

Business Process Digitalization and Cloud Computing

14. Camunda Practice

Andrea Morichetta, Phd

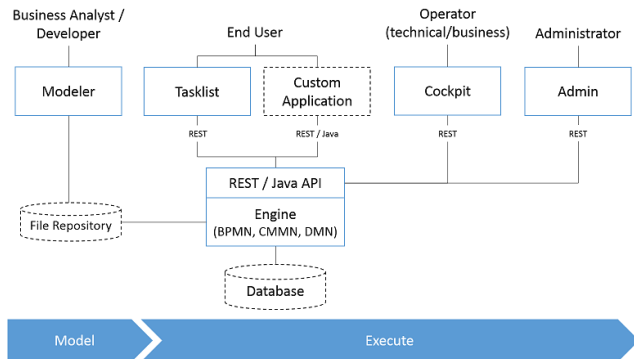
Computer Science Division

November 27, 2018



Introduction

- Camunda is a **Java-based framework** supporting BPMN for **workflow** and **process automation**
- CMMN for Case Management
- DMN for Business Decision Management

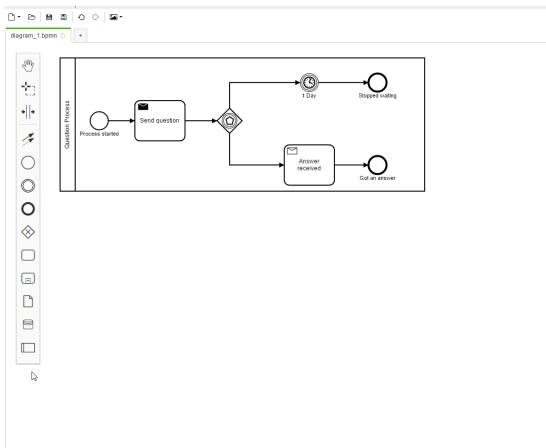


Process Engine

- The process engine is a **Java library** responsible for executing BPMN 2.0 processes, CMMN 1.1 cases and DMN 1.1 decisions
- Spring Framework Integration
- CDI/Java EE Integration
- Runtime Container Integration

Modeler

- Camunda Modeler is a **desktop application** for modeling BPMN, DMN and CMMN.
- It allows you to model files located directly on your local file system.
- **bpmn.io** the online version



BPMN 2.0 Properties for Execution

The screenshot displays the Camunda Modeler interface. The main workspace shows a BPMN diagram with the following elements:

- Start event (circle) leading to a task "Calculate Order Amount".
- Decision diamond with the expression "Amount greater than 50 dollars?".
- "Yes" path leading to a task "Ship Order Immediately".
- "No" path leading to a task "Collect Order in Batch".

The "Ship Order Immediately" task is selected, and its properties are shown in the right-hand panel, which is highlighted with a red border. The panel is titled "ServiceTask_0ipxatt" and contains the following sections:

- General:** Id: ServiceTask_0ipxatt
- Details:** Implementation Type: Java Class (selected)
- Listener:** Add Execution Listener (+)
- Documentation:** Documentation (empty text area)

Tasklist

- Camunda Tasklist is used to **organize work on tasks** that are supposed to be complete by an user.
- The **engine assigns those tasks to end users** as defined in the model.
- When end users click on a task in Camunda Tasklist, they will see a task form that displays information and asks them to enter data.

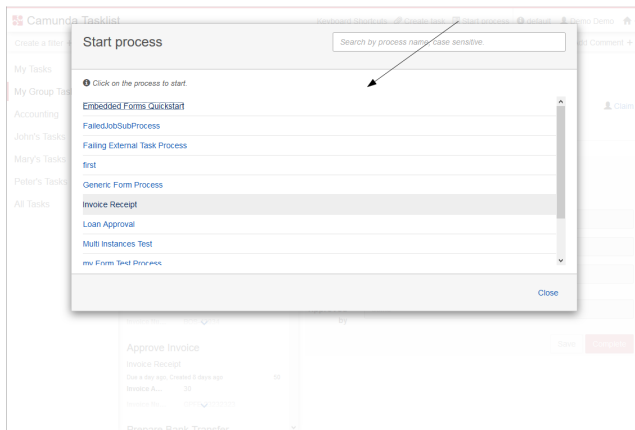
The screenshot displays the Camunda Tasklist web application. On the left, a sidebar shows a navigation menu with categories like 'My Tasks', 'My Group Tasks (7)', 'Accounting', 'John's Tasks', 'Mary's Tasks', 'Peter's Tasks', and 'All Tasks'. The main area shows a list of tasks, with the top one being 'Prepare Bank Transfer' (Invoice Receipt) due in 21 hours. The right-hand side of the interface shows the detailed task form for 'Prepare Bank Transfer'. The form includes a title 'Prepare Bank Transfer', a sub-header 'Invoice Receipt', and a 'Set follow-up' button set to 'In 21 hours'. Below this, there are tabs for 'Form', 'History', 'Diagram', and 'Description'. The main content area contains the instruction: 'Please prepare the bank transfer for the following invoice'. The form fields are: 'Invoice Document' (invoice.pdf), 'Creditor' (Bobby's Office Supplies), 'Amount' (900), 'Invoice Number' (BOS-43934), and 'Approved by' (demo). At the bottom right of the form are 'Save' and 'Complete' buttons.

Dashboard

The screenshot displays the Camunda Tasklist interface. On the left, a navigation sidebar lists task categories: My Tasks, My Group Tasks (7), Accounting, John's Tasks, Mary's Tasks, Peter's Tasks, and All Tasks. A 'Filters' callout points to the sidebar. The main task list shows several 'Prepare Bank Transfer' tasks, each with an 'Invoice Receipt' type, due dates, and values. A 'Filter results' callout points to the top of the task list. On the right, the 'Task view' for a 'Prepare Bank Transfer' task is shown. It includes an 'Invoice Receipt' header, a 'Set follow-up' button, and tabs for 'Form', 'History', 'Diagram', and 'Description'. The 'Form' tab is active, displaying fields for 'Invoice Document' (invoice.pdf), 'Creditor' (Bobby's Office Supplies), 'Amount' (900), 'Invoice Number' (BOS-43934), and 'Approved by' (demo). 'Save' and 'Complete' buttons are at the bottom right. A 'Task view' callout points to the task view header.

