



Exercises

Table of Contents

Preface	3
1. Exercise: Modeling a basic process	4
2. Exercise: Adding data and specifying contracts	7
3. Exercise: Dynamically controlling the process flow with data	13
4. Exercise: Creating forms	17
5. Exercise: Configuring actors	25
6. Exercise: Using a connector to send an email	27

Preface

The goal of the exercises is to create a basic leave request process.

The process will execute in the following way:

1. An employee requests a leave
2. The employee's manager validates the request

In order to implement this process we are going to follow an iterative approach mixing theoretical content and practical exercises. Each exercise will allow you to build an executable and more complete version of the process.

Each exercise is divided in 3 sections:

- **Goal** - Presents the objectives of the exercise.
- **Instructions overview** - Contains an overview of the main steps required for completing the exercise.
- **Step by step instructions** - Contains a detailed description of the steps required for completing the exercise.

1. Exercise: Modeling a basic process

Goal

The goal of this exercise is to create a first basic version the leave request process diagram.

At this stage the process is executable but has very limited value from the business point of view as it does not yet contain forms or data. We will extend it in the upcoming exercises.

Instructions overview

In order to complete the exercise, create a "Leave request diagram" diagram in version 1.0 containing a "Leave request" process in version 1.0.

The process pool should contain the following BPMN elements:

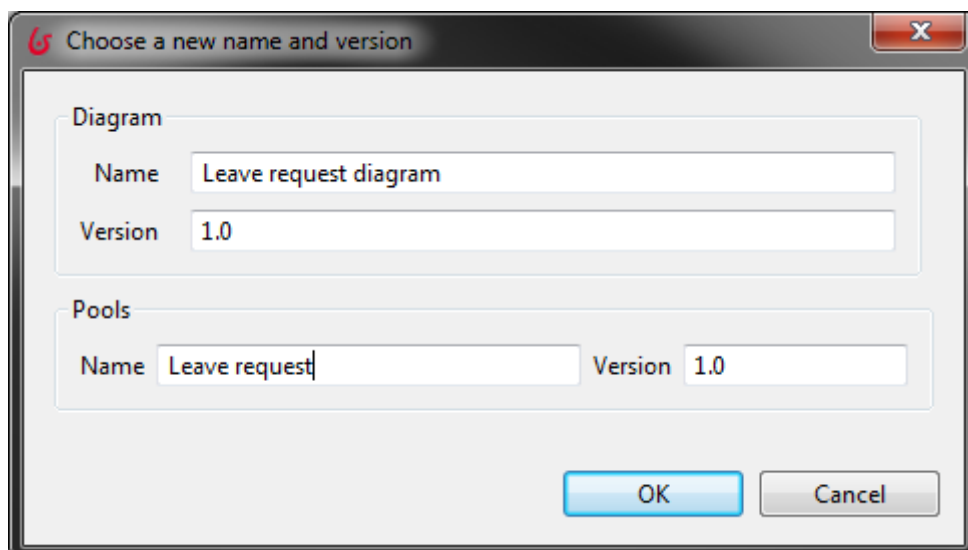
- A start event named "Fill request"
- A human task named "Validate request"
- A terminate end event named "End - Request processed"

Step by step instructions

1. **Start the Bonita Studio.**
2. **Create a new process diagram.**
3. **Set the diagram and process names.**

Click on the "Diagram / Rename" top menu.

Enter "Leave request diagram" as the diagram name and "Leave request" as the pool name.



4. **Rename the start event into "Fill request".**

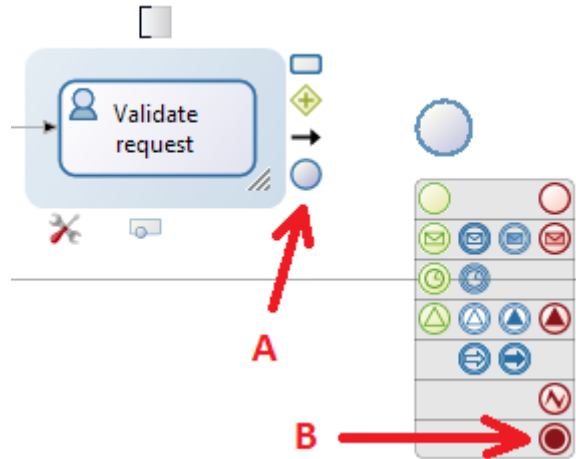
Select the start event from the diagram.

Navigate to the "General / General" tab and enter the new name.

5. **Rename the human task into "Validate request".**

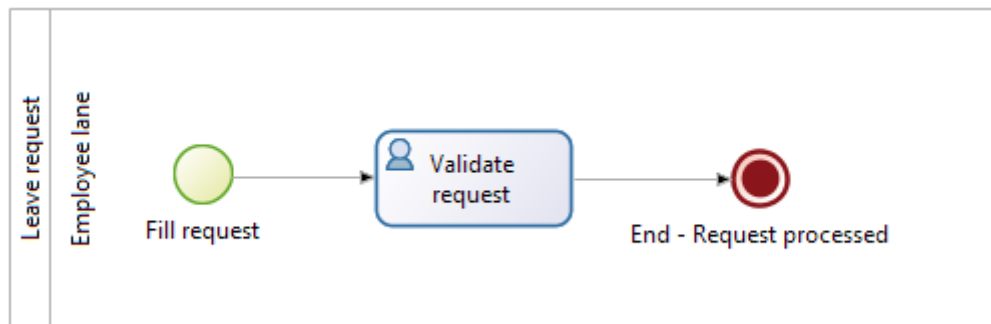
6. **Add a terminate end event named "End - Request processed".**

Select the human task, perform drag and drop from the contextual event icon (A) and select the terminate end event (B).



Once the event is created, rename it.

7. **Make sure that your diagram matches this:**




8. **Save the process.**

Click on the "Save" button from the Studio's top menu bar.

9. **Run the process to test it.**



Click on the  button of the Studio's top menu bar.

Your browser should open and display an automatically generated case start form. Submit the form.

You should be redirected to the task inbox of the Bonita Portal. Select the "Validate request" task and click on the top right "Do it" button.

