

Exercise 13

Creating and executing 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

- Camunda BPMN Modeler 1.1.1
- Camunda BPMN runtime 7.5.0

Steps.

1. Let's get our project initiated. Firstly let's make a directory structure:
`mkdir ~/bpnn/`

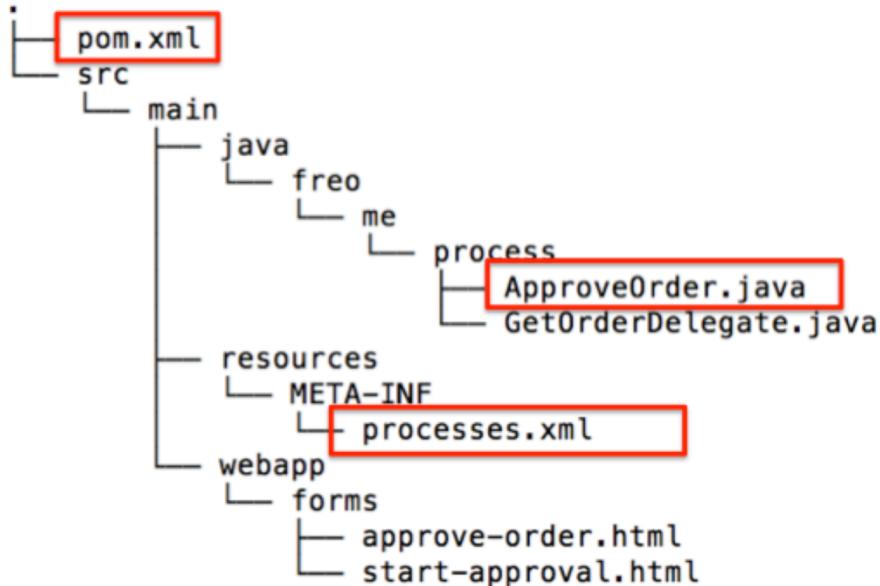
2. Now lets grab a simple default project layout:

```
cd ~/bpnn  
wget http://freo.me/bpmn-kit -O process-archive.zip  
unzip process-archive.zip
```



© Paul Fremantle 2012. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

3. The main files here are the build (pom.xml), a generic process descriptor XML (see <https://docs.camunda.org/manual/latest/user-guide/process-applications/the-processes-xml-deployment-descriptor/#empty-processes-xml>) and a basic Application class (ApproveOrder). We'll ignore the other files for the moment.



