



Talend Connectors for BPM

Reference Guide

6.4.1

Adapted for v6.4.1. Supersedes previous releases.

Publication date: June 29, 2017

Copyright © 2012-2017 Talend Inc. All rights reserved.

Notices

All brands, product names, company names, trademarks and service marks are the properties of their respective owners.

End User License Agreement

The software described in this documentation is provided under **Talend's** End User License Agreement (EULA) for commercial products. By using the software, you are considered to have fully understood and unconditionally accepted all the terms and conditions of the EULA.

To read the EULA now, visit <http://www.talend.com/legal-terms/us-eula>.

Table of Contents

Preface	v
General information	v
Purpose	v
Audience	v
Typographical conventions	v
Feedback and Support	v
Talend connectors for BPM	1
Talend ESB Web Service Client	2
Configuring the Talend ESB Web Service Client connector	2
Scenario: Using the Talend ESB Web Service Client connector to retrieve data from the <i>CRMService</i>	9
Talend ESB REST Client	26
Configuring the ESB REST Client connector	26
Scenario 1: Retrieving data from a REST service from within a process using the Talend ESB REST Client connector	32
Scenario 2: Updating user information to a remote database from within a process using Talend ESB REST Client connector	41

Preface

General information

Purpose

This Reference Guide provides use cases and details about how to set parameters for the major connectors provided by **Talend**. At this stage **Talend** only provides specific connectors as add-ons for Bonita BPM 6.5 SP.

Information presented in this document applies to **Talend** connectors for BPM delivered with **Talend 6.4.1**.

Audience



This guide is for users and administrators using **Talend** connectors for BPM.



The layout of GUI screens provided in this document may vary slightly from your actual GUI.

Typographical conventions

This guide uses the following typographical conventions:

- text in **bold**: window and dialog box buttons and fields, keyboard keys, menus, and menu options,
- text in **[bold]**: window, wizard, and dialog box titles,
- text in *courier*: system parameters typed in by the user,
- text in *italics*: file, schema, column, row, and variable names,
- The  icon indicates an item that provides additional information about an important point. It is also used to add comments related to a table or a figure,
- The  icon indicates a message that gives information about the execution requirements or recommendation type. It is also used to refer to situations or information the end-user needs to be aware of or pay special attention to.
- Any command is highlighted with a grey background or code typeface.

Feedback and Support

Your feedback is valuable. Do not hesitate to give your input, make suggestions or requests regarding this documentation or product and find support from the **Talend** team, on **Talend Community** at:

<https://community.talend.com/>



Talend connectors for BPM

Talend connectors for BPM are used to connect tasks or processes to information and resources provided by **Talend** as add-ons to Bonita BPM. Upon installation, you can select them from the connectors list in Bonita BPM and use them wherever connectors are supported within a process model.

This chapter describes the how to use the connectors provide by **Talend** in your process design. To install these connectors, unzip *talend-bonita-connectors-<version>.zip* under the *Talend-ESB-<version>\add-ons\bpm* \ directory of your **Talend** ESB Runtime to get the *talend-bonita-connectors-<version>* folder, and follow the installation steps detailed in the text file *README_TALEND.txt* included in the folder.

Talend ESB Web Service Client

Talend ESB Web Service Client connector calls any ESB Web service Provider (SOAP) provided by Talend ESB. This makes it the primary connector to leverage the Talend ESB Providers and Routes from within a process.

This connector provides support for:

- Request/Response and One Way service operations.
- Authentication by:
 - SAML Token (using the Talend ESB Security Token Service)
 - Username-Token
 - Basic HTTP Authentication
- Service Activity Monitoring (using the Talend ESB SAM Server)
- Service Discovery (using the Talend ESB Service Locator)
- Business correlation

Configuring the Talend ESB Web Service Client connector

To configure the **Talend ESB Web Service Client** connector, proceed as follows:

1. If you have not added the connector, select the **Talend ESB Web Service Client** connector from the connector list;

If you have already added it to the process or a task of the process, select the connector and click **Edit** in the **Connectors** tab of the **General** view for the process or task.

A configuration wizard opens to guide you through the configuration.

The following sections describe how to access this connector list with the real-world use case provided below.

Talend ESB Web Service Client

Name the connector
Specify connector parameters

Name *
(CRMService_getCRMInformation)

Description

Select event * enter start suspend resume finish

If connector fails...

Named Error

< Back Next > Finish Cancel

- Fill in the following information and click **Next** to go to the next step.

Field	Description
Name *	Specifies the name of the connector.
Description	Description of the connector.
Select event *	Specifies the event which triggers the connector: <ul style="list-style-type: none"> enter - event occurs when the flow enters this process step; start - event occurs when the flow starts; suspend/resume - event occurs when the user presses the suspend/resume button on the web-form; finish - event occurs when the flow exits this process step
If connector fails...	Specifies what to do if the connector fails consuming the Web service: <ul style="list-style-type: none"> Puts in failed state - puts the step in failed state and terminates the process; Ignore error and continue process - continues the process ignoring the error; Throw error event - throws a user-specified error.
Named Error	Specifies the name of the error which should be thrown if the connector fails. Available only if the Throw error event option is selected.



Fields followed by an asterisk (*) are required.

3. The following step of the wizard allows you to define a WSDL file or a WSDL URL (using ?WSDL notation) or to select a service from the *Talend ESB - Service Locator* service.

4. Fill in the following information and click **Next** to go to the next step.

Field	Description
Use WSDL file	This option allows you to select the WSDL file which describes the Talend ESB Web service. It is possible to browse to a file location of the WSDL file or to specify a ?WSDL URL where the WSDL can be directly retrieved from the Service Provider.
Use Service Locator	This option allows you to select the particular ESB Web service from the ESB Service Locator, which provides a list of all services available within the Talend ESB.

5. In this step, you can define the **Port**, **Operation** and the related request mapping. The wizard provides the specific values in drop-down fields and in the **Parameters** view depending on the WSDL selected.

Configure your Web service invocation using XML as input
Configure Web service with XML

WSDL URL:

Service:

Port:


Operation:

Parameters:

Parameter	Value
Whole variable	

< Back Next > Finish Cancel

6. Fill in the following information and click **Next** to go to the next step.

Field	Description
WSDL URL	URL where the WSDL is served (as specified in Step 2).
Service	Service name which should be consumed (the selection box shows only the values defined in the selected WSDL).
Port	Service port (the selection box shows only the values defined in the selected WSDL).
Operation	Name of the operation which will be called. The control allows selecting only available operations (as defined in the selected WSDL).
Parameters	<p>Table which contains the parameters for the operation:</p> <ul style="list-style-type: none"> • Parameter - parameter name which shows the request xml in a structured way. • Value - parameter value which provides different options to use static values, expressions or process variables for the request parameters. <p> For a parameter dealing with data of type Date that needs to be converted to a string, you may use <code>javax.xml.bind.DatatypeConverter</code> for the conversion to avoid possible errors related to your locale setting.</p>

7. This step allows you to use or not the *Talend ESB Infrastructure* services for this consumer request.

Configure TESB Features
Set all necessary information to have your TESB Features working

▼ Service Registry (If Service Registry is used, "SAM", "Security" and "Business Correlation" settings will be ignored)

Use Service Registry ☐

Authentication Username

Password

☐ Show password

Authorization Role Name

Security Token

Propagate Type

Alias

Certificate Password

☐ Show password

Correlation ID

▶ Service Locator

▶ Service Activity Monitoring

▶ Security

▶ Business Correlation

Save connector configuration...

Test Configuration

< Back Next > Finish Cancel

8. Fill in the following information and click **Next** to go to the next step.

Feature	Field / option	Description
Service Registry	Use Service Registry	Select this check box to enable the Service Registry feature, which provides dynamic endpoint lookup and allows services to be redirected based upon information retrieved from the registry. This feature works in Talend ESB Runtime only. With the Service Registry feature enabled, all the settings for the Service Activity Monitoring , Security , and Business Correlation features will be ignored.
	Authentication Username and Password	Enter the user name and password for authentication. This credential information will not be used if Propagate using Certificate is selected from the Propagate Type list.

Feature	Field / option	Description
	Authentication Role Name	If SAML token is registered in the service registry, specify the client's role if required.
	Security Token	Specify a security token string or select a variable that contains the desired token string to pass to the service the connector calls.
	Propagate Type	Select between: Propagate using U/P (default): STS authentication using authentication username and password. This option works only when a Security Token is specified. Propagate using Certificate : STS authentication using an alias and certificate password pair.
	Alias and Certificate Password	Enter the certificate alias and password for STS authentication. This credential information is used only if Propagate using Certificate is selected from the Propagate Type list.
	Correlation ID	Specify your correlation ID or select a process variable for your correlation ID to pass to the service the connector calls, in case Business Correlation is enabled based on the policies applied for the service. If you leave this field empty, and if Business Correlation is enabled, a random UUID will be generated automatically at runtime.
Service Locator	Use Service Locator	If this option is selected, the provider endpoint (URL) will be requested at runtime from the Talend ESB Service locator. In this way, the process is loosely coupled to the provider so the provider can be used based on availability, regardless of where they are hosted.
	Service Locator Properties	Optional: additional metadata can be set as a filter for the list of service providers. For example, if the provider uses a <i>location</i> metadata to specify the data center location they are running with (values <i>EU</i> and <i>US</i> for example), a row in this table with 'locator' values <i>EU</i> would only retrieve an endpoint where the provider is located in a <i>EU</i> data center.
Service Activity Monitoring	Use Service Activity Monitoring	Select this check box to enable monitoring the service calls with Service Activity Monitoring. In this way, request from the process to a ESB Provider is also tracked as a Service Activity event. If the Service Registry feature is enabled, this check box does not take effect.
	Service Activity Monitoring Custom Info	The table allows you to provide additional user-defined metadata which is stored together with the Service Activity Event.
Security	Use Security	This option allows you to secure the Web service call by standard-based security mechanism as provided by Talend ESB. If the Service Registry feature is enabled, this check box does not take effect.
	Security Type	Select between: <ul style="list-style-type: none"> • Basic: Basic HTTP authentication • SAML Token: SAML Token authentication (using the Talend ESB Security Token Service) • Username Token: Username-Token authentication
	Username and Password	Enter the user name and password for authentication. This credential information will not be used if Propagate using Certificate is selected from the Propagate Type list.
	Use Authorization	Select this check box to enable authorized call.

Feature	Field / option	Description
		<p>This option takes effect when SAML Token is selected from the Security Type list.</p> <p>For more information about the management of user roles and rights, see <i>Talend Administration Center User Guide</i> and <i>Talend ESB Infrastructure Services Configuration Guide</i>.</p>
	Role Name	With the Use Authorization check box selected, specify the client's role if required.
	Security Token	Specify a security token string or select a variable that contains the desired token string to pass to the service the connector calls.
	Propagate Type	<p>Select between:</p> <ul style="list-style-type: none"> • Propagate using U/P (default): STS authentication using authentication username and password. This option works only when a Security Token is specified. • Propagate using Certificate: STS authentication using alias and certificate password.
	Alias and Certificate Password	Enter the certificate alias and password for STS authentication.
		Enter the certificate password for STS authentication.
Business Correlation	Use Business Correlation	<p>Select this check box to enable the Business Correlation feature. This will pass the specified correlation ID via the request message header to the service the connector calls, so that chained service calls will be grouped under this correlation ID.</p> <p>With this feature enabled, the connector will also extract the correlation ID from the response message header and store it in the process variable specified in the next step for further use in the process.</p> <p>If the Service Registry feature is enabled, this check box does not take effect.</p>
	Correlation ID	Specify your correlation ID or select a process variable for your correlation ID to pass to the service the connector calls. If you leave this field empty, a random UUID will be generated automatically at runtime.

9. Set the parameters as follows to map the Web service response to variables/data types of the process, then click **Finish**.

Map outputs of this connector to process variables.

Connector output: providedscripts.BonitaXML.evaluateXF goes to Destination variable: cRM_Status

Buttons: Create data..., < Back, Next >, Finish, Cancel

Field	Description
Connector output	Allows you to select the connector output or its part for storing it in the specified variable. The list also allows you to open a dialog box which provides a structured view of the response parameter.
Destination variable	Variable where the connector output should be stored. With the Create data... option, you can create new process variables within this dialog box.

Scenario: Using the Talend ESB Web Service Client connector to retrieve data from the *CRMService*

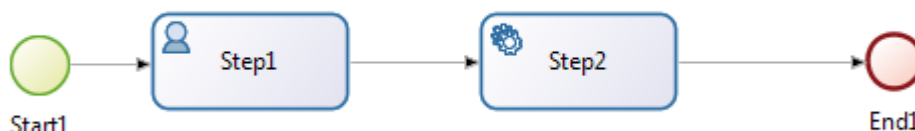
This scenario explains how to configure the **Talend ESB Web Service Client** connector to retrieve data from the *CRMService* which is provided with the *Rent-a-Car* example of Talend ESB. The data returned by the process, as well as exceptions, are then displayed in a Web browser.

Prerequisites:

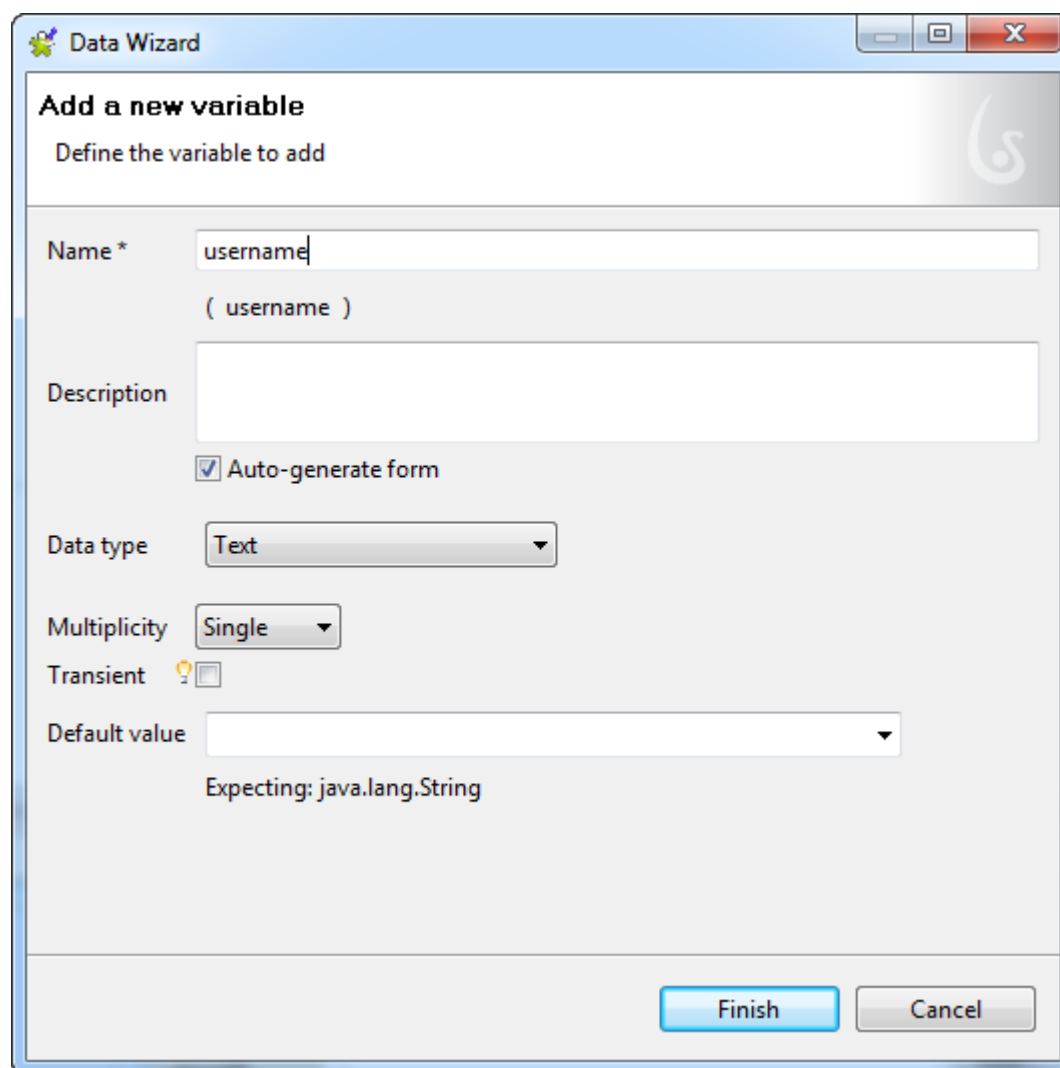
- Talend ESB Runtime (OSGI Container) v5.1.1 is installed and running.
- STS, Service Locator and Service Activity Monitoring is enabled.
- The *Rent-a-Car* demo is built and installed as described in the *README.txt* file (which is part of the example) using the `-Pa11` option to enable all the features (Service Locator, Service Activity Monitoring and STS).
- The *CRMService* is deployed and started.

Creating a new process

1. Click **New** from the **Process** menu to create a new process, which already has a start event and a human task.
2. Drag and drop a new task, **Step2**, from the context palette of **Step1**.
3. Drag and drop an End event, **End1**, from the context palette of **Step2**.



