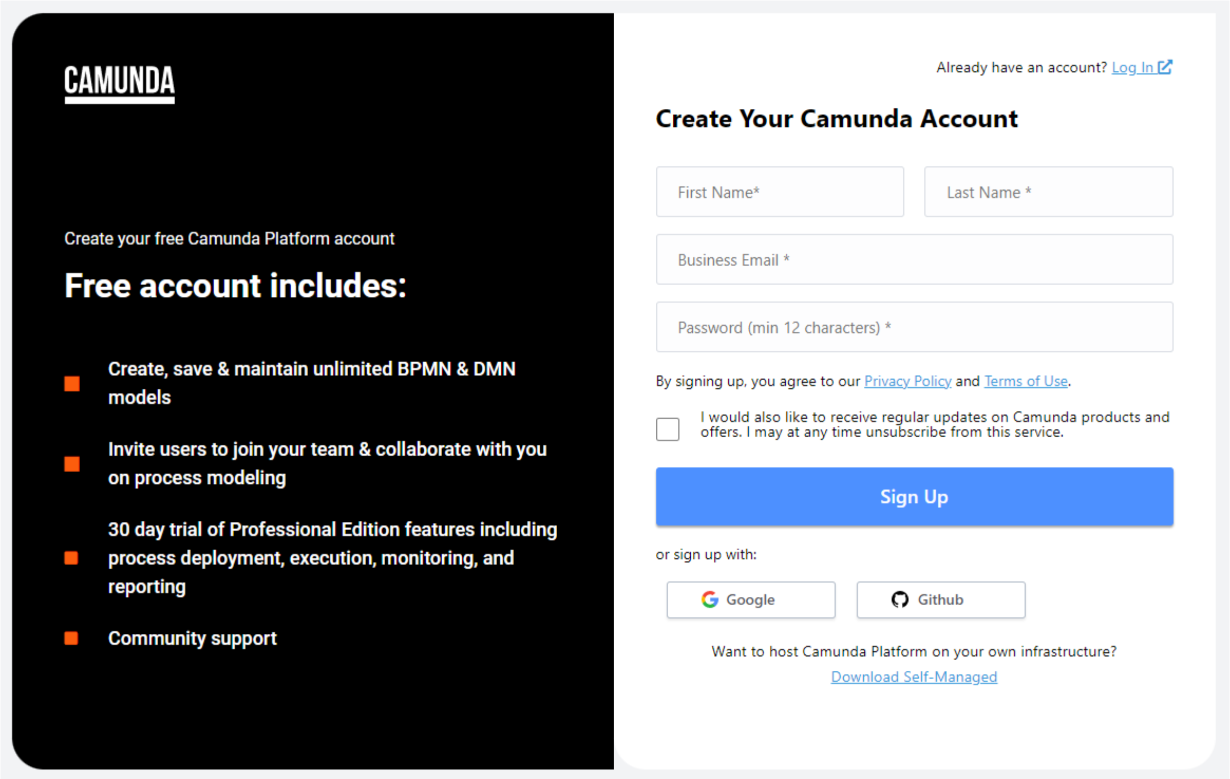
Camunda Cloud workshop instruction

Table of content

I Camunda Cloud setup

1. Create a free account on Camunda Cloud.
   a. go to https://camunda.com/platform/ and click on **Try Free**
   b. provide your details and click on *Sign Up* - page should look like below:



Create your free Camunda Platform account

Free account includes:

- Create, save & maintain unlimited BPMN & DMN models
- Invite users to join your team & collaborate with you on process modeling
- 30 day trial of Professional Edition features including process deployment, execution, monitoring, and reporting
- Community support

Already have an account? Log In

Create Your Camunda Account

By signing up, you agree to our Privacy Policy and Terms of Use.

□ I would also like to receive regular updates on Camunda products and offers. I may at any time unsubscribe from this service.

or sign up with:

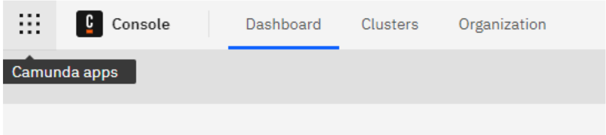
Want to host Camunda Platform on your own infrastructure?
Download Self-Managed

   c. You should receive confirmation on your email. Check your email box and verify your account.
   d. Log in to Camunda Cloud - open https://camunda.com/platform/ and click on Log In.

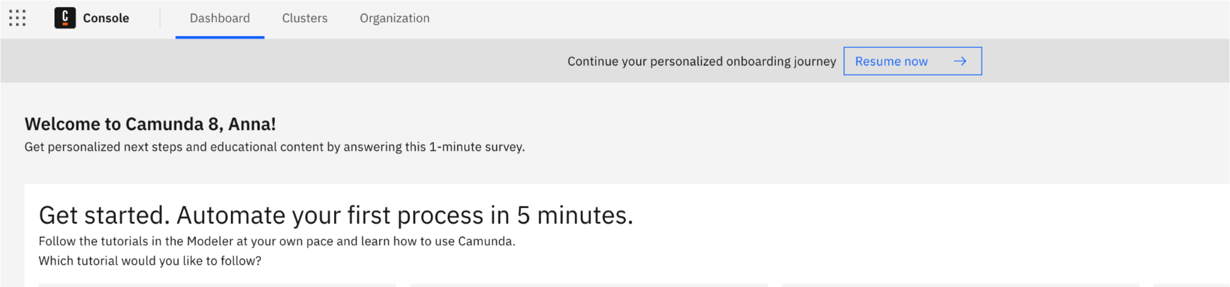
2. Create a cluster on Camunda Cloud.
   a. Once you are logged in, you should see the Console view of Camunda. Make sure you see

in the top left corner.

If not, click on Camunda apps in the top left corner and select **Console:**



b. In the menu on the top select **Clusters**:



Continue your personalized onboarding journey

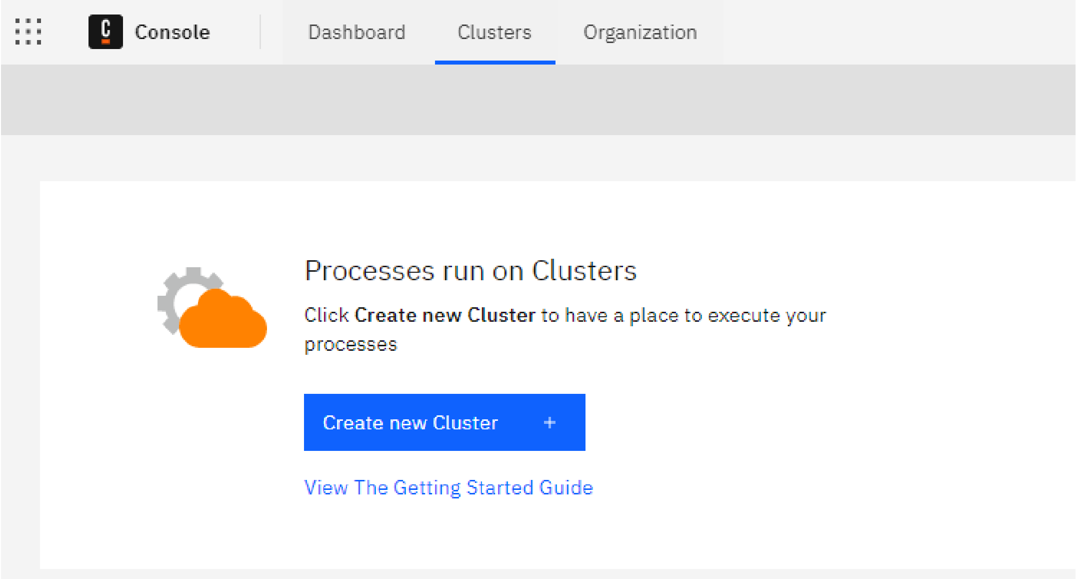
Welcome to Camunda 8, Anna!

Get personalized next steps and educational content by answering this 1-minute survey.

Get started. Automate your first process in 5 minutes.

Follow the tutorials in the Modeler at your own pace and learn how to use Camunda.
Which tutorial would you like to follow?

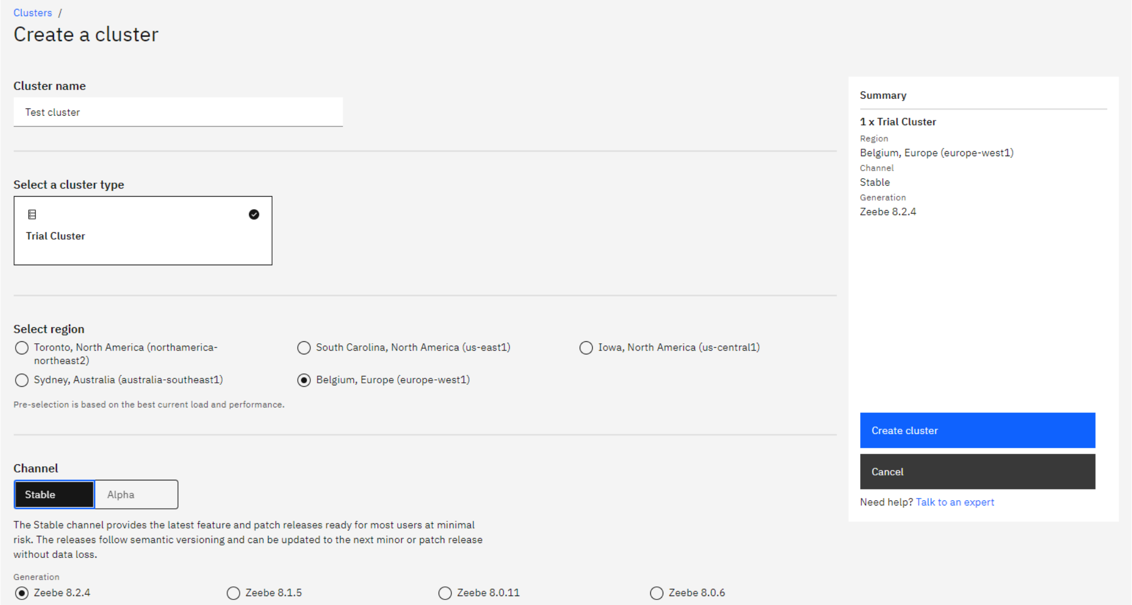
and then click on **Create new cluster:**



Processes run on Clusters

Click **Create new Cluster** to have a place to execute your processes

c. Provide all details for your cluster as below and then click on Create cluster



Create a cluster

Cluster name: Test cluster

Select a cluster type: Trial Cluster ✓

Select region
○ Toronto, North America (northamerica-northeast2)
○ South Carolina, North America (us-east1)
○ Iowa, North America (us-central1)
○ Sydney, Australia (australia-southeast1)
● Belgium, Europe (europe-west1)

Pre-selection is based on the best current load and performance.

Channel: **Stable** | Alpha

The Stable channel provides the latest feature and patch releases ready for most users at minimal risk. The releases follow semantic versioning and can be updated to the next minor or patch release without data loss.