10 directories, 6 files

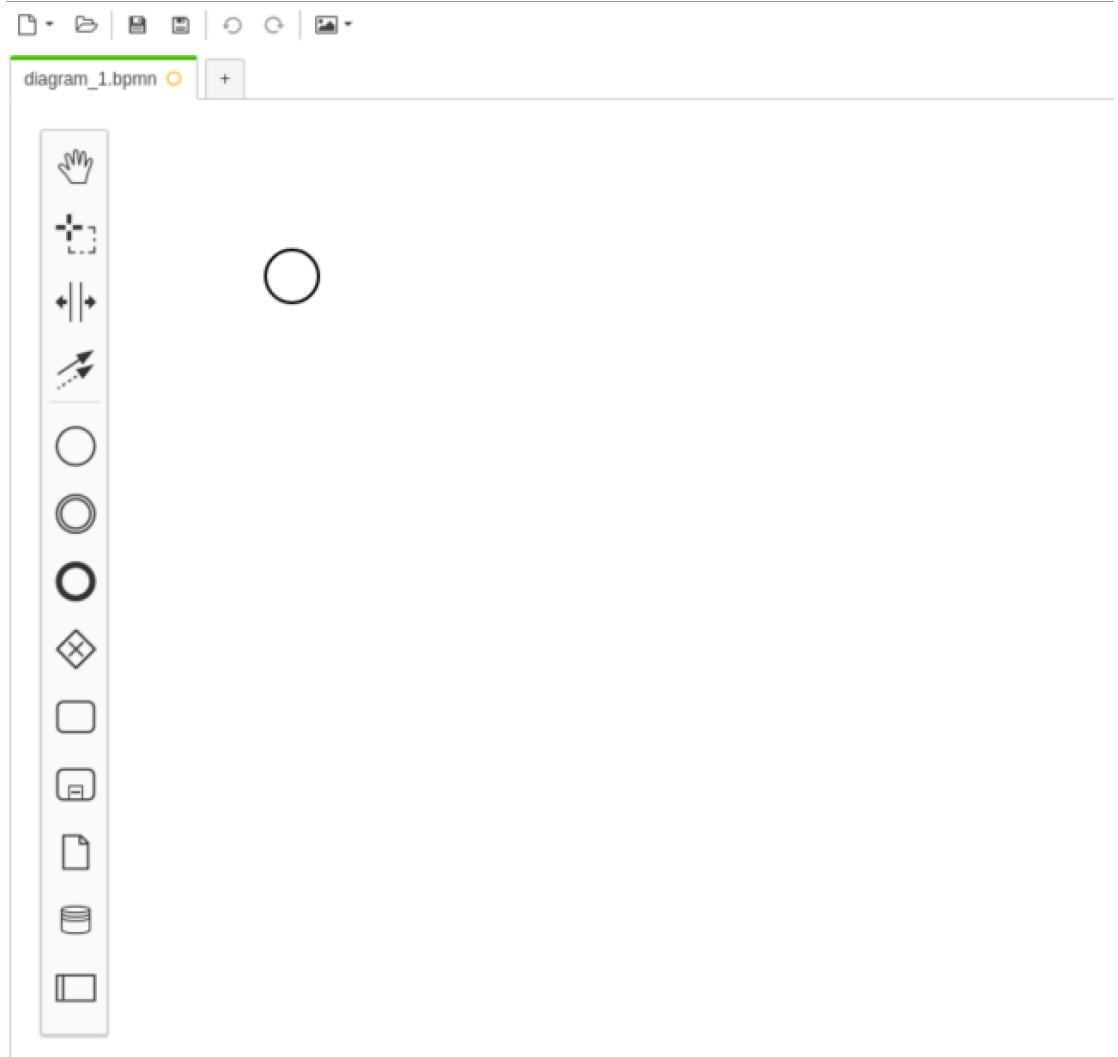
Take a look at those three.

With the three mentioned files, this will create a WAR that will run *whatever* BPMN process we deploy into the **src/main/resources** folder.

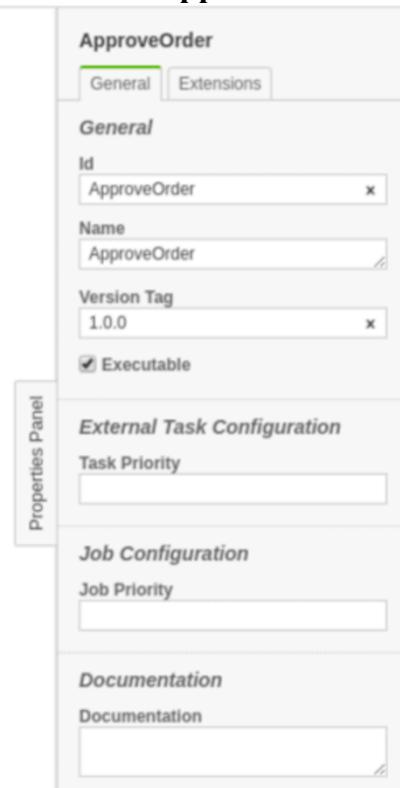
4. Let's create our first BPMN.
5. Start the Camunda BPMN Modeler
cd ~/camunda-modeler
./camunda-modeler
6. Click on Create a BPMN diagram.