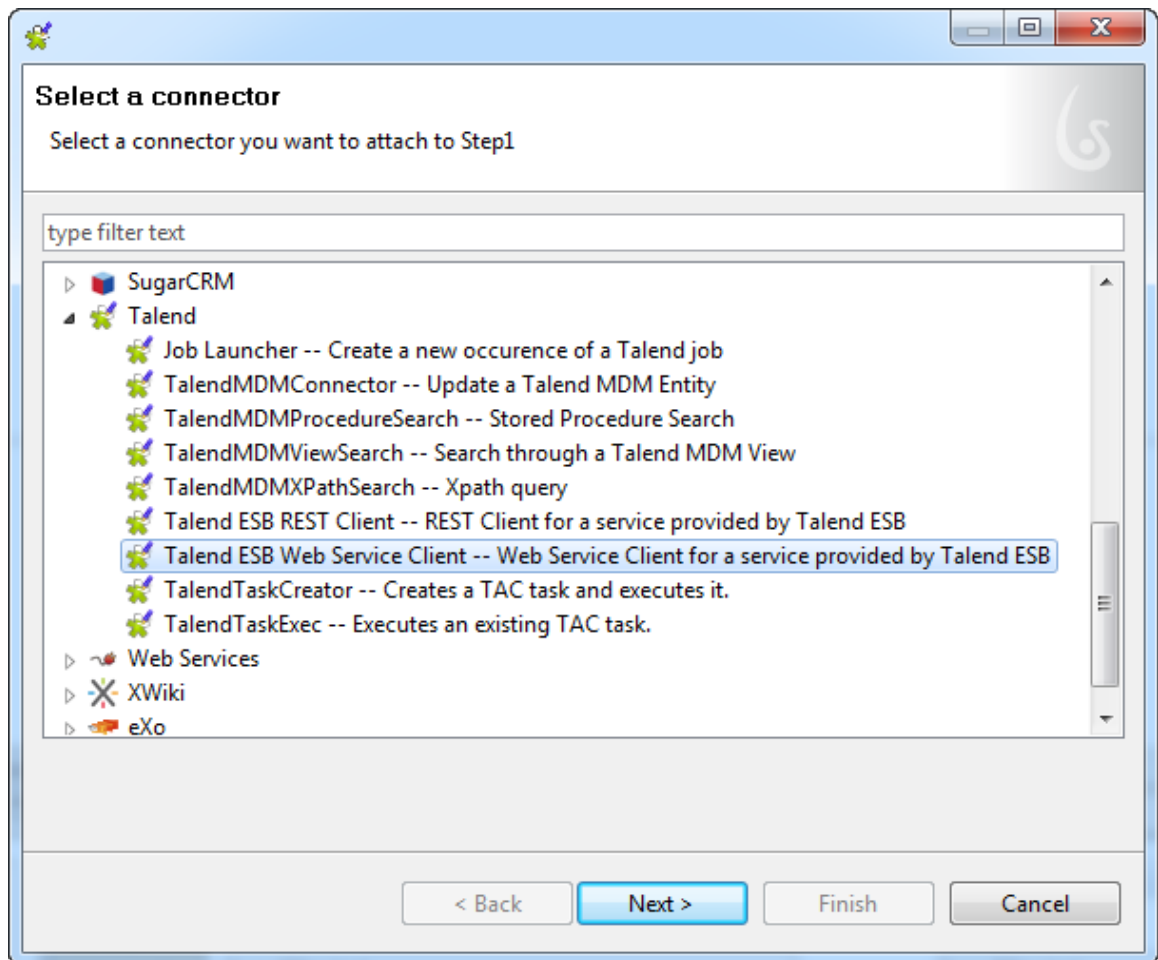
4. Click the task labelled **Step 1** in the process design workspace. From the **General** view, select the **Data** tab and click the **Add...** button.
5. Add the variable *username*, select the **Auto-generate form** check box. You may also specify the default value in the **Default value** field.



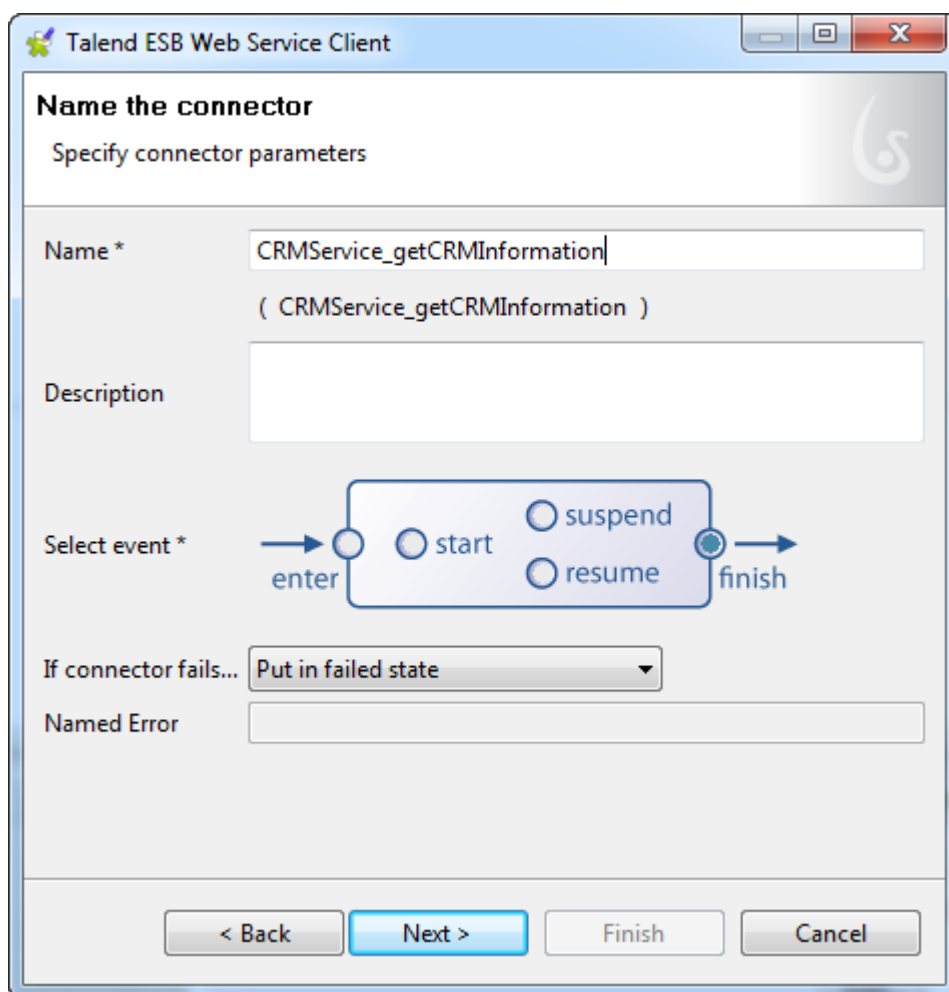
6. Click **Finish**. A form allowing you to enter the *username* variable is generated.

Adding and configuring the Talend ESB Web Service Client connector

1. From the **General** view, select the **Connectors** tab and click the **Add...** button. Then in the connector wizard select **Talend > Talend ESB Web Service Client** then click **Next**.

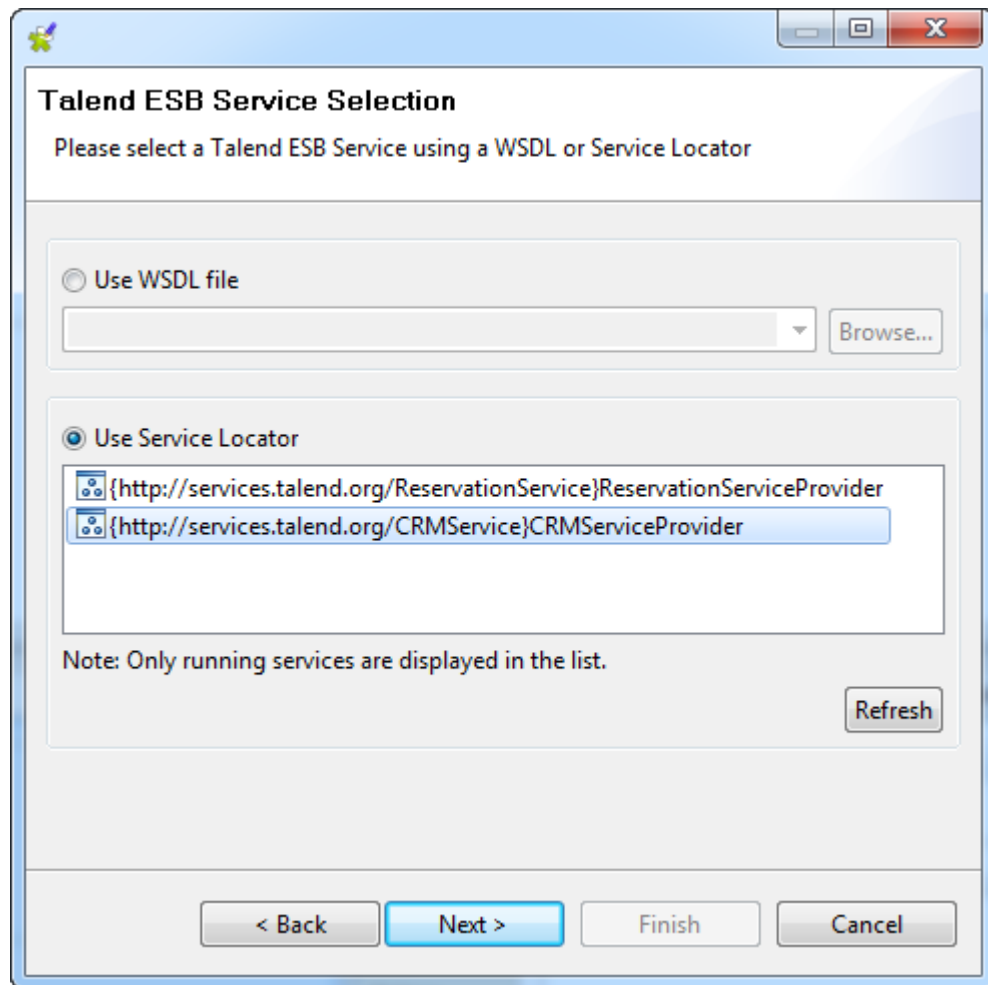


2. Name the connector *CRMService_getCRMInformation* and click **Next**.



The image shows a screenshot of the 'Talend ESB Web Service Client' dialog box. The title bar reads 'Talend ESB Web Service Client'. The main heading is 'Name the connector', followed by the subtitle 'Specify connector parameters'. The 'Name *' field contains the text 'CRMService_getCRMInformation', with '(CRMService_getCRMInformation)' displayed below it. The 'Description' field is empty. The 'Select event *' section features a state transition diagram with an 'enter' event, a 'start' state, and two options: 'suspend' and 'resume'. The diagram ends with a 'finish' event. Below this, the 'If connector fails...' dropdown menu is set to 'Put in failed state', with an empty 'Named Error' field underneath. At the bottom, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

3. Select the **Use Service Locator** option and select the *CRMServiceProvider* line from the available services list. Click **Next**.



4. From the **Operation** list, select **getCRMInformation**, expand the **Whole variable** line, select **username**, assign the value of the *username* variable to it by clicking its **Value** field and selecting the variable, and then click **Next**.

Configure your Web service invocation using XML as input
Configure Web service with XML

WSDL URL:

Service:

Port:

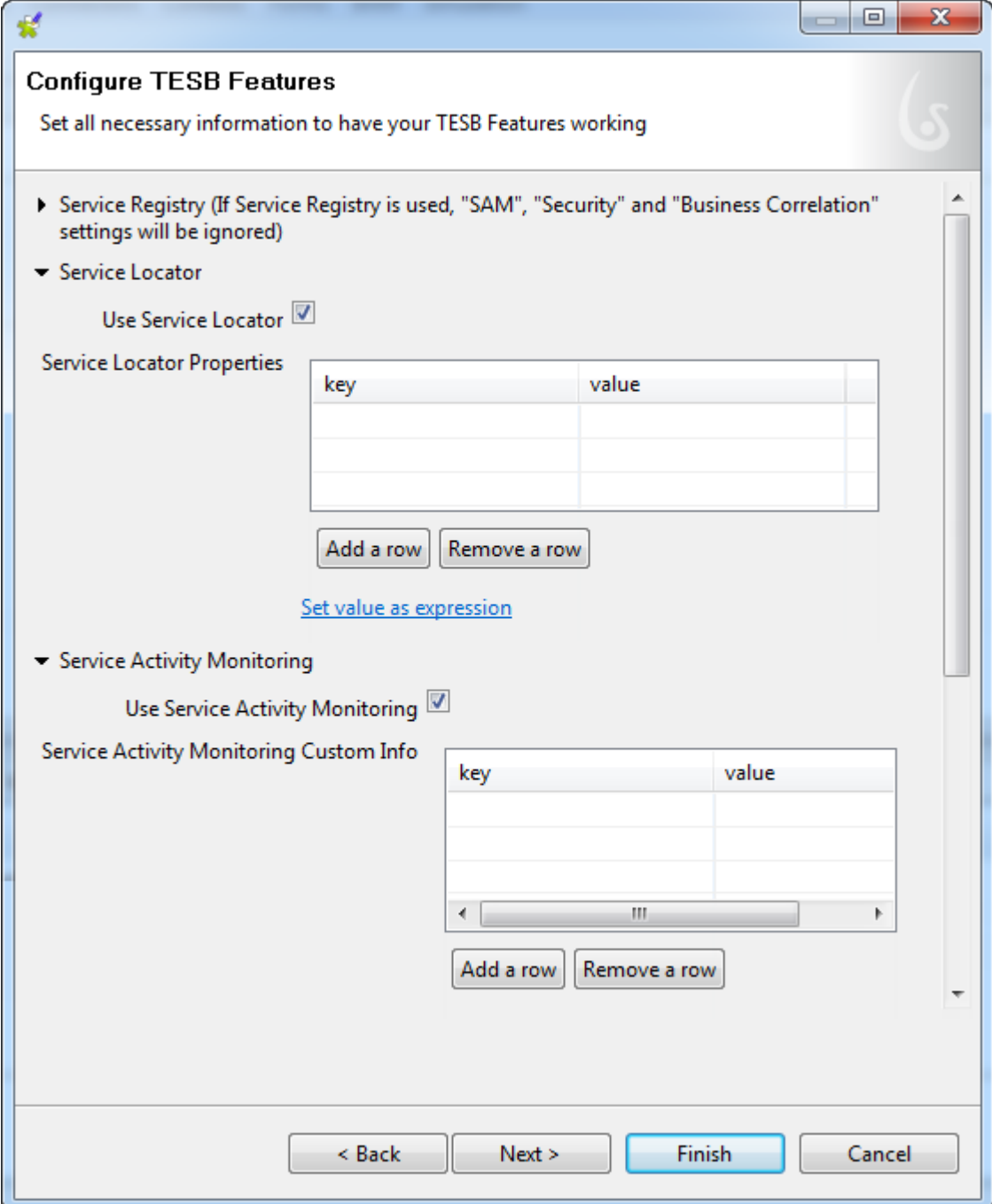
Operation:

Parameters:

Parameter	Value
Whole variable	
tns:LoginUser	
name	
username	\${username}
email	
... Append as child	

< Back Next > **Finish** Cancel

5. Select the **Use Service Locator** and **Use Service Activity Monitoring**.



Configure TESB Features
Set all necessary information to have your TESB Features working

► Service Registry (If Service Registry is used, "SAM", "Security" and "Business Correlation" settings will be ignored)

▼ Service Locator

Use Service Locator ☒

Service Locator Properties

key	value

[Set value as expression](#)

