SIEMENS

Polarion Connector for Mendix 1.0

Developer's Guide

Contents

Getting started with Polarion Connector

Introduction

Polarion is an application lifecycle management (ALM) enterprise solution. Software is an integral part of product development across all industries and ALM enables manufacturers to continuously integrate, verify and validate the growing software content they build into their products. Together with Teamcenter, Polarion is an integral part of the product development process and systems driven product development.

Polarion Connector for Mendix enables Mendix developers to access ALM data from Polarion or create and modify ALM data in Polarion.

This documentation provides guidance on using Polarion Connector for Mendix. It assumes that you are familiar with Mendix concepts, processes, and terminology for application development.

Prerequisites

To use Polarion Connector for Mendix, you need the following:

- Mendix Business Modeler 8.5.0
- Appropriate Polarion licenses
- A running and accessible Polarion instance. The minimum Polarion version supported is version 20R1.

Download Polarion Connector, the sample application, and dependencies

Polarion Connector for Mendix is available on the Mendix App Store. To add the connector to your project:

1. Open your Mendix Modeler project.
2. Click the AppStore icon  on the menu bar to access the Mendix App Store.
3. Search or navigate to the Polarion Mendix Connector and click the link.

4. On the Polarion Connector for Mendix page, click Download.
   In the **Import Module** dialog box, click Import.

The connector is imported in your project in the **App Store Modules** folder.

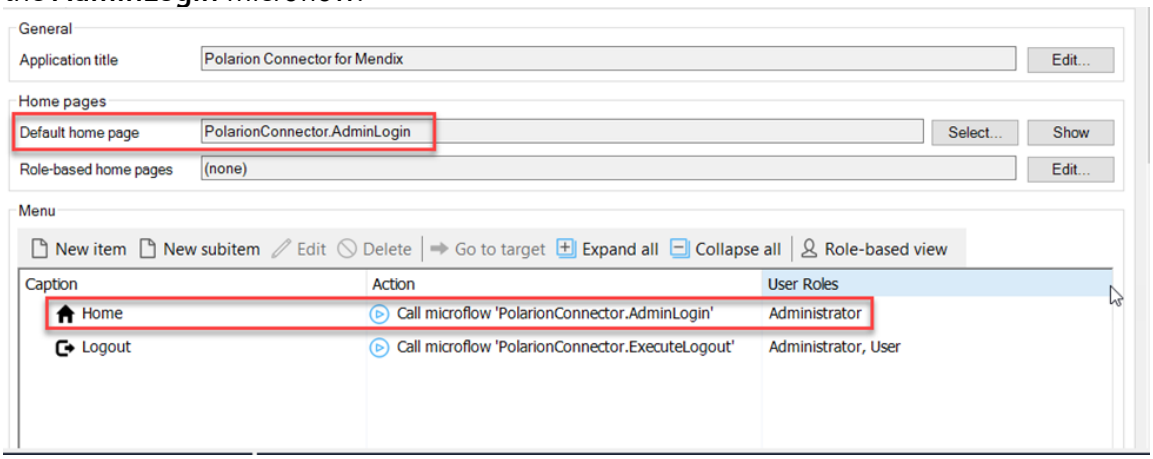
Similarly, download the following applications from the Mendix App Store:

- (Optional) Siemens PLM Software UI Resources. This application should be in the project that contains the Polarion Connector.
- (Optional) Polarion Sample Application. The Sample Application contains sample microflows. You *do not* require to download the Sample Application in the same project that also contains the Polarion Connector.

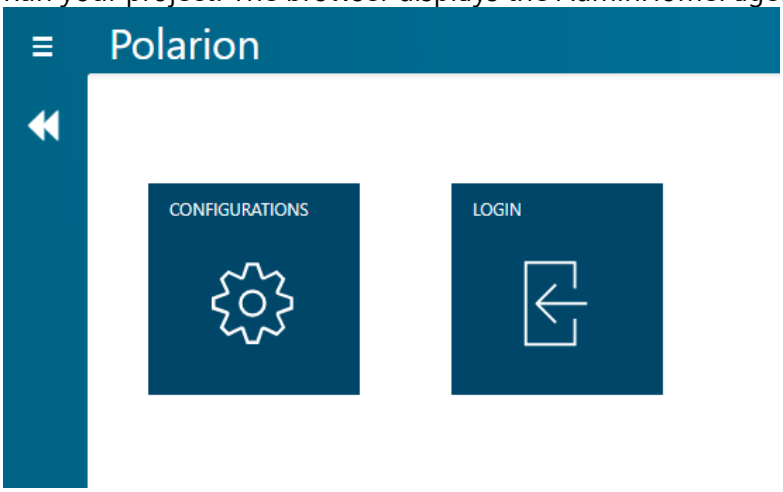
Configure to connect to Polarion

To help administrators configure the connection to Polarion, the Polarion Connector contains configuration microflows and pages. To configure the connection to Polarion, the Mendix administrator must do the following:

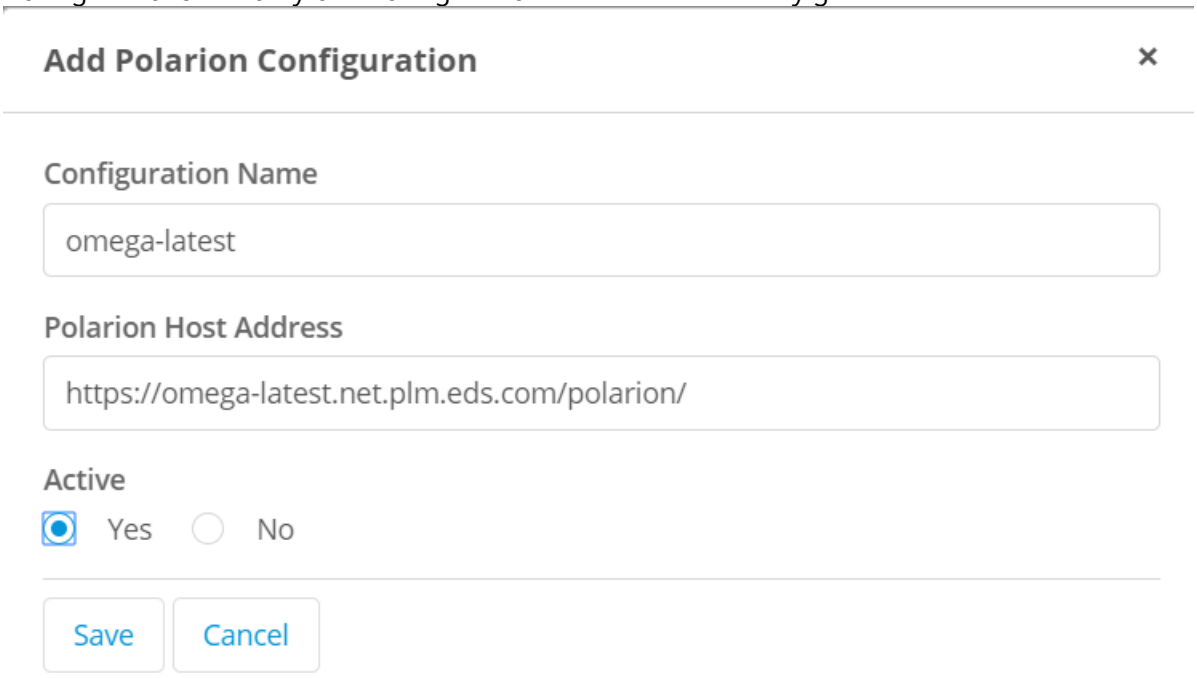
1. In the navigation, update the Default home page and the Home menu item to connect to the **AdminLogin** microflow.



2. Run your project. The browser displays the AdminHomePage.



3. Click the **CONFIGURATIONS** tile in the web browser.
4. Click **New** in the Configurations page.
5. In the Add Polarion Configuration dialog box, specify the required fields such as the **Polarion Host Address** and click **Save**. Be sure to toggle on the **Yes** radio button for the **Active** option selection. The Configurations page is used for creating multiple Polarion Configurations but only one configuration can be active at any given time.

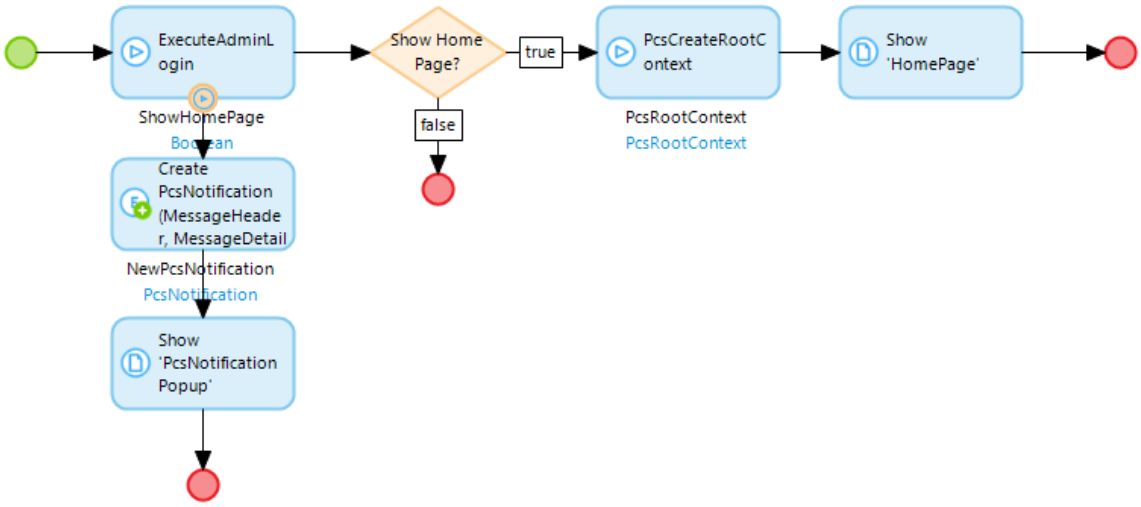


6. Once you have added the Polarion configuration information, log on to Polarion server using the **LOGIN** icon.

    i.

In the previous steps, the configuration shows the Home page included in the Polarion Connector for administrators. If you want to customize the page that you finally see, create a new microflow that contains the core microflow, **ExecuteAdminLogin**. Update your **Default home page** and the **Home** menu item to connect to the new microflow you create.

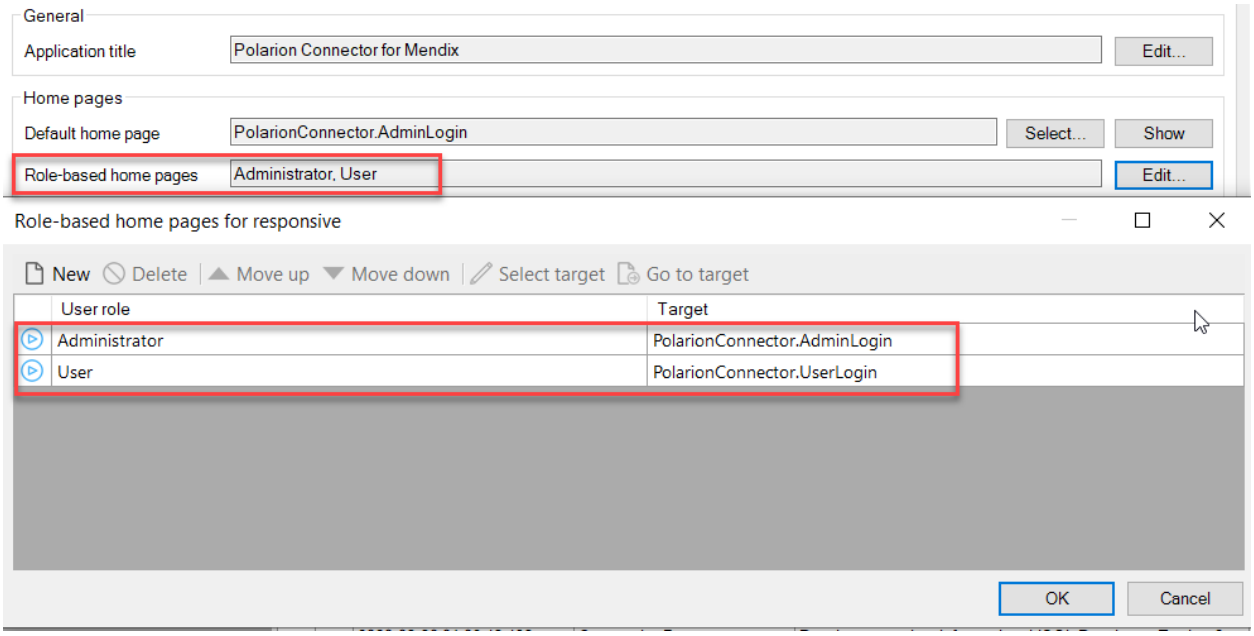
As an example, the Polarion Sample Application has a microflow called **Act_AdminLogin** that uses the **ExecuteAdminLogin** microflow. Refer to that microflow for help.



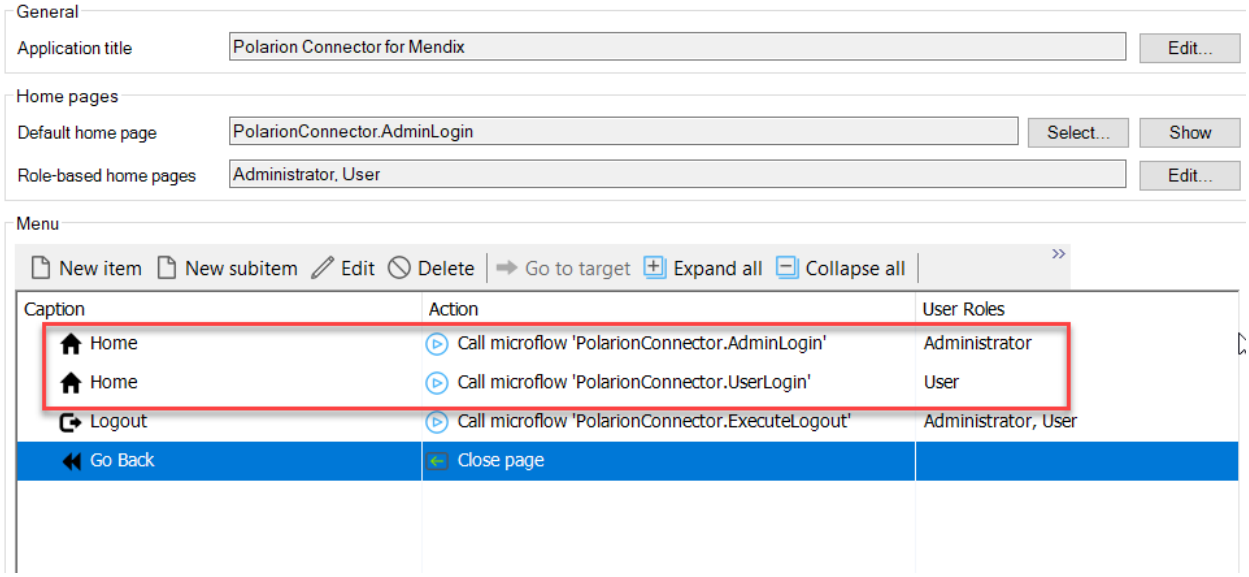
Configure to connect to Polarion in production mode

You can use the **AdminLogin** and the **UserLogin** microflows to configure the security of the Polarion Connector when the security level of your Mendix project is set to **Production.** Configure the security as follows:

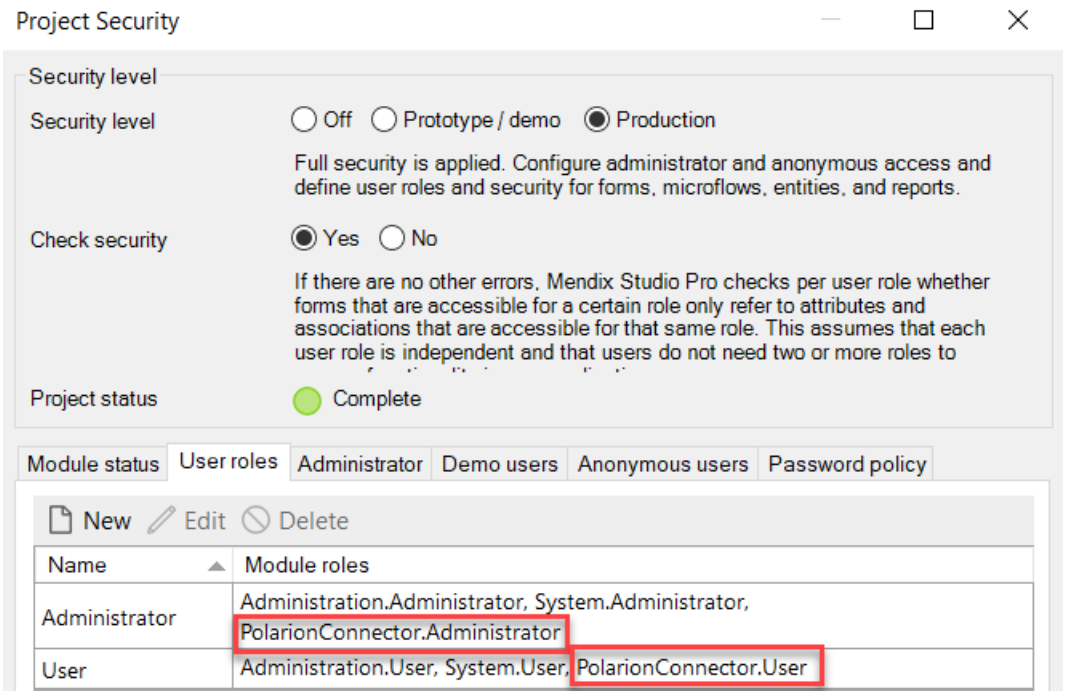
1. In navigation, update the **Role-based home pages** to connect to **AdminLogin** and **UserLogin** microflows based on the role.



2. In navigation, update the Home menu items to include the **AdminLogin** and **UserLogin** microflows.



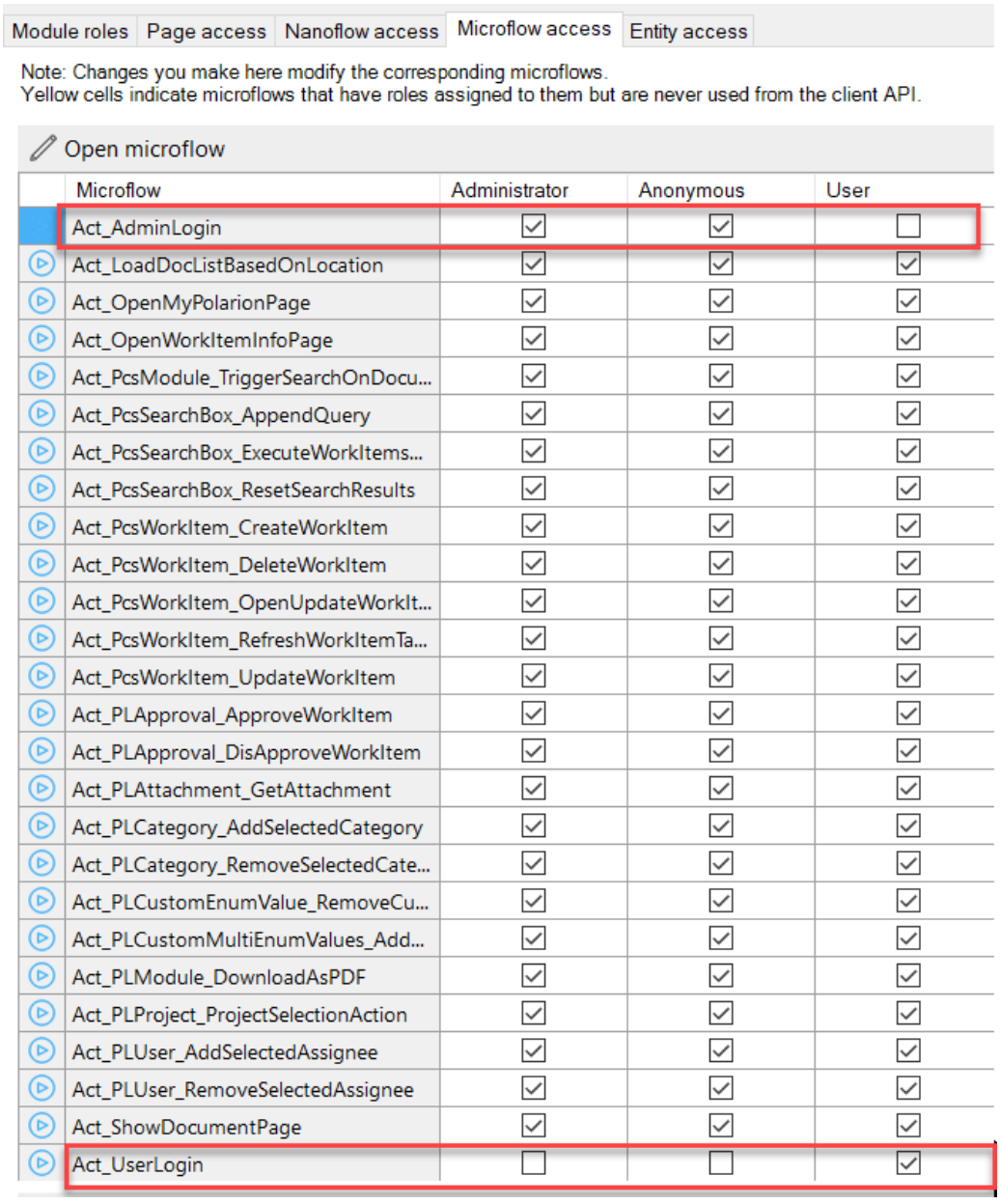
3. In the security page of your project, add the respective administrator and user roles in the **User roles** tab.