On the dashboard of Tasklist you see an overview of pending tasks.

Start a Process



To start a process instance via Tasklist, click on Start process in the header menu and select a process out of the displayed list of process definitions.

Task View

The screenshot displays the Camunda Tasklist interface. On the left, a sidebar shows a navigation menu with categories: 'My Tasks', 'My Group Tasks (7)', 'Accounting', 'John's Tasks', 'Mary's Tasks', 'Peter's Tasks', and 'All Tasks'. The 'My Group Tasks (7)' section is expanded, showing a list of tasks. The top task is 'Prepare Bank Transfer', which is selected and its details are shown in the main view.

The main view is titled 'Prepare Bank Transfer' and includes the following information:

- Invoice Receipt**
- Set follow-up ...** (with a dropdown menu set to 'in 21 hours')
- Accounting** (category)
- Claim** (action)
- Form** (selected tab), History, Diagram, Description

The task description reads: 'Please prepare the bank transfer for the following invoice'. Below this, there is a form with the following fields:

- Invoice Document**: invoice.pdf
- Creditor**: Bobby's Office Supplies
- Amount**: 900
- Invoice Number**: BOS-43934
- Approved by**: demo

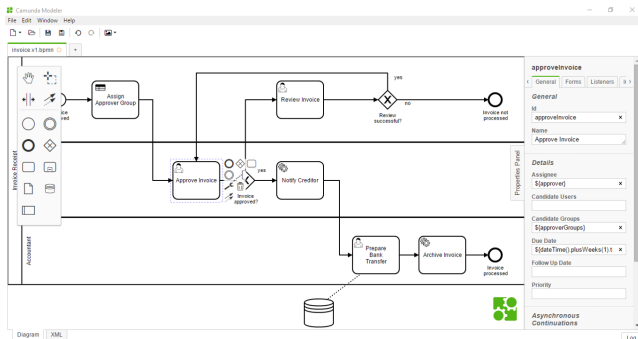
At the bottom right of the form, there are two buttons: 'Save' and 'Complete'.

On the right section of the dashboard, you can see the task view. Here you can work on tasks and perform the following operational actions.

Task Detail Tabs

The screenshot displays the Camunda Tasklist interface. On the left, a sidebar lists task categories: My Tasks, My Group Tasks (7), Accounting, John's Tasks, Mary's Tasks, Peter's Tasks, and All Tasks. The main area shows a list of tasks, with the top one selected: 'Prepare Bank Transfer' (Invoice Receipt, Due in 21 hours, Created 6 days ago, Invoice Amount 900, Invoice No. BOS-4384). Below this, other tasks like 'Approve Invoice' and another 'Prepare Bank Transfer' are visible. The right-hand pane shows the detailed view of the selected task. At the top, it says 'Prepare Bank Transfer' and 'Invoice Receipt'. Below that, there are options for 'Set follow-up ... in 21 hours' and 'Accounting Claim'. A tabbed interface is present with four tabs: 'Form', 'History', 'Diagram', and 'Description'. The 'Form' tab is active and highlighted with a red underline. A black arrow points to the 'Description' tab. The form content includes the instruction 'Please prepare the bank transfer for the following invoice' and a list of fields: 'Invoice Document' (invoice.pdf), 'Creditor' (Bobby's Office Supplies), 'Amount' (900), 'Invoice Number' (BOS-4384), and 'Approved by' (demo). At the bottom right of the form are 'Save' and 'Complete' buttons.

Implementing Initial Assignment



When a task is created, it is initially assigned to a user or group

Claiming a task in Tasklist

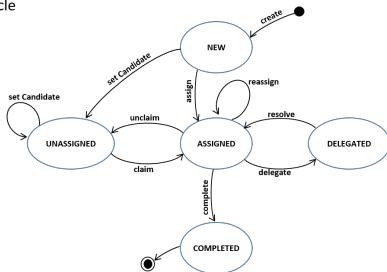
In Tasklist, a user can only work on a task if the task is assigned to that user.

This means that a user must claim a task if it is not yet assigned to him.

Task Lifecycle

The diagram shows the task lifecycle and supported transitions managed by Camunda BPM.

Task Lifecycle



Task Lifecycle States

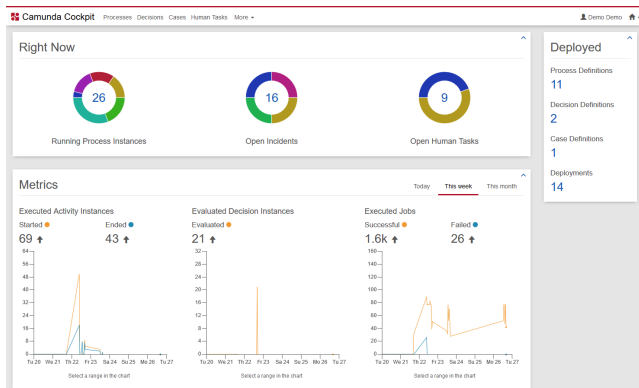
- New
- Unclaimed
- Claimed/Assigned
- Delegated
- Completed
- Deleted

Task Lifecycle Operations

- Create
- Set Candidate
- Claim
- Unclaim
- Assign
- Reassign
- Delegate
- Resolve
- Complete
- Delete
- Suspend

Camunda Cockpit

- A web application for process monitoring and operations that allows you to search for process instances, inspect their state and repair broken instances.



Process Definition View

The process definition view provides you with information about the definition and the status of a process.

The screenshot displays the Camunda Cockpit interface for the 'Invoice Receipt' process. The top navigation bar includes 'Processes', 'Decisions', 'Cases', 'Human Tasks', and 'More'. The breadcrumb trail is 'Dashboard > Processes > Invoice Receipt: Runtime | History'. The left sidebar contains metadata for the process definition, including version, ID, key, name, and tenant. The main area shows a BPMN diagram with tasks like 'Approve Invoice', 'Prepare Bank Transfer', and 'Archive Invoice', along with a decision diamond 'Invoice approved?'. Annotations include 'Instance counter' pointing to a task, and 'Incident indicator' pointing to a task. Below the diagram, the 'Process Instances' tab is active, showing a table of running instances with columns for State, ID, Start Time, and Business Key. A 'Select a filter' button is visible above the table.