▼ Service Activity Monitoring

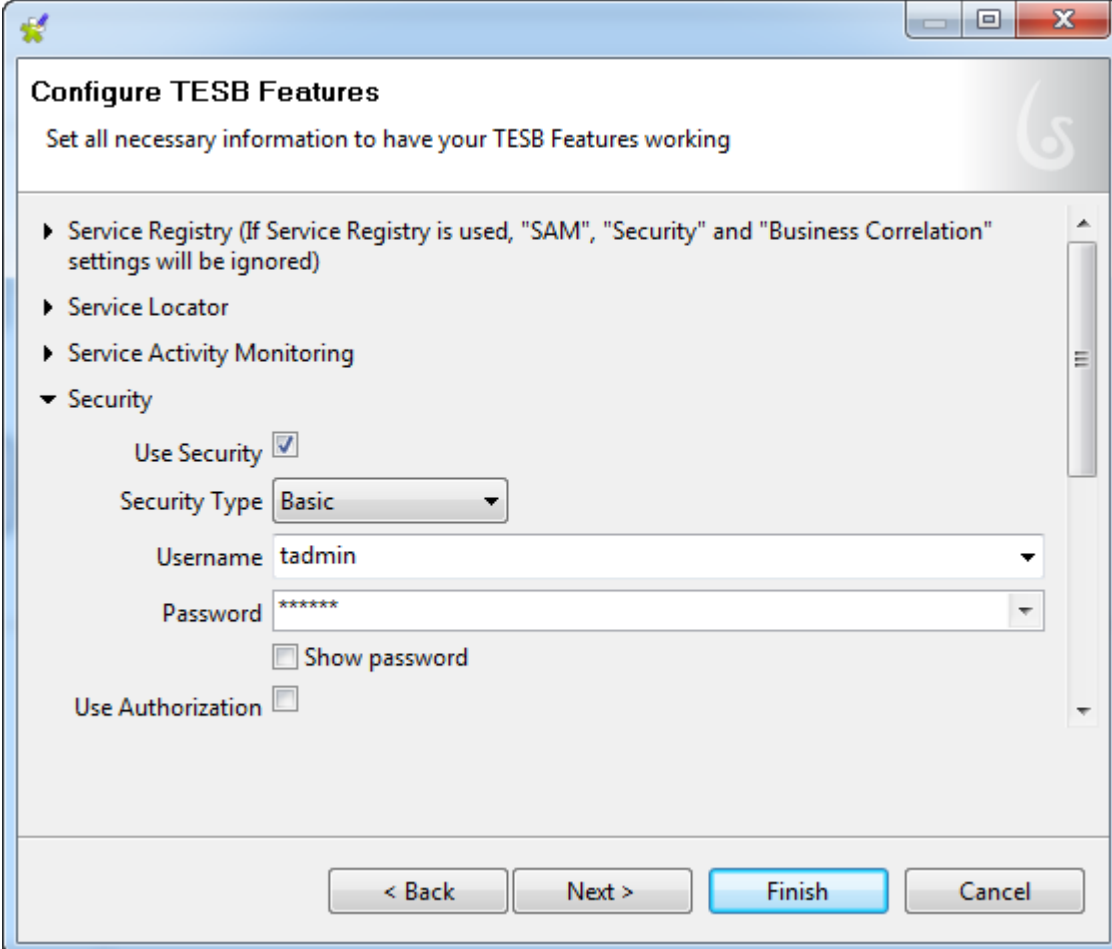
Use Service Activity Monitoring ☒

Service Activity Monitoring Custom Info

key	value

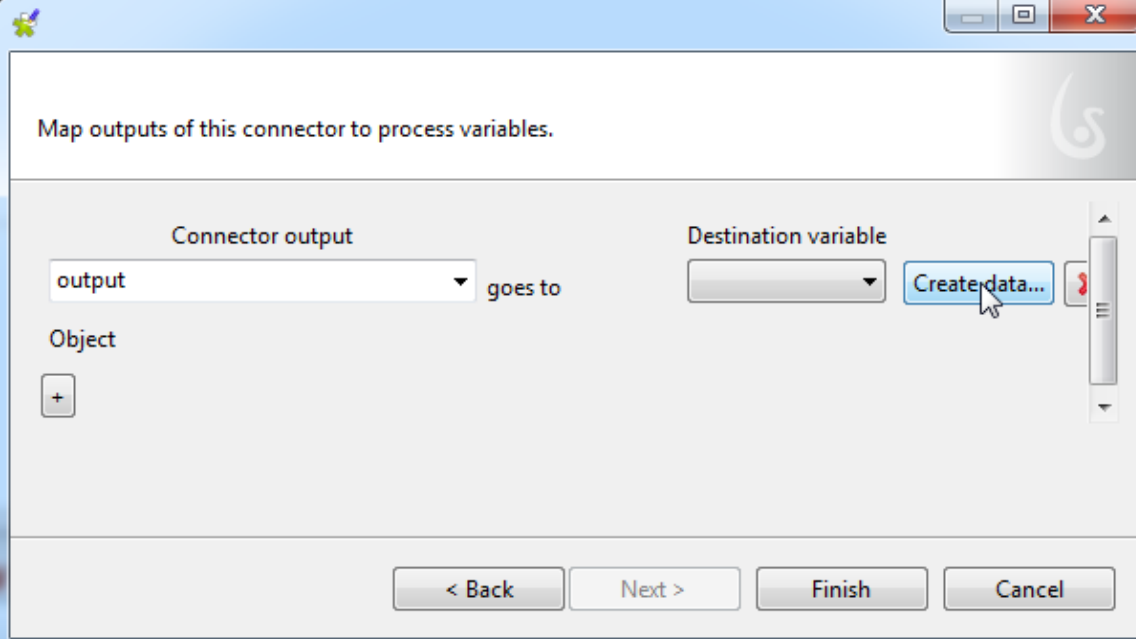
< Back Next > **Finish** Cancel

6. Select the **Use Security** check box, select **Basic** from the **Security Type** list, and enter *tadmin* in both **Username** and **Password** fields. Then click **Next**.



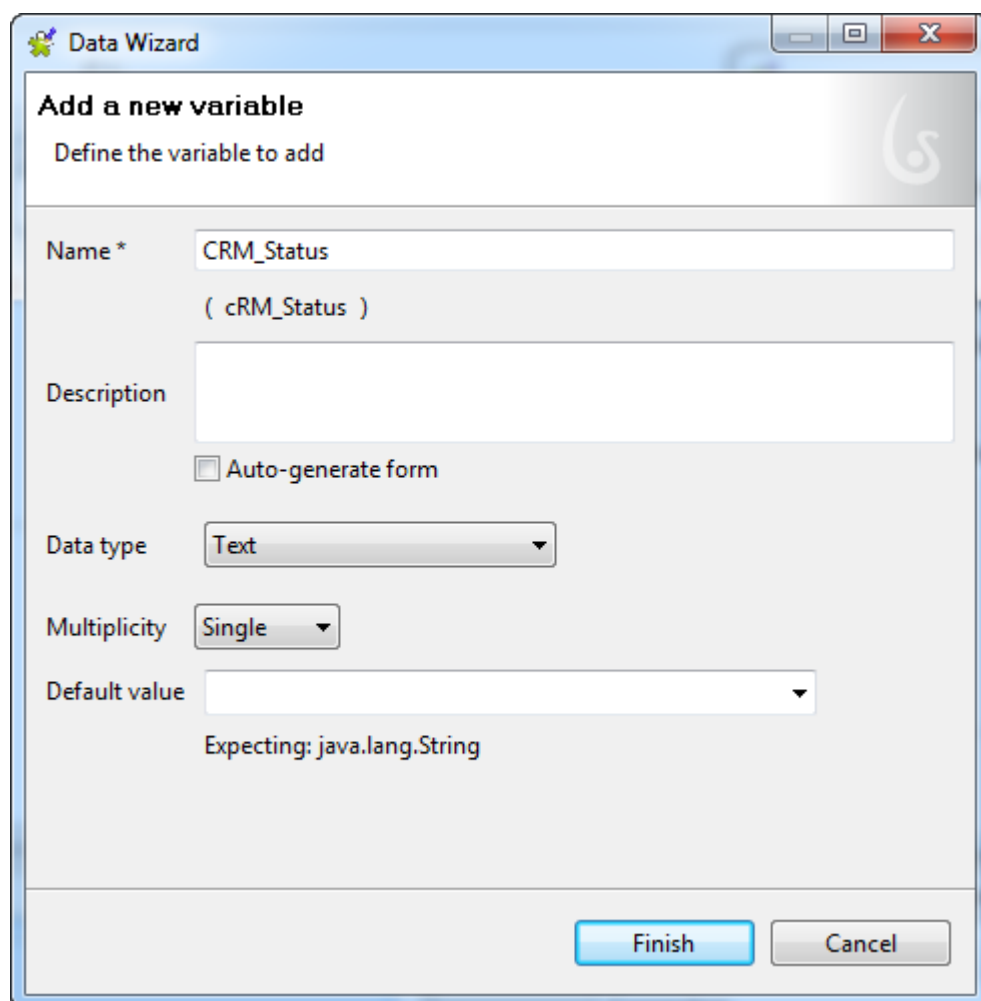
The dialog box is titled "Configure TESB Features" with a subtitle "Set all necessary information to have your TESB Features working". It contains a list of features: Service Registry, Service Locator, Service Activity Monitoring, and Security. The Security section is expanded, showing "Use Security" checked, "Security Type" set to "Basic", "Username" as "tadmin", and "Password" as "*****". There are checkboxes for "Show password" and "Use Authorization". Navigation buttons at the bottom include "< Back", "Next >", "Finish", and "Cancel".

7. Click the **Create data...** button.



The dialog box is titled "Map outputs of this connector to process variables." It shows a mapping table with "Connector output" and "Destination variable" columns. The "output" connector output is mapped to an empty destination variable field. A "Create data..." button is next to the empty field. There is also an "Object" section with a "+" button. Navigation buttons at the bottom include "< Back", "Next >", "Finish", and "Cancel".

8. Name the variable *CRM_Status*, clear the **Auto-generate form** check box, then click **Finish** to close the variable window.

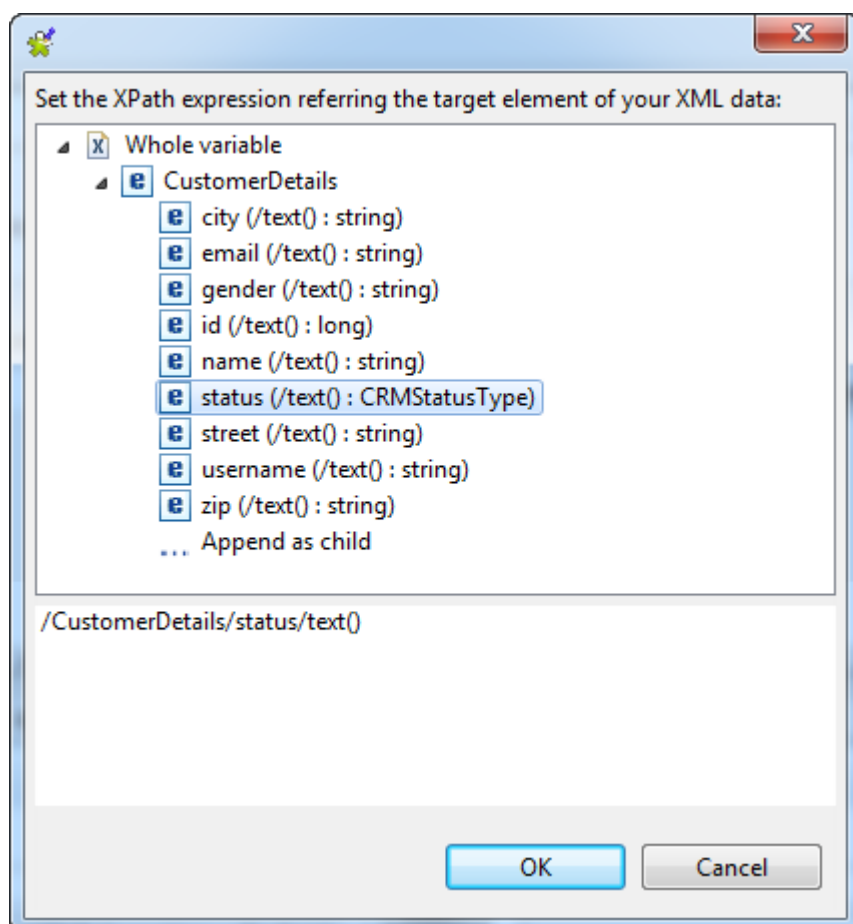


The image shows a 'Data Wizard' dialog box with the title 'Add a new variable'. The subtitle is 'Define the variable to add'. The dialog contains the following fields and options:

- Name ***: A text box containing 'CRM_Status'. Below it, the text '(CRM_Status)' is displayed.
- Description**: An empty text box.
- Auto-generate form**: An unchecked checkbox.
- Data type**: A dropdown menu showing 'Text'.
- Multiplicity**: A dropdown menu showing 'Single'.
- Default value**: An empty text box.
- Below the default value field, the text 'Expecting: java.lang.String' is displayed.

At the bottom right, there are two buttons: 'Finish' and 'Cancel'.

9. Select the output connector named **output...**, expand the **Whole variable** line, and select **status**. Then click **OK** to close the window.

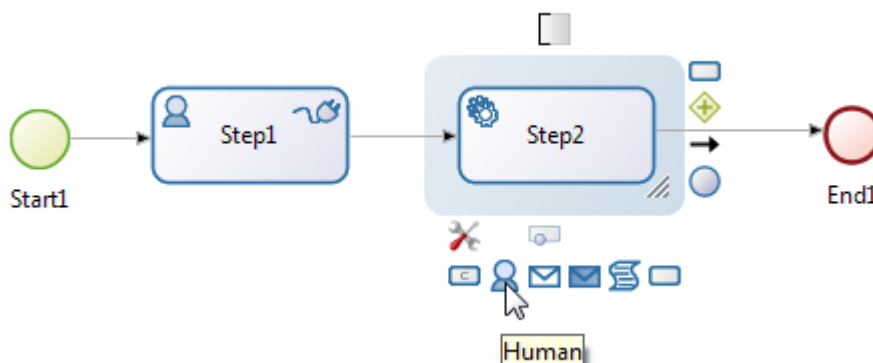


In this way, a request to the *CRMService* (getCRMInformation) Operation is defined with a user name as a request parameter and the *status* of this user within the CRM Application as a result.

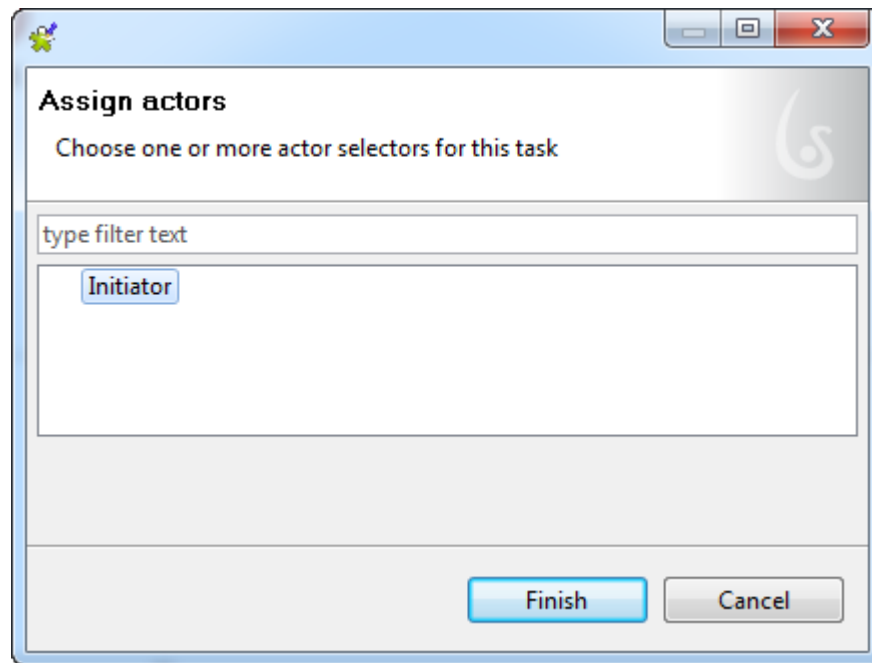
Displaying the operation result in a Web browser

Next, you need to display the *CRM_Status* variable in a process Web User Interface to show the returned status.

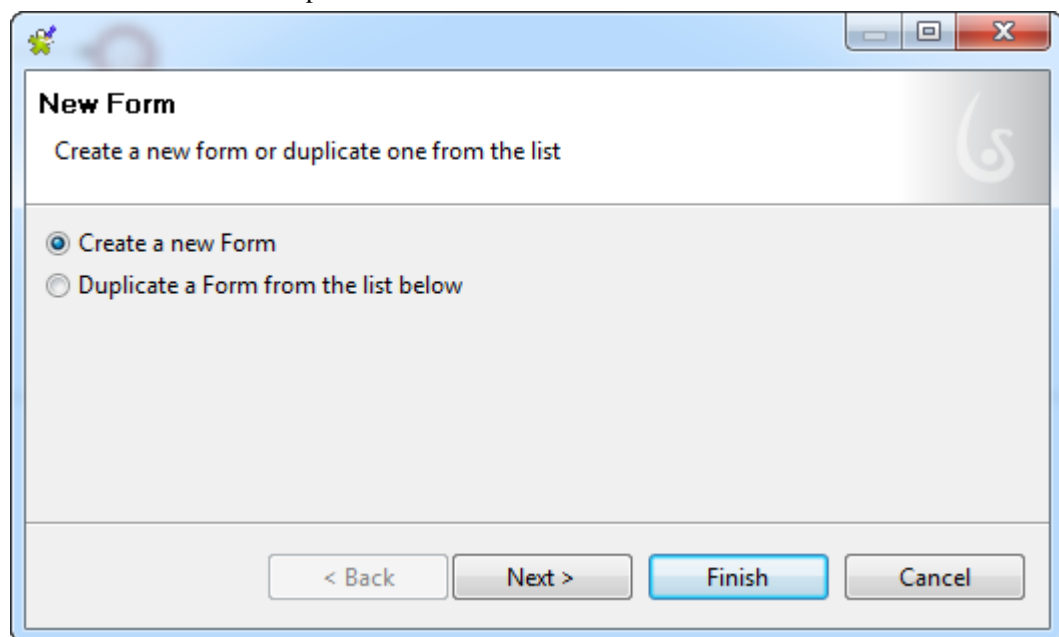
1. Click the task labelled **Step2** in the process design workspace, click the tools icon which is on the bottom-left corner, and then select the **Human** icon. The task type is changed to *Human*.