The screenshot displays the Bonitasoft web application interface. At the top, the Bonitasoft logo is on the left, and the user's name 'Walter Bates' and a 'Settings' link are on the right. Below the header, there are three tabs: 'Tasks' (selected), 'Cases', and 'Processes'. On the left sidebar, there are sections for 'To do' (with a refresh button), 'My tasks', 'Available tasks', 'Done', and 'Processes' (with a 'Leave request 1.0' task). The main content area shows a search bar, 'TAKE' and 'RELEASE' buttons, and a table with one task: 'Validate request' (1 - Leave request) with a due date of 'in 59 min'. To the right of the task table, there is a 'Validate request' section with a 'DO IT' button (highlighted with a red box) and a 'MORE' button. Below this, there is a 'Comments' section with a text input field and an 'ADD COMMENT' button.

Welcome: Walter Bates | User | Settings

Tasks Cases Processes

Search...

TAKE RELEASE

Validate request

No description.

Case: 1
Due date: in 59 min
Process: Leave request
Priority: Normal

Comments

No data

Type new comment

+ ADD COMMENT

Submit the form of the "Validate request" task to complete your case.

2. Exercise: Adding data and specifying contracts

Goal

The goal of this exercise is to continue the leave request process implementation by:

1. setting up a Business Data Model (BDM)
2. specifying contracts for the case instantiation and human tasks

Once completed, the process will be executable with automatically generated forms.



Warning

It is mandatory to strictly observe the case and syntax of the technical names provided in the instructions. Failing to do so will result in errors.

Instructions overview

Duplicate the process diagram from the previous exercise to create a 2.0 version.

Create a "LeaveRequest" BDM with the following attributes:

Name	Type	Mandatory
requestorId	Long	No
leaveStart	Date	Yes
dayCount	Integer	Yes
isApproved	Boolean	No

Declare an instance "request" of type "LeaveRequest" on your pool.

Generate case instantiation contract from the "request" data and with the following mandatory items:

- leaveStart
- dayCount

Add these two constraints on the case instantiation contract:

- "leaveStart" must be in the future
- "dayCount" must be strictly greater than zero

Initialize the "request" BDM with the following code:

```
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import com.company.model.LeaveRequest;

// Retrieve current process instance
ProcessAPI processAPI = apiAccessor.getProcessAPI();
ProcessInstance processInstance = processAPI.getProcessInstance(processInstanceId);

// Initiate new leave request
LeaveRequest newRequest = new LeaveRequest();
newRequest.setRequestorId(processInstance.getStartedBy());
```

```
newRequest.setLeaveStart(requestInput.get("leaveStart"));
newRequest.setDayCount(requestInput.get("dayCount"));
return newRequest;
```

Add a step contract on the "Validate request" task with the following item:

- isApproved (Boolean)

Add an operation on the "Validate request" task to set the "isApproved" field of the "request" BDM with the value returned by the contract.

Step by step instructions

1. Duplicate the process diagram from the previous exercise to create a 2.0 version.

In the Studio's top menu, click on "Diagram / Duplicate".

Update the process diagram AND pool version numbers.

2. Create the "LeaveRequest" BDM

Navigate to the "Development / Business Data Model / Manage" top menu.

Click on "Add" (A) in the "List of Business Objects" and name the object "LeaveRequest" (B) (this is a technical name so it should not contain spaces or special characters).

With the "LeaveRequest" object selected, add the following attributes (C):

Name	Type	Mandatory
requestorId	Long	No
leaveStart	Date	Yes
dayCount	Integer	Yes
isApproved	Boolean	No

Manage Business Data Model
Add Business Objects to the list and edit their details in the right pane tabs.

List of Business Objects

A **Add** button is circled in red. The **LeaveRequest** object is listed with a red 'B' next to it.

LeaveRequest

Description: [Text Area]

Attributes

C **Add** button is circled in red. The attributes table is shown below:

Name	Type	Multiple	Mandatory
requestorId	LONG	<input type="checkbox"/>	<input type="checkbox"/>
leaveStart	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>
dayCount	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>
isApproved	BOOLEAN	<input type="checkbox"/>	<input type="checkbox"/>

Details for isApproved
No details available

Package: com.company.model

Buttons: Finish, Cancel

Click on "Finish".

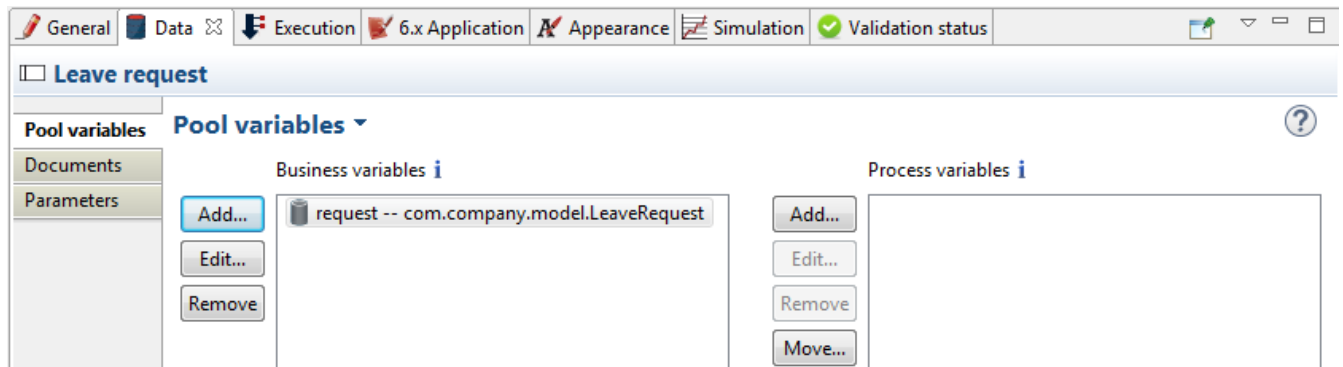
3. Declare an instance of the "LeaveRequest" BDM on your process

Select the process pool, and navigate to the "Data / Pool variables" tab.

In the "Business variables" section, click on "Add".