Generation
● Zeebe 8.2.4 ○ Zeebe 8.1.5 ○ Zeebe 8.0.11 ○ Zeebe 8.0.6

Summary

1 x Trial Cluster

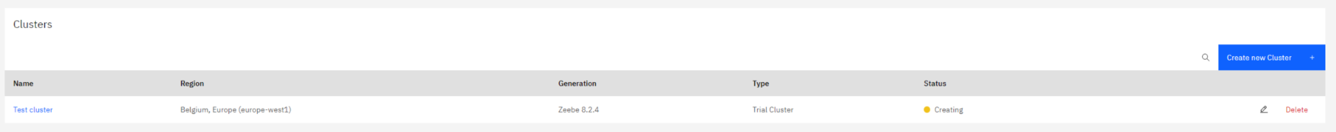
Region
Belgium, Europe (europe-west1)

Channel
Stable

Generation
Zeebe 8.2.4

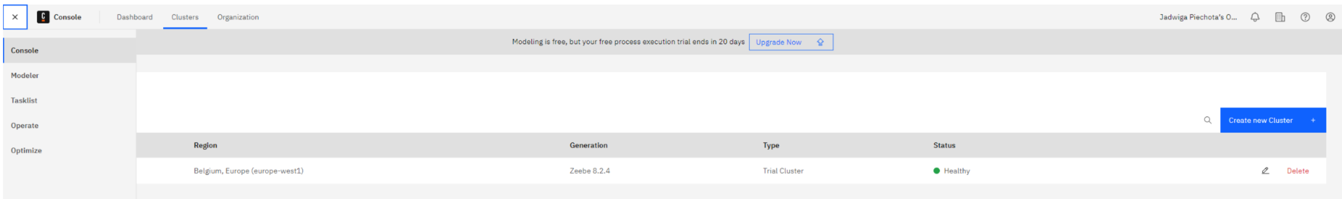
Need help? Talk to an expert

d. You should be able to see you cluster in with Creating status:



Name	Region	Generation	Type	Status		
Test cluster	Belgium, Europe (europe-west1)	Zeebe 8.2.4	Trial Cluster	● Creating		Delete

After a few minutes the cluster should be provisioned. The status will look like this:

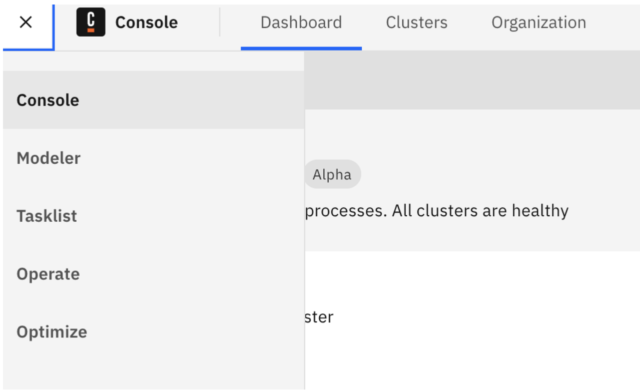


Modeling is free, but your free process execution trial ends in 20 days

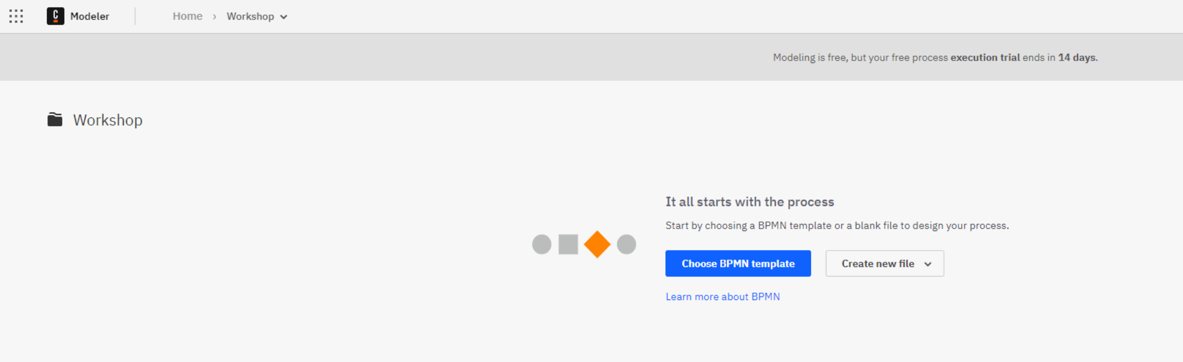
Region	Generation	Type	Status		
Belgium, Europe (europe-west1)	Zeebe 8.2.4	Trial Cluster	● Healthy		Delete

3. Create a project.

a. Go to the Modeler - click on Camunda apps on the top left corner and choose Modeler:



b. When You open Modeler, click on New project. When new project is opened, name the project - at the top bar (exemplary name can be workshop as on below picture):

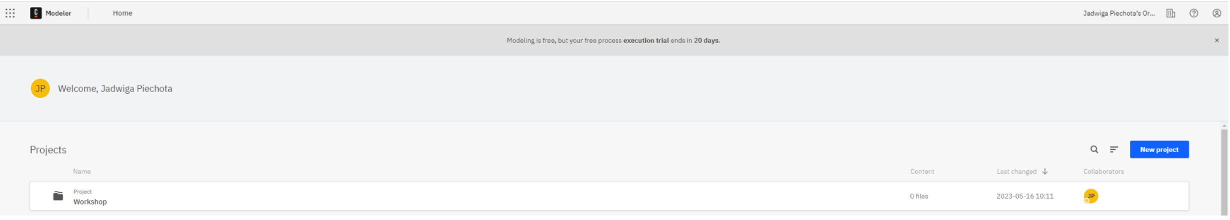


Modeling is free, but your free process **execution trial** ends in **14 days.**

It all starts with the process

Start by choosing a BPMN template or a blank file to design your process.

your project list should look like this:



Modeling is free, but your free process **execution trial** ends in **20 days.**

Welcome, Jadwiga Piechota

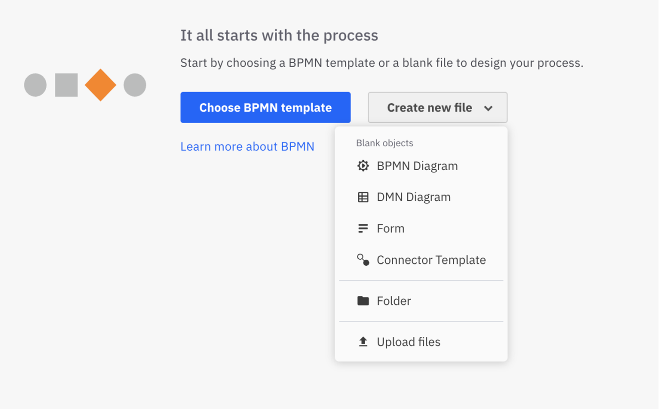
Name	Content	Last changed ↓	Collaborators
Project Workshop	0 files	2023-05-16 10:11	JP

II First Exercise - Create your first workflow in Camunda Cloud Modeler

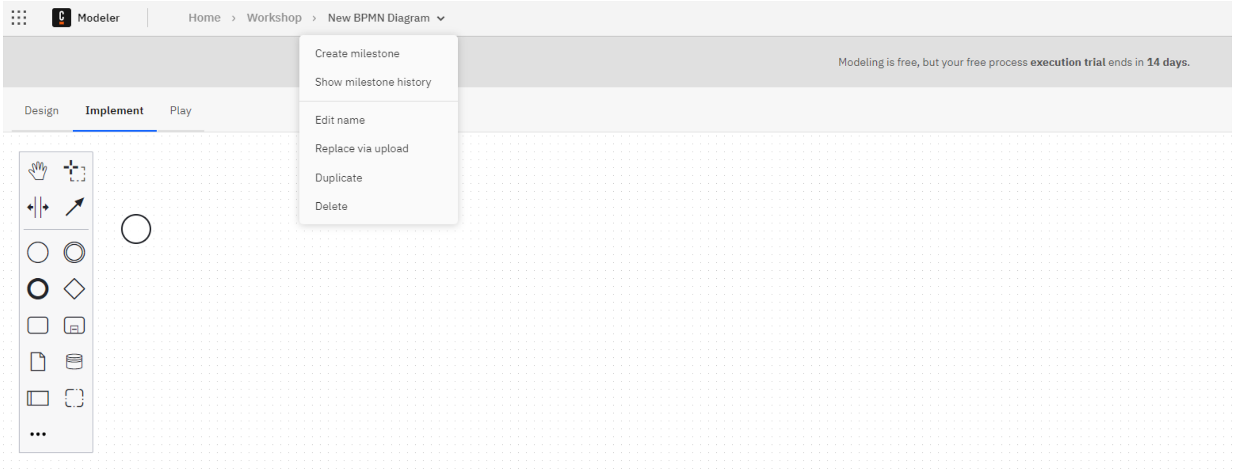
Task description:

Define a process that needs to automate bank account creation.
Customer requests a bank account. Next, the bank employee needs to review the request to decide if the request can be processed. If the decision is successful, the other employee has to provision the account. Otherwise, the request is rejected and the account is not created.

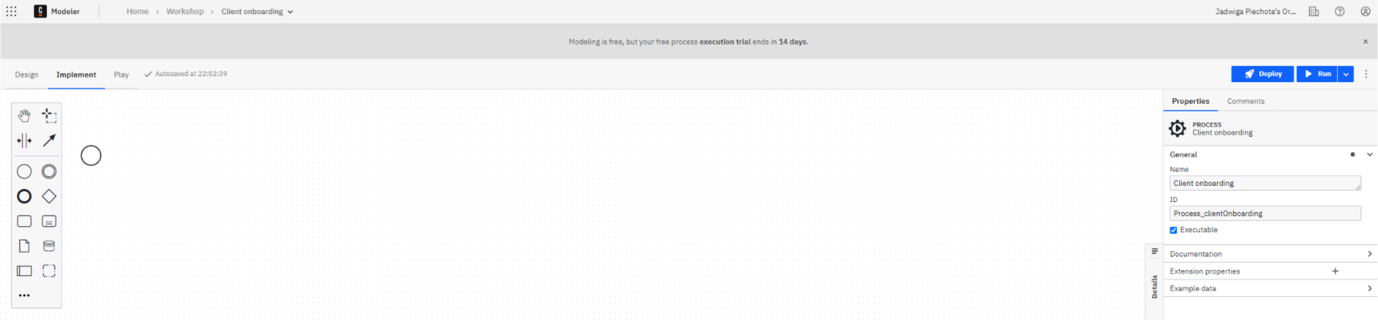
1. Create your first BPMN
   a. Open your project, click on Create new file and choose *BPMN Diagram*.



   b. Name your BPMN diagram with a descriptive name e.g. Client onboarding. Click on the default name (top bar) and choose *Edit name*:


Modeler
Home › Workshop › New BPMN Diagram
Create milestone
Show milestone history
Edit name
Replace via upload
Duplicate
Delete
Modeling is free, but your free process execution trial ends in 14 days.
Design
Implement
Play

c. Open *Details* section (right hand side). Next, click on the *General* tab. Fill in Name and ID. Name refers to a custom model name. ID is a unique identifier which is important when you connect your model with other applications.



Modeling is free, but your free process **execution trial** ends in **14 days.**

✓ Autosaved at 22:52:39

PROCESS
Client onboarding

General

Name: Client onboarding

ID: Process_clientOnboarding

☑ Executable

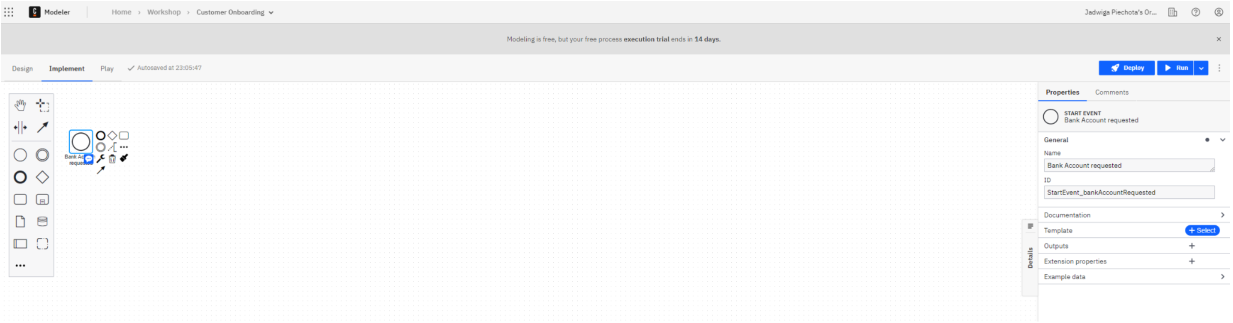
Documentation ›
Extension properties +
Example data ›

On the screen you should see a new, empty BPMN diagram with the start event.

2. Name the start event.

a. Click on start event icon: ○ and open *Details* section -> *General* tab. Fill the fields as below:

Name	Bank Account requested
ID	StartEvent_bankAccountRequested



Modeling is free, but your free process **execution trial** ends in **14 days.**

✓ Autosaved at 23:05:47

START EVENT
Bank Account requested

General

Name: Bank Account requested

ID: StartEvent_bankAccountRequested

Documentation ›
Template + Select
Outputs +
Extension properties +
Example data ›

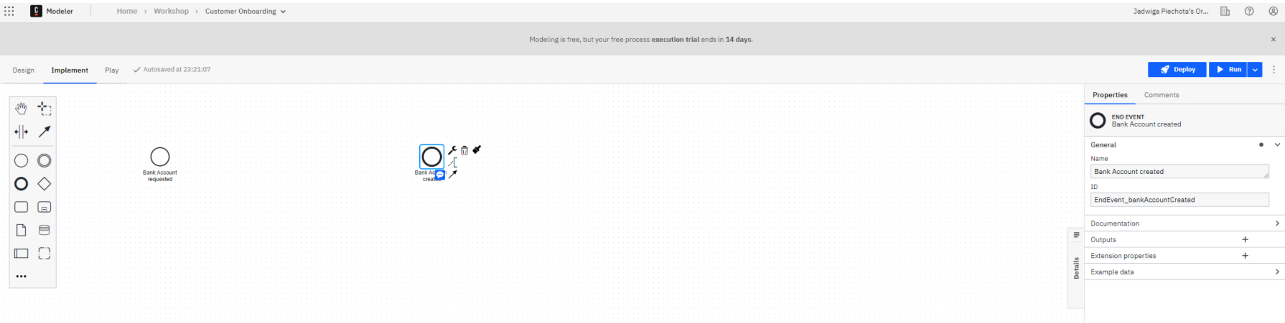
You can also change the name by double clicking on start event icon.