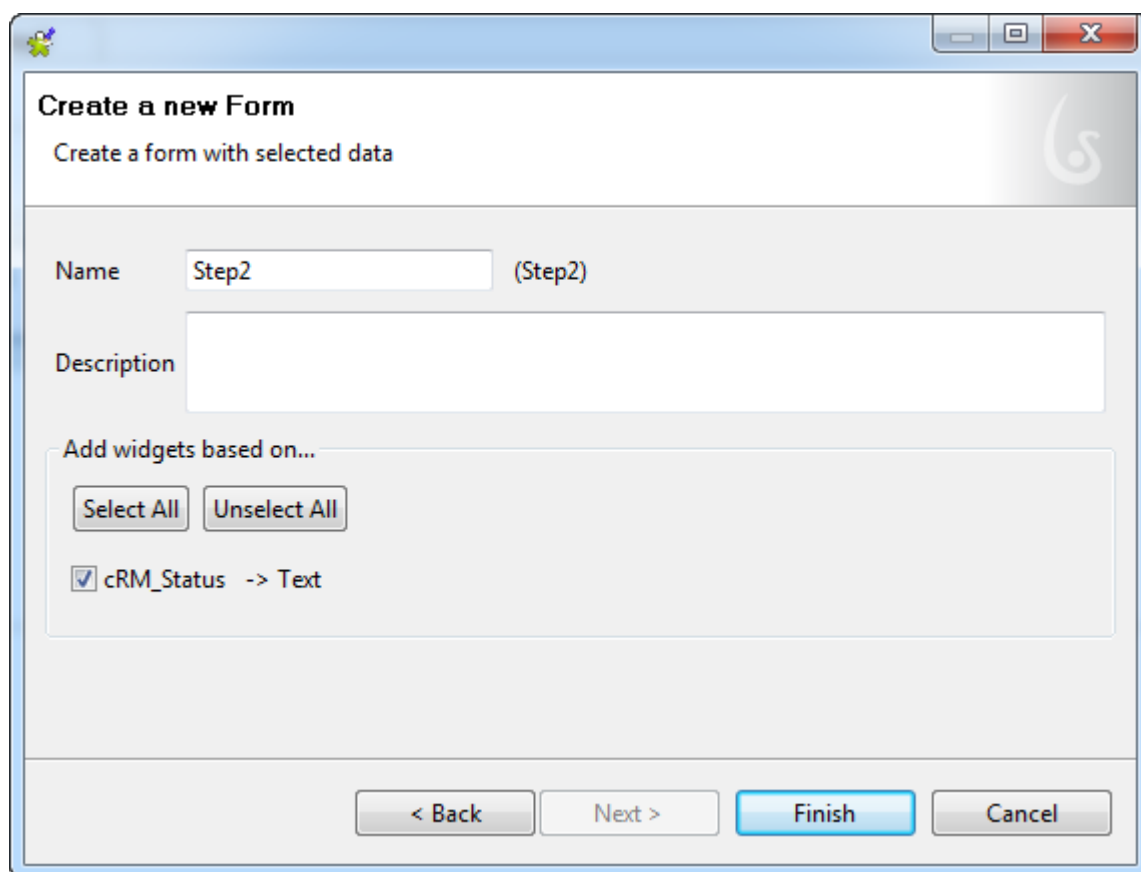
2. From the **General** view, select the **Actors** tab, click the **Choose...** button to open the [Assign actors] dialog box. Select **Initiator** and click **Finish** to close the dialog box.



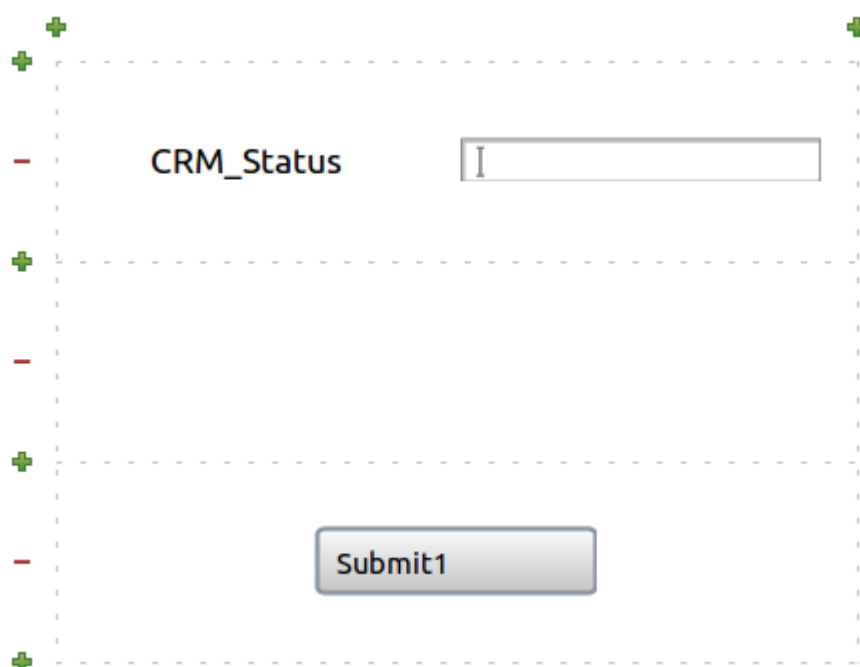
3. From the **Application** view, click the **Entry Pageflow** tab and click the **Add...** button next to the first table.
4. Select the **Create a new form** option and click **Next**.



5. On the next step, configure which variables will be shown on the form. Select the *cRM_Status* check box and click **Finish**.



A form will be created that will display the *CRM_Status* variable in the process Web User Interface to show the returned status.



6. Run the process. After deploying the bundle, the studio opens the Web page in your default browser.

MyProcess

Case start : MyProcess

Submit

7. Click the **Submit** button to start the process and complete the first step.
8. Fill in the **user name** field and click the **Submit** button. The `getCRMInformation` *CRMService* Operation is called and the `username` parameter is populated with the user-specified value.

MyProcess

Step1

From: May 26, 2012 6:17 PM To: Priority: **Normal**

username:

Submit

9. After a few seconds, the **Step2** form is returned by the server, showing the result of the operation, and Talend ESB shows the corresponding log messages.

MyProcess

Step2

From: May 28, 2012 3:13 PM To: Priority: **Normal**

admin

CRM_Status

Submit1

Managing exceptions

The following example shows how to deal with errors while consuming the Web service.

1. Delete the connection between the **Step1** and **Step2** tasks, drop the **XOR Gateway** from the **Palette** onto the workspace between **Step1** and **Step2**, then connect **Step1** to the **Gateway** and connect the **Gateway** to **Step2**.
2. Select the **Gateway** output line. On the **General** view, enter the condition `cRM_Status != null` in the **Condition** field.



General
Application
Appearance
Simulation

General

Name

Description

Condition

☐ Default flow

3. On the **General** view of the **Step1** icon, select the **Connectors** tab then select the *CRMService_getCRMInformation* connector and click **Edit**. Select the **Ignore error and continue process** reaction in the field named **If connector fails....** Click **Next** and then **Finish**.

Talend ESB Web Service Client

Name the connector
Specify connector parameters

Name * CRMService_getCRMInformation
(CRMService_getCRMInformation)

Description

Select event *
enter start suspend resume finish

If connector fails... Ignore error and continue process

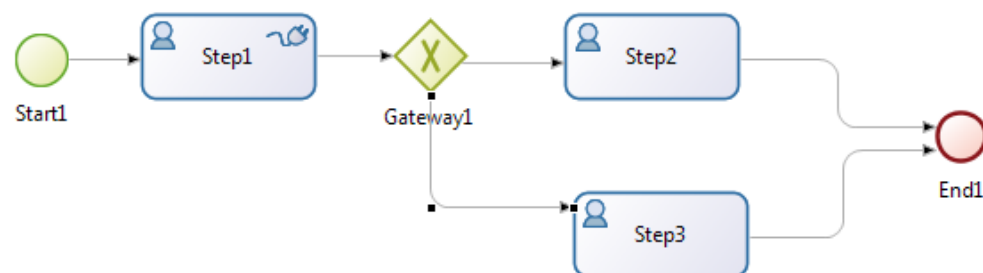
Named Error

< Back Next > Finish Cancel

This allows the process to continue regardless of the response returned by the Web service.

Later you will be able to process exceptional situations. Now, as the process will continue even in case of exception, you are able to process errors.

4. Drop a new human task, **Step3**, from the Palette onto the process design workspace, and assign the **Initiator** actor to it as we did for **Step2**.
5. Add a branch to the **Gateway** and connect it to **Step3**, and then connect **Step3** to the end event of the process.
6. Select the alternative **Gateway** branch. On the **General** view, enter the condition `cRM_Status == null` in the **Condition** field.

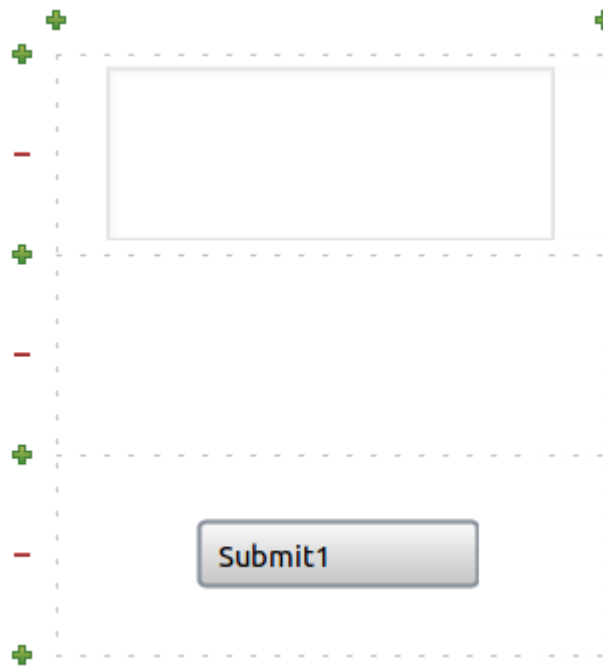


The screenshot shows the 'General' tab of a connector configuration in Talend Studio. The 'Name' field is empty. The 'Description' field is empty. The 'Condition' field is set to 'cRM_Status == null'. There is a checkbox for 'Default flow' which is unchecked.

7. Create a new form for **Step3**, with the check box for the *cRM_Status* variable cleared. This form will display the error message.

The screenshot shows the 'Create a new Form' dialog box. The 'Name' field is 'Step3 (Step3)'. The 'Description' field is empty. The 'Add widgets based on...' section shows a checkbox for 'cRM_Status -> Text' which is unchecked. The 'Finish' button is highlighted.

8. Add the **Message** element on the form. Select the **Data** tab of the **General** view and enter the text: *Error occurred while calling the CRM Service!* in the **Initial value** field in the **Form** perspective.



Next, emulate the exceptional situation: shut down the OSGI container of Talend ESB so that the Web services are no longer available.

9. Run your process. After trying to consume the unavailable service, you get the following screen.

MyProcess

Step3	
From: May 28, 2012 3:42 PM	To: Priority: Normal
admin	
Error occurred while calling the CRM Service!	
<input type="button" value="Submit1"/>	

Talend ESB REST Client

The **Talend ESB REST Client** connector calls a REpresentational State Transfer (REST) Web service provided by Talend ESB. This makes it a primary connector to leverage the Talend ESB Providers and Routes from within a process.

This connector provides support for:

- Request/Response and One Way service operations.
- Security types:
 - Basic HTTP authentication
 - Digest authentication
- Service Activity Monitoring (using the Talend ESB SAM Server)
- Service Discovery (using the Talend ESB Service Locator)
- Business Correlation

Configuring the ESB REST Client connector

To configure the **Talend REST Client** connector, proceed as follows:

1. If you have not added the connector, select the **Talend ESB REST Client** connector from the connector list;

If you have already added it to the process or a task of the process, select the connector and click **Edit** in the **Connectors** tab of the **General** view for the process or task.

A configuration wizard opens to guide you through the configuration.

2. Complete the following information and click **Next** to go to the next step.

Field / option	Description
Name *	Name of the connector.
Description	Description of the connector.
Select event *	Event that triggers the connector: <ul style="list-style-type: none"> • enter - event occurs when the flow enters this process step; • start - event occurs when the flow starts; • suspend/resume - event occurs when the user presses the suspend/resume button on the web-form; • finish - event occurs when the flow exits this process step
If connector fails...	Action to take if the connector fails consuming the Web service: <ul style="list-style-type: none"> • Put in failed state - puts the step in failed state and terminates the process; • Ignore error and continue process - continues the process ignoring the error; • Throw error event - throws a user-specified error.
Named Error	Specifies the name of the error which should be thrown if the connector fails. Available only if the Throw error event option is selected.



Fields followed by an asterisk (*) are required.

3. The following step of the wizard allows you to define the properties of the REST service to be called from within the process.

4. Fill in the following information and click **Next** to go to the next step.

Field / option	Description
Endpoint URL	URL of the REST service to be invoked. When the Use Service Locator check box is selected on the next page of this wizard, this field is not used and the URL of the REST service will be obtained from the Service Locator server automatically.
Relative Path	Relative path of the REST service to be invoked. If Use Service Locator is disabled: You can enter any of the first part of the address in the Endpoint URL field, and the second part in the Relative Path field. If Use Service Locator is enabled: The URL part will be given by the Service Locator. In this case, you need to know the URL part, and specify the rest part in Relative Path . This depends on the service you request.
Method	From this list, select an HTTP method that describes the desired action. The specific meanings of the HTTP methods are subject to definitions of your Web service provider. Listed below are the generally accepted HTTP method definitions: - GET : retrieves data from the server end based on the given parameters. - POST : uploads data to the server end based on the given parameters. - PUT : updates data based on the given parameters, or if the data does not exist, creates it. - DELETE : removes data based on the given parameters.

Field / option	Description
Accept Type	Select the media type the client end is prepared to accept for the response from the server end. Available options are XML , JSON , and ANY . When ANY is selected, the response message can be of any type and will be transformed into a string.
Query	Specify the URI query parameters in the form of name-value pairs. This option is mostly used with the GET method.
Request	This option is required only when the POST or PUT HTTP method is selected. Content Type: Select XML , JSON , or FORM according to the media type of the content to be uploaded to the server end. Request: Payload of the request message.
Security	Use Security: Select this check box if authentication is required on the REST server end. Once selected, you need to provide your username and password. Security Type: choose a security type between Basic HTTP and Digest .
Headers	Type in the name-value pair(s) for HTTP headers to define the parameters of the requested HTTP operation.
Timeouts	Connection timeout: Set the amount of time, in seconds, that the client will attempt to establish a connection before it times out. If set to 0, the client will continue to attempt to open a connection indefinitely. (default: 30) Receive timeout: Set the amount of time, in seconds, that the client will wait for a response before it times out. If set to 0, the client will wait indefinitely. (default: 60)

- This step allows you to use or not the *Talend ESB Infrastructure* services for this consumer request.

Talend ESB REST Client

Configure TESB Features
Set all necessary information to have your TESB Features working

▼ Service Locator

Use Service Locator ☐

Service Name

Service Namespace

Service Locator Properties

key	value

Add a row Remove a row

[Set value as expression](#)

► Service Activity Monitoring

▼ Business Correlation

Use Business Correlation ☐

Correlation ID

Save connector configuration...

Test Configuration

< Back Next > Finish Cancel

6. Fill in the following information and click **Next** to go to the next step.

Feature	Field / option	Description
Service Locator	Use Service Locator	Select this check box to enable the Service Locator feature. With this feature enabled, the provider endpoint (URL) will be requested at runtime from the Talend ESB Service locator. In this way, the process is loosely coupled to the provider so the provider can be used based on availability, regardless of where they are hosted.
	Service Name	Name of the service to be invoked.
	Service Namespace	Namespace of the service to be invoked.
	Service Locator Properties	Optional: additional metadata can be set as a filter for the list of service providers. For example, if the provider uses a <i>location</i> metadata to specify the data center location they are running with (values <i>EU</i> and <i>US</i> for example), a row in this table

Feature	Field / option	Description
		with 'locator' values <i>EU</i> would only retrieve an endpoint where the provider is located in a <i>EU</i> data center.
Service Activity Monitoring	Use Service Activity Monitoring	Select this check box to enable monitoring the service calls with Service Activity Monitoring. In this way, request from the process to a ESB Provider is also tracked as a Service Activity event.
	Service Activity Monitoring Custom Info	The table allows you to provide additional user-defined metadata which is stored together with the Service Activity Event.
Business Correlation	Use Business Correlation	Select this check box to enable the Business Correlation feature. This will pass a correlation ID to the service it calls via the request HTTP header, so that chained service calls will be grouped under this correlation ID. With this feature enabled, the connector will also extract the correlation ID from the response HTTP header and store it in the process variable specified in the next step for further use in the process.
	Correlation ID	Specify your correlation ID or select a process variable for your correlation ID to pass to the service the connector calls. If you leave this field empty, a random UUID will be generated automatically at runtime.

7. Set the parameters as follows to map the Web service response to variables/data types of the process, and then click **Finish**.

Field	Description
Connector output	Allows you to select the connector output or its part for storing it in the specified variable. The list also allows you to open a dialog box which provides a structured view of the response parameter.
Destination variable	Variable where the connector output should be stored. With the Create data... option, you can create new process variables within this dialog box.

Scenario 1: Retrieving data from a REST service from within a process using the Talend ESB REST Client connector

This scenario explains how to configure the **Talend ESB REST Client** connector to retrieve data from a RESTful Web service provided through a demo Job of Talend ESB.

