

# Exercise 11

*Creating a BPMN flow*

## **Prior Knowledge**

*Understand simple BPMN*

*Using Eclipse / Maven*

## **Objectives**

Understand the basics of the BPEL specification, and be able to create and execute a business process using the BPEL tooling in Eclipse. Deploy the BPEL into the WSO2 BPS and be able to track instances etc.

## **Software Requirements**

- Java Development Kit 7
- WSO2 Developer Studio 3.7
- Camunda BPMN Modeler 2.7.0
- Camunda Engine 7.3.0 including Tomcat

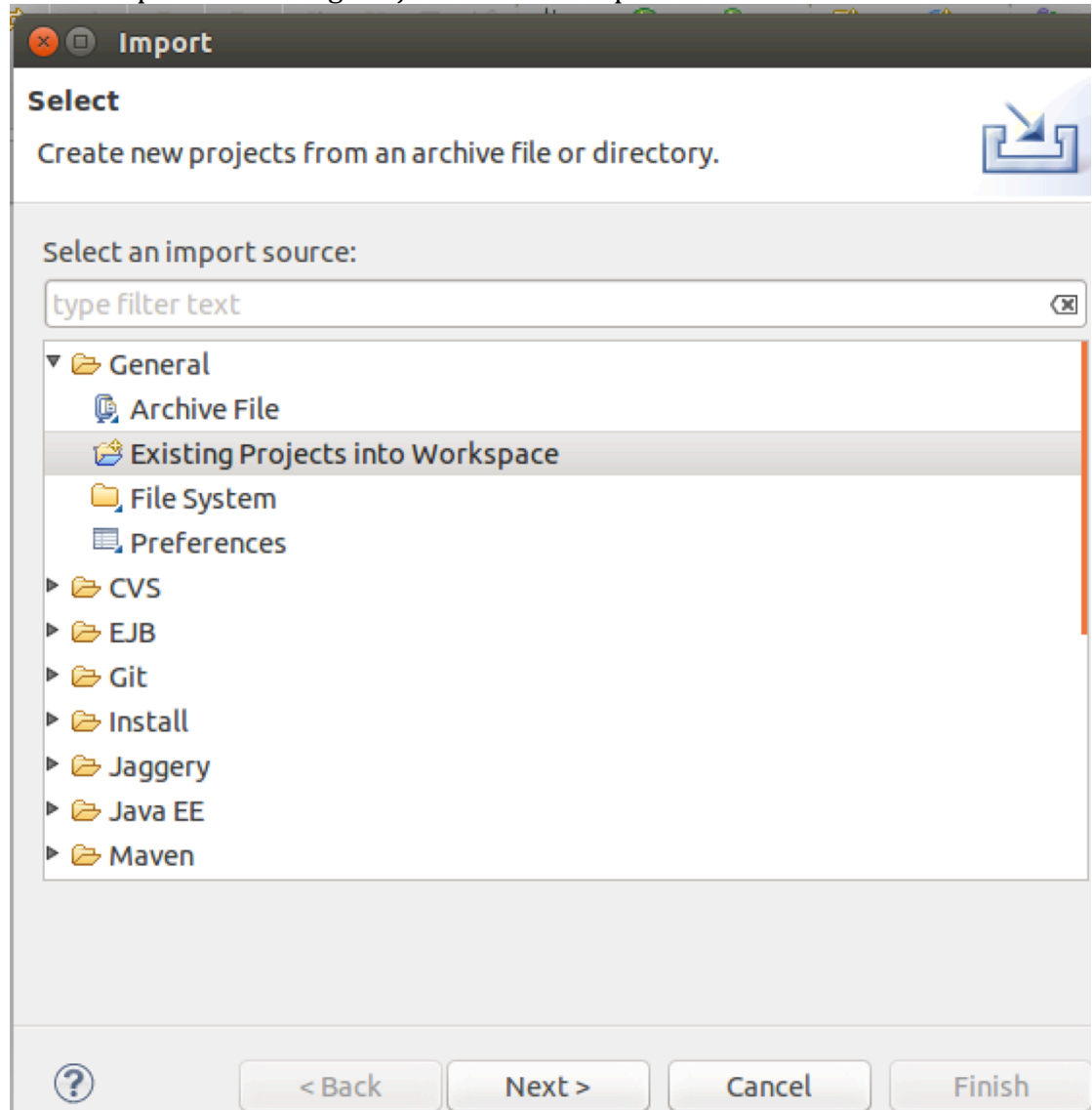
*This exercise is a shortened / modified version of the Camunda BPMN 2.0 Getting Started Guide. I recommend you work through that in your own time, but this exercise is a bit more efficient. I've also modified it to be about Coffee and not Loans ☺. The Camunda guide is here: <http://docs.camunda.org/get-started/bpmn20/>*