3. Name the end event.

a. Add end event - from the menu on the left hand select end event symbol ◯ and drop it on the diagram.

b. Click on end event icon on the diagram and and open *Details* section -> *General* tab. Fill the fields as below:

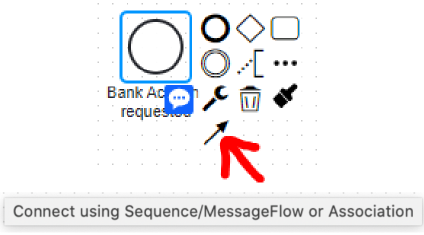
Name	Bank Account created
ID	EndEvent_bankAccountCreated

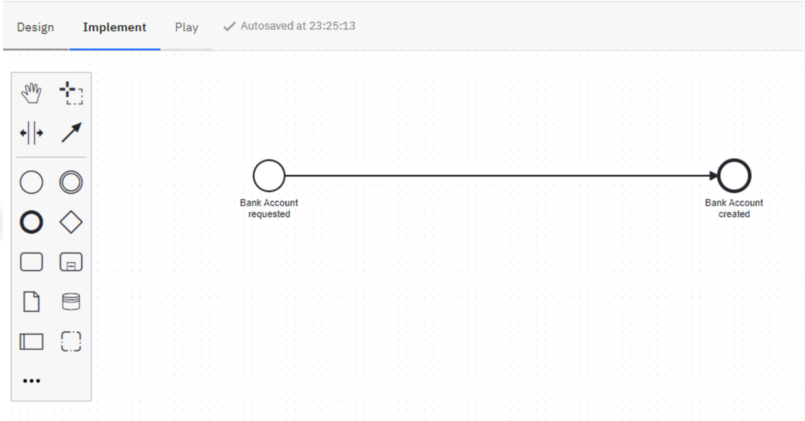


Name: Bank Account created

ID: EndEvent_bankAccountCreated

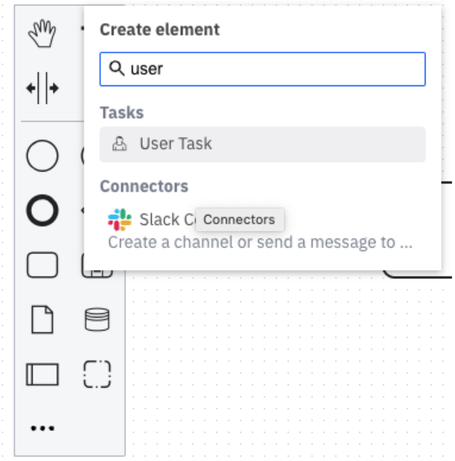
4. Connect start event with end event. Click on start event, select the arrow symbol located on the bottom and move the mouse over the end event. Click on the end event. You diagram should look like below:



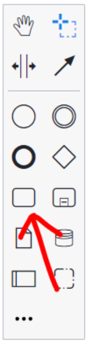


5. Now, let's add our first User Task.

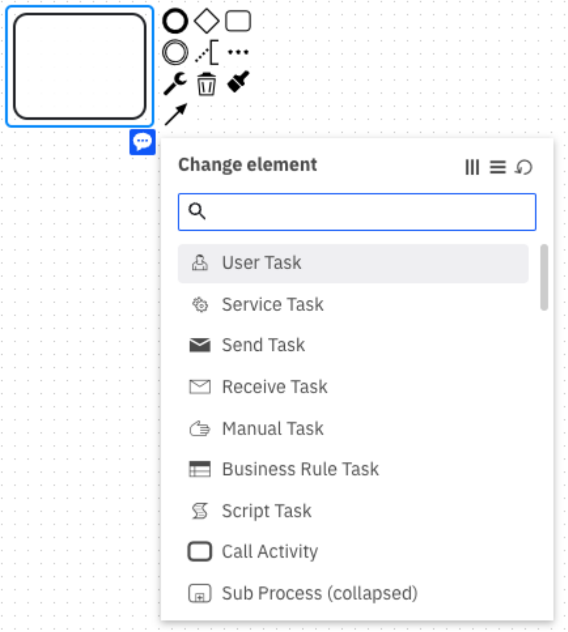
   a. Click on tree dots in the menu on the left ••• .
   Then in search type User Task and add it to the diagram.



You can also select Create Task icon from the left menu:



click on the monkey wrench icon and select User task:



Change element

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

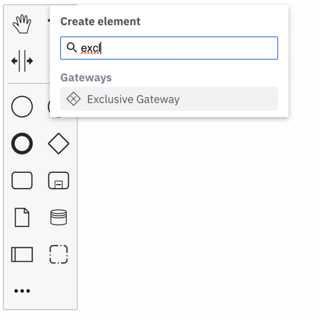
b. Name user task Review client request.

Name	Review client request
ID	Activity_reviewClientRequest

c. We want to attach the form for this task in order to be able to take an action by the human actor to execute the work and move forward.
Click on the user task, then open *Details* section -> *Form* tab on the right.
Select the *Type* as **Camunda Forms** and paste below json into the box called *Form JSON configuration*:

```
{
  "components": [
    {
      "label": "Is request approved?",
      "type": "checkbox",
      "id": "Field_1n7851c",
      "key": "automaticProcessing",
      "description": "Bank Account creation request"
    }
  ],
  "type": "default",
  "id": "Form_0w8g0i0",
  "schemaVersion": 4
}
```

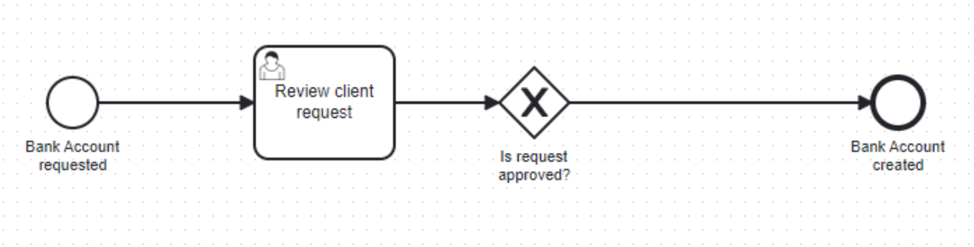
6. Add exclusive Gateway.
    a. Click three dots in the menu on the left and search Exclusive Gateway. Add the element to the model between task *Review client request* and end event. Try to drop the element on the line that connects both elements.



    b. Name exclusive gateway with label *Is request approved?*

Name	Is request approved?
ID	Gateway_isRequestApproved

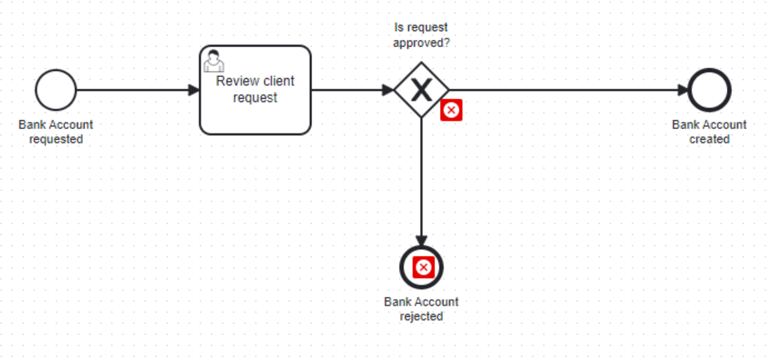
    c. You diagram should look like below:



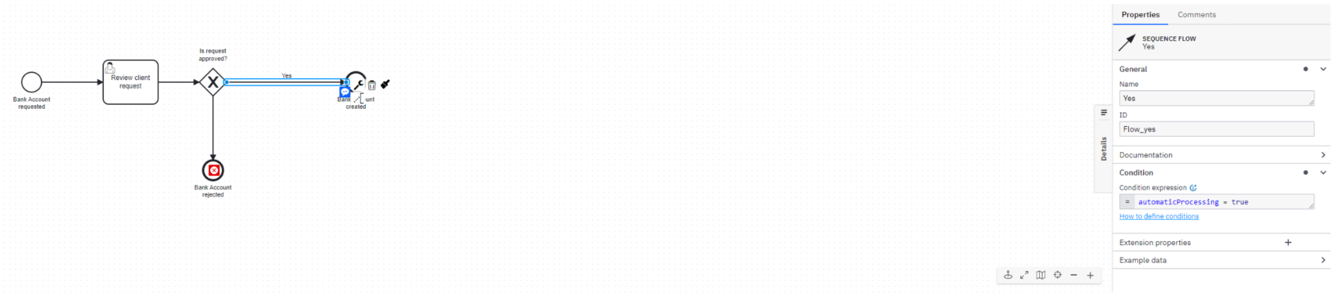
7. Add End Event. Select end event icon ○ in the menu on the left.
    a. Name this Event as *Bank Account rejected*

Name	Bank Account rejected
ID	EndEvent_bankAccountRejected

    b. Add the end event below the exclusive gateway. Connect Gateway with End Event. You diagram should look like below:



8. Now we are going to fix problems you can see on the diagram.
    a. Click on the arrow from the exclusive gateway to the end event called *Bank Account created*, open *Details* section -> *General* tab. Adjust the name and ID.
    Then, open *Details* section -> *Condition* tab on the right. Set the condition matching the output flow under *Condition expression*: automaticProcessing = true. It means when the variable called *automaticProcessing* is set to true by the bank employee who is reviewing the request, we follow that path when the process run.



Properties | Comments

SEQUENCE FLOW
Yes

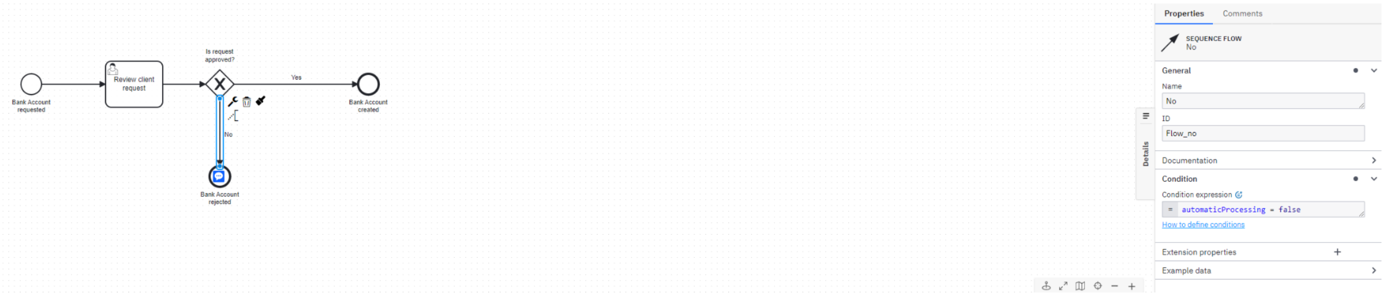
General
Name: Yes
ID: Flow_yes

Documentation

Condition
Condition expression: = automaticProcessing = true
How to define conditions

Extension properties +
Example data

    b. Click on the arrow from the exclusive gateway to the end event called *Bank Account rejected*, open *Details* section -> *General* tab. Adjust the name and ID.
    Then, open *Details* section -> *Condition* tab on the right. Set the condition matching the output flow under *Condition expression*: automaticProcessing = false. It means when the variable called *automaticProcessing* is set to false by the bank employee who is reviewing the request, we follow that path when the process run.



