Pega Process Fabric Hub

17 February 2023

CONTENTS

Registering Pega applications in Pega Process Fabric Hub

For seamless management of work that include assignments from multiple applications, register your applications in Pega Process Fabric Hub. By using a unified worklist in Pega Process Fabric Hub, you save time and provide a consistent user experience.

Before you begin: Prepare to register your applications:

- Ensure that the application that you want to register is present on a Pega Platform system that supports HTTPS and has valid certificates.
- Ensure that the operator IDs in the system that hosts your applications and in the system that hosts Pega Process Fabric Hub are identical. The operator ID is case-sensitive. Other operator details can differ between the systems.
- Ensure that you have an *Application*: `Administrators`, *Application*: `Authors`, or your custom access group with administrative privileges in the system that hosts your applications.

> **Note:** If you use a custom access group other than *Application*: `Administrators` or *Application*: `Authors`, add the access group to the *AllowedRegistrationAccessGroups* dynamic system setting. For more information, see Modification of the standard mappings published to Pega Process Fabric Hub and Editing a dynamic system setting . Ensure that the access group is present in your operator record. For more information, see Defining user contact information and application access .

In Pega Process Fabric Hub, you can use Interwoven Worklist that consolidates assignments from your numerous registered applications. By using Interwoven Worklist, users save time and complete daily assignments without switching context between many applications.

Register your applications in Pega Process Fabric Hub by completing the following tasks:

- **Installing the Pega Process Fabric Hub connector component**
  To connect your applications to Pega Process Fabric Hub, download and install the Pega Process Fabric Hub connector component in your remote application that you want to register with Pega Process Fabric Hub.
- **Enabling Constellation applications for registration**
  Merge seamless work management and an innovative UI framework by enabling Constellation applications for registration with Pega Process Fabric Hub. By default, Constellation applications are not compatible with Pega Process Fabric Hub, but after you provide a correct data transform configuration, your applications can publish data to Pega Process Fabric Hub for unified case and assignment management in your organization.
- **Establishing communication between Pega Process Fabric Hub and Pega Platform by editing dynamic system settings**
  For a seamless user experience, ensure that the Pega Platform system that hosts your applications communicates correctly with Pega Process Fabric Hub. Add the Pega Process Fabric Hub URL to a dynamic system setting in your system so that your applications within the system can send assignments to Pega Process Fabric Hub.
- **Updating a keystore for Pega Platform to connect to Pega Process Fabric Hub**
  To create a safe and reliable connection between your applications and Pega Process Fabric Hub, update the keystore in your Pega Platform system that points to Pega Process Fabric Hub.
- **Providing access rights for Pega Process Fabric Hub by editing a service package**

To ensure that Pega Process Fabric Hub can establish a connection with the applications in your Pega Platform system, configure appropriate access rights for Pega Process Fabric Hub by editing a service package.

- **Configuring a token profile for a remote application**
  If you host your remote application on the same system on which you also host your instance of Pega Process Fabric Hub, configure a truststore in a token profile of your remote application to establish a secure and reliable connection between your application and Pega Process Fabric Hub.
- **Configuring synchronization with Pega Process Fabric Hub**
  If you host your applications in Pega Platform version 8.1 or 8.2, ensure that queue processors and the AsyncProcessor requestor run correctly so that Pega Process Fabric Hub can successfully synchronize data with your application.
- **Creating a secure connection with Pega Process Fabric Hub by configuring OAuth 2.0 authentication**
  Provide a secure connection between the applications that your Pega Platform system hosts and Pega Process Fabric Hub by creating an OAuth 2.0 client registration. As a result, you receive a secure ID and password so that you can ensure that the communication between your applications and Pega Process Fabric Hub is safe.
- **Activating applications in Pega Process Fabric Hub**
  Start processing assignments from multiple applications within one window by activating your applications in Pega Process Fabric Hub. After you register and activate your application, you can process Interwoven Worklist that includes tasks from different applications, without logging in and out between the applications.
- **Modification of the default access groups with administrative privileges**
  To register an application in Pega Process Fabric Hub, your access group needs to have administrative privileges. The default administrative access groups are *Application*`:Administrators` and *Application*`:Authors`. To meet your unique business requirements, you can create your custom access group with administrative privileges.
- **Filtering assignments for publication**

Publish only specific assignments from your remote application with the Pega Process Fabric Hub Connector component to Pega Process Fabric Hub. For example, you can prevent assignments that are related to a specific case type from publication.

- **Modification of the standard mappings published to Pega Process Fabric Hub**
  Extensions pass fields from your application to Pega Process Fabric Hub so that Pega Process Fabric Hub reflects data from your application. Overriding default extensions maps the values between your application and Pega Process Fabric Hub in a way that is accurate for your unique business scenario.
- **Filtering case types for publication**
  Specify which case types you want to allow or block when you publish case type data from your remote application with the connector component to Pega Process Fabric Hub.
- **Filtering cases for publication**
  Publish only those cases that meet a specific condition from your remote application to Pega Process Fabric Hub.

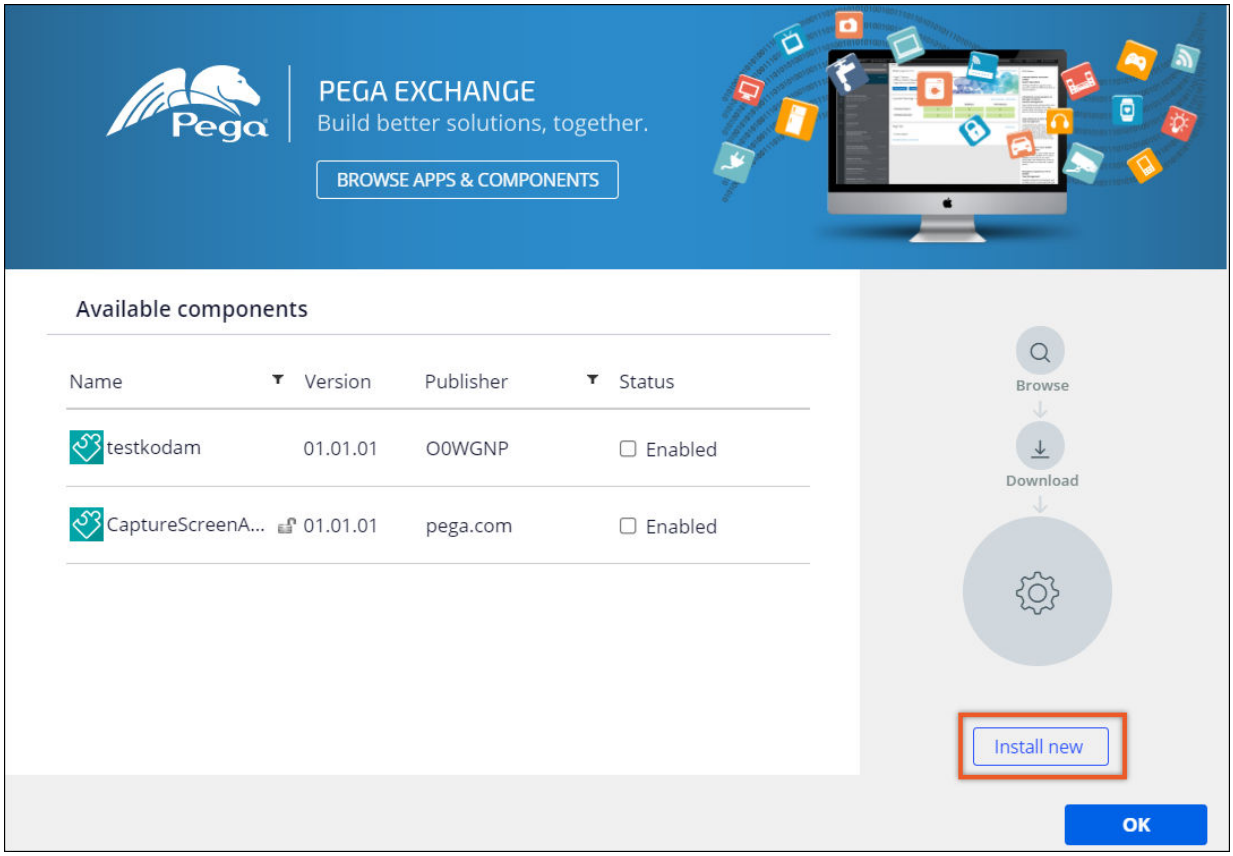
Installing the Pega Process Fabric Hub connector component

To connect your applications to Pega Process Fabric Hub, download and install the Pega Process Fabric Hub connector component in your remote application that you want to register with Pega Process Fabric Hub.The component is responsible for tracking the creation of new assignments and publishing assignment details to Pega Process Fabric Hub. The component also provides all the necessary transformation capabilities before publishing assignments.