Prerequisites:

- You have imported the ESB demo project your *Talend Studio*.
- You have launched the demo Job *DemoREST* in the **Integration** perspective of your *Talend Studio*, and you can see information like the following when entering `http://localhost:8090/services/customers` in your Web browser:

```
- <customers>
  - <customer>
    <id>1</id>
    <firstName>Theodore</firstName>
    <city>Tallahassee</city>
    <lastName>Roosevelt</lastName>
  </customer>
  - <customer>
    <id>2</id>
    <firstName>George</firstName>
    <city>Helena</city>
    <lastName>Grant</lastName>
  </customer>
  - <customer>
    <id>3</id>
    <firstName>Warren</firstName>
    <city>Harrisburg</city>
    <lastName>Tyler</lastName>
  </customer>
</customers>
```

Creating a new process and adding a variable for the input data (Step one)

1. Create a new process and select the **Step1** element in the process model.
2. From the **General** view of the **Details** panel, select the **Data** tab and click the **Add...** button.

Add a new variable
Define the variable to add

Name *
(customer_ID)

Description

☒ Auto-generate form

Data type

Multiplicity

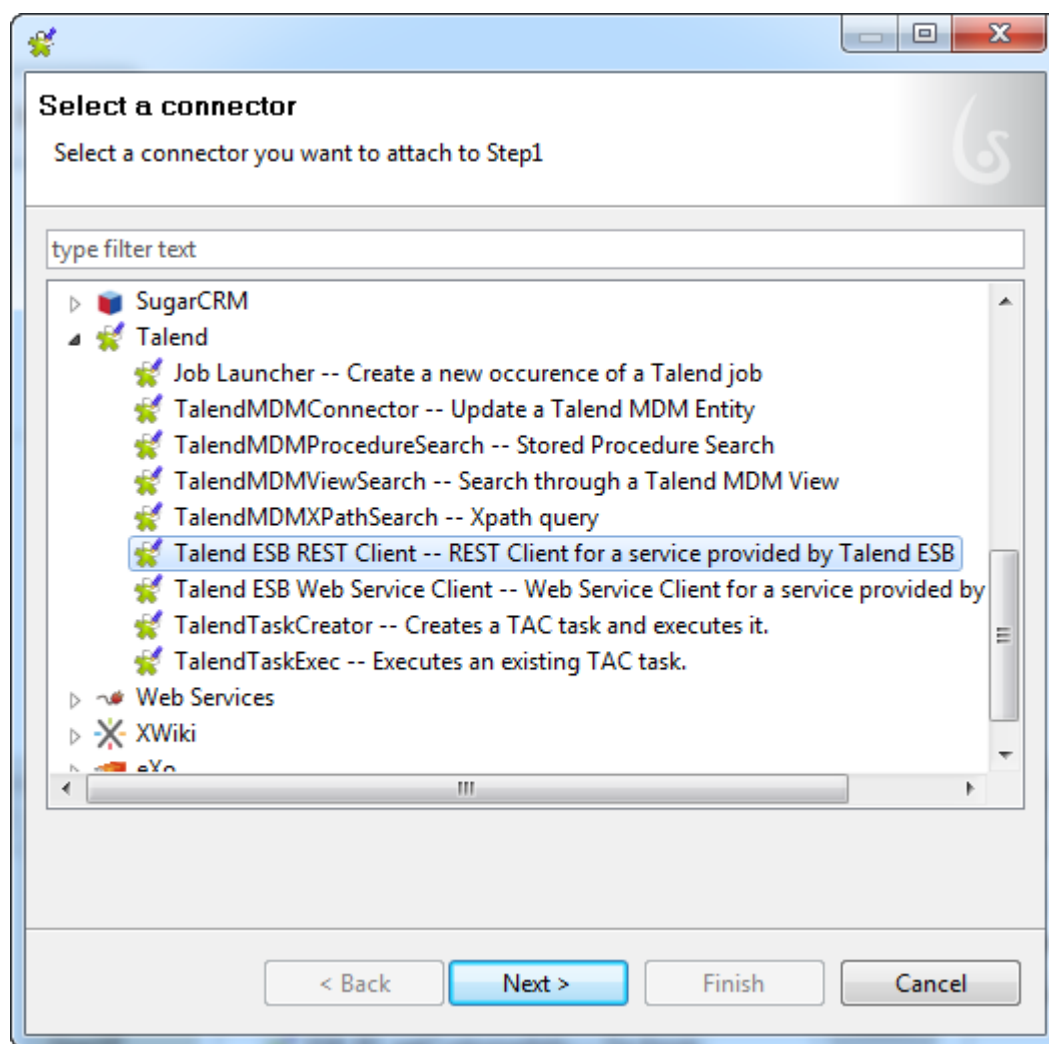
Transient ☐

Default value
Expecting: java.lang.String

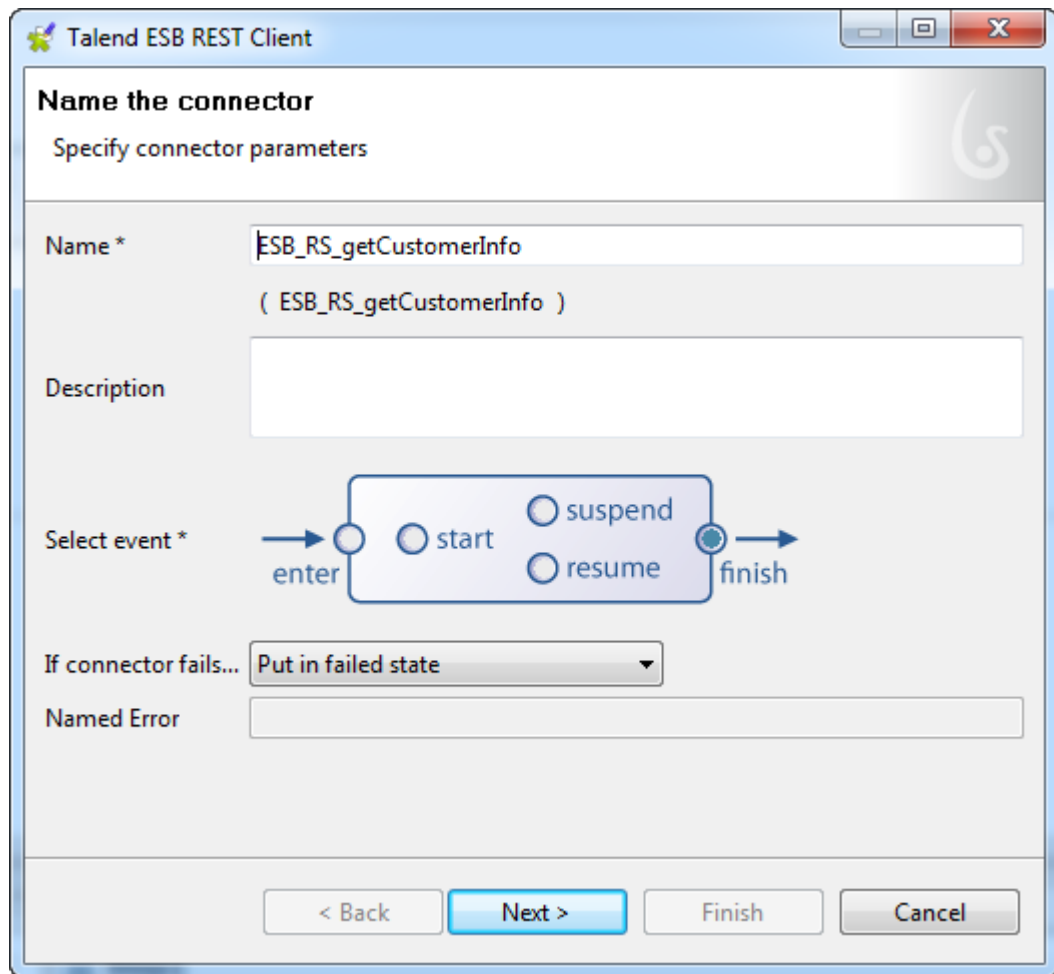
3. Name the variable *Customer_ID*, select the **Auto-generate form** check box. You can also specify the default value in the **Default value** field.
4. Click **Finish**. A form allowing you to enter the value for the *Customer_ID* variable will be generated.

Adding and configuring the Talend ESB REST Client connector

1. From the **General** view, select the **Connectors** tab and click the **Add...** button. A window opens.



2. Select **Talend > Talend ESB REST Client** then click **Next** to open the connector configuration wizard.



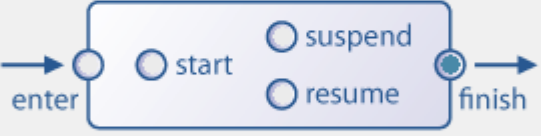
The image shows the 'Talend ESB REST Client' configuration window. The title bar says 'Talend ESB REST Client'. The main heading is 'Name the connector' with a subtitle 'Specify connector parameters'. The 'Name *' field contains 'ESB_RS_getCustomerInfo' and a hint '(ESB_RS_getCustomerInfo)'. The 'Description' field is empty. The 'Select event *' section shows a state machine diagram with 'enter', 'start', 'suspend', 'resume', and 'finish' events. The 'If connector fails...' dropdown is set to 'Put in failed state'. The 'Named Error' field is empty. At the bottom are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

Talend ESB REST Client

Name the connector
Specify connector parameters

Name *
(ESB_RS_getCustomerInfo)

Description

Select event * 

If connector fails...

Named Error

< Back Next > Finish Cancel

3. Give the connector a name, *ESB_RS_getCustomerInfo* for example, and click **Next**.

The REST Service configuration page opens.

The screenshot shows the 'Talend ESB REST Client' window with the title 'Configure your REST Service'. Below the title is the instruction 'Set all necessary information to have your REST Service working'. The main configuration area includes:

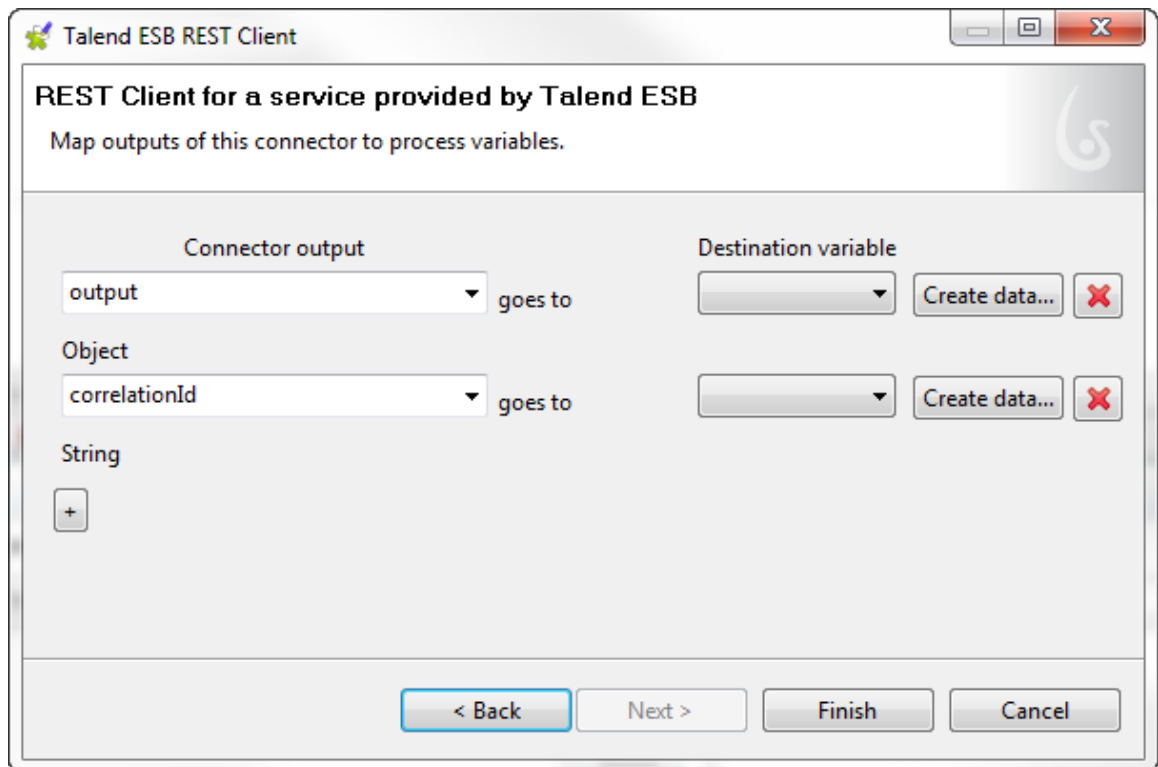
- Endpoint URL** (not used if SL enabled): A text field containing 'http://localhost:8090/services/customers'.
- Relative Path**: A dropdown menu showing '\${customer_ID}'.
- Method ***: A dropdown menu showing 'GET'.
- Accept Type**: A dropdown menu that is currently empty.

Below these fields is a list of expandable sections: Query, Request, Security, Headers, and Timeouts. To the right of the configuration fields are two buttons: 'Save connector configuration...' and 'Test Configuration'. At the bottom of the window are four navigation buttons: '< Back', 'Next >', 'Finish' (highlighted in blue), and 'Cancel'.

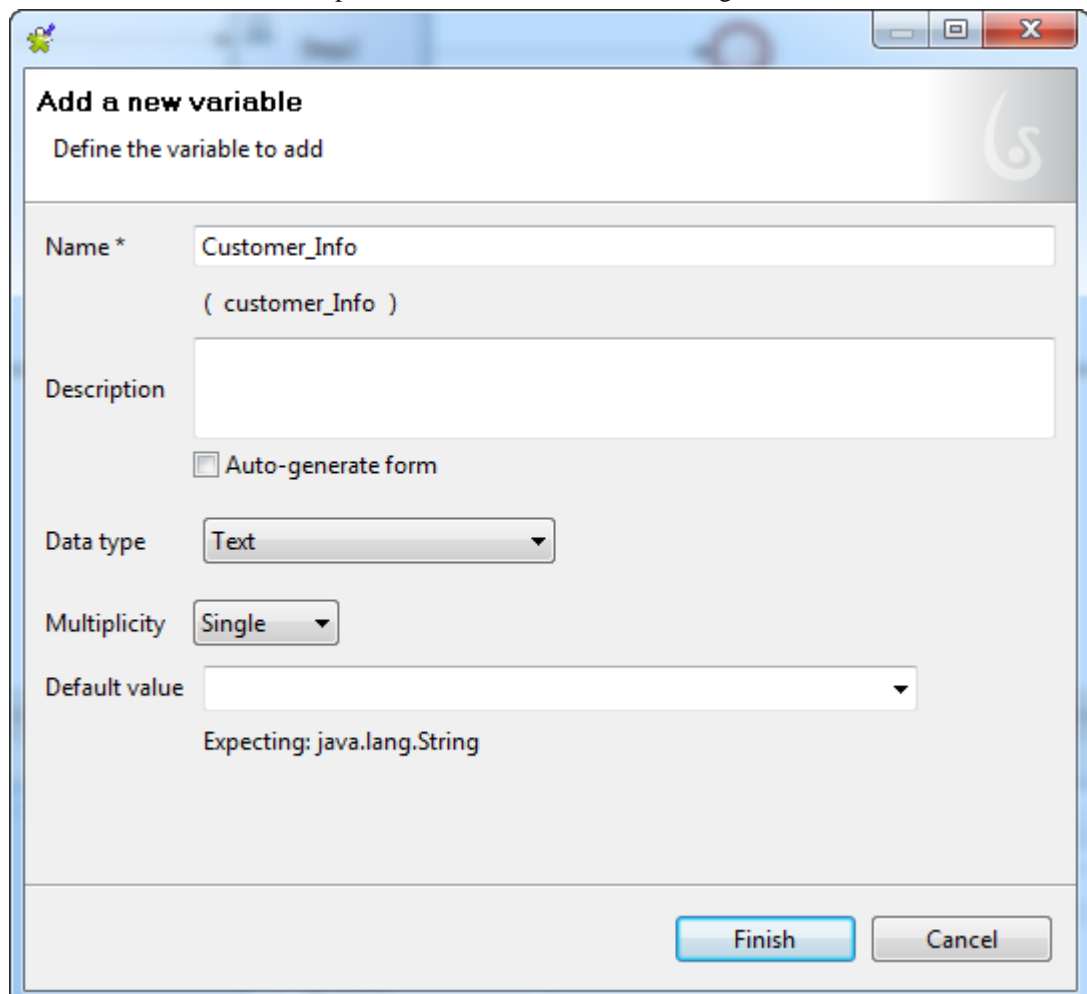
4. Configure the REST Service.

- Fill the **Endpoint URL** field with the URL of the REST Service implement by the *DemoREST* Job, `http://localhost:8090/services/customers` in this example.
- Fill in the **Relative Path** field with `${customer_ID}`, which corresponds to the variable you just created. This allows you to enter the relative path on your Web browser when the process is executed.
- From the **Method** list, select **GET**. This operation will retrieve data from the REST service to be called.

5. Click **Next**. The **Configure TESB Feature** page opens to let you configure service locator information, service activity monitoring, and business correlation. As we do not use any of these options in this example, simply click **Next** to proceed with connector output mapping.