4. Run your project and ensure that administrators and users can view their configured login pages.

With the configuration done in the previous steps, it shows the Home pages included in the Polarion Connector. If you want to customize the home pages that you finally see, call **ExecuteAdminLogin** and the **ExecuteUserLogin** microflows in the new microflows you create. In the security page of Polarion Connector, select the appropriate permissions for the new microflows you created. Update your **Default home page** and the **Home** menu item in Project Explorer's Navigation editor to connect to the microflows you create.

As an example, the Polarion Sample Application has an additional role defined, called **Anonymous,** to enable anonymous users sign in. It also has microflows called **Act_AdminLogin** and **Act_UserLogin** which are setup in the Navigation menu to enable role-based access to the home pages. These microflows have appropriate permissions setup in the Security menu for controlling access to Anonymous, Administrator and User roles respectively.



Module roles | Page access | Nanoflow access | Microflow access | Entity access

Note: Changes you make here modify the corresponding microflows.
Yellow cells indicate microflows that have roles assigned to them but are never used from the client API.

Open microflow

Microflow	Administrator	Anonymous	User
Act_AdminLogin	☑	☑	☐
Act_LoadDocListBasedOnLocation	☑	☑	☑
Act_OpenMyPolarionPage	☑	☑	☑
Act_OpenWorkItemInfoPage	☑	☑	☑
Act_PcsModule_TriggerSearchOnDocu...	☑	☑	☑
Act_PcsSearchBox_AppendQuery	☑	☑	☑
Act_PcsSearchBox_ExecuteWorkItems...	☑	☑	☑
Act_PcsSearchBox_ResetSearchResults	☑	☑	☑
Act_PcsWorkItem_CreateWorkItem	☑	☑	☑
Act_PcsWorkItem_DeleteWorkItem	☑	☑	☑
Act_PcsWorkItem_OpenUpdateWorkIt...	☑	☑	☑
Act_PcsWorkItem_RefreshWorkItemTa...	☑	☑	☑
Act_PcsWorkItem_UpdateWorkItem	☑	☑	☑
Act_PLApproval_ApproveWorkItem	☑	☑	☑
Act_PLApproval_DisApproveWorkItem	☑	☑	☑
Act_PLAttachment_GetAttachment	☑	☑	☑
Act_PLCategory_AddSelectedCategory	☑	☑	☑
Act_PLCategory_RemoveSelectedCate...	☑	☑	☑
Act_PLCustomEnumValue_RemoveCu...	☑	☑	☑
Act_PLCustomMultiEnumValues_Add...	☑	☑	☑
Act_PLModule_DownloadAsPDF	☑	☑	☑
Act_PLProject_ProjectSelectionAction	☑	☑	☑
Act_PLUser_AddSelectedAssignee	☑	☑	☑
Act_PLUser_RemoveSelectedAssignee	☑	☑	☑
Act_ShowDocumentPage	☑	☑	☑
Act_UserLogin	☐	☐	☑

Please note that in order to facilitate a full anonymous access of the Sample Application, the Anonymous role has been granted full administrator access to all the pages, entities, and microflows of the application. For a production deployment of your custom application (or if you use some features from this sample application), the Anonymous access may need to be amended or restricted according to your business requirements.

Understanding the Polarion Connector Domain Model

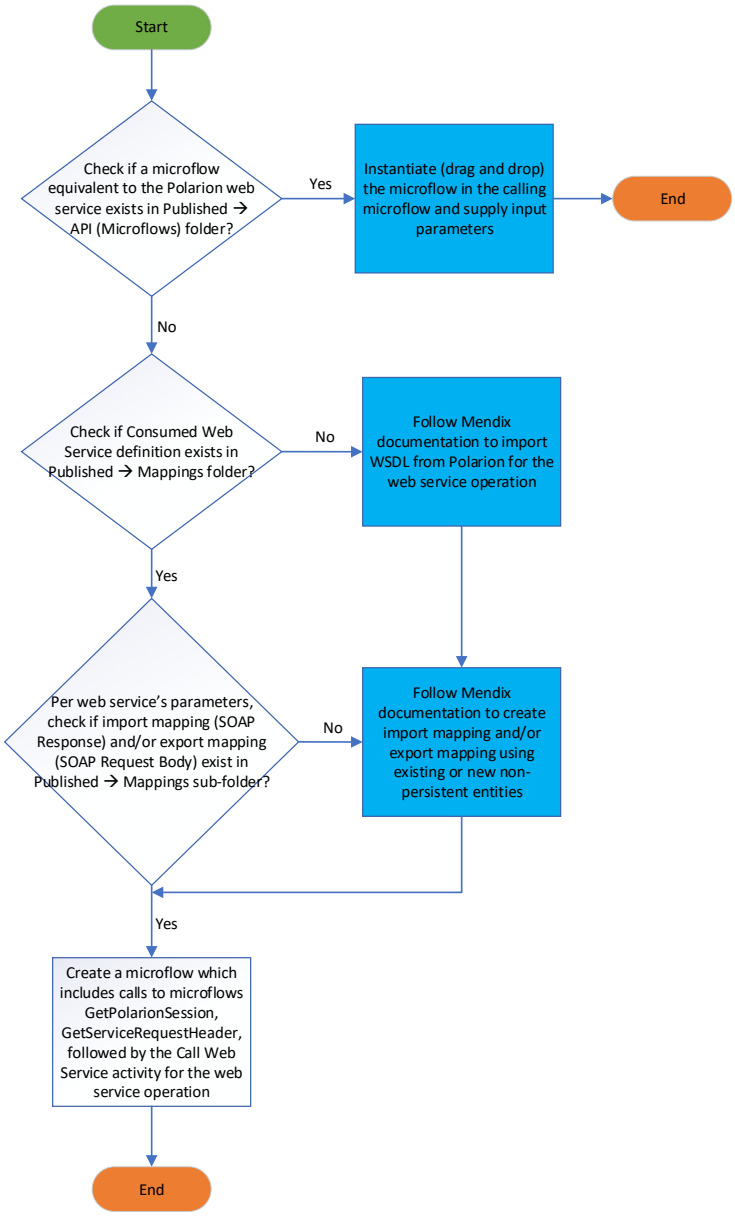
The Domain Model is a data model that describes the information in your application domain in an abstract way. It is central to the architecture of your application. The Domain Model consists of entities and their relations that are represented by associations.

The Polarion Connector for Mendix Domain Model represents Polarion Work Item types, associated objects and their properties.

You can view the Polarion Connector Domain Model by navigating to PolarionConnector→Domain Model from the **Project Explorer**.

To export the Domain Model documentation, right-click the **Project Explorer** and choose **Export documentation**. The Domain Model documentation is exported as an HTML file.

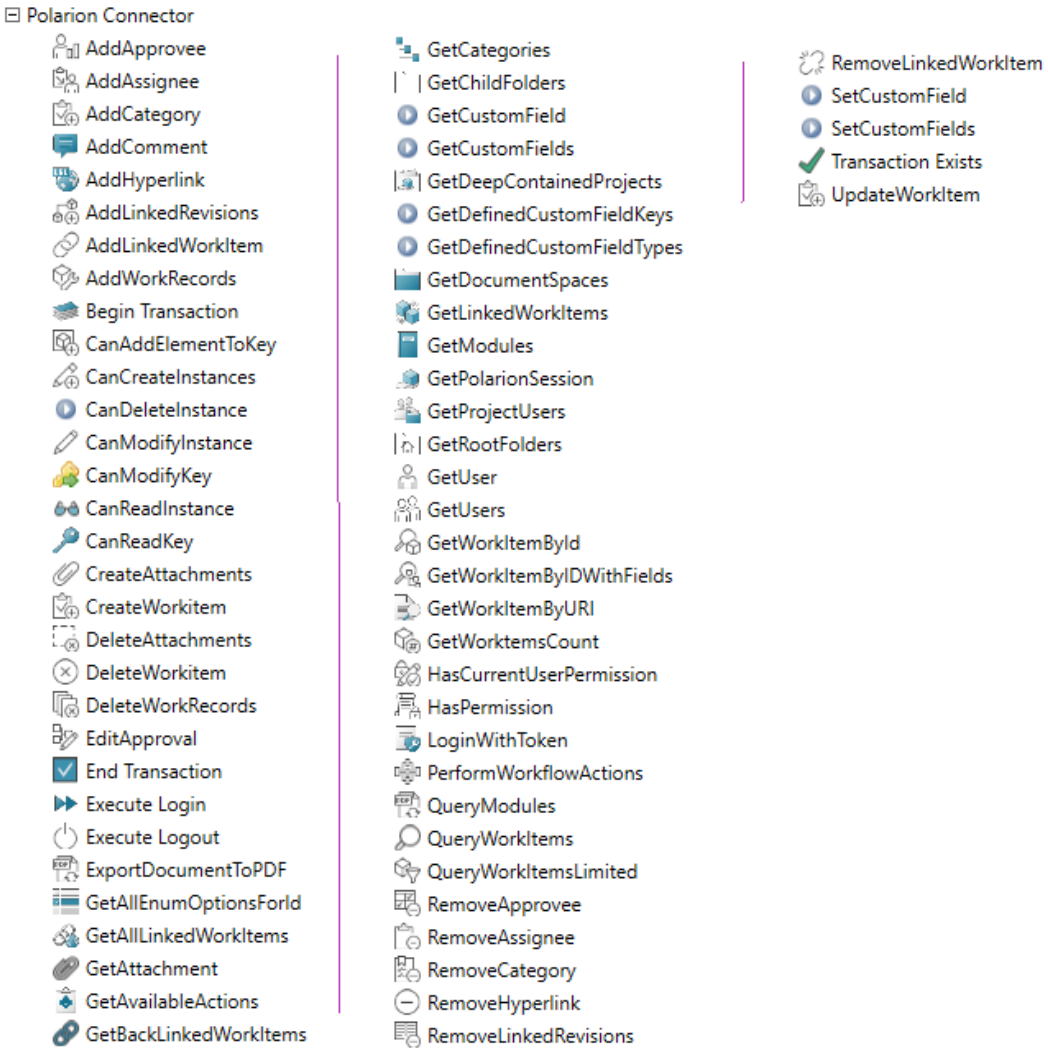
Using Polarion web services through Polarion Connector: process workflow



Using Polarion services included with Polarion Connector

Introduction

Polarion web services are provided through microflows which are referred to as “Polarion service microflows” from here on in this document. You can see the available Polarion services microflows in the **Polarion Connector** section of the **Toolbox.**



Supported services

For information about the supported Polarion services, see Polarion Connector services.

Process to use Polarion service microflows available in the Polarion Connector

1. Design a microflow as per your business logic.
2. Drag a Polarion service microflow from the Polarion Connector category of the toolbox into your microflow.
3. Specify input parameters for the Polarion service microflow.
4. Specify how data is retrieved.
5. Test your application.

Example of using the available Polarion services microflows

The Sample Application has microflows that make use of the Polarion services microflows available in the Polarion Connector. Two such examples are the **Act_PcsWorkItem_CreateWorkItem** and **Act_PcsSearchBox_ExecuteWorkItemsSearch** microflows in the Sample Application. Act_PcsWorkItem_CreateWorkItem makes use of the **CreateWorkItem** microflow from Polarion Connector which in turn calls the **createWorkItem** Polarion service. Similarly, Act_PcsSearchBox_ExecuteWorkItemsSearch makes use of the **QueryWorkItemsLimited** microflow from Polarion Connector which in turn calls **queryWorkItemsLimited** Polarion service. For additional details, download the Sample Application from the Mendix App Store and import it into your project.

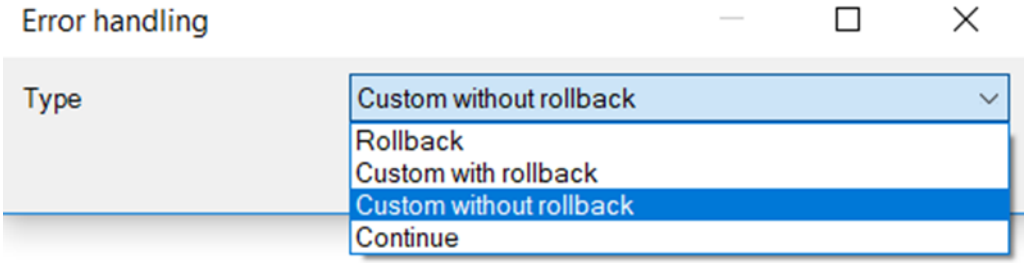
How to handle errors

It is a good practice to set up error handling on all your microflows that call Polarion services microflows. The Sample Application provides the microflow **Sub_HandleException** as an example for error handling. Developers can make use of this microflow or write a similar one of their own.

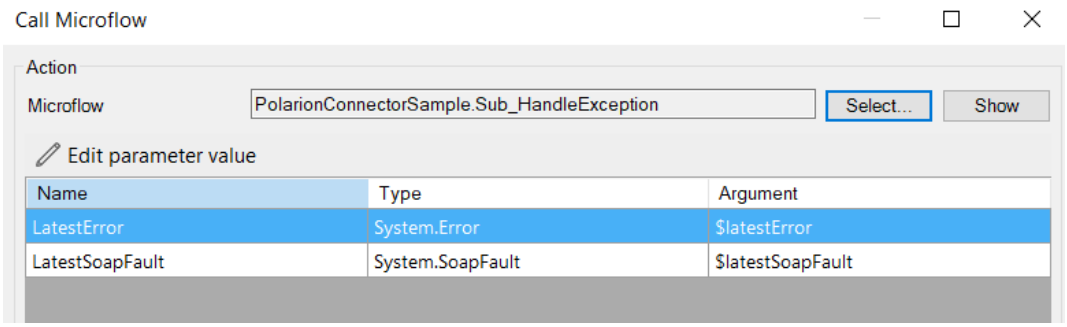
To handle errors using the **Sub_HandleException** microflow:

1. In your microflow, right-click your Polarion service microflow and choose **Set error handling**.

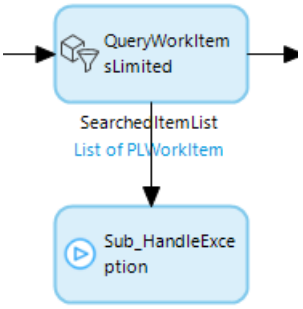
2. In the **Error handling** dialog box, choose the error handling component as **Custom without rollback**. Note that the **Custom with rollback** error handling is already done within the Connector's Polarion service microflow and, it is therefore not needed in the Sample Application.



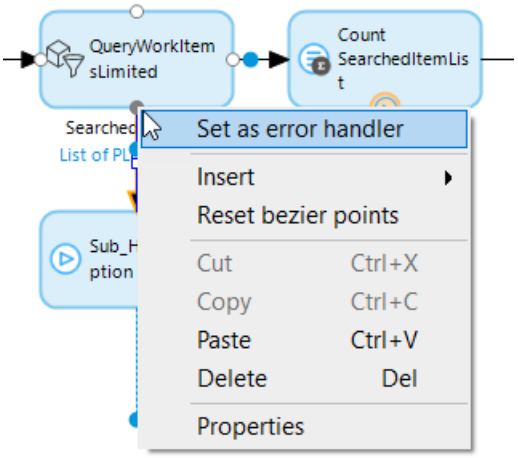
3. Drag the **Microflow call** service from the **Toolbox** to your microflow.
    a. Update the **Action** section of the microflow and select the **Sub_HandleException** microflow.
    b. Update the **LatestError** and **LatestSoapFault** parameters and set them to system maintained objects latestError and latestSoapFault.



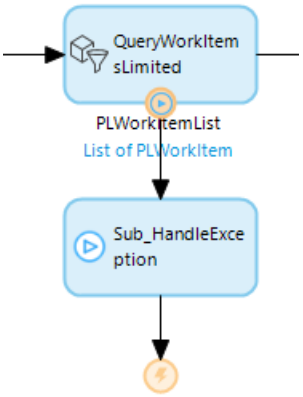
4. Connect the **Microflow call** activity (set to **Sub_HandleException** microflow in the previous step) with the Polarion service microflow from where the error will originate.



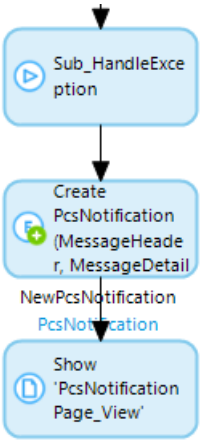
5. Right-click the anchor point of the Polarion service microflow and choose **Set as error handler**.



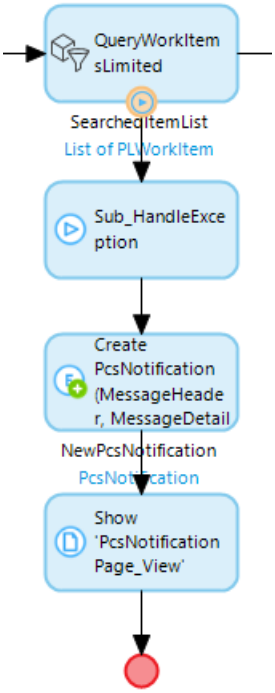
6. Note if the current microflow is a sub-microflow then use Error Event action from Toolbox as the next action after the **Sub_HandleException** microflow. For this case, proceed with the next step in the calling microflow.



7. Formulate a message to notify the user of the error. This message can be shown to the user either through the Show Page or Show Message actions from the Toolbox. The Sample Application commonly uses the method of creating **PcsNotification** object and sets one of its member attributes to error message details ($latestError/Message) which are then displayed on the **PcsNotificationPage_View** page.



8. Specify an End Event after the error message has been displayed to the user and to end the error handling part for your microflow.



Extending the Domain Model

Guidelines to extend the Polarion Connector Domain Model

The Domain Model in Mendix consists of objects/entities and their attributes and associations. It is analogous to the Polarion data model of Work Items which are a major focal point of data and activity in Polarion. Work Item Types represent various artifacts - requirements, tasks, change requests, for example. The Domain Model of Polarion Connector works for all these types but, if and when, a need arises to extend the Domain Model, ensure that:

- You extend the Domain Model in a separate module and not the Polarion Connector Domain Model. This will ensure that you can smoothly get updated versions of the Polarion Connector module.
- You use the methods of generalization (inheritance) or associated entities if you want to add additional information to some specializations of an entity but not to others.
- The entities and associations preferably should match the corresponding Polarion Work Item names, its related features and their properties. You can find information about out-of-the-box Polarion Work Items, its types and properties in the Configure Work Items and Polarion SDK documentation.

Process to extend the domain model while using available Polarion service microflows

- Create the required entities in a separate module as a subtype of an existing entity (specialized entity).
- For example, due to some special handling needs you want to create a separate entity for the *Supplier Part* business artifact, you can create it as a specialization entity of the *PLWorkItem* entity. *PLWorkItem* entity would be the generalization entity (or the super class) in this case.
- If the required entity is available but the required attributes are not available, create a specialization of the entity in a new module and then add the required attributes to this new entity.
  Caution
  Siemens recommends that you always create new entities and not make any changes to the entities that come with Polarion Connector.
- Design a main microflow as per your business logic.
- Drag a Polarion service microflow from the **Polarion Connector** section of the toolbox into the microflow.
- Instantiate and specify input parameters to the Polarion service microflow.

- Specify new (if needed) or existing import/export mappings for the Polarion service microflow.
- Specify how data is retrieved in the main microflow.
- Test your application.

Example: Extend the Domain Model to access additional information

To access other Polarion object types or attributes which are not already defined in Polarion Connector's Domain Model, you must add their definition to your app's Domain Model. This process is the same regardless of whether the object or attribute is out-of-the-box or custom.