> **Before you begin:** Download the Pega Process Fabric component from Pega Marketplace. Ensure that the component that you download matches the version of your instance of Pega Platform. For more information, see Pega Process Fabric Hub connector for Pega Platform.

> **Note:** You only install the Pega Process Fabric component once on your Pega Platform system. The installation impacts all the applications in your Pega Platform system.

1. In the header of Dev Studio, click the name of the application, and then click Definition.
2. On the Definition tab, in the **Enabled components** section, click Manage components.
3. In the dialog box, click Install new, as in the following figure:



Installing a component

4. In the dialog box, navigate to the `.zip` file that contains the Pega Process Fabric Hub connector component, and then click **Open**.

   Result: The component installation is in progress. After the installation is complete, the component status in the **Available components** is `Enabled`.

5. Click **OK** to close the dialog box.
6. On the application rule form, click **Save**.

For example:

What to do next:

Proceed with the installation process:

- If your remote application uses Constellation on Pega Platform version 8.6, prepare the application for registration. For more information, see Enabling Constellation applications for registration.
- If you build your remote application by using UI-Kit or Theme Cosmos, or you host your application on Pega Platform version 8.5 or earlier, create a connection between your remote application and Pega Process Fabric Hub. For more information, see Establishing communication between Pega Process Fabric Hub and Pega Platform by editing dynamic system settings.

Enabling Constellation applications for registration

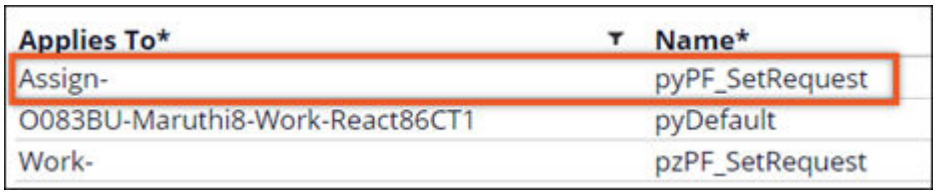
Merge seamless work management and an innovative UI framework by enabling Constellation applications for registration with Pega Process Fabric Hub. By default, Constellation applications are not compatible with Pega Process Fabric Hub, but after you provide a correct data transform configuration, your applications can publish data to Pega Process Fabric Hub for unified case and assignment management in your organization.

Note: Enabling Constellation applications applies only to applications that you build in Pega Platform™ version 8.6 by using Constellation. If you build your application in the UI-Kit or Theme Cosmos, or an earlier version of Pega Platform, skip this procedure and continue the registration process. For more information, see Establishing communication between Pega Process Fabric Hub and Pega Platform by editing dynamic system settings.

For more information, see Constellation architecture and Traditional UI architecture .

After you install the Pega Process Fabric Hub connector component, your application includes an empty *Assign-.pyPF_SetRequest* data transform and an empty *Work-.pyPF_SetCaseRequest* data transform. When you provide correct configuration for the data transforms, you can register your Constellation application with Pega Process Fabric Hub.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Data Model** category, and then click **Data Transform**.
3. In the list of data transform instances, open the **Assign-.pyPF_SetRequest** data transform, as in the following figure:



Opening the Assign-.pyPF_SetRequest data transform

4. Click **Save as**, and then save the rule in your application ruleset.
5. On the **Definition** tab, define the data transform by entering the following values:

Row	Action	Target	Relation	Source
1	When	`RequestPage.status != "Resolved"`	Not applicable	Not applicable
1.1	Remove	`RequestPage.actions`	Not applicable	Not applicable
1.2	Update Page	`RequestPage.actions(<APPEND>)`	Not applicable	Not applicable
1.2.1	Set	`.ID`	`equal to`	`"redirect"`
1.2.2	When	`!Application.pyIsConstellationApp`	Not applicable	Not applicable
1.2.2.1	Comment	`"?"+@(Pega-RULES:Utilities).pzEncryptURLActionString(tools,"Global","pyActivity=pyMobileSnapStart&Action=openAssignment"`	Not applicable	Not applicable

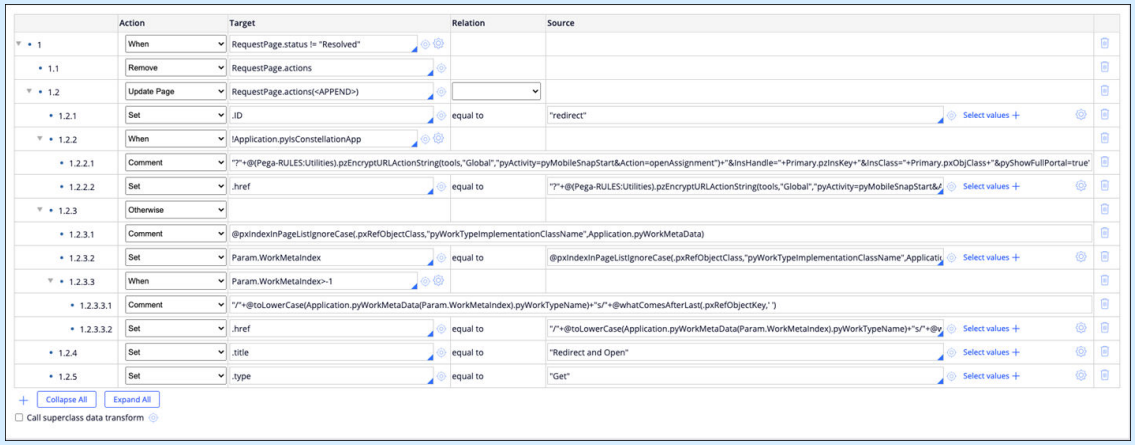
Row	Action	Target	Relation	Source
		) +"&InsHandle="+Primary.pzInsKey+"&InsClass="+Primary.pxObjClass+"&pyShowFullPortal=true"		
1.2.2.2	Set	.href	equal to	"?"+@(Pega-RULES:Utilities).pzEncryptURLActionString(tools,"Global","pyActivity=pyMobileSnapStart&Action=openAssignment") +"&InsHandle="+Primary.pzInsKey+"&InsClass="+Primary.pxObjClass+"&pySh

Row	Action	Target	Relation	Source
				`owFullPortal=true"`
1.2.3	Otherwise	not applicable	Not applicable	Not applicable
1.2.3.1	Comment	`@pxIndexInPageListIgnoreCase(.pxRefObjectClass,"pyWorkTypeImplementationClassName",Application.pyWorkMetaData)`	Not applicable	Not applicable
1.2.3.2	Set	`Param.WorkMetaIndex`	`equal to`	`@pxIndexInPageListIgnoreCase(.pxRefObjectClass,"pyWorkTypeImplementationClassName",Application.pyWorkMetaData)`
1.2.3.3	When	`Param.WorkMetaIndex>-1`	Not applicable	Not applicable

Row	Action	Target	Relation	Source
1.2.3.3.1	Comment	"/"+@toLowerCase(Application.pyWorkMetaData(Param.WorkMetaIndex).pyWorkTypeName)+" <s/ or es/ or ies/ >"+@whatComesAfterLast(.pxRefObjectKey,' ')	Not applicable	Not applicable
1.2.3.3.2	Set	.href	equal to	"/"+@toLowerCase(Application.pyWorkMetaData(Param.WorkMetaIndex).pyWorkTypeName)+"s/"+@whatComesAfterLast(.pxRefObjectKey,' ')

Row	Action	Target	Relation	Source
1.2.4	Set	`.title`	`equal to`	`"Redirect and Open"`
1.2.5	Set	`.type`	`equal to`	`"Get"`

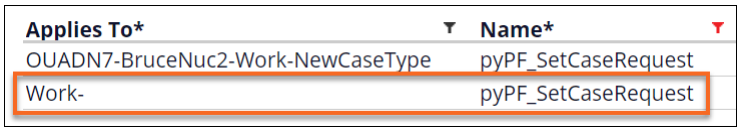
For example: The following figure shows a correctly completed data transform:



Assign-.pyPF_SetRequest data transform

6. On the **Pages & Classes** tab, define data pages that the data transform accesses by entering the following values:
    a. In the **Page name** field, enter `RequestPage`, and then, in the **Class** field, enter `$NONE`.
    b. Click **Add item**.
    c. In the **Page name** field, enter `RequestPage.actions()`, and then, in the **Class** field, enter `$NONE`.
7. Click **Save**.
8. In the navigation pane of Dev Studio, click **Records**.
9. Expand the **Data Model** category, and then click **Data Transform**.