## **Steps.**

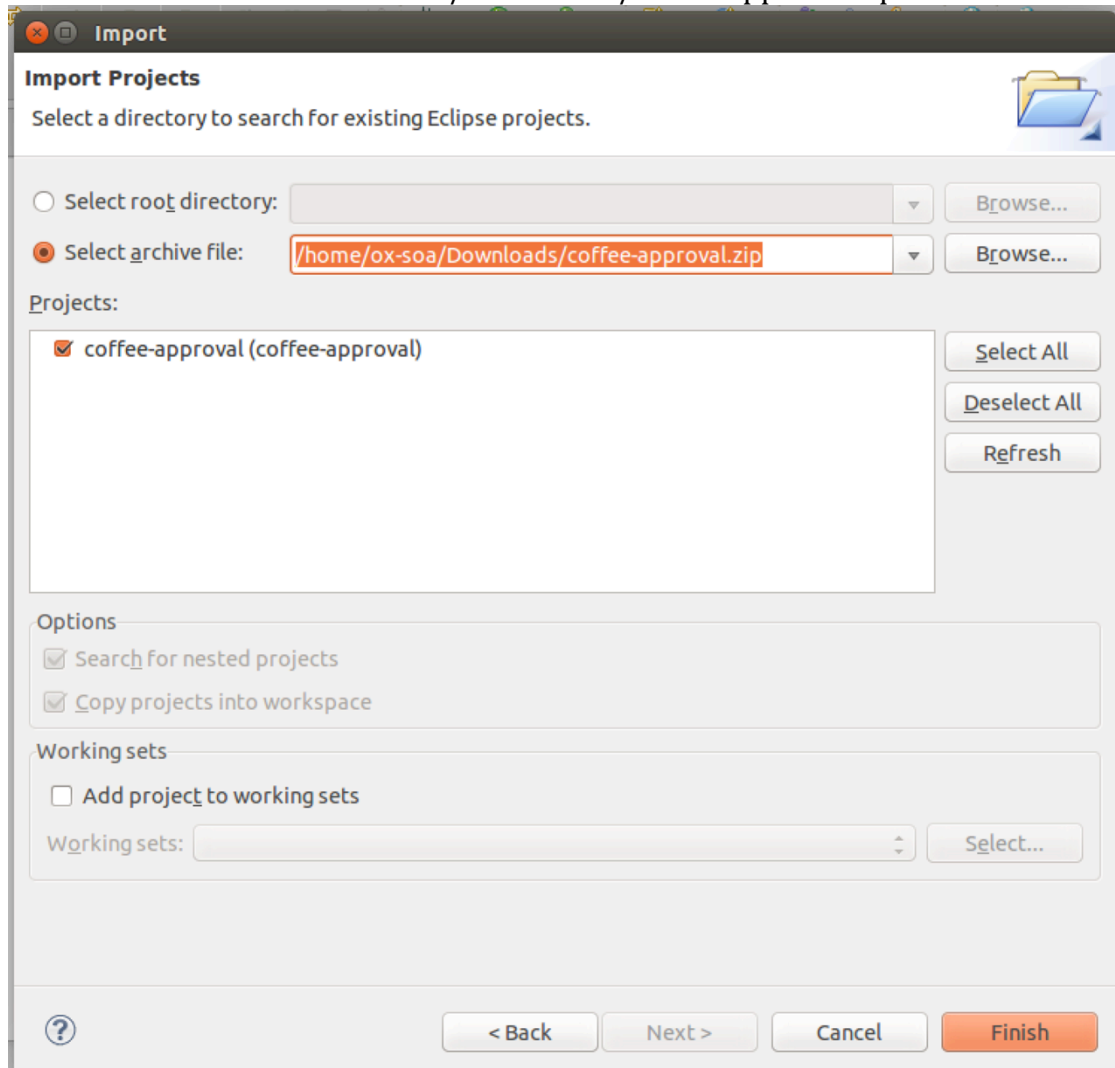
1. Start up Eclipse



## 2. File -> Import -> Existing Projects into Workspace

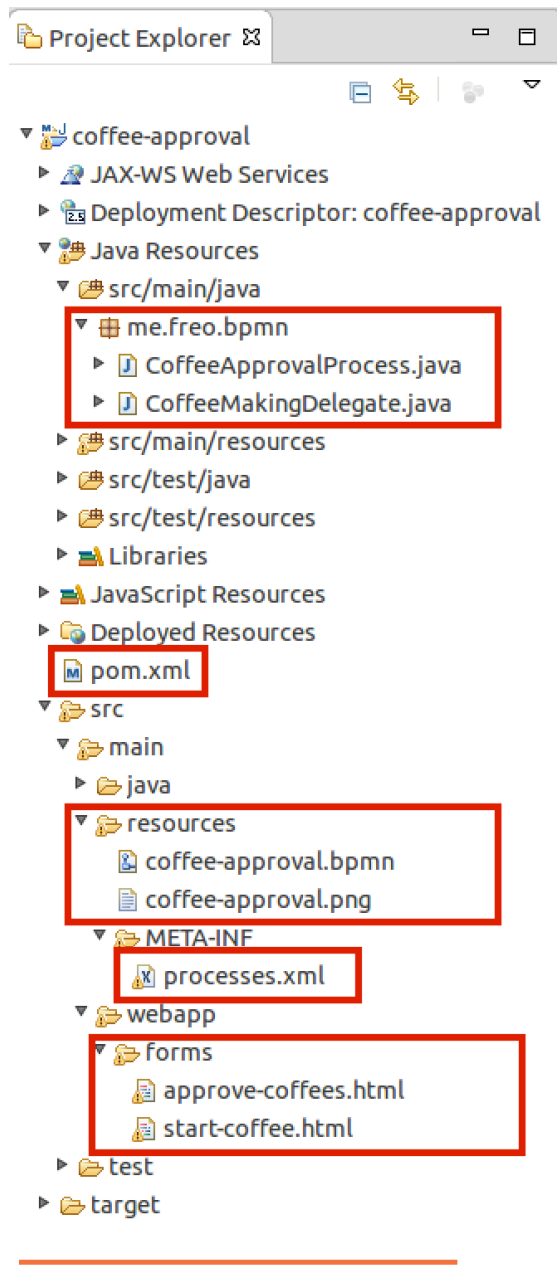


3. Click Next  
Select Archive File and choose ~/Downloads/coffee-approval.zip



4. Click **Finish**
5. You should have a project imported. Now expand the view so you can see the directories:





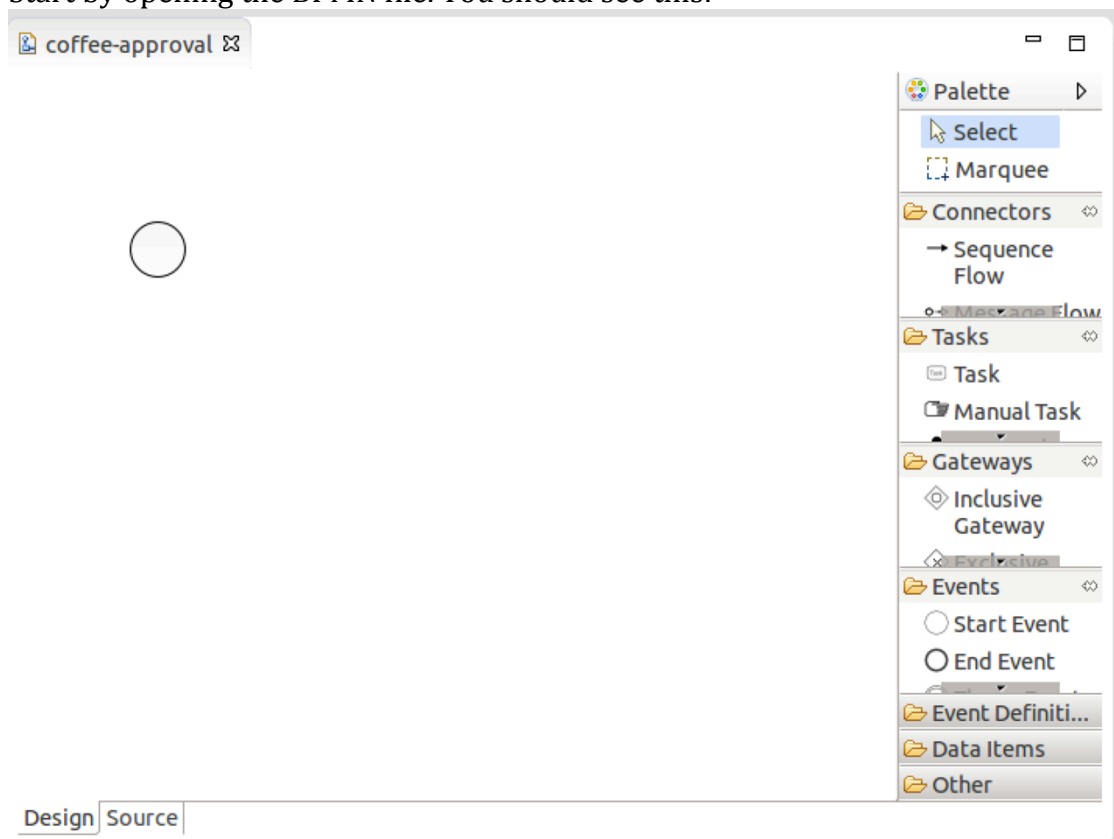
6. Lets examine each of these files in turn. The list is in the order we will come across them in the exercise, not in the graphical order:

pom.xml	This is the Maven build file that contains the Camunda BPMN dependencies and is used to build the WAR file we will deploy.
processes.xml	This is a deployment descriptor for the BPMN engine. It is documented here, but we will not be changing it: <a href="http://docs.camunda.org/manual/latest/reference/deployment-descriptors/">http://docs.camunda.org/manual/latest/reference/deployment-descriptors/</a>
CoffeeApproval Process.java	This is a class annotated with the @ProcessApplication annotation. This class is basically empty but is used to tell the BPMN engine that there is a process in this

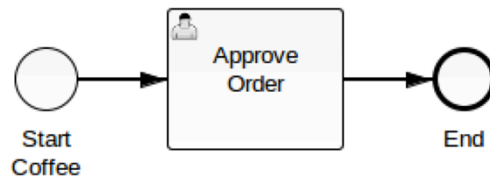


	WebApp WAR file
coffee-approval.bpmn	This is the process definition in the BPMN XML file. We will be using the graphical BPMN2 editor to edit this. This is the heart of the whole system.
coffee-approval.png	This is an auto-generated picture of the process, created from the BPMN XML above.
approve-coffees.html <i>and</i> start-coffee.html	These are HTML forms with extra Camunda specific tags. These are used by the Camunda engine to create Web Forms for the process to appear in each user's task list.
CoffeeMakingDelegate	This is a class that can be used to call out to Service Logic from the process, e.g. to our RESTful service.

7. Take a look at each of these files just to get a brief idea.
8. In order to save time, I have created the first three files and these won't need any modification.
9. Start by opening the BPMN file. You should see this:



10. The circle depicts the start of a process. Extend the diagram to look like the following. *Hint, Shift-Enter starts a newline in the text edit boxes.*



11. Make sure that the Task is a “User Task”.  
12. Each object has properties that can be browsed. If you click on the background canvas you can edit the Processes overall properties. Make sure the process is executable and called **coffee-process**

Process Id:

Name:

Is Executable: ☒

Data Objects:

name

Add or remove table entry by right-click.

Docum...tion:

13. Edit the properties for the Approve Order Task:

Id: **ApproveTask**  
Name: **Approve Order**  
Assignee: **demo**

**User Task "Approve Order"**

**General**

Id:

Name:

Assignee:

Candid...Users:

Candid...roups:

Form Key:

Due Date:

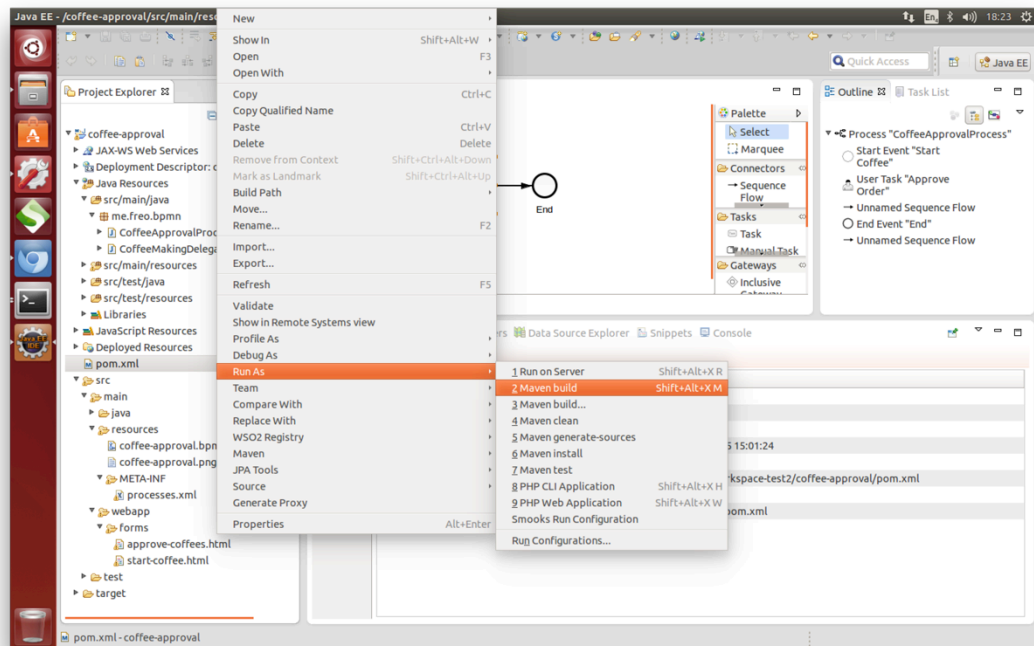
The due date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2012-03-01T15:30:23). See for more information [user guide](#).