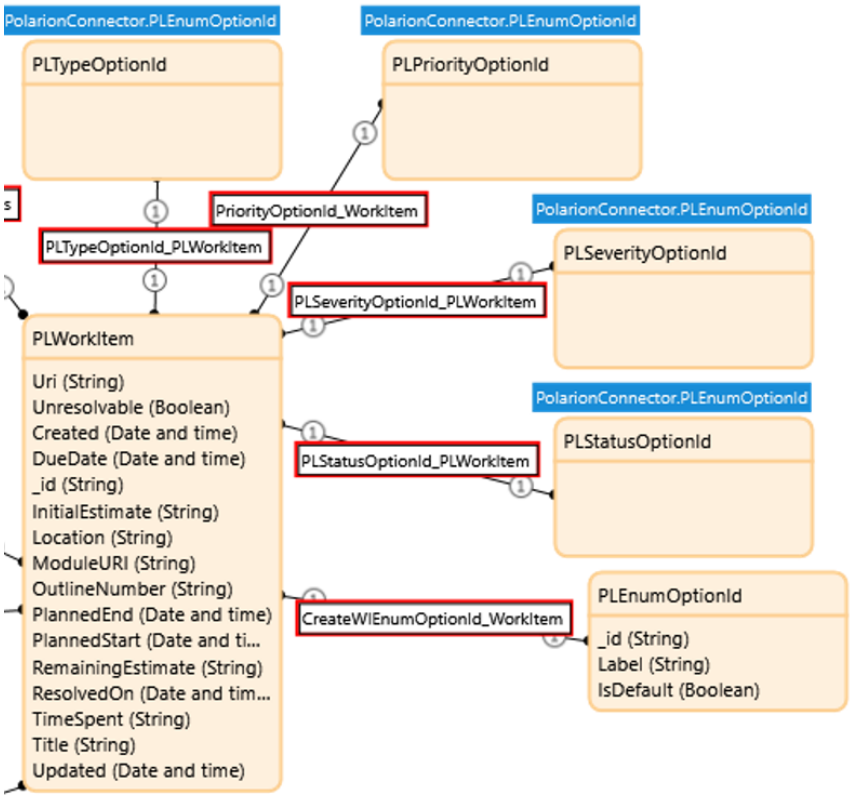
A more common scenario likely to happen is that the desired objects and attributes may already be present in the Doman Model but are linked to the main entity through different levels of associations. They are thus not easily accessible via the main entity to the Mendix UI widgets such as a Data Grid or a Template Grid. The example below tackles such a scenario of the Work Items overview page WorkItemInfoPage_Overview in the Sample Application. This page is populated by an initial query of Work Items and its associations in the microflow Act_OpenWorkItemInfoPage but this microflow also needs to pass a singleton view of the retrieved data to the overview page. For this purpose, the example also illustrates the composition of the Domain Model entities needed to support this feature. That is where we start first.

I. Examine the PLWorkItem entity and its associations

Examine the PLWorkItem entity and its associations to determine which attributes you want to retrieve.

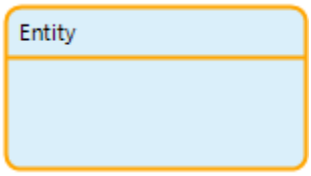
a. This entity has associations representing Type, Priority, Severity and Status which are of interest for displaying on the Work Items page.

b. Attributes Type, Priority, Severity and Status have enum (enumeration) options.

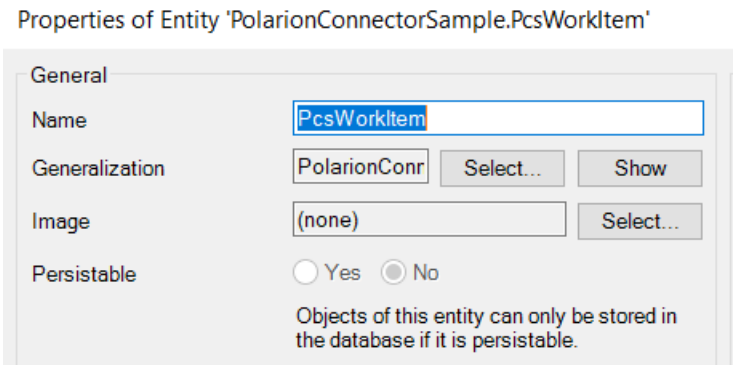


II. Create a new Domain Model entity

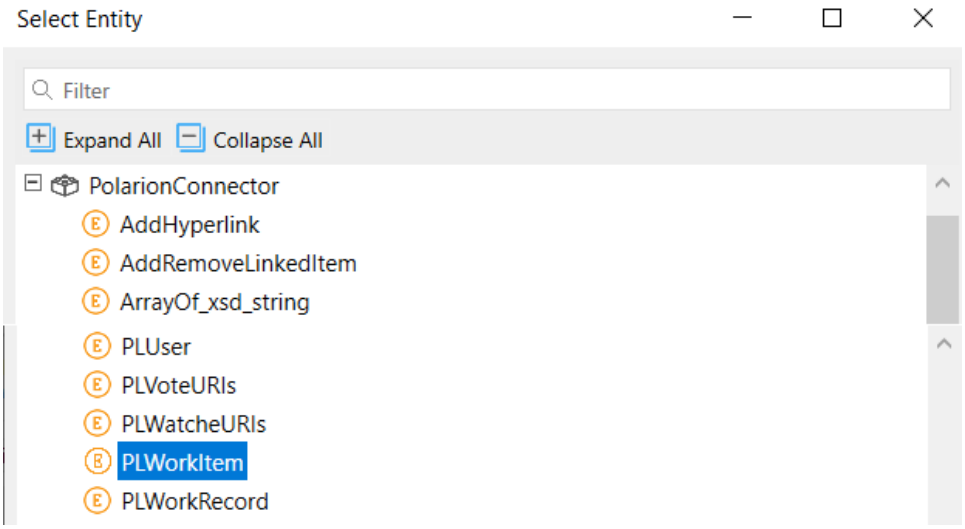
1. In your module's Domain Model, create a new entity.



2. Change the entity's name to somewhat resemble the name of Polarion Connector Work Item type you intend to extend. Also, switch the Persistable property of the entity to be a non-persistent entity.



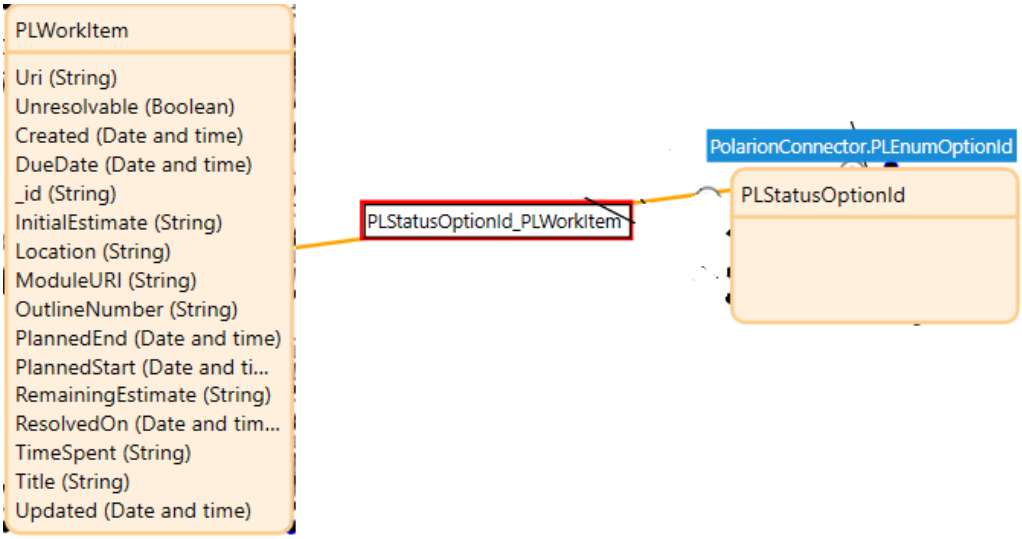
3. Define the new entity's *generalization* to be the Polarion Connector Work Item type you intend to extend.



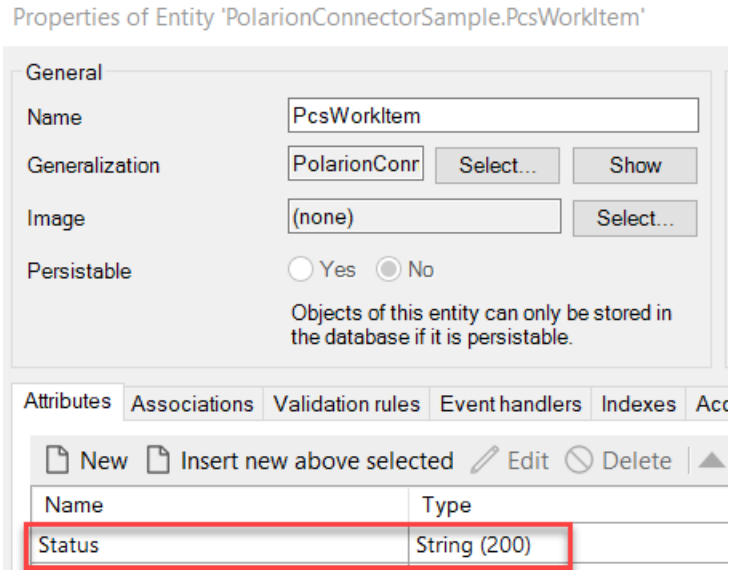
III. Define new attributes to match the Polarion Connector Work Item Type attributes

1. Inspect the Polarion Connector associations, attributes, attribute types, and other parameters to create an equivalent attribute in your module's Domain Model. As an example, note that the attribute Status is of type PLStatusOptionId, that is, it is an enumeration option with an id and it is linked to PLWorkItem through the association PLStatusOptionId_PLWorkItem. Since, the name (label) of the Status enumeration option id is to be displayed on a UI page, you can choose to represent it as a simple string attribute in your module's Domain Model.

- Polarion Connector



- Polarion Connector Sample Application

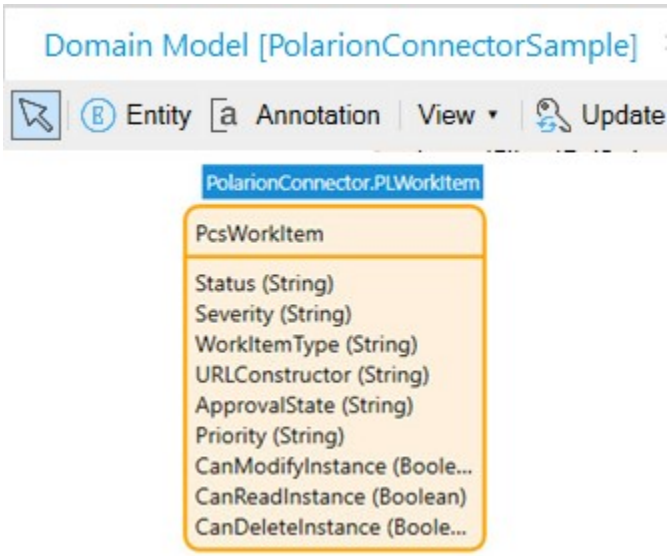


Other attributes such as Type, Priority, and Severity have a similar schema and they can be modeled the same way in your Domain Model.

IV. Complete Entity Definitions

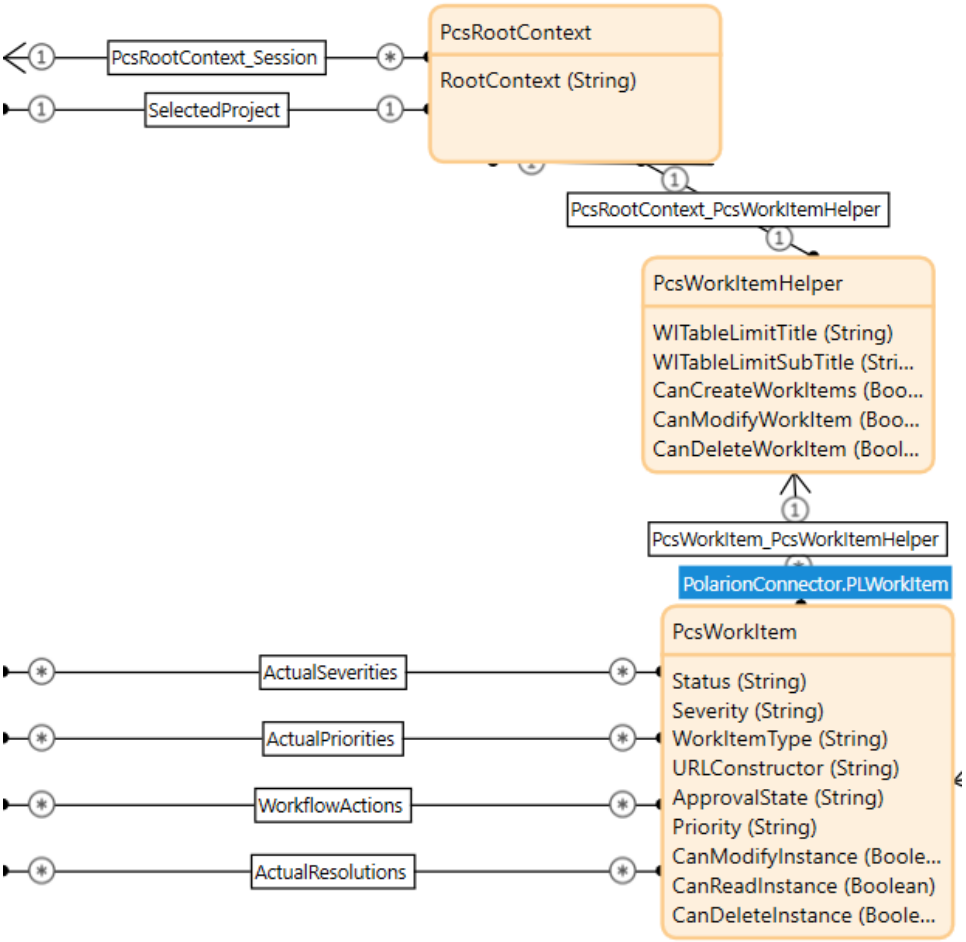
Finish the process of defining other attributes with appropriate data types, entities and associations needed by your module's Domain Model entity.

- For the PcsWorkItem domain entity, additional attributes are added to track permissions, Approval State, URL path to Polarion Work Item, etc.



- PcsRootcontext is an entity that sets up all the context information needed by various microflows. Through its various associations, it establishes links to the session entity, selected project, search criteria entity and PcsWorkItemHelper entity, etc.
- PcsWorkItemHelper is an entity created in the Domain Model with associations to carry a list of PcsWorkItem and PcsRootContext entities.

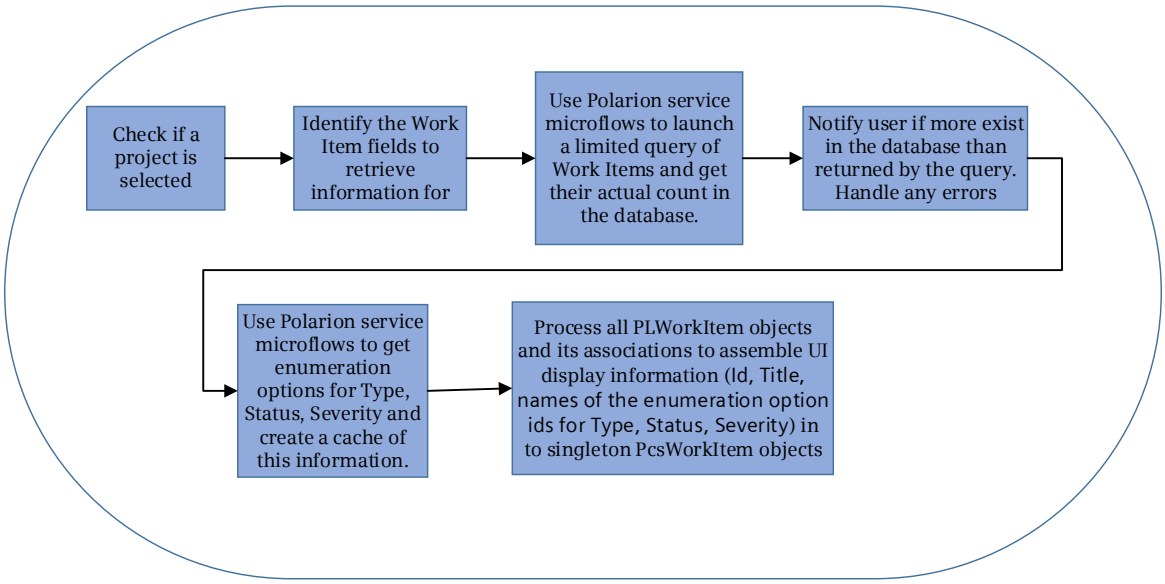
The Domain Model will be similar to the graphic below. You can now start using your new Domain Model entities and its attributes.



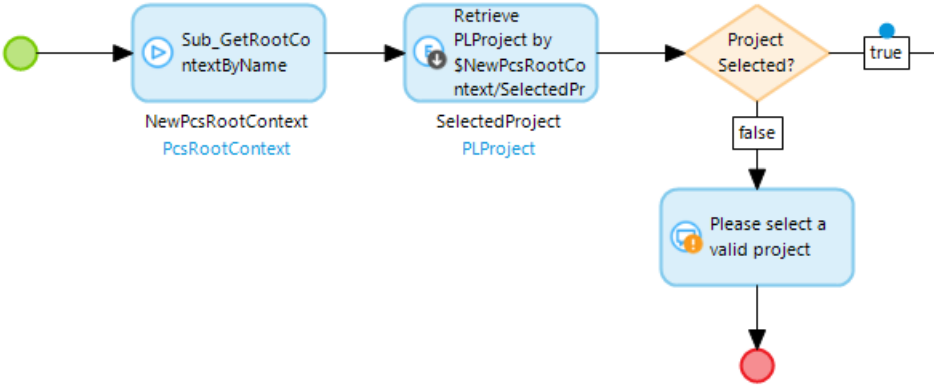
Repeat this process for each object whose properties you want to work with in your module.

V. Build Microflow and Sub-Microflows

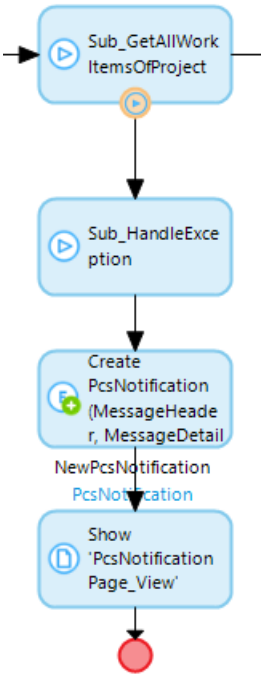
1. Microflows allow you to express the custom logic for your module. Plan your microflow and decide what services you want to use. The Polarion services microflows are available in the Toolbox under the Polarion Connector category. For querying Work Items, a microflow such as Act_OpenWorkItemInfoPage typically consists of the following activities:



2. In the first few steps, ensure that a root context is available for Work Items, and a project selection is made from the initial landing page.

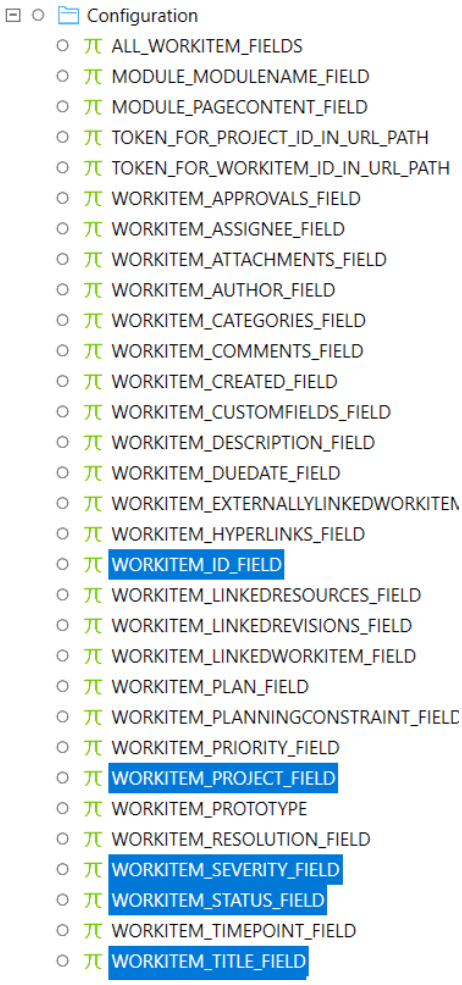


3. Create a sub-microflow to launch the query. In our example it is Sub_GetAllWorkItemsOfProject. The required input to this sub-microflow is PcsWorkItemHelper and through its association PcsRootContext_PcsWorkItemHelper, the PcsRootContext is also passed in. Set Sub_HandleException microflow as the error handler of type Custom without rollback for this microflow. Refer to How to handle errors to complete the steps. Your microflow appears as in the graphic below.

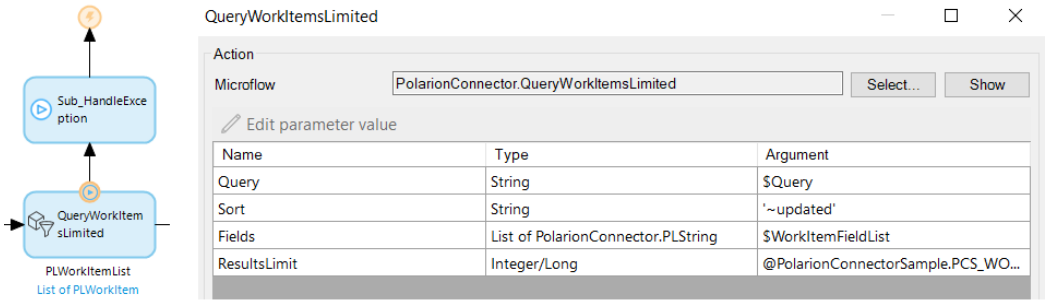


The sub-microflow Sub_GetAllWorkItemsOfProject further calls other sub-microflows some of which are described below.