10. In the list of data transform instances, open the pyPF_SetCaseRequest data transform, as in the following figure:



Applies To*	Name*
OUADN7-BruceNuc2-Work-NewCaseType	pyPF_SetCaseRequest
Work-	pyPF_SetCaseRequest

Opening the Work-.pyPF_SetCaseRequest data transform

11. Click Save as, and then save the rule in your application ruleset and corresponding case type class.
12. On the Definition tab, define the data transform by entering the following values:

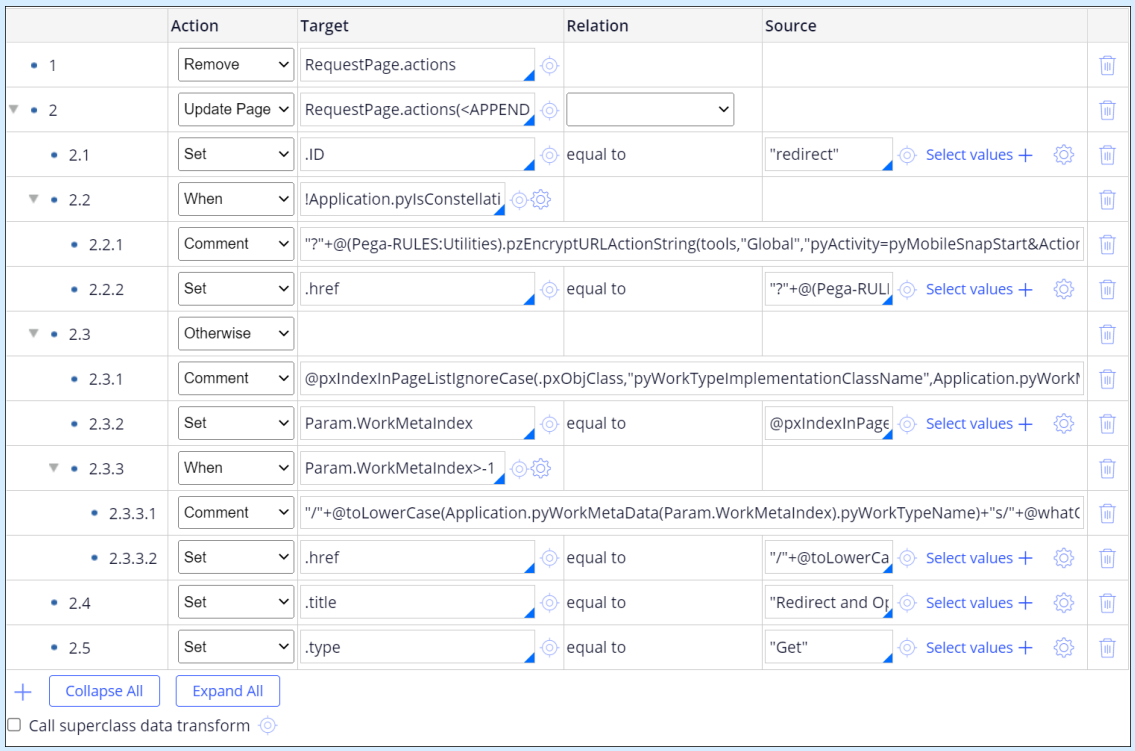
Row	Action	Target	Relation	Source
1	Remove	`RequestPage.actions`	Not applicable	Not applicable
2	Update Page	`RequestPage.actions(<APPEND>)`	Not applicable	Not applicable
2.1	Set	`.ID`	`equal to`	`"redirect`
2.2	When	`!Application.pyIsConstellationApp`	Not applicable	Not applicable
2.2.1	Comment	`"?"+@(Pega-RULES:Utilities).pzEncryptURLActionString(tools,"`	Not applicable	Not applicable

Row	Action	Target	Relation	Source
		Global","pyActivity=pyMobileSnapStart&Action=openWorkByHandle&InsHandle="+Primary.pzInsKey+"&pyShowFullPortal=true&ThreadName="+Primary.pyID)		
2.2.2	Set	.href	equal to	"?"+@(Pega-RULES:Utilities).pzEncryptURLActionString(tools,"Global","pyActivity=pyMobileSnapStart&Action=openWorkByHandle&InsHandle="+Primary.pzInsKey+"&pyShowF

Row	Action	Target	Relation	Source
				ullPortal=true&ThreadName="+Primary.pyID)
2.3	Otherwise	Not applicable	Not applicable	Not applicable
2.3.1	Comment	@pxIndexInPageListIgnoreCase(.pxObjClass,"pyWorkTypeImplementationClassName",Application.pyWorkMetaData)	Not applicable	Not applicable
2.3.2	Set	Param.WorkMetaIndex	equal to	@pxIndexInPageListIgnoreCase(.pxObjClass,"pyWorkTypeImplementationClassName",Application.pyWorkMetaData)

Row	Action	Target	Relation	Source
2.3.3	When	`Param.WorkMetaIndex>-1`	Not applicable	Not applicable
2.3.3.1	Comment	`"/"+@toLowerCase(Application.pyWorkMetaData(Param.WorkMetaIndex).pyWorkTypeName)+"s/"+@whatComesAfterLast(.pzInsKey," ")`	Not applicable	Not applicable
2.3.3.2	Set	`.href`	`equal to`	`"/"+@toLowerCase(Application.pyWorkMetaData(Param.WorkMetaIndex).pyWorkTypeName)+"s/"+@whatComesAfterLast(.pzInsKey,' ')`
2.4	Set	`.title`	`equal to`	`"Redirect and Open"`
2.5	Set	`.type`	`equal to`	`"Get"`

For example: The following figure shows a correctly completed data transform:



Work-.pyPF_SetCaseRequest data transform

13. On the **Pages & Classes** tab, define the data pages that the data transform accesses:
    a. In the **Page name** field, enter `RequestPage`, and then, in the **Class** field, enter `$NONE`.
    b. Click **Add item**.
    c. In the **Page name** field, enter **RequestPage.actions()**, and then, in the **Class** field, enter `$NONE`.
    d. Click **Save**.

What to do next: Continue with registering your application. For more information, see Establishing communication between Pega Process Fabric Hub and Pega Platform by editing dynamic system settings.

Establishing communication between Pega Process Fabric Hub and Pega Platform by editing dynamic system settings

For a seamless user experience, ensure that the Pega Platform system that hosts your applications communicates correctly with Pega Process Fabric Hub. Add the Pega Process Fabric Hub URL to a dynamic system setting in your system so that your applications within the system can send assignments to Pega Process Fabric Hub.

Before you begin: Prepare your Pega Platform system to create the connection with Pega Process Fabric Hub:

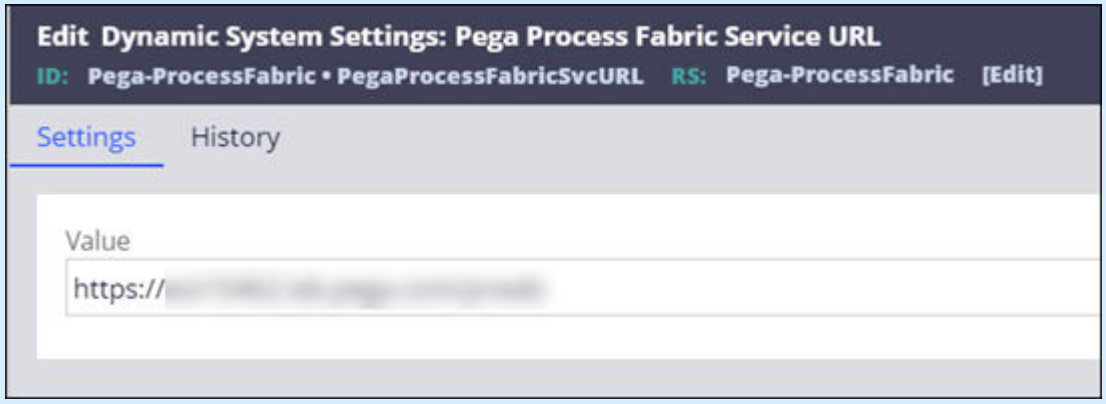
- Obtain the Pega Process Fabric Hub URL by contacting your administrator.
- Add the Pega Process Fabric connector component to your remote application. For more information, see Installing the Pega Process Fabric Hub connector component.

Note: Because the dynamic system settings impact your entire Pega Platform environment, you only need to edit the settings once. The configuration affects all the applications in your Pega Platform system.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **SysAdmin** category, and then click **Dynamic System Settings**.

3. In the list of the dynamic system settings instances, select PegaProcessFabricSvcURL.
4. On the Settings tab, in the Value field, enter the Pega Process Fabric Hub URL.

For example: Enter `https://`*sample URL*, as in the following figure:



Dynamic system settings configuration for the Pega Process Fabric Hub

5. Click Save.

For example:

Updating a keystore for Pega Platform to connect to Pega Process Fabric Hub

To create a safe and reliable connection between your applications and Pega Process Fabric Hub, update the keystore in your Pega Platform system that points to Pega Process Fabric Hub.