Properties | Comments

SEQUENCE FLOW
No

General
Name: No
ID: Flow_no

Documentation

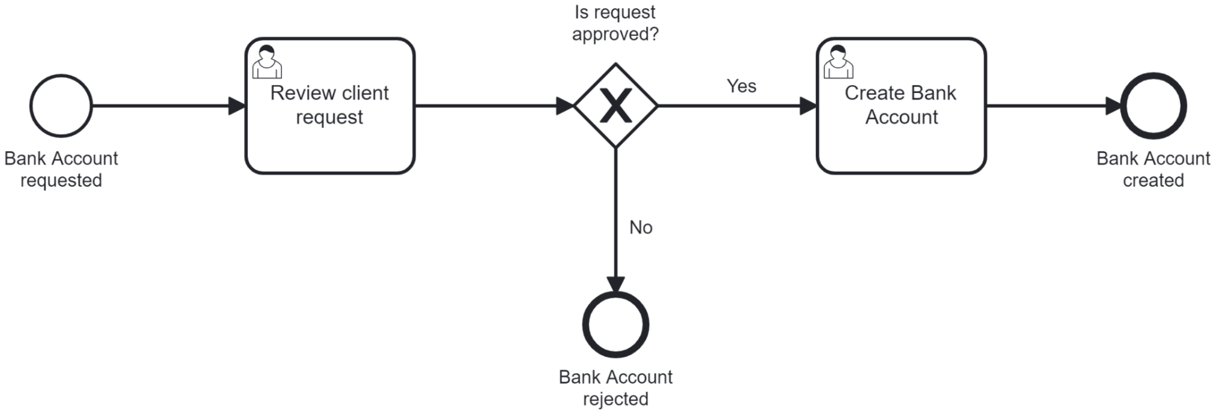
Condition
Condition expression: = automaticProcessing = false
How to define conditions

Extension properties +
Example data

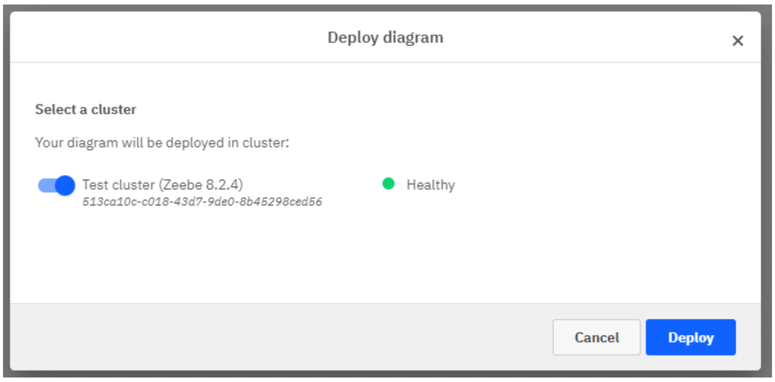
9. Create the second User Task.
    a. Choose User Task from the menu on the left and drop it on the diagram between the exclusive gateway and the end event called *Bank Account created*.
    b. Name user task.

Name	Create Bank Account
ID	Activity_createBankAccount

10. You final diagram should look like below:



11. Now it is time to deploy your process to the workflow engine. The purpose of that step is to provide the workflow engine with your process definition. Click on 🚀 Deploy button.

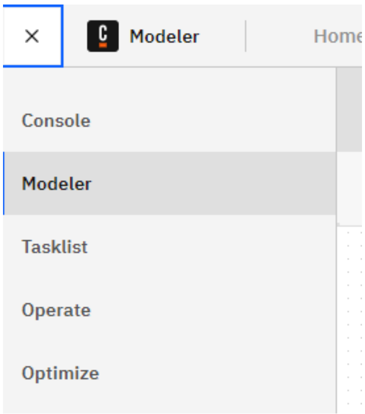

Deploy diagram ×

Select a cluster

Your diagram will be deployed in cluster:

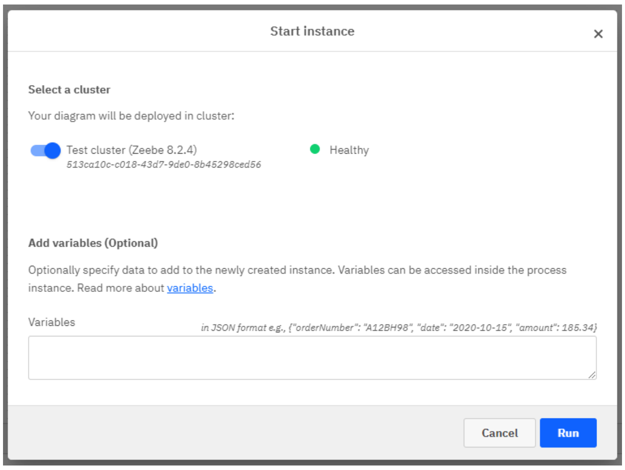
Test cluster (Zeebe 8.2.4) ● Healthy
513ca10c-c018-43d7-9de0-8b45298ced56

12. In the Camunda apps click on *Operate* tab.



You should see your model deployed - look for your model name that was created in point 1c.

13. Now it is time to create an instance of your model. Come back to the Modeler view and click ▶ Run



Start instance ×

Select a cluster

Your diagram will be deployed in cluster:

Test cluster (Zeebe 8.2.4) ● Healthy
513ca10c-c018-43d7-9de0-8b45298ced56

Add variables (Optional)

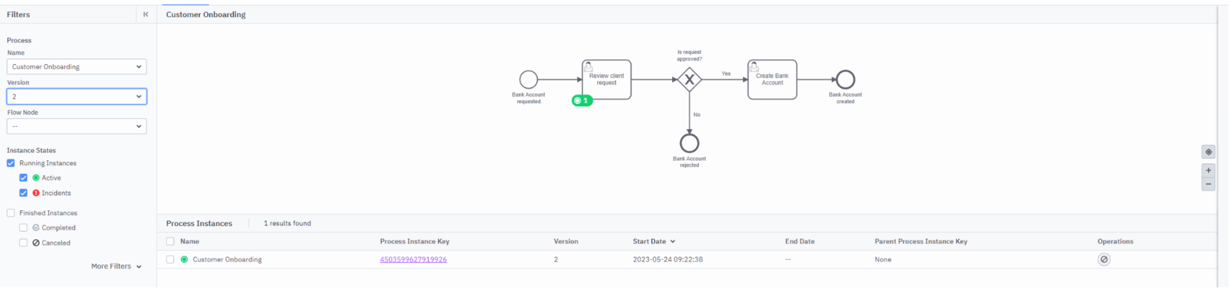
Optionally specify data to add to the newly created instance. Variables can be accessed inside the process instance. Read more about variables.

Variables — in JSON format e.g., {"orderNumber": "A12BH98", "date": "2020-10-15", "amount": 185.34}

Cancel | Run

Amazing! Your first BPMN model is deployed and an instance is created.
Play around. Check if the model behaves as expected.

14. When you come back to *Operate* view and click on your process, you will see the instance that is currently active:



Filters

Process
Name: Customer Onboarding
Version: 2
Flow Node: --

Instance States
☑ Running Instances
☑ ● Active
☑ ● Incidents
□ Finished Instances
□ Completed
□ Canceled

More Filters

Customer Onboarding

Process Instances — 1 results found

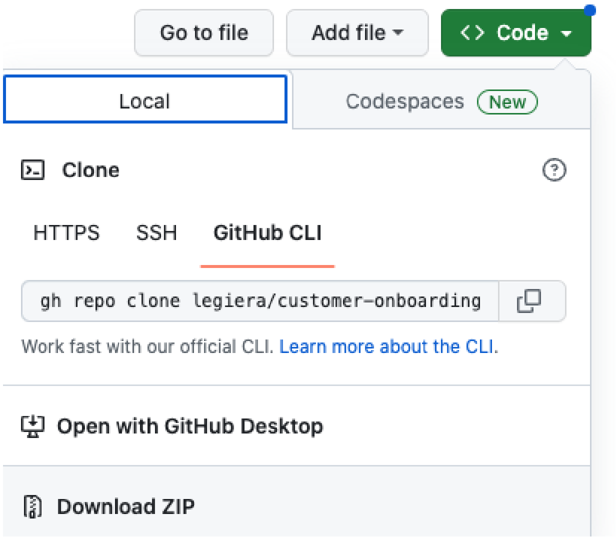
□ Name	Process Instance Key	Version	Start Date	End Date	Parent Process Instance Key	Operations
□ ● Customer Onboarding	4503599627919926	2	2023-05-24 09:22:38	--	None	⊘

As you can see, bank employee need to perform an action in order to decide if the account should be created. Open Camunda apps -> *Tasklist* tab. You would see the active task that needs to be actioned - that is the form you pasted in the user task in point 5c.
Open the task, assign to yourself and decide if the account should be created. Come back to Operate view and see what happened to the process instance.

III Clone and set up project

1. Clone repository:

   git clone **https://github.com/legiera/customer-onboarding-camunda-8-springboot.git**
   or download it as a zip file. Once the zip file is downloaded, extract it.

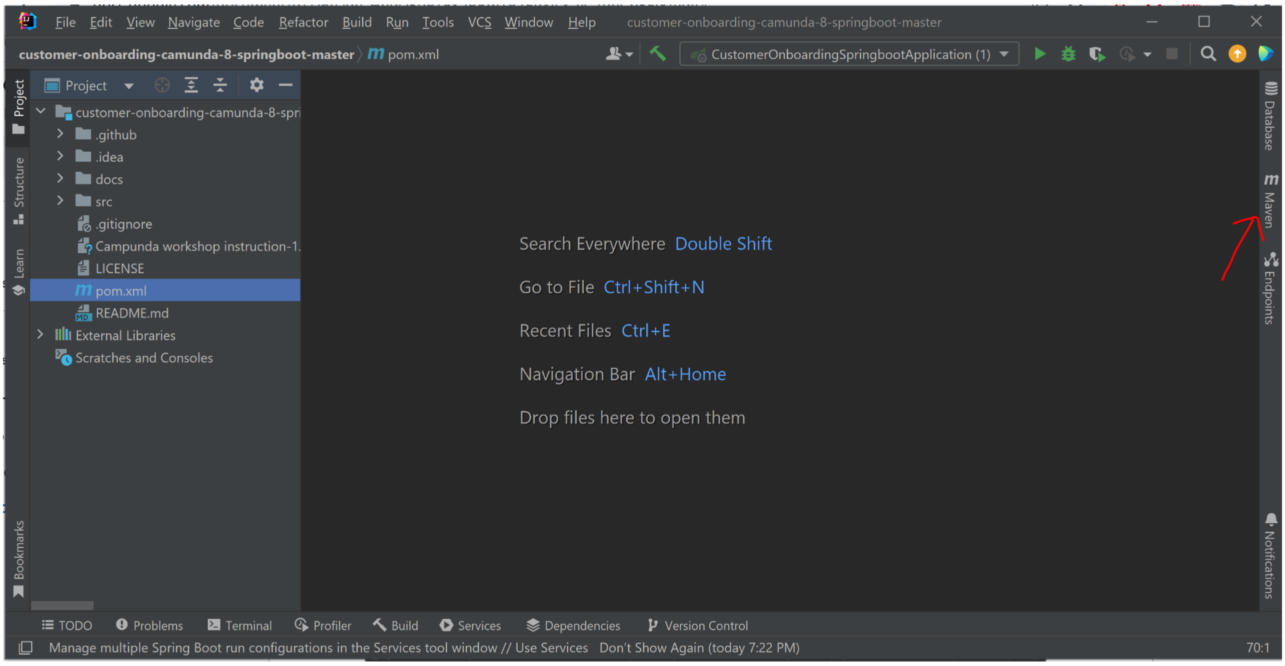


```
gh repo clone legiera/customer-onboarding
```

Work fast with our official CLI. Learn more about the CLI.

2. Open project in Intellij. Click on Projects -> Open -> select folder with downloaded project.

3. Project needs to be built with Maven. If you do not see the maven icon on the right hand side (see below), follow the instruction: click in Intellij top menu View -> Tool Windows -> Maven. You should now see the maven tab on the right.