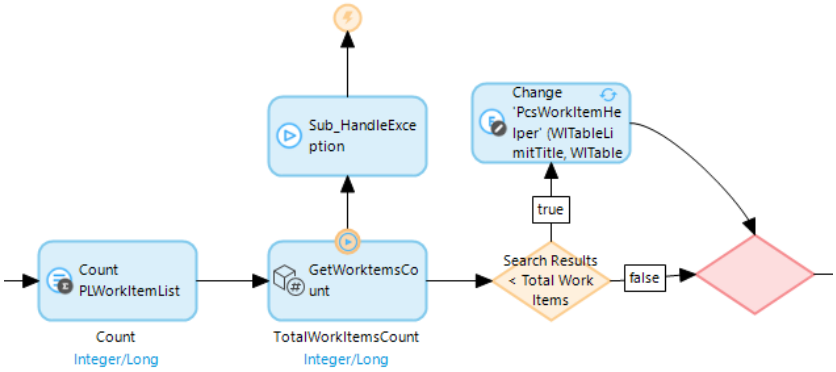
a. Prepare a list of input Work Item attributes to fetch through the query. Use the constants of type **WORKITEM_<field name>_FIELD** defined in the Configuration folder of the Polarion Connector. Microflow Sub_GetWorkItemFields creates such a list of fields with Work Item type, status, project, id, title, and severity.



b. Select the appropriate Polarion service microflow, microflow **QueryWorkItemsLimited** in this case, from Toolbox and drag it to your microflow. Microflow QueryWorkItemsLimited calls the Polarion service **queryWorkItemsLimited** and returns a list of PLWorkItem. The input parameters needed for this microflow are the Lucene query string, sort criteria, list of fields to fetch, and the limit on the number of Work Items to fetch. The constant PCS_WORKITEMS_RESULTS_LIMIT sets the default limit of Work Items to retrieve from the query to 500. Set the error handler and trigger an error event to be handled in the calling microflow. The inputs needed and the extended microflow are shown in the graphic below.



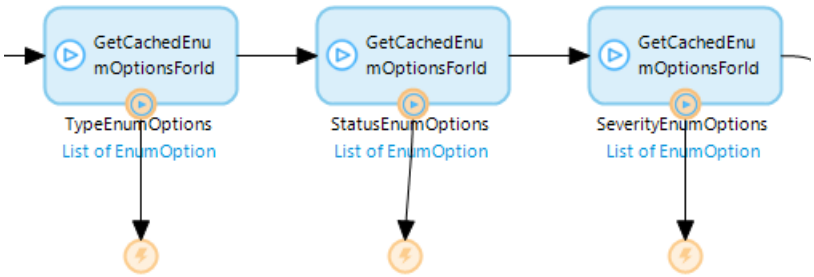
c. Count the number of Work Items returned by the limited query and compare it with the actual number of Work Items present in Polarion project's database. Warn the user if more exist in the Polarion database than what was returned by the query. Make use of Polarion service microflow **GetWorkItemsCount** from Polarion Connector category of the Toolbox to get a total count of the number of Work Items existing in Polarion database. This microflow internally calls the Polarion Service **getWorkItemsCount** to which the Lucene query string is passed. Also, don't forget to do proper error handling as mentioned in the previous steps. The extended microflow now appears similar to the graphic below.



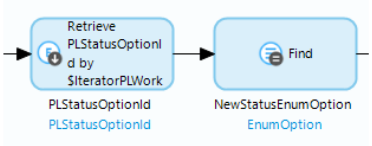
d. Next task is to extract all the pertinent information from the list of PLWorkItem and its associations to create an equivalent list of PcsWorkItem. For this, the sub-microflow Sub_CreateWorkItemTableData performs these functions:

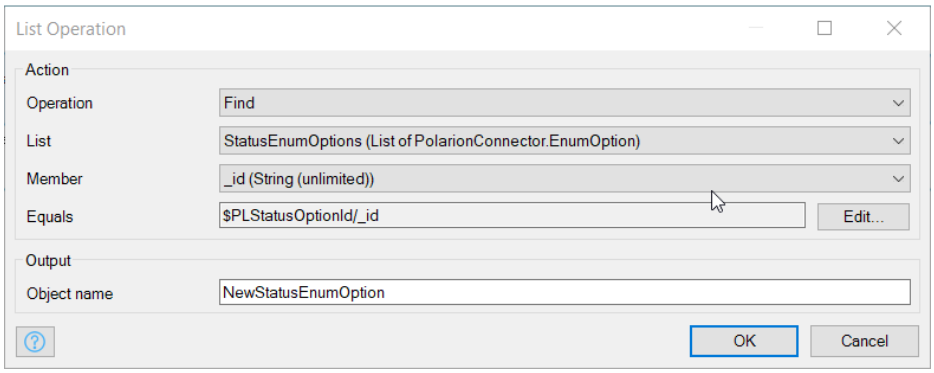
   i. Accumulates information on create, read, modify, and delete permissions by calling Polarion service microflows **CanCreateInstances**, **CanReadInstance**, **CanModifyInstance**, and **CanDeleteInstance**.

   ii. Creates a cache of enumeration options for Type, Status and Severity by calling a sub-microflow **GetCachedEnumOptionsForId** which internally calls the Polarion service microflow **GetAllEnumOptionsForId**. The microflow GetCachedEnumOptionsForId has an in-built logic to first check if enumeration options exist in the session cache. If found, it returns the cached data otherwise it fetches data from Polarion and caches it.

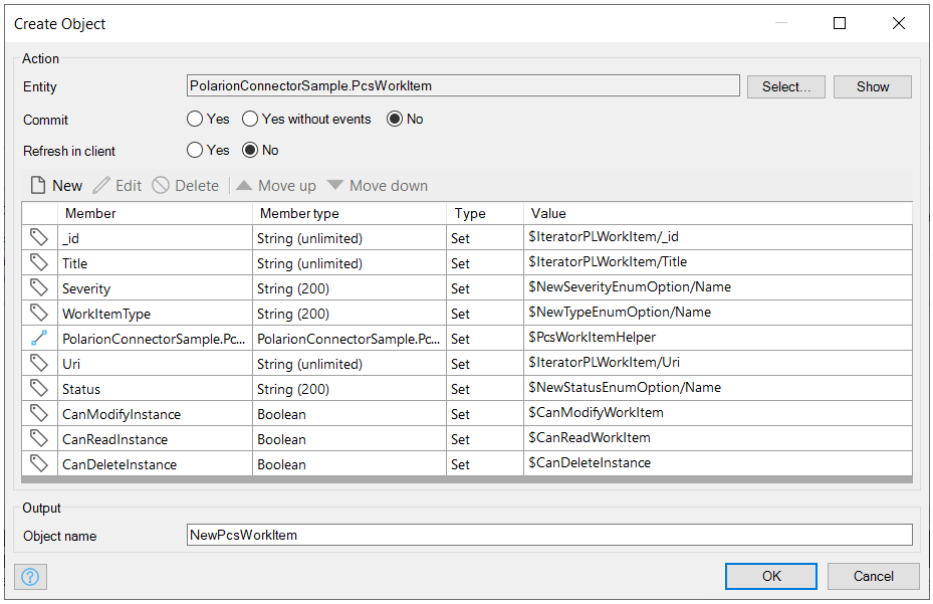


iii. Extracts enumeration option information for each of the PLWorkItem by matching their assigned enumeration option ids to the elements in the cache (generated in the previous step). This is repeated for each of the enumeration option types - Type, Status, Severity. Below image takes Status's enumeration option as an example and gives details of the related microflow activities.

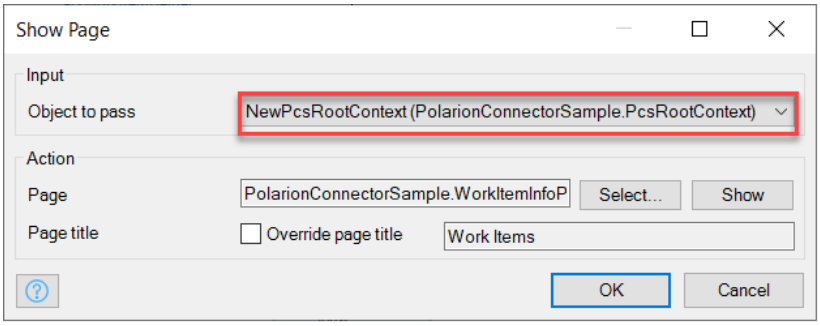


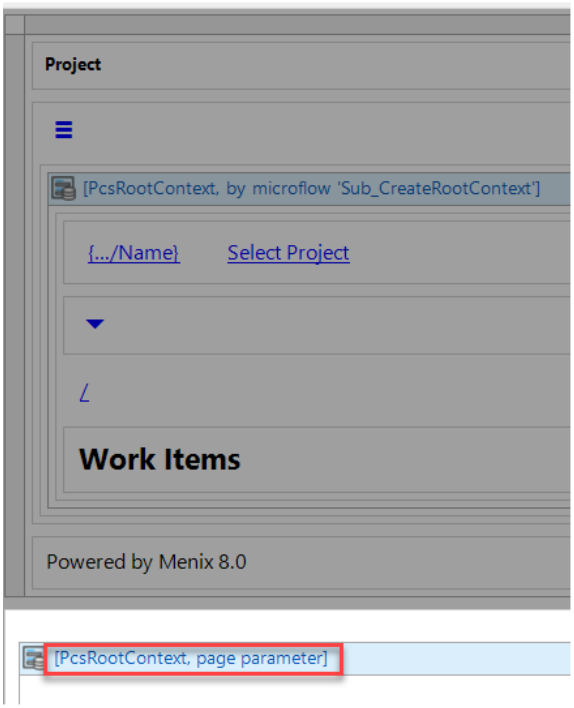


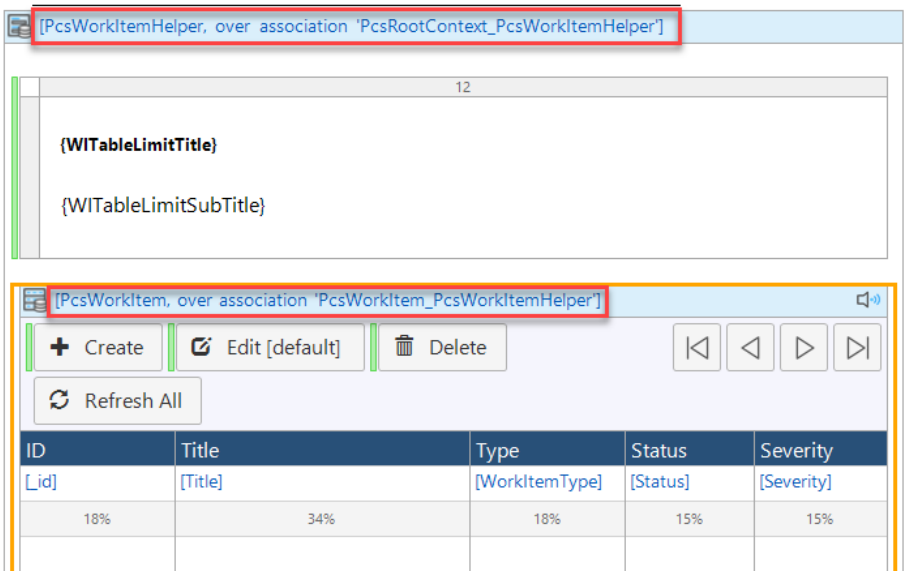
iv. Creates a PcsWorkItem object for each of the PLWorkItem by assembling all the relevant information to be processed by the UI pages. This includes the attribute information needed for display purposes on pages, for example Id, Title, and names (labels) of the enumeration option types - Type, Status, Severity.



4. At this point, you have all the Work Items returned by the query and you also have all their attribute information in the form of a list of PcsWorkItem, ready to be displayed on the WorkItemInfoPage_Overview page. Note that the object passed in as page parameter is PcsRootContext. Upon page initialization, PcsRootContext's associations are navigated (PcsRootContext → PcsWorkItemHelper → PcsWorkItem) through nested data views to access the list of PcsWorkItem and displayed in the data grid.


Show Page
Input
Object to pass
NewPcsRootContext (PolarionConnectorSample.PcsRootContext)
Action
Page
PolarionConnectorSample.WorkItemInfoP
Select...
Show
Page title
Override page title
Work Items
OK
Cancel

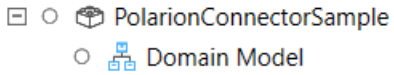

Project
[PcsRootContext, by microflow 'Sub_CreateRootContext']
{.../Name}
Select Project
Work Items
Powered by Menix 8.0
[PcsRootContext, page parameter]


[PcsWorkItemHelper, over association 'PcsRootContext_PcsWorkItemHelper']
12
{WITableLimitTitle}
{WITableLimitSubTitle}
[PcsWorkItem, over association 'PcsWorkItem_PcsWorkItemHelper']
Create
Edit [default]
Delete
Refresh All
ID
Title
Type
Status
Severity
[_id]
[Title]
[WorkItemType]
[Status]
[Severity]
18%
34%
18%
15%
15%

Example: Extend the domain model to call a service to create a Work Item

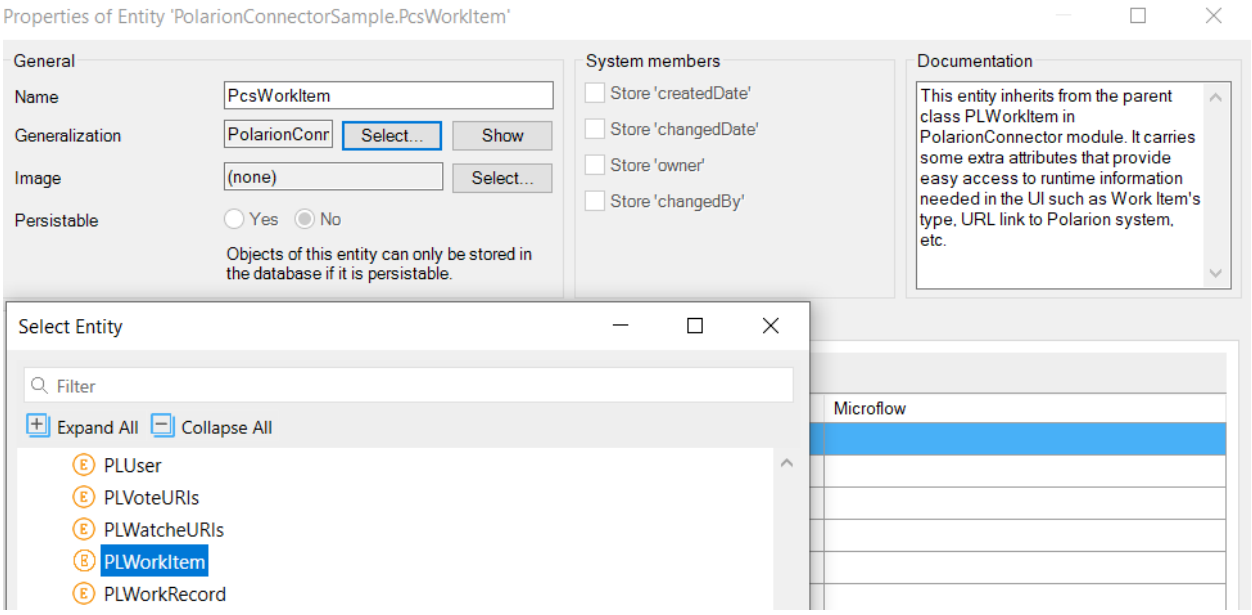
This example takes the Create Work Item feature of the Sample Application and illustrates the composition of the main microflow Act_PcsWorkItem_CreateWorkItem which is the logic engine behind this feature.

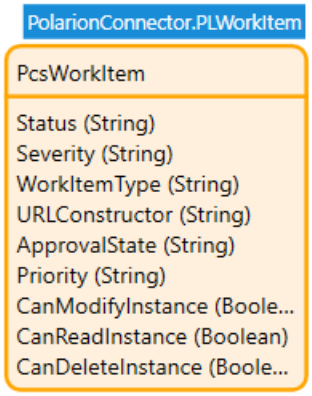
1. Create a new module that represents your own business area. Skip this step if the module already exists.



2. Define the Domain Model.

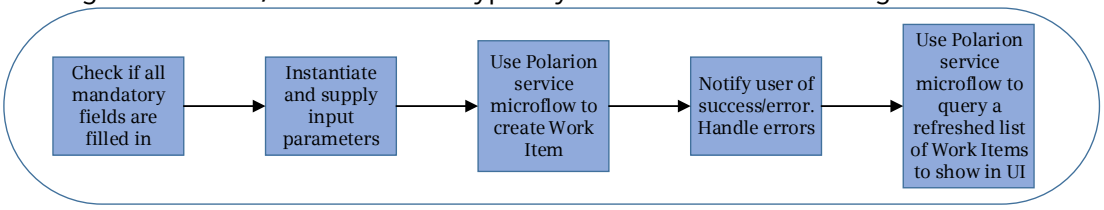
   Create a new non-persistent entity PcsWorkItem by selecting Generalization as **PolarionConnector.PLWorkItem**. Skip this step if the entity already exists.
   By the virtue of inheritance, PcsWorkItem will have access to all the same associations that were defined for the parent PLWorkItem in PolarionConnector's Domain Model. As a good practice, ensure that any new attributes or associations of the PcsWorkItem entity reflect the appropriate business intent.



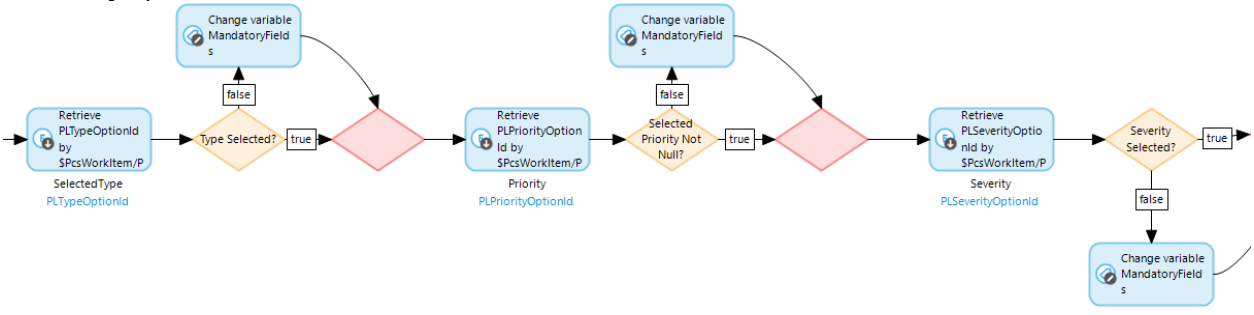


Tip: You can find information on out-of-the-box Polarion Work Items, its types and properties in the Configure Work Items and Polarion SDK documentation.

3. Plan your microflow and decide what services you want to use. The Polarion services microflows are available in the Toolbox under the Polarion Connector category. For creating a Work Item, the microflow typically consists of the following activities:

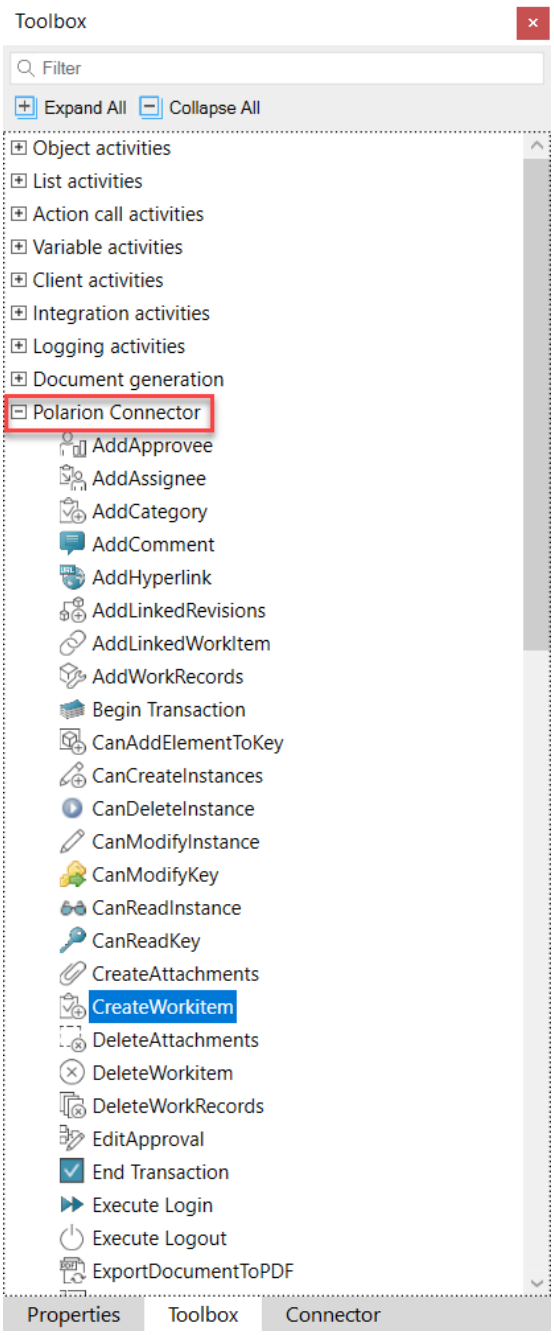


4. Ensure inputs for the mandatory fields such as Title, Type, Severity and Priority have been provided by the user. These are needed to create a Work Item and they can be obtained through a UI input page like CreateWorkItemPage_New included in the Sample Application. These mandatory inputs are retrieved through associated entities of PcsWorkItem (inherited from PLWorkItem) – PLTypeOptionId, PLSeverityOptionId, PLPriorityOptionId.