Name the variable "request" and select the "LeaveRequest" business object.

Click on "Finish".



4. Set up the case instantiation contract

Select the process pool, and navigate to the "Execution / Contract / Inputs" tab.

Click on "Add from data".

Select the "request" BDM and click on "Next".


Uncheck the "requestorId" and "isApproved" attributes.

Add Contract input

Select request attributes to add to contract
Choose the fields you want to reuse in your contract.

Input name: of type COMPLEX

Attribute name	Attribute type	Input type
<input type="checkbox"/> requestorId	LONG	TEXT
<input checked="" type="checkbox"/> leaveStart	DATE	DATE
<input checked="" type="checkbox"/> dayCount	INTEGER	INTEGER
<input type="checkbox"/> isApproved	BOOLEAN	BOOLEAN

 Contract elements and mapping operations will be generated and can be edited manually.
 Storage operations to update business variables can also be generated.
 Select the option that suits your needs:

☒ Auto-generate storage operations
☐ No, thanks. I'll manually define how to use the contract.

Click on "Finish" and dismiss the warning message about the incomplete initialization of the business variable.

☒ General
 ☒ Data
 ☒ Execution
 ☒ 6.x Application
 ☒ Appearance
 ☒ Simulation
 ☒ Validation status

☐ Leave request

Connectors in

Contract

Instantiation form

Overview page

Connectors out

Process instantiation inputs ▾

Inputs Constraints

Add from data...

Add

Add child

Remove

Name *	Type	Multiple	Description
requestInput	COMPLEX	<input type="checkbox"/>	
leaveStart	DATE	<input type="checkbox"/>	
dayCount	INTEGER	<input type="checkbox"/>	

Still in the "Execution / Contract" tab, switch to the "Constraints" tab.

Add the following constraint:

Property	Value
Name	leaveStartIsFuture
Expression	<code>return requestInput.leaveStart.compareTo(new Date()) > 0;</code>
Error message	Leave start must be in the future

Add a second constraint:

Property	Value
Name	atLeastOneDay
Expression	<code>return requestInput.dayCount > 0;</code>
Error message	Day count must be greater than zero

Name *	Expression *	Error message
leaveStartIsFuture	<code>return requestInput.leaveStart.compareTo(new Date()) > 0;</code>	Leave start must be in the future
atLeastOneDay	<code>return requestInput.dayCount > 0;</code>	Day count must be greater than zero

5. Update the "request" BDM initialization

Select the process pool, and navigate to the "Data / Pool variables" tab.

Select the "request" business variable and click on "Edit".

Click on the "Pencil" icon next to the "Default value" field to open the expression editor.

Clear all of the generated code and replace it by the following:

```
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import com.company.model.LeaveRequest;

// Retrieve current process instance
ProcessAPI processAPI = apiAccessor.getProcessAPI();
ProcessInstance processInstance = processAPI.getProcessInstance(processInstanceId);

// Initiate new leave request
LeaveRequest newRequest = new LeaveRequest();
newRequest.setRequestorId(processInstance.getStartedBy());
newRequest.setLeaveStart(requestInput.get("leaveStart"));
newRequest.setDayCount(requestInput.get("dayCount"));
return newRequest;
```

This will initialize the BDM from the contract data and set the process initiator as the request author.

6. Set up the "Validate request" step contract

Select the "Validate request" task and navigate to the "Execution / Contract / Inputs" tab.

Click on "Add" then set the input name to "isApproved" and its type to "BOOLEAN".

7. Add an operation on "Validate request" to update the request

With the "Validate request" task selected, navigate to the "Execution / Operations" tab.

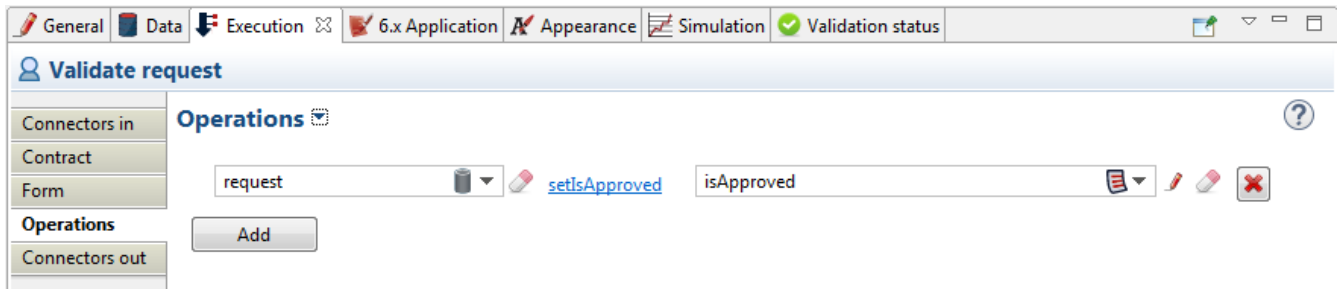
Click on "Add"

Select "request" as the target.

Click on the "Takes value of" link and switch to the "Use a Java method" operator type.

Select the "setIsApproved" method and click on "OK".

In the rightmost dropdown, select the "isApproved" contract input.



8. Save the process.

Click on the "Save" button from the Studio's top menu bar.

9. Run the process to test your contract constraints.

Run the process and use the automatically generated forms to test your constraints.

Tips:

- for the "leaveStart" input, enter a date in the YYYY-MM-DD format.
- for the "isApproved" input, enter a boolean value: either "true" or "false".

Attention : this is a temporary form generated automatically for testing. Before you put your process into production, create and map the necessary forms.

Leave request

requestInput - <no description defined in contract for this input>

leaveStart - <no description defined in contract for this input>

Expecting a DATE value. (2015-05-31)

dayCount - <no description defined in contract for this input>

Expecting a INTEGER value.

Start

3. Exercise: Dynamically controlling the process flow with data

Goal

The goal of this exercise is to extend the existing process by adding a dynamic validation for the leave request. This will be achieved with the use of an exclusive gateway which will lead one of two possible execution paths: one for the approved request and one for the rejected request.

Instructions overview

Duplicate the process diagram from the previous exercise to create a 3.0 version.