File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
customer-onboarding-camunda-8-springboot-master
customer-onboarding-camunda-8-springboot-master 〉 pom.xml
CustomerOnboardingSpringbootApplication (1)
Project
customer-onboarding-camunda-8-spri
.github
.idea
docs
src
.gitignore
Campunda workshop instruction-1.
LICENSE
pom.xml
README.md
External Libraries
Scratches and Consoles
Structure
Learn
Bookmarks
Database
Maven
Endpoints
Notifications
TODO
Problems
Terminal
Profiler
Build
Services
Dependencies
Version Control
70:1

Search Everywhere Double Shift

Go to File Ctrl+Shift+N

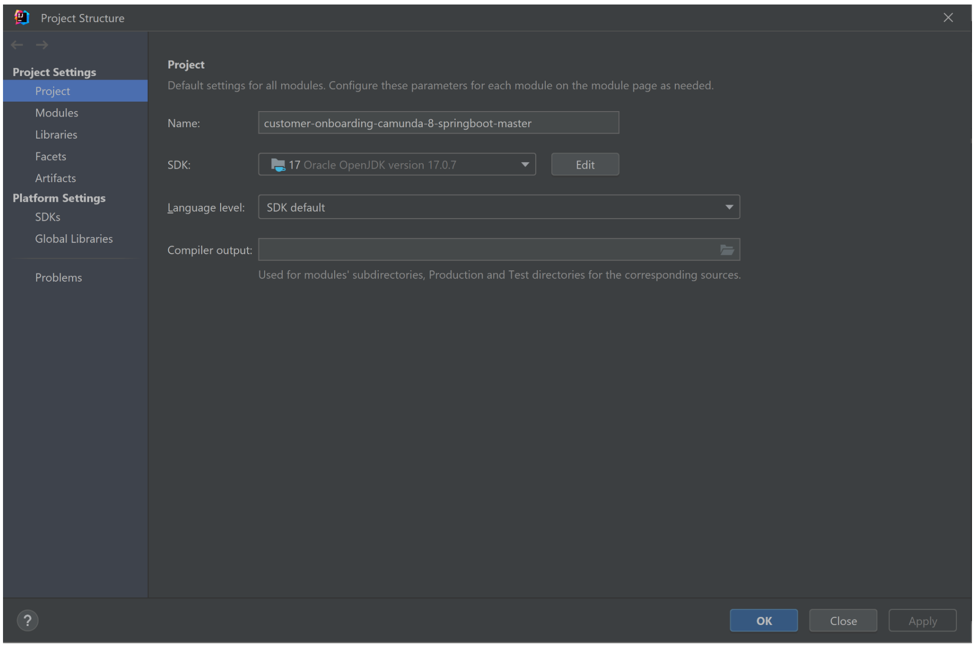
Recent Files Ctrl+E

Navigation Bar Alt+Home

Drop files here to open them

Manage multiple Spring Boot run configurations in the Services tool window // Use Services Don't Show Again (today 7:22 PM)

4. Make sure that the JDK version is set up correctly for your project. Open File -> Project Structure… -> Project under Project Settings. Choose SDK 17 from drop-down list:



Project

Default settings for all modules. Configure these parameters for each module on the module page as needed.

Name: customer-onboarding-camunda-8-springboot-master

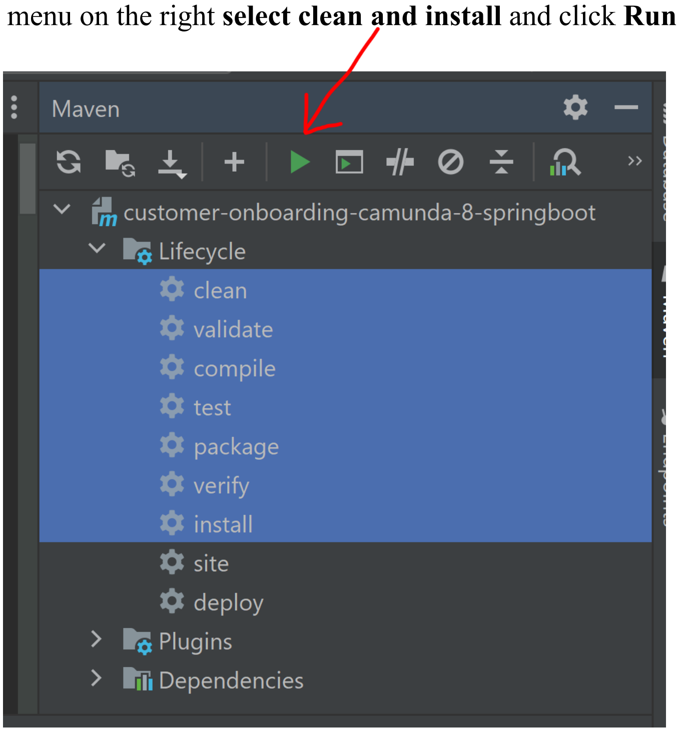
SDK: 17 Oracle OpenJDK version 17.0.7

Language level: SDK default

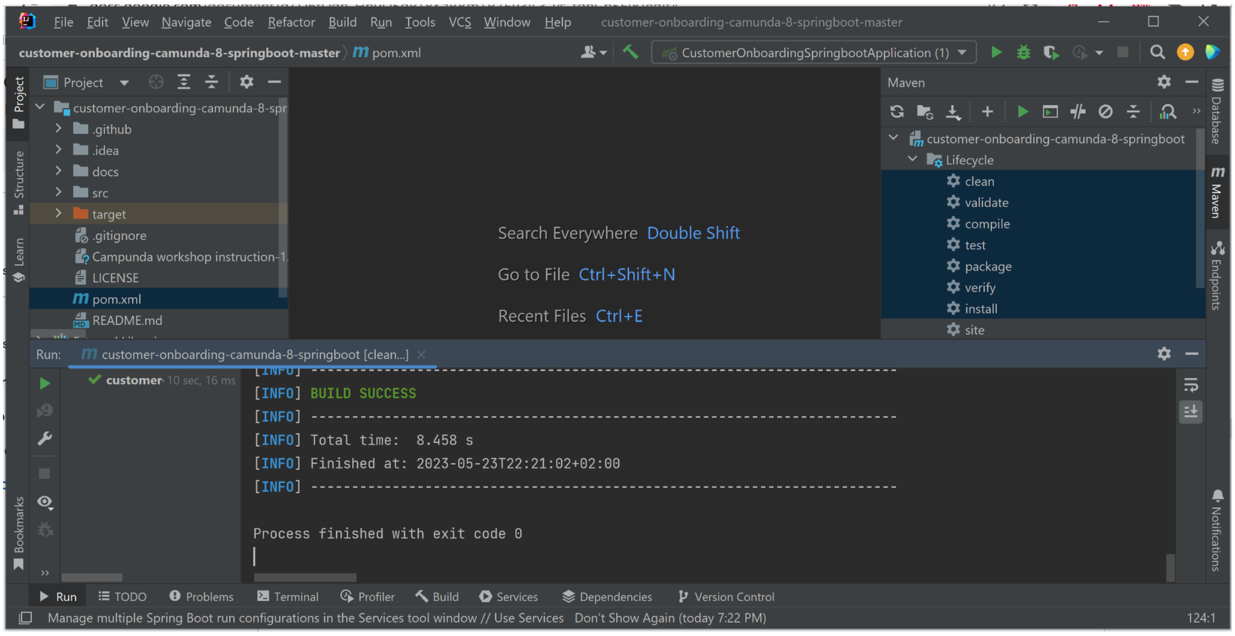
Compiler output:

Used for modules' subdirectories, Production and Test directories for the corresponding sources.

5. Build the project: **double click CTRL** and type ***maven clean install*** or click **shift** and in the maven menu on the right **select clean and install** and click **Run Maven Build** button**:**

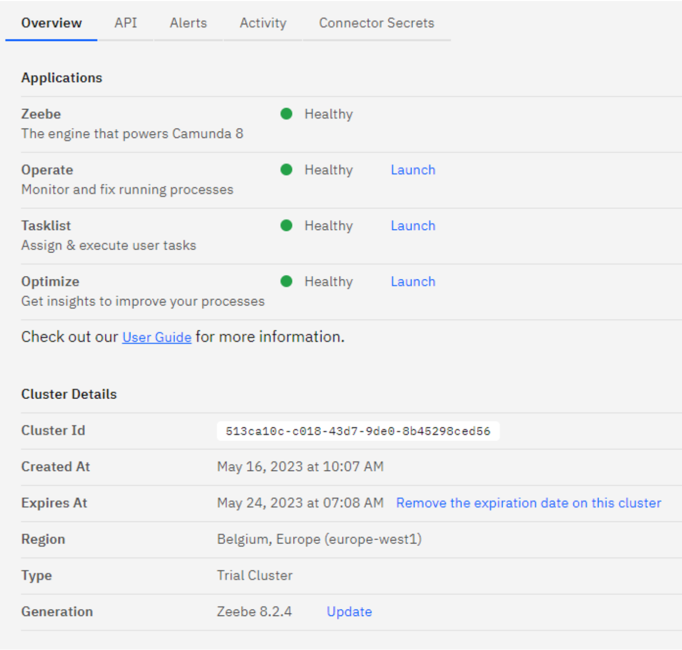


when the build is finished, you should see information that build completed successfully, see below:


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
customer-onboarding-camunda-8-springboot-master
customer-onboarding-camunda-8-springboot-master 〉 pom.xml
CustomerOnboardingSpringbootApplication (1)
Project
customer-onboarding-camunda-8-spr
.github
.idea
docs
src
target
.gitignore
Campunda workshop instruction-1
LICENSE
pom.xml
README.md
Search Everywhere Double Shift
Go to File Ctrl+Shift+N
Recent Files Ctrl+E
Maven
customer-onboarding-camunda-8-springboot
Lifecycle
clean
validate
compile
test
package
verify
install
site
Run: customer-onboarding-camunda-8-springboot [clean...]
customer 10 sec, 16 ms
Run
TODO
Problems
Terminal
Profiler
Build
Services
Dependencies
Version Control
Manage multiple Spring Boot run configurations in the Services tool window // Use Services Don't Show Again (today 7:22 PM)
124:1

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  8.458 s
[INFO] Finished at: 2023-05-23T22:21:02+02:00
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

6. Create Client Credential
   Credentials will be used in order to connect to your cluster from outside of Camunda Cloud. They are necessary for security reasons to make sure only authorised applications can execute operations on your cluster.
   a. Open Camunda Cloud. Go to your Cluster - click on Camunda apps on the top left corner and choose Console -> Clusters tab. Click on the name of your cluster. You should see below details:



Applications

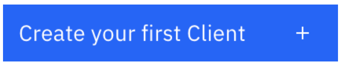
Zeebe The engine that powers Camunda 8	● Healthy	
Operate Monitor and fix running processes	● Healthy	Launch
Tasklist Assign & execute user tasks	● Healthy	Launch
Optimize Get insights to improve your processes	● Healthy	Launch

Check out our User Guide for more information.

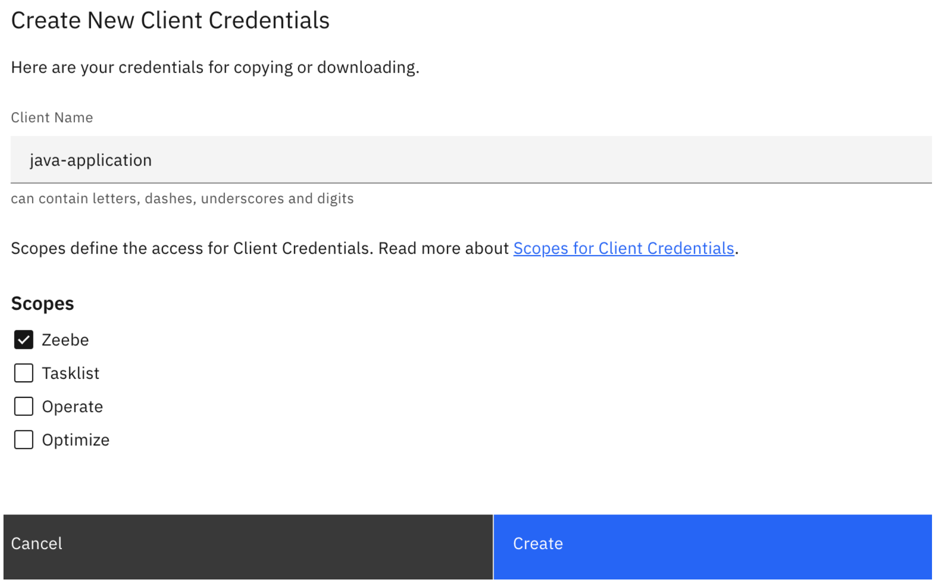
Cluster Details

Cluster Id	513ca10c-c018-43d7-9de0-8b45298ced56
Created At	May 16, 2023 at 10:07 AM
Expires At	May 24, 2023 at 07:08 AM Remove the expiration date on this cluster
Region	Belgium, Europe (europe-west1)
Type	Trial Cluster
Generation	Zeebe 8.2.4 Update

   b. Then select **API tab** and click:



c. Provide details as below and click Create:



Create New Client Credentials

Here are your credentials for copying or downloading.