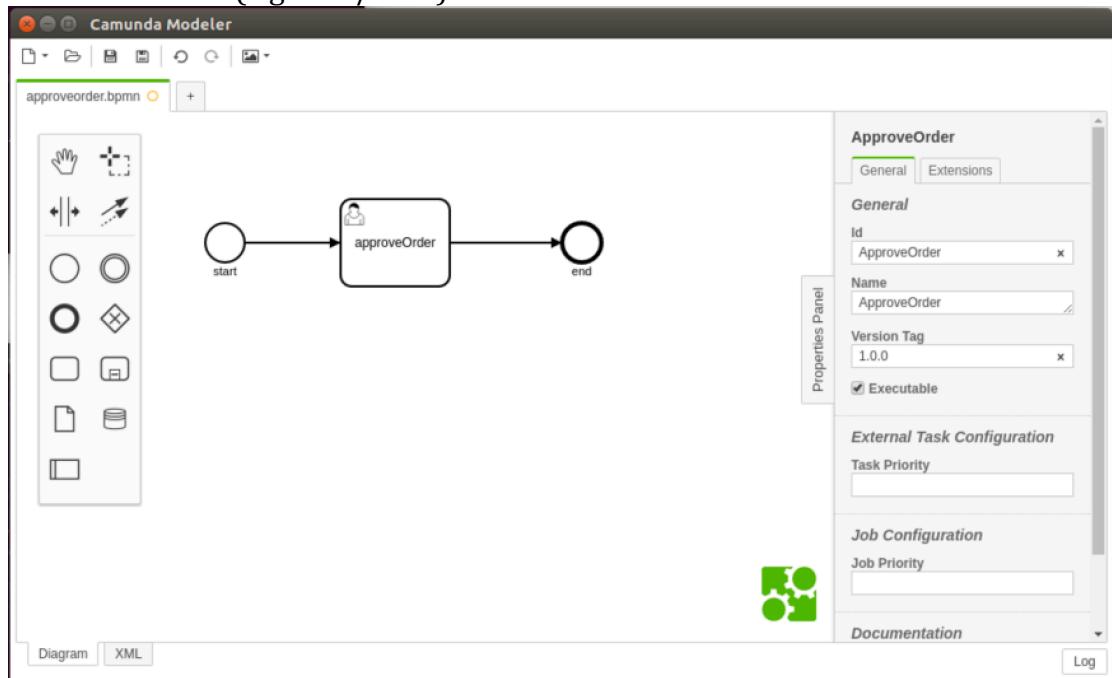
7. You should see:



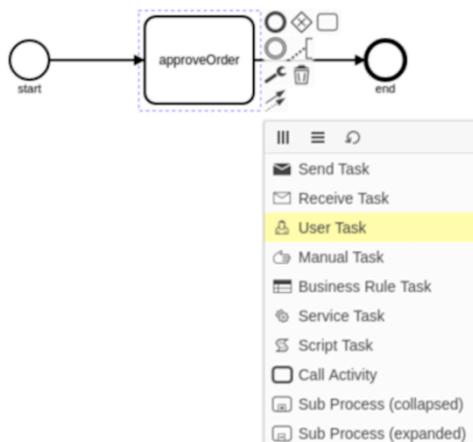
8. Make sure the properties panel is expanded.
9. Make sure the **Executable** tag is ticked, and change the process id and name to be **ApproveOrder**. Change the version number to be 1.0.0:



10. Use the tool to draw a simple process like this. Make the **id** of each object match the name (e.g start/start).

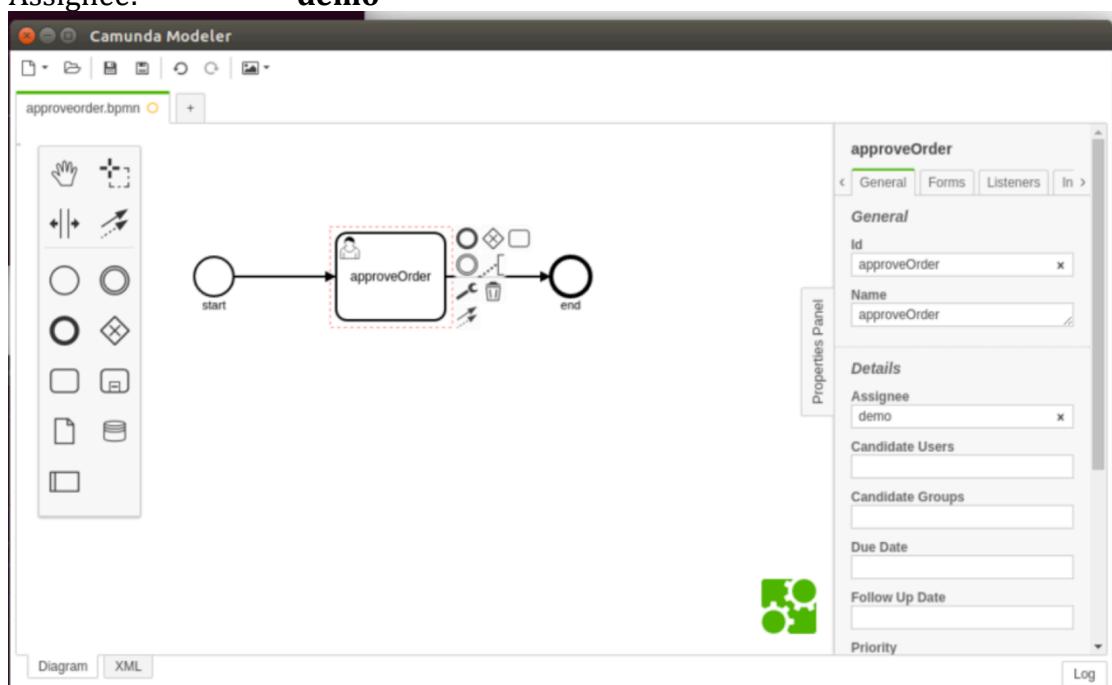


11. In order to make the task into a User Task (with the little “man” icon), click on the spanner/wrench and choose User Task.



12. Edit the properties for the Approve Order Task:

Id: **ApproveOrder**
Name: **ApproveOrder**
Assignee: **demo**



13. Save the file as `~/bpmn/src/main/resources/approveorder.bpmn`
Hint: Sometimes there is a bug with the BPMN editor and the Save dialog box appears behind the main window. To solve it, click on the new icon that appears in the Launcher.



14. Now you can test your process.

15. In the BPMN directory, type:

```
mvn clean install
```

16. You should see some text in the console like:

```
[INFO] Installing /home/oxsoa/bpmn/target/approve-order-0.1.0.war to /home/oxsoa/.m2/repository/freeme/process/approve-order/0.1.0/approve-order-0.1.0.war
[INFO] Installing /home/oxsoa/bpmn/pom.xml to /home/oxsoa/.m2/repository/freeme/process/approve-order/0.1.0/approve-order-0.1.0.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.245 s
[INFO] Finished at: 2016-06-14T09:02:34+01:00
[INFO] Final Memory: 22M/290M
[INFO] -----
oxsoa@oxsoa:~/bpmn$
```

17. The Camunda server is running on 8080, so make sure none of your other servers is still running on 8080.

18. Start the server:

```
cd ~/servers/camunda/server/apache-tomcat-8.0.24
bin/catalina.sh run
```

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

```
oxsoa@oxsoa:~/servers/camunda/server/apache-tomcat-8.0.24
14-Jun-2016 09:08:07.393 INFO [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars 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.
14-Jun-2016 09:08:07.394 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-07015 Detected @ProcessApplication class 'org.camunda.bpm.example.invoice.InvoiceProcessApplication'
14-Jun-2016 09:08:07.401 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-08024 Found processes.xml file at file:/home/oxsoa/servers/camunda/server/apache-tomcat-8.0.24/webapps/camunda-invoice/WEB-INF/classes/META-INF/processes.xml
14-Jun-2016 09:08:07.457 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-08023 Deployment summary for process archive 'camunda-invoice':
assign-approver-groups.dmn
invoice.v2.bpmn

14-Jun-2016 09:08:07.917 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-07021 ProcessApplication 'camunda-invoice' registered for DB deployments [13711c20-313b-11e6-b553-024208065a2e, 139d8356-313b-11e6-b553-024208065a2e]. Will execute process definitions
invoice[version: 1, id: invoice:1:137fc223-313b-11e6-b553-024208065a2e]
invoice[version: 2, id: invoice:2:13a328a9-313b-11e6-b553-024208065a2e]
Deployment does not provide any case definitions.
14-Jun-2016 09:08:07.923 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo SPIN-01010 Discovered Spin data format provider: org.camunda.spin.impl.json.jackson.format.JsonFormatProvider[name = application/json]
14-Jun-2016 09:08:07.923 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo SPIN-01010 Discovered Spin data format provider: org.camunda.spin.impl.xml.dom.format.DomXmlDataFormatProvider[name = application/xml]
14-Jun-2016 09:08:07.925 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo SPIN-01009 Discovered Spin data format: org.camunda.spin.impl.xml.dom.format.DomXmlDataFormat[name = application/xml]
14-Jun-2016 09:08:07.925 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo SPIN-01009 Discovered Spin data format: org.camunda.spin.impl.json.jackson.format.JsonFormatProvider[name = application/json]
14-Jun-2016 09:08:08.095 WARNING [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logWarn ENGINE-09004 Warnings during parsing:
* Exclusive Gateway 'reviewSuccessful_gw' has outgoing sequence flow 'reviewSuccessful' without condition which is not the default flow. We assume it to be the default flow, but it is bad modeling practice, better set the default flow in your gateway. | invoice.v1.bpmn | line 0 | column 0
14-Jun-2016 09:08:09.253 INFO [localhost-startStop-1] org.camunda.bpm.example.invoice.service.NotifyCreditorService.execute
... Now notifying creditor Bobby's Office Supplies

14-Jun-2016 09:08:09.648 INFO [localhost-startStop-1] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-08050 Process application camunda-invoice successfully deployed
14-Jun-2016 09:08:09.649 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory /home/oxsoa/servers/camunda/server/apache-tomcat-8.0.24/webapps/camunda-invoice has finished in 3,081 ms
14-Jun-2016 09:08:09.649 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory /home/oxsoa/servers/camunda/server/apache-tomcat-8.0.24/webapps/engine-rest
14-Jun-2016 09:08:10.910 INFO [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars 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.
14-Jun-2016 09:08:11.025 INFO [localhost-startStop-1] org.jboss.resteasy.logging.impl.Slf4jLogger.info Deploying javax.ws.rs.core.Application: class org.camunda.bpm.engine.rest.impl.application.DefaultApplication
14-Jun-2016 09:08:11.114 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory /home/oxsoa/servers/camunda/server/apache-tomcat-8.0.24/webapps/engine-rest has finished in 1,465 ms
```