A keystore is a file that contains keys and certificates that you use for encryption, authentication, and serving content over HTTPS. To connect to Pega Process Fabric

Hub, you update a keystore data instance that points to the URL of your Pega Process Fabric Hub instance.

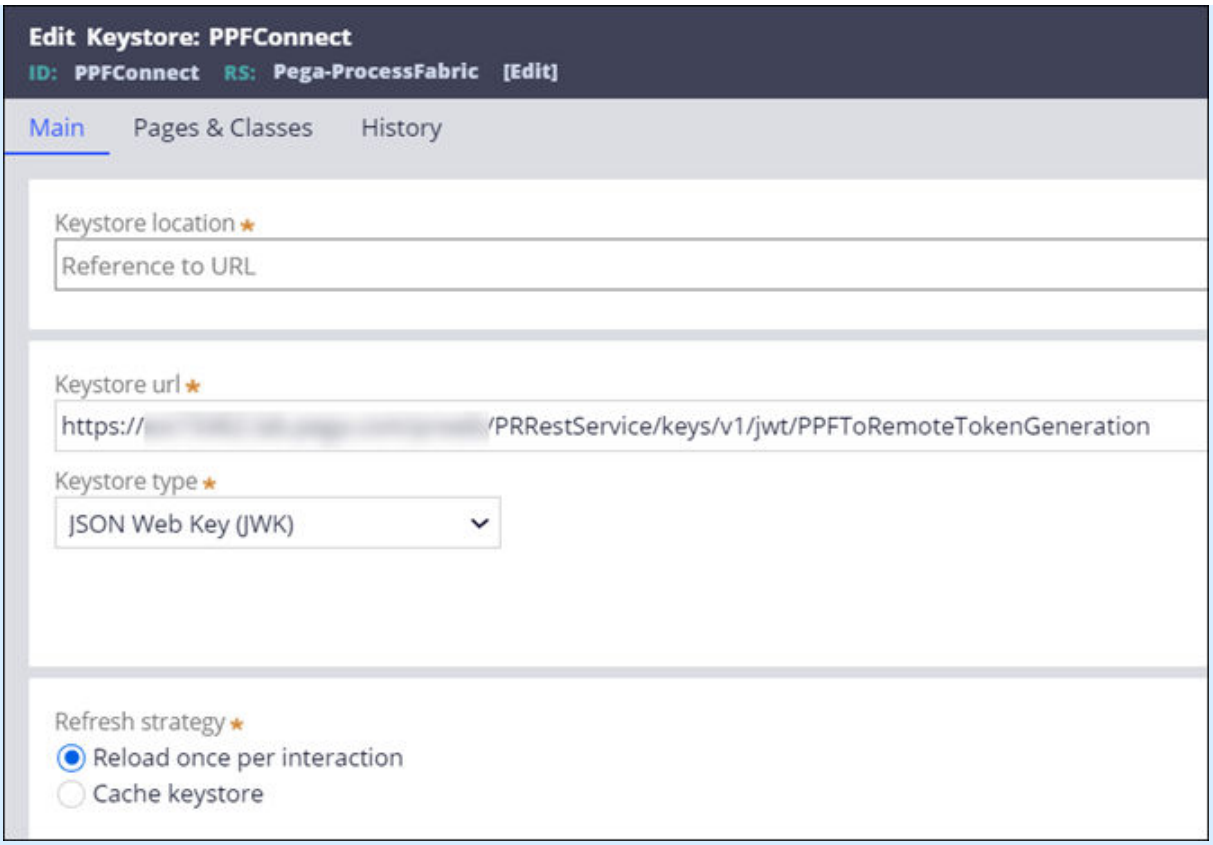
> **Note:** Adding the Pega Process Fabric Hub URL to your keystore is a system-wide action, so you only edit the keystore once for your Pega Platform system. The configuration impacts all of the applications in your Pega Platform system.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **Keystore**.
3. In the list of keystore instances, select **PPFConnect**.
4. On the **Main** tab, in the **Keystore location** field, press the Down arrow key, and then select **Reference to URL**.
5. In the **Keystore URL** field, enter the URL of the site that hosts Pega Process Fabric Hub.
   You only change the part of the URL that indicates the hosting server, as in the following example.

   > **For example:** Enter `https://`*hosting server URL*`/PRRestService/keys/v1/jwt/PPFToRemoteTokenGeneration`.

6. In the **Keystore type** list, select **JSON Web Key (JWK)**.
7. In the **Refresh strategy** list, select **Reload once per interaction**.
8. Click **Save**.

> **Result:** The following figure shows a complete keystore configuration:



Keystore configurations that support communication with the Pega Process Fabric Hub

Providing access rights for Pega Process Fabric Hub by editing a service package

To ensure that Pega Process Fabric Hub can establish a connection with the applications in your Pega Platform system, configure appropriate access rights for Pega Process Fabric Hub by editing a service package.

A service package is a collection of services that controls access to the listeners for services with defined access groups, such as authentication and REST services. By updating a service package, you ensure that Pega Process Fabric Hub can connect to your registered applications.

Note: Service package settings are system-wide and affect all applications in your instance of Pega Platform. You only need to configure these settings once and the configuration affects all applications. The configuration resets only after you reinstall the Pega Process Fabric Hub connector component from your remote application.

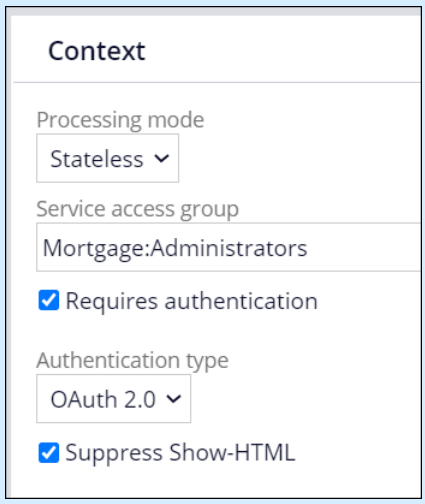
1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Integration-Resources** category, and then click **Service Package**.
3. In the list of the service package instances, select **ppfconnect**.
4. On the **Context** tab, in the **Processing mode** list, select **Stateless**.
5. In the **Service access group** field, enter the name of an access group in the application that you want to connect to the Pega Process Fabric Hub.

   For example: Enter *YourApplication*`:Administrators`.

   Note: If you use a custom access group other than *Application*`:Administrators` or *Application*`:Authors`, add the access group to the *AllowedRegistrationAccessGroups* dynamic system setting. For more information, see Modification of the standard mappings published to Pega Process Fabric Hub and Editing a dynamic system setting . Ensure that the access group is present in your operator record. For more information, see Defining user contact information and application access .

6. Select the **Requires authentication** checkbox.
7. In the **Authentication type** list, select **OAuth 2.0**.
8. Select the **Suppress Show-HTML** checkbox.

For example: The following figure shows a sample configuration for the service package:



Service package configurations

9. In the **Methods** section, view the newly created REST services by clicking **Refresh**.

Result: The *ppfconnect* service package now includes the *app*, *assignments/{assignmentID}/canperform*, and *applications/{applicationName}/actions/{action}* REST services that you use to communicate with Pega Process Fabric Hub, as in the following figure:

REST methods to communicate with Pega Process Fabric Hub

10. Click Save.

What to do next:

- If you host your application in Pega Platform version 8.1 or 8.2, create synchronization between your application and Pega Process Fabric Hub. For more information, see Configuring synchronization with Pega Process Fabric Hub.
- If you host your application in Pega Platform version 8.3 or later, list your application in the System Runtime Context. For more information, see Including an application in System Runtime Context .

Configuring a token profile for a remote application

If you host your remote application on the same system on which you also host your instance of Pega Process Fabric Hub, configure a truststore in a token profile of your remote application to establish a secure and reliable connection between your application and Pega Process Fabric Hub.

Note: If you host your instance of Pega Process Fabric Hub and your remote application on different systems, or if you retain the default configuration of a truststore for your remote application, you can continue the application registration process without performing this procedure.

Before you begin:

- Ensure that you host both your remote application and Pega Process Fabric Hub on the same system.
- Configure a keystore for your remote application. For more information, see Updating a keystore for Pega Platform to connect to Pega Process Fabric Hub.

When you add the Pega Process Fabric Hub Connector component to your application, the system automatically adds a *PPFConnect* token profile to your application. By default, the *PPFConnect* token profile points to a *PPFConnect* truststore, which supports only scenarios in which you host Pega Process Fabric Hub and your remote applications on different systems. If you host your Pega Process Fabric Hub instance and your remote application on the same system, you need to provide a truststore that supports this configuration.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Security** category, and then click **Token Profile**.
3. In the list of token profile instances, click **PPFConnect**.
4. On the **Processing** tab, in the **Security** section, in the **Truststore** field, enter `PPFKS`.

