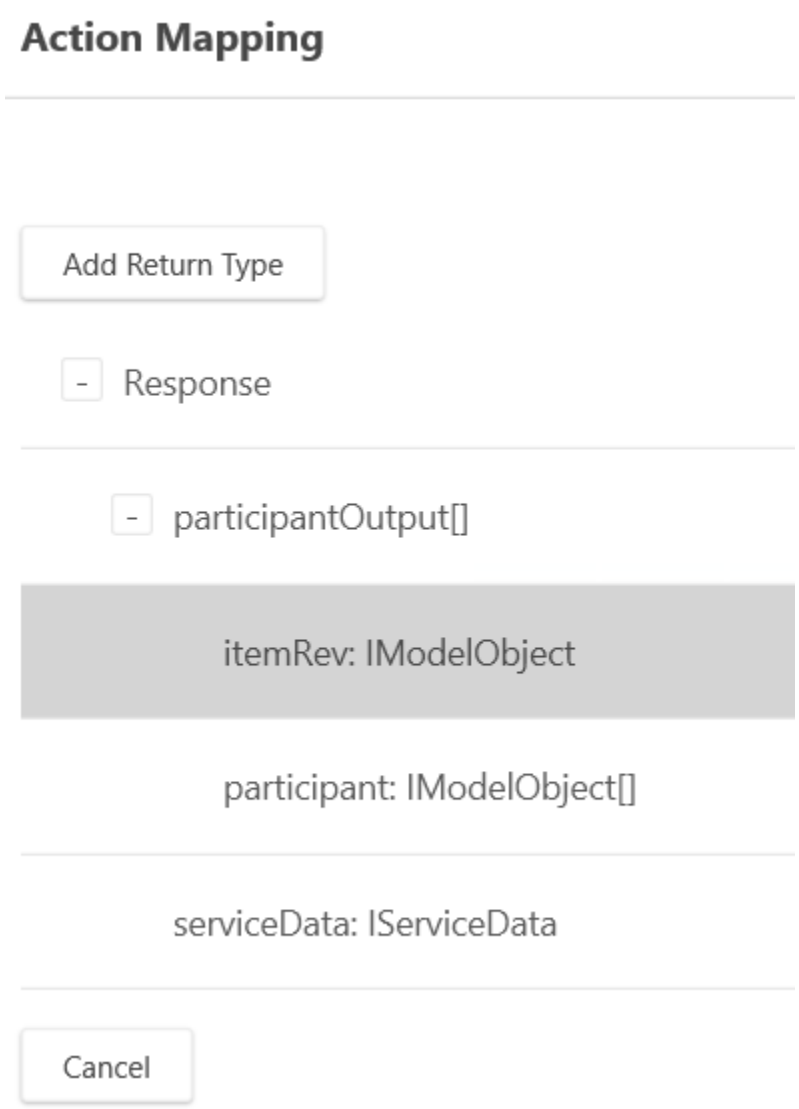
Leaf node selected	Intermediate node selected
	

b. To add a return type:

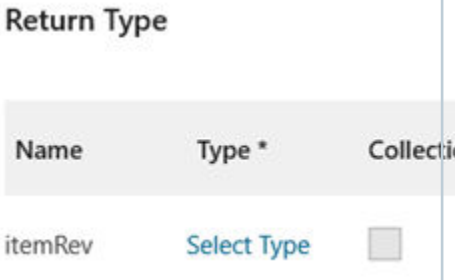
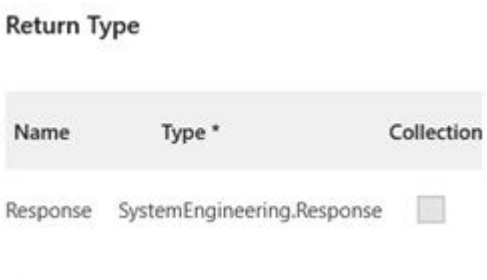
A. In the **Return Type** section, click **Edit** .



B. In the **Action Mapping** dialog box, select the return type and click **Add Return Type**.



You can either select a leaf node or an intermediate node from the tree. Depending on what you select, a parameter of that type is created.

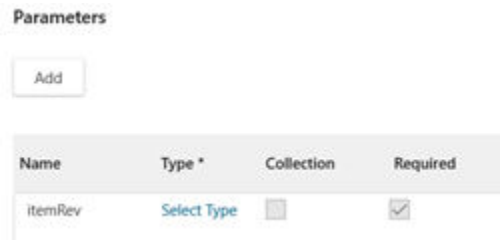
Leaf node selected	Intermediate node selected
	

3. **Map the parameters to entities in the OData service**

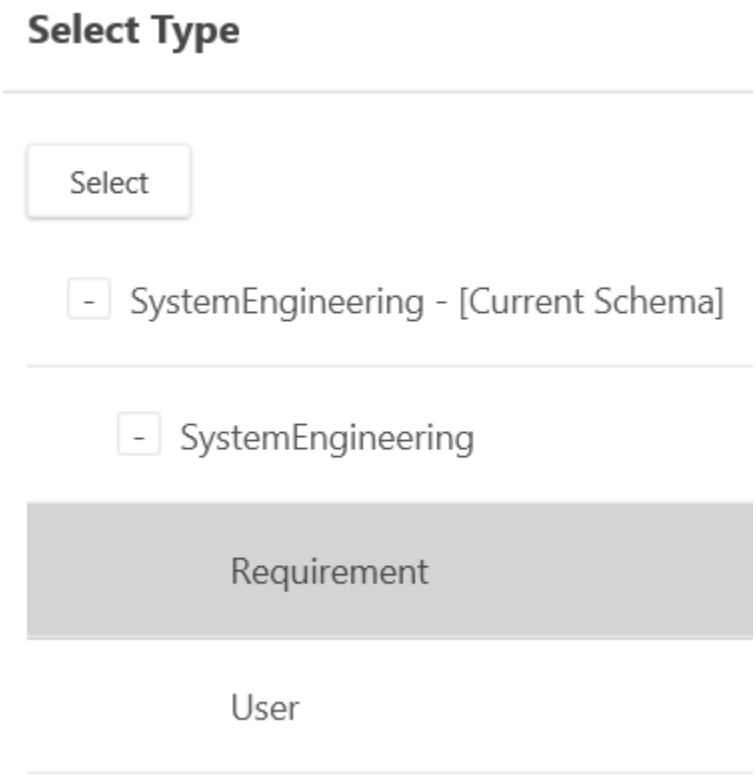
   After you add request and return parameters, you must map these parameters to the OData entities. Depending on whether your parameters are individual entities or complex types, the mapping table differs.

   a. To map a request type, when your parameters are individual entities:

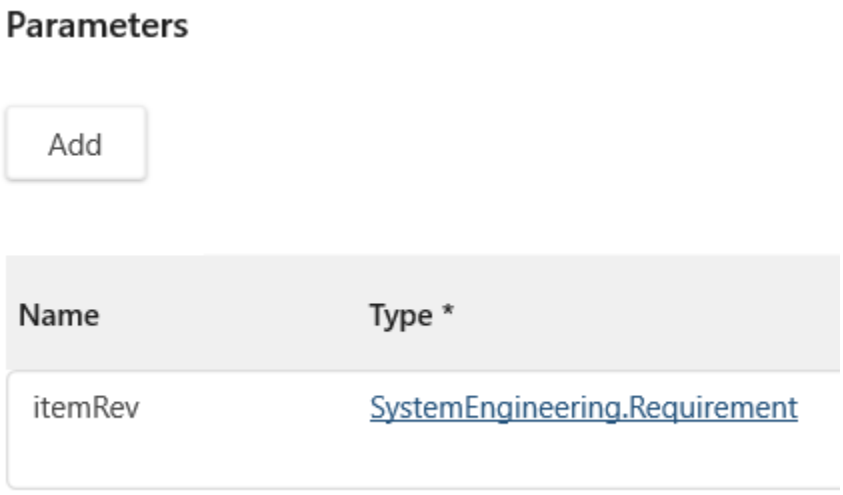
      A. Under the **Parameters** section, click **Select Type**.



      B. In the **Select Type** dialog box, navigate to the business object that you want to map to the entity, select it, and click **Select**.

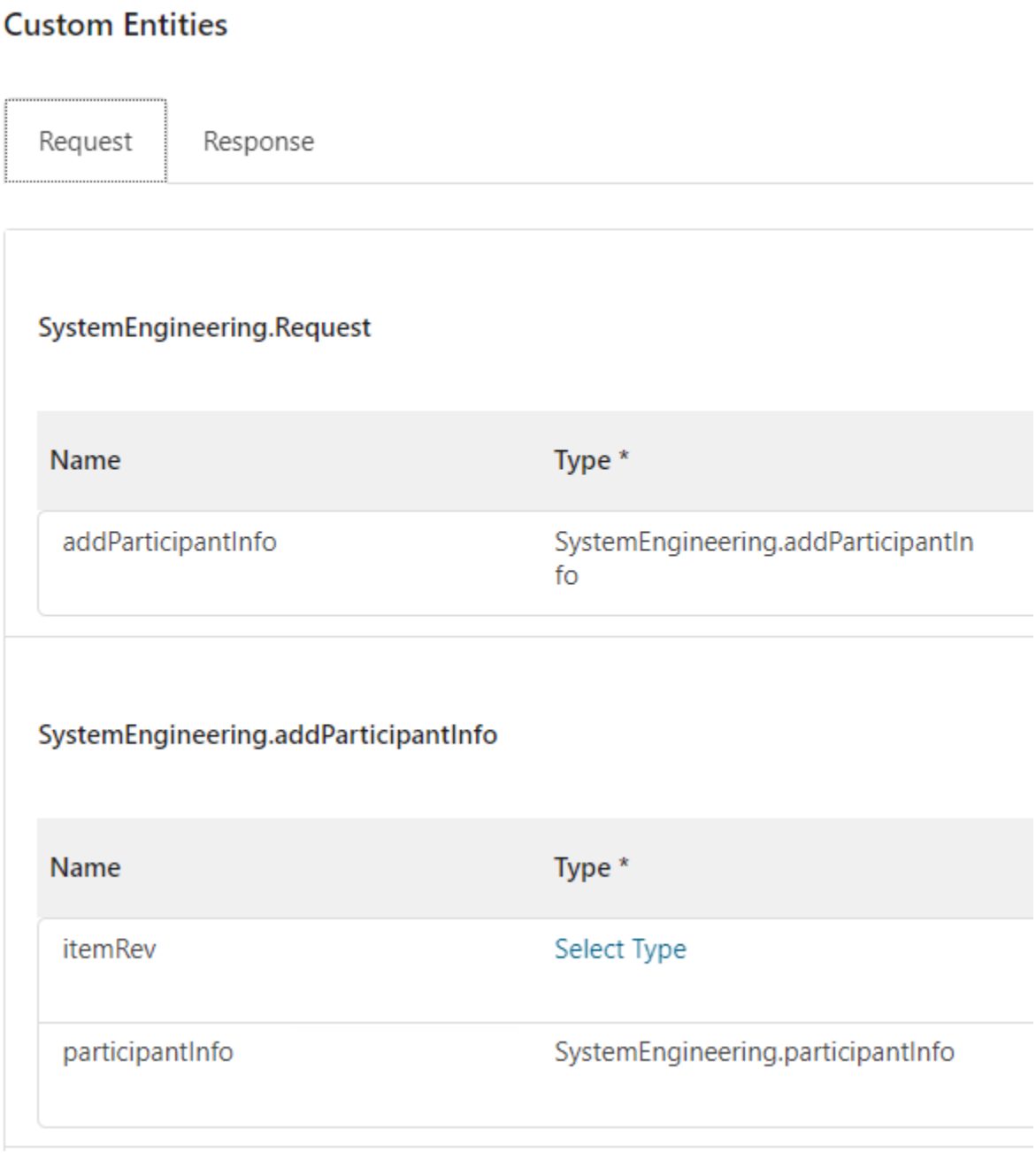


      The **Parameters** section now shows your mapped entities.

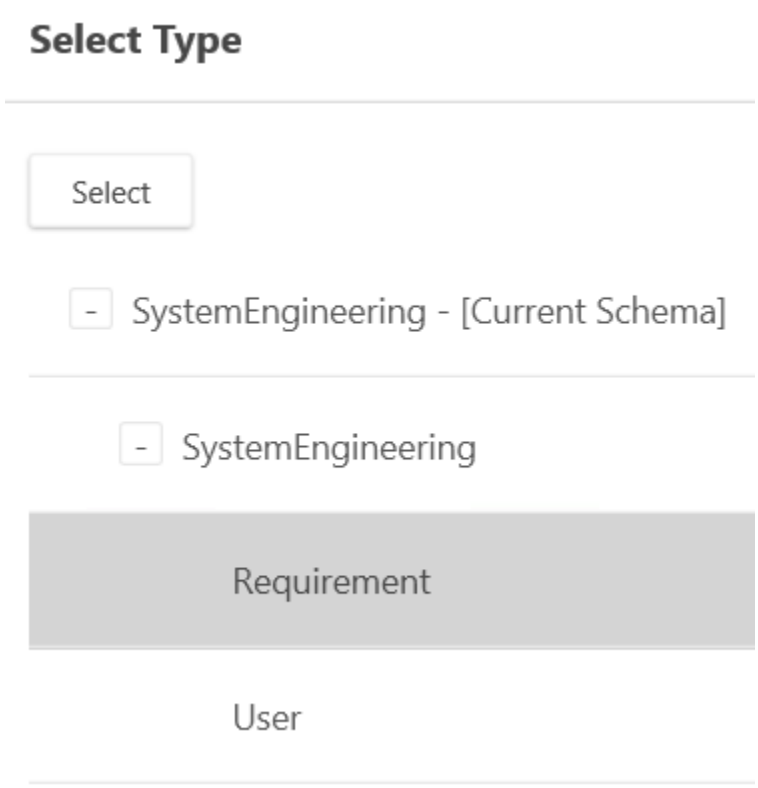


b. To map a request type, when your parameters are complex types:

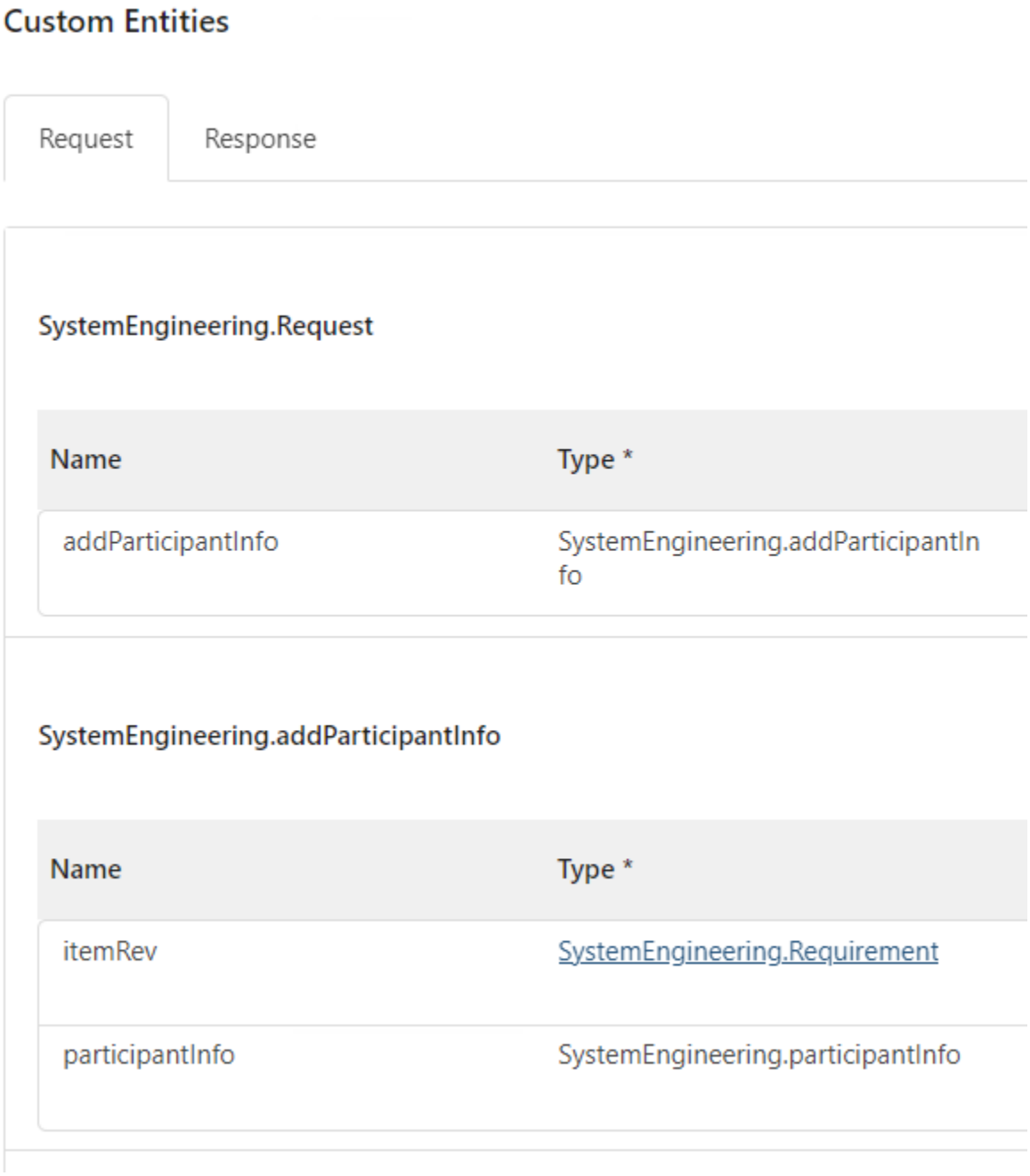
   A. Under the **Custom Entities** section, navigate to the business object where mapping is required, and click **Select Type**.



B. In the **Select Type** dialog box, navigate to the business object that you want to map to the entity, select it, and click **Select**.

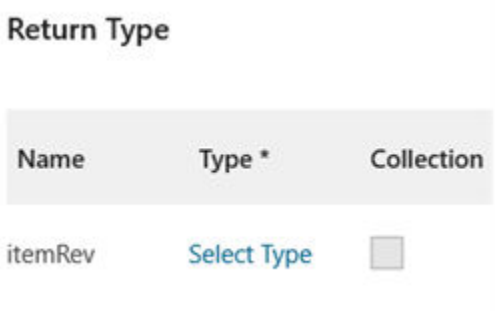


The mapping is now seen in the **Custom Entities** section.

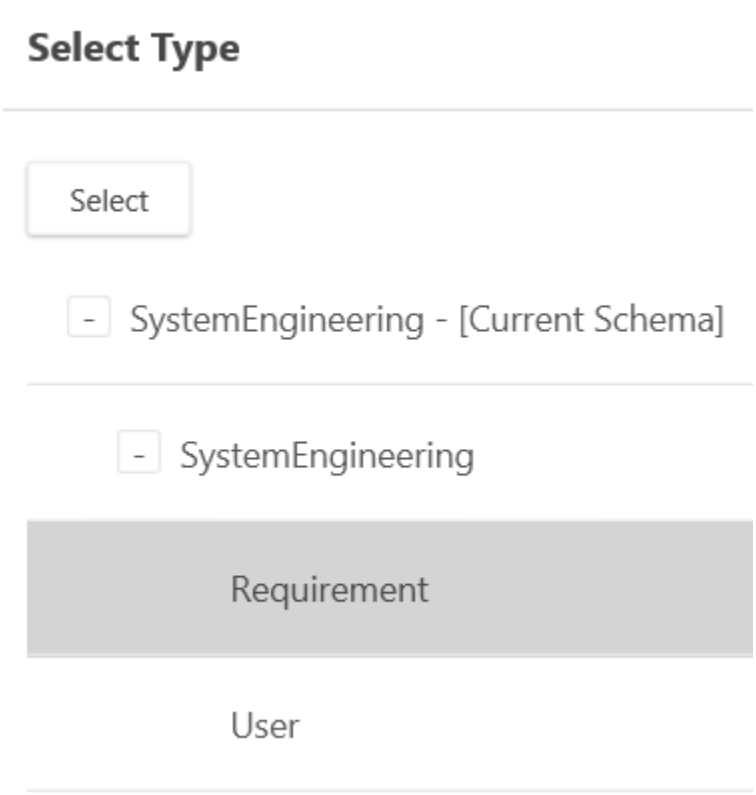


c. To map a return type, when your parameters are individual entities:

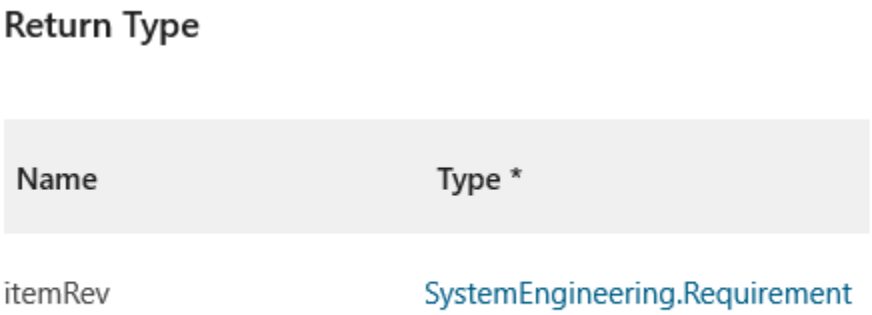
   A. Under the **Return Type** section, click **Select Type**.



B. In the **Select Type** dialog box, navigate to the business object that you want to map to the entity, select it, and click **Select**.

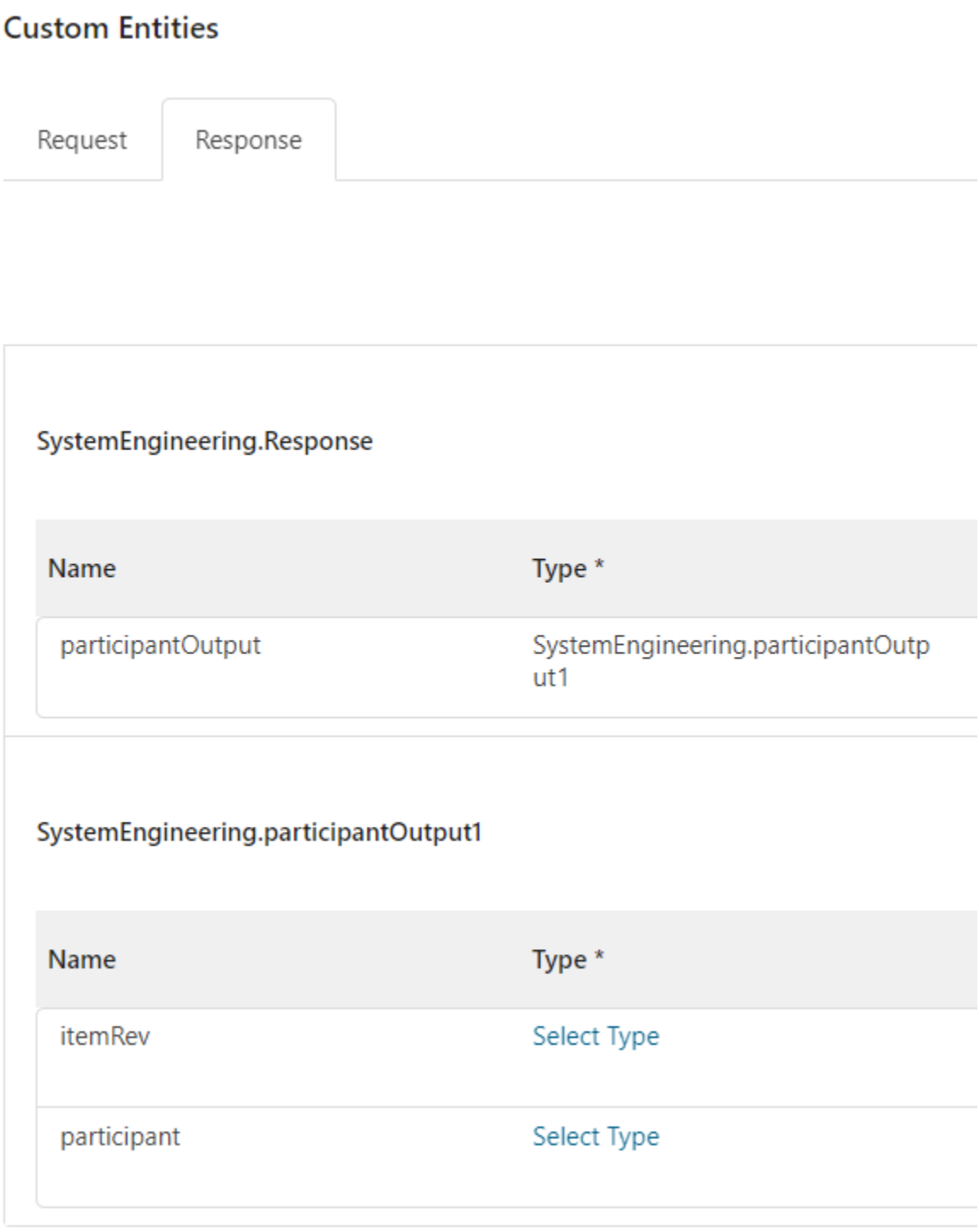


The **Parameters** section now shows your mapped entities.

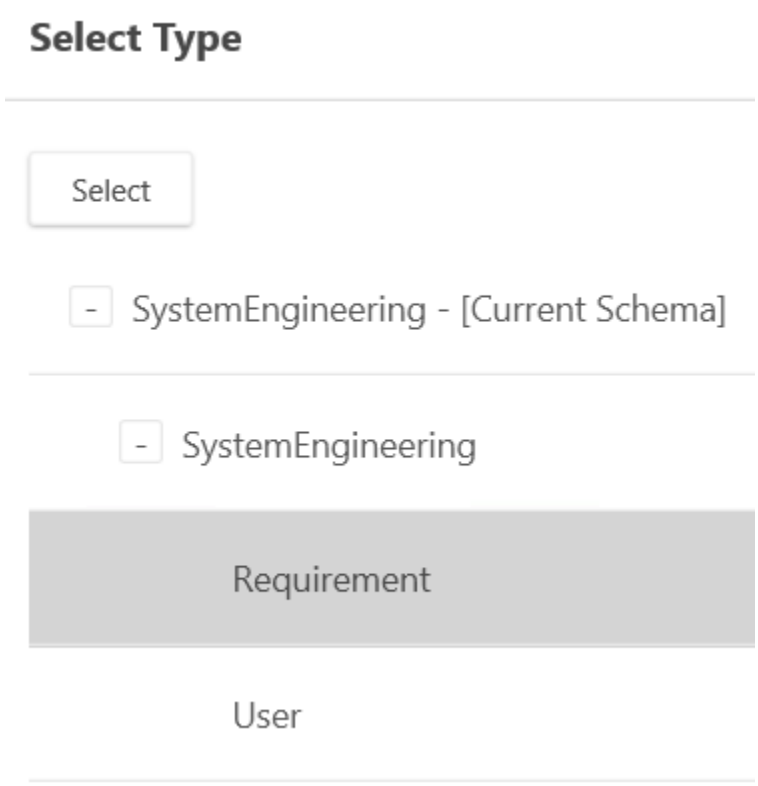


d. To map a return type, when your parameters are complex types:

   A. Under the **Custom Entities** section, click the **Response** tab and navigate to the business object where mapping is required, and click **Select Type**.



B. In the **Select Type** dialog box, navigate to the business object that you want to map to the entity, select it, and click **Select**.



You can view the mapping is now seen in the **Custom Entities** section.

Custom Entities

Request | Response

SystemEngineering.Response

Name	Type *
participantOutput	SystemEngineering.participantOutput1

SystemEngineering.participantOutput1

Name	Type *
itemRev	SystemEngineering.Requirement
participant	Select Type

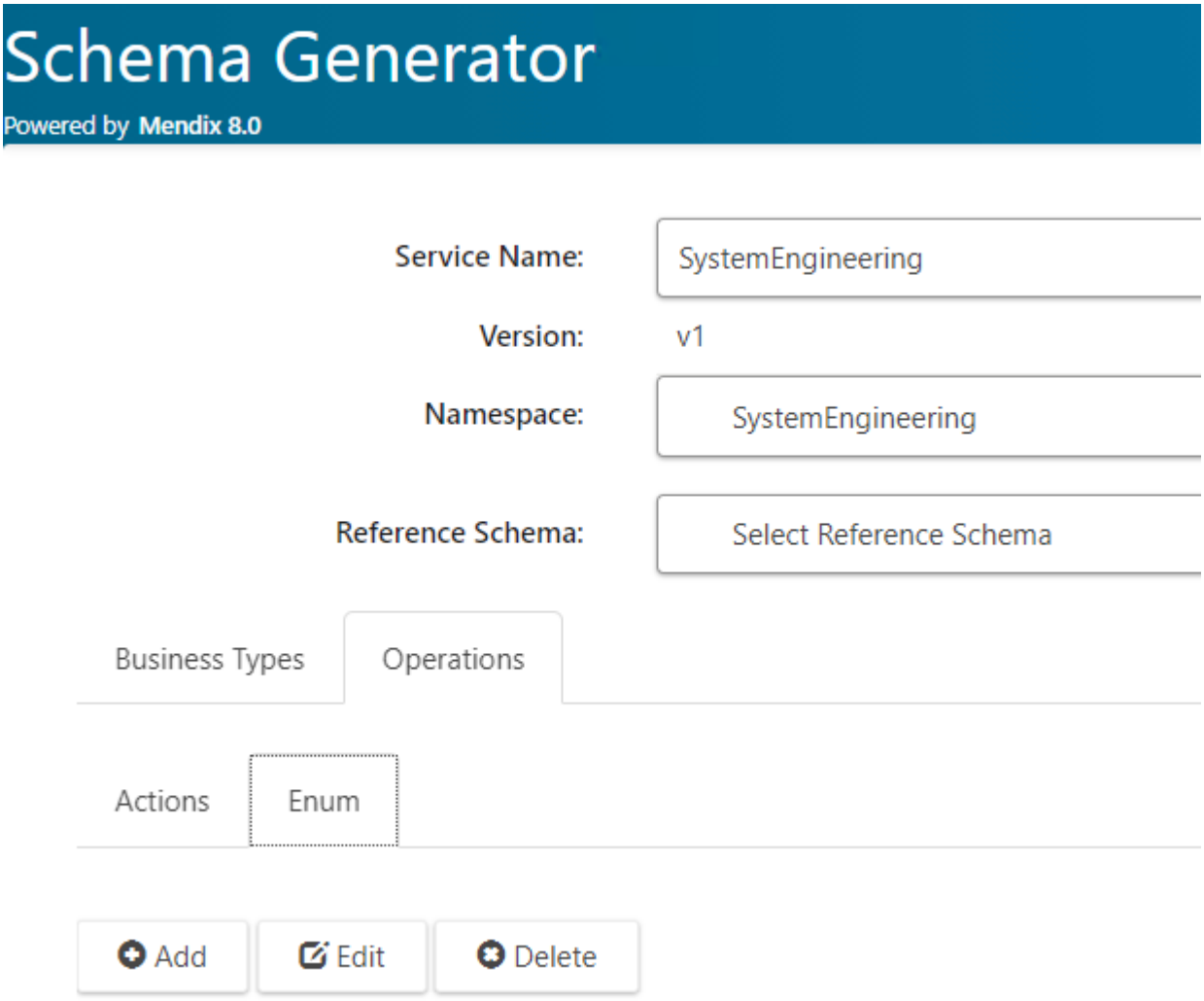
Create enums to map the input values of the Teamcenter SOA operation

If the Teamcenter SOA operation input parameter has enum values, and if you have chosen such an operation in the previous step, enum types are automatically generated in the background and associated with the input parameters.