6. Click the **Create data** button to open the [Add a new variable] dialog box.

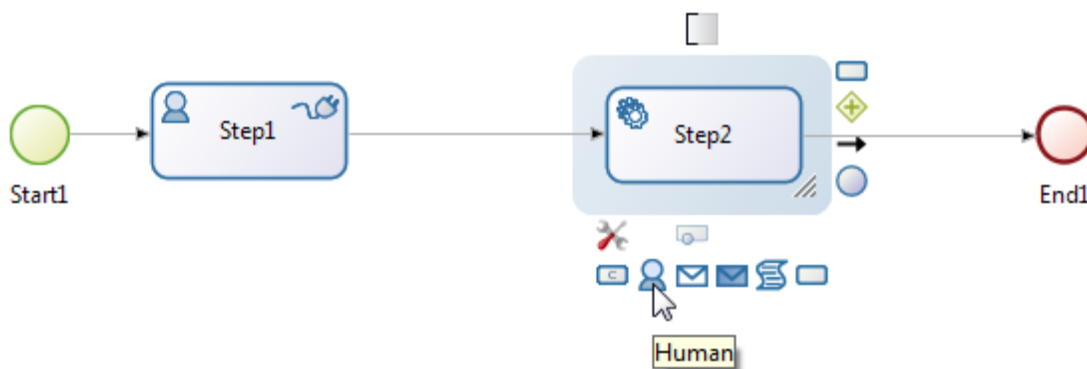


7. Name the variable *Customer_Info*, clear the **Auto-generate form** check box, then click **Finish** to close the dialog box.
8. Click **Finish** to close connector configuration wizard.

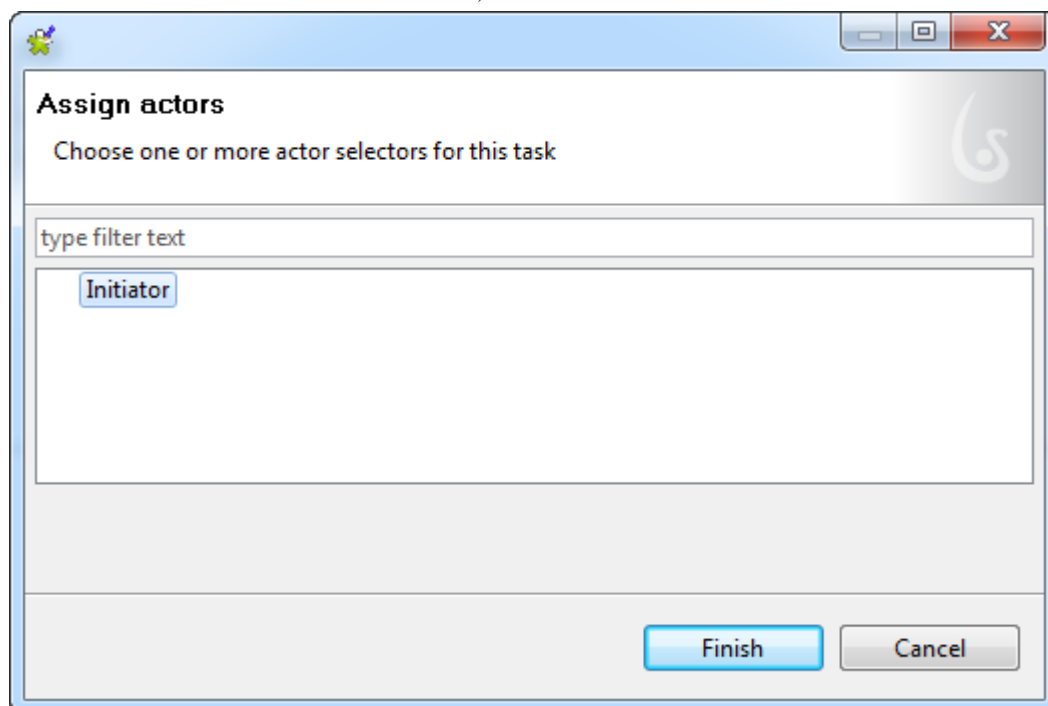
Next, you have to display the *Customer_Info* variable in a process Web User Interface to show the returned information.

Configuring response display (Step two)

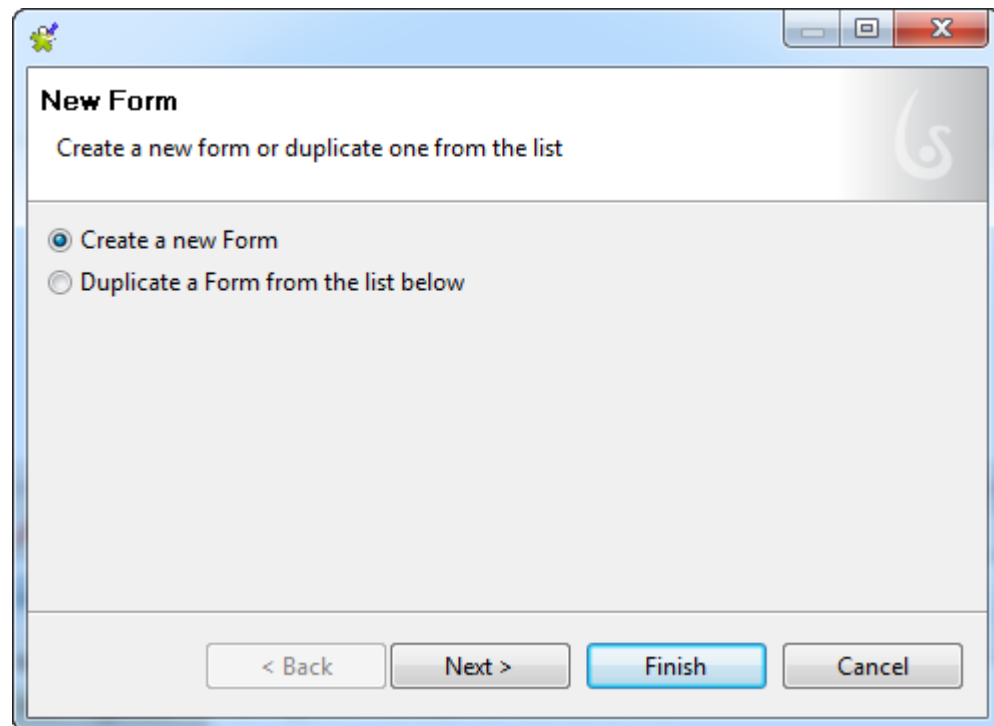
1. Add a new task (Step 2) and an end event to the process model.
2. Click the **Step2** icon in the process model, then click the tools icon on the bottom-left corner.



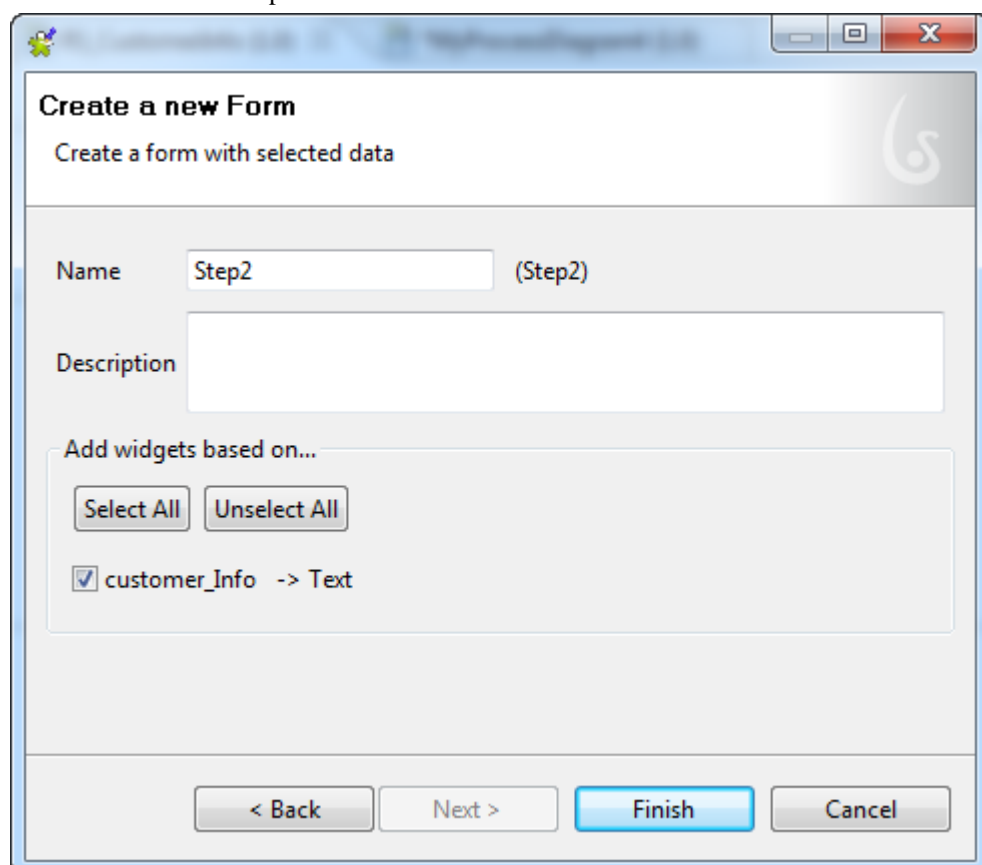
3. Select the **Human** icon. The task type is changed to *Human*.
4. From the **General** view for this new human task, select the **Actors** tab and click the **Choose...** button.



5. Select **Initiator** and click **Finish** to close the window.
6. From the **Application** view, click the **Entry Pageflow** tab and click the **Add...** button next to the first table.



7. Select the **Create a new form** option and click **Next**.

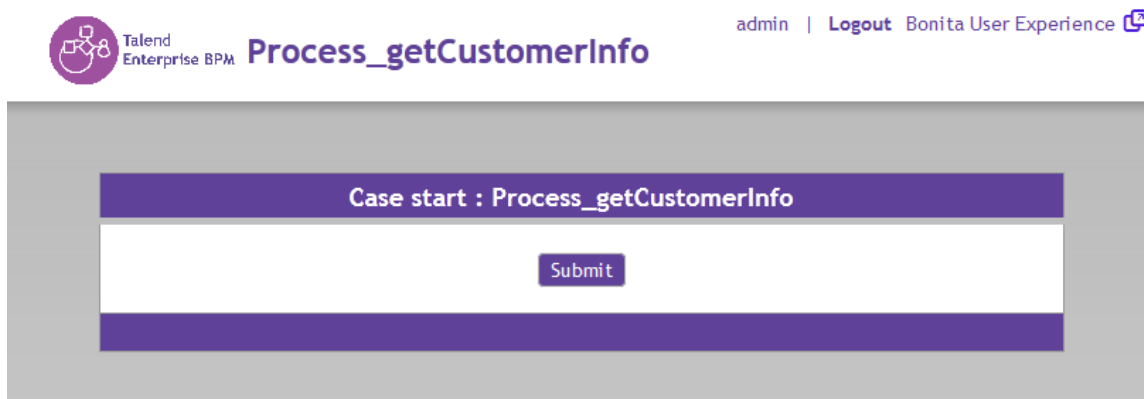


8. On the next step, configure which variables will be shown on the form. Select the *customer_Info* check box and click **Finish**. A newly created form displays the *customer_Info* variable.

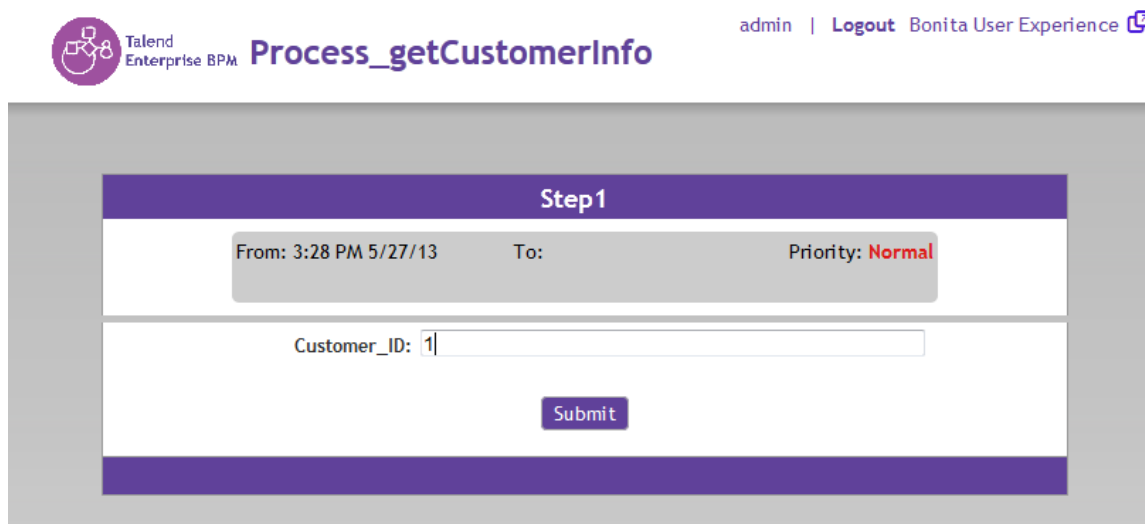
A screenshot of a web form titled "Customer_Info". The form is enclosed in a dashed border with green plus signs at the corners. It contains a text input field with the letter "I" and a button labeled "Submit1".

Running the process and retrieve customer information from the ESB REST service

1. Click the **Run process** button to run the process. The studio opens the process page in your default browser.



2. Click the **Submit** button to start the process and call the REST service.



Talend Enterprise BPM | admin | Logout Bonita User Experience

Process_getCustomerInfo

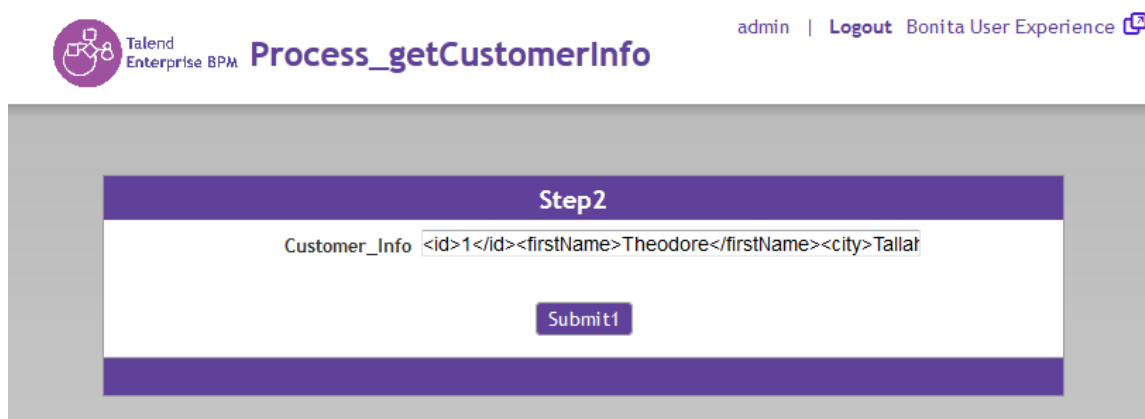
Step1

From: 3:28 PM 5/27/13 To: Priority: **Normal**

Customer_ID:

Submit

3. In the **Customer_ID** field, enter a customer ID, *1* for example, and click the **Submit** button. The HTTP GET operation is performed and the customer information corresponding to the customer ID you just entered is displayed in the **Customer_Info** field.



Talend Enterprise BPM | admin | Logout Bonita User Experience

Process_getCustomerInfo

Step2

Customer_Info

Submit1

Scenario 2: Updating user information to a remote database from within a process using Talend ESB REST Client connector

This scenario explains how to configure the **Talend ESB REST Client** connector to update a record of user information to a remote database using the HTTP POST method.

Prerequisites:

- In the **Integration** perspective of your *Talend Studio*, you have created a Job as described in the scenario on how to implement a REST service accepting HTTP POST requests in the *Talend Components Reference Guide*.
- The above-mentioned Job is running as a REST server to expose a REST service that accepts HTTP POST requests, and you can see the following service definition information when entering `http://localhost:8045/users?_wadl` in your Web browser:

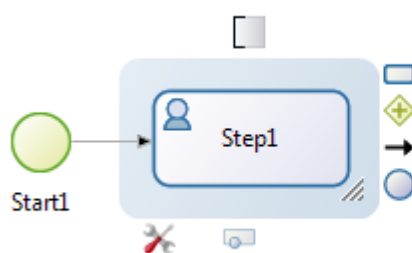
```

- <application>
  <grammars/>
  - <resources base="http://localhost:8045/users">
    - <resource path="/">
      - <resource path="post/{id}/{first_name}/{last_name}">
        <param name="last_name" style="template" type="xs:string"/>
        <param name="id" style="template" type="xs:int"/>
        <param name="first_name" style="template" type="xs:string"/>
      - <method name="POST">
        <request/>
        - <response>
          <representation mediaType="application/xml"/>
          <representation mediaType="text/xml"/>
          <representation mediaType="application/json"/>
        </response>
      </method>
    </resource>
  </resources>
</application>

```

Creating a new process and adding variables for the input data (Step one)

1. Create a new process, rename the process pool to *Update User Info*, and select the **Step1** element in the process model.



2. From the **General** view of the **Details** panel, select the **Data** tab and click the **Add...** button to create an input variable.