Note: If you use a customized truststore, enter the name of your truststore instead of `PPFKS`.

5. Click Save.

What to do next:

- If you host your application in Pega Platform™ version 8.1 or 8.2, configure synchronization in your remote application. For more information, see Configuring synchronization with Pega Process Fabric Hub.
- If you host your application in Pega Platform version 8.3 or later, configure OAuth 2.0 authentication for your remote application. For more information, see Creating a secure connection with Pega Process Fabric Hub by configuring OAuth 2.0 authentication.

Configuring synchronization with Pega Process Fabric Hub

If you host your applications in Pega Platform version 8.1 or 8.2, ensure that queue processors and the AsyncProcessor requestor run correctly so that Pega Process Fabric Hub can successfully synchronize data with your application.

Note: Configuring queue processors and the AsyncProcessor requestor applies only to applications that you host in Pega Platform versions 8.1 and 8.2. If you host your application in Pega Platform version 8.3 or later, omit configuring queue processors and the AsyncProcessor requestor, and continue with configuring OAuth 2.0 authentication. For more information, see Creating a secure connection with Pega Process Fabric Hub by configuring OAuth 2.0 authentication.

Queue processors and the AsyncProcessor requestor synchronize assignments, operator to work queue mappings, and work queue metadata. After you install the

Pega Process Fabric Hub connector component, your remote application includes the following queue processors:

- *pyPF_PublishAssignment*
- *pyPF_PublishBulkAssignments*
- *pyPF_PublishBulkOperators*
- *pyPF_PublishBulkWorkQueues*

Note: Because queue processor and requestor settings affect your entire Pega Platform instance, the configurations that you apply in one application also affect other applications in your environment.

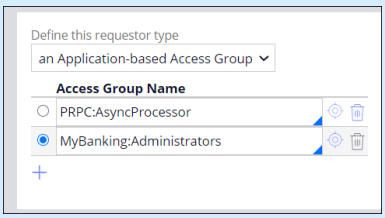
1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **SysAdmin** category, and then click **Requestor Type**.
3. In the list of requestor type instances, click the **ASYNCPROCESSOR** requestor type for the system that hosts your application.

For example: Select the Asyncprocessor requestor type for the Pega system:

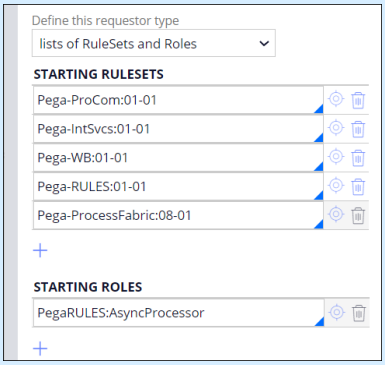
SYSTEM NAME	REQUESTOR TYPE	LABEL
pega	ASYNCPROCESSOR	ASYNCPROCESSOR Starting connection
pega	PORTAL	PORTAL Starting connection
pega	BROWSER	BROWSER Starting connection
pega	Batch	BATCH Starting connection
pega	APP	Application Starting connection
prpc	PORTAL	PORTAL Starting connection
prpc	BROWSER	BROWSER Starting connection
prpc	Batch	BATCH Starting connection
prpc	ASYNCPROCESSOR	ASYNCPROCESSOR Starting connection
prpc	APP	Application Starting connection

AsyncProcessor type in the list of requestor types

4. Define how the requestor type runs in the system:

Choices	Actions
Use an access group	a. In the Define this requestor type list, select an Application-based Access Group. b. Click Add item. c. In the Access Group Name field, enter an access group from your application that you want to register with Pega Process Fabric Hub. d. Make the application access group active by clicking the radio button next to the access group name. **Result:** The AsyncProcessor requestor type can communicate with the Pega Process Fabric Hub based on the access group, as in the following example: 

Choices	Actions
	AsyncProcessor requestor type configurations based on access group
Use rulesets and roles	a. In the Define this requestor type list, select list of RuleSets and Roles. **Note:** AsyncProcessor requestor type configuration that is based on rulesets and roles is deprecated. Use the rulesets and roles option only if your AsyncProcessor requestor type is already configured to support rulesets and roles. b. In the Starting rulesets column, create a list of rulesets by clicking Add row and entering the following rulesets in this order: `Pega-ProCom:01-01`, `Pega-IntSvcs:01-01`, `Pega-WB:01-01`, and `Pega-RULES:01-01`. c. Click Add row, and then add the Pega Process Fabric ruleset for a version that matches your version of Pega Platform.

Choices	Actions
	For example: For Pega Platform 8.1, enter `Pega-ProcessFabric:08-01`. d. In the Starting roles field, enter `PegaRULES:AsyncProcessor`. **Result:** The AsyncProcessor requestor type can communicate with Pega Process Fabric Hub based on the list of rulesets, as shown in the following example:  **AsyncProcessor requestor type configurations based on rulesets and roles**

5. Click Save.

Creating a secure connection with Pega Process Fabric Hub by configuring OAuth 2.0 authentication

Provide a secure connection between the applications that your Pega Platform system hosts and Pega Process Fabric Hub by creating an OAuth 2.0 client registration. As a result, you receive a secure ID and password so that you can ensure that the communication between your applications and Pega Process Fabric Hub is safe.

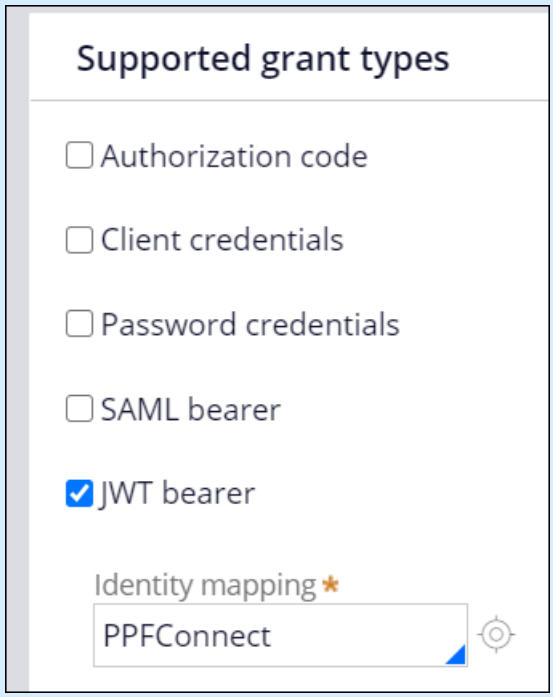
By using the OAuth 2.0 protocol, Pega Process Fabric Hub can communicate securely with your applications over HTTPS. You define OAuth 2.0 client registration data instances so that Pega Process Fabric Hub can access Pega Platform REST services. When you define the OAuth 2.0 client registration, you receive a client ID and a client secret that you use when you register your application in Pega Process Fabric Hub to create a secure connection. Client IDs and client secrets are unique to every operator in your system. After you create the client ID and client secret, you can use these credentials to register multiple applications that your operator can then access as an administrator.

1. Create a new OAuth 2.0 client registration:
    a. In the header of Dev Studio, click **Create** > **Security** > **OAuth 2.0 Client Registration**.
    b. In the **Short description** field, provide some descriptive information for your client registration.
    c. In the **Client Name** field, provide a client name for your OAuth 2.0 client registration.
    d. Click **Create and open**.
2. On the **Client information** tab, in the **Client Credentials** section, in the **Type of client** parameter, select **Confidential**.
3. In the **Supported grant types** section, select the **JWT bearer** check box.

Note: Ensure that you clear the remaining check boxes.