Definition Metadata:

- Definition Version: 3
- Version Tag: null
- Definition ID: invoice:3:15b3c6e2-5cc7-11e7-88ba-...
- Definition Key: invoice
- Definition Name: Invoice Receipt
- History Time To Live: null
- Tenant ID: null
- Deployment ID: 15a8b11e-5cc7-11e7-88ba-606720b6...

Instances Running:

- current version: 8
- all versions: 12

Related:

- Reports
- Migration

State	ID	Start Time	Business Key
✓	38cf3386-5cc7-11e7-88ba-606720b6f99d	2017-06-29T14:33:57	
✓	161c4cb8-5cc7-11e7-88ba-606720b6f99d	2017-06-29T14:32:59	
✓	16bcd1ee-5cc7-11e7-88ba-606720b6f99d	2017-06-24T14:33:00	
✗	16a55308-5cc7-11e7-88ba-606720b6f99d	2017-06-15T14:33:00	
✓	811a3861-50f5-11e7-a286-606720b6f99d	2017-06-09T13:35:02	
✓	80cd5816-50f5-11e7-a286-606720b6f99d	2017-06-09T13:35:01	

Process Instance View

The screenshot displays the Camunda Cockpit interface for a process instance. The breadcrumb navigation shows: Dashboard > Processes > Comercases Process : 3dda40b3-6090-11e7-8c0d-9081835c525b : Runtime | History. The left sidebar contains a 'Filter' section with a tree view of process elements. The main area shows a BPMN diagram with a 'MESSAGE TASK STOP' activity highlighted. Below the diagram is a 'Variables' tab with a table of instance variables.

Name	Value	Type	Scope	Actions
nrOfInstances	5	Integer	MessageTask_2#m...	[Edit] [Delete]
nrOfCompleted...	0	Integer	MessageTask_2#m...	[Edit] [Delete]
nrOfActiveInsta...	5	Integer	MessageTask_2#m...	[Edit] [Delete]

This view allows you to drill down into a single process instance and **explore its running activities** as well as the variables, tasks, jobs, etc.

Deployment View

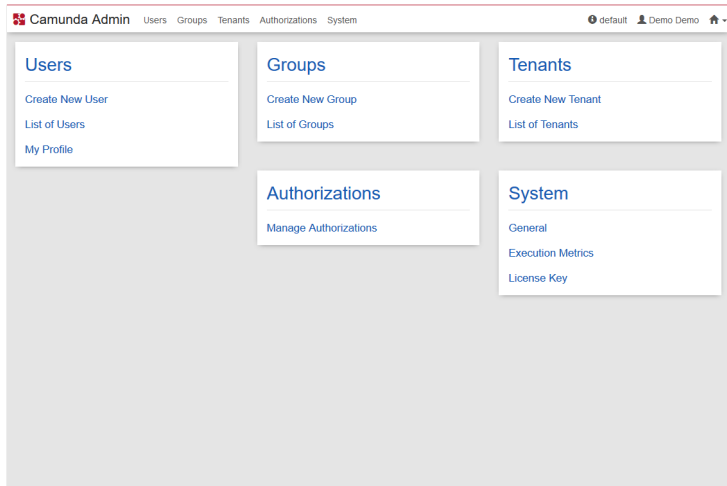
The screenshot displays the Camunda Cockpit interface. On the left, the 'Deployment Time' dropdown is set to 'Deploy'. Below it, a search bar contains 'third-test-processes'. A list of deployments is shown, with 'third-test-processes' selected. To the right of this list, a 'Search' box is highlighted. Below the deployment list, a 'Resources' section is visible, with 'failing-subprocess.bpmn' selected. A 'Deployments' box is also highlighted. The main area shows the BPMN diagram for 'failing-subprocess.bpmn' (Version: 1). The diagram includes a Start Event, a Service Task, a Subprocess, another Service Task, and an End Event. A 'Resource content' box is highlighted. Below the diagram, a 'Definitions' table is shown:

Name	Key	Instance Count
FailingSubProcess	FailingSubProcess	2

The deployment view of Cockpit shows an overview of all **deployments**, their **resources** and the content of these resources.

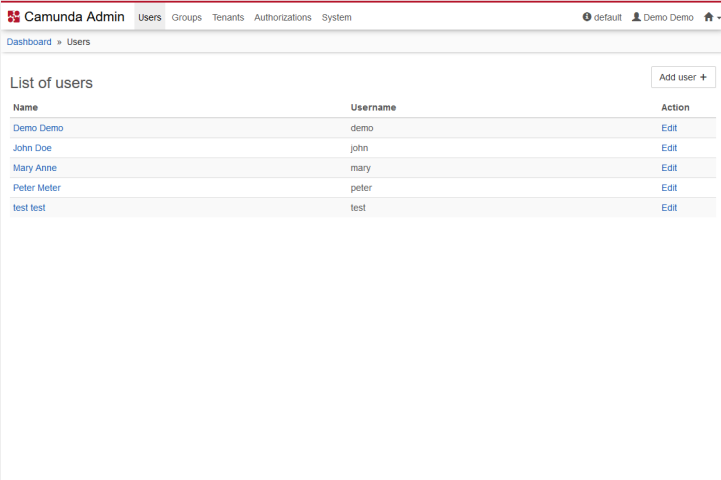
Camunda Admin

- A web application that allows you to manage users, groups and authorizations.



User Management

The Users menu allows you to **add, edit and delete user profiles**. Furthermore, you can manage group membership and change passwords.



The screenshot shows the Camunda Admin interface for user management. The top navigation bar includes 'Camunda Admin' and tabs for 'Users', 'Groups', 'Tenants', 'Authorizations', and 'System'. The current page is 'Users', with a breadcrumb 'Dashboard > Users'. A table titled 'List of users' displays a list of users with columns for Name, Username, and Action. An 'Add user +' button is located in the top right corner of the table area.