Modify the diagram to add the following elements after the "Validate request" task:

- an exclusive gateway named "Is approved?"
- a service task named "Notify request approved"
- a terminate end event named "End - Request rejected"
- a service task named "Notify request rejected"
- a terminate end event named "End - Request approved"

Configure the transitions going out the exclusive gateway so that they control dynamically the process flow based on process data.

Step by step instructions

1. **Duplicate the process diagram from the previous exercise to create a 3.0 version.**
2. **Remove the terminate end event.**

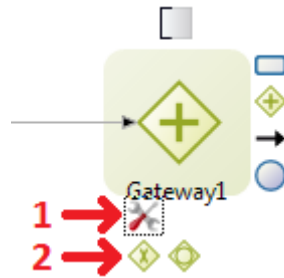
Select the event from the diagram and hit "delete" on your keyboard.

3. **Add an exclusive gateway named "Is approved?" next to the human task "Validate request".**

Select the "Validate request" task, perform a drag and drop from the gateway contextual icon.



Once you have created the gateway, use the context "tool" icon (1) to change its type to "Exclusive" (2).



Rename the gateway into "Is approved?".

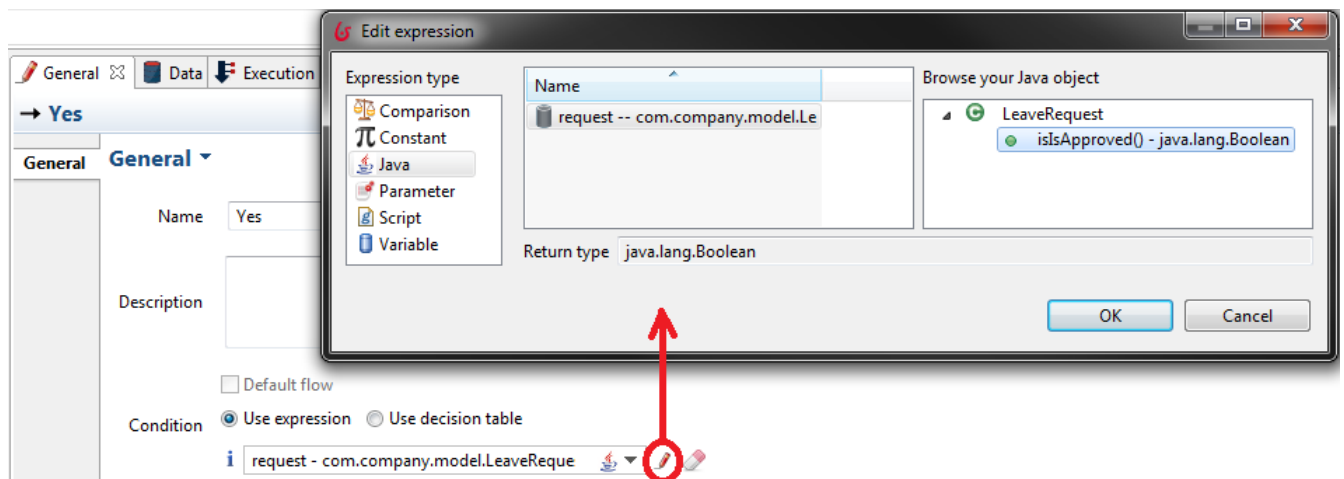
4. Create a service task named "Notify request approved" after the gateway.
5. Add a terminate end event named "End - Request approved" next to the newly added service task.
6. Create a service task named "Notify request rejected" after the gateway.
7. Add a terminate end event named "End - Request rejected" next to the newly added service task.
8. Name the two transitions going out of the gateway

The transition leading to the "Notify request approved" task should be named "Yes" and the other "No".

9. Configure condition on the "Yes" transition to use the "isApproved" attribute of the "request" BDM.

Select the transition and click on the pencil icon next to the "Condition" dropdown to open the expression editor.

In the expression editor, select "Java" as the expression type, select the "request" variable and the "isApproved" method.



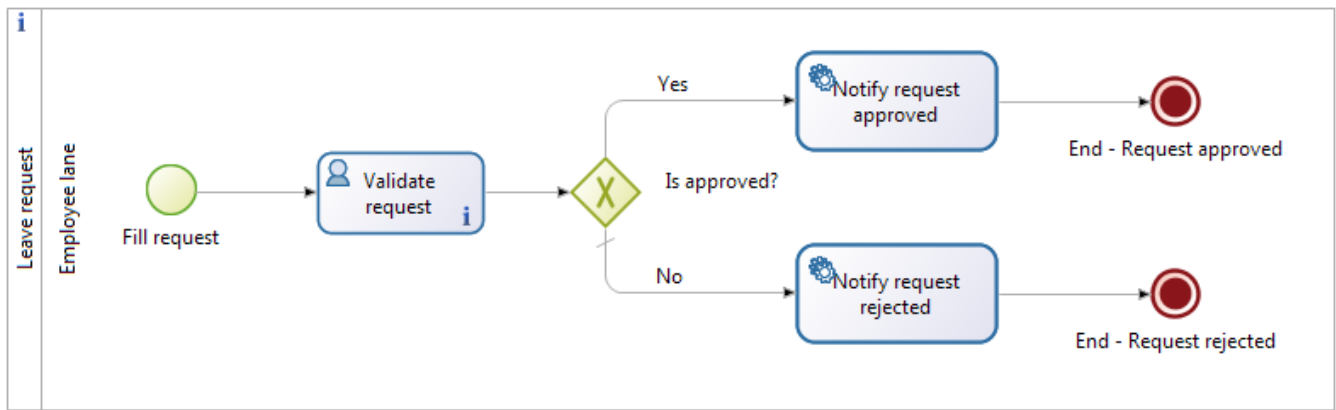
10. Set the "No" transition as the default flow.

Select the "No" transition and check the "Default flow" property.

11. Validate the diagram.

Once the transition conditions are properly set, you can re-validate the process diagram. To do so, navigate to the "Validation status" tab and click on "refresh". You may safely ignore the 3 warnings related to the UI Designer.