4. In the **Identity mapping** field, enter `PPFConnect`.

For example: The following figure shows correct configurations of supported grant types and identity mapping:



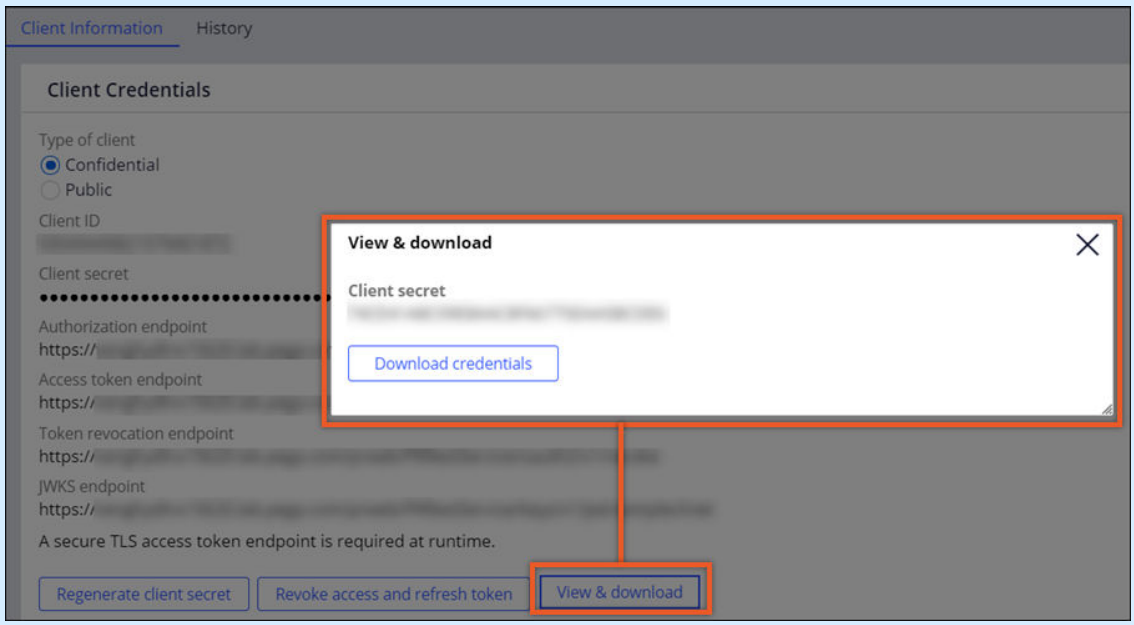
Security configurations for connection with Pega Process Fabric Hub

5. In the **Client information** section, click View & download.

Result: The **View & download** window opens and displays your client secret.

6. In the **View & download** window, click Download credentials.

For example: The following figure shows how to view and download the secure credentials:



Obtaining client ID and password

7. In the **Save As** dialog box, select a secure location to save the file that stores your credentials, and then click Save.
8. Click Save.

Result: You use the client ID and the client secret to register every application that you associate with a current operator.

Activating applications in Pega Process Fabric Hub

Start processing assignments from multiple applications within one window by activating your applications in Pega Process Fabric Hub. After you register and activate your application, you can process Interwoven Worklist that includes tasks from different applications, without logging in and out between the applications.

1. Log in to Pega Process Fabric Hub as a configurator.
2. In the navigation pane of Pega Process Fabric Hub, click **Registered applications**.
3. In the header of the **Registered applications** section, click **Register application**.
4. In the **Register application** dialog box, enter the details of the application that you want to add:
    a. In the **Platform** list, select the platform that stores your application. The default option is `Pega`.
    b. In the **System URL** field, enter the application alias URL for applications that you host in Pega Platform version 8.4 or later, or the system URL for applications that you host in Pega Platform version 8.3 or earlier. Use the `https://domain/context-root/auth-servlet-path` format.

    > **For example:** Enter `https://uplustelco.com/prweb/PRAuth/SSO`.

    As part of the remote system URL, you can provide authentication servlet details. Pega Process Fabric Hub uses authentication information when launching the URL to render remote assignments.

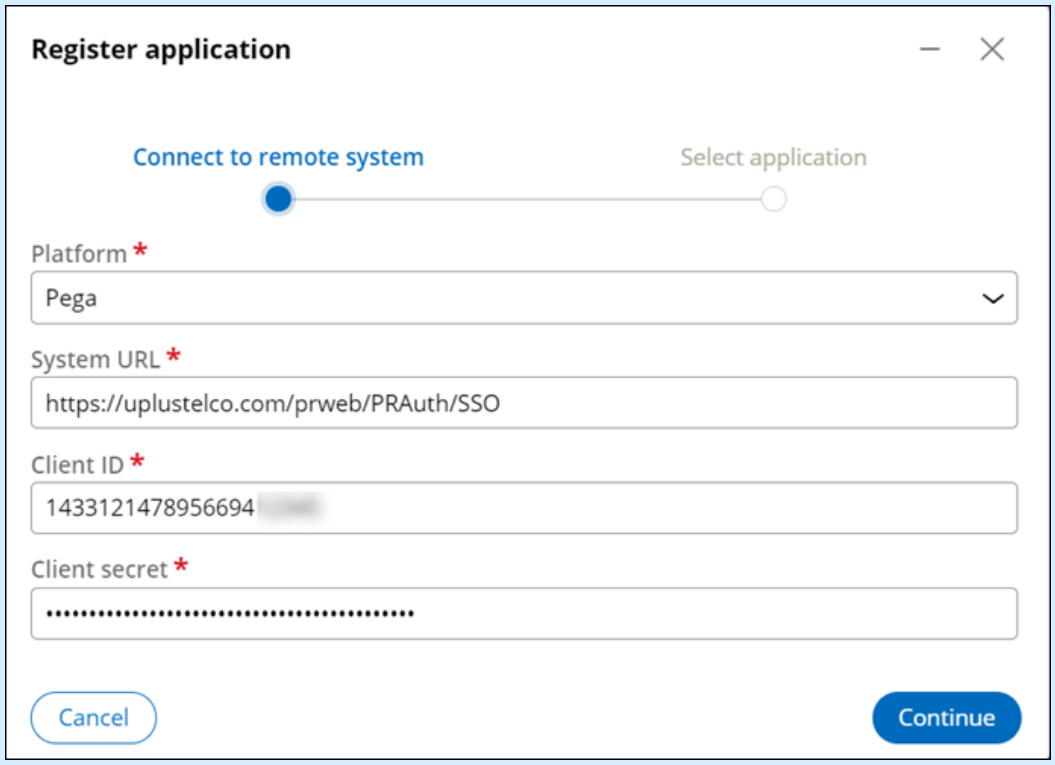
    If you have single sign on (SSO) configured for your application, providing the authentication servlet details ensures that the process involves the correct servlet and users that are already authenticated do not need to perform any additional authentication. As a result, your application uses SSO when a user opens an assignment by using Interwoven Worklist or Next Best Work in Pega Process Fabric Hub.

    For a non-SSO-based system, the recommendation is to use the Application URL alias, which you can find on the Application Definition tab of your application.

    c. In the **Client ID** field, enter the OAuth 2.0 client registration ID that you generate when creating the OAuth 2.0 client registration.

d. In the **Client Secret** field, enter the client secret that you generate when creating the OAuth 2.0 client registration.
e. Click **Continue**.

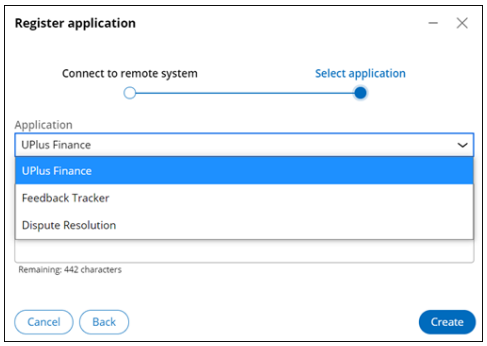
For example: The following figure shows sample data that an application registration requires:



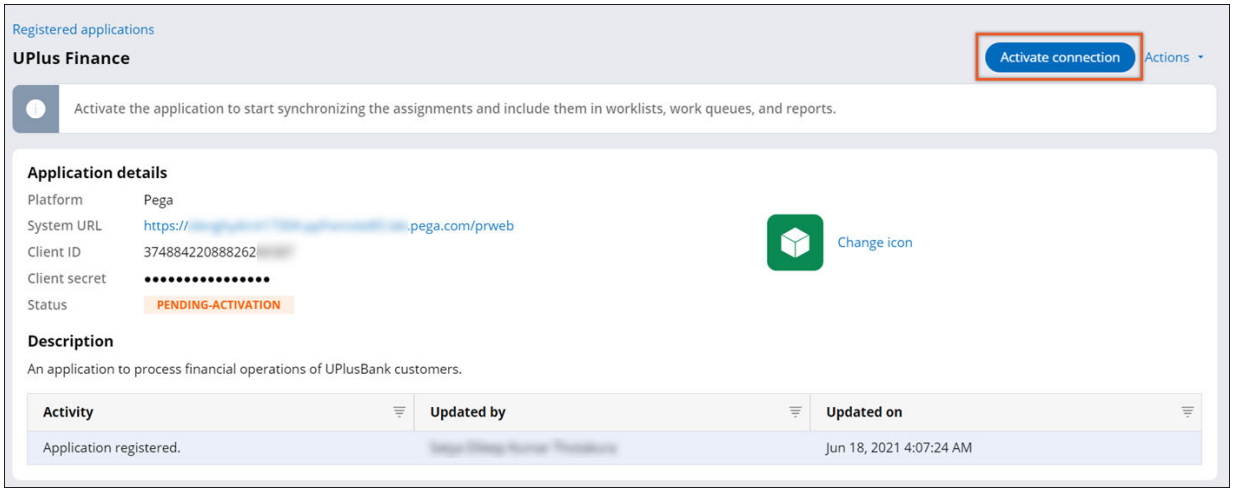
Registering an application

In the **System URL** field, the `uplustelco.com` is a domain name, `prweb` is a root context, and `PRAuth/SSO` is the authentication servlet.