Follow Date:

14. Now you can test your process.



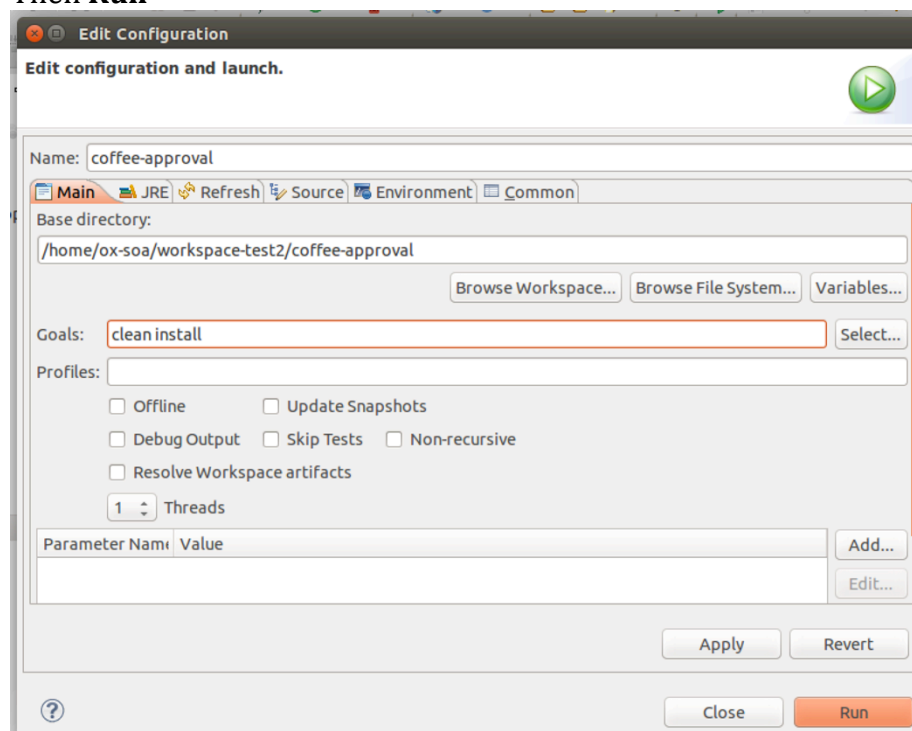
15. Let's use Maven to build the Process. Right Click on the pom.xml, and choose **Run As -> Maven Build**



16. The first time you do this, it will configure it. From then on you can do this without configuration.

In the Goals box type: **clean install**

Then **Run**



**17. You should see some text in the console like:**

```
[INFO] Assembling webapp [coffee-approval] in [/home/ox-soa/workspace-test2/coffee-approval/target/coffee-approval]
[INFO] Processing war project
[INFO] Copying webapp resources [/home/ox-soa/workspace-test2/coffee-approval/src/main/webapp]
[INFO] Webapp assembled in [20 msecs]
[INFO] Building war: /home/ox-soa/workspace-test2/coffee-approval/target/coffee-approval-0.0.1-SNAPSHOT.war
[INFO]
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ coffee-approval ---
[INFO] Installing /home/ox-soa/workspace-test2/coffee-approval/target/coffee-approval-0.0.1-SNAPSHOT.war to /home/
[INFO] Installing /home/ox-soa/workspace-test2/coffee-approval/pom.xml to /home/ox-soa/.m2/repository/me/freo/bpmn
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 2.909s
[INFO] Finished at: Mon Sep 14 18:27:00 BST 2015
[INFO] Final Memory: 14M/210M
[INFO]
```

**18. Ignore any red logging errors at the top of the console**

**19. We now need to copy the generated WAR file into the server to see if it works:**

```
cp workspace/coffee-approval/target/coffee-approval-0.0.1-SNAPSHOT.war servers/camunda/server/apache-tomcat-7.0.62/webapps/
```

*All one line!*

**20. I have adjusted the port that the Camunda server is on from the default of 8080 to be on 8090, so it doesn't clash with anything else (I hope!).**

**21. Start the server:**

```
cd ~/servers/camunda/server/apache-tomcat-7.0.62
```

```
bin/catalina.sh run
```

**22. We haven't yet copied our process across, but you should see the server deploy a default process:**

```
INFO: Detected @ProcessApplication class
org.camunda.bpm.example.invoice.InvoiceProcessApplication
Sep 14, 2015 7:47:26 PM
org.camunda.bpm.container.impl.deployment.ParseProcessesXmlStep
parseProcessesXmlFiles
INFO: Found process application file at file:/home/ox-soa/servers/camunda/server/apache-tomcat-7.0.62/webapps/camunda-invoice/WEB-INF/classes/META-INF/processes.xml
Sep 14, 2015 7:47:26 PM
org.camunda.bpm.container.impl.deployment.DeployProcessArchiveStep
logDeploymentSummary
INFO: Deployment summary for process archive 'camunda-invoice':

    invoice.bpmn
    invoice.png

Sep 14, 2015 7:47:26 PM
org.camunda.bpm.engine.impl.application.ProcessApplicationManager
logRegistration
INFO: ProcessApplication 'camunda-invoice' registered for DB deployments
[02e5497c-5af0-11e5-9e5b-56847afe9799]. Will execute process definitions

    invoice[version: 1, id: invoice:1:02f0e23f-5af0-11e5-9e5b-56847afe9799]
Deployment does not provide any case definitions.
Sep 14, 2015 7:47:27 PM
org.camunda.bpm.container.impl.RuntimeContainerDelegateImpl
deployProcessApplication
INFO: Process Application camunda-invoice successfully deployed.
```





23. Now we can copy over our process WAR. In a new command window:

```
cp ~/workspace/coffee-approval/target/coffee-approval-0.0.1-SNAPSHOT.war ~/servers/camunda/server/apache-tomcat-7.0.62/webapps/
All on one line!
```