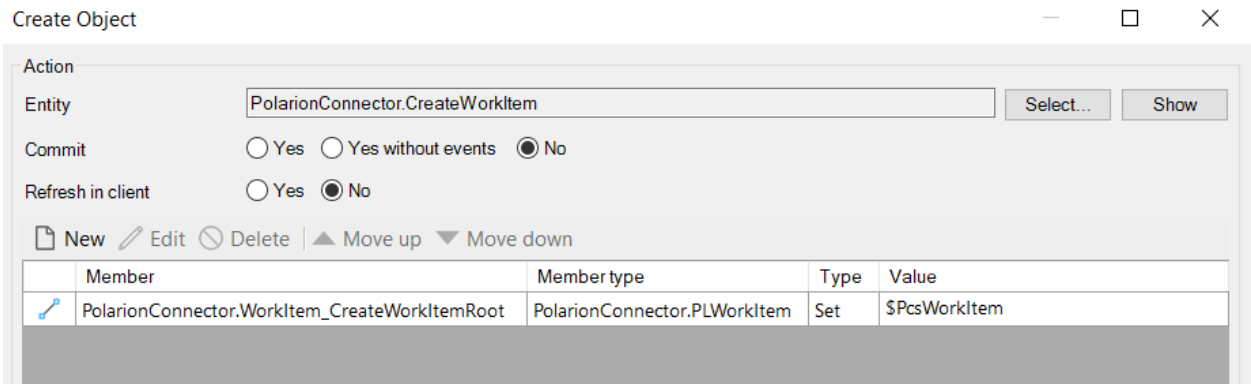
5. Specify any special processing if needed. For example, Description field's content needs to be treated as regular text instead of markup text.

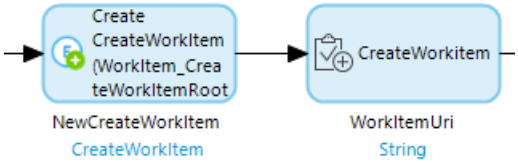
6. Select the appropriate Polarion service microflow from Toolbox and drag it to your microflow. For example, you can use the **CreateWorkItem** microflow under the Polarion Connector category to create a Work Item.



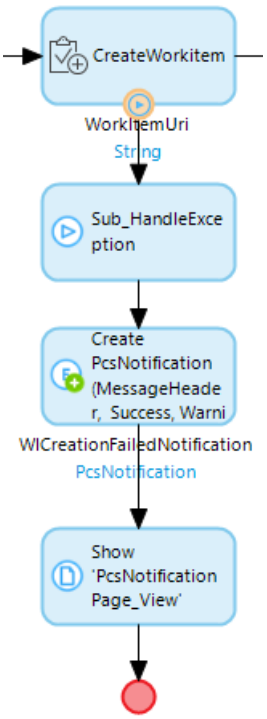
7. In your microflow, specify the input parameters that the Polarion service microflow will use. The CreateWorkItem microflow requires only one input parameter:

- An input object of entity *PolarionConnector.CreateWorkItem* which has an association called *WorkItem_CreateWorkItemRoot* to *PLWorkItem*. Create input object of type CreateWorkItem with its association set to PcsWorkItem. PcsWorkItem carries all the information gathered from the input page CreateWorkItemPage_New which is eventually passed on to the Polarion service **createWorkItem**. Your microflow is now extended as follows:

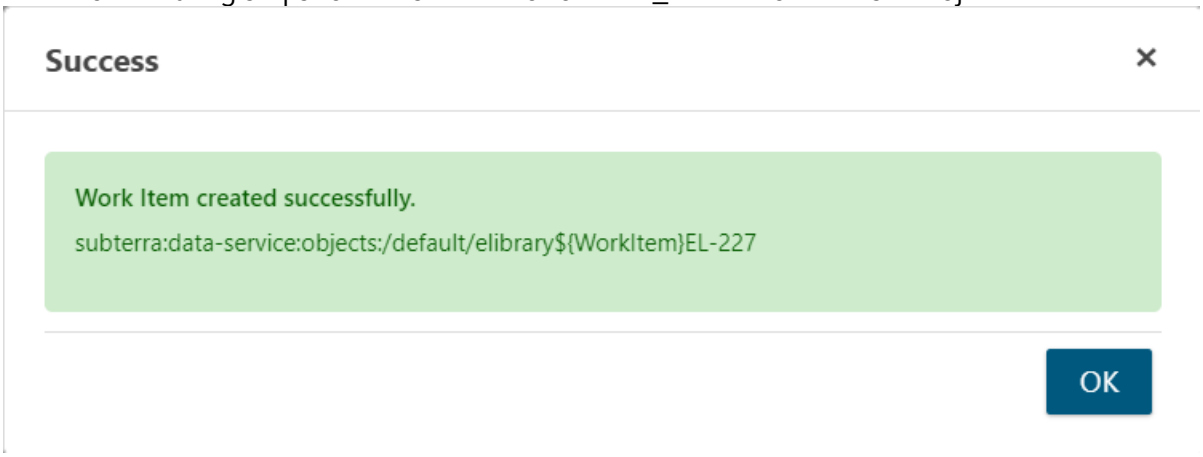




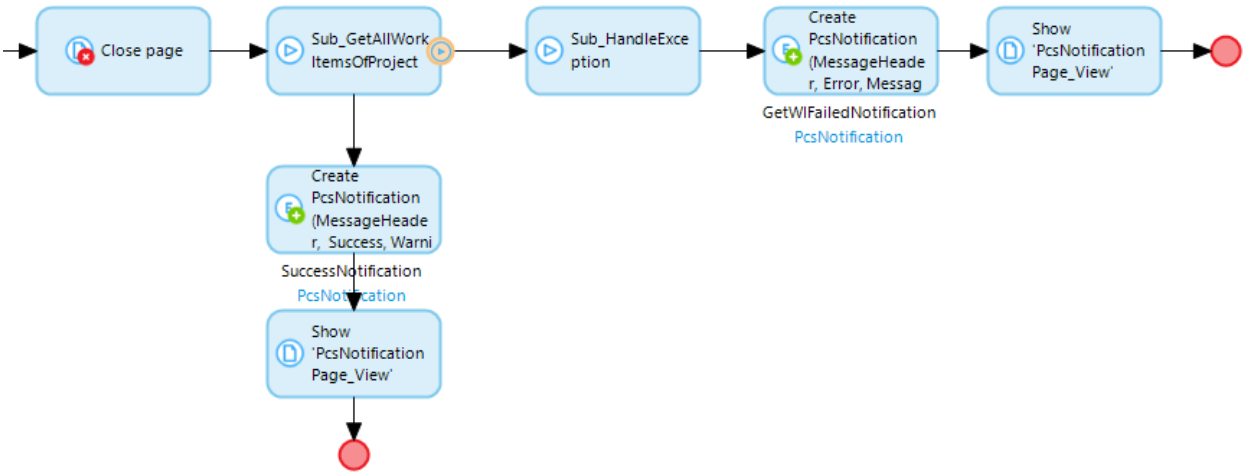
8. In the previous step, a successful CreateWorkItem microflow returns a valid Work Item URI returned from the Polarion service createWorkItem but this is also where error handling should be done to cover errors and exceptions thrown by this service. Set Sub_HandleException microflow as the error handler of type Custom without rollback for CreateWorkItem microflow. Refer to How to handle errors to complete the steps. Your microflow appears as follows:



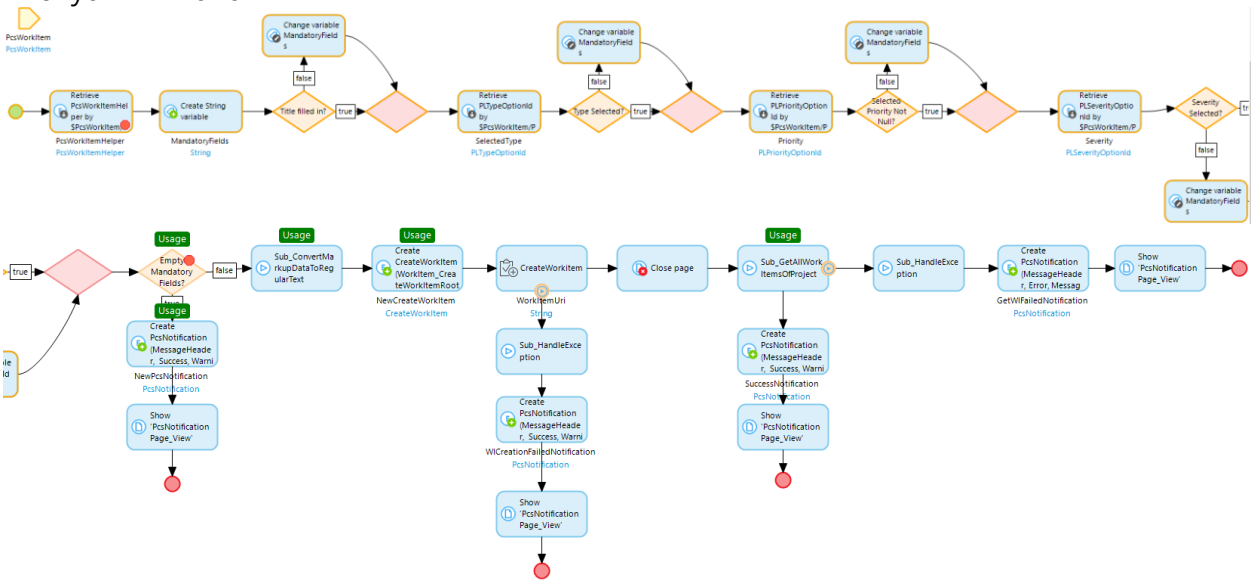
9. Display a success message and a refreshed list of Work Items in the client UI. For this, query Polarion to fetch an updated list of Work Items. A new sub-microflow Sub_GetAllWorkItemsOfProject does this job. This sub-microflow is very much a part of the microflow Act_OpenWorkItemInfoPage explained in the preceding topic. Complete the error handling steps for the sub-microflow Sub_GetAllWorkItemsOfProject.



Your microflow appears as follows:



10. Test your microflow.

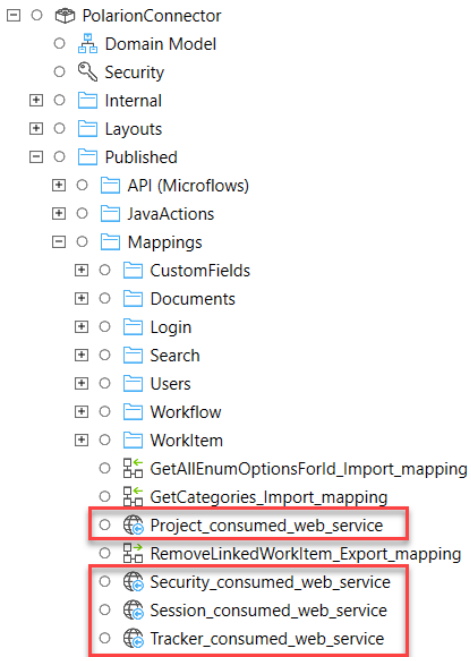


Using Polarion services not included in Polarion Connector

The Polarion Connector supports a library of Polarion web services through microflows written to call these services. For a list of supported services, see Polarion Connector services. If the web service you intend to use is not in this list, you can follow the steps described below in this section to easily write a new microflow in your module and call the desired service. The steps described here are also illustrated by taking the Polarion web service **queryWorkItemUrisLimited,** currently not in Polarion Connector's list of supported services, as an example. 'queryWorkItemUrisLimited' service queries for Work Item URIs with a given limit. Refer to **Polarion Web Services API** for a complete list.

I. Examine the Consumed Web Service Definition

Check if a Consumed Web Service definition for your intended web service exists in Polarion Connector's *Published* → *Mappings* folder. Currently, web service definitions for ProjectWebService, SecurityWebService, SessionWebService and TrackerWebService already exist in the Polarion Connector. If a newer one is desired, create a new Consumed Web Service and import its WSDL (Web Services Definition Language) from **Polarion Web Services API**. The example web service queryWorkItemUrisLimited is part of the TrackerWebService definition and therefore a new one is not needed.

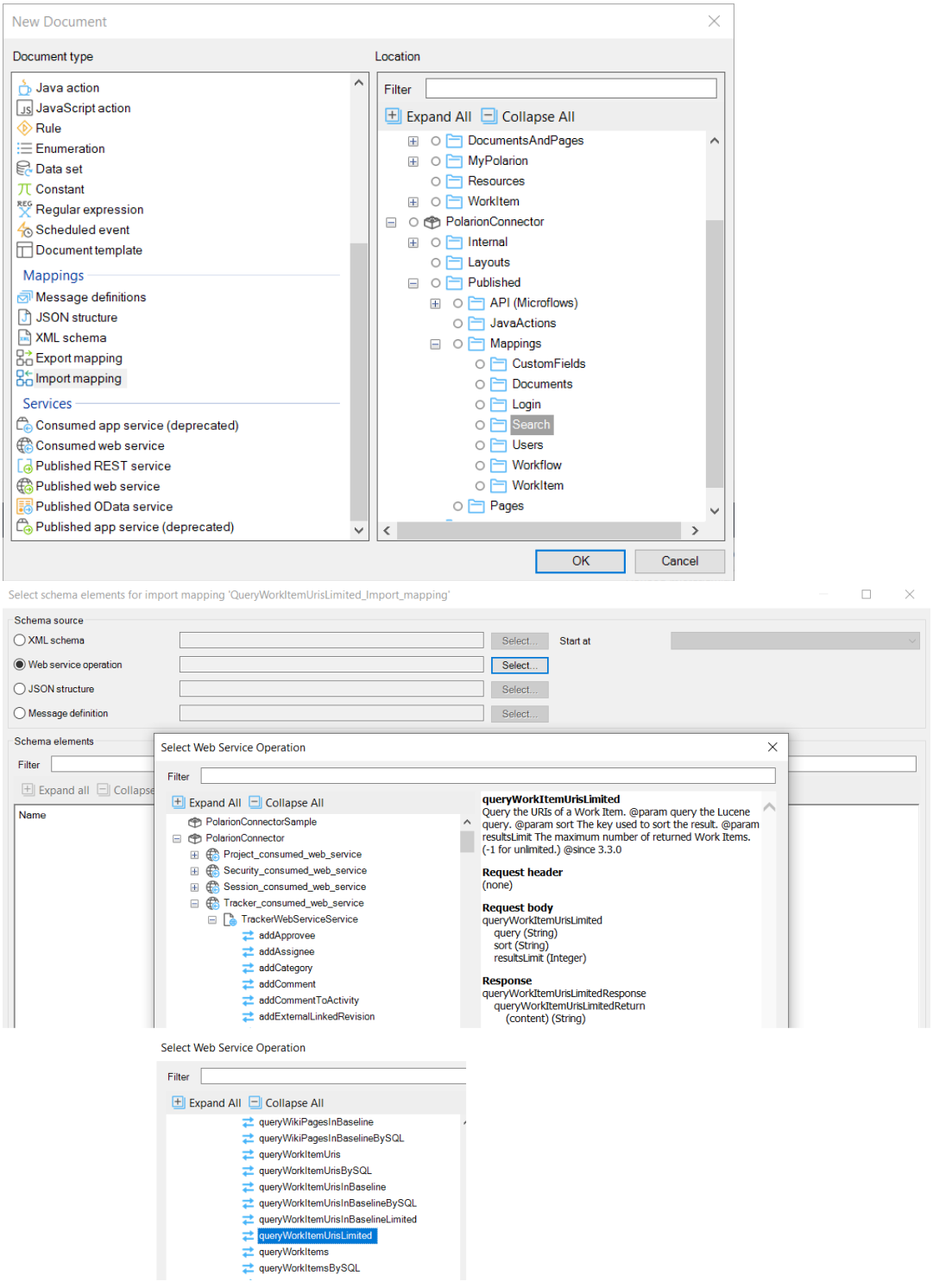


II. Create Import Mappings

Examine the intended web service's return parameters and create an import mapping to map the web service's SOAP Response to your Domain Model's entities.

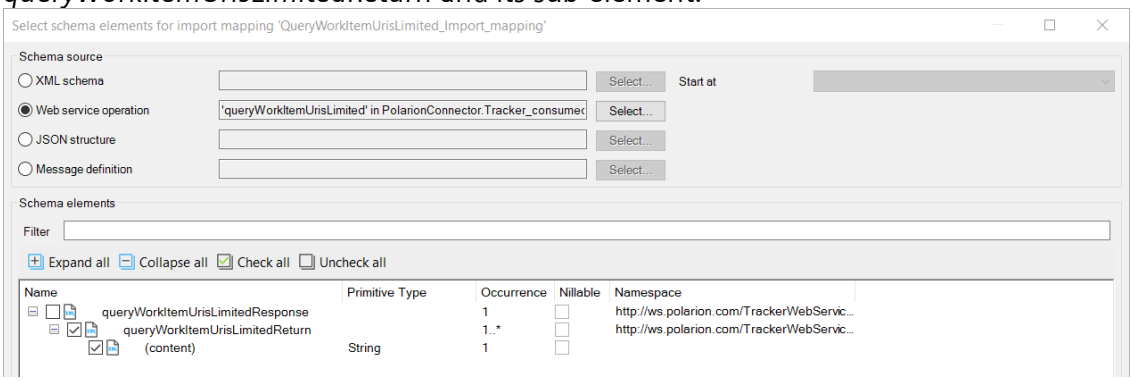
a. Check if an appropriate import mapping exists to map the web service's SOAP Response to your Domain Model's entities. Existing import mappings can be found in sub-folders under Polarion Connector's *Published* → *Mappings* folder.

b. If none of the existing mappings suits, create a new import mapping starting with the command *File* → *New Document...* → *Import mapping*. Toggle on *Web service operation* selection and expand the applicable consumed web service definition to select your web service of interest. The images below show the example case of selecting queryWorkItemUrisLimited service for creating import mapping.

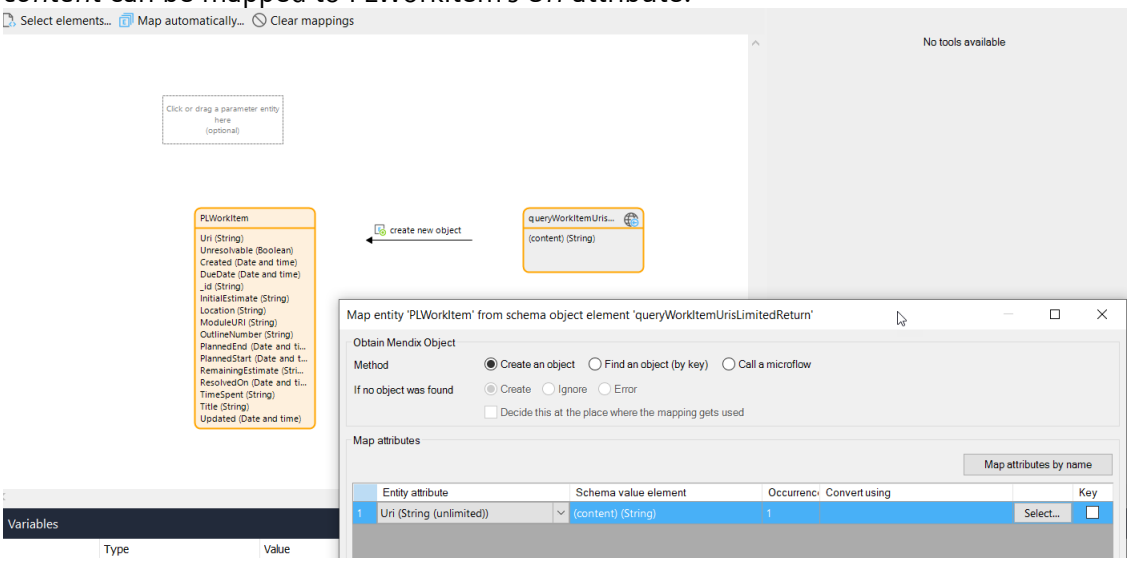


c. From the *Select schema elements for import mapping* ... dialog box, you can optionally select the entities and associations to map as return values of the web service. For the

example web service queryWorkItemUrisLimited, we need to map only the *queryWorkItemUrisLimitedReturn* and its sub-element.