Name	Username	Action
Demo Demo	demo	Edit
John Doe	john	Edit
Mary Anne	mary	Edit
Peter Meter	peter	Edit
test test	test	Edit

Group Management

The Groups menu allows you to add, edit and delete user groups. Beside that you can view the members of groups

Camunda Admin Users Groups Tenants Authorizations System default Demo Demo

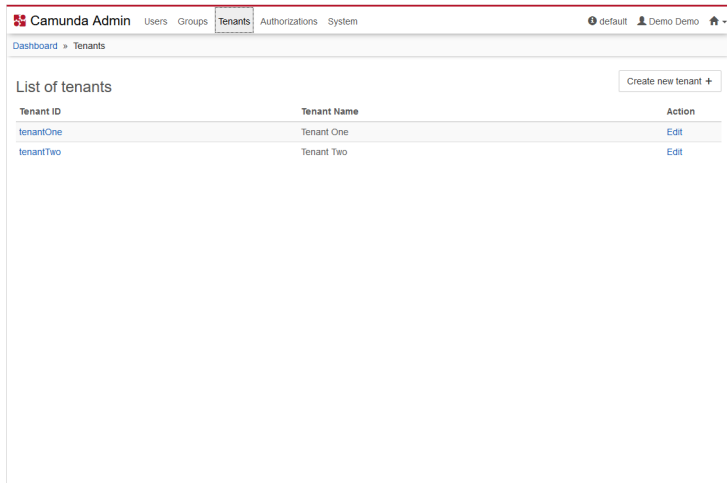
Dashboard > Groups

Create new group +

Group ID	Group Name	Group Type	Action
accounting	Accounting	WORKFLOW	Edit
camunda-admin	camunda BPM Administrators	SYSTEM	Edit
management	Management	WORKFLOW	Edit
sales	Sales	WORKFLOW	Edit

Tenant Management

The Tenants menu allows you to add, edit and delete tenants. Beside that you can view which user or group is a member of a tenant.



The screenshot shows the Camunda Admin web interface. The top navigation bar includes 'Camunda Admin', 'Users', 'Groups', 'Tenants' (highlighted), 'Authorizations', and 'System'. On the right, there are links for 'default', 'Demo Demo', and a home icon. Below the navigation, a breadcrumb trail shows 'Dashboard > Tenants'. The main content area is titled 'List of tenants' and features a 'Create new tenant +' button. A table lists two tenants:

Tenant ID	Tenant Name	Action
tenantOne	Tenant One	Edit
tenantTwo	Tenant Two	Edit

Authorization Management

Manage **authorizations** for a variety of resources (e.g., Applications, Groups, Filters).

The screenshot displays the Camunda Admin interface for managing authorizations. The top navigation bar includes 'Camunda Admin', 'Users', 'Groups', 'Tenants', 'Authorizations', and 'System'. The user 'Jonny Prosciutto' is logged in. The breadcrumb trail shows 'Dashboard > Authorizations'. A sidebar on the left lists various management options, with 'Application' selected. The main content area is titled 'Application Authorizations' and features a 'Create new authorization +' button. Below this is a table with the following data:

Type	User / Group	Permissions	Resource ID	Action
ALLOW	accounting	ACCESS	tasklist	Edit Delete
ALLOW	camunda-admin	ALL	*	Edit Delete
ALLOW	management	ACCESS	tasklist	Edit Delete
ALLOW	sales	ACCESS	tasklist	Edit Delete

At the bottom of the interface, it states: 'Date and Time displayed in local timezone: Europe/Berlin' and 'Powered by camunda BPM / v\$(project.version)'.

Authorization Management

- **Application Access:** Set the authorizations for the new group and the created users
- **Filter Access:** The filter can set authorization in the tasklist
- **Member Visibility:** Depending on the users authorization, Tasklist will show you information about your colleagues and groups.
- **Application-Specific Permissions:** It is possible to set permission for every applications and specific instances

Camunda in Practice

Prerequisites

Before downloading Camunda, make sure you have a JRE (Java Runtime Environment), or better, a **JDK** (Java Development Kit) installed. The Java 8 is recommended.

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Download the Runtime

- Download a distribution of the Camunda BPM platform.
- You can choose from different distributions for various application servers. In this tutorial, we will use the **Apache Tomcat-based distribution**.

[Download Camunda from here](#)

- Execute the script named start-camunda.bat (for Windows users) or start-camunda.sh (for Unix users).
- This script will start the application server and open a welcome screen in your web browser or go to `http://localhost:8080/camunda-welcome/index.html`.

Camunda External Task

Create a new BPMN Diagram

1. Select activity
2. Click wrench icon and change type to Service Task

Payment Retrieval Requested

Charge Credit Card

- Send Task
- Receive Task
- User Task
- Manual Task
- Business Rule Task
- Service Task**
- Script Task
- Call Activity
- Sub Process (collapsed)
- Sub Process (expanded)

Properties Panel

General

Id: Task_192t1nu

Name: Charge Credit Card

Asynchronous Continuations

Asynchronous Before

Asynchronous After

Documentation

Element Documentation

Diagram XML

Log

Configure the Service Task

The screenshot shows the Camunda Modeler interface. In the center, a BPMN diagram titled 'diagram_4.bpmn' is displayed. It features a 'Payment Retrieval Requested' start event leading to a 'Charge Credit Card' service task, which is currently selected and highlighted with a dashed blue border. To the right of the task, there are several icons representing different types of events and connectors. On the right side of the interface, the 'Properties Panel' is open, showing the configuration for the selected 'Task_19pt1nu'. The panel has tabs for 'General', 'Listeners', and 'Input/Output'. The 'General' tab is active, displaying the following configuration:

- General**
 - Id: Task_19pt1nu
 - Name: Charge Credit Card
- Details**
 - Implementation: External
 - Topic: charge-card
- External Task Configuration**
 - Task Priority: [empty field]
- Asynchronous Continuations**
 - Asynchronous Before
 - Asynchronous After
- Documentation**
 - Element Documentation: [empty field]

A red text box with a white border is overlaid on the diagram, containing the text: "If it's not already visible, click on the Properties Panel tab". A red arrow points from this text box to the 'Properties Panel' tab icon in the top right corner of the interface.