24. If everything is going well, you can go back to the other terminal window and see the log, and you will see the coffee-approval process be deployed:

```
INFO: Server startup in 12741 ms
Sep 14, 2015 7:51:39 PM org.apache.catalina.startup.HostConfig deployWAR
INFO: Deploying web application archive /home/ox-soa/servers/camunda/server/apache-tomcat-7.0.62/webapps/coffee-approval-0.0.1-SNAPSHOT.war
Sep 14, 2015 7:51:40 PM org.apache.catalina.startup.TldConfig execute
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
Sep 14, 2015 7:51:40 PM
org.camunda.bpm.application.impl.ServletProcessApplicationDeployer
onStartup
INFO: Detected @ProcessApplication class me.freo.bpmn.CoffeeApprovalProcess
Sep 14, 2015 7:51:40 PM
org.camunda.bpm.container.impl.deployment.ParseProcessesXmlStep
parseProcessesXmlFiles
INFO: Found process application file at file:/home/ox-soa/servers/camunda/server/apache-tomcat-7.0.62/webapps/coffee-approval-0.0.1-SNAPSHOT/WEB-INF/classes/META-INF/processes.xml
Sep 14, 2015 7:51:40 PM
org.camunda.bpm.container.impl.deployment.DeployProcessArchiveStep
logDeploymentSummary
INFO: Deployment summary for process archive 'coffee-approval':

    coffee-approval.png
    coffee-approval.bpmn

Sep 14, 2015 7:51:40 PM
org.camunda.bpm.engine.impl.application.ProcessApplicationManager
logRegistration
INFO: ProcessApplication 'Coffee Approval Process' registered for DB
deployments [a28f4ce0-5b11-11e5-a9e9-56847afe9799]. Will execute process
definitions

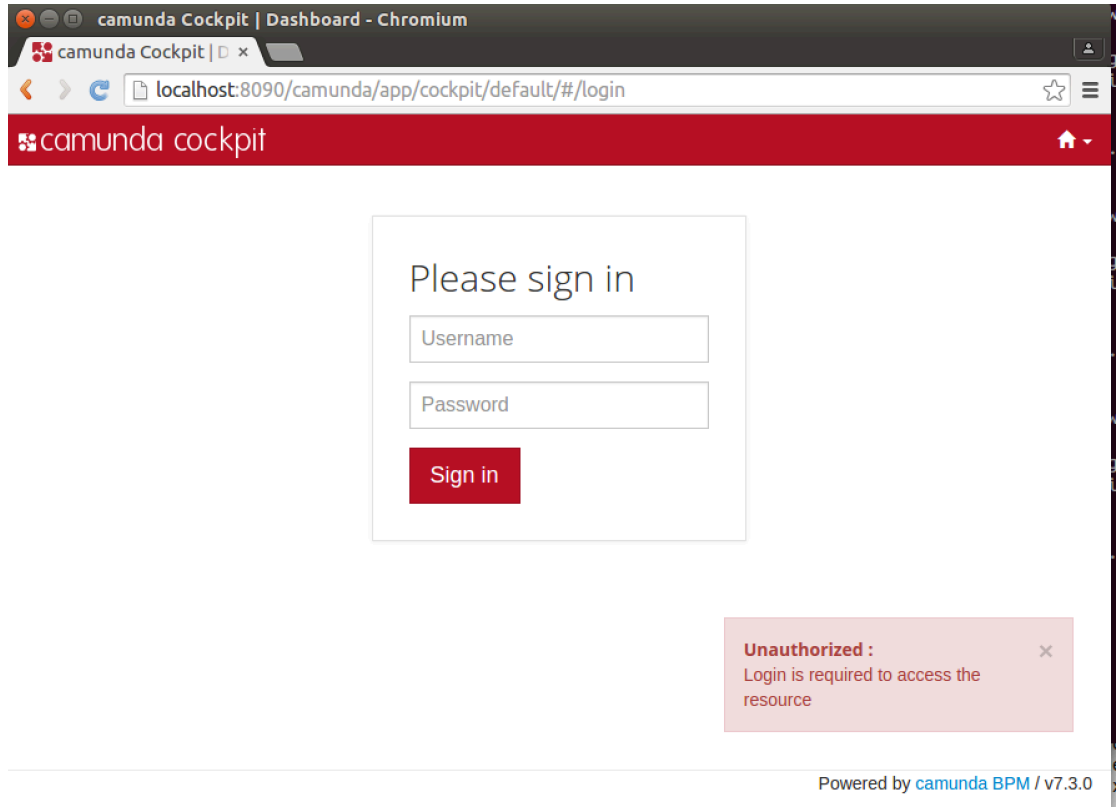
    coffee-approval[version: 1, id: coffee-approval:1:a293b9b3-5b11-11e5-a9e9-56847afe9799]
Deployment does not provide any case definitions.
Sep 14, 2015 7:51:40 PM
org.camunda.bpm.container.impl.RuntimeContainerDelegateImpl
deployProcessApplication
INFO: Process Application Coffee Approval Process successfully deployed.
Sep 14, 2015 7:51:40 PM org.apache.catalina.startup.HostConfig deployWAR
INFO: Deployment of web application archive /home/ox-soa/servers/camunda/server/apache-tomcat-7.0.62/webapps/coffee-approval-0.0.1-SNAPSHOT.war has finished in 611 ms
```

25. Let's test the process now.

26. Start up a browser (Chromium) and browse to  
<http://localhost:8090/camunda>



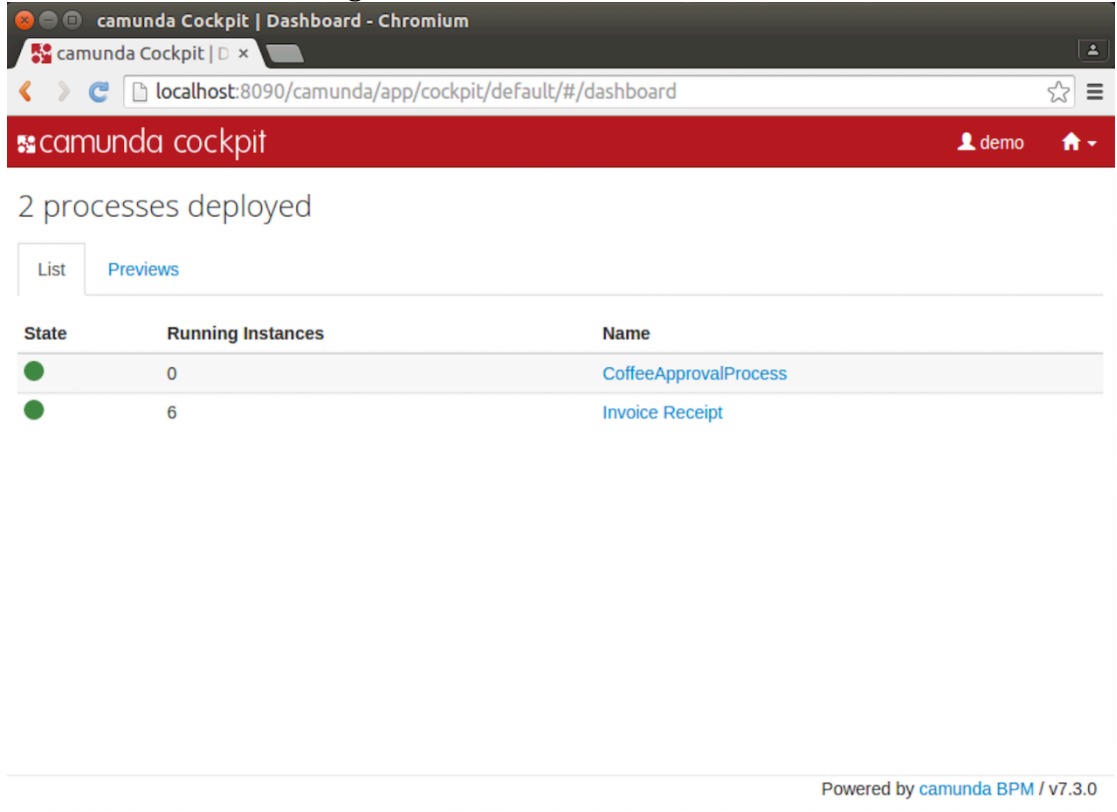
## 27. You should see:



## 28. Sign in with *demo/demo*



29. You should see something similar to this:



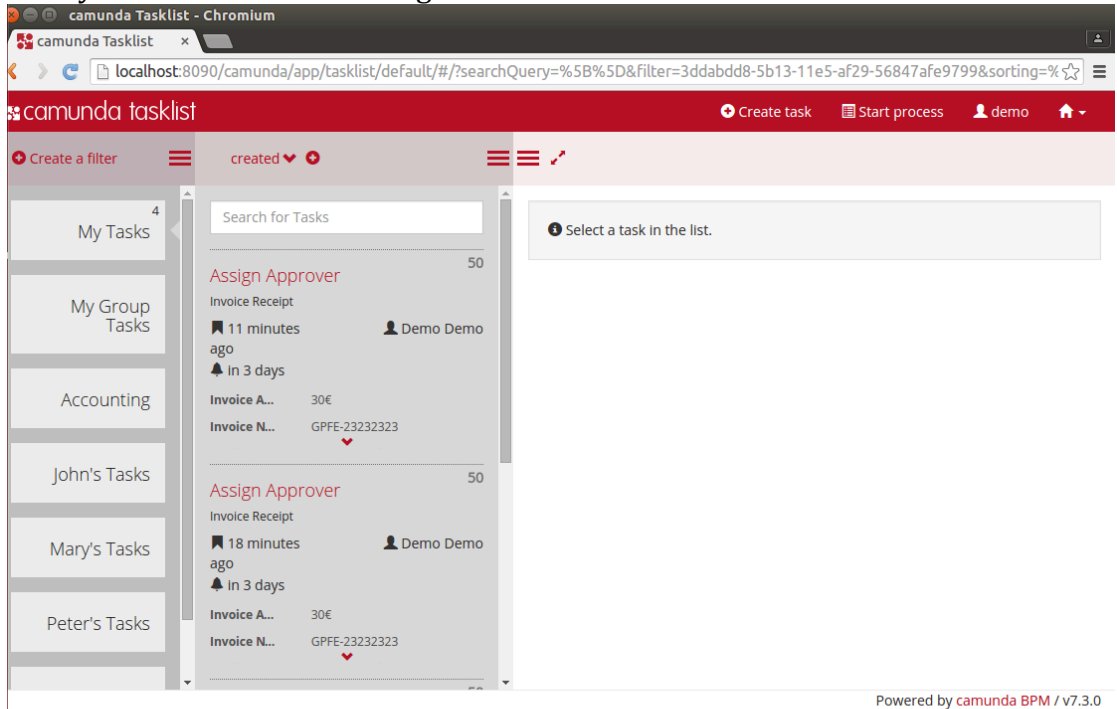
The screenshot shows the Camunda Cockpit Dashboard in a Chromium browser. The address bar indicates the URL is `localhost:8090/camunda/app/cockpit/default/#/dashboard`. The dashboard header is red with the Camunda logo and a user profile icon labeled 'demo'. Below the header, it states '2 processes deployed'. There are two tabs: 'List' (selected) and 'Previews'. A table displays the deployed processes:

State	Running Instances	Name
	0	<a href="#">CoffeeApprovalProcess</a>
	6	<a href="#">Invoice Receipt</a>

At the bottom right, it says 'Powered by camunda BPM / v7.3.0'.

30. Select the little Home icon in the corner and choose Tasklist

31. Now you should see something like this:

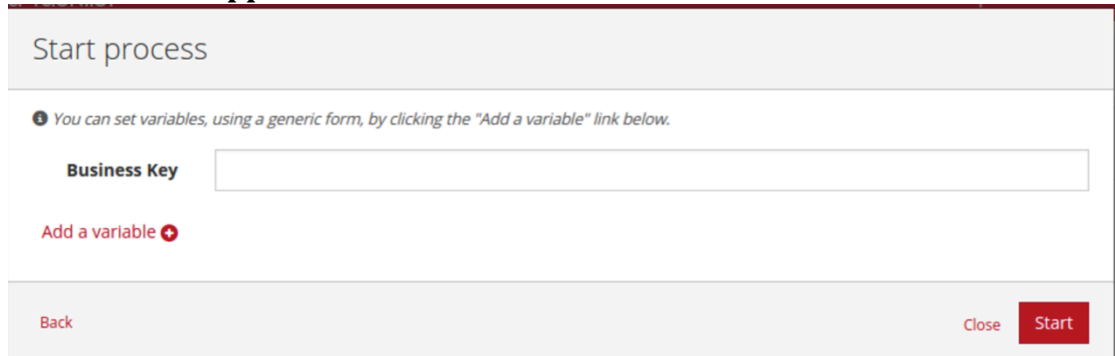


The screenshot shows the Camunda Tasklist interface in a Chromium browser. The address bar indicates the URL is `localhost:8090/camunda/app/tasklist/default/#/?searchQuery=%5B%5D&filter=3ddabdd8-5b13-11e5-af29-56847afe9799&sorting=%5B%5D`. The header is red with the Camunda logo and navigation icons for 'Create task', 'Start process', 'demo', and a home icon. Below the header, there's a 'Create a filter' button and a 'created' filter dropdown. A sidebar on the left lists task categories: 'My Tasks' (4), 'My Group Tasks', 'Accounting', 'John's Tasks', 'Mary's Tasks', and 'Peter's Tasks'. The main area displays a list of tasks. The first task is 'Assign Approver' for 'Invoice Receipt', assigned to 'Demo Demo', 11 minutes ago, and due in 3 days. It shows two sub-tasks: 'Invoice A...' with a value of 30€ and 'Invoice N...' with a value of GPFE-23232323. The second task is identical. A message box on the right says 'Select a task in the list.' At the bottom right, it says 'Powered by camunda BPM / v7.3.0'.