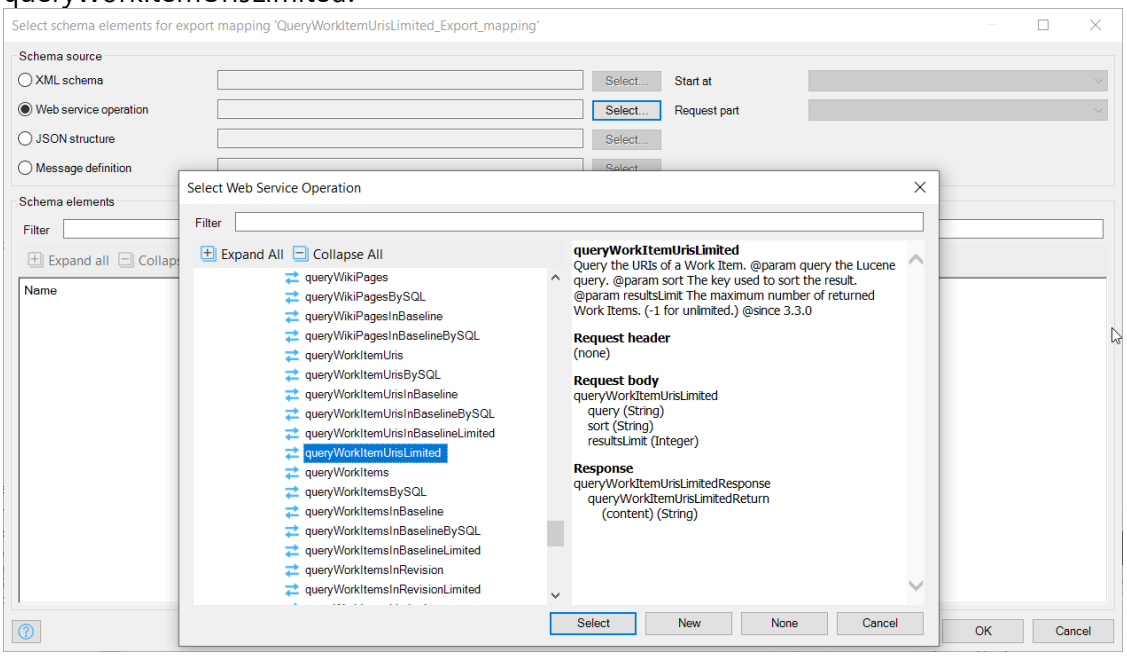
d. In the editor, double click the selected elements (on the right of the arrow) and select suitable existing entities from your Domain Model in the *Select Entity* dialog box. You also have the option of either creating new entities and then mapping them or using the *Map automatically...* option from the editor's toolbar. Polarion Connector's Domain Model is quite extensive, and, in many cases, its pre-existing entities and associations may prove to be sufficient for your import mapping needs. In the example case, *queryWorkItemUrisLimitedReturn* can be mapped directly to PLWorkItem entity and *content* can be mapped to PLWorkItem's *Uri* attribute.

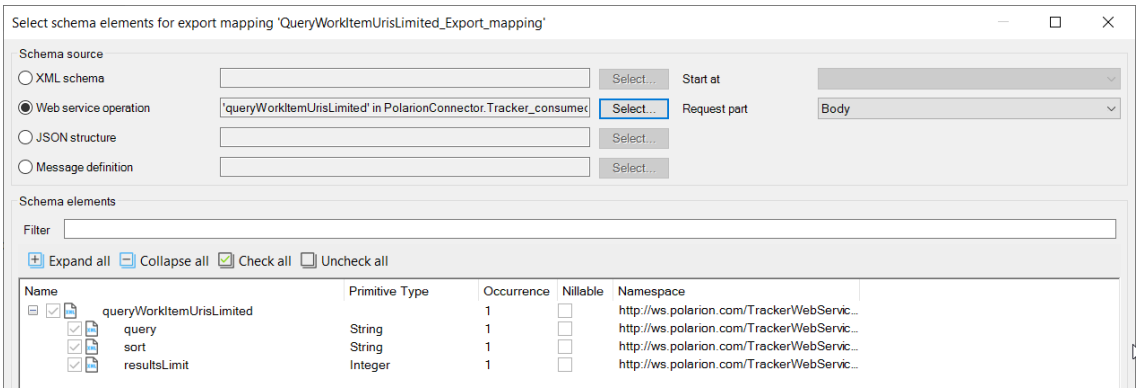


III. Create Export Mapping

Examine the intended web service's input parameters and create an export mapping to generate an XML for the SOAP request.

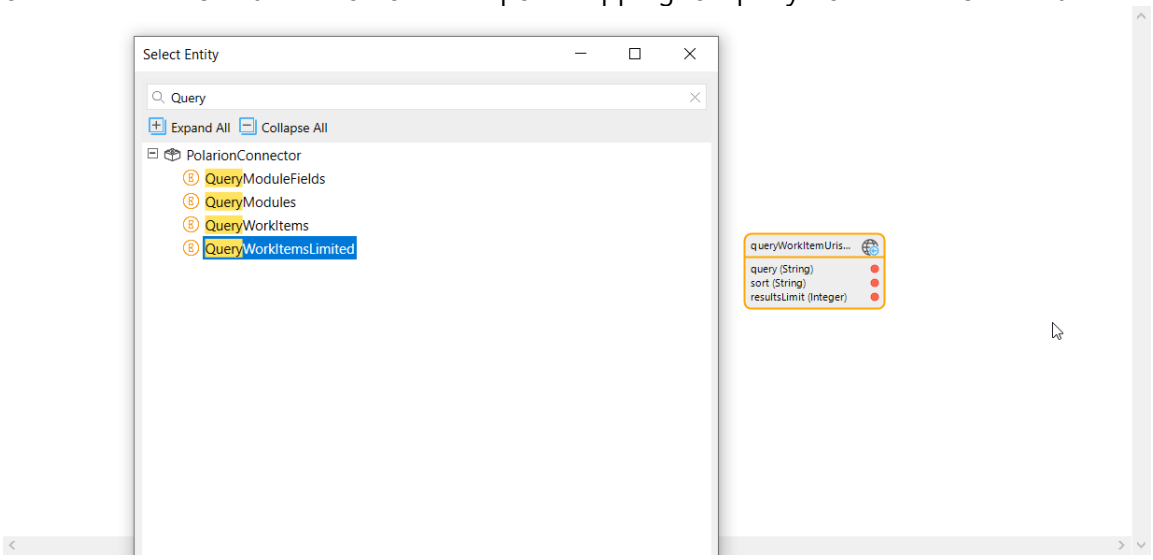
a. Start with the command *File → New Document... → Export mapping*. Toggle on *Web service operation* selection and expand the applicable consumed web service definition to select your web service of interest. Keep the default selection of all elements in the *Select schema elements for export mapping ...* dialog box to complete this step. The images below show the example case of creating export mapping for the web service queryWorkItemUrisLimited.

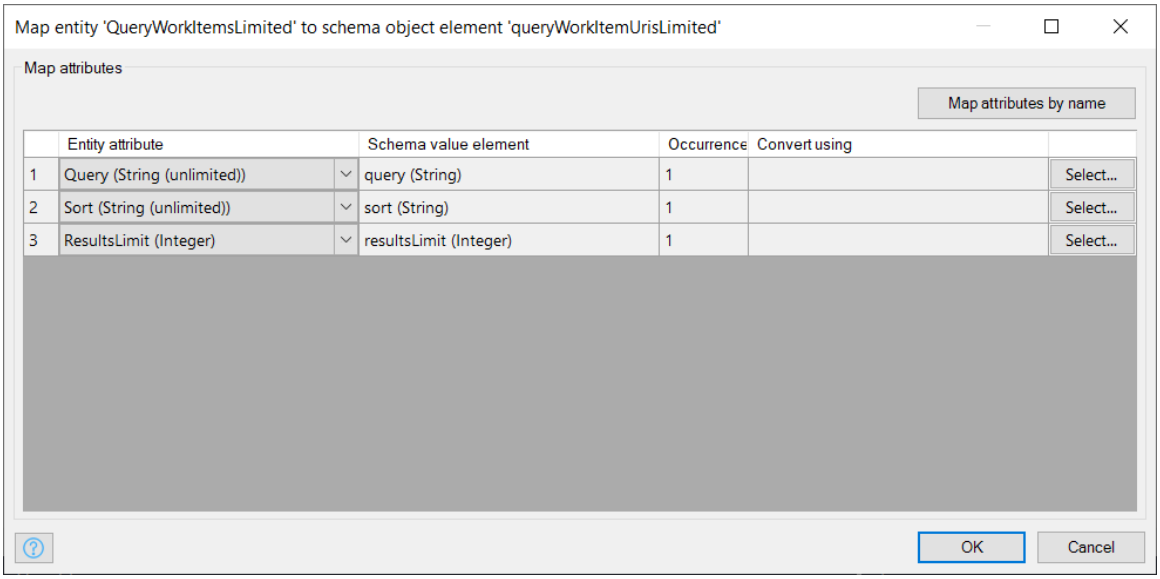


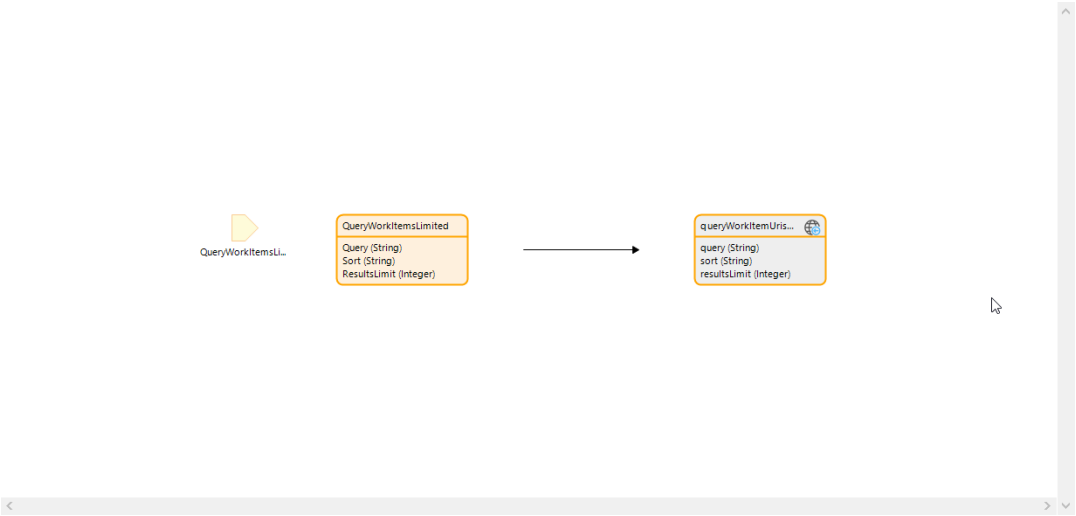


b. In the editor, double click the selected elements (on the right of the arrow) and select a suitable existing entity from your Domain Model in the *Select Entity* dialog box. You also have the option of creating new entities and then mapping them or using the *Map automatically...* option from the editor's toolbar. Polarion Connector's Domain Model is quite extensive, and, in many cases, its pre-existing entities and associations may prove to be sufficient for your export mapping needs. In the example case, there is a pre-existing entity *QueryWorkItemsLimited* which nicely fits the export mapping since its attributes match one-to-one with the input parameters of the service

queryWorkItemUrisLimited. Following images show the selection of this entity, mapping of the attributes and creation of the export mapping for queryWorkItemUrisLimited.







IV. Use the 'Call web service' activity

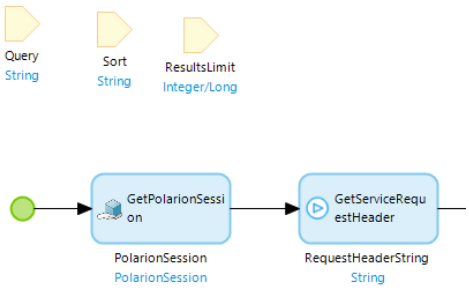
Mendix's 'call-web-service' activity can be used to call an imported web service operation. Compose a wrapper microflow to stage the required input needed for 'call-web-service' activity to call the desired Polarion service. This microflow is also responsible for processing the response received back from the web service and returning it to the calling microflow. Such microflows have also been referred to as "Polarion services microflows" in this document.

a. Start your microflow by instantiating all the input parameters needed for calling the Polarion web service. For the example case of queryWorkItemUrisLimited service, the input parameters to be passed in are: Lucene query string, sort criteria and the number of Work Items to fetch.

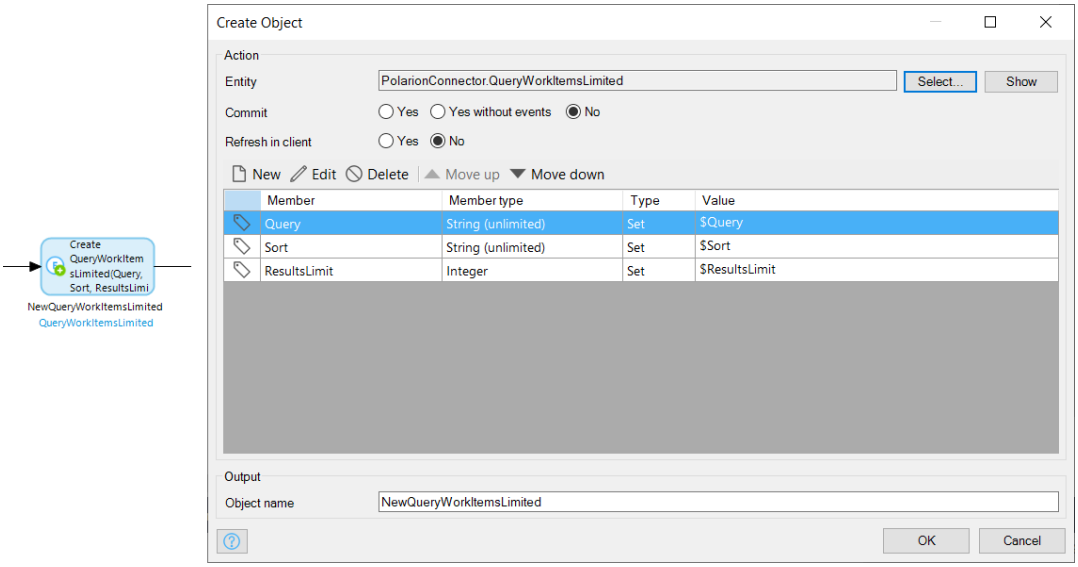
b. Initialize Polarion session information. This needs to be done for each Polarion service microflow you write. There are two sub-microflows available in Polarion Connector responsible for handling this:

   I. GetPolarionSession - Retrieves the existing Polarion Session from Mendix database. Creates a new Polarion Session if none is found.

   II. GetServiceRequestHeader - Creates a web service request header string with Polarion login cookie which is used to populate the SOAP Request Header template of the 'call-web-service' activity.

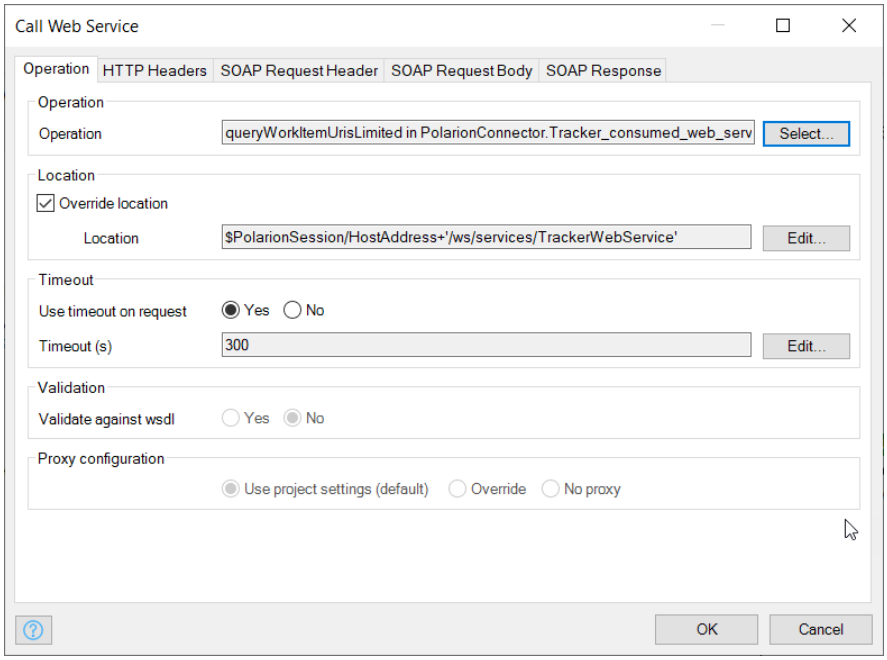


c. Create the export mapping object with its attribute values initialized to the input parameters. For the example case, create a QueryWorkItemsLimited object and set its attributes as shown.

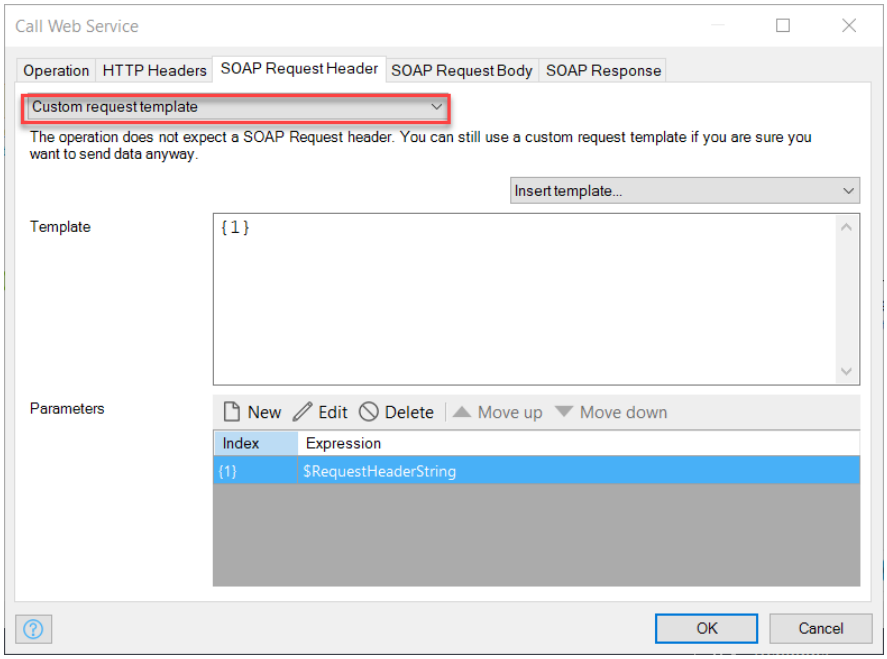


d. Drag and drop the 'call-web-service' activity in to your microflow and start populating the information in its various tabs:

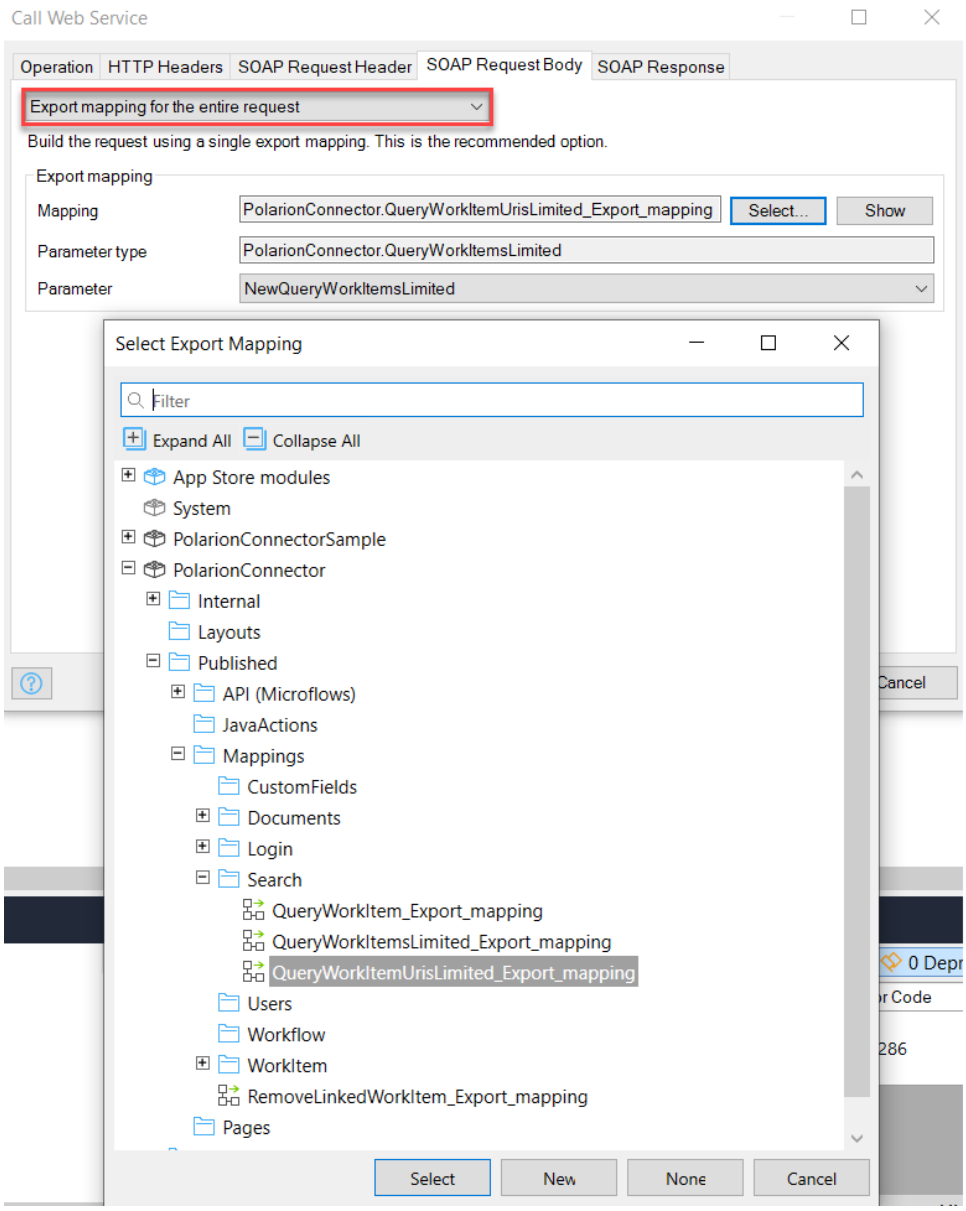
I. Operation tab: Select the applicable web service in the Operation field. For the Location field, toggle on the 'Override location' option and set Location to the URL address of Polarion server's web service operation. The URL is an expression composed of a concatenation of Polarion server's host address and a standard default path to the web service. The host address is extracted from the Polarion session object returned by the earlier GetPolarionSession call. Set the Timeout setting to the default of 300 seconds to throw an exception when the web service takes too long to respond after connecting. The image below shows the Operation tab for the example web service queryWorkItemUrisLimited.