Configure Properties for Process

The screenshot displays the Camunda Modeler interface. On the left, a BPMN diagram titled 'diagram_4.bpmn' shows a process flow: a start event 'Payment Retrieval Requested' leads to a task 'Charge Credit Card' (indicated by a gear icon), which then leads to an end event 'Payment Received'. A vertical toolbar on the left contains various BPMN symbols. The right-hand side features a configuration panel for the selected process, titled 'payment-retrieval'. This panel has tabs for 'General', 'Listeners', and 'Extensions'. The 'General' tab is active and contains the following fields:

- Id:** 'payment-retrieval' (with a dropdown arrow). A red callout box 'Set Process Id' points to this field.
- Name:** 'Payment Retrieval' (with a dropdown arrow). A red callout box 'Set Process Name' points to this field.
- Version Tag:** (empty text field).
- Executable:** A checked checkbox. A red callout box 'Mark the process to be executable' points to this checkbox.
- External Task Configuration:** A section with a 'Task Priority' text field.
- Job Configuration:** A section with a 'Job Priority' text field.
- Candidate Starter Configuration:** A section with a 'Candidate Starter Groups' text field, a note 'Specify more than one group as a comma separated list.', and a 'Candidate Starter Users' text field, also with a note 'Specify more than one user as a comma separated list.'
- History Configuration:** A section with a 'Log' button.

At the bottom of the interface, there are 'Diagram' and 'XML' buttons, and a 'Log' button in the bottom right corner of the configuration panel.

Implement an External Task Worker

Prerequisites

Make sure you have the following tools installed:

- JDK 1.8
- An IDE for Java projects (e.g. Eclipse)

Create a New Maven Project

New Maven Project
Configure project

Artifact

Group Id:

Artifact Id:

Version:

Packaging:

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:

▶ **Advanced**

Add Camunda External Task Client Dependency

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd" >
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.camunda.bpm.getstarted</groupId>
  <artifactId>charge-card-worker</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <dependencies>
    <dependency>
      <groupId>org.camunda.bpm</groupId>
      <artifactId>camunda-external-task-client</artifactId>
      <version>1.0.0</version>
    </dependency>
  </dependencies>
</project>
```


Add the Java Class

- Create a new package *org.camunda.bpm.getstarted.chargecard*
- Add the Java Class add a Java class, e.g. *ChargeCardWorker*

```
import java.util.logging.Logger;
import org.camunda.bpm.client.ExternalTaskClient;

public class ChargeCardWorker {
    private final static Logger LOGGER = Logger.getLogger(ChargeCardWorker.class.getName());

    public static void main(String[] args) {
        ExternalTaskClient client = ExternalTaskClient.create()
            .baseUrl("http://localhost:8080/engine-rest")
            .build();

        // subscribe to an external task topic as specified in the process
        client.subscribe("charge-card")
            .lockDuration(1000) // the default lock duration is 20 seconds, but you can override this
            .handler((externalTask, externalTaskService) -> {
                // Put your business logic here

                // Get a process variable
                String item = (String)externalTask.getVariable("item");
                Long amount = (Long) externalTask.getVariable("amount");
                LOGGER.info("Charging credit card with an amount of " + amount + " for the item " + item + "...");

                // Complete the task
                externalTaskService.complete(externalTask);
            })
            .open();
    }
}
```

Run the Worker

Errors

In case of errors please make sure that the compiler is 1.8 for the jdk in windows – > preferences – > java – > compiler – >configure project specific settings

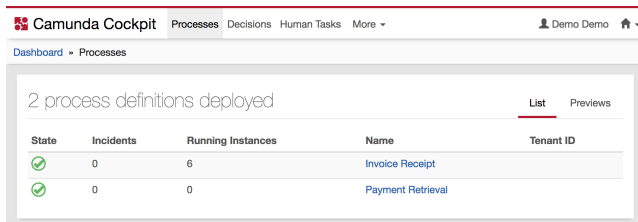
You can run the Java application by right clicking on the class ChargeCardWorker and choosing Run as Java.

Deploy the Process "Payment Retrieval"

The image displays two screenshots of the Camunda Modeler interface. The top screenshot shows a BPMN diagram with three elements: 'Payment Retrieval Requested' (start event), 'Charge Credit Card' (task), and 'Payment Received' (end event). A red arrow points to a 'Deploy Current Diagram' button in the top right, which is labeled 'Deploy button'. The right-hand panel shows configuration options for the 'payment-retrieval' process, including 'General', 'Listeners', and 'Extensions' tabs, and fields for 'Id', 'Name', 'Version Tag', 'Executable', 'External Task Configuration', and 'Job Configuration'. The bottom screenshot shows the same diagram with a 'Deploy Diagram' dialog box open. The dialog contains the text 'Specify deployment details and deploy this diagram to Camunda.' and a 'Deployment Name' field with the value 'Payment Retrieval'. There is also a 'Tenant Id' field and 'Deploy' and 'Cancel' buttons.

Verify the Deployment with Cockpit

- Go to `http://localhost:8080/camunda/app/cockpit`



The screenshot displays the Camunda Cockpit interface. At the top, the navigation bar includes the Camunda logo, the text "Camunda Cockpit", and tabs for "Processes", "Decisions", "Human Tasks", and "More". On the right side of the navigation bar, there is a user profile icon labeled "Demo Demo" and a home icon. Below the navigation bar, the breadcrumb "Dashboard » Processes" is visible. The main content area features a summary card that says "2 process definitions deployed" with two tabs: "List" (which is selected and underlined) and "Previews". Below this summary is a table with the following data:

State	Incidents	Running Instances	Name	Tenant ID
✓	0	6	Invoice Receipt	
✓	0	0	Payment Retrieval	

Start a Process Instance

REST Client: Make a POST request to the following endpoint:

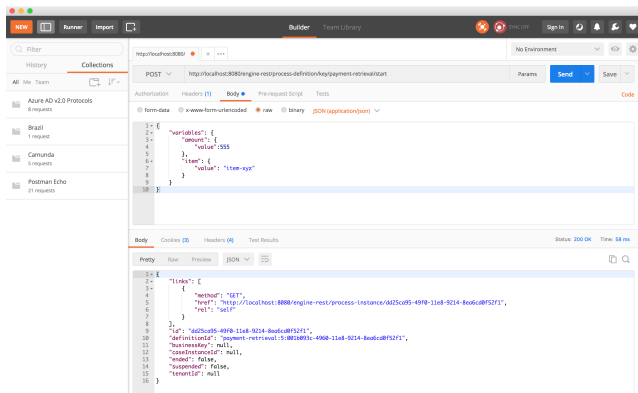
`http://localhost:8080/engine-rest/process-definition/key/
payment-retrieval/start`

The JSON Body should look like this:

```
{
  "variables": {
    "amount": {
      "value": 555,
      "type": "long"
    },
    "item": {
      "value": "item-xyz"
    }
  }
}
```

Start a Process Instance

Make sure you are setting the headers correctly to **Content-Type: application/json**



The screenshot shows the Postman interface with a REST client request and its response. The request is a POST to `http://localhost:8080/engine-rest/process-definition/key/payment-retrieval/start`. The body is a JSON object with variables and an item.