Add a new variable
Define the variable to add

Name *
(id)

Description

☐ Auto-generate form

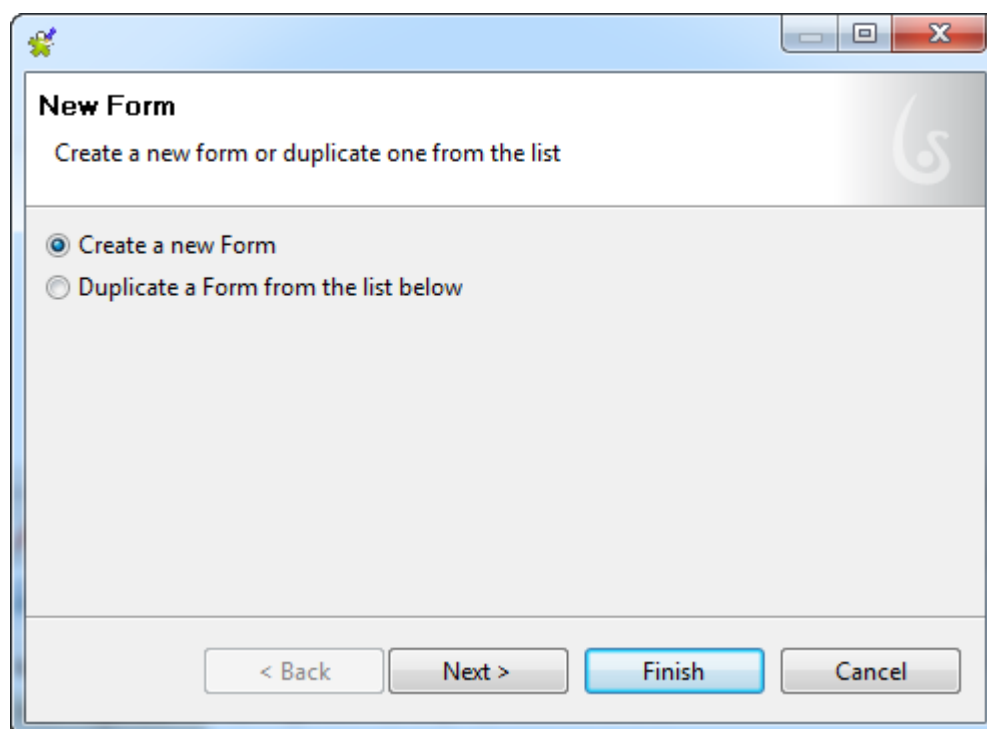
Data type

Multiplicity

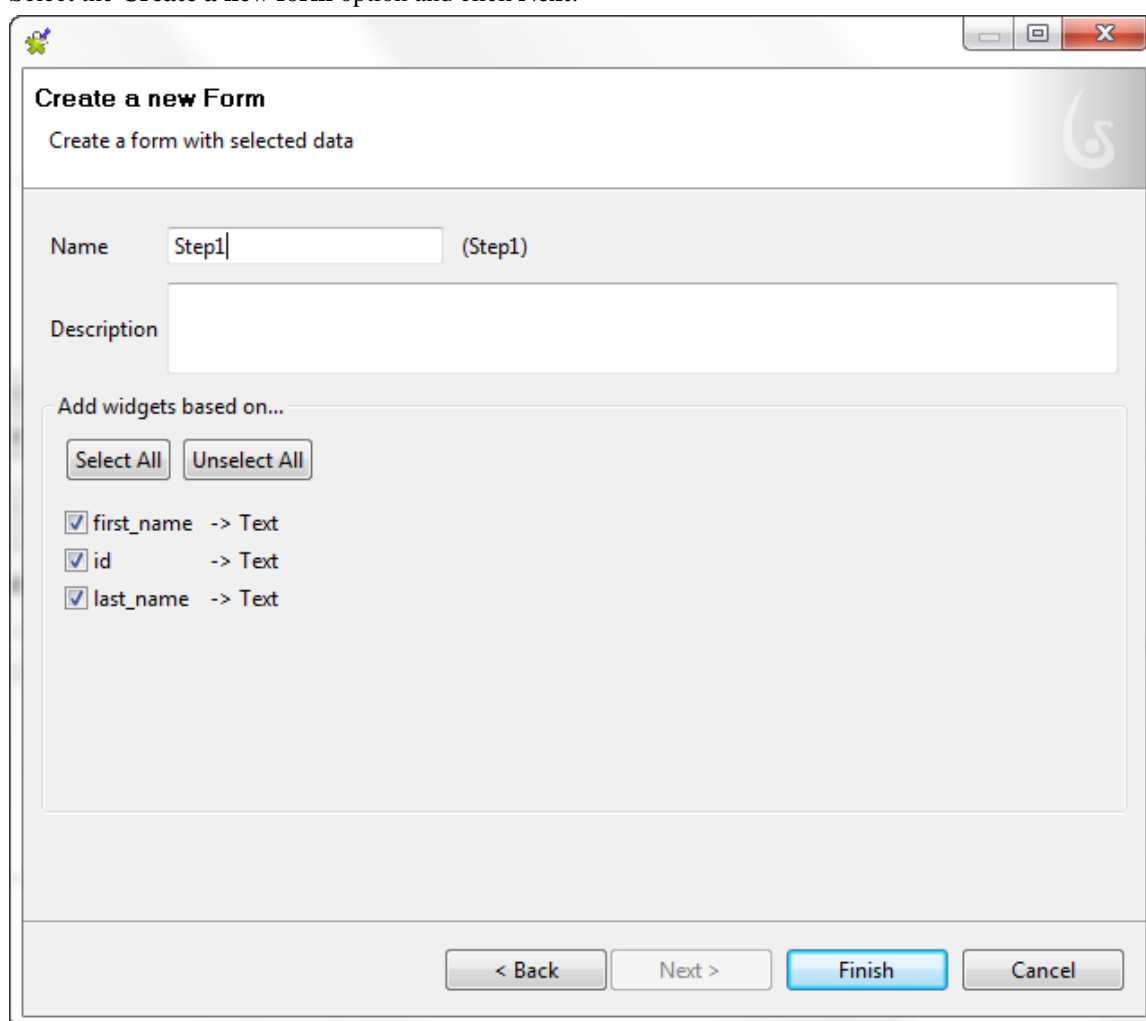
Transient ☐

Default value
Expecting: java.lang.String

3. Name the variable *id*, clear the **Auto-generate form** check box because we will create a form manually to show the input variables. If needed, you can specify the default value in the **Default value** field. When done, click **Finish**.
4. Repeat the steps above to create two other input variables named *first_name* and *last_name*, with the **Auto-generate form** check box cleared.
5. From the **Application** view, click the **Entry Pageflow** tab and click the **Add...** button next to the first table.



6. Select the **Create a new form** option and click **Next**.

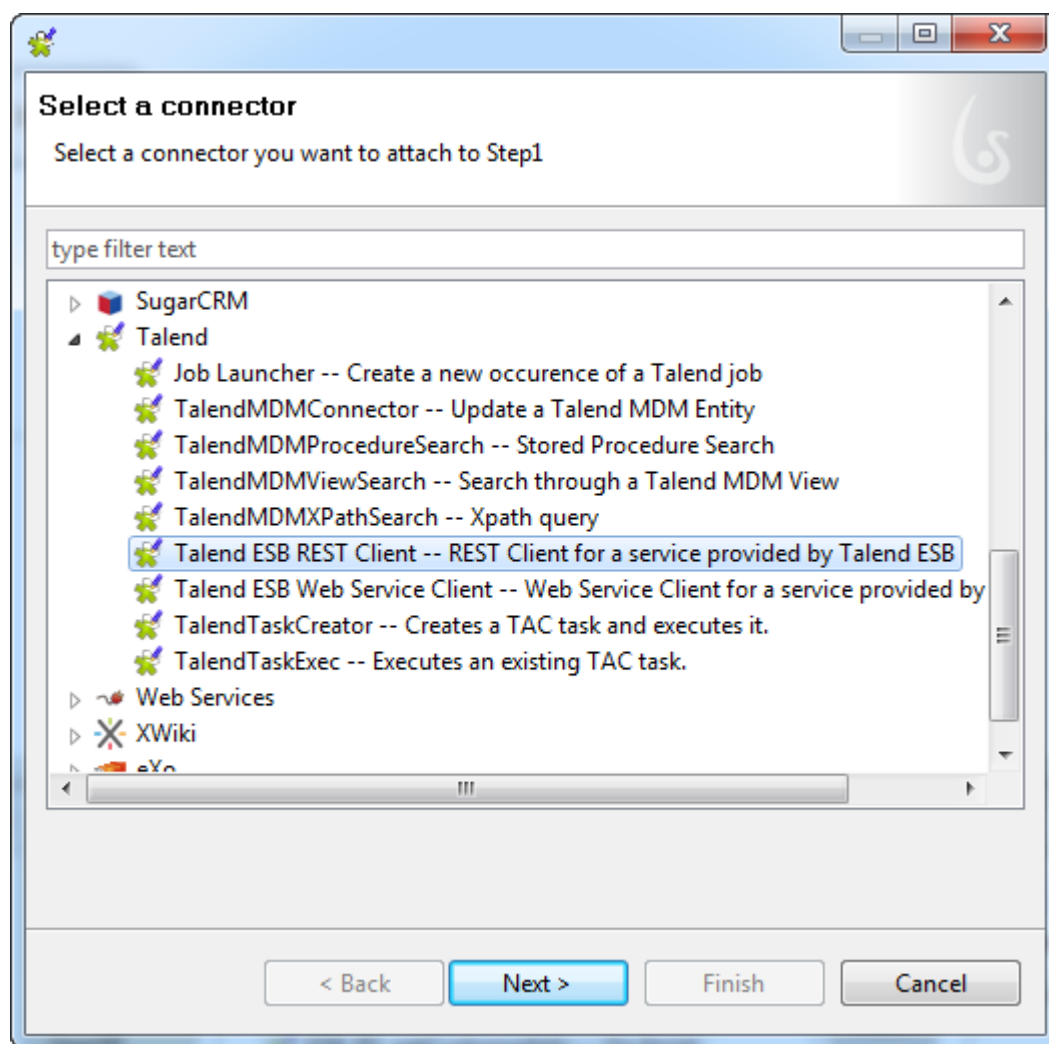


7. On the next step, configure which variables will be shown on the form. Select the *id*, *first_name* and *last_name* check boxes and click **Finish**. A newly created form displays the selected variables. Edit the form so that the variables are displayed in the way you want.

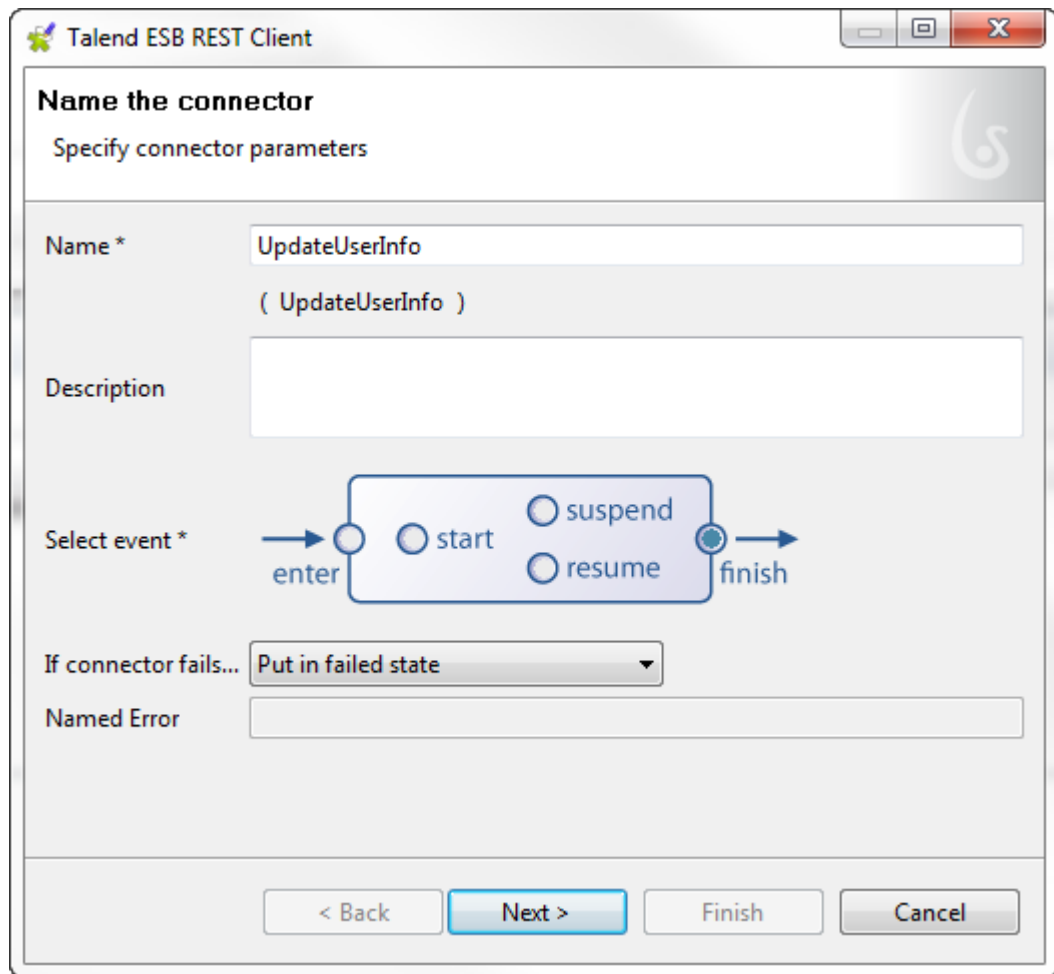
The image shows a web form with a dashed border. On the left side of the form, there are four green plus signs and four red minus signs, each corresponding to a row. The first row contains the text 'User ID' followed by a text input field. The second row contains the text 'First Name' followed by a text input field. The third row contains the text 'Last Name' followed by a text input field. The fourth row contains a 'Submit' button. The form is currently empty, with no data entered in the input fields.

Adding and configuring the Talend ESB REST Client connector

1. From the **General** view, select the **Connectors** tab and click the **Add...** button. A window opens.



2. Select **Talend > Talend ESB REST Client** then click **Next** to open the connector configuration wizard.



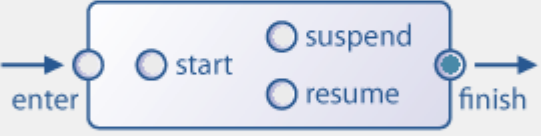
The image shows the 'Name the connector' dialog box in Talend ESB REST Client. The window title is 'Talend ESB REST Client'. The main heading is 'Name the connector' with a subtitle 'Specify connector parameters'. The 'Name *' field contains 'UpdateUserInfo' and '(UpdateUserInfo)' below it. The 'Description' field is empty. The 'Select event *' section shows a state machine diagram with 'enter', 'start', 'suspend', 'resume', and 'finish' events. The 'If connector fails...' dropdown is set to 'Put in failed state'. The 'Named Error' field is empty. At the bottom are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

Talend ESB REST Client

Name the connector
Specify connector parameters

Name *
(UpdateUserInfo)

Description

Select event * 

If connector fails...

Named Error

< Back Next > Finish Cancel

3. Give the connector a name, *UpdateUserInfo* for example, and click **Next**.

The REST Service configuration page opens.

Talend ESB REST Client

Configure your REST Service
Set all necessary information to have your REST Service working

Endpoint URL (not used if SL enabled)

Relative Path

Method *

Accept Type

► Query

▼ Request

Content Type

Request

► Security

► Headers

► Timeouts

Save connector configuration...