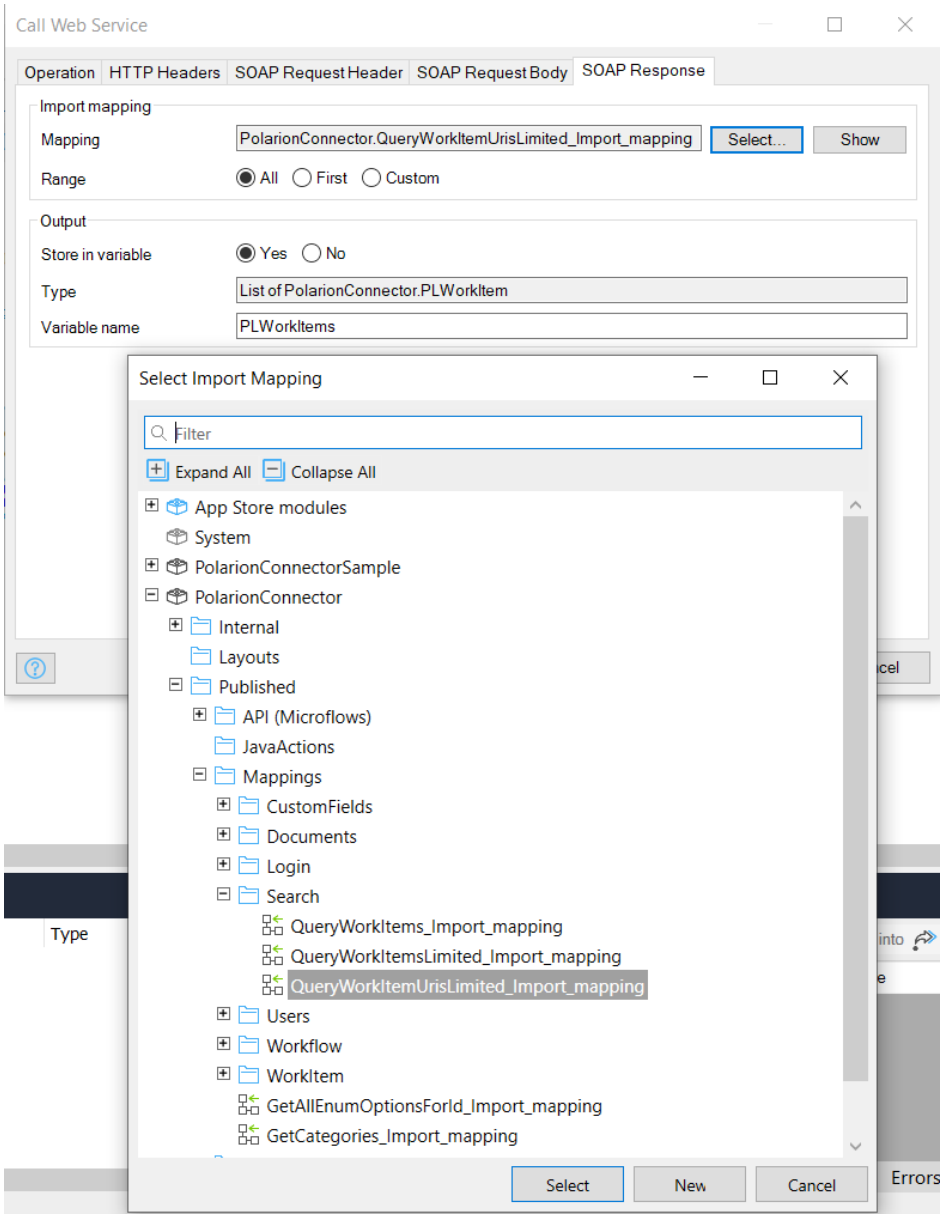
II. SOAP Request Header tab: Use the 'Custom request template' option. Define a parameter for the header string generated by the earlier GetServiceRequestHeader call. This string has the XML structure of the SOAP envelope in plain text and it also has the login cookie embedded in it. Set the Template field to the parameter you just defined which in turn points to the header string which has all the required information to establish an authenticated connection. The image below shows the SOAP Request Header tab for the example web service queryWorkItemUrisLimited.



III. SOAP Request Body tab: Select the option 'Export mapping for the entire request'. In the Mapping field, you can choose an existing export mapping pre-created to use for this request. This was explained earlier in the Create Export Mapping section. The 'Parameter Type' field is automatically set to the entity based on the export mapping selected. If applicable, the Parameter field can be set to an object or list that you want to use as parameter for the mapping. The image below shows the SOAP Request Body tab for the example web service queryWorkItemUrisLimited.

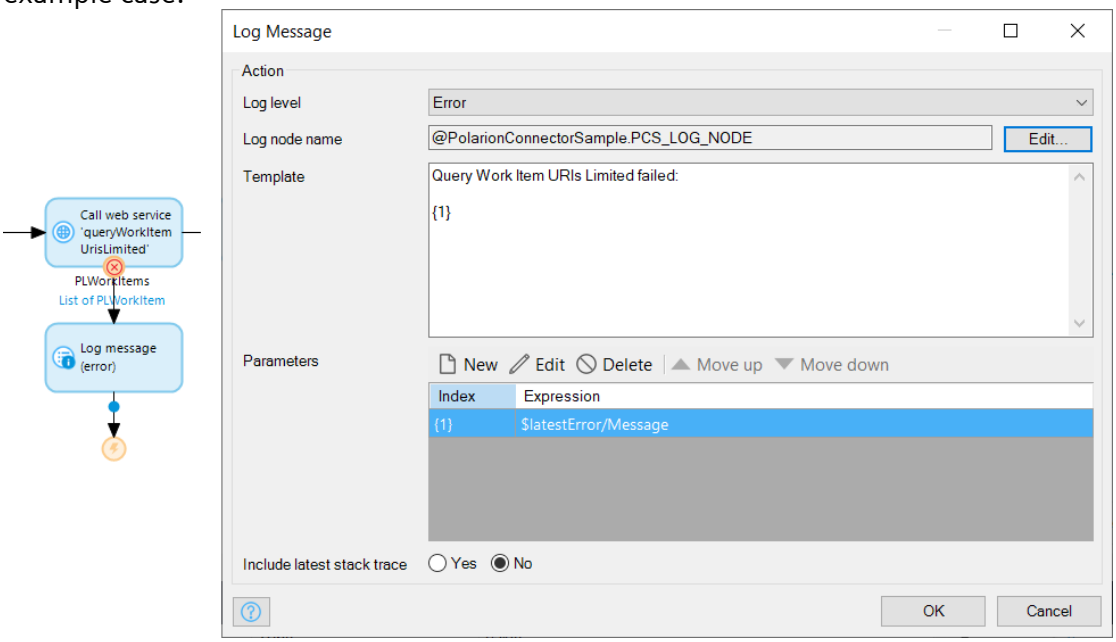


IV. SOAP Response tab: In the Mapping field, you can choose an existing import mapping pre-created to use for this request. This was explained earlier in the Create Import Mappings section. The Type field is automatically set to an entity or a list based on the import mapping selected. Assign a variable in the 'Variable name' field to store the result of the 'call-web-service' activity. The image below shows the SOAP Response tab for the example web service queryWorkItemUrisLimited.

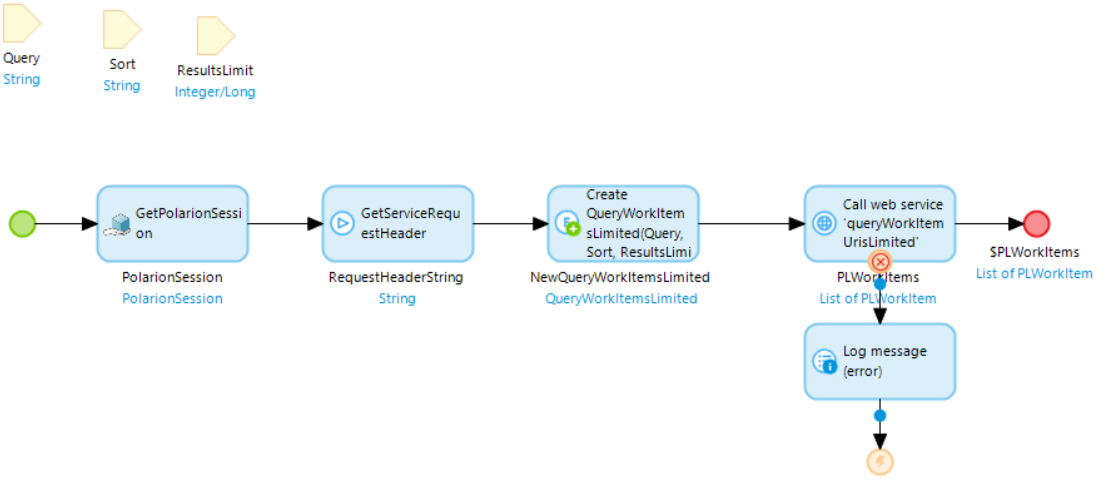


e. Set error handler of type *Custom with rollback* for 'call-web-service' activity to handle all errors that occur during the processing of the service request. This includes networking errors (HTTP errors connecting to the Polarion server), session time-out, and parsing errors. These errors can optionally be logged to a custom log node for your module using the Log Message activity. The errors are processed as an Error Event, and the calling microflow must also set an error handler on the Polarion service microflow to process the returned error and must also display the $latestError/Message and then exit the microflow. Below images show the error handling and details of Log Message for the

example case.



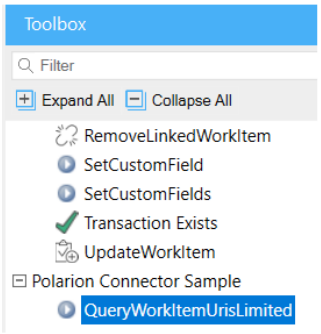
f. Optionally, perform any further processing per your needs before ending the microflow by returning applicable variable, object, list or nothing. The complete Polarion service microflow for the example case of queryWorkItemUrisLimited service is shown in the image below.

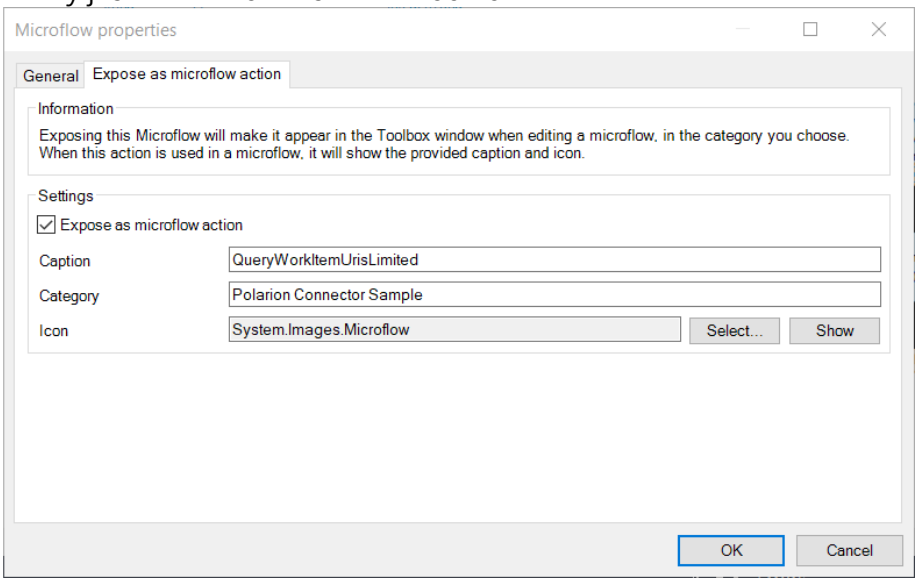


g. Test your microflow.

V. Expose as Microflow Action

Optionally, by right-clicking in the microflow editor and selecting *Properties* and then further selecting the *Expose as microflow action* option, your microflow will be exposed in the *Toolbar* under a category of your choice with an icon of your choice. By doing so, your microflow will become a drag and drop activity just like the others in the Toolbox.

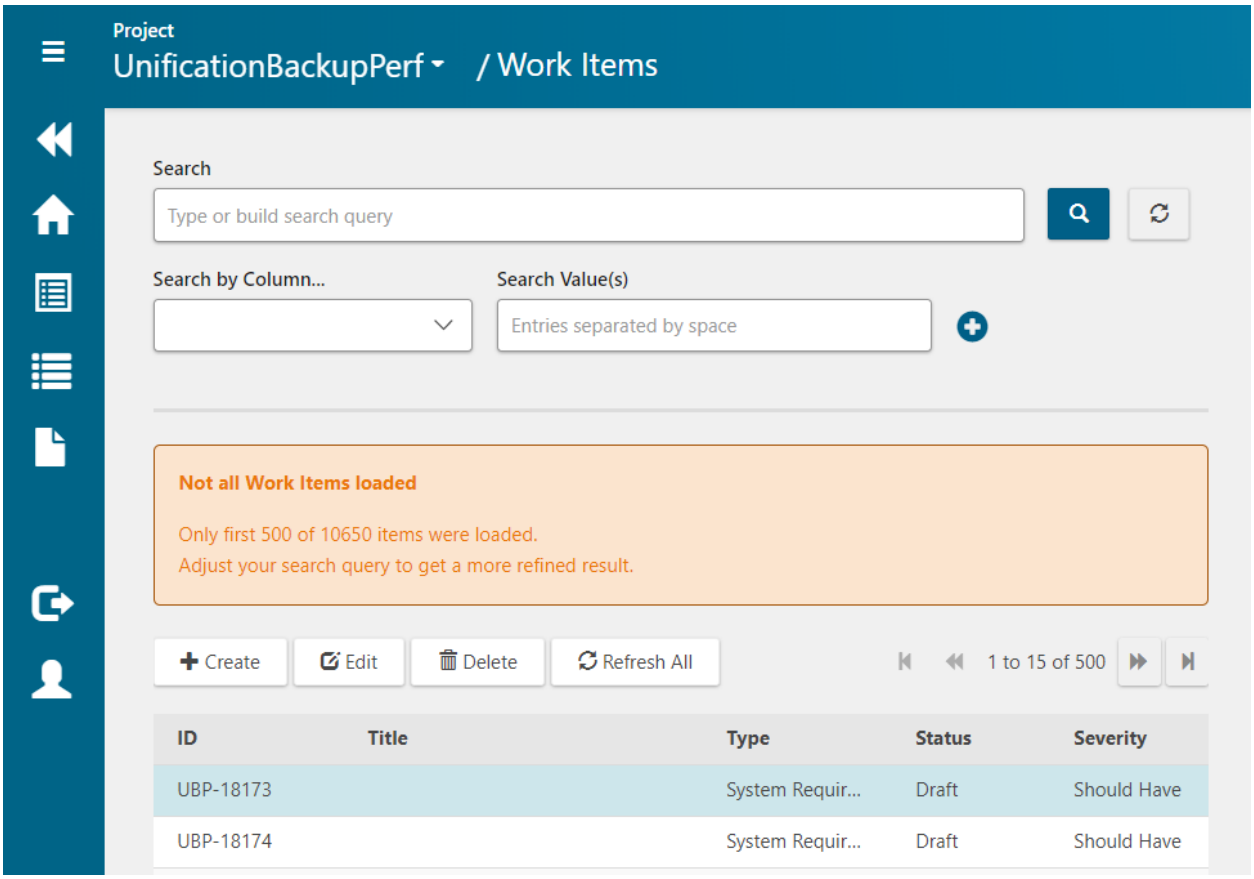




Configuration Settings

Query Limits

The Polarion service microflow QueryWorkItemsLimited expects a number to be passed in which sets a limit on the number of search results returned. As an example, the constant PCS_WORKITEMS_RESULTS_LIMIT in the Sample Application sets a default limit of 500 on the number of Work Items to be retrieved wherever the microflow QueryWorkItemsLimited is called. This limit can be adjusted up or down according to the needs of the customizers. When this limit is hit by the microflow, a warning message is displayed (see the image below). It informs the user that the number of Work Items existing in the selected project's database are more than the query limit and that the user should resort to a refined query if their Work Item(s) of interest is not found. In the case of Sample application, the user can refine the query by providing a search criteria in the search fields.



Runtime Settings

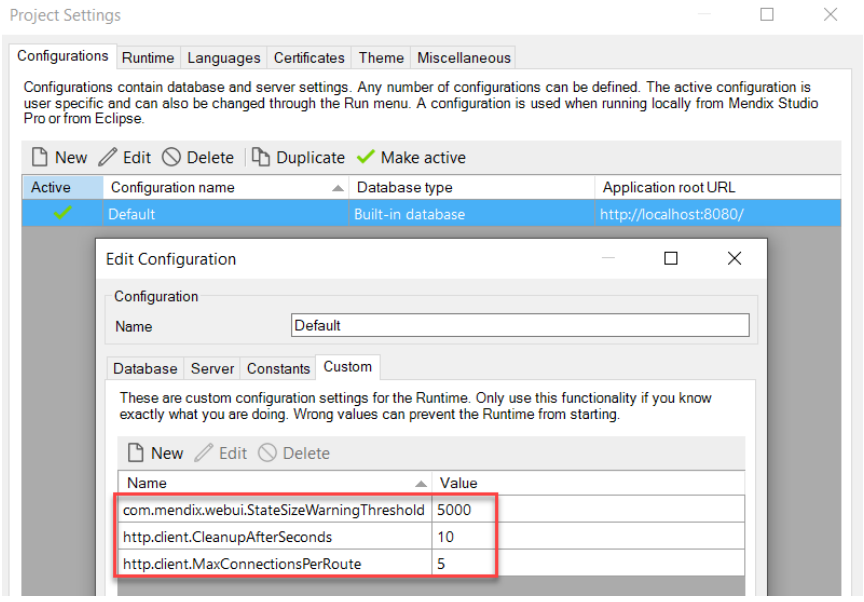
Custom runtime settings as described in below mentioned sections are included in the Sample Application. These settings can be changed according to your needs by going to Project Settings → Configurations → Edit → Custom.

Http Client Settings

A user may experience a hang of the Sample Application if multiple consecutive exceptions are encountered from Polarion web services within a short period of time. This is a known issue with Mendix 8.5 where the Http client is not closed (till it times out) after receiving an error from a published web service. To mitigate this issue, runtime settings of *http.client.CleanupAfterSeconds* is set to 10 seconds and *http.client.MaxConnectionsPerRoute* is set to 5 in the Sample Application. This allows the Sample Application to have 5 Http clients per session and close any unused Http clients after 10 seconds. These settings can be adjusted up or down by customizers according to their expected application usage patterns. More information about these settings is described in the online documentation at Runtime Customization.

State Threshold Level

Sample Application pages may need to process a large amount of non-persistable objects to display the data. By default, Mendix Runtime will log a warning message to console in case the number of objects exceeds the State Threshold Level. This level can be configured with the custom setting of *com.mendix.webui.StateSizeWarningThreshold* which is set to 5000 to allow processing of larger amounts of objects and to prevent the warning from being logged too often. This setting can be adjusted up or down by customizers according to their expected application usage patterns. More information about this setting is described in the online documentation at Runtime Customization.