```
1 = {
2   "variables": {
3     "amount": {
4       "value": "555"
5     },
6     "item": {
7       "value": "item-xyz"
8     }
9   }
10 }
```

The response is a JSON object with links and instance details.

```
1 = {
2   "links": [
3     {
4       "method": "GET",
5       "href": "http://localhost:8080/engine-rest/process-instance/625c95-49f8-11e8-9214-8e6ca8f52f1",
6       "rel": "self"
7     }
8   ],
9   "id": "625c95-49f8-11e8-9214-8e6ca8f52f1",
10  "definitionId": "payment-retrieval:5:001e093c-4968-11e8-9214-8e6ca8f52f1",
11  "businessKey": null,
12  "caseInstanceId": null,
13  "ended": false,
14  "suspended": false,
15  "tenantId": null
16 }
```

Add a User Task

The screenshot shows the Camunda Modeler interface. The main workspace displays a BPMN diagram with the following elements:

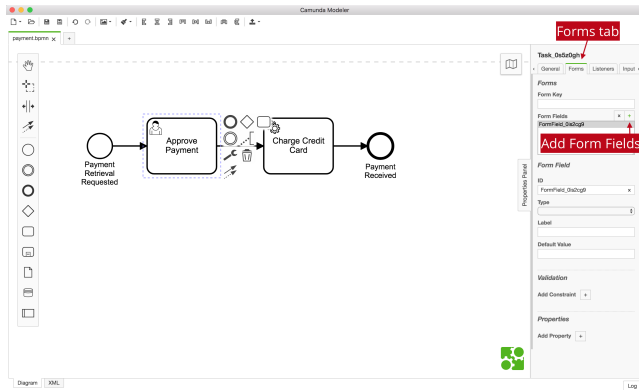
- Start event: Payment Retrieval Requested
- Task: Approve Payment (highlighted with a blue dashed border)
- Exclusive Gateway
- Task: Charge Credit Card
- End event: Payment Received

The right-hand panel is titled "Task_Dedupgh" and contains the following configuration options:

- General**
 - Id: task_Dedupgh
 - Name: Approve Payment
- Details**
 - Assignee: alexis (highlighted with a red box and arrow, with the text "Set demo as Assignee" next to it)
 - Candidate Users: [empty]
 - Candidate Groups: [empty]
- Due Date**
 - [empty]
 - The due date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-28T08:04:00)
- Follow Up Date**
 - [empty]
 - The follow up date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-06-28T08:04:00)
- Priority**
 - [empty]
- Asynchronous Continuations**
 - Asynchronous Before
 - Asynchronous After

At the bottom of the panel, there is a "Log" button and a green icon.

Configure a basic form in the User Task



● Field1

- ▶ ID: amount
- ▶ Type: long
- ▶ Label: Amount

● Field2

- ▶ ID: item
- ▶ Type: string
- ▶ Label: Item

● Field3

- ▶ ID: approved
- ▶ Type: boolean
- ▶ Label: Approved?

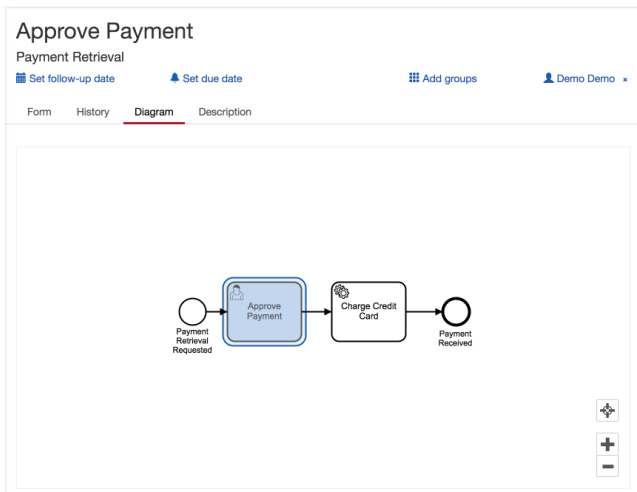
Work on the Task

- After the deploy of the process "*Payment Retrieval*"
- Go to Tasklist <http://localhost:8080/camunda/app/tasklist>
- Click on start process

The screenshot shows the 'Start process' dialog box in the Camunda Tasklist application. The dialog has a title bar 'Camunda Tasklist' and a subtitle 'Start process'. Below the title bar, there is a message: 'You can set variables, using a generic form, by clicking the "Add a variable" link below.' The main area contains a 'Business Key' input field and a table for adding variables. The table has columns for 'Name', 'Type', and 'Value'. Two variables are listed: 'amount' of type 'Long' with value '555', and 'item' of type 'String' with value 'item-xyz'. At the bottom of the dialog, there are 'Back', 'Close', and 'Start' buttons.

Add a varia... +	Name	Type	Value
Rem... x	amount	Long	555
Rem... x	item	String	item-xyz





Approve Payment in the Task List



Approve Payment in the Task List

Approve Payment

Payment Retrieval

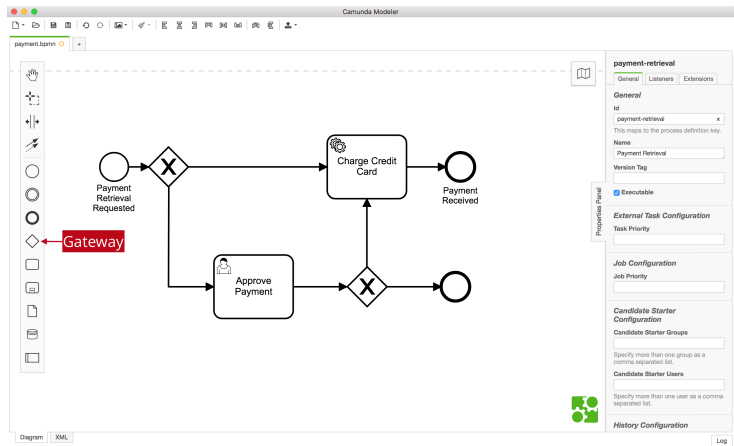
 Set follow-up date  Set due date  Add groups  Demo Demo ▾

Form History Diagram Description

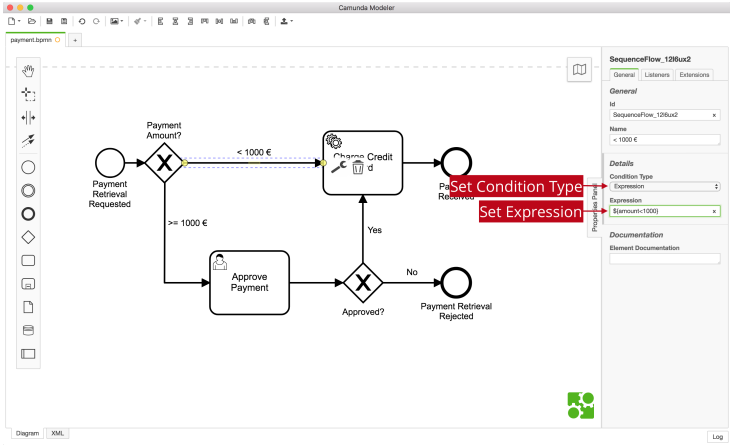
Amount

Item
Approve?

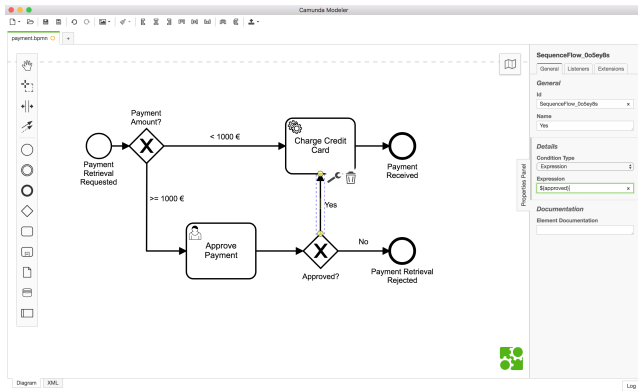
Add Two Gateways



Configure the Gateways



Configure the Gateways



Work on the Task

- After the deploy of the process *"Payment Retrieval"*
- Go to Tasklist <http://localhost:8080/camunda/app/tasklist>
- Click on start process
- make sure you use an amount that is larger or equal to 1000 in order to see the User Task Approve Payment

Camunda Tasklist

Keyboard Shortcuts Create task Start process Demo Demo

Start process

Business Key

Add a variable...

	Name	Type	Value