Client Name

java-application

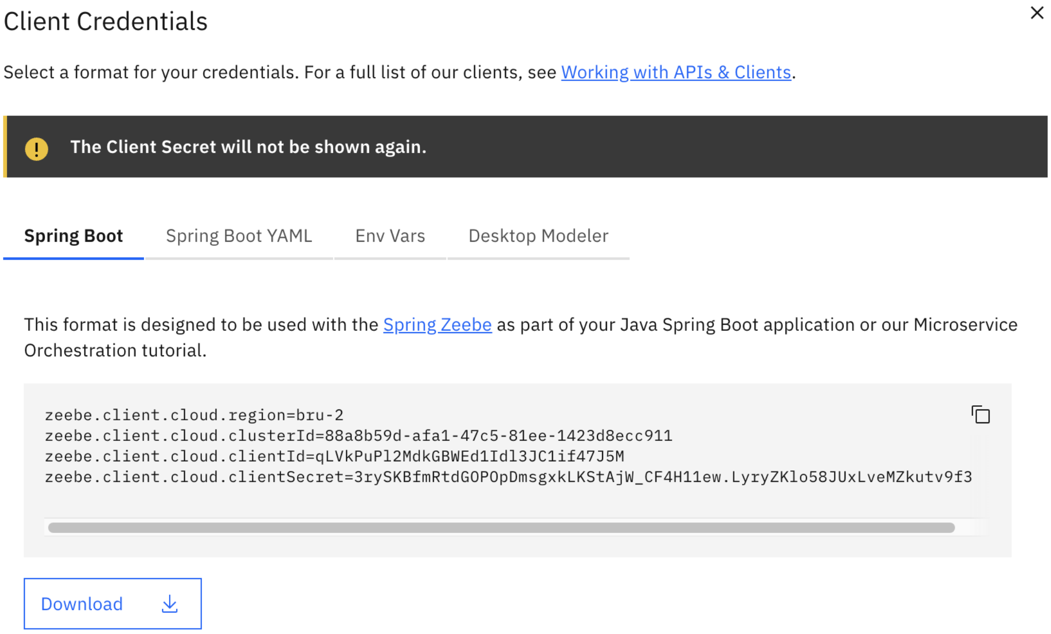
can contain letters, dashes, underscores and digits

Scopes define the access for Client Credentials. Read more about Scopes for Client Credentials.

Scopes

☑ Zeebe
□ Tasklist
□ Operate
□ Optimize

d. You should see details of your **Client Secret**:



Client Credentials

Select a format for your credentials. For a full list of our clients, see Working with APIs & Clients.

The Client Secret will not be shown again.

This format is designed to be used with the Spring Zeebe as part of your Java Spring Boot application or our Microservice Orchestration tutorial.

```
zeebe.client.cloud.region=bru-2
zeebe.client.cloud.clusterId=88a8b59d-afa1-47c5-81ee-1423d8ecc911
zeebe.client.cloud.clientId=qLVkPuPl2MdkGBWEd1Idl3JC1if47J5M
zeebe.client.cloud.clientSecret=3rySKBfmRtdGOPOpDmsgxkLKStAjW_CF4H11ew.LyryZKlo58JUxLveMZkutv9f3
```

e. Download your **Client Secret** and copy it to the clipboard.

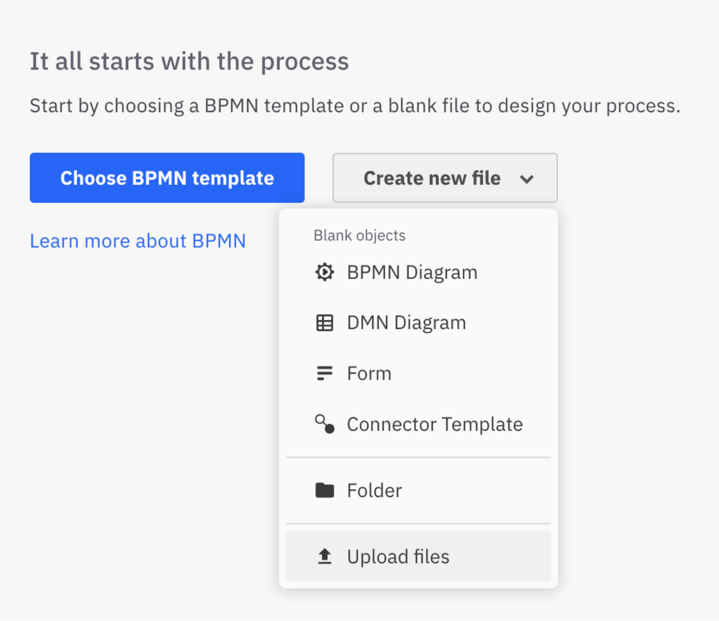
7. Open your project in Intellij and update credentials that you have just created.
Open file located under *<your_base_path>\customer-onboarding-camunda-8-springboot-master\src\main\resources\application.properties*
and paste the content **application.properties** you downloaded in point 6e.

IV Second Exercise - Create Simple Bank Account System

Task description:

Define a process that needs to **automate bank account creation**. Customer requires a bank account. The external system needs to score the customer. Next, the bank **employee needs to review the request** in order to decide whether an account for this Client can be created. If the decision is **successful, the request is sent to microservice** to provision the bank account. **Otherwise, the request is rejected** and the bank account is not created.

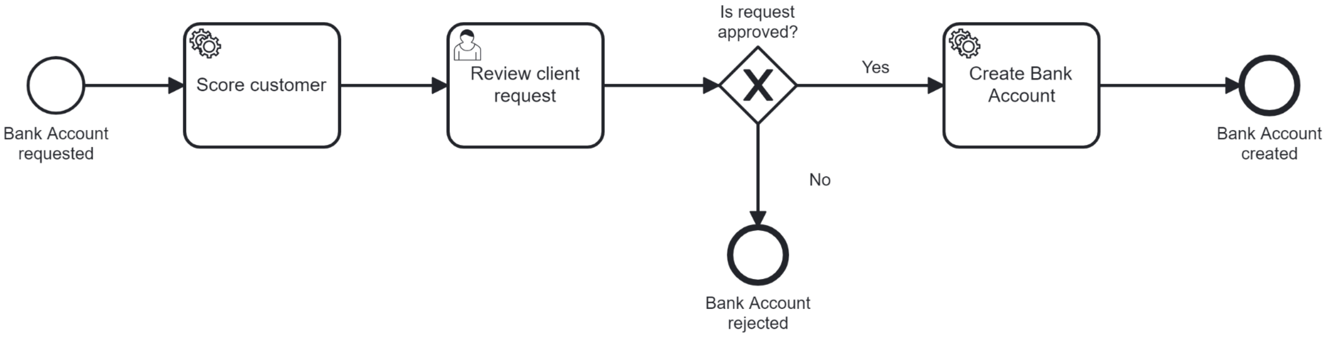
1. For this exercise, we are providing the ready BPMN diagram that needs to be uploaded to Camunda Cloud.
   Locate BPMN in the project - it is located under *<your_base_path>\customer-onboarding-camunda-8-springboot-master\src\main\resources\bank-account-simple.bpmn.*
2. Upload BPMN to Camunda Cloud
   a. Open Camunda Cloud and create a new project - follow the steps 3a and 3b from the section I of that instruction.
   b. Open your project and select Create new file -> Upload files. Find BPMN from point 1 and click Open.



It all starts with the process

Start by choosing a BPMN template or a blank file to design your process.

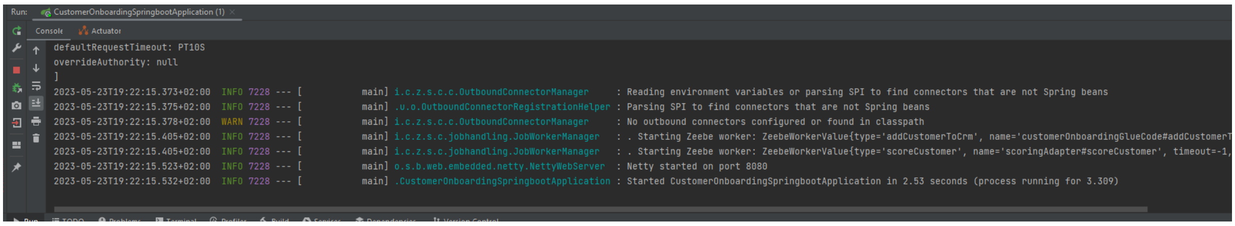
c. You should now be able to open the uploaded diagram in the *Modeler*. It should look like below:



3. Deploy the process - click on Implement tab and then hit Deploy

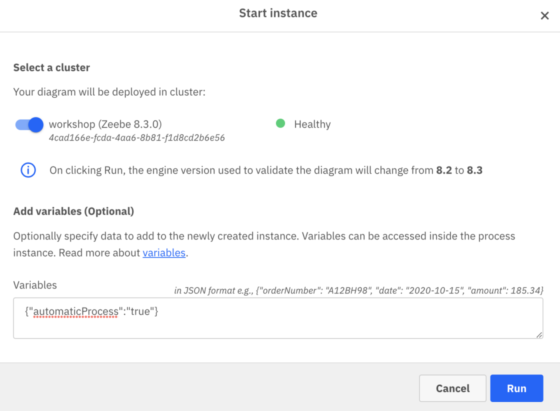
4. Open Camunda apps -> *Operate* tab. You should see the definition of your process deployed to the Zeebe engine.

5. Now it is time to run the Java application. Come back to Intellij.
    a. Open the main class where application startup method is located: *<your_base_path>\customer-onboarding-camunda-8-springboot-master\src\main\java\io\camunda\workshop\customer\CustomerOnboardingSpringbootApplication.java*
    b. Right click on **CustomerOnboardingSpringbootApplication** - > 'Run CustomerOnboardingSpringbootApplication'.
    c. When process starts, you should see below logs in the console:



```
defaultRequestTimeout: PT10S
overrideAuthority: null
]
2023-05-23T19:22:15.373+02:00  INFO 7228 --- [           main] i.c.z.s.c.c.OutboundConnectorManager     : Reading environment variables or parsing SPI to find connectors that are not Spring beans
2023-05-23T19:22:15.375+02:00  INFO 7228 --- [           main] .u.o.OutboundConnectorRegistrationHelper : Parsing SPI to find connectors that are not Spring beans
2023-05-23T19:22:15.378+02:00  WARN 7228 --- [           main] i.c.z.s.c.c.OutboundConnectorManager     : No outbound connectors configured or found in classpath
2023-05-23T19:22:15.405+02:00  INFO 7228 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='addCustomerToCrm', name='customerOnboardingGlueCode#addCustomerT
2023-05-23T19:22:15.405+02:00  INFO 7228 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='scoreCustomer', name='scoringAdapter#scoreCustomer', timeout=-1,
2023-05-23T19:22:15.523+02:00  INFO 7228 --- [           main] o.s.b.web.embedded.netty.NettyWebServer  : Netty started on port 8080
2023-05-23T19:22:15.532+02:00  INFO 7228 --- [           main] .CustomerOnboardingSpringbootApplication : Started CustomerOnboardingSpringbootApplication in 2.53 seconds (process running for 3.309)
```

6. Come back to the model in Camunda Cloud and create the process instance.
    a. Click on Open Camunda apps -> *Modeler* tab. Select **Implement Tab** and click on ▶ Run
    b. Start an instance with the following variable *{ "automaticProcess": "true" }* as below:



Start instance ×

Select a cluster

Your diagram will be deployed in cluster:

workshop (Zeebe 8.3.0) ● Healthy
4cad166e-fcda-4aa6-8b81-f1d8cd2b6e56

On clicking Run, the engine version used to validate the diagram will change from **8.2** to **8.3**

Add variables (Optional)

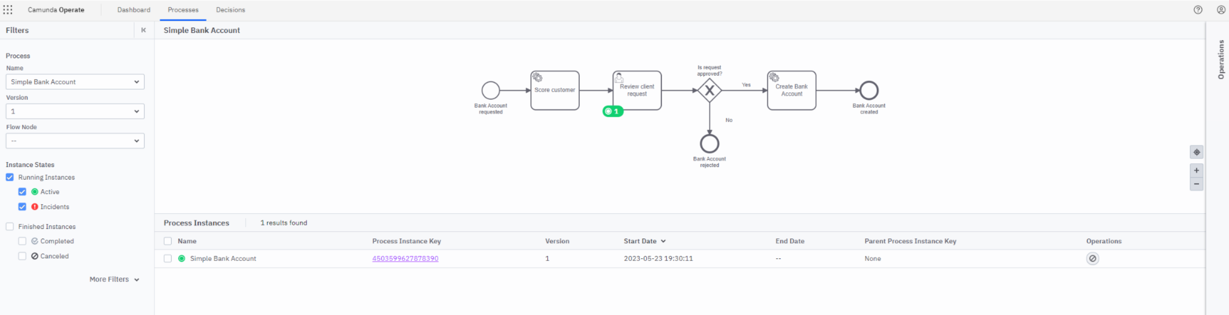
Optionally specify data to add to the newly created instance. Variables can be accessed inside the process instance. Read more about variables.