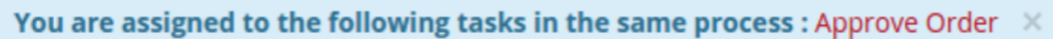
32. Click on **Start Process**

33. Choose **CoffeeApprovalProcess**



34. Enter anything you like in Business Key and then click **Start**

35. Do you remember where you set the assignee for the User Task in the BPMN process to **demo**? Well this has just happened and you are logged in as demo and hence the portal has popped up a message saying you've been assigned a task to work on:



36. Click on **Approve Order**



37. You should see something like this:

## Approve Order

CoffeeApprovalProcess

Set follow-up date

Set due date

Add groups

Demo Demo ✕

Form

History

Diagram

Description

*You can set variables, using a generic form, by clicking the "Add a variable" link below.*

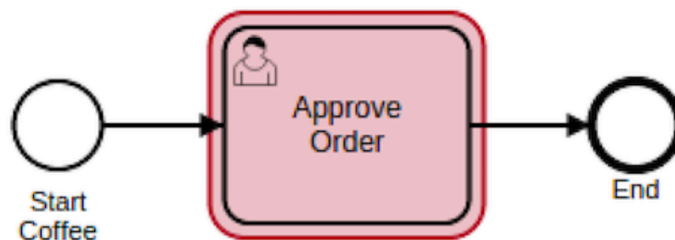
Add a vari... +

Load Vari... ⓘ

Complete

38. Click on the **Diagram** tab

You will see the process you designed, now with the current step highlighted:



39. Go back to the **Form** tab and click Complete. This will “complete” this instance of the process.



**40.** This process works, but let's be honest, it sucks a bit.

In order to improve it, we need to get some real information when we start the process (the name of the customer, the type of coffee that they want). Then we need to have some approval step (did the payment go through ok?). Then we can make the coffee.

*Of course we could be much cooler: we could model the whole of the barista/customer interaction in swimlanes. But let's do this step at a time.*

**41.** Go back to the BPMN diagram in the Eclipse editor. I have already created a form to capture the name and coffee order. We need to associate this with the start event.

**42.** Click on the start event, and select the properties tab. In the **Form Key** field type:  
embedded:app:forms/start-coffee.html

Take a look at this HTML file.

This means that the form will be embedded into the Task list, and the form is inside the webapp at the path given.

The screenshot shows the Eclipse IDE with the properties view for a BPMN Start Event. The event is named "Start Event 'Start Coffee'". The properties are as follows:

General	Id:	StartEvent_1
	Name:	Start
	Form Key:	embedded:app:forms/start-coffee.html
	Asynch...efore:	<input type="checkbox"/> More information on asynchronous continuation can be found in the <a href="#">user guide</a> .
	Asynch...After:	<input type="checkbox"/> More information on asynchronous continuation can be found in the <a href="#">user guide</a> .

**43.** Do the same for the User Task, but with the other form specified.

Also take a look at the other HTML file and spot the difference.

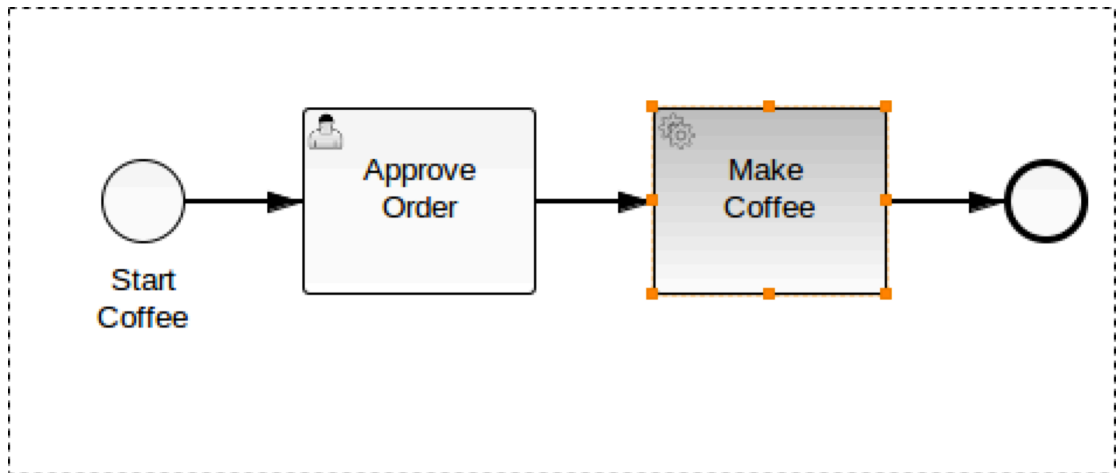
The screenshot shows the Eclipse IDE with the properties view for a BPMN User Task. The task is named "User Task 'Approve Order'". The properties are as follows:

General	Id:	ApproveTask
	Name:	Approve
	Assignee:	demo
	Candid...Users:	
	Candid...roups:	
	Form Key:	embedded:app:forms/approve-coffees.html
	Due Date:	

The due date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2012-03-01T15:30:23). See for more information [user guide](#).

**44.** Now let's add a task to actually "make" the coffees. Delete the End event (the bold circle on the right). Add a new Service Task and a new End event, so that the process looks like this:





45. Now select the properties of the Service Task, and click on the radio button to the right of the word Class (it may be only half visible?). This will enable a button entitled **Choose Class**. Click on it.