Rem... x	<input type="text" value="amount"/>	<input type="text" value="Long"/>	<input type="text" value="555"/>
Rem... x	<input type="text" value="item"/>	<input type="text" value="String"/>	<input type="text" value="item-xyz"/>

Back Close Start

Add a Business Rule Task to the Process

The screenshot shows the Camunda Modeler interface. The main workspace displays a BPMN diagram for a process named 'payment.bpmn'. The diagram starts with a start event 'Payment Retrieval Requested', followed by a decision diamond 'Payment Amount?'. The decision has two outgoing flows: one labeled '< 1000 €' and another labeled '>= 1000 €'. The '>= 1000 €' flow leads to a task 'Approve Payment'. A context menu is open over the 'Approve Payment' task, listing various task types. The 'Business Rule Task' option is highlighted in yellow. The right-hand side of the interface shows the 'Properties Panel' for the selected task, 'Task_0s5z0gh'. The 'General' tab is active, showing fields for 'Id' (Task_0s5z0gh), 'Name' (Approve Payment), 'Assignee' (demo), and 'Due Date'. The 'Asynchronous Continuations' section is also visible.

Camunda Modeler

payment.bpmn x

Payment Retrieval Requested

Payment Amount?

< 1000 €

>= 1000 €

Approve Payment

- Task
- Send Task
- Receive Task
- Manual Task
- Business Rule Task**
- Service Task
- Script Task
- Call Activity
- Sub Process (collapsed)
- Sub Process (expanded)

Task_0s5z0gh

General Forms Listeners Input

General

Id Task_0s5z0gh

Name Approve Payment

Details

Assignee demo

Candidate Users

Candidate Groups

Due Date

The due date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-08-26T09:54:00)

Follow Up Date

The follow up date as an EL expression (e.g. \${someDate}) or an ISO date (e.g. 2015-08-26T09:54:00)

Priority

Asynchronous Continuations

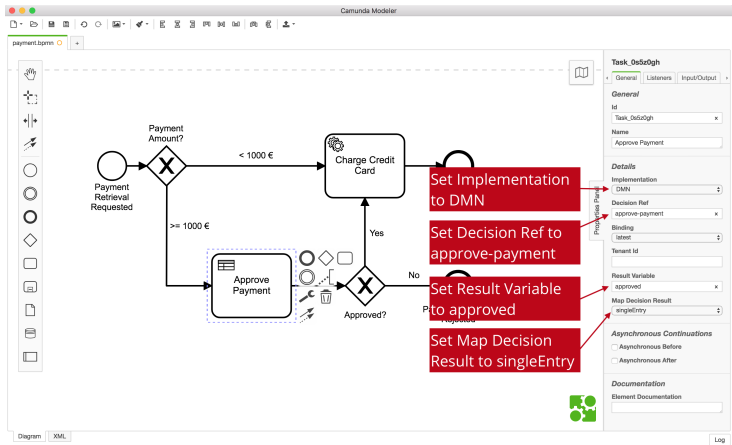
Asynchronous Before

Asynchronous After

Diagram XML

Log

Configure the Business Rule Task



Create a DMN table

The screenshot shows the Camunda Modeler interface. At the top, the title bar reads "Camunda Modeler". Below it, there are tabs for "payment.bpmn" and "diagram_1.dmn". The main workspace displays a BPMN task named "Approve Payment" with a sub-label "approve-payment". To the right of the task is a "View DRD" button. Below the task, a DMN table is visible. The table has three columns: "Input", "Output", and "Annotation". The "Input" column contains a single row with the value "string". The "Output" column contains a single row with the value "string". The "Annotation" column is empty. At the bottom left, there is a "Decision Table" button and a "XXL" button. At the bottom right, there is a "Log" button.

U	Input +	Output +	Annotation
-	-	-	
	string	string	
+ -	-	-	-

DMN Table

or the input column, use item as the Input Expression and Item as the Input Label:

The screenshot shows the Camunda Modeler interface for editing a Decision Model Notation (DMN) table. The table is titled "Approve Payment" and is currently in edit mode. The table structure is as follows:

U	Input +	Output +	Annotation
	Item	Input Expression	
	string	Item	
		Enter simple FEEL expression or change to script.	
		Input Label: <input type="text" value="Item"/>	
		Input Variable: <input type="text" value="collInput"/>	

At the bottom of the interface, there are buttons for "Decision Table", "XXL", and "Log".

DMN Table

Use approved as the Output Name and Approved as the Output Label for the output column Approved

The screenshot shows the Camunda Modeler interface with a DMN table for the decision 'Approve Payment'. The table is structured as follows:

U	Input +	Output +	Annotation
	Output Name approved	Approved	
	Output Label Approved	string	
+	-	-	-