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

```
cp ~/bpmmn/target/approve-order-0.1.0.war  
~/servers/camunda/server/apache-tomcat-8.0.24/webapps
```

All on one line!

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

```
14-Jun-2016 09:13:24.717 INFO [localhost-startStop-2] org.apache.jasper.servlet.TldScanner.scanJars 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.  
14-Jun-2016 09:13:24.718 INFO [localhost-startStop-2] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-07015 Detected @ProcessAp  
plication class 'freo.me.process.ApproveOrder'  
14-Jun-2016 09:13:24.720 INFO [localhost-startStop-2] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-08024 Found processes.xml  
file at file:/home/oxsoa/servers/camunda/server/apache-tomcat-8.0.24/webapps/approve-order-0.1.0/WEB-INF/classes/META-INF/processes.  
xml  
14-Jun-2016 09:13:24.724 INFO [localhost-startStop-2] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-08023 Deployment summary  
for process archive 'Order Approval Process':  
  
    approveorder.bpmn  
  
14-Jun-2016 09:13:24.757 INFO [localhost-startStop-2] org.camunda.commons.logging.BaseLogger.logInfo ENGINE-07021 ProcessApplication  
'Order Approval Process' registered for DB deployments [8ce46b42-313b-11e6-ae62-024208065a2e, ddb75bdb-3207-11e6-bbd0-024208065a2e].  
Will execute process definitions  
  
    ApproveOrder[version: 1, id: ApproveOrder:1:8cf708e4-313b-11e6-ae62-024208065a2e]  
    ApproveOrder[version: 2, id: ApproveOrder:2:ddba902d-3207-11e6-bbd0-024208065a2e]  
Deployment does not provide any case definitions.
```

22. Let's test the process now.

23. Start up a browser (Chromium) and browse to
<http://localhost:8080/camunda/app/cockpit/>

24. You should see:



Please sign in

Username

Password

Sign in

25. Sign in with *demo/demo*



26. You should see something similar to this:

The screenshot shows the Camunda Cockpit dashboard in a Mozilla Firefox browser. The URL is `localhost:8080/camunda/app/cockpit/default/#/`. The dashboard has three main sections: Processes, Decisions, and Deployments.

Section	Value
Processes	2 process definitions 24 running instances 0 process instances with incidents
Decisions	1 decision definition
Deployments	4 deployments
Batches	0 running batches 0 completed batches

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



28. Now you should see something like this:

Camunda Tasklist - Mozilla Firefox

localhost:8080/camunda/app/tasklist/default/#/?sea

Search

Camunda Tasklist

Keyboard Shortcuts Create task Start process Demo Demo

Create a filter + CREATION_DATE + < > ↻

SEARCH_PLACEHOLDER

Select a task in the list.