> Note:
>
> Automatically generated enum types cannot be edited or removed.

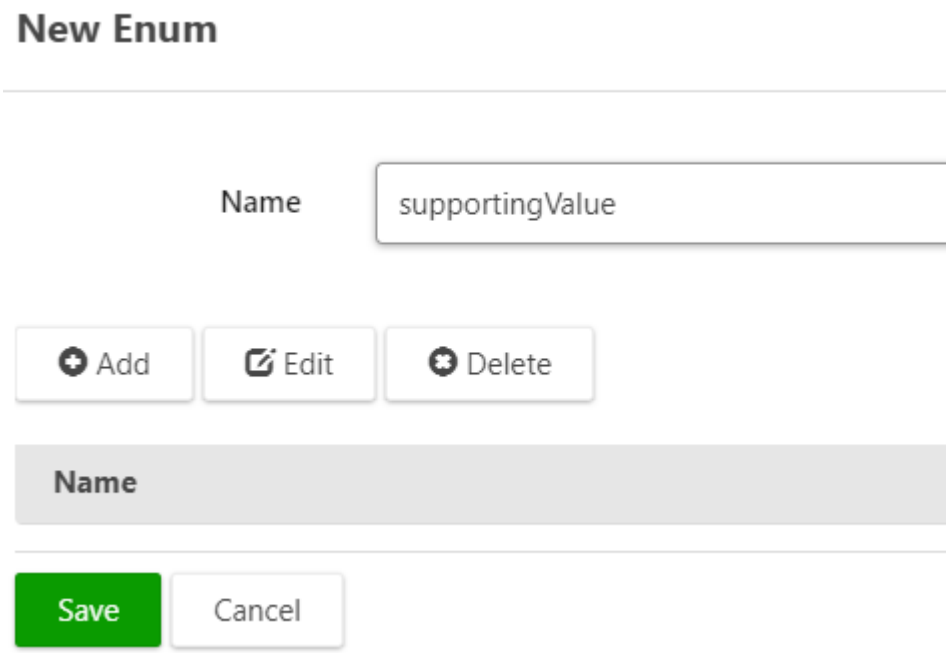
Alternatively, if the Teamcenter SOA operation input has provided a list of values in the description, and if you have chosen such an operation in the previous step, you can explicitly create an enum type with the required values as follows:

1. On the Schema Generator page, click the **Operations** tab and then click the **Enum** tab.

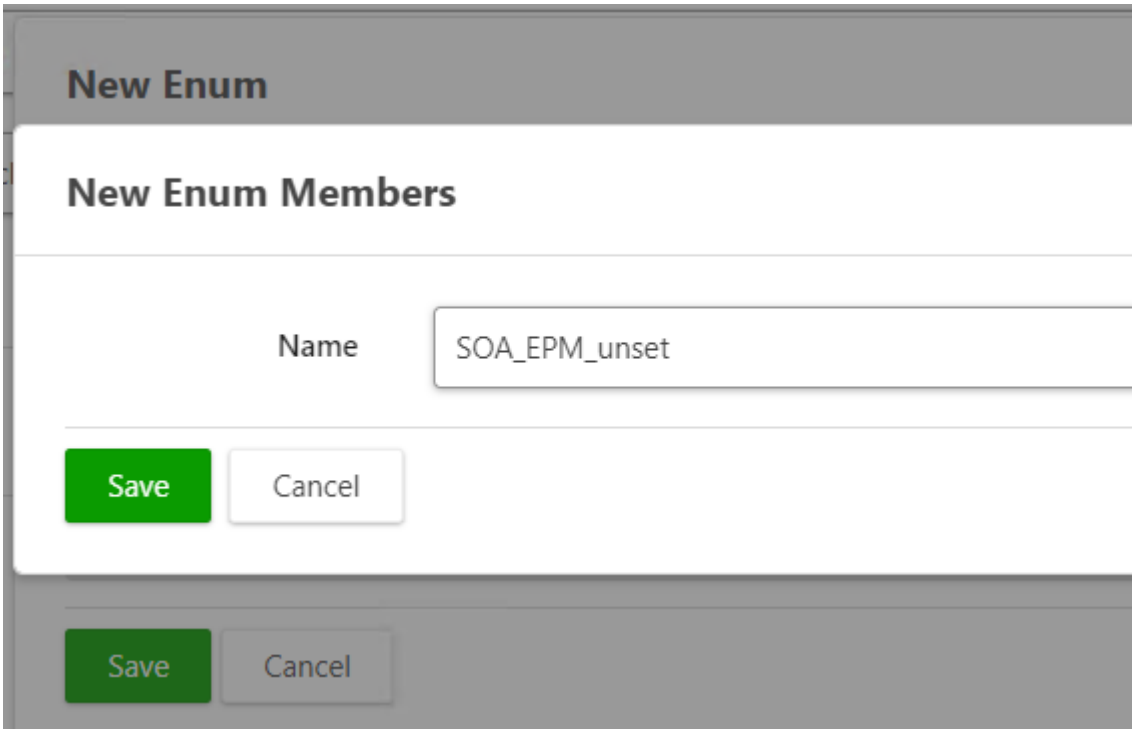


2. Click **Add**.

3. In the **New Enum** dialog box, type the name of the input value and click **Add**.

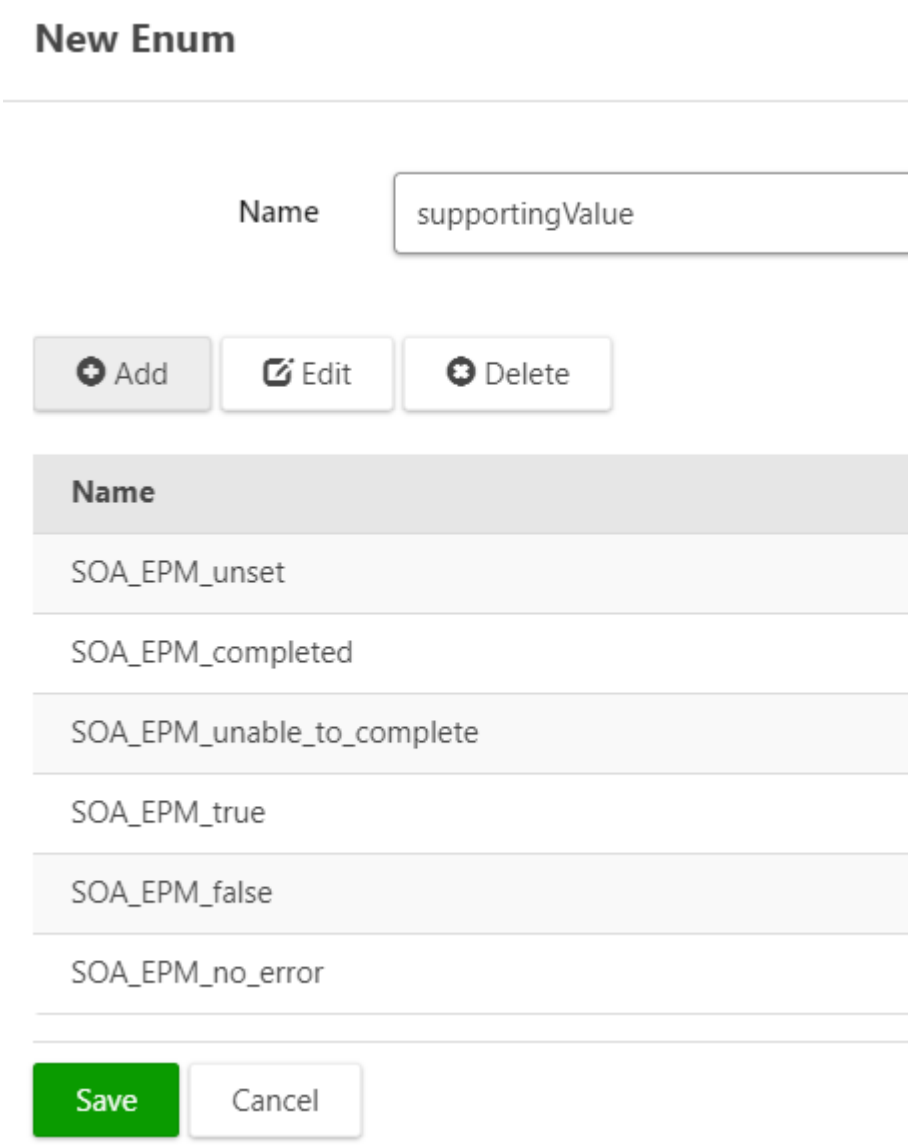


4. In the **New Enum Members** dialog box, type a name for the enum value and click **Save**.



Similarly, add more enum values.

5. In the **New Enum** dialog box, click **Save**.

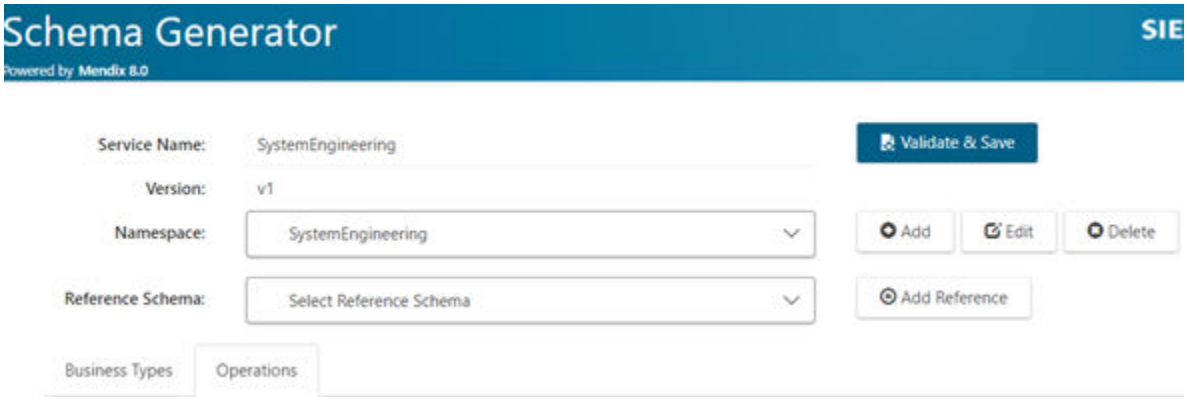


You can now assign the enum values you created to an action parameter.

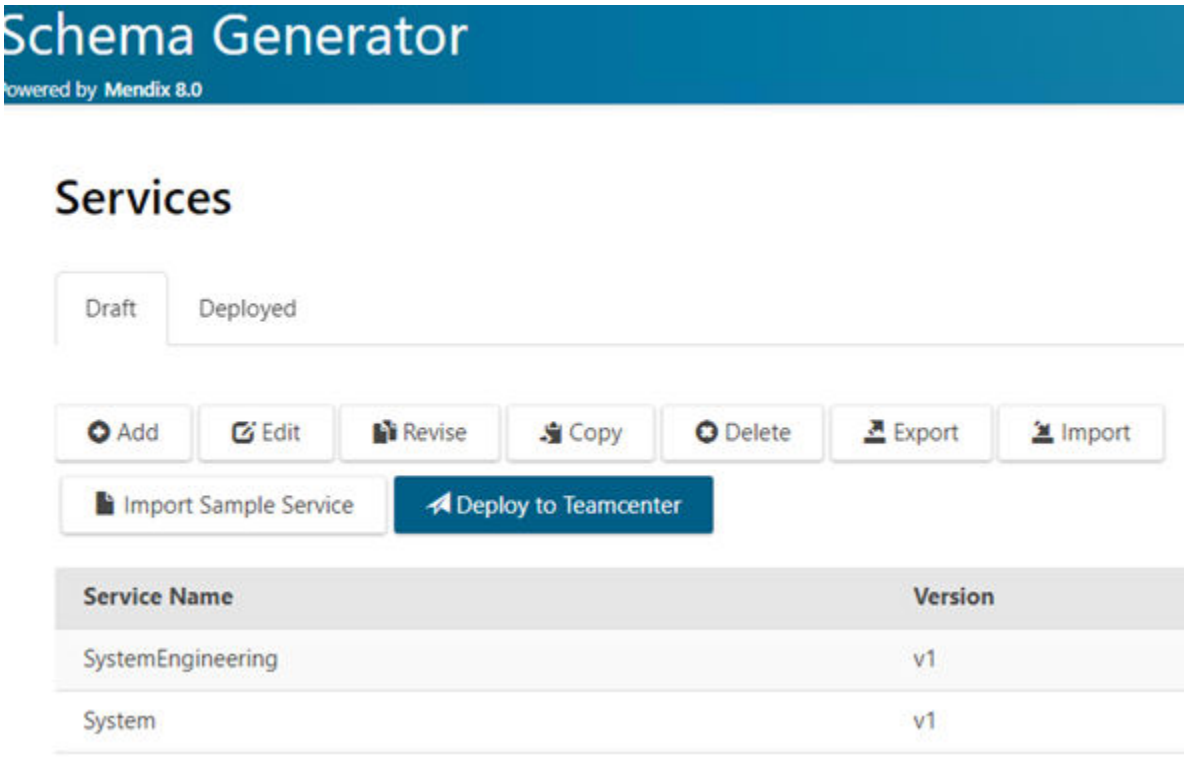
Step 4: Generate the service

After adding objects, properties, and operations, you can generate the service. To generate the service:

- Click **Validate & Save**.



The new service appears in the **Draft** tab of the Services page.