Test Configuration

< Back Next > Finish Cancel

4. Configure the REST Service.

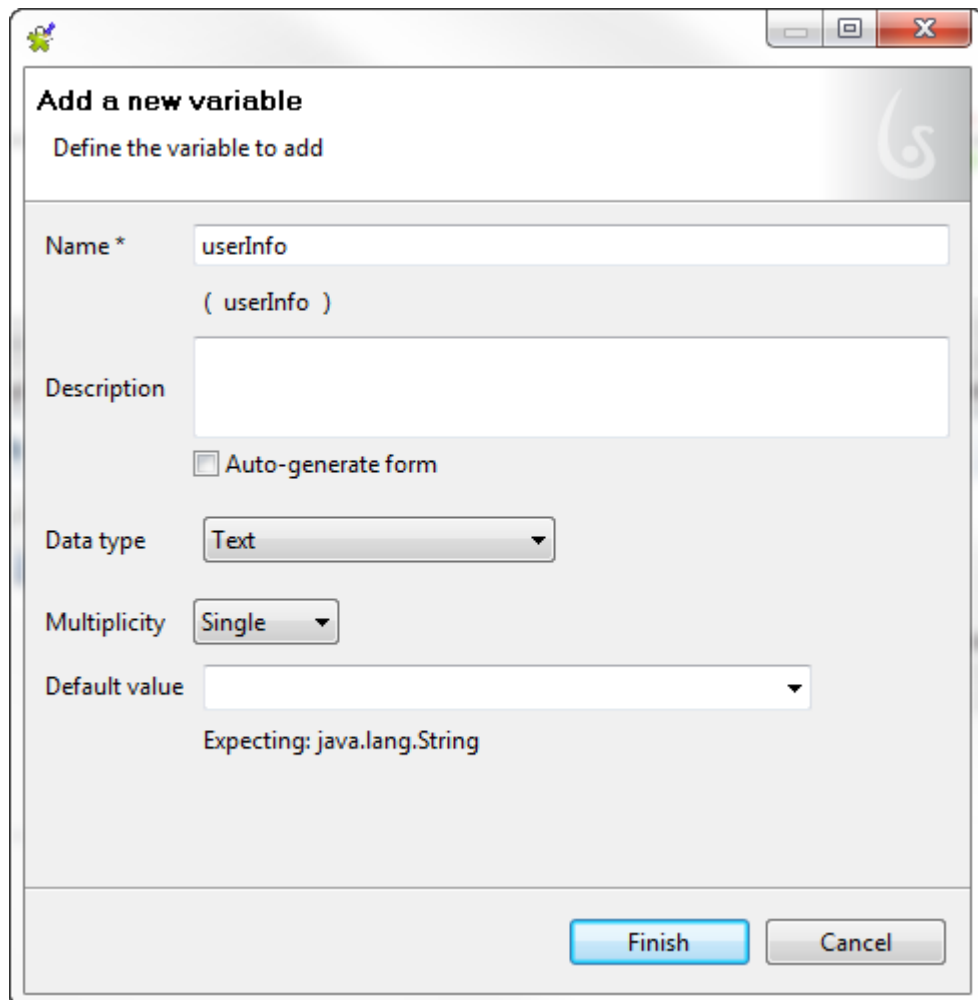
- Fill the **Endpoint URL** field with the URL of the REST Service implement by the Job mentioned above, `http://localhost:8045/users` in this example.
- Fill in the **Relative Path** field with `/post/${id}/${first_name}/${last_name}`. This allows you to enter the information you want to update to the server end via the resource path when the process is executed.
- From the **Method** list, select **POST**, to update the specified information to the remote server through the called REST service.
- From the **Accept Type** list, select **ANY**, to transform any type of response message into a string.
- Expand the **Request** node, select **XML** from the **Content Type** list, and fill the **Request** field with the request payload structure:

```
<id></id><first_name></first_name><last_name></last_name>
```

- Leave all the other parameters as they are.

5. Click **Next**. The **Configure TESB Feature** page opens to let you configure service locator information, service activity monitoring, and business correlation. As we do not use any of these options in this example, simply click **Next** to proceed with connector output mapping.
6. Click the **Create data** button to open the [Add a new variable] dialog box.

Name the variable *userInfo*, clear the **Auto-generate form** check box, then click **Finish** to close the dialog box.



Add a new variable
Define the variable to add

Name *
(userInfo)

Description

☐ Auto-generate form

Data type

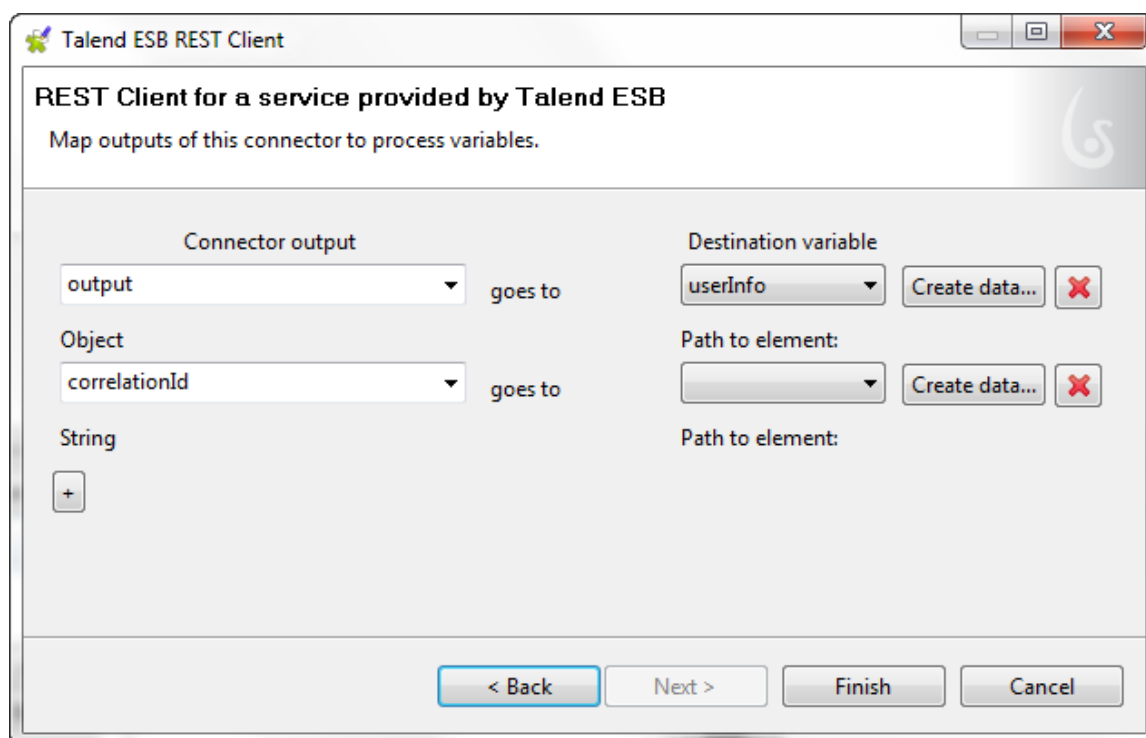
Multiplicity

Default value

Expecting: java.lang.String

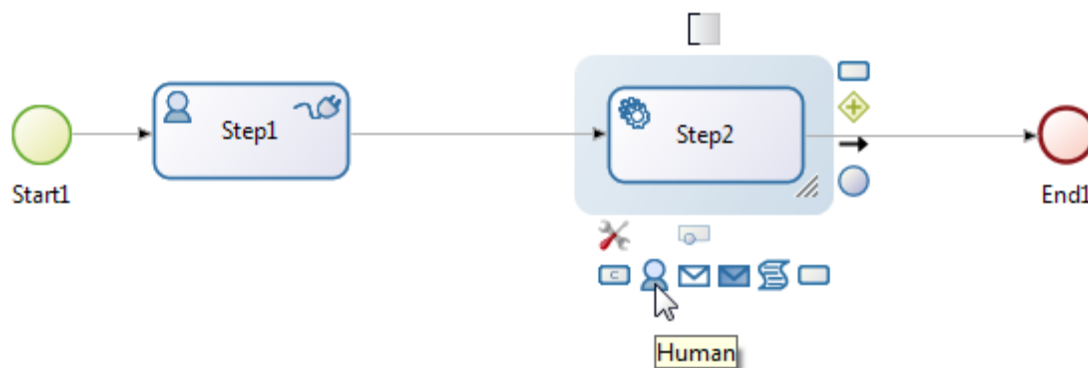
Finish **Cancel**

7. Remove the correlation ID output mapping if you want, which is not required in this example, and click **Finish** to close connector configuration wizard.

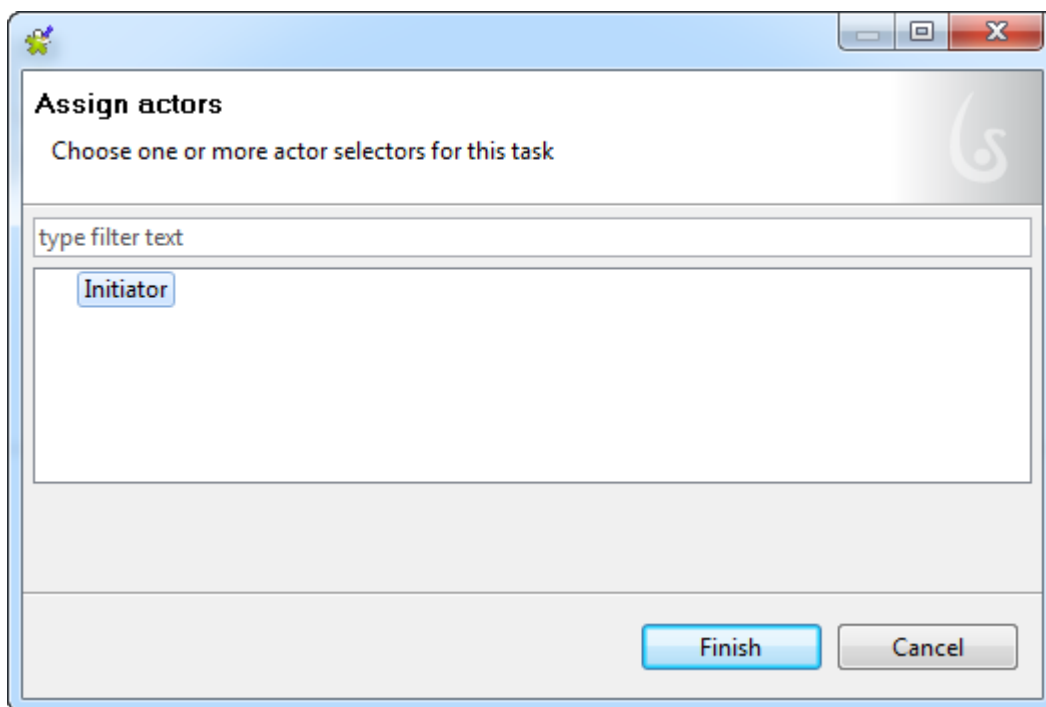


Configuring response display

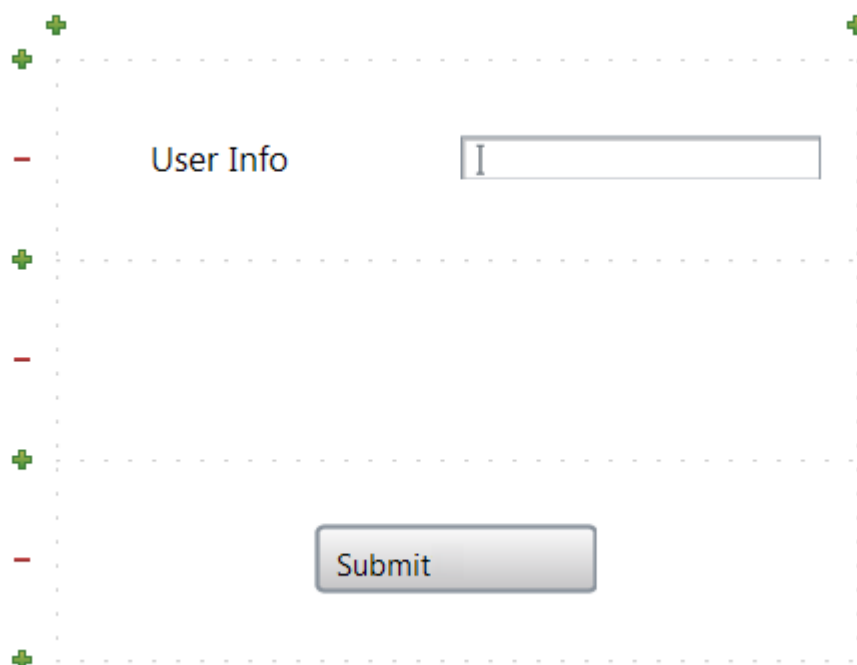
1. Add a new task (Step 2) and an end event to the process model.
2. Click the **Step2** icon in the process model, then click the tools icon on the bottom-left corner.



3. Select the **Human** icon. The task type is changed to *Human*.
4. From the **General** view for this new human task, select the **Actors** tab and click the **Choose...** button.

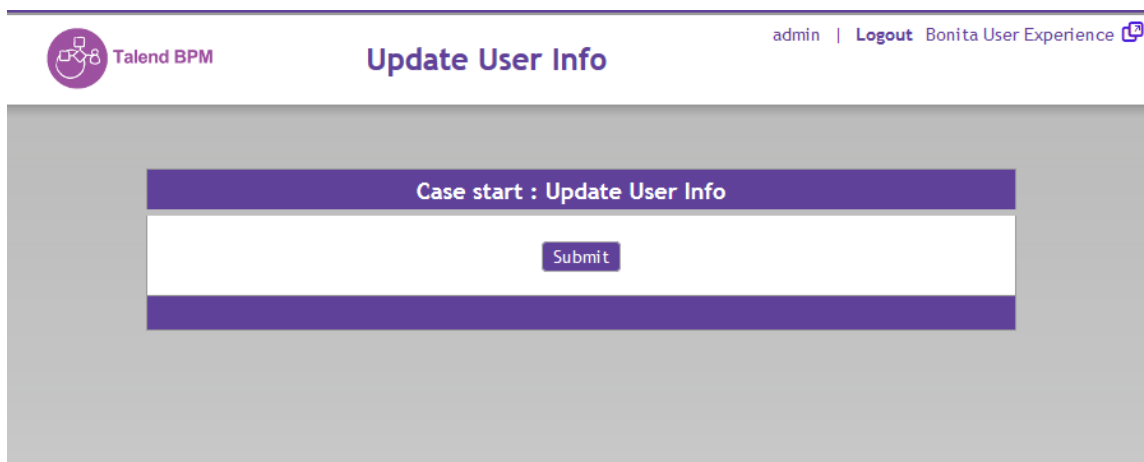


5. Select **Initiator** and click **Finish** to close the window.
6. From the **Application** view, click the **Entry Pageflow** tab, click the **Add...** button next to the first table, and select the **Create a new form** option and click **Next** to create a form for this step.
7. In the **[Create a new Form]** dialog box, select the *userInfo* variable and click **Finish**. A newly created form displays the *userInfo* variable. Edit the form as needed.

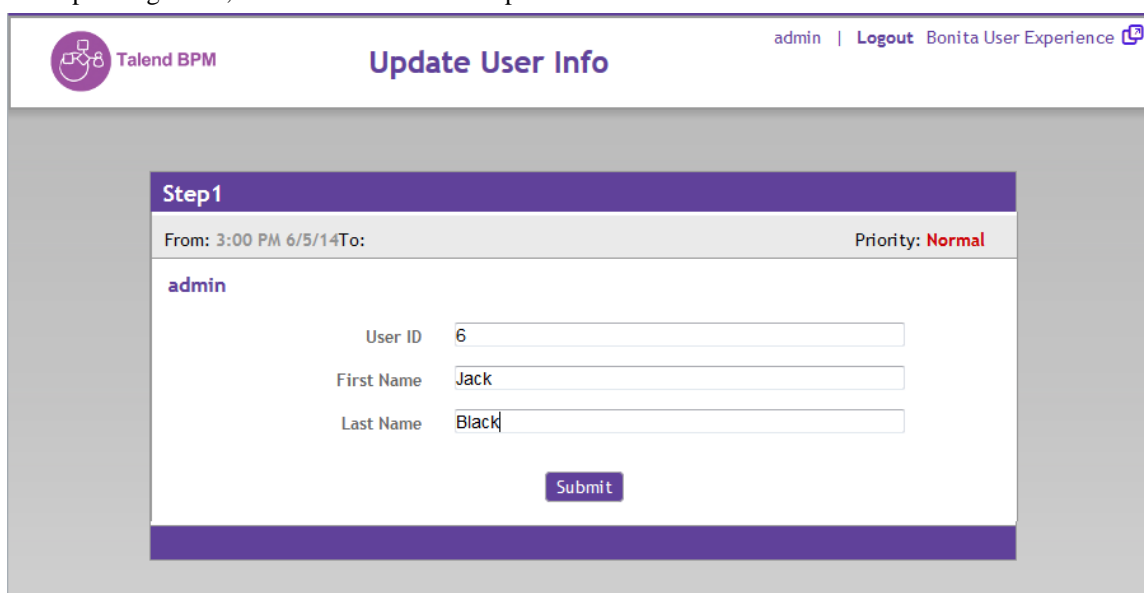
A screenshot of a form titled 'User Info'. The form is enclosed in a dashed border with green plus signs at the corners and red minus signs on the left side, indicating it is a draggable component. Inside the form, there is a text input field with a cursor. At the bottom center of the form is a 'Submit' button.

Running the process and updating user information to the server

1. Click the **Run process** button on the toolbar to run the process. The **Update User Info** page opens in your default browser.



2. Click the **Submit** button to call the REST service. Enter the user ID, first name, and last name in the corresponding fields, and click **Submit** to be updated the information to the server end.



The HTTP POST operation is performed and the response message corresponding to the information you just entered is displayed in the **User Info** field.

Step2

From: 3:01 PM 6/5/14 To: Priority: **Normal**

admin

User Info: `<id>6</id><first_name>Jack</first_name><last_name>Black`

Submit

- Switch to the **Integration** perspective and check the **Run** console of the Job running as the REST server. The server end information shows that the HTTP POST request sent from process has been successfully handled and the user information has been updated to the database.

Execution

Run **Kill** **Clear**

Payload:
`<id></id><first_name></first_name><last_name></last_name>`

tLogRow_1		
id	first_name	last_name
1	John	Smith
2	Martin	Reagan
3	James	White
4	Jenny	Jackson
5	Robert	Thomson
6	Jack	Black

[INFO]: org.apache.cxf.interceptor.LoggingOutInterceptor - Outbound Message

ID: 1
 Response-Code: 200
 Content-Type: application/xml
 Headers: {Content-Type=[application/xml], Date=[Thu, 05 Jun 2014 02:37:34 GMT]}

Payload: `<?xml version="1.0" encoding="UTF-8"?><user><id>6</id><first_name>Jack</first_name><last_name>Black</last_name></user>`

☐ Line limit ☒ Wrap