12. Make sure that your diagram matches this:



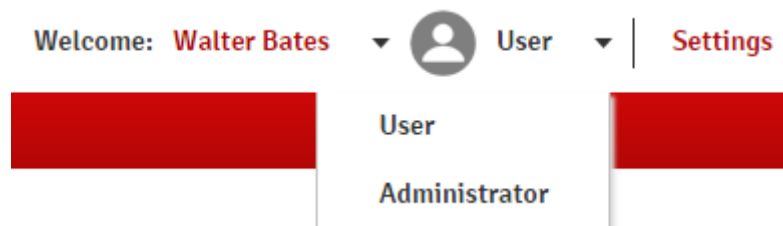
13. Execute twice the process to test the different paths.

Warning: do not hit twice the "Run" button as this would redeploy the process and clear the history. In order to create the second process instance, you will need to navigate in the Bonita Portal.

To do so, navigate to the "Processes" menu and select the "Leave request" process definition with version 3.0 and click on "Start".

14. Check that the process is correctly executed in the Portal administration interface.

In the Portal, switch to the "Administrator" profile with the top right dropdown.



Navigate to the "BPM / Cases" menu (A) then, switch to the "Archived cases" tab (B) and click on the "..." link in the action column (C) to view the case details.

Bonitasoft Welcome: **Walter Bates** Administrator Settings

BPM Organization Resources Applications

Tasks
Cases
Processes

Cases
Open cases
Archived cases

Filters

Process name **All** Process version **All** Search...

Case list **1-2 of 2**

	ID	Process name	Version	Start date	Started by	End date	State	Actions
<input type="checkbox"/>	8	Leave request	3.0	06/05/2015 5:55 PM	walter.bates	06/05/2015 5:55 PM	completed	...
<input type="checkbox"/>	9	Leave request	3.0	06/05/2015 5:57 PM	walter.bates	06/05/2015 5:58 PM	completed	...

Check the "Done tasks" section to validate the process execution:

BACK **OVERVIEW**

Case id: 8 - Process: Leave request

Process version: 3.0 Started on: 06/05/2015 5:55 PM Started by: Walter Bates

Last updated: 06/05/2015 5:55 PM State: completed

Done tasks

- Task name: Notify request approved
Description: No description.
Performed on: 06/05/2015 5:55 PM
- Task name: Is approved?
Description: No description.
Performed on: 06/05/2015 5:55 PM
- Task name: Validate request
Description: No description.
Performed on: 06/05/2015 5:55 PM

1 - 3 of 3

Comments

System 10 min ago

The task "Validate request" is now assigned to walter.bates

1 of 1

4. Exercise: Creating forms

Goal

The goal of this exercise is to provide user friendly forms for the execution of the process.

Instructions overview

Duplicate the process diagram from the previous exercise to create a 4.0 version.

Create the following forms:

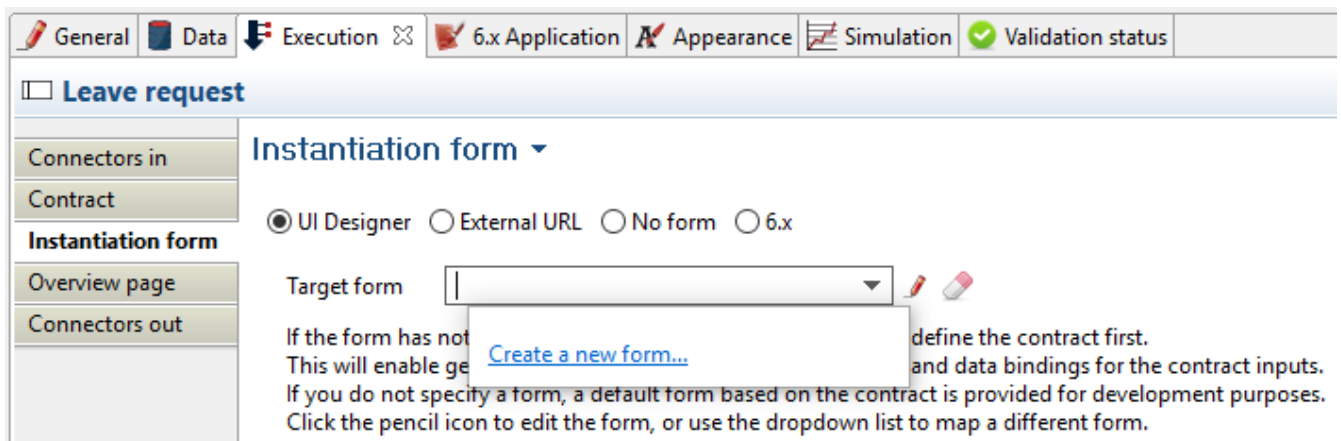
- A case instantiation form at pool level that sets the "leaveStart" and "dayCount" data.
- A form for the "Validate request" task that displays the "requestor" user details, the "leaveStart" and "dayCount" data in read-only mode and that sets the "isApproved" data.

Step by step instructions

1. **Duplicate the process diagram from the previous exercise to create a 4.0 version.**
2. **Create a case instantiation form.**

Select the process pool, and navigate to the "Execution / Instantiation form" tab.

Use the "Target form" dropdown to create a new form.



This will open the UI Designer in a browser with a form automatically generated from your instantiation contract.

3. **Rename the form.**

Use the text field on top of the screen to rename the form into "fillLeaveRequest" (this is a technical name so it should not contain spaces or special characters).

Click on "Save".

4. **Add a form container**

Drag and drop a "Form container" widget from the widget palette located on the left on a new row under the "Request contract" widget.

Tip: A dashed zone allowing to insert a new row appears when hovering the separation between two widgets.



The form container allows to validate the content entered in the form (required fields and field formatting).