5. Continue the process depending on the registration status:

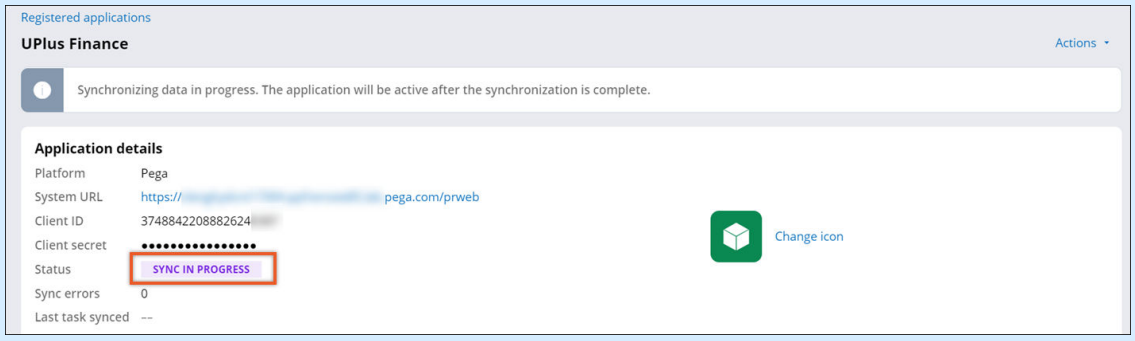
Choices	Actions
Application registration is successful	a. In the **Register application** dialog box, in the Application list, select an application that you want to register, and then click **Create**, as in the following figure:  **Selecting an application** **Result:** The landing page of your application opens. Your application has the `Pending-Activation` status. b. Activate your application by performing actions from step 6.
Application registration fails	Analyze and troubleshoot issues that cause registration failure. For more information, see Troubleshooting application registration errors.

6. On the landing page of your application, click **Activate connection**, as in the following figure:



Activating a new application

Result: Your application has the `Sync in progress` status, as shown in the following figure:



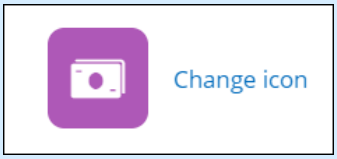
Activating a newly registered application

7. View the activation result by clicking **Actions** > **Refresh**, and then verify the activation status:

Choices	Actions
The status is `Sync-InProgress`	The synchronization process is still in progress. Wait until the synchronization is complete and the activation automatically changes the status to `Active`.
The status is `Active`	Your application is active in Pega Process Fabric Hub. You can now manage work in your Interwoven Worklist.
The status is `Activation-Failure`	Determine and eliminate the cause of the activation failure: a. Analyze the activation error message. For more information, see Troubleshooting application registration errors. b. Fix the issue as necessary. c. In the **Activation details** section, click **Activate connection**, and then refresh the page.

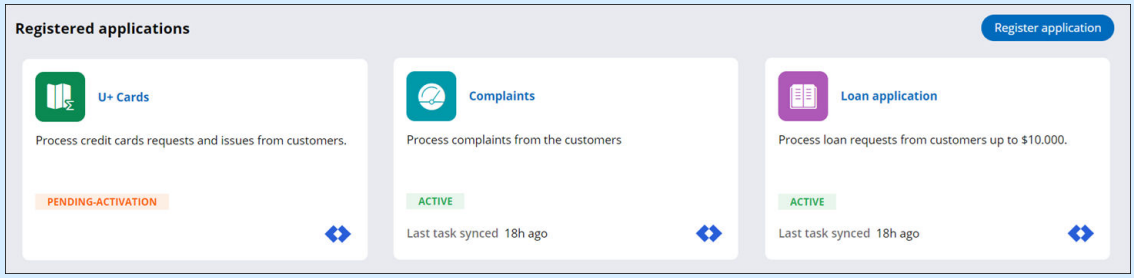
8. **Optional:** To use an application icon that corresponds with your business requirements, in the **Application details** section, click **Change icon**, and then in the icons list, select a new graphic.

For example: For a loan requests application, select an icon with a stack of banknotes, as in the following example:



A sample icon to associate with an application

Result: A list of your registered applications displays icons that are distinguishable and relevant for your applications, as in the following figure:



Registered applications list with distinguishable icons

Result: Pega Process Fabric Hub now fetches assignments from your registered application.

For example:

What to do next:

- Configure the priority of assignments that Pega Process Fabric Hub fetches from your registered application by assigning appropriate weighting. For more information, see Setting the weighting for your applications.
- Ensure that Pega Process Fabric Hub reflects the latest settings of your application, for example, after you edit work queues, and matches your current business needs by managing application life cycle. For more information, see Managing applications in Pega Process Fabric Hub.

Modification of the default access groups with administrative privileges

To register an application in Pega Process Fabric Hub, your access group needs to have administrative privileges. The default administrative access groups are *Application*:`Administrators` and *Application*:`Authors`. To meet your unique business requirements, you can create your custom access group with administrative privileges. To use your custom access group to register applications in Pega Process Fabric Hub, edit the *AllowedRegistrationAccessGroups* dynamic system setting.

The default value of the *AllowedRegistrationAccessGroups* dynamic system setting is `All:Administrators,All:Authors`. You edit the dynamic system setting by adding your custom access group in the comma-separated list.

For more information, see Editing a dynamic system setting .

Filtering assignments for publication

Publish only specific assignments from your remote application with the Pega Process Fabric Hub Connector component to Pega Process Fabric Hub. For example, you can prevent assignments that are related to a specific case type from publication.

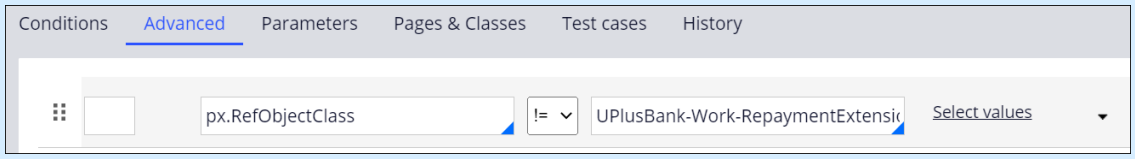
Before you begin:

- Ensure that your remote application has the Pega Process Fabric Hub Connector installed. For more information, see Installing the Pega Process Fabric Hub connector component and Pega Process Fabric Hub Connectors.
- You perform the configuration in your remote application. For more information, see Registering and managing remote applications in Pega Process Fabric Hub.

Saving the When rule for publishing real-time information about assignments in your ruleset

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Decision** category, and then click **When**.
3. In the list of When rule instances, open the *pyPF_EnablePublishRealtime* When rule that applies to the *Assign-* class.
4. On the rule form header, click **Save as**.
5. On the **Save as When** form, in the **Context** section, in the **Add to ruleset** list, select your application ruleset.
6. Click **Create and open**.
7. On the **Advanced** tab, update the condition and logic string.

For example: The following figure shows a configuration that prevents the assignments for the RepaymentExtension case type from publication:



A sample configuration of the pyPF_EnablePublishRealtime When rule

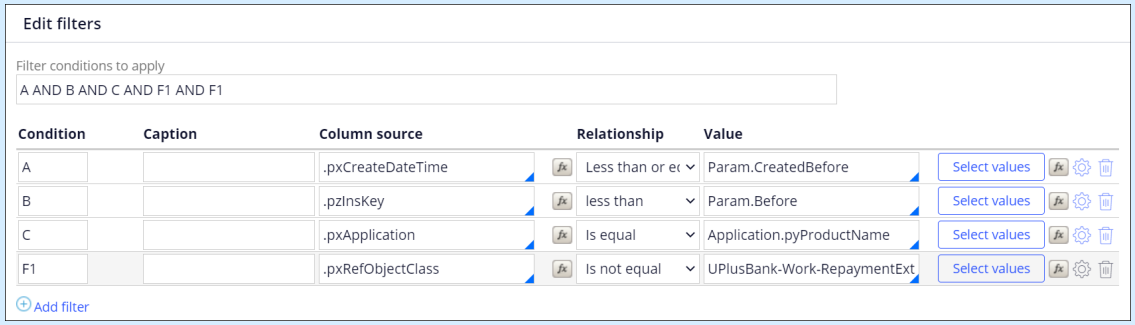
8. Click **Save**.

Saving the report definition rules for publishing bulk information about assignments in your ruleset

9. In the navigation pane of Dev Studio, click **Records**.
10. Expand the **Reports** category, and then click **Report definition**.
11. In the list of report definition instances, open the *pyPF_PublishBulk* report definition that applies to the *Assign-Worklist* class.

12. On the rule form header, click Save as.
13. On the **Save as Report Definition** form, in the **Context** section, in the Add to ruleset list, select your application ruleset.
14. Click Create and open.
15. In the **Edit filters** section, update the filter conditions.