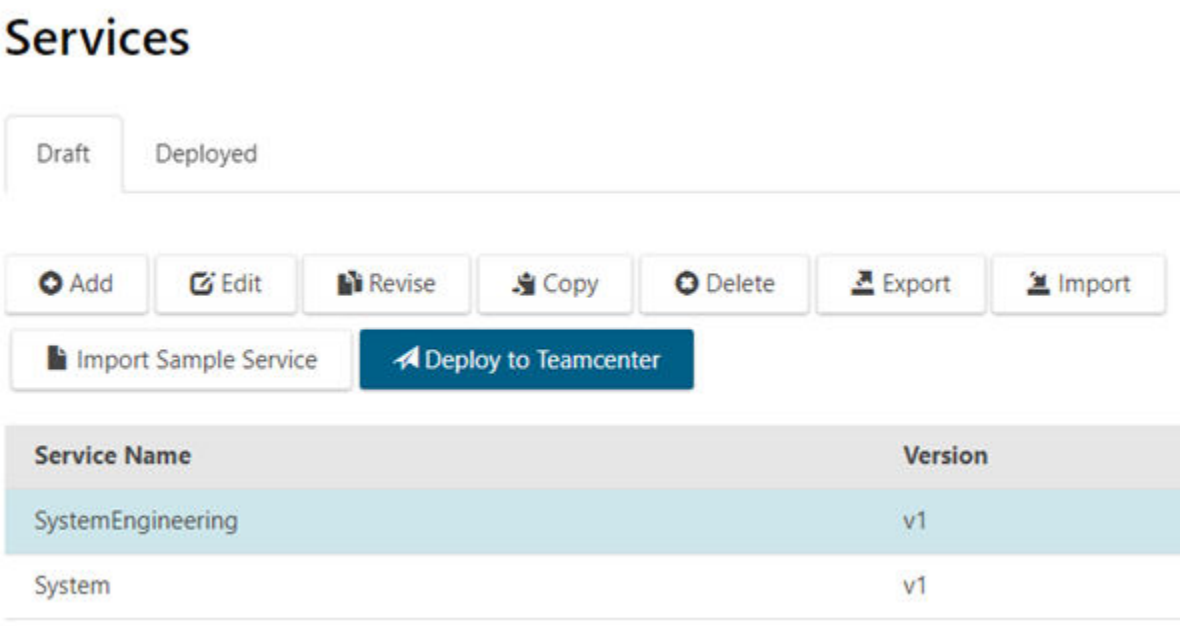


After generating the service, you must deploy the service to Teamcenter.

Step 5: Deploy a service to Teamcenter

To deploy the service to Teamcenter:

- In the **Services** page, choose a service and click **Deploy to Teamcenter**.



If this is a new service, version 1 of the service is deployed.

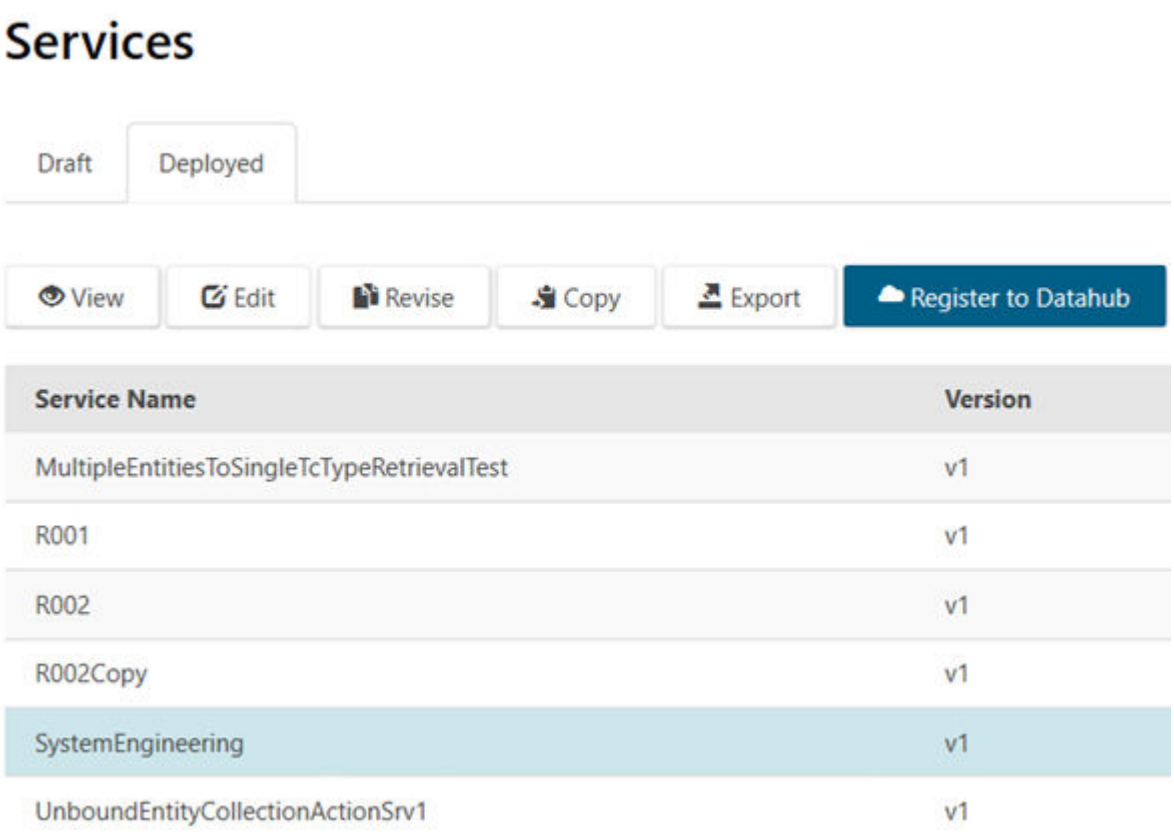
The service now appears in the **Deployed** tab.

After deploying the service to Teamcenter, you can deploy it to the Mendix Data Hub.

Step 6: Register a service to Mendix Data Hub

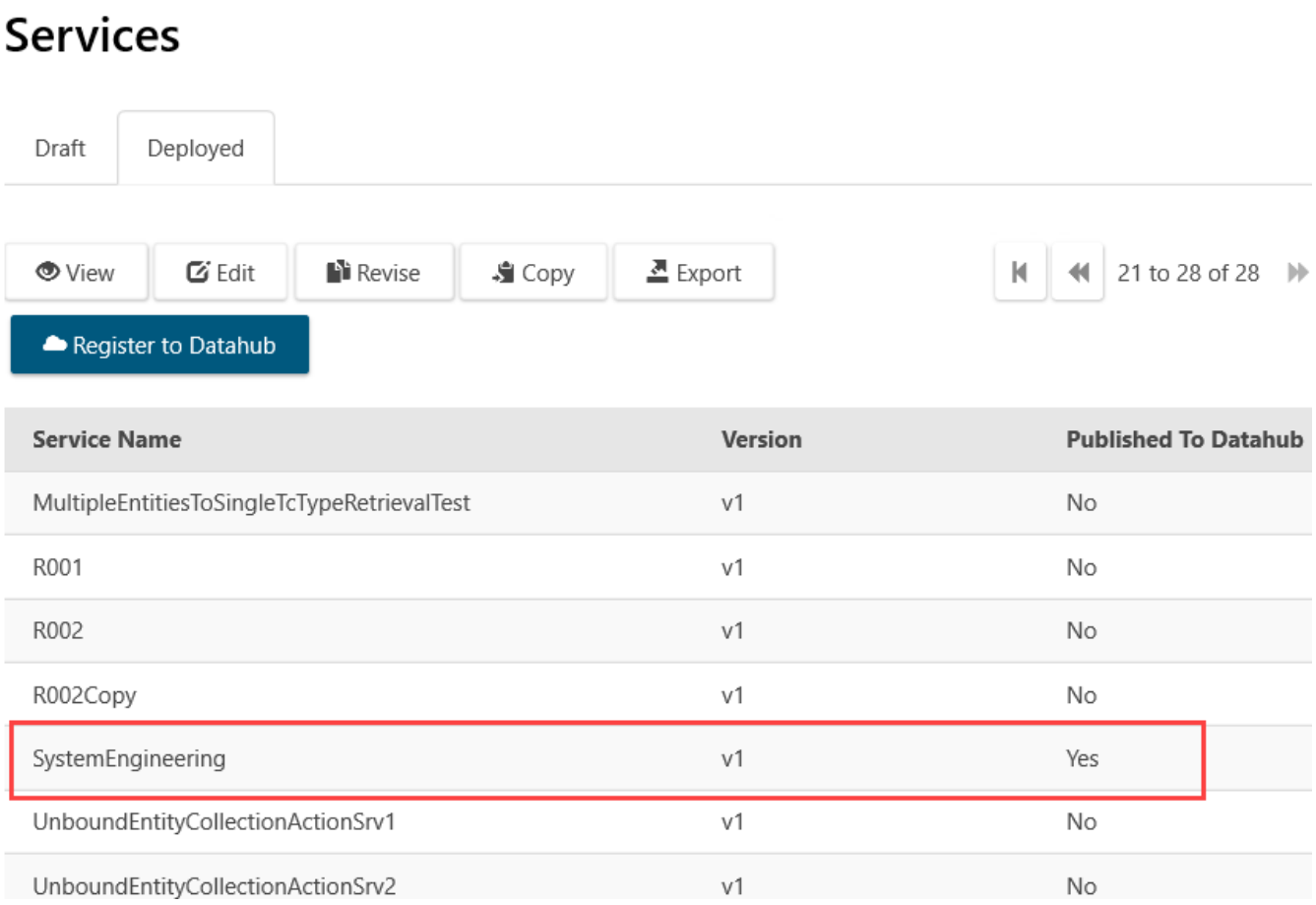
Only services that are deployed to Teamcenter can be registered to Mendix Data Hub.

1. To register a service to Mendix Data Hub, from the Deployed tab of the **Services** page, choose a service and click **Register to Datahub**.



2. If you have not logged on to Mendix Data Hub, a login dialog box appears. Log on to Mendix Data Hub.

The service is published to the Mendix Data Hub.



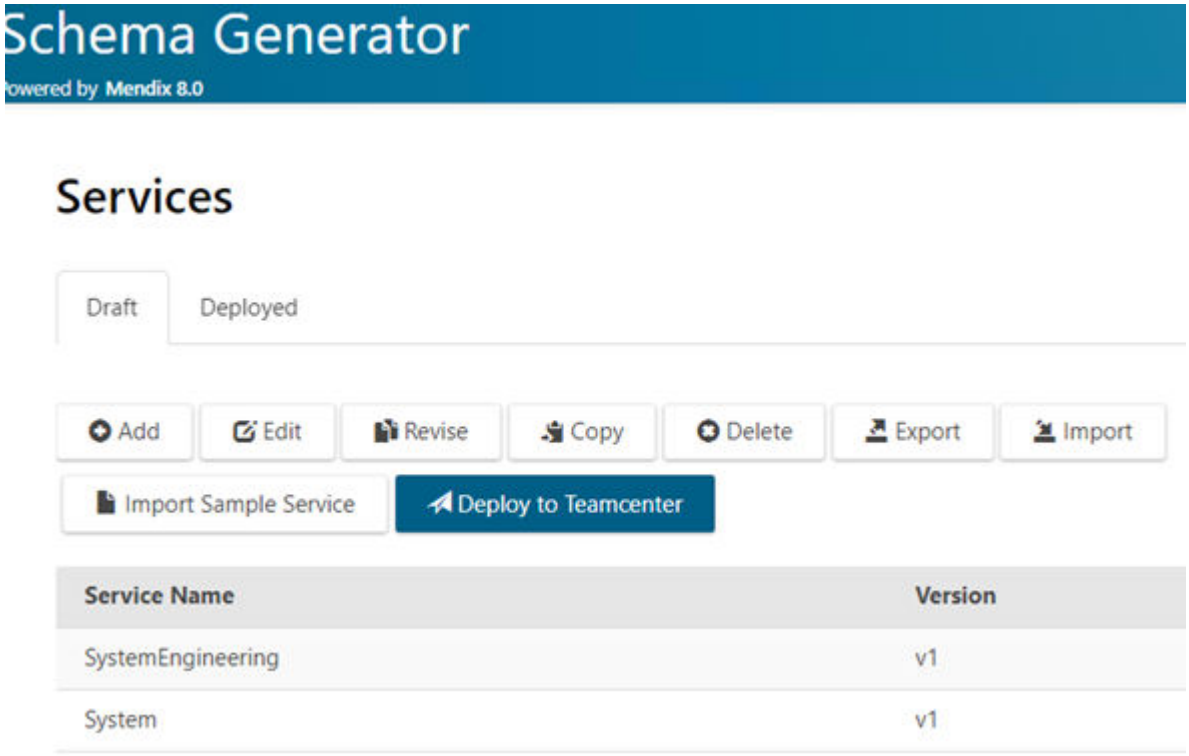
Create a new service using existing services

You can create a new service using an existing service in the following ways:

- Add an existing service as a reference in the new service you are creating.
- Create a copy of an existing service and make the changes you need.

Add an existing service as a reference in the new service

1. Click **Home** to get to the Services page.
2. From the **Draft** tab click **Add**.



3. In the **New Service** dialog box, specify the name of the new service and click **Create**.

4. Click **Add Reference** next to the **Namespace** list.

5. In the **Add Reference** dialog box, select a service from either the **Draft** or **Deployed** tab and click **Add**.

6. Select objects and properties and create a namespace and then generate the service.

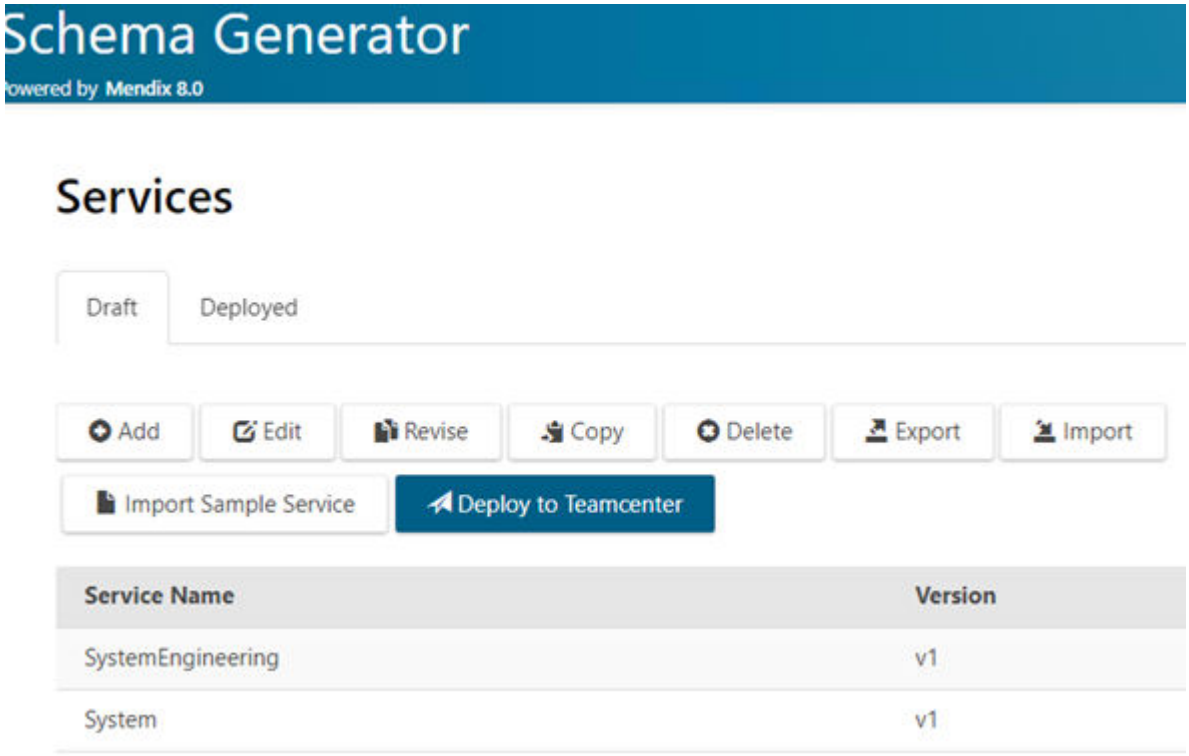
   See Create a new service for more information.