Variables *in JSON format e.g., {"orderNumber": "A12BH98", "date": "2020-10-15", "amount": 185.34}*

{"automaticProcess":"true"}

Cancel Run

7. View status of your instance. Open Camunda Operate - click on Camunda apps -> *Operate* tab. Locate your instance and click on it. You should see the following:



Process

Name: Simple Bank Account

Version: 1

Flow Node: --

Instance States

☑ Running Instances

☑ Active

☑ Incidents

□ Finished Instances

□ Completed

□ Canceled

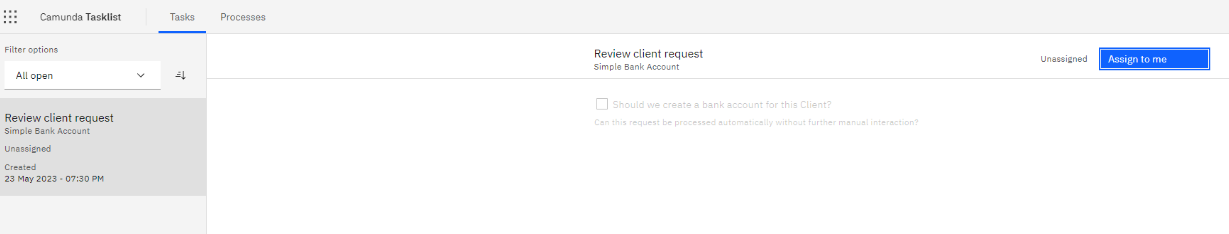
Process Instances 1 results found

Name	Process Instance Key	Version	Start Date	End Date	Parent Process Instance Key	Operations
Simple Bank Account	4503599627878390	1	2023-05-23 19:30:11	--	None	

You see that instance is waiting for a human actor to make a decision if the account should be created.

WARNING! If you do not see the instance (now or later), you probably deployed more than one time so please change the version from 1 to All to see all instances.

8. Open the Tasklist - click on Camunda apps -> *Tasklist* tab.
   a. You should see **Review client request** task.



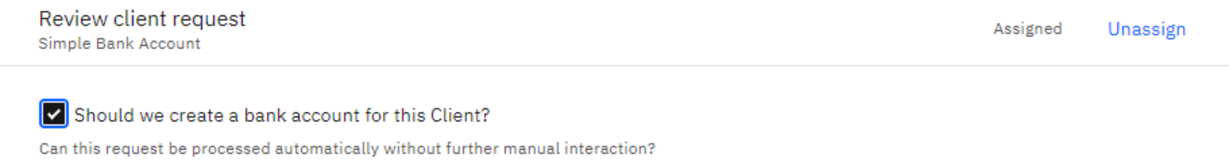
Filter options: All open

Review client request
Simple Bank Account
Unassigned
Created
23 May 2023 - 07:30 PM

Review client request
Simple Bank Account
Unassigned

□ Should we create a bank account for this Client?
Can this request be processed automatically without further manual interaction?

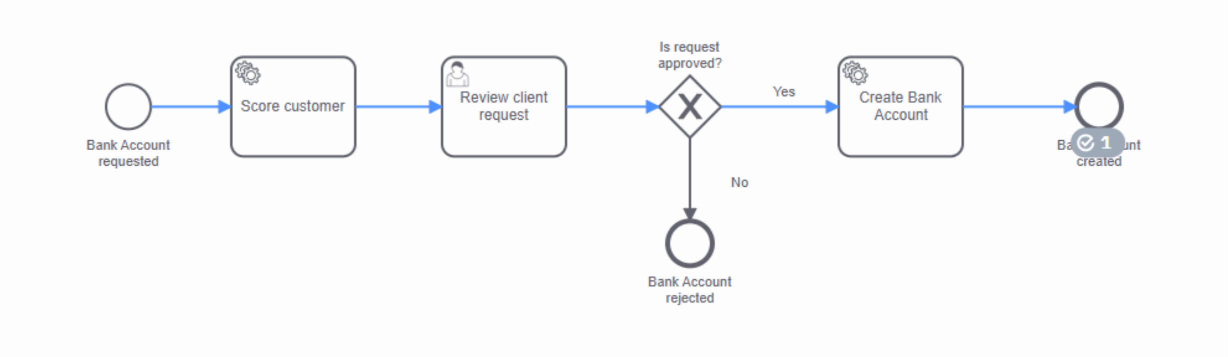
   b. Click on that new task and choose:
   c. Select box and click on **Complete Task**:



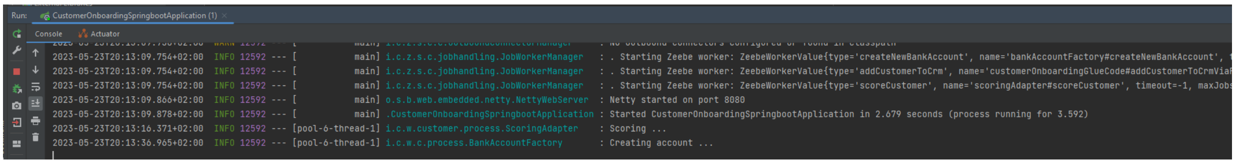
Review client request
Simple Bank Account
Assigned

☑ Should we create a bank account for this Client?
Can this request be processed automatically without further manual interaction?

9. Go back to your instance in Operate tab. Now you should see that token is processed and account is created:



10. Check also logs in the Intellij. What you can see? What happened?
    The logs should be as follows:



```
2023-05-23T20:13:09.754+02:00  INFO 12592 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='createNewBankAccount', name='bankAccountFactory#createNewBankAccount', 
2023-05-23T20:13:09.754+02:00  INFO 12592 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='addCustomerToCrm', name='customerOnboardingGlueCode#addCustomerToCrmVia
2023-05-23T20:13:09.754+02:00  INFO 12592 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='scoreCustomer', name='scoringAdapter#scoreCustomer', timeout=-1, maxJobs
2023-05-23T20:13:09.866+02:00  INFO 12592 --- [           main] o.s.b.web.embedded.netty.NettyWebServer  : Netty started on port 8080
2023-05-23T20:13:09.878+02:00  INFO 12592 --- [           main] .CustomerOnboardingSpringbootApplication : Started CustomerOnboardingSpringbootApplication in 2.679 seconds (process running for 3.592)
2023-05-23T20:13:16.371+02:00  INFO 12592 --- [pool-6-thread-1] i.c.w.customer.process.ScoringAdapter    : Scoring ...
2023-05-23T20:13:36.965+02:00  INFO 12592 --- [pool-6-thread-1] i.c.w.c.process.BankAccountFactory       : Creating account ...
```

11. No let’s create new instance from our machine (the same operation as in point 6):
    a. Open Command Prompt
    b. Type the following command:

    ```
    curl -X PUT http://localhost:8080/customer
    ```

    c. Alternatively open the link in your browser http://localhost:8080/ and click on Open bank account.

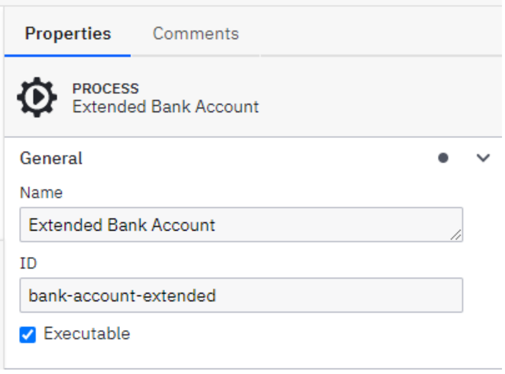
12. We should see the same effect as before - repeat steps 7-10.

V Third exercise - Create Extended Bank System

Task description:

Define a process that needs to **automate bank account creation**. Customer requires a bank account. The external system needs to score the customer. **If clients earnings are above 10k PLN and age is above 18 years old, a bank account creation request is sent to Create Bank Account service. If age is under 18 and earnings are negative the request is rejected automatically.** In other cases the bank employee needs to review the request in order to decide whether an account for this Client can be created. If the decision is successful, the other employee has to perform some steps in order to provision the bank account. Next, the bank account creation request is sent to the Create Bank Account service. Otherwise, the request is rejected and the bank account is not created.

1. Let's start with the BPMN model. For simplicity, please modify our existing model from section IV.
    a. Open Camunda Modeler - click on Camunda apps -> *Modeler* tab.
    b. Find the bank-account-simple model, click on three dots on the right hand side and choose *Duplicate.*
    c. Change the name of the diagram - open a copy of the model and click on top to change the name (see point 1b in section II of this instruction). Name it **bank-account-extended**.
    d. Change the name of the model to **Extended Bank Account** and model ID to **bank-account-extended.**



PROCESS
Extended Bank Account

General

Name: Extended Bank Account

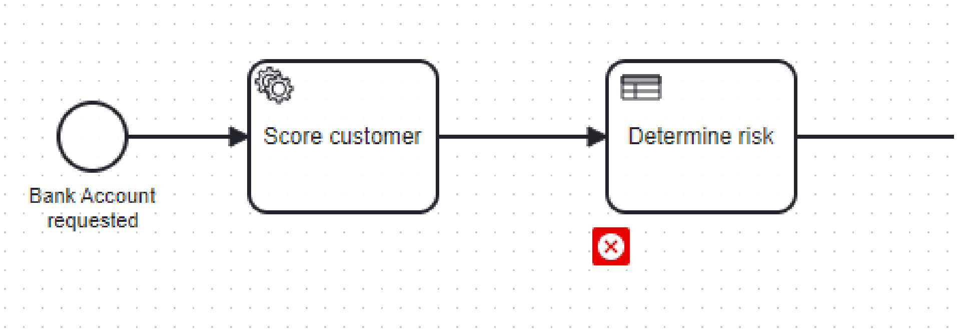
ID: bank-account-extended

☑ Executable

2. Create a **Business Rule Task** to automate decision-making.
    a. Open the model in the Modeler, then click ••• in the edit menu on the left and search for **Business Rule Task.** Add it to your model behind **Score Customer** service task with name **Determine risk:**

Name	Determine risk
ID	Activity_determineRisk

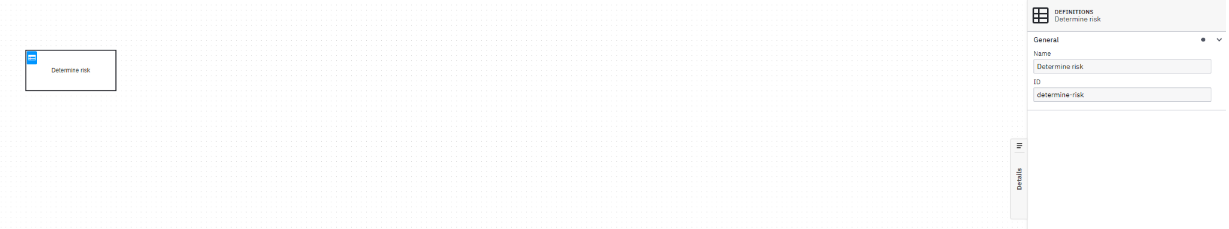
At the current moment, it should look like below:



3. Create New DMN Diagram in Modeler.
    a. Open your existing project where **bank-account-extended** sits.
    b. Click on New -> DMN Diagram. Name the diagram with *Determine risk*.
    c. Open *Details* section (right hand side). Next, click on the *General* tab. Fill in Name and ID. Name refers to a custom model name. ID is a unique identifier which is important when you connect your model with other applications.