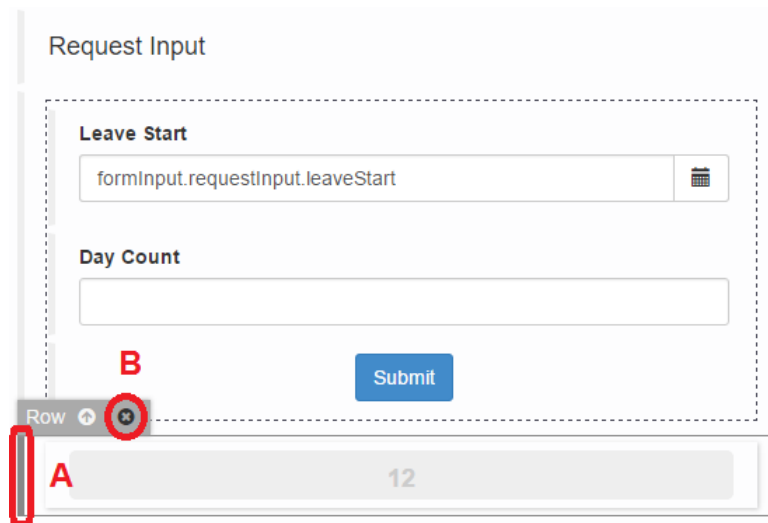
5. Move the existing widgets in the form container

Select the "Leave Start" widget and drag and drop it inside the form container.

Repeat the operation for the "Day Count" and "Submit" widgets while adding new rows.

Once the widgets are grouped under the form container, select and remove the empty rows.

Tip: Hover on the grey bar located on left of a row (A) to display the actions including the removal (B).



6. Update the form title

Select the top most widget "Request Input" and edit the following properties:

Property	Value
Text	Fill a New Leave Request
Text level	Level 2
Alignment	Center

7. Update the "Leave start" widget

Select the "Leave start" widget and edit the following properties:

Property	Value
Required	yes
Placeholder	Start date of your leave
Date format	yyyy-MM-dd

8. Update the "Day count" widget

Select the "Day count" widget and edit the following properties:

Property	Value
Required	yes
Label	Number of days
Placeholder	Number of leave days
Min value *	1

* not to be confused with the "Value min length" property.

9. Clear the initial form values (this will show the placeholders at runtime)

In the lower "Variables" section, click on the "Pencil" icon for the "formInput" variable.

Replace the JSON values with this:

```
{
  "requestInput" : {
    "leaveStart" : null,
    "dayCount" : null
  }
}
```


10. Add a variable to handle potential form submit errors

In the lower "Variables" section, click on "Create a new variable".

Set the variable name to "error", leave its type to "String" and its value empty then click on "Save".

11. Update the "Submit" widget

Select the "Submit" widget.

Click on the "binding" icon  next to the "Disabled" property.

Enter the following content in the text field that just appeared (this will disable the button when the form cannot be validated):


```
$form.$invalid
```

Set the "Failed response value" property to "error" (this allows to retrieve submission error details if any).

12. Dynamically display the error

Drag a "Text" widget from the palette and drop it under the "Submit" button.

Set the "CSS classes" property to "text-danger".

Click on the "binding" icon  next to the "Hidden" property.

Enter the following content in the text field that just appeared (this displays the widget only when there is an error):

```
error == null
```

Replace the "Text" property with the following content:

```
<b>Error :</b> {{error | json}}
```

13. Check the form's appearance

Make sure that the form looks like this in the editor:

The screenshot shows a form editor interface. At the top, the title 'Fill a New Leave Request' is centered. Below it, a dashed box contains the form structure. The first field is 'Leave Start *' with a text input containing 'formInput.requestInput.leaveStart' and a calendar icon. The second field is 'Number of days *' with a text input containing 'Number of leave days'. A blue 'Submit' button is positioned below the second field. At the bottom left, there is an error message placeholder: 'Error : {{error | json}}'.

Click on the top most "Preview" button.

Make sure that the form preview looks like this:

The screenshot shows the form preview. The title 'Fill a New Leave Request' is centered at the top. Below it, the 'Leave Start *' field has a text input with the placeholder 'Start date of your leave' and a calendar icon. The 'Number of days *' field has a text input with the placeholder 'Number of leave days'. A blue 'Submit' button is at the bottom center.

Check for following points:

- the widget presenting the error is not visible
- the "Submit" button is disabled by default (this is due to the validation provided by the form container)
- the "Submit" button is enabled when the form is valid

Note: the form cannot be submitted from the preview mode even when it is valid.

14. Save the form

Use the top "Save" button to save the form.

15. Create a form for the "Validate request" task.

In the Studio, select the "Validate request" task and navigate to the "Execution / Form" tab.

Use the "Target form" drop down to create a new form.

16. Rename the form

In the UI Designer, rename the form into "validateLeaveRequest" and save it.

17. Remove unneeded variables

In the lower "variables" section of the screen, remove the following variables:

- formInput
- formOutput

18. Retrieve business data from the ongoing request

Click on the "Create a new variable" button and configure the variable with the following properties:

Property	Value
Name	request
Type	External API
API URL	../{{context.request_ref.link}}

Note: here, we are calling the Bonita REST APIs to fetch our "request" BDM data. We are using the "context" variable that exposes links to the process instance data including our request in the form of "request_ref". We can then retrieve the request thanks to the "link" attribute which provides the URL that needs to be called to retrieve the object.

19. Retrieve the requestor's user data

Create a new variable with the following properties:

Property	Value
Name	requestor
Type	External API
API URL	../API/identity/user/{{request.requestorId}}

20. Add a variable for error handling

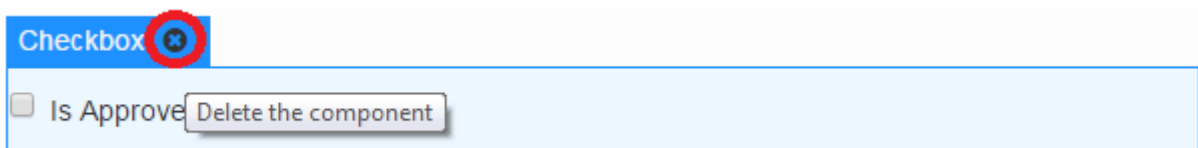
Create a new variable named "error", leave its type to "String" and its value empty then save it.

21. Remove the "formInput" et "formOutput" variables

Use the "trash bin" icon to remove the "formInput" and "formOutput" variables.