My Tasks (8)

My Group Tasks

Accounting

John's Tasks

Mary's Tasks

Peter's Tasks

All Tasks

Review Invoice

Invoice Receipt Demo Demo
Due in 2 days, created 11 minutes ago 50
Invoice A... 10.99
Invoice N... PSAG 5342

Review Invoice

Invoice Receipt Demo Demo
Due in 2 days, created 11 minutes ago 50
Invoice A... 10.99
Invoice N... PSAG 5342

Review Invoice

Invoice Receipt Demo Demo
Due in 2 days, created 12 minutes ago 50
Invoice A... 10.99
Invoice N... PSAG 5342

Review Invoice

Invoice Receipt Demo Demo
Due in 2 days, created 12 minutes ago 50
Invoice A... 10.99
Invoice N... PSAG 5342

Review Invoice

Powered by camunda BPM / v7.5.0

Those tasks are from the default process that comes with Camunda.
Ignore those.

29. Click on **Start Process**



30. Choose **ApproveOrder**

Start process

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

Business Key

+

Back Close Start

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

32. 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:

You are assigned to the following tasks in the same process : ×
approveOrder

33. Click on **Approve Order**



34. You should see something like this:

approveOrder

ApproveOrder

Set follow-up date Set due date Add groups Demo Demo x

Form History Diagram Description

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

Business Key

+ ④

Complete

35. Click on the **Diagram** tab

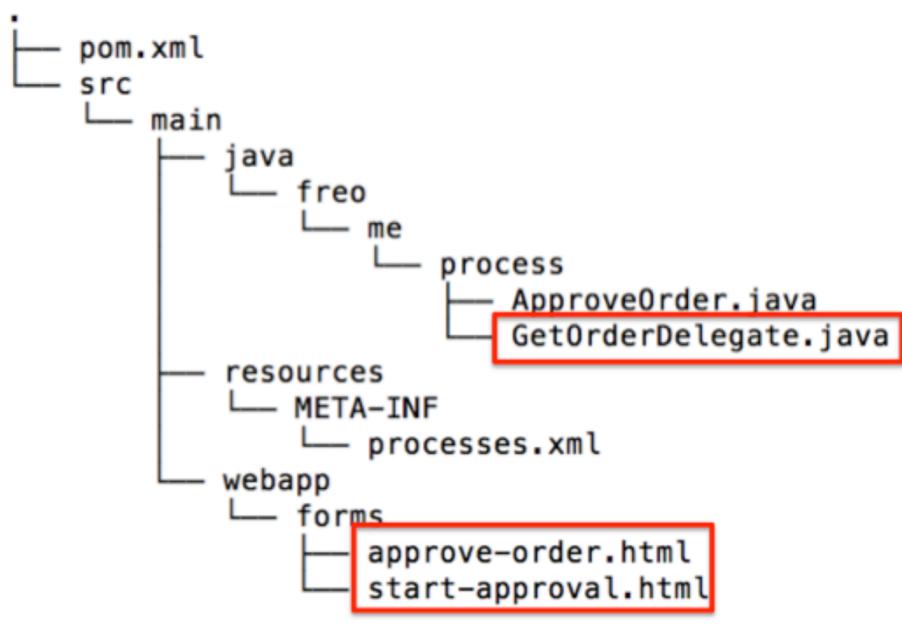
You will see the process you designed, now with the current step highlighted:



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



37. This process works, but lets be honest, it is really pointless.
38. To make the process more interesting we need to tie it in with our existing order processing system. A simple way to do that would be to take an order ID as input, and then gather information about that order to put in front of the approver.
39. If you look at the directory, you should see that there are three other files that came with process-archive.zip that will help:



40. There are two forms already deployed in the `src/main/webapp/forms` directory. Take a look at these.
41. These are standard HTML with some extensions for Camunda that enable the developer to specify fields that will be displayed or collected and to map those fields to process instance variables.