The namespaces from the selected service is added to the service you are creating. You cannot edit the namespaces of the referenced service. You can, however, delete namespaces.

Create a copy of an existing service with modifications

1. Click **Home** to get to the Services page.

2. From the **Draft** or the **Deployed** tab, select a service and click **Copy**.



3. You can add new namespaces, modify the contents of existing namespaces, or make breakable changes.

Administering services

Edit a service

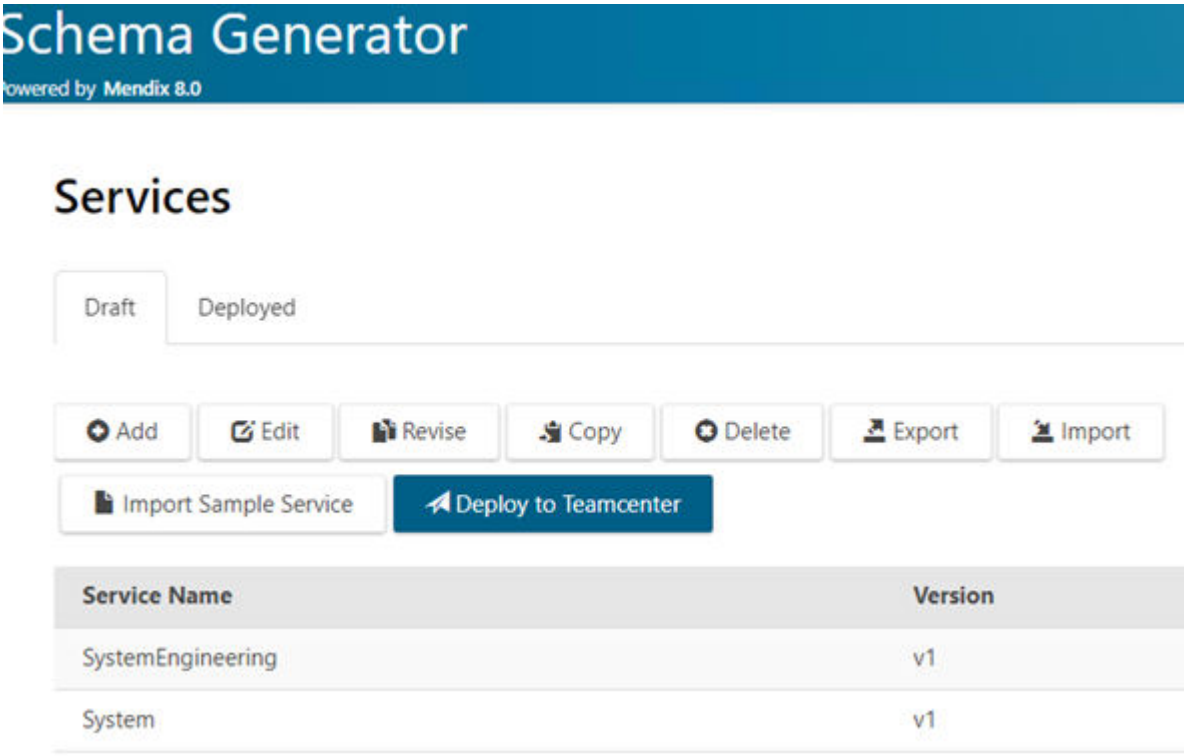
Editing a service allows you to update the objects and properties within your service. Editing allows you to make only non-breakable changes.

> Note:
>
> Actions in a service cannot be edited. To modify an action, you must first delete it and then create a new action.

To edit a service:

1. Click **Home** to get to the Services page.

2. From the **Draft** tab, select a service and click **Edit**.

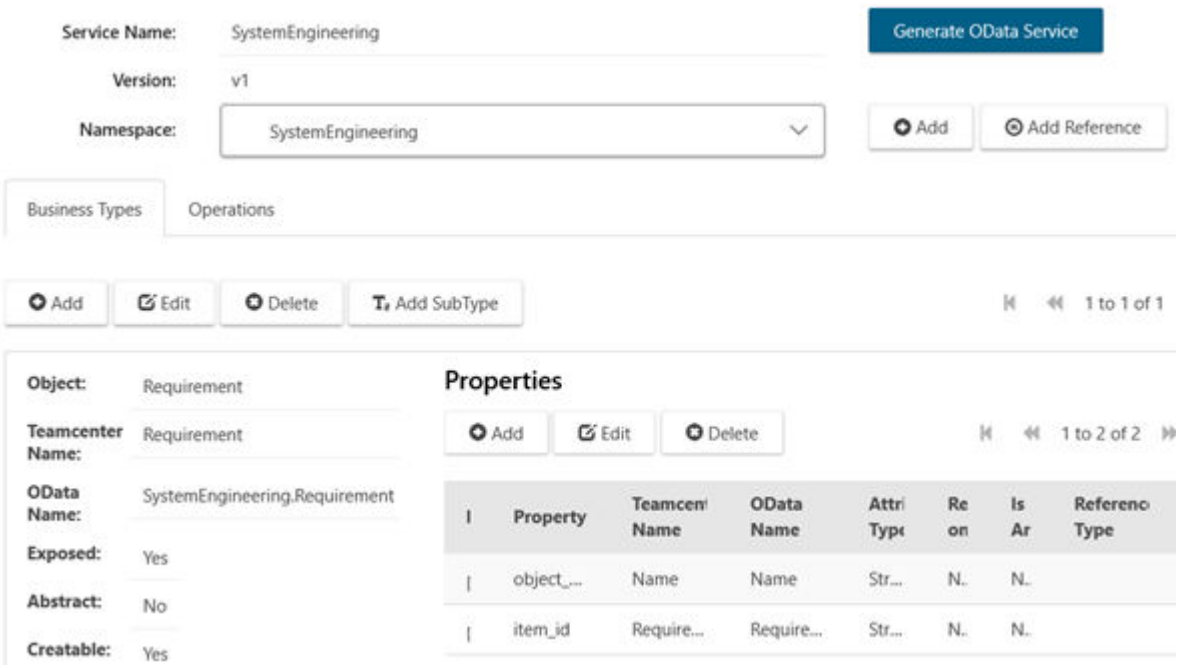


To edit a deployed service that is not available in the **Draft** tab, click the **Deployed** tab, select the required service, and click **Edit**.

3. Select the service and click **Edit**.

   You can:

   - Create new namespaces.
   - Add an external reference—Add data from other services to the service you are editing.
   - Add, edit, or delete new objects.
   - Create a subtype of an object.
   - Add, edit, or delete new properties.



Note:

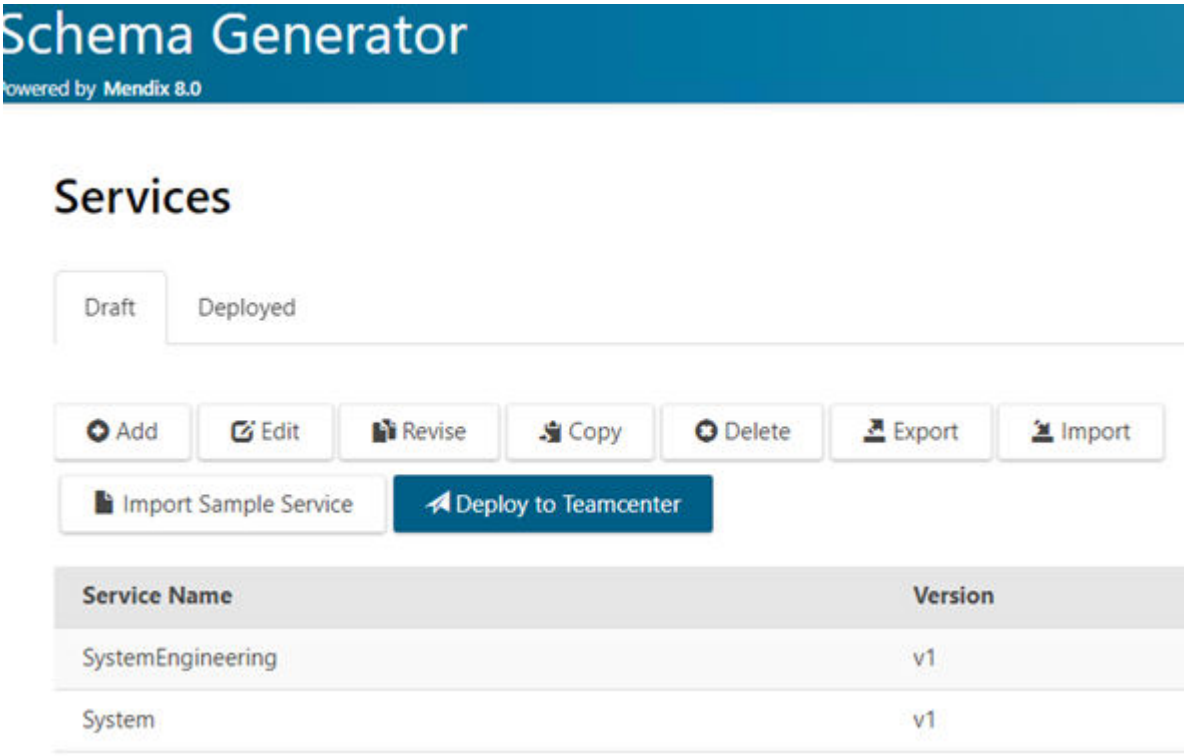
Only non-breakable changes are allowed if the service is deployed to Teamcenter. You cannot modify existing objects or properties.

Revise a service

When you want to make breakable changes, that is, modify existing objects and properties, to a deployed service, you must revise the service. To revise a service:

1. Click **Home** to get to the Services page.

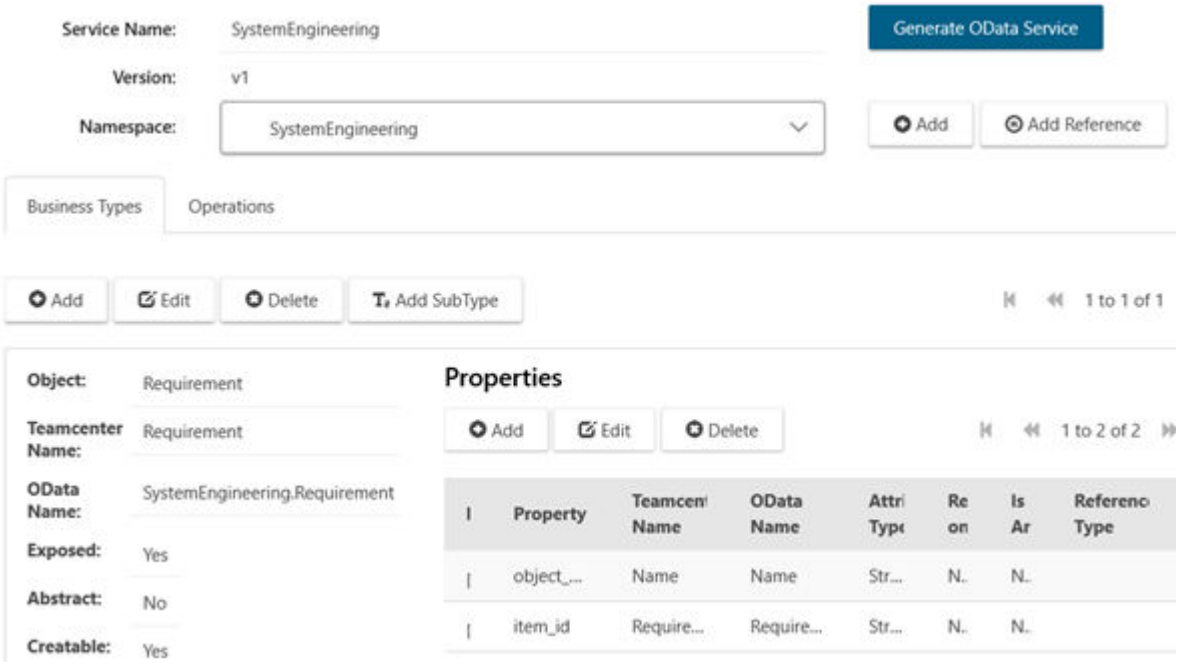
2. From the **Draft** or the **Deployed** tab, select a service and click **Revise**.



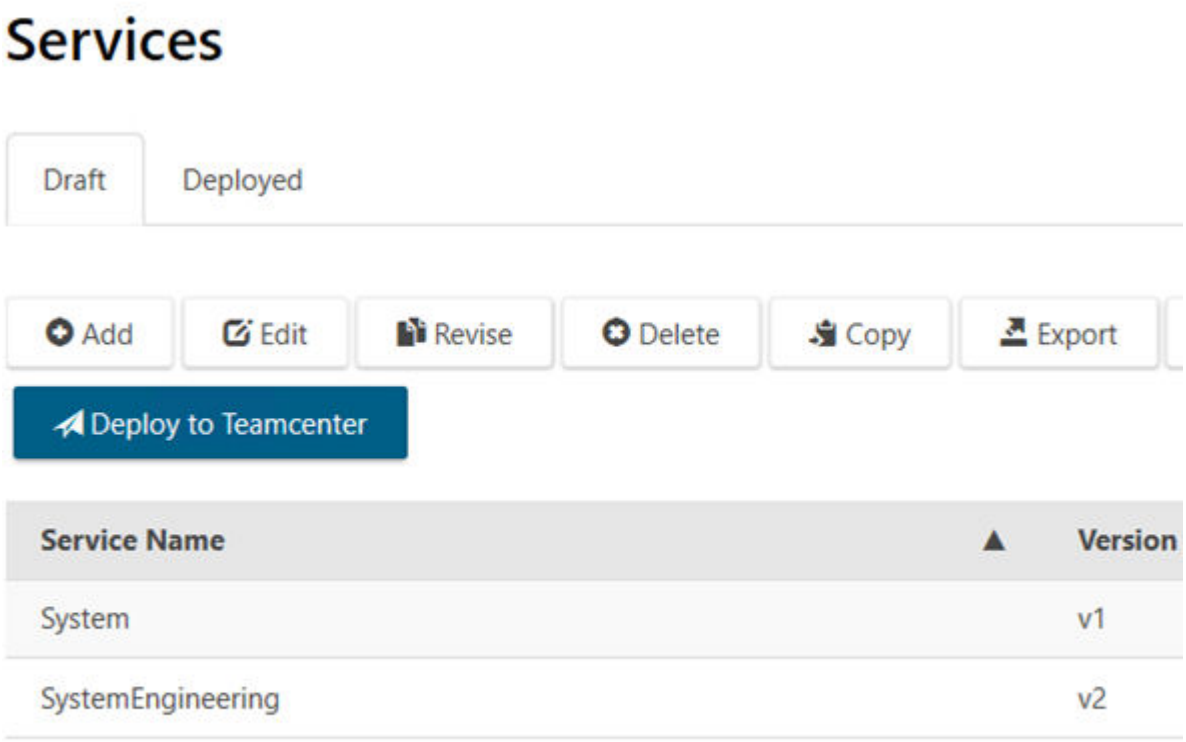
3. Select the service and click **Revise**.