Name	Determine risk
ID	determine-risk

the diagram should look like below:

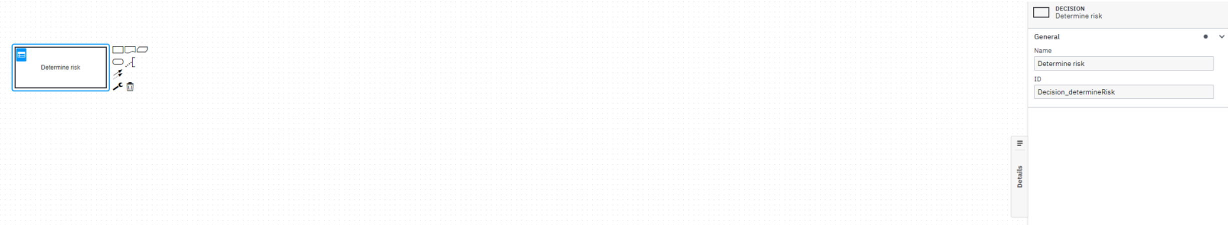


Name
Determine risk
ID
determine-risk

    d. Next click on the DMN element and fill in the same values as in point c. **WARNING!!!** This is an important step that determines later how your DMN can be connected with a BPMN diagram.

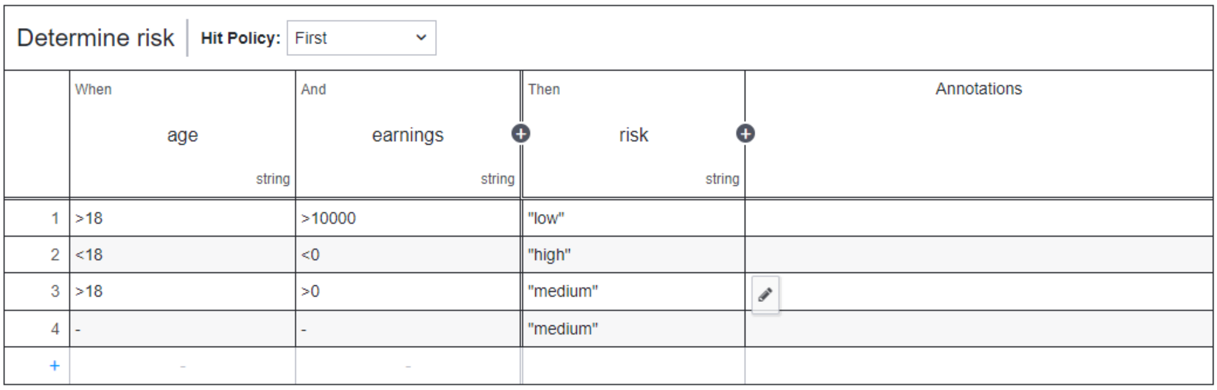
Name	Determine risk
ID	Decision_determineRisk

It should look as below:



Name
Determine risk
ID
Decision_determineRisk

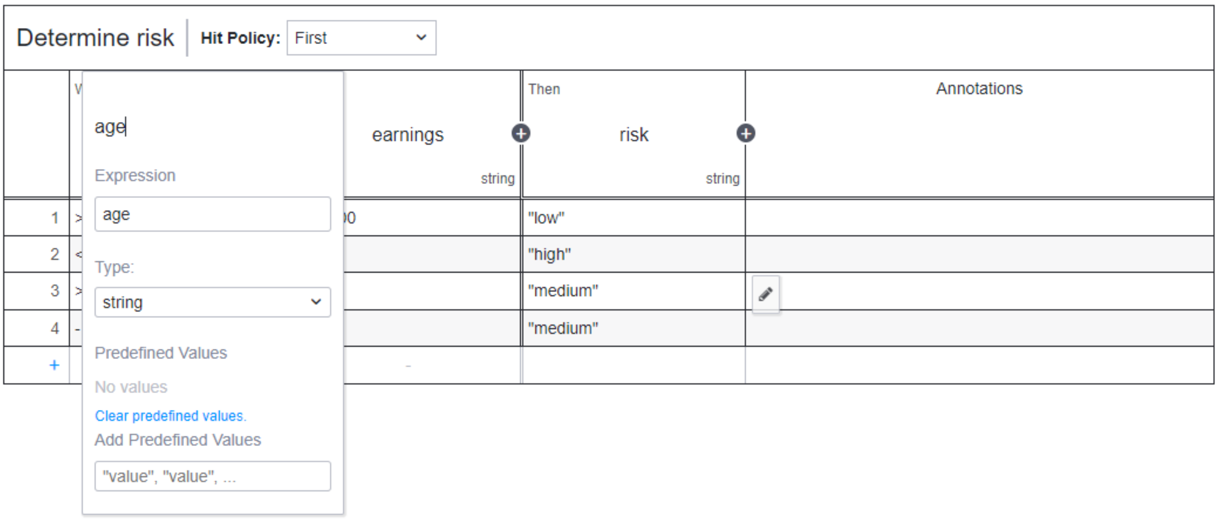
4. Edit decision table - click on ▤ to define your logic. You can try to fill in the table yourself or copy ready solution as below:



Determine risk | **Hit Policy:** First

	When age string	And earnings string	Then risk string	Annotations
1	>18	>10000	"low"	
2	<18	<0	"high"	
3	>18	>0	"medium"	
4	-	-	"medium"	
+	-	-		

<u>Remember to set up each input and output:</u>



Determine risk | **Hit Policy:** First

age

Expression: age

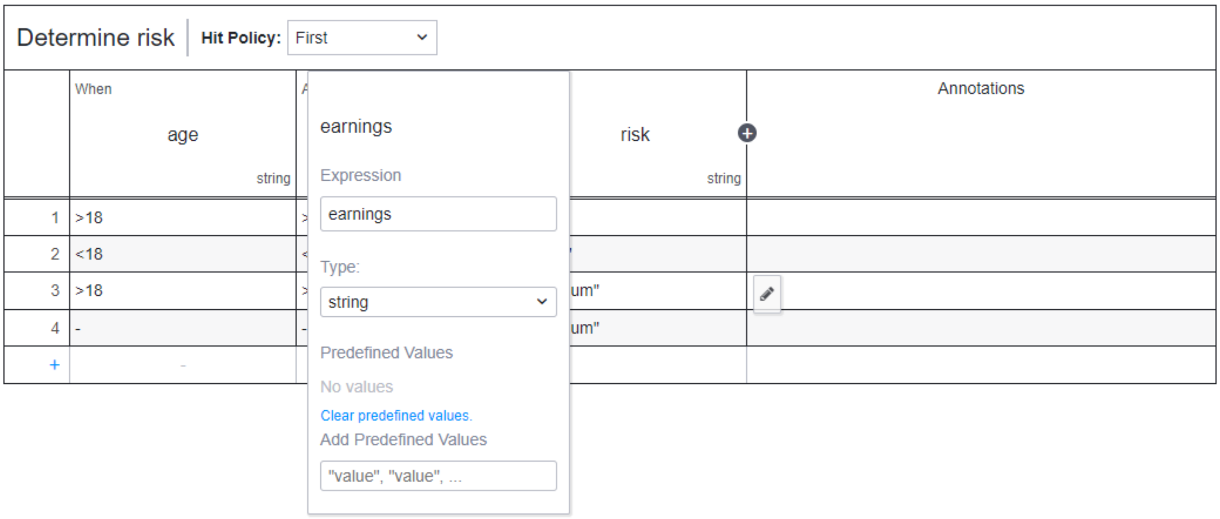
Type: string

Predefined Values

No values

Clear predefined values.

Add Predefined Values: "value", "value", ...



Determine risk | **Hit Policy:** First

earnings

Expression: earnings

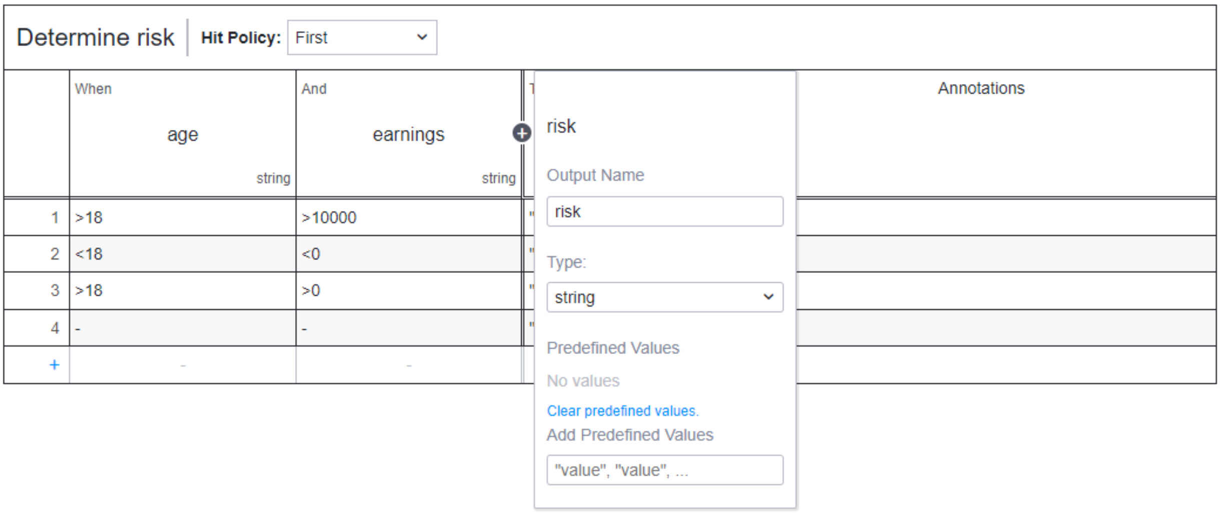
Type: string

Predefined Values

No values

Clear predefined values.

Add Predefined Values: "value", "value", ...



Determine risk | Hit Policy: First

	When age string	And earnings string	Annotations
1	>18	>10000	
2	<18	<0	
3	>18	>0	
4	-	-	
+	-	-	

risk

Output Name: risk

Type: string

Predefined Values

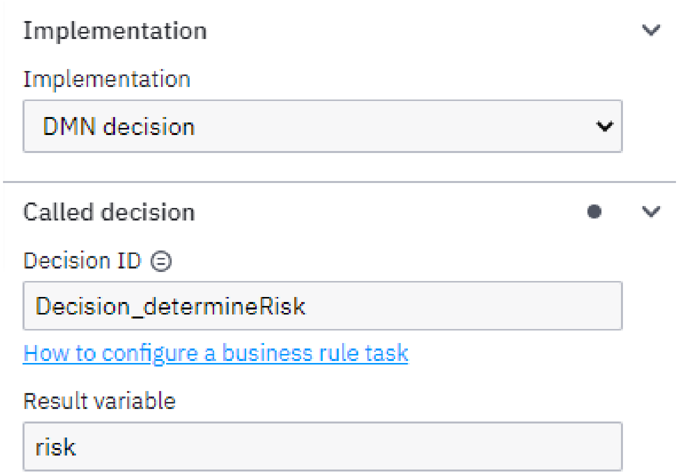
No values

Clear predefined values.

Add Predefined Values: "value", "value", ...

5. Deploy your DMN diagram. Click on **Deploy** button on the top right corner.

6. Connect DMN diagram with your BPMN diagram.
   a. Open **bank-account-extended** model and link Business Rule task to our created DMN model. Click on Determine risk business rule task and add below properties:



Implementation

Implementation: DMN decision

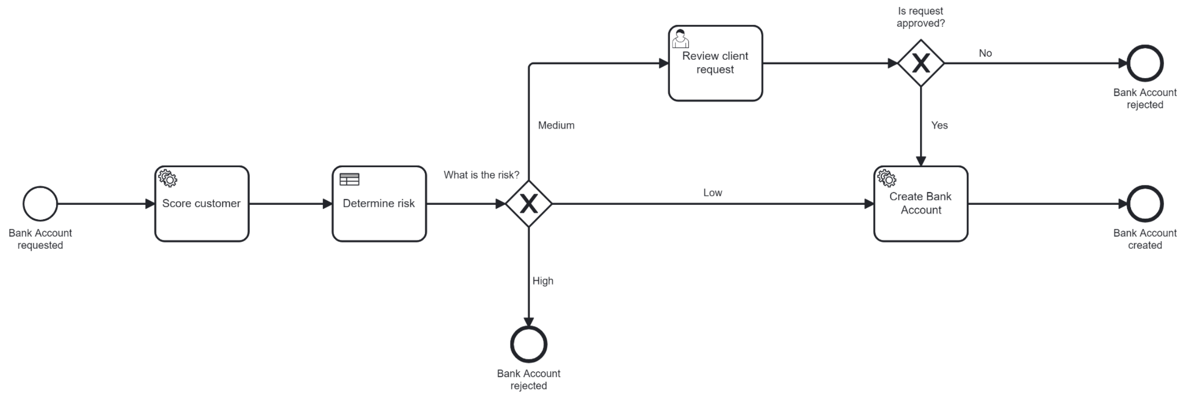
Called decision

Decision ID: Decision_determineRisk