Service Task "Make Coffee"

**General**

Id: ServiceTask\_2

Name: Make

Class:  Choose Class

Expression:

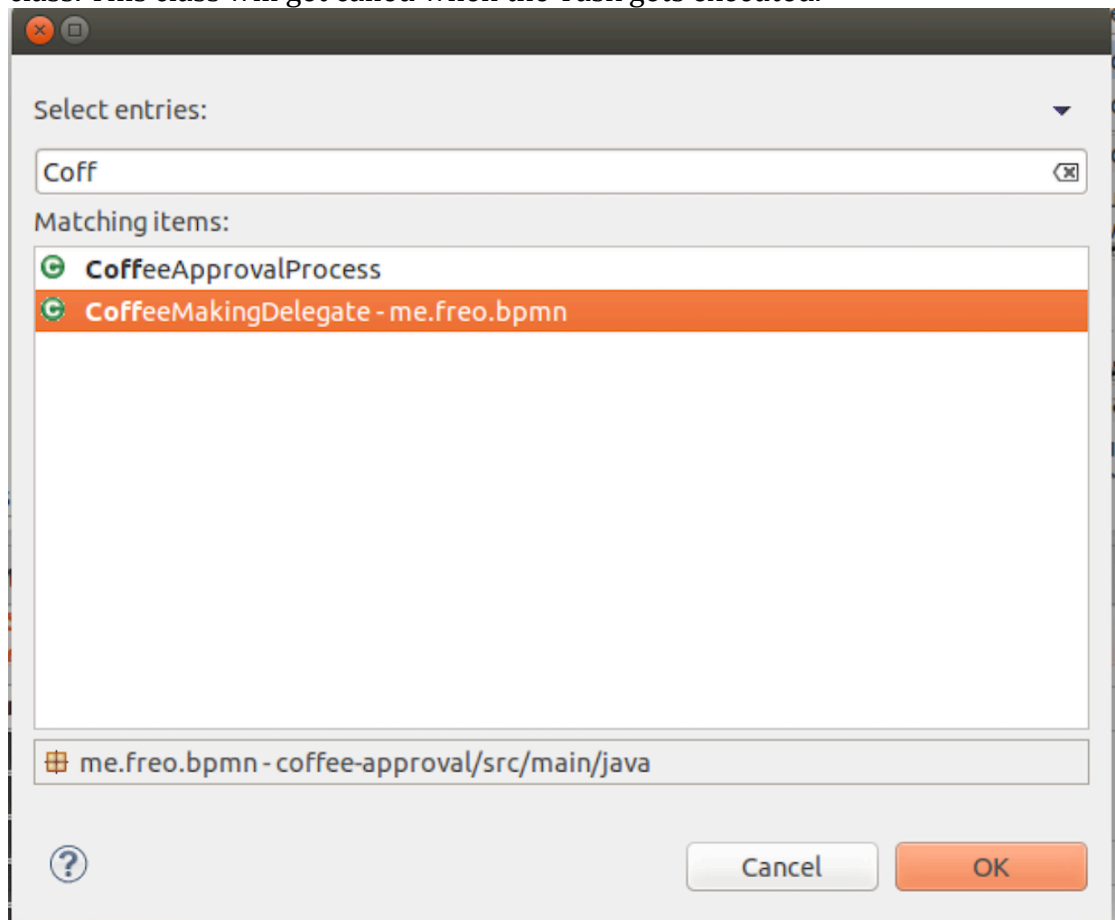
ExpressionDelegate:

Result Variable:

Asynchronous Before: ☐ More information on asynchronous continuation can be found in the [user guide](#).

Asynchronous After: ☐ More information on asynchronous continuation can be found in the [user guide](#).

46. Type in Coffee into the box and then select the CoffeeMakingDelegate class. This class will get called when the Task gets executed.



47. Click OK.
48. Save the BPMN.
49. Take a look at the Java code for this class. It should make some sense now.
50. Rebuild the WAR file using the pom.xml / Run As-> Maven Build from above.
51. Recopy the file to the webapps directory (as above).
52. Check to see if it deployed correctly by checking the terminal window where Tomcat is running (*catalina.sh run*)





53. Now go back to the task list and start the process again. You should now see a nice form inviting you to enter more useful information:

Start process

Customer Name

Type of Coffee

Back

Close

Start

54. Fill in the form and hit **Start**

55. Now when you go to Approve the Order you will see more useful info:

Customer Name

Paul the Hipster

Type of Coffee

Flat White

Approve?

☐

Save

Complete

56. When you approve it this time, go check on the logs in the terminal window:

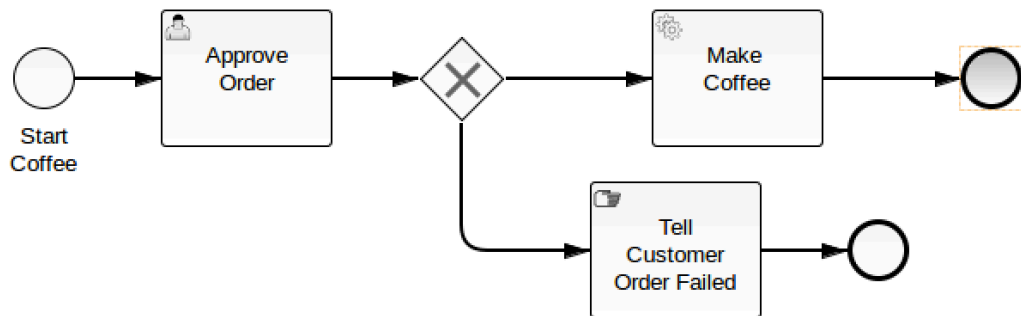
```
Sep 14, 2015 8:55:37 PM me.freo.bpmn.CoffeeMakingDelegate execute  
INFO: Processing request by 'Paul the Hipster'  
Sep 14, 2015 8:55:37 PM me.freo.bpmn.CoffeeMakingDelegate execute  
INFO: Processing request for 'Flat white'
```

57. *Actually approved?*

We never checked if the Barista approved making the coffee. Edit the process to add an Exclusive Gateway and actions to take depending if it is approved.



58. It should look a bit like this:



59. The secret to making this work is to put some conditions on the flows (the arrows). On one arrow put the condition `${approved}`, and on the other put the condition `${!approved}`. Do this in the condition boxes in the properties.

Unnamed Sequence Flow

General

Id: SequenceFlow\_10

Name:

Condition: \${approved}

Check that the log only gets appended when you click the approve button on the task.

60. Easy extension: Try out the BPMN.IO App installed in Chromium. This is a Chrome extension that creates and edits BPMN.

### 61. Extension 2

Extend the Java Delegate so that it actually calls the REST service multiple times to order the coffee.

### 62. Extension 3

You could also ask for card information in the start form, and then extend the process to pay as well.

### 63. Extension 4

You could also adjust it to handle ordering multiples of the same drink, by asking how many at the start.