For example: The following figure shows filters that prevent the assignments for the RepaymentExtension case type from publication:



A sample configuration of the pyPF_PublishBulk report definition

16. Click Save.
17. In the list of report definition instances, open the *pyPF_PublishBulk* report definition that applies to the *Assign-WorkBasket* class.
18. Repeat steps 12 through 16.

Modification of the standard mappings published to Pega Process Fabric Hub

Extensions pass fields from your application to Pega Process Fabric Hub so that Pega Process Fabric Hub reflects data from your application. Overriding default extensions maps the values between your application and Pega Process Fabric Hub in a way that is accurate for your unique business scenario.Additional fields that Pega Process Fabric Hub includes are `Customer`, `Product`, and `Account`. With the additional fields, you can

present more detailed assignment data in Pega Process Fabric Hub. If your application stores related data in fields with different names, by overriding default extensions, you ensure that information between your application and Pega Process Fabric Hub passes correctly.

Standard assignment data

Pega Process Fabric Hub includes the following standard fields to store assignment data:

- `RegistrationID`
- `AssingTo`
- `AssignToType`
- `CaseTypeClass`
- `Status`
- `TaskID`
- `Label`
- `Priority`
- `TaskCreateDateTime`
- `TaskCreateOperator`

The `Customer`, `Product`, and `Account` fields are additional and you can use these fields to pass more detailed data to Pega Process Fabric Hub by using extensions.

You add the Pega Process Fabric extensions to your application when you install the Pega Process Fabric Hub connector component. The extensions use both bulk sync and real-time APIs to pass data from your applications to Pega Process Fabric Hub. If you want to pass values from fields in your application, first you need to identify the class that stores the field value in your Pega Platform system.

The *Assign-* classes

To pass values that the system stores in the *Assign-Worklist* or *Assign-WorkBasket* classes, you override the following items:

- The *Assign-Worklist.pyPF_PublishBulk* and *Assign-WorkBasket.pyPF_PublishBulk* report definitions by adding your columns to these reports.

  The bulk sync that runs during the registration activation process uses these reports to fetch assignment data from the database. However, real-time sync uses assignment BLOB for sourcing assignment data. As a result, overriding the *Assign-Worklist.pyPF_PublishBulk* and *Assign-WorkBasket.pyPF_PublishBulk* report definitions impacts only bulk sync.

- The *Assign-.pyPF_SetRequest* data transform by adding your field, for example `RequestPage.label =` *Primary.pxTaskLabel*.

  Both bulk sync and real-time sync use this data transform to prepare the task request.

The *Work-* classes

To pass values that the system stores in your *Work-* class, you override the following items:

- The *Work-.pyPF_GetCases* report definition by adding your column to this report.

  Both bulk sync and real-time sync use this report definition to fetch case information in the task request preparation.

- The *Work-.pyPF_SetRequest* data transform by adding your field, for example the `RequestPage.caseLabel =` *Primary.pyLabel* format.

  Both bulk sync and real-time sync use this data transform to prepare the task request.

Other classes

If the value belongs to a class that is different than the *Assign-* or *Work-* class, you override the *System-Queue-PPF-Connect-Assign* data transform by adding your field, for

example `RequestPage.accountID = Primary.pyAccountId`. Both bulk sync and real-time sync use this data transform to prepare the task request.

Order of preference

Avoid including a field in multiple extensions. However, if your business process requires the same field in multiple extensions, the system considers classes that store fields in the following order:

1. The *System-Queue-PPF-Connect-Assign.pyPF_SetRequest* data transform
2. Your *Work-* class
3. The *Work-* class
4. The *Assign-* class

For example, if you include the same field in the *Work-* and *Assign-* classes, the system references the field from the *Work-* class.

Filtering case types for publication

Specify which case types you want to allow or block when you publish case type data from your remote application with the connector component to Pega Process Fabric Hub.

Before you begin:

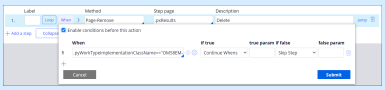
- Ensure that your remote application has the Pega Process Fabric Hub Connector for Pega Platform version 8.6 that is compatible with Pega Process Fabric Hub version 2.1. For more information, see Pega Process Fabric Hub Connectors and Installing the Pega Process Fabric Hub connector component.
- You perform the configuration in your remote application. For more information, see Registering and managing remote applications in Pega Process Fabric Hub.

By default, all case types in the **Cases & data** tab are published to Pega Process Fabric Hub.

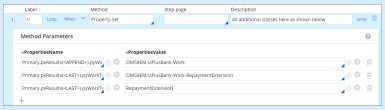
You filter the case types that you want to publish to Pega Process Fabric Hub from your remote application by updating a Pega Process Fabric Hub extension activity rule in your application.

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Technical** category, and then click **Activity**.
3. In the list of Activity rule instances, open the *pyPF_GetCaseSyncClassesExt* Activity rule that applies to the *Code-Pega-List* class.
4. On the rule form header, click **Save as**.
5. On the **Save As Activity** form, in the **Context** section, in the **Add to ruleset** list, select your application ruleset.
6. Click **Create and open**.
7. On the **Steps** tab, update the list of case types that your application publishes to Pega Process Fabric Hub:

Choices	Actions
Remove a case type	a. Click **Add a step**. b. Add a loop by clicking **Loop**, and then, in the **Repeat** section, select **For each embedded page**. c. Click **Submit**. d. Add a When condition by clicking **When**. e. Select the **Enable conditions before this action** checkbox. f. In the **When** field, enter `.pyWorkTypeImplementati`

Choices	Actions
	`onClassName==`*the name of the implementation class of the case type that you want to remove*. g. In the **if false** field, select **Skip Step**. **For example:**  **A sample configuration of the pyPF_GetCaseSyncClassesExt extension activity** h. Click **Submit**. i. In the **Method** field, enter `Page-Remove` j. In the **Step page** field, enter `.pxResults`. **Note:** Ensure that you have the *.pxResults* data page and the *Embed-Application-WorkMetaData* class added to the pages and classes list on the

Choices	Actions
	Pages & Classes tab of the extension activity.
Add a case type	a. Click Add a step. b. In the Method field, enter `Property-Set`, and then click Expand to see method parameters. c. In the PropertiesName filed, enter `Primary.pxResults(<APPEND>).pyWorkPoolName`, and then in the PropertiesValue fields, enter the name of the work pool for the case type that you want to add. d. Click Add item. e. In the PropertiesName filed, enter `Primary.pxResults(<LAST>).pyWorkTypeImplementationClassName`, and then in the PropertiesValue fields, enter the name of the implementation class of the case type that you want to add. f. Click Add item. g. In the PropertiesName filed, enter `Primary.pxResults(<LAST>).pyWorkTypeName`, and then in the PropertiesValue fields, enter the name of the case type that you want to add. **For example:**

Choices	Actions
	 **A sample configuration of the pyPF_GetCaseSyncClassesExt extension activity**

8. Click Save.

Filtering cases for publication

Publish only those cases that meet a specific condition from your remote application to Pega Process Fabric Hub.

Before you begin:

- Ensure that your remote application has the Pega Process Fabric Hub Connector. For more information, see Installing the Pega Process Fabric Hub connector component and Pega Process Fabric Hub Connectors.
- You perform the configuration in your remote application. For more information, see Registering and managing remote applications in Pega Process Fabric Hub.

By default, your remote application publishes all the cases of the selected case types to Pega Process Fabric Hub.

Saving the When rule for publishing real-time information about cases in your ruleset

1. In the navigation pane of Dev Studio, click **Records**.
2. Expand the **Decision** category, and then click **When**.
3. In the list of When rule instances, open the *pyPF_EnablePublishRealtime* When rule that applies to the *Work-* class.
4. On the rule form header, click **Save as**.
5. On the **Save as When** form, in the **Context** section, in the **Apply to** field, enter the class of the case type.
6. In the **Add to ruleset** list, select your application ruleset.
7. Click **Create and open**.
8. On the **Advanced** tab, update the condition and logic string.

For example: The following figure shows a configuration for publishing cases for the RepaymentExtension case type only if the value of the *Region* property is `NORTH`:

Configuration for publishing only the cases in which the value of the .Region property is NORTH

9. Click **Save**.

Saving the report definition rule for publishing bulk information about cases in your ruleset

10. In the navigation pane of Dev Studio, click **Records**.
11. Expand the **Reports** category, and then click **Report definition**.
12. In the list of Report Definition rule instances, open the *pyPF_PublishBulk* Report Definition rule that applies to the *Work-* class.

13. On the rule form header, click **Save as**.
14. On the **Save as Report Definition** form, in the **Context** section, in the **Add to ruleset** list, select your application ruleset.
15. Click **Create and open**.
16. Update the filter conditions.
17. Click **Save**.