42. There is also a “Delegate” Java class. Here is the code listing:

```
public class GetOrderDelegate implements JavaDelegate {  
    private final static Logger LOGGER =  
        Logger.getLogger("APPROVAL-REQUESTS");  
  
    public void execute(DelegateExecution de) throws Exception {  
  
        String id = (String) de.getVariable("id");  
        LOGGER.info("Processing request by '" + de.getVariable("id"));  
  
        Client client = ClientBuilder.newClient();  
        WebTarget target = client.target("http://localhost:80")  
            .path("purchase").path(id);  
  
        Response response =  
            target.request(MediaType.APPLICATION_JSON).get();  
        if (response.getStatus() == 200) {  
  
            JSONObject json =  
                new JSONObject(response.readEntity(String.class));  
            de.setVariable("lineItem", json.get("lineItem"));  
            de.setVariable("date", json.get("date"));  
            de.setVariable("quantity", json.get("quantity"));  
            de.setVariable("customerNumber",  
                json.get("customerNumber"));  
            de.setVariable("poNumber", json.get("poNumber"));  
  
        } else {  
            throw new BpmnError("ID NOT FOUND");  
        }  
    }  
}
```

43. This is a class specific to Camunda that can be called from a “ServiceTask”. As you can see, it takes a process variable (id) and uses it to look up a PO from our purchase server, using JAX RS REST client code.

44. Let's use these extra files.

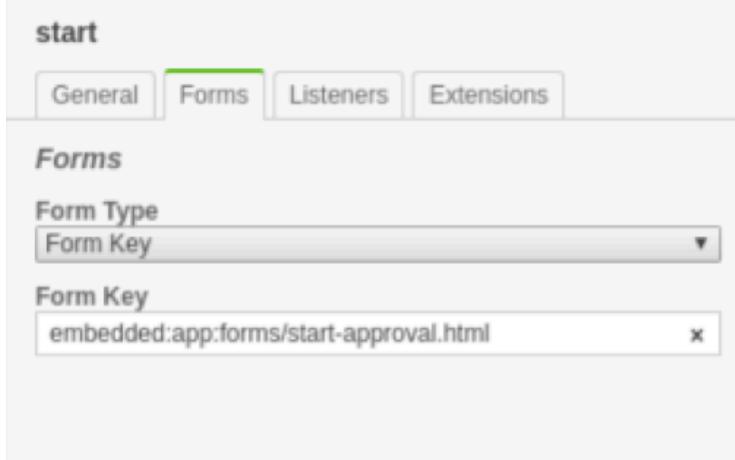
45. Firstly expand the BPMN flow to look like this:



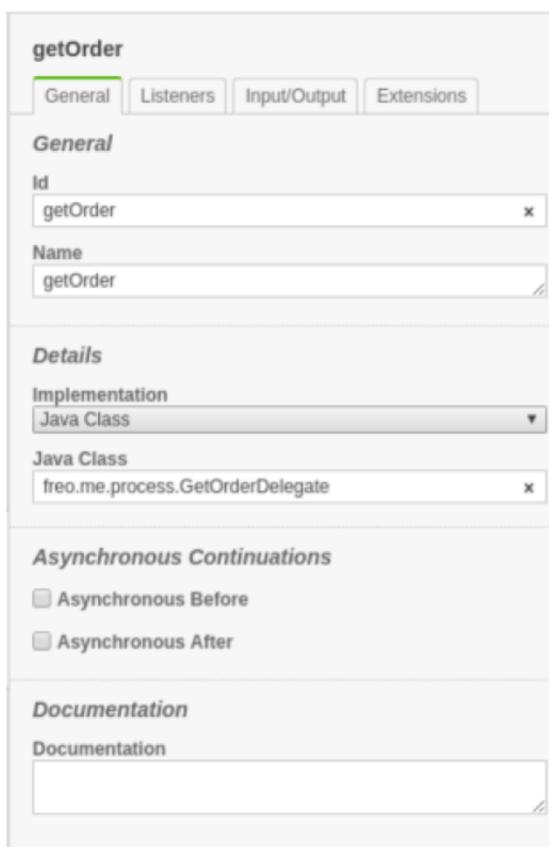
46. The getOrder task is a Service Task.