4. Make the changes. You can:

   - Add an external reference—that is add data from other services to the service you are editing.
   - Add, edit, or delete objects.
   - Create subtype of objects.
   - Add, edit, or delete new properties.



5. After revising the service, click **Generate OData Service**. The new version of the service is displayed.



You can now deploy the new version to Teamcenter or Mendix Data Hub.

Delete a service

While you cannot delete a service that is deployed to Teamcenter, you can delete a service from your local repository as follows:

1. Click **Home** to get to the Services page.

2. From the **Draft** tab, select a service and click **Delete**.

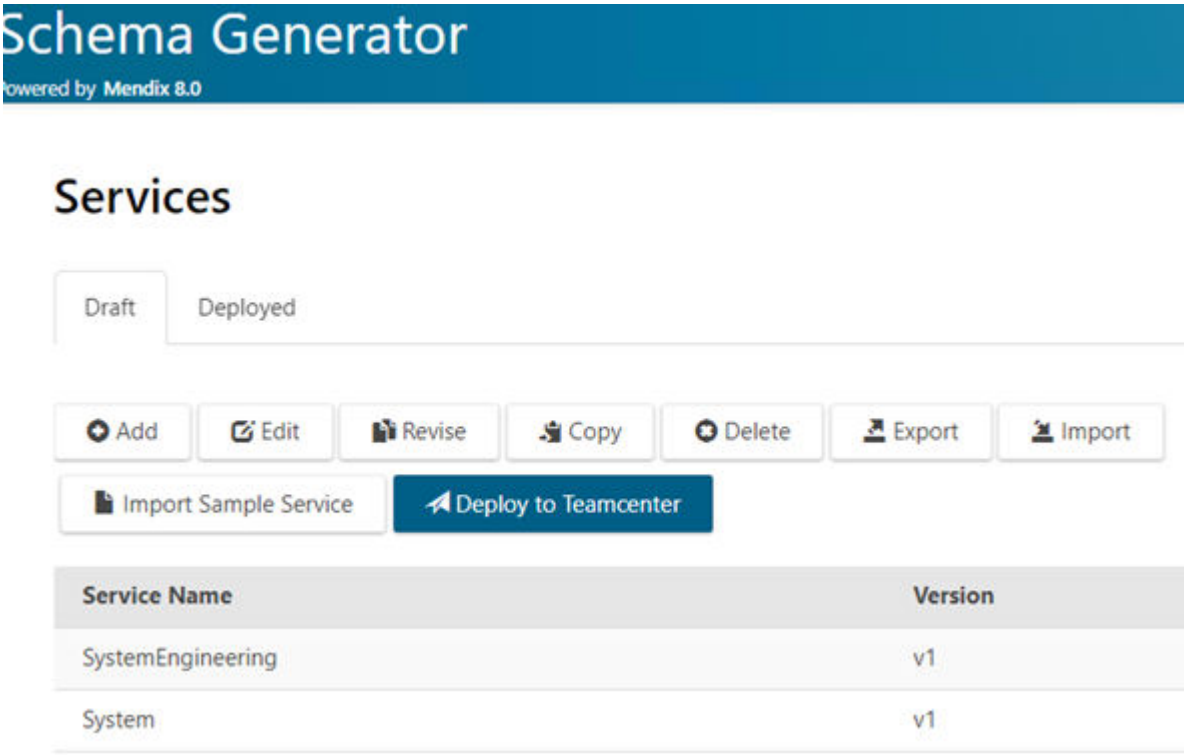
The service is deleted from your local services list.

Download a service

You can download a service to review the schema and mapping files. To download a service:

1. Click **Home** to access the Services page.

2. From the **Draft** or **Deployed** tab, select a service and click **Export**.



3. The service is downloaded as a ZIP file. Save the ZIP file and extract its contents.

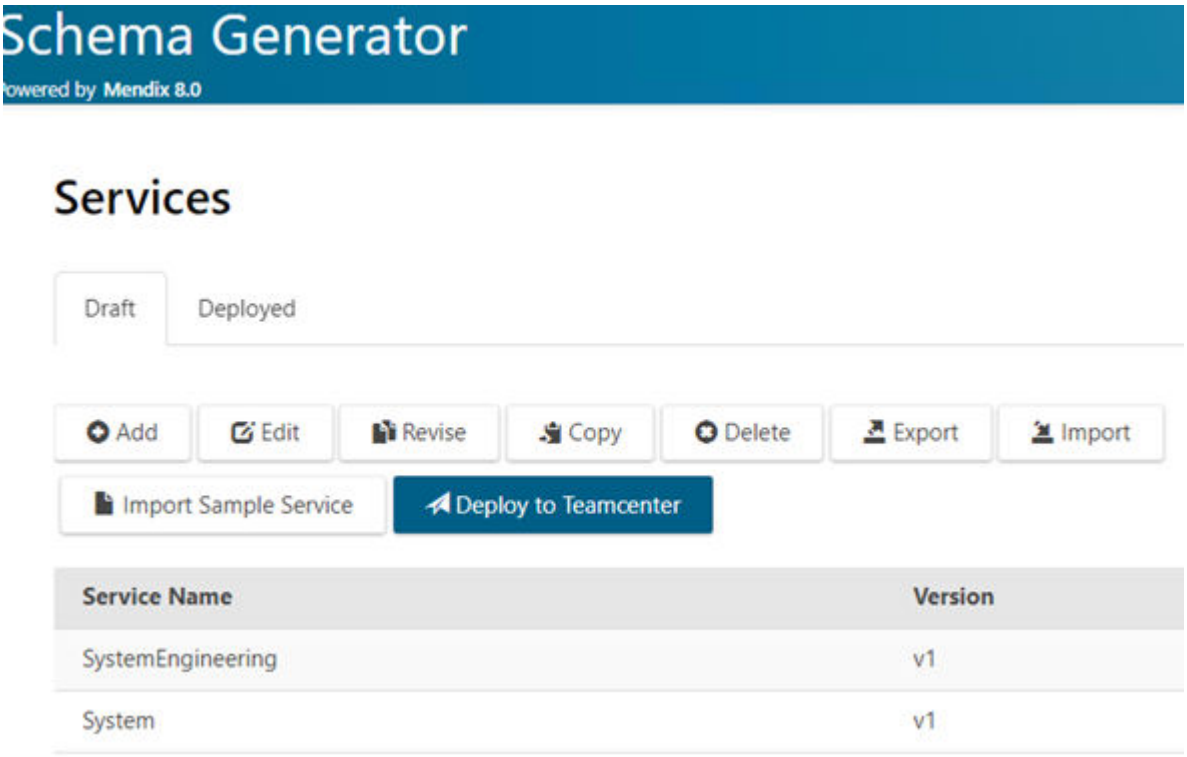
   The ZIP file contains the *TcMapping* JSON file that contains the mapping information.

Import a service

You can move a service from one Teamcenter environment to another by importing a service as follows:

1. Click **Home** to get to the Services page.

2. From the **Draft** tab, select a service and click **Import**.



3. In the **Import Service** dialog box, browse to the service ZIP file and click **Import**.

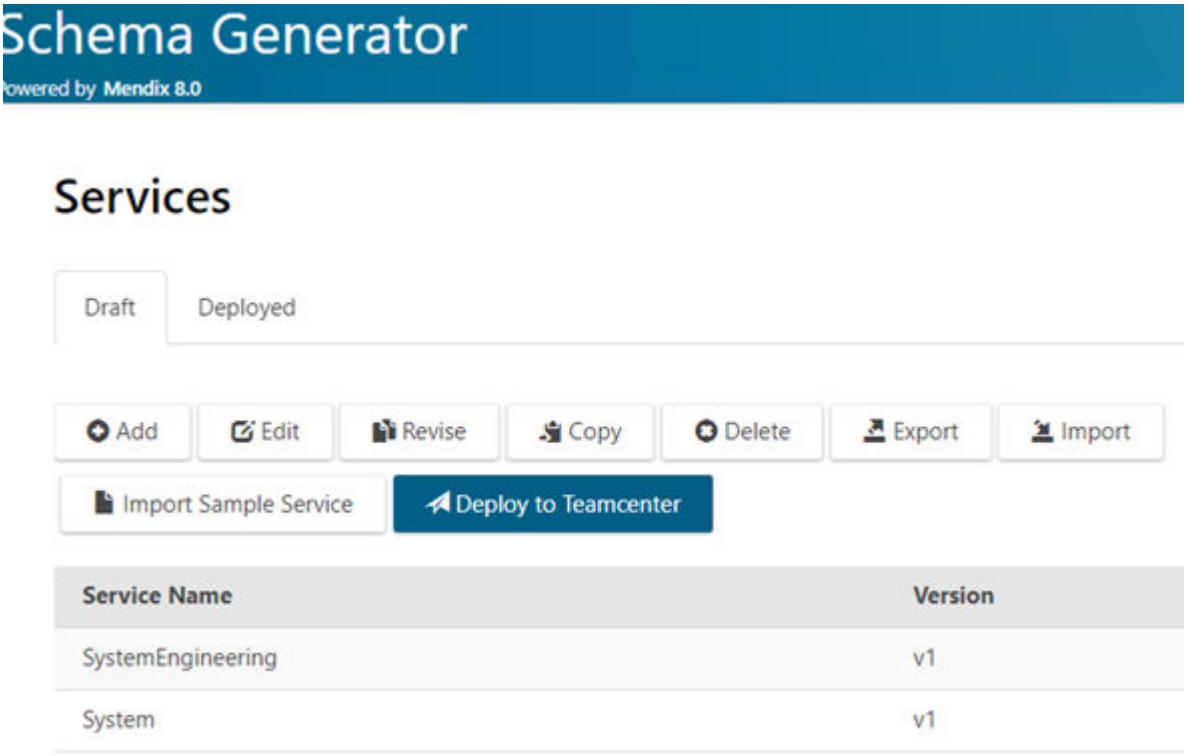
4. In the **Overwrite Service** dialog box, click **Yes** to overwrite the existing services with the new service.

The service you imported replaces the last revision of the service.

Create a subtype of an object

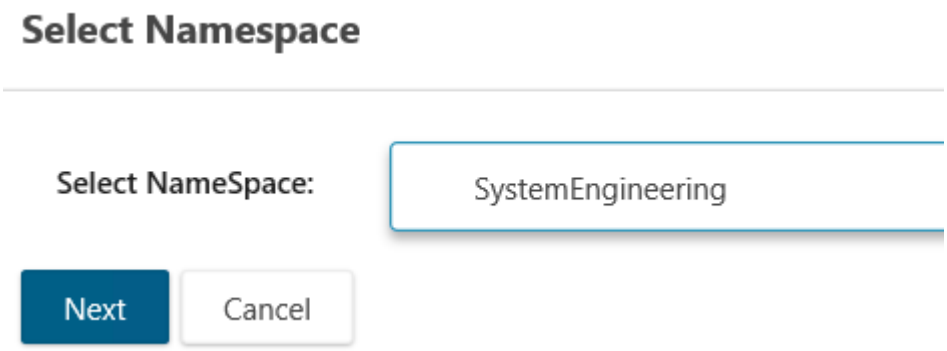
1. Click **Home** to get to the Services page.