At the bottom of the interface, there are tabs for 'Decision Table' and 'XXL', and a 'Log' button in the bottom right corner.

DMN Table

The screenshot shows the Camunda Modeler interface for editing a Decision Model Notation (DMN) table. The table is titled "Approve Payment" and is named "approve-payment". It is currently in "U" (Unapproved) state. The table has two columns: "Input" and "Output". The "Input" column has two rows: "1" with value "Item-xyz" and "2" with value "Item-zzz". The "Output" column has one row with value "Approved". The "Annotation" column is empty. A dropdown menu is open over the "Type" field, showing a list of data types: string, boolean, integer, long, double, and date. The "string" type is selected. The interface also includes a "View DRD" button, a "Decision Table" tab, and a "Log" button.

U	Input	Output	Annotation
	Item	Approved	
	st:	string	
1	"Item-xyz"		
2	"Item-zzz"		
+			

Deploy the DMN Table

The screenshot shows the Camunda Modeler interface with a 'Deploy Diagram' dialog box open. The dialog contains the following text and fields:

Deploy Diagram
Specify deployment details and deploy this diagram to Camunda.

Deployment Name

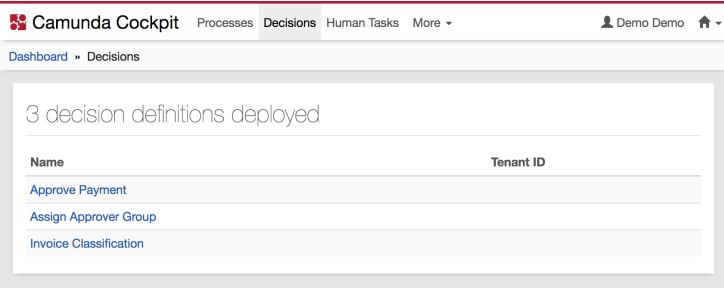
Tenant Id

In the background, the 'Approve Payment' decision table is visible. A red box in the top toolbar highlights the 'Deploy' button.

	Input +
U	Item
	string
1	"Item-xyz"
2	not("Item-xyz")
+	-

Verify the Deployment with Cockpit

`http://localhost:8080/camunda/app/cockpit`



The screenshot shows the Camunda Cockpit interface. The top navigation bar includes the Camunda logo, the text "Camunda Cockpit", and menu items for "Processes", "Decisions", "Human Tasks", and "More". On the right, there is a user profile "Demo Demo" and a home icon. Below the navigation bar, the breadcrumb "Dashboard » Decisions" is visible. The main content area displays the message "3 decision definitions deployed" above a table. The table has two columns: "Name" and "Tenant ID". Three decision definitions are listed in the table:

Name	Tenant ID
Approve Payment	
Assign Approver Group	
Invoice Classification	

Start the Process

Camunda Tasklist Keyboard Shortcuts Create task Start process Demo Demo Home

Create a filter

My Tasks (3)

My Group T

Accounting

John's Task

Mary's Task

Peter's Task

All Tasks

Start process

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

Business Key

Add a varia... +	Name	Type	Value
Rem... x	<input type="text" value="item"/>	<input type="text" value="String"/>	<input type="text" value="item-zzz"/>
Rem... x	<input type="text" value="amount"/>	<input type="text" value="Long"/>	<input type="text" value="1200"/>

[Back](#) Close Start

Assign Reviewer

Inspect using Cockpit and Tasklist

The screenshot displays the Camunda Cockpit interface for a decision task named "Approve Payment". The breadcrumb navigation shows the path: Dashboard > Decisions > Approve Payment > 919e5322-529f-11e8-9724-d2c19956f38d. The left sidebar contains metadata for the instance, including Instance ID, Definition Version, Definition ID, Definition Key, Definition Name, Tenant ID, Deployment ID, Process Instance ID, Case Instance ID, and Decision Requirements.

The main content area features a decision table titled "Approve Payment" with the definition key "approve-payment". The table has four columns: "U" (User), "Input", "Output", and "Annotation".

U	Input	Output	Annotation
	Item	Approved	
	string	boolean	
1	"item-xyz"	true	-
2	not("item-xyz")	false	-

Below the decision table, there are tabs for "Inputs" and "Outputs". The "Inputs" tab is active, showing a table with columns "Name", "Type", and "Value".

Name	Type	Value
Item	String	item-zzz

You'll see that depending on the input, the worker will either charge or not charge the credit card.

Questions?