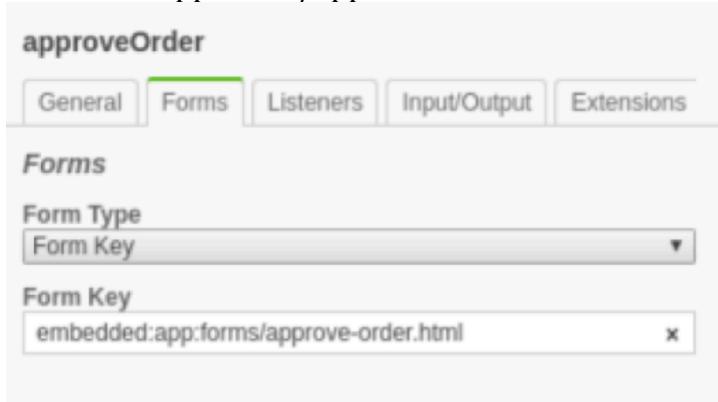
47. Go to the properties of the Start event and add a Form, with the Form Type is **Form Key** and the value is:
embedded:app:forms/start-approval.html



48. Edit the properties of the Service Task getOrder:
Implementation: Java Class
Java Class: freo.me.process.GetOrderDelegate



49. Add a form to the approveOrder User Task and reference the other form:
embedded:app:forms/approve-order.html



50. Now we should be ready to run this. **Save the BPMN**.

51. Rebuild the WAR file:

```
cd ~/bpmmn  
mvn clean install
```

52. Recopy the WAR to the Camunda directory:

```
cp ~/bpmmn/target/approve-order-0.1.0.war  
~/servers/camunda/server/apache-tomcat-8.0.24/webapps
```

53. Make sure your backend purchase service is up and running. This is the docker-compose service from Exercise 9¹.

You can check that the service is up and running by browsing:
<http://localhost/purchase>

You should see something like



54. Keep this window open as you are going to need that order id.

¹ If you didn't get this working, you can run the service using the following command lines:

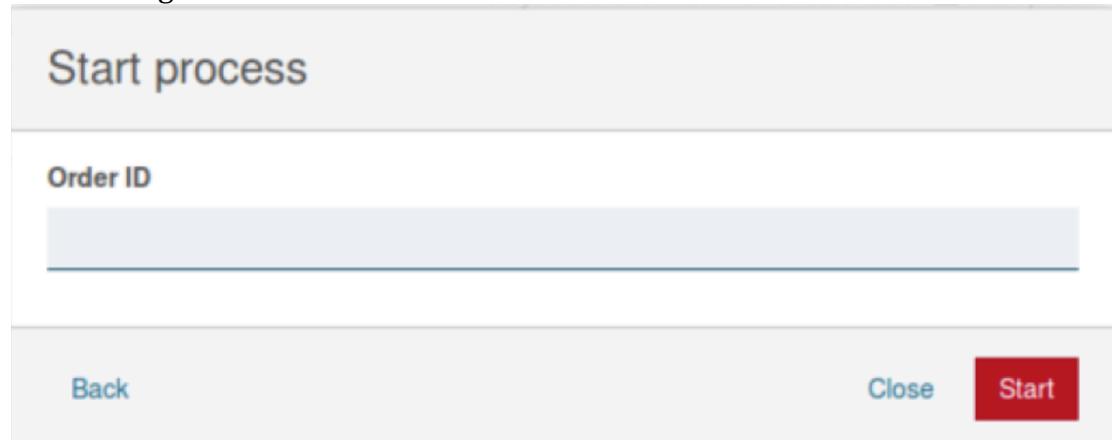
```
cd ~  
git clone https://github.com/pzfreo/P0ResourceComplete.git  
cd P0ResourceComplete  
gradle clean shadowJar  
sudo docker-compose up
```



55. If Camunda's Tomcat is already running it should have redeployed the process, otherwise start up the Tomcat again.

56. Go back to the Camunda tasklist

57. Now when you start the ApproveOrder process, you should see a new form asking for the id:



The screenshot shows a web-based form titled "Start process". It has a single input field labeled "Order ID" with a placeholder text area below it. At the bottom right are three buttons: "Back" (in blue), "Close" (in blue), and a large red button labeled "Start".

58. Copy and paste the id from the previous browser window and click **Start**

59. Once again you should be notified that you are the approver for this process instance. Now you can click to approve that.

60. You should see a form that has data collected from the REST service. You can click the **Approve** tickbox and then Complete.

61. Check out the Cockpit and the Admin windows (from the little House icon). There are a lot of features in the package that you would need for a real process management scenario (e.g. adding users, creating tenants, adding new approvers to existing processes, checking on the state of processes, etc).

62. That's all.

63. Extension

If you did the node.js/MongoDB Bonus exercise you will also have a catalogue to look up prices and a customer database. You could extend this process to create an invoice for the order.