Limited Support

Following are some of the known issues with limited support in Polarion Connector:

1. All custom fields for a Work Item are supported except for specialized custom fields like *Test Steps* and *Feature Selection*.
2. During update of a Work Item, *Linked OSLC Resources* field will not be updated.
3. Most of the information structures linked to a Work Item are available in the Connector except for *Watches*.
4. Where hierarchical recursive data types are encountered, only the first level is processed. For example, if a Work Item has an association to Plan and Plan has recursive association to itself (Work Item -> Plan -> Plan), only the Work Item and its first level Plan is retrieved.
5. Fetching images for enumerations (Work Item type icons, custom enumeration icons, icons for Work Items, Documents, etc...) is not supported.
6. Limited support of Documents, for example - getting Document attachments, working with comments, etc. is not supported.

Appendix A: Polarion Connector services (Microflow APIs)

Microflow 'AddApprovee'

Adds Work Item approvals and changes the status of approval.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with approval data associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AddAssignee'

Adds Work Item assignee(s).

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with assignee data associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AddCategory'

Adds Categories to a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with category data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AddComment'

Adds comments to a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with comment data associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AddHyperlink'

Adds hyperlinks to a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with hyperlink data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AddLinkedRevisions'

Adds linked revisions to a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with linked revision data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AddLinkedWorkItem'

Adds linked Work Items.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with linked work item data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AddPolarionConfiguration'

Creates a new Polarion configuration record in the database

Parameters

Name	Type	Documentation
PolarionConfiguration	PolarionConnector.PolarionConfiguration	Polarion configuration

Return type

Nothing

Back to top

Microflow 'AddWorkRecords'

Creates Work Records to a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with work record data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'AdminLogin'

Shows admin page if valid polarion session exists otherwise shows default home page

Parameters

This microflow has no parameters.

Return type

Nothing

Back to top

Microflow 'BeginTransaction'

Starts an explicit transaction for the current session.

Parameters

This microflow has no parameters.

Return type

Nothing

Back to top

Microflow 'CanAddElementToKey'

This microflow checks if the current user can add elements to the collection at given key of the object. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputKey	String	The key of the field that contains the collection.
InputURI	String	The uri of the given object.

Return type

Boolean

Back to top

Microflow 'CanCreateInstances'

This microflow checks if the current user can create instances of the given prototype in the given project. For all Work Item types, the prototype class is 'WorkItem' (see constant WORKITEM_PROTOTYPE). It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputProjectId	String	The id of the project to check the permission in, null to check global permissions.
InputPrototype	String	The prototype of the objects. For a Work Item, the prototype as set in the global constant WORKITEM_PROTOTYPE should be passed in.

Return type

Boolean

Back to top

Microflow 'CanDeleteInstance'

This microflow checks the Polarion permissions to see if the current user can modify the given object. The URI of the given object is passed in. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputURI	String	The uri of the given object.

Return type

Boolean

Back to top

Microflow 'CanModifyInstance'

This microflow checks the Polarion permissions to see if the current user can modify the given object. The URI of the given object is passed in. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputURI	String	The uri of the given object.

Return type

Boolean

Back to top

Microflow 'CanModifyKey'

This microflow checks the Polarion permissions to see if the current user can modify the field with given key of the object. The URI of the given object and the key of the given field are passed in. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputKey	String	The key of the given field.
InputURI	String	The uri of the given object.

Return type

Boolean

Back to top

Microflow 'CanReadInstance'

This microflow checks the Polarion permissions to see if the current user can read the given object. The URI of the given object is passed in. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputURI	String	The uri of the given object.

Return type

Boolean

Back to top

Microflow 'CanReadKey'

This microflow checks the Polarion permissions to see if the current user can read the field with given key of the object. The URI of the given object and the key of the given field are passed in. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputKey	String	The key of the given field.
InputURI	String	The uri of the given object.

Return type

Boolean

Back to top

Microflow 'CreateAttachments'

Creates Work Item attachments.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with attachment data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'CreateWorkItem'

Creates a new Work Item with the given content. The Project and the Type must be set or the Work Item will not be created. The URI MUST NOT be set, otherwise the creation will fail.

Parameters

Name	Type	Documentation
CreateWorkItem	PolarionConnector.CreateWorkItem	Data of the Work Item to be created.

Return type

String

Back to top

Microflow 'DeleteAttachments'

Removes Work Item attachments.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with attachment data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'DeleteWorkItem'

Deletes a Work Item from Polarion.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item to be deleted.

Return type

Nothing

Back to top

Microflow 'DeleteWorkRecords'

Deletes Work Records of a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with work record data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'EditApproval'

Changes the status of an approval.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with approval data associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'EndTransaction'

Ends the explicit transaction of the current session by either commit or rollback.

Parameters

Name	Type	Documentation
Rollback	Boolean	If true the transaction is rolled back otherwise it is commited.

Return type

Nothing

Back to top

Microflow 'ExecuteAdminLogin'

Shows admin home page if valid Polarion session exists

Parameters

This microflow has no parameters.

Return type

Boolean

Back to top

Microflow 'ExecuteLogin'

Logs a user into Polarion and creates a session.

Parameters

Name	Type	Documentation
Credentials	PolarionConnector.Credentials	User credential information.

Return type

Nothing

Back to top

Microflow 'ExecuteLogout'

Ends current Polarion session.

Parameters

This microflow has no parameters.

Return type

Nothing

Back to top

Microflow 'ExecuteUserLogin'

Shows user home page if valid Polarion session exists

Parameters

This microflow has no parameters.

Return type

Boolean

Back to top

Microflow 'ExportDocumentToPDF'

This action allows you to export and download document as a PDF file. It uses "exportDocumentToPDF" webservice call.
Note: This action is available from Polarion V20R1 onwards.

Parameters

Name	Type	Documentation
PLModule	PolarionConnector.PLModule	PLModule object with associated PLProject object. (PLProject should not be null) and "uri" attribute is not null.

Return type

Nothing

Back to top

Microflow 'GetAllEnumOptionsForId'

Gets all enumeration options.

Parameters

Name	Type	Documentation
GetAllEnumOptionsForIdInput	PolarionConnector.GetAllEnumOptionsForIdInput	Input object to retrieve enum options for. Project ID and EnumID should be provided.

Return type

List of PolarionConnector.EnumOption

Back to top

Microflow 'GetAllLinkedWorkItems'

Gets all linked work items (direct, back) of a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item information object

Return type

List of PolarionConnector.PLLinkedWorkItem

Back to top

Microflow 'GetAttachmentFile'

Downloads a file attached to Work Item.

Parameters

Name	Type	Documentation
PLAttachment	PolarionConnector.PLAttachment	Attachment of the Work Item
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object

Return type

Nothing

Back to top

Microflow 'GetAvailableActions'

Gets all the available actions that can be used on the workflow object.

Parameters

Name	Type	Documentation
WorkFlowObjectURI	String	URI of any WorkflowObject, e.g. WorkItem or Module.

Return type

List of PolarionConnector.WorkflowAction

Back to top

Microflow 'GetBackLinkedWorkItems'

Gets the back linked Work Items of a Work Item.

Parameters

Name	Type	Documentation
WorkItemUri	String	Work Item URI

Return type

List of PolarionConnector.PLLinkedWorkItem

Back to top

Microflow 'GetCachedCustomFieldTypes'

Gets all custom field definitions for a Work Item Type. First checks if the Work Item custom field definitions exists in session cache. If found returns the cached data otherwise fetches data from Polarion and caches it.

Parameters

Name	Type	Documentation
ProjectId	String	ID of the Project
WorkItemTypeId	String	Work Item Type ID
WorkItemURI	String	The URI of the Work Item

Return type

List of PolarionConnector.CustomFieldType

Back to top

Microflow 'GetCachedEnumOptionsForId'

Gets all enumeration options. First checks if the enumeration options exists in session cache. If found returns the cached data otherwise fetches data from Polarion and caches it.

Parameters

Name	Type	Documentation
EnumID	String	ID of the enumeration to get the options for
ObjectTypeID	String	ID of the Work Item Type
ProjectID	String	ID of the Project

Return type

List of PolarionConnector.EnumOption

Back to top

Microflow 'GetCategories'

Get the categories defined for the give project id.

Parameters

Name	Type	Documentation
ProjectID	String	ID of the Project

Return type

List of PolarionConnector.PLCategory

Back to top

Microflow 'GetChildFolders'

This action uses 'getChildFoldes' webservice calls to get list of folders contained in a parent folder and returns list of PLModules with location fields.

Parameters

Name	Type	Documentation
PLModule	PolarionConnector.PLModule	PLModule object with associated PLProject object. (PLProject should not be null) and location field set.

Return type

List of PolarionConnector.PLModule

Back to top

Microflow 'GetConstructedWorkItemURL'

This microflow returns a constructed URL link to access the input Work Item in Polarion. It makes use of two constants TOKEN_FOR_PROJECT_ID_IN_URL_PATH and TOKEN_FOR_WORKITEM_ID_IN_URL_PATH to construct the URL path. The URL returned by this microflow is of the default type <Polarion Host Address>#/project/<ProjectId>/workitem?id=<WorkItemId>. Write your own microflow similar to this one if your installation does not follow the default install paths.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	The Work Item

Return type

String

Back to top

Microflow 'GetCustomField'

Gets custom field of a Work Item.

Parameters

Name	Type	Documentation
GetCustomFieldInput	PolarionConnector.GetCustomFieldInput	Input object with Work Item URI and custom field name to be retrieved.

Return type

PolarionConnector.PLCustom

Back to top

Microflow 'GetCustomFields'

Gets all the custom fields of a Work Item.

Parameters

Name	Type	Documentation
IgnoreCustomFieldsList	List of PolarionConnector.IgnoreCustomFields	List of custom fields to be ignored.
PLWorkItem	PolarionConnector.PLWorkItem	Work Item of custom fields to be retrieved.

Return type

Nothing

Back to top

Microflow 'GetCustomFieldTypes'

Gets all custom field definitions for a specific Work Item.

Parameters

Name	Type	Documentation
WorkItemURI	String	The URI of the Work Item

Return type

List of PolarionConnector.CustomFieldType

Back to top

Microflow 'GetDeepContainedProjects'

This action retrieves all available Projects for current user by using "getRootProjectGroup" and "getDeepContainedProjects" webservice calls.
Note: This action retrieves flat list of all available projects for logged in user.

Parameters

This microflow has no parameters.

Return type

List of PolarionConnector.PLProject

Back to top

Microflow 'GetDefinedCustomFieldKeys'

Gets all custom field keys defined for a Work Item Type in a Project.

Parameters

Name	Type	Documentation
GetDefinedCustomFieldKeysInput	PolarionConnector.GetDefinedCustomFieldKeysInput	Input object having information required to retrieve custom field keys. Project ID and Work Item Type ID should be provided.

Return type

List of PolarionConnector.PLString

Back to top

Microflow 'GetDefinedCustomFieldTypes'

Gets all custom field definitions for a Work Item Type.

Parameters

Name	Type	Documentation
GetDefinedCustomFieldKeysInput	PolarionConnector.GetDefinedCustomFieldKeysInput	Input object having information required to retrieve custom field definitions. Project ID and Work Item Type ID should be provided.

Return type

List of PolarionConnector.CustomFieldType

Back to top

Microflow 'GetDocumentSpaces'

This action invokes "getDocumentSpaces" webservice to fetch list of spaces for a given project.

Parameters

Name	Type	Documentation
PLModule	PolarionConnector.PLModule	PLModule object with associated PLProject object. (PLProject should not be null).

Return type

List of PolarionConnector.PLModule

Back to top

Microflow 'GetLinkedWorkItems'

Gets direct Linked Work Items of a Work Item.

Parameters

Name	Type	Documentation
ProjectID	String	Project ID
WorkItemID	String	Work Item ID

Return type

List of PolarionConnector.PLLinkedWorkItem

Back to top

Microflow 'GetModules'

This action uses 'getModules' webservice call to return list of available Modules/Documents relative to Module location for a given project.
a. If this location field is available with PLModule object, action will return list of Modules relative to available location.
b. If location field at PLModule object is empty, action "GetDocumentSpaces" will be run to get all available locations and getModules call on those locations.

Parameters

Name	Type	Documentation
PLModule	PolarionConnector.PLModule	PLModule object with associated PLProject object. (PLProject should not be null) and location field (optional)

Return type

List of PolarionConnector.PLModule

Back to top

Microflow 'GetPolarionConfiguration'

Retrieves active Polarion configuration from Mendix database. Returns empty configuration object and throws a warning message if there is no active configuration found.

Parameters

This microflow has no parameters.

Return type

PolarionConnector.PolarionConfiguration

Back to top

Microflow 'GetPolarionSession'

Retrieves the existing Polarion Session from Mendix database. Creates a new Polarion Session if none is found.

Parameters

This microflow has no parameters.

Return type

PolarionConnector.PolarionSession

Back to top

Microflow 'GetProjectUsers'

Gets users of a project.

Parameters

Name	Type	Documentation
ProjectID	String	ID of the project to get the users for.

Return type

List of PolarionConnector.PLUser

Back to top

Microflow 'GetRootFolders'

This action calls 'getRootFolders' and returns a list of parent folders (top-level) for a project.(Sorted in ascending order by name)

Parameters

Name	Type	Documentation
PLModule	PolarionConnector.PLModule	PLModule object with associated PLProject object. (PLProject should not be null).

Return type

List of PolarionConnector.PLModule

Back to top

Microflow 'GetServiceRequestHeader'

Creates a webservice request header string with Polarion login cookie which can be used as value to Request Header of webservice activity

Parameters

Name	Type	Documentation
PolarionSession	PolarionConnector.PolarionSession	Polarion session

Return type

String

Back to top

Microflow 'GetUser'

Get the User information by User ID.

Parameters

Name	Type	Documentation
UserID	String	User ID

Return type

PolarionConnector.PLUser

Back to top

Microflow 'GetUsers'

Gets all users.

Parameters

This microflow has no parameters.

Return type

List of PolarionConnector.PLUser

Back to top

Microflow 'GetWorkItemById'

Gets Work Item by ID and its field information as per the input mapping fields.

Parameters

Name	Type	Documentation
GetWorkItemByIdInput	PolarionConnector.GetWorkItemByIdInput	Input object with data required to fetch Work Item

Return type

PolarionConnector.PLWorkItem

Back to top

Microflow 'GetWorkItemByIDWithFields'

Gets a Work Item by ID with only the fields specified filled in with values from Polarion.

Parameters

Name	Type	Documentation
ProjectID	String	Project ID
WorkItemFieldList	List of PolarionConnector.PLString	List of Work Item fields to be fetched from Polarion.
WorkItemID	String	Work Item ID

Return type

PolarionConnector.PLWorkItem

Back to top

Microflow 'GetWorkItemByURI'

Gets Work Item by URI and its field information as per the input mapping fields.

Parameters

Name	Type	Documentation
WorkItemURI	String	Work Item URI

Return type

PolarionConnector.PLWorkItem

Back to top

Microflow 'GetWorkItemsCount'

Counts the number of Work Items returned by the given query.

Parameters

Name	Type	Documentation
Query	String	The Lucene query to be used.

Return type

Integer/Long

Back to top

Microflow 'HasCurrentUserPermission'

This microflow checks the Polarion permissions to see if the current user is granted the given permission for the given project. For example, to check modify permissions for Work Items of a given project for a given user, pass in the string 'com.polarion.persistence.object.WorkItem.modify' as an input parameter. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputPermission	String	The permission to check. For example, to check modify permission set it to 'com.polarion.persistence.object.WorkItem.modify'.
InputProjectId	String	The id of the project to check the permission in, null to check global permissions.

Return type

Boolean

Back to top

Microflow 'HasPermission'

This microflow checks the Polarion permissions to see if the given user is granted the given permission for the given project. For example, to check modify permissions for Work Items of a given project for a given user, pass in the string 'com.polarion.persistence.object.WorkItem.modify' as an input parameter. It returns true if the permission is granted, false otherwise.

Parameters

Name	Type	Documentation
InputPermission	String	The permission to check. For example, to check modify permission set it to 'com.polarion.persistence.object.WorkItem.modify'.
InputProjectId	String	The id of the project to check the permission in, null to check global permissions.
InputUserId	String	The id of the user to check the permission for

Return type

Boolean

Back to top

Microflow 'LoginWithToken'

Logs in user to Polarion using token.

Parameters

Name	Type	Documentation
Mechanism	String	The mechanism which client request for authentication (server must be configured for given mechanism otherwise login will be rejected)
Token	String	The token of the user to log-in
UserName	String	The name of the user to log-in

Return type

Nothing

Back to top

Microflow 'PerformWorkflowAction'

Executes a workflow action. GetAvailableActions can be used to get the actions that can be performed.

Parameters

Name	Type	Documentation
ActionID	Integer/Long	The Id of the action to execute.Use GetAvailableActions microflow to get action information
WorkflowObjectURI	String	URI of any WorkflowObject, e.g. WorkItem or Module.

Return type

Nothing

Back to top

Microflow 'PerformWorkItemWorkflowAction'

Executes a workflow action.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with workflow action data associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'QueryModules'

QueryModule microflow uses 'queryModules' webservice to performs a Lucene search on Live Documents based on query input with fields and returns a list of searched Document.

Parameters

Name	Type	Documentation
PLModule	PolarionConnector.PLModule	PLModule object with associated PLProject object. (PLProject should not be null)
QueryModuleFieldsList	List of PolarionConnector.QueryModuleFields	List of QueryModuleFields objects containing an array of Module Fields to retrieve from query.
QueryModules	PolarionConnector.QueryModules	Query object to pass. This object will contain query, sort and resultsLimit attributes.

Return type

List of PolarionConnector.PLModule

Back to top

Microflow 'QueryWorkItems'

Action QueryWorkitems invokes "queryWorkitems" webservice that performs a Luncene query on workitems to retrive a list of searched workitems.

Parameters

Name	Type	Documentation
QueryWorkItems	PolarionConnector.QueryWorkItems	Query object to pass. This object has to contain lucene query.
WorkItemFieldsList	List of PolarionConnector.PLString	List of PLString object containing arry of fields to be retrieved from search.

Return type

List of PolarionConnector.PLWorkItem

Back to top

Microflow 'QueryWorkItemsLimited'

Queries for Work Items.

Parameters

Name	Type	Documentation
Fields	List of PolarionConnector.PLString	The keys of the fields that should be filled in.
Query	String	The Lucene query to be used.
ResultsLimit	Integer/Long	The maximum number of returned Work Items. (-1 for unlimited.)
Sort	String	The field to be used for sorting.

Return type

List of PolarionConnector.PLWorkItem

Back to top

Microflow 'RemoveApprovee'

Removes an approving user.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with approval data associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'RemoveAssignee'

Removes Work Item assignee(s).

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with assignee data associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'RemoveCategory'

Removes Work Item categories.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with category data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'RemoveHyperlink'

Removes Work Item hyperlinks.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with hyperlink data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'RemoveLinkedRevisions'

Removes linked revisions associated to a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with linked revision data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'RemoveLinkedWorkItem'

Removes linked work items associated to a Work Item.

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with linked work item data associated to it.Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'SavePolarionConfiguration'

Saves Polarion configuration to the database

Parameters

Name	Type	Documentation
PolarionConfiguration	PolarionConnector.PolarionConfiguration	Polarion configuration to be saved

Return type

Nothing

Back to top

Microflow 'SetConfigurationAsActive'

Sets the input Polarion configuration record as Active and all other configuration records as InActive.

Parameters

Name	Type	Documentation
PolarionConfiguration	PolarionConnector.PolarionConfiguration	Polarion Configuration to set active

Return type

Nothing

Back to top

Microflow 'SetCustomField'

Sets a custom field of a Work Item.

Parameters

Name	Type	Documentation
SetCustomFieldInput	PolarionConnector.SetCustomFieldInput	Custom field information to be set to Work Item.

Return type

Nothing

Back to top

Microflow 'SetCustomFields'

Sets the custom fields to a Work Item,

Parameters

Name	Type	Documentation
PLWorkItem	PolarionConnector.PLWorkItem	Work Item object with custom fields associated to it. Work Item URI should be set otherwise it will fail.

Return type

Nothing

Back to top

Microflow 'ShowLoginPage'

Displays the login page after it is determined that there is no valid pre-existing session. It also returns a boolean to the calling microflow to take action accordingly (for example, show Home page).

Parameters

This microflow has no parameters.

Return type

Boolean

Back to top

Microflow 'TransactionExists'

Checks if there is a transaction exist for the current session.

Parameters

This microflow has no parameters.

Return type

Boolean

Back to top

Microflow 'UpdateWorkItem'

Updates a Work Item. The Work Item passed has to contain the valid URI of the Work Item to be updated.

Parameters

Name	Type	Documentation
UpdateWorkItem	PolarionConnector.UpdateWorkItem	Work Item data to be updated.

Return type

Nothing

Back to top

Microflow 'UserLogin'

This microflow shows the login page (via ShowLoginPage microflow) and then the default Home page for user roles.

Parameters

This microflow has no parameters.

Return type

Nothing

Back to top

Siemens Industry Software

Headquarters

Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe

Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific

Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens Digital Industries Software

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing, and electronics design meet tomorrow. Our solutions help companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities, and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit siemens.com/software or follow us on LinkedIn, Twitter, Facebook, and Instagram.

Siemens Digital Industries Software —
Where today meets tomorrow.