22. Remove the "Is Approved" checkbox

Select the "Is Approved" checkbox and use the removal link that appears when hovering over the widget with the cursor.



23. Remove the "Submit" button

24. Add a form title

Drag a "Title" widget from the palette and drop it on the top row of the form.

Configure the widget as following:

Property	Value
Text	Validate a leave request
Title level	Level 2
Alignment	center

25. Add a widget to display the requestor

Drag an "Input" widget from the palette and drop it on a new row under the form title.

Configure the widget as following:

Property	Value
Read-only	yes
Label	Requestor
Value	requestor.firstname + " " + requestor.lastname

26. Add a widget to display the leave start date

Drag an "Input" widget from the palette and drop it on a new row under the requestor widget.

Configure the widget as following:

Property	Value
Read-only	yes
Label	Leave start
Value	request.leaveStart date

Note: we are using an AngularJS filter "| date" to display the date in a user-friendly format.

27. Add a widget to display the number of days

Drag an "Input" widget from the palette and drop it on a new row under the "Start date" widget.

Configure the widget as following:

Property	Value
Read-only	yes
Label	Number of days
Value	request.dayCount

28. Add a widget to reject the request

Drag a "Button" widget from the palette and drop it on a new row at the bottom of the form.

Configure the widget as following:

Property	Value
Width	6
Label	Reject

Property	Value
Alignment	right
Style	danger
Data sent on click	{ "isApproved" : false }
Failed response value	error
Target URL on success	/bonita

29. Add a widget to approve the request

Drag a "Button" widget from the palette and drop it on the "6 column" zone located on the right of the "Reject" button.


Configure the widget as following:

Property	Value
Label	Approve
Style	success
Data sent on click	{ "isApproved" : true }
Failed response value	error
Target URL on success	/bonita

30. Dynamically display potential submit errors

Drag a "Text" widget from the palette and drop it under the "Submit" button.

Set the "CSS classes" property to "text-danger".

Click on the "binding" icon  next to the "Hidden" property.

Enter the following content in the text field that just appeared (this display the widget only when there is an error):

```
error == null
```

Replace the "Text" property with the following content:

```
<b>Error :</b> {{error | json}}
```

31. Check the form's appearance

Make sure that the form looks like this in the editor:

Validate a leave request

Requestor
requestor.firstname + " " + requestor.lastname

Leave start
request.leaveStart | date

Number of days
request.dayCount

Reject

Approve

Error : {{error | json}}

32. Save the form

Use the top "Save" button to save the form.

33. Test the process execution

Run the process from the Studio and execute all of its steps.

Validate that the proper execution path was taken at the end of process execution by looking at the case history in the Bonita Portal.

5. Exercise: Configuring actors

Goal

The goal of this exercise is to add collaboration to the existing process by dispatching the forms between two actors: a requestor and a validator.

Instructions overview

Duplicate the process diagram from the previous exercise to create a 5.0 version.

Add a "Validator" lane to the diagram and move the "Validate request" task in it.

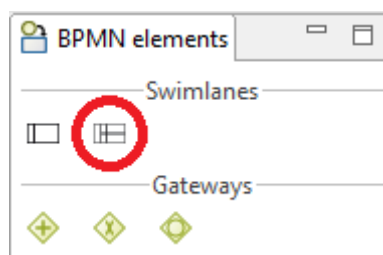
Add an actor filter of type "Initiator manager" on the "Validator" lane.

Step by step instructions

1. **Duplicate the process diagram from the previous exercise to create a 5.0 version.**

2. **Add a "Validator" lane to the process.**

Select the "Lane" element from the BPMN palette and click in the process pool.



With the lane selected, navigate to the "General / Lane" tab and set its name to "Validator".

3. **Rename the other lane into "Requestor".**

4. **Drag and drop the "Validate request" task into the "Validator" lane.**

5. **Add an "Initiator manager" actor filter on the "Validator" lane.**

Select the "Validator" lane and navigate to the "General / Actor" tab.

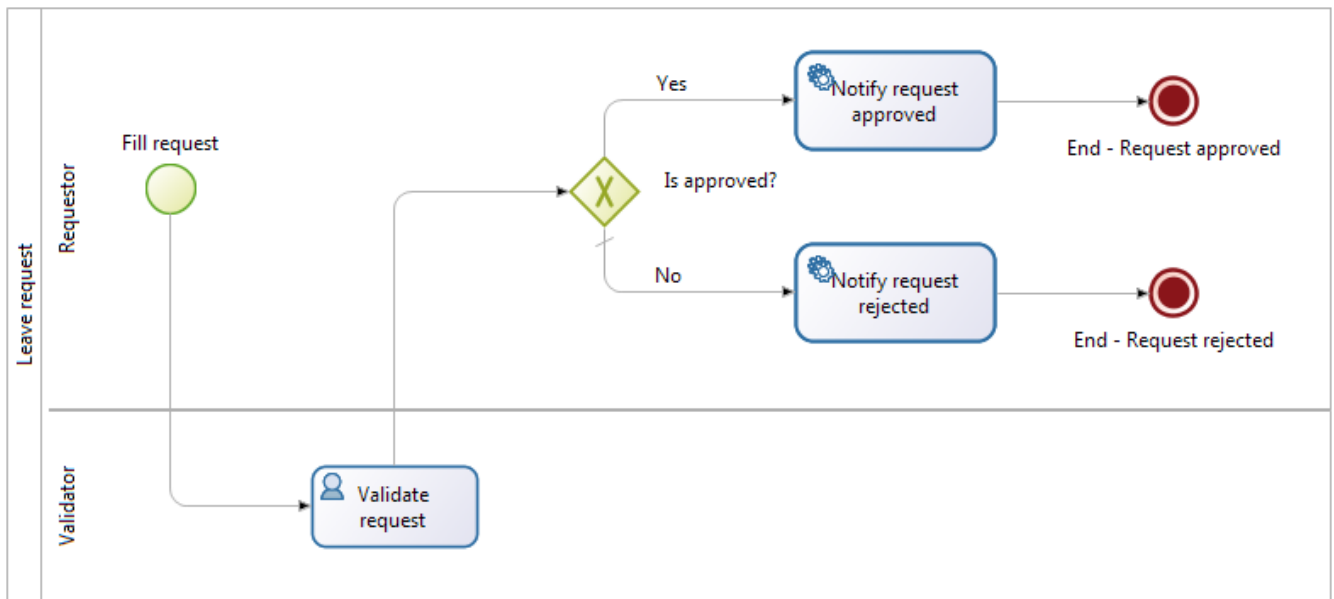
Select the "Employee actor" from the dropdown menu.