2. From the **Draft** or the **Deployed** tab, select a service and click **Edit**.

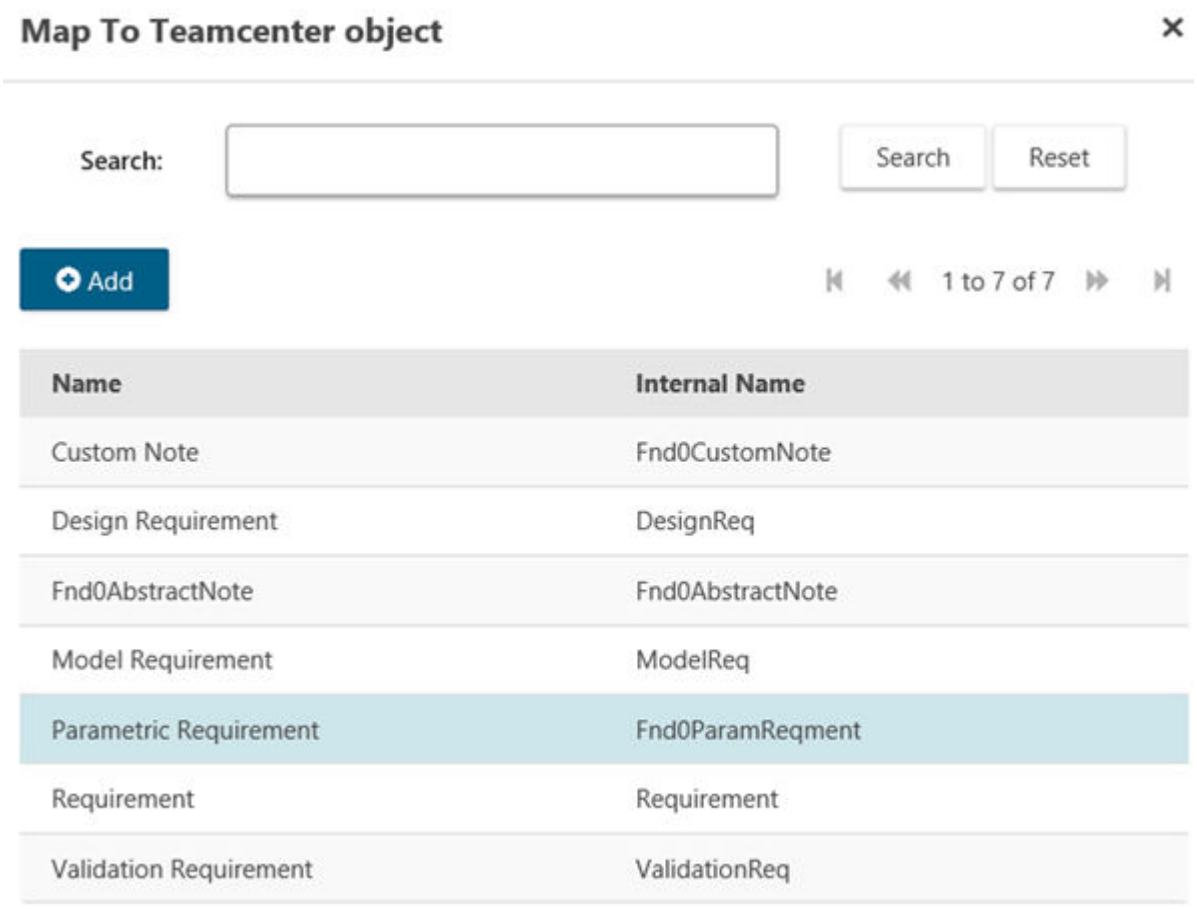


3. Select an object and click **Add SubType**.

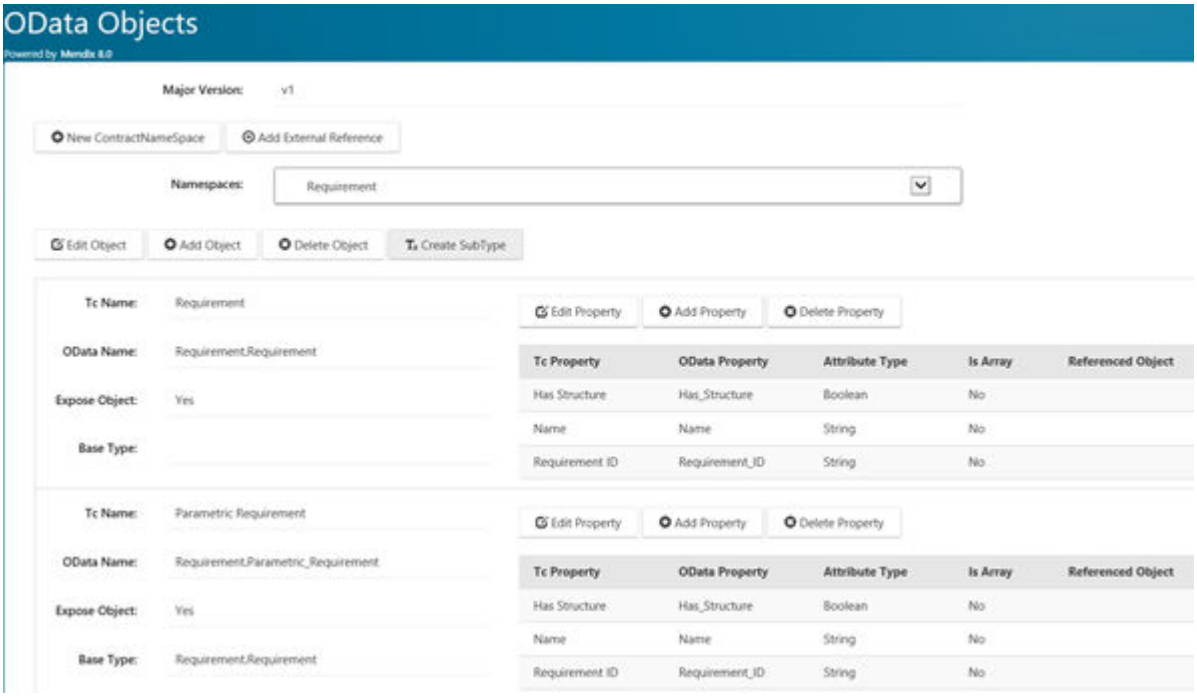
4. In the **Select Namespace** dialog box, select the namespace under which you want to create the subtype and click **Next**.



5. In the **Map to Teamcenter object** object dialog box, select the object that the subtype object maps to.



6. Click **Add** to create the subtype of the object.



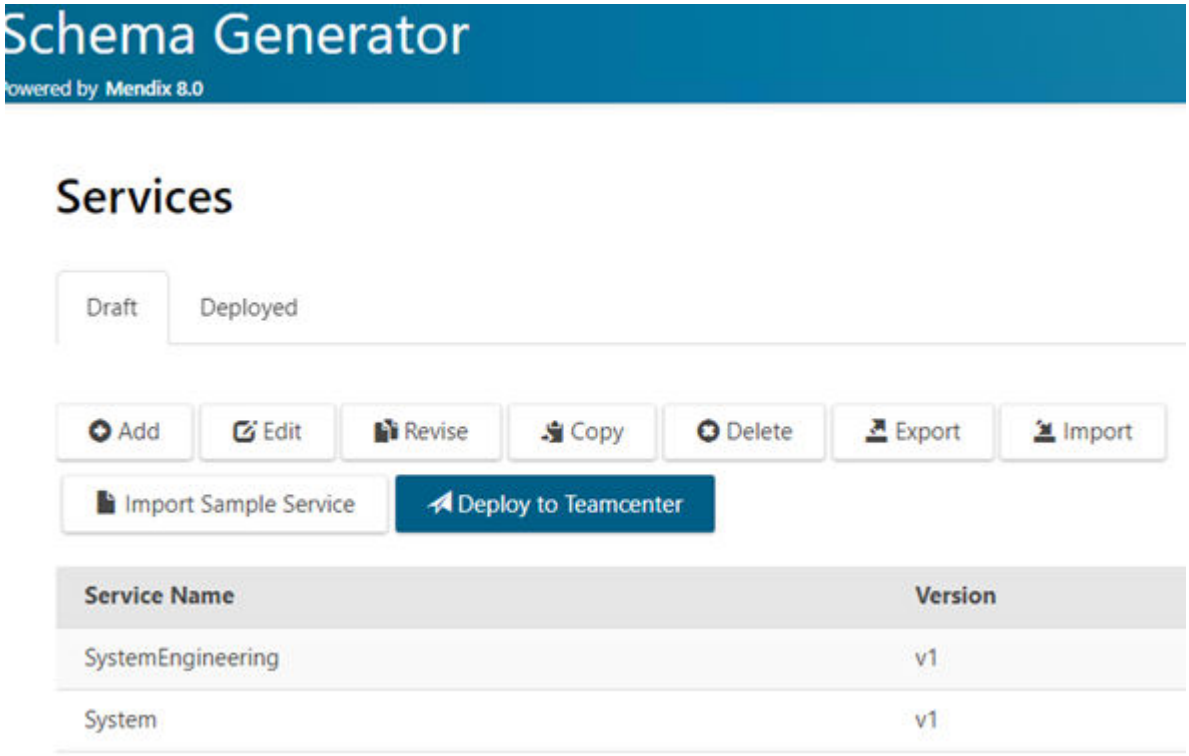
Delete a namespace

You can delete a namespace of a service that has not been deployed yet. To delete the namespace of a deployed service, you must create a major version of the service. Specifically, you must revise the service.

To delete a namespace:

1. Click **Home** to get to the Services page.

2. From the **Draft** tab, select a service and click **Delete**.



3. In the **Confirmation** dialog box, click **OK**.

7. Using the Teamcenter OData service in Mendix

Overview of using OData service in Mendix

To use the Teamcenter OData service, the Mendix developer must:

- Import the Teamcenter Login Module app into the Mendix project.
- Capture the Teamcenter connection configuration by creating a page or hardcoding it.
- Use the retrieve cookie mechanism after successful logon to get the cookie from the database. This cookie can be used anywhere in Mendix.
- Drop services from Mendix Data Hub to the entity diagram.

  Or

  Import the service using the CSDL file.
- Call the header microflow from your page to obtain cookie information.

Import the Teamcenter Login Module app into your Mendix project

Contact your Siemens Digital Industries Software representative to get the Teamcenter Login Module app.

After you get the app, import it into your Mendix project.

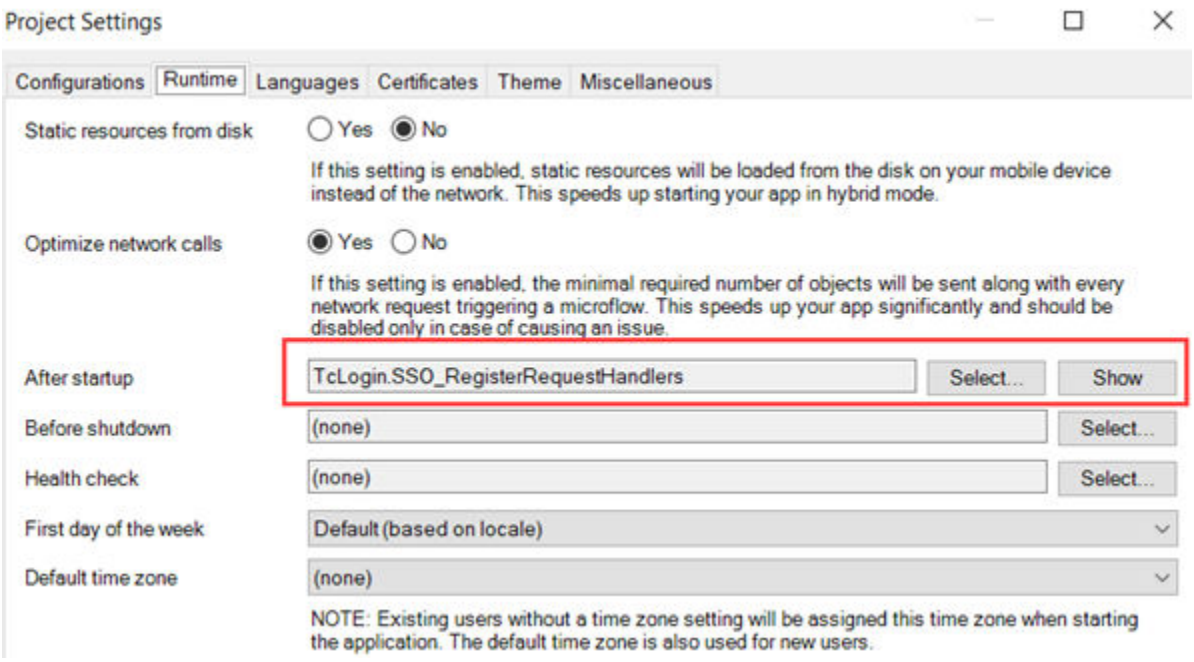
Set up the connection to Teamcenter using the Teamcenter Login Module app

To use OData services, you must first connect to Teamcenter by capturing the following information:

- Teamcenter configuration information
  - You can hardcode this information in your microflows, or you can obtain it by creating a page.
  - See the documentation on the TeamcenterConfiguration domain entity in Mendix for more information about the information that must be captured.
- Teamcenter logon information

  - Create a page to obtain Teamcenter credentials.
  - See the documentation on the Credentials domain entity in Mendix for more information about the information that must be captured.

- (Optional) If you are using Security Services (Teamcenter SSO), ensure that you add the **SSO_RegisterRequestHandlers** microflow to your startup microflow.



- Once logon is successful, a cookie is generated in the Mendix database. You can use this cookie to access Teamcenter information.

You can use the following Java actions in your custom microflows:

- **Login**: Use this for logging on to Teamcenter.
- **Logout**: Use this for logging out of Teamcenter. After logout, the cookie for the session is deleted.
- **Retrieve Cookie**: Use this for obtaining the cookie from the Mendix database.

You can also use sample microflows from the **Use_me**>**Microflows** directory.

Update the Teamcenter OData service with cookie information

You must update the Teamcenter OData service with cookie information so that the service can access Teamcenter. To update the service:

- From the **Settings** tab of the Teamcenter OData Service, in the **HTTP Headers** section, select the header microflow from the **Headers from microflow** box.

  You can choose the sample **RetrieveHeaders** microflow from the **Use_Me** >**Microflows** folder.

Access Teamcenter OData services

Access Teamcenter OData services from Mendix Data Hub

1. From Mendix Studio, click the **Connector** tab and search for the Teamcenter OData service.
2. Drop the service on to the **Domain Model** of your app.

The service is created in your app folder.

Access Teamcenter OData services by uploading CSDL files

1. Right-click a folder and choose **Add other** > **Consumed OData Service**.
2. In the **Add Consumed OData Service** dialog box, type in a name for the service and click **OK**.
3. Click **Select a file from disk** and select a CSDL schema file.
4. In the **Select Metadata** dialog box, select the objects and properties as required and click **OK**.

The domain model and the service are added to the folder.

8. Supported OData features

Requesting data

- Requesting entity collections

  ```
  <ServiceRoot>/Revisions
  ```

- Support for selecting only specific properties using $select

  ```
  <ServiceRoot>Revisions?$select=Name,Id
  ```

- Requesting an individual entity by ID

  ```
  <ServiceRoot>/Revisions(<ID>)
  ```

- Requesting referenced entity

  ```
  <ServiceRoot>/Revisions(<ID>)/Owner
  ```

- Requesting an individual property

  ```
  <ServiceRoot>/Revisions(<ID>)/Name
  ```

- Requesting an individual property of a referenced entity

  ```
  <ServiceRoot>/Revisions(<ID>)/Owner/Name
  ```

- Requesting an individual property raw value

  ```
  <ServiceRoot>/Revisions(<ID>)/Name/$value
  ```

Filtering

- Filter the data based on primitive properties

  ```
  <ServiceRoot>Revisions?$filter=Name eq 'abc'
  ```

- Filter the data based on navigation properties

  ```
  <ServiceRoot>Revisions?$filter=Owner/Name eq 'john'
  ```

- Filter the data based on navigation properties (Collection) using the lambda operator *any*

Get the list of revisions with the status `TCM Released`

```
<ServiceRoot>Revisions?$filter=StatusList/any(var0:(var0/Name eq 'TCM
Released')
```

Note that the lambda operator *all* is not supported yet.

- Request associated or referenced objects along with the primary resource using $expand

```
<ServiceRoot>Revisions?$expand=Owner

<ServiceRoot>Revisions?$expand=Owner($filter=UserName eq 'john')
```

Sorting

- `<ServiceRoot>Revisions?$orderby=Name asc`
- `<ServiceRoot>Revisions?$orderby=Name desc`

Paging

- Request a specific page from the query result

```
<ServiceRoot>Revisions?$top=2&$skip=5
```

Actions

- The service author can define OData actions to perform complicated read and writes on OData business entities. The actions defined are consumed by end users of the application. Teamcenter OData Service can be mapped to any of the published Teamcenter SOA operation. Thus when the OData action endpoint request is fired, the mapped Teamcenter SOA operation is executed.

  For example, if the service author want to expose the operation of adding users to an item revision:

  - Expose the Teamcenter business object as an OData entity. For example, Item Revision.
  - Add a new action such as **addParticipant** and map it to the Teamcenter SOA operation **Participant-2018-11-Participant/addParticipants**.
  - Deploy the service to Teamcenter.
  - Consume the action end point using a REST client.

    POST *Service-URL***/***Service-name***.svc/***Service-version***/addParticipants**

```
{
    "wso":{"uid": "<UID>"},
    "assignee":{
        "uid":"<UID>"
        },
    "participantType":"<PARTICIPANT_TYPE>",
}
```

9. Using the Teamcenter OData Sample App

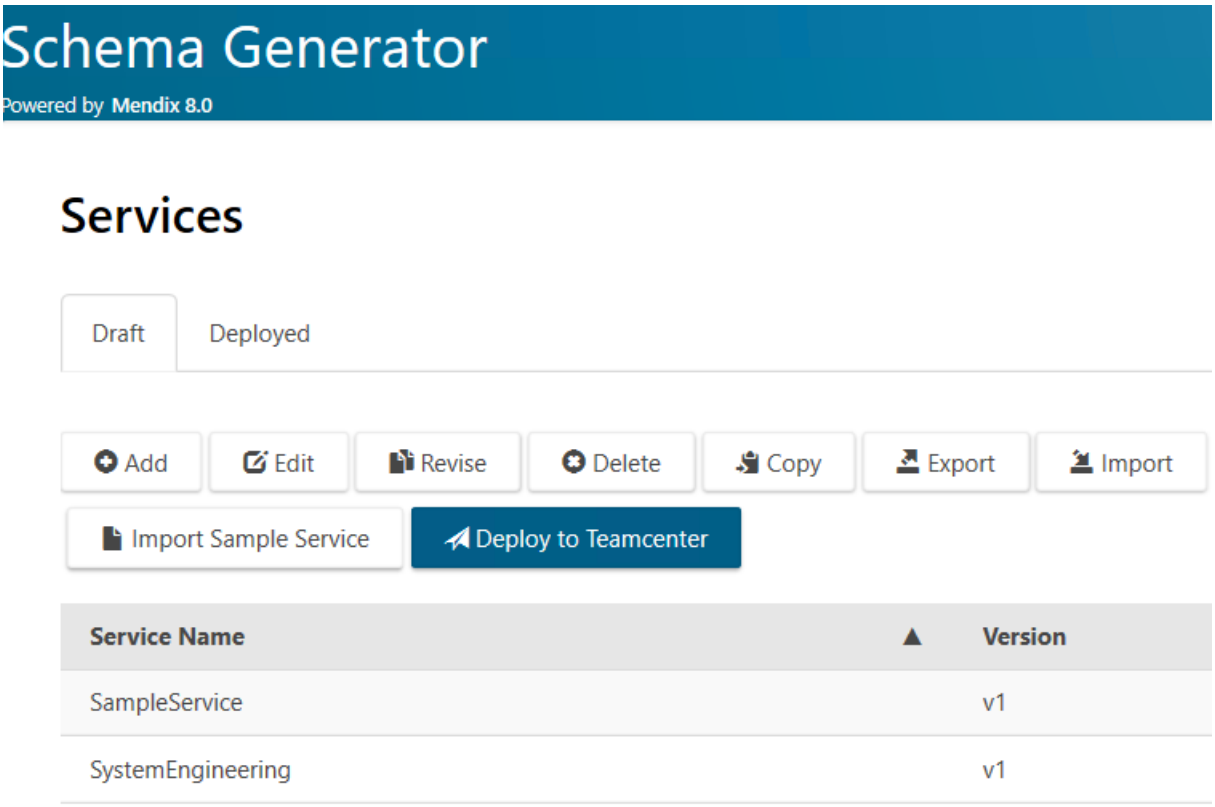
Prerequisites

- Install Teamcenter with the OData template and microservice.
- Install Change Manager and Schedule Manager.

Setup OData Schema for the sample app

You must import and deploy this service to Teamcenter.

1. In the **Services** page, click **Import Sample Service** to import the Teamcenter OData service **NewScheduleService**.



2. Deploy **NewScheduleService** to Teamcenter.
3. Download the OData Sample app.

   Contact your Siemens Digital Industries Software service representative for information on how to download the OData Sample app.

4. Launch the OData Sample app using Mendix Studio Pro 8.16 using the following command from the Mendix Studio Pro *modeler* directory:

```
C:\apps\Mendix\8.16.0.9638\modeler>studiopro.exe --enable-data-hub
--enable-non-integer-keys --enable-one-way-navigable --enable-non-
countable-entities --enable-attribute-capabilities
```

5. Update the **Default Value** entry of the **NewScheduleService_Location** constant with the hostname and port values of the Teamcenter environment and with the service version in the following format:

   http://hostname:port/tc/micro/ODataService/v1/NewScheduleService.svc/service-version-number

   Example:

   `http://18.191.238.55:7001/tc/micro/ODataService/v1/NewScheduleService.svc/v1`

6. Update the **TeamcenterConfiguration** constant with the Teamcenter URL.

   Example:

   `http://18.191.238.55:7001/tc`

Creating data for the OData Sample app

Prerequisites

The following applications must be installed in Teamcenter.

- Change Manager
- Schedule Manager

To create data for the OData Sample app:

1. Create a Change Request revision.
2. Create an Item revision.
3. Relate the Item revision to the Change Request revision with the **CMHasImpactedItem** relation.
4. Create a Schedule.
5. Create tasks under the Schedule.

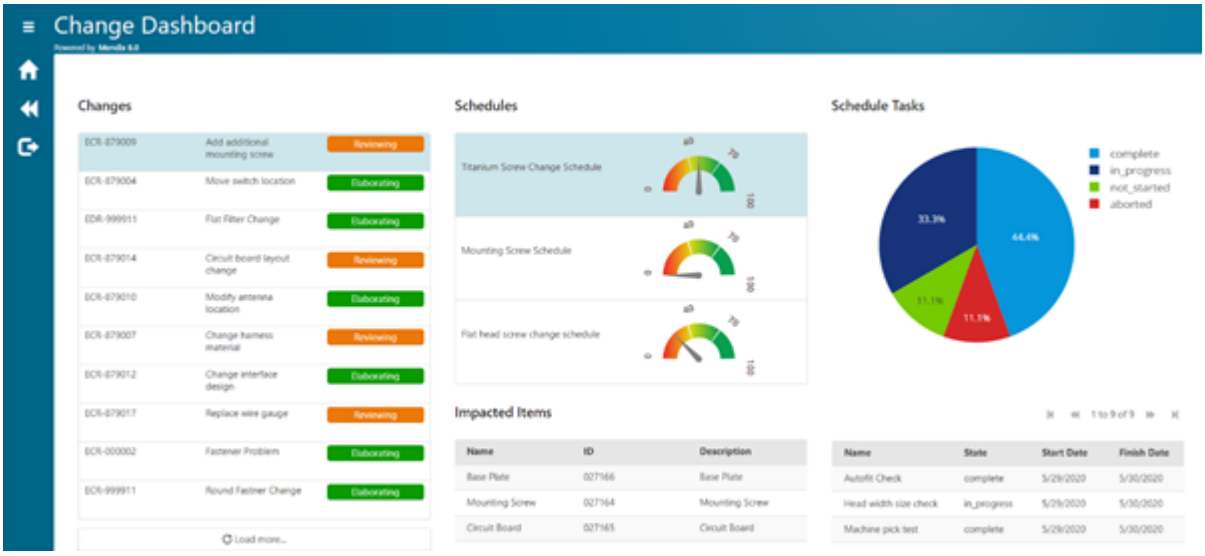
6. Relate the Change Request revision and the Schedule with the **CMHasWorkBreakdown** relation.

Run the OData Sample app

1. Run the OData Sample app from Medix Studio Pro.

2. Log on to Teamcenter as a user with credentials that have access to data created previously.

3. If the schema is deployed and the sample data is available, the dashboard is rendered as follows:



10. Using the sample Postman collection

Overview of the sample Postman collection

The OData services that you create using the Teamcenter OData Framework can be used with any REST client. To help you understand how to use these services, a sample collection is available that can be used with the Postman tool.

Get the Postman collection to query Teamcenter Schedule Manager data

To use this collection, you require a service that can query Teamcenter Schedules. You can use the service bundled with the Teamcenter Sample Application or create and deploy an OData service with Schedule and Schedule Tasks.

The Postman collection is a JSON file called *SampleScheduleManagerSchema.postman_collection*. You can access this file as follows:

1. In Mendix Studio, open the ODataMappingApp.

2. Click **Project** > **Show Project Directory in Explorer**.

   This opens the project directory of the ODataMappingApp in Windows.

3. Open the **Resources** directory to get the *SampleScheduleManagerSchema.postman_collection* JSON file.

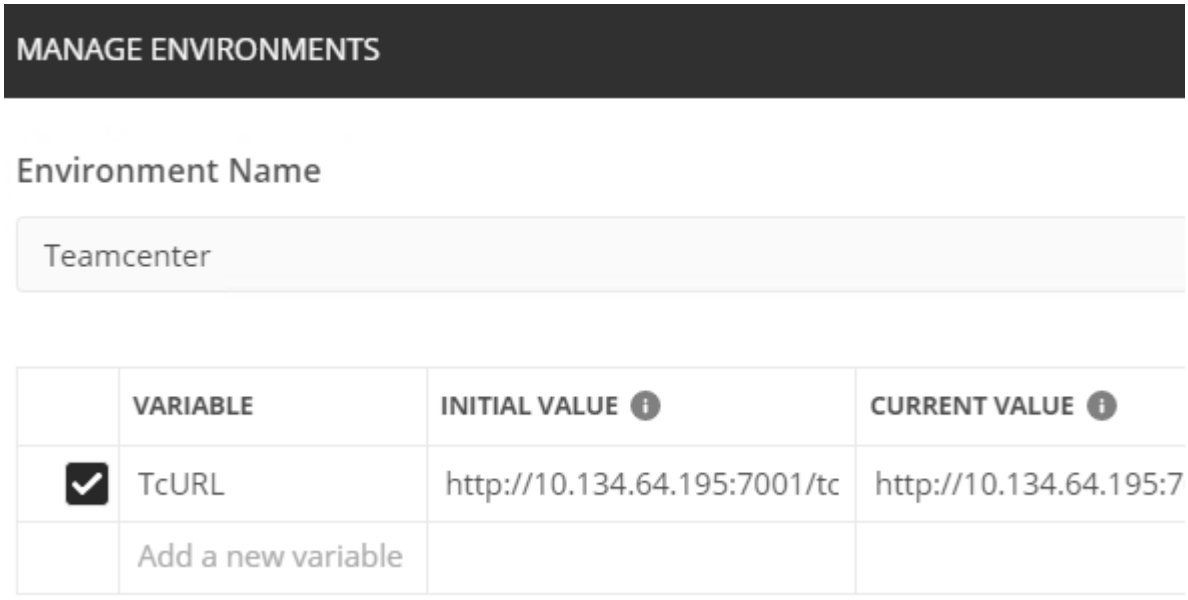
Import the sample collection to Postman and set it up

1. Get the *SampleScheduleManagerSchema.postman_collection* JSON file from the ODataMappingApp project directory.

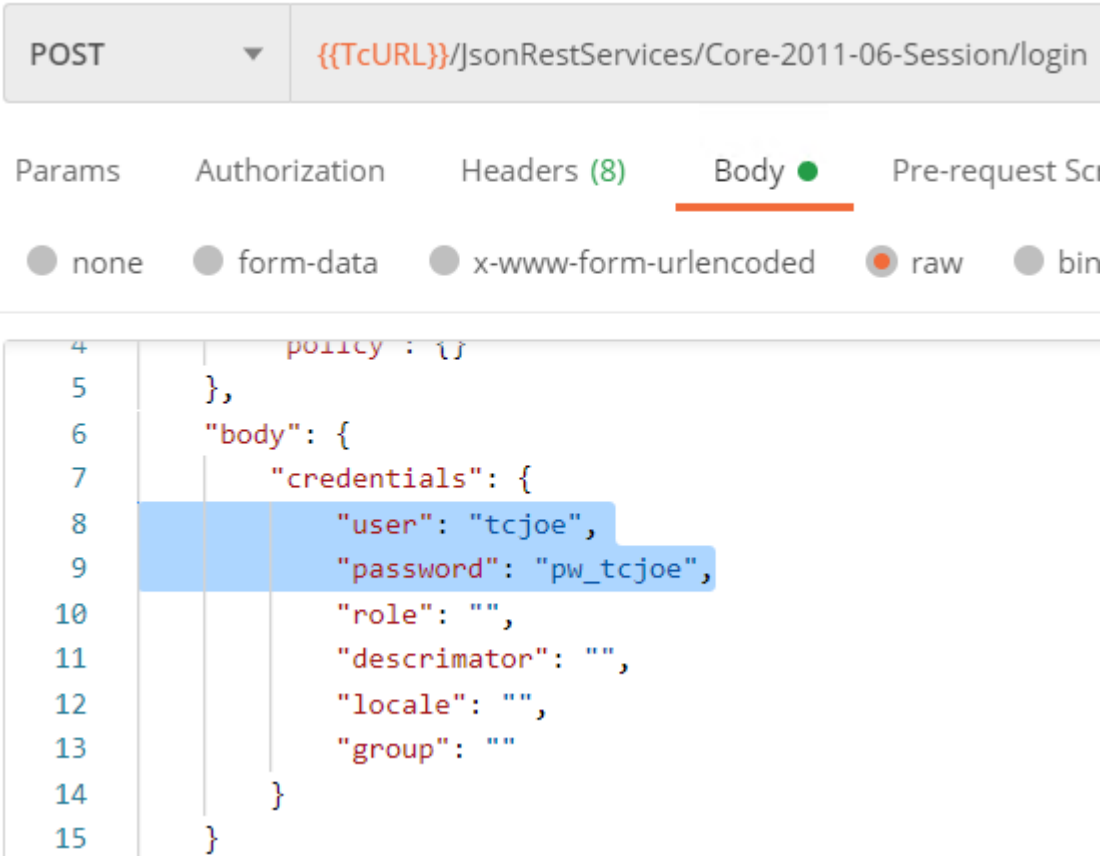
2. Import the *SampleScheduleManagerSchema.postman_collection* JSON file into Postman.

3. Create a Postman environment with the following variable:

   - **TcURL**: Specifies the Teamcenter server url, for example, http://10.134.64.195:7001/tc



- In the Teamcenter Login request, in the **credentials** section, update the values of **user** and **password** with the Teamcenter username and password.



4. Run the Teamcenter Login request.

Once the logon is successful, you can run REST queries and work with Teamcenter data.

Before using the queries, ensure that you update the service name and version to match with the one you are using currently.

Queries available in the collection

Query	Description	Example
Metadata	Provides the metadata of the service. This will list all the objects, the properties, and the operations that are available in the service.	*{{TcURL}}*`/micro/ODataService/v1/SampleService.svc/v1/$metadata`
All Objects query	Lists all the objects for the specified entity.	*{{TcURL}}*`/micro/ODataService/v1/SampleService.svc/v1/Schedules` Lists all the schedules in the **SampleService** service.
Filter query	Filters the result of a query based on a specified filter. Note that you can only filter on those properties that are available in the service.	*{{TcURL}}*`/micro/ODataService/v1/SampleService.svc/v1/Schedules?$filter=(Name eq 'Schedule1')` Filters all schedule tasks in the **SampleService** service based on the property **Name** with value **Schedule1**.
Expand query	Queries the reference of the specified object.	*{{TcURL}}*`/micro/ODataService/v1/SampleService.svc/v1/Schedule_Tasks?$expand=ScheduleTag` Lists all the schedules that are a part on the schedule tasks.
Navigation Query	Filters the reference of the specified object.	*{{TcURL}}*`/micro/ODataService/v1/SampleService.svc/v1/Schedule_Tasks?$expand=ScheduleTag&$filter=(ScheduleTag/Name eq 'Schedule1')` Only shows schedule tasks for the schedule with the name **Schedule1**.
Navigation Filter	Filters the entity and its association.	*{{TcURL}}*`/micro/ODataService/v1/SampleService.svc/v1/Schedule_Tasks?$filter=(Name eq 'Task2' and ScheduleTag/Name eq 'Schedule1')`

Query	Description	Example
		Shows schedule tasks that have the task name as **Task2** and the schedule name as **Schedule1**.
Key property Query	Filters on key property.	*{{TcURL}}*`/micro/ODataService/v1/SampleService.svc/v1/Schedule_Tasks('AWf5aUf15UdT8B')` Filters all schedule tasks in the SampleService service based on the key filter, that is, UID **AWf5aUf15UdT8B**.