Click on the "Set" button of the actor filter.

Select an "Initiator manager" filter and click on "Next".

Name the filter "requestorManager" and click on "Finish".

6. **Make sure that the diagram looks like this:**



7. Execute the process with the two actors.

Run the process from the Studio (user "Walter Bates" will be used).

Submit the "Fill leave request" form. If the actors are properly set, the "Validate request" task should not be available anymore.

Disconnect from the Portal by clicking on the user name in upper right corner of the screen and clicking on "Logout".



Connect as user "helen.kelly" with "bpm" as password.

If the actor filter ran correctly, the "Validate request" tasks should now be available in the inbox.

6. Exercise: Using a connector to send an email

Goal

The goal of this exercise is to demonstrate an interaction between Bonita BPM and an external system with the use of a connector. In this context, we will notify the leave request initiator about his request's status with an email connector.



Warning

Depending on your network configuration, your firewall or the security settings of your email server you may not be able to send an email from Bonita BPM.

Instructions overview

Duplicate the process diagram from the previous exercise to create a 6.0 version.

Add an email connector on the "Notify request approved" and "Notify request rejected" service tasks. These will send an email to the requestor with the request's validation status.

The following code will be used to retrieve the requestor's email in the connector:

```
import org.bonitasoft.engine.api.IdentityAPI;
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import org.bonitasoft.engine.identity.ContactData;

// Get APIs
ProcessAPI processAPI = apiAccessor.getProcessAPI();
IdentityAPI identityAPI = apiAccessor.getIdentityAPI();
// Get process initiator
ProcessInstance processInstance = processAPI.getProcessInstance(processInstanceId);
long initiatorUserId = processInstance.getStartedBy();
// Get email from process initiator
ContactData contactData = identityAPI.getUserContactData(initiatorUserId, false);
return contactData.getEmail();
```

Step by step instructions

1. **Duplicate the process diagram from the previous exercise to create a 6.0 version.**
2. **Test the email connector to obtain the right SMTP configuration.**

Navigate to the "Development / Connectors / Test connector" top menu.

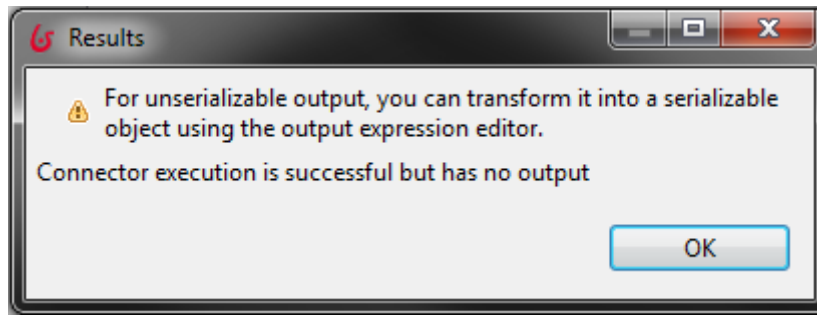
Select the "Email (SMTP)" connector by using either the text filter or the "Messaging" category and click on "Next".

Fill the connection parameters based on your email account settings and click on "Next".

Enter your email in the "From" and "To" fields and click on "Next".


Enter "Bonita test" as the subject, click on the "Test" button and confirm the test without checking any dependency.

At this stage, you should get a message similar to this one:



Make sure that you receive the email (this may take a couple of minutes). If you do not receive it, update the connector configuration and re-test it.



Once the configuration is valid, click on .

Name the configuration "emailConfig" and save it.

Close the connector test interface.

3. Add an email connector on the "Notify request approved" task.

Select the task, navigate to the "Execution / Connectors in" tab and click on "Add".

Select the "Email (SMTP)" connector.

Name it "sendRequestApprovedEmail" and click on "Next".



Do not fill in the parameters but click on .

Select the "emailConfig" and move to the "Email addressee" configuration page.

Enter your email address in the "From" field.

Use the "pencil" icon to edit the expression of the "To" field.

Set the "Expression type" to "Script", name the script "getRequestorEmail" and paste the following in the code edition zone:

```
import org.bonitasoft.engine.api.IdentityAPI;
import org.bonitasoft.engine.api.ProcessAPI;
import org.bonitasoft.engine.bpm.process.ProcessInstance;
import org.bonitasoft.engine.identity.ContactData;

// Get APIs
ProcessAPI processAPI = apiAccessor.getProcessAPI();
IdentityAPI identityAPI = apiAccessor.getIdentityAPI();
// Get process initiator
ProcessInstance processInstance = processAPI.getProcessInstance(processInstanceId);
long initiatorUserId = processInstance.getStartedBy();
// Get email from process initiator
ContactData contactData = identityAPI.getUserContactData(initiatorUserId, false);
return contactData.getEmail();
```

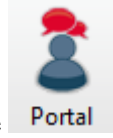
Move to the next page and set "Leave request approved" as the subject.

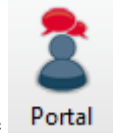
Click on "Finish".

4. Add an email connector on the "Notify request rejected" task.

Repeat the previous steps by naming the connector "sendRequestRejectedEmail" and setting "Leave request rejected" as the subject.

5. Update the email of the test user "Walter Bates".



Open the Portal from the Studio by using the  button from the toolbar.

Switch to the "Administrator" profile then navigate to the "Organization / Users" menu.

Select the user "Walter Bates" and click on the "More" button.

Click on "Edit user".

Select the "Business card" tab

Enter your email in the "Email" field.

Save your changes and close the Portal.

6. Execute the process twice to test the different paths and check that you receive the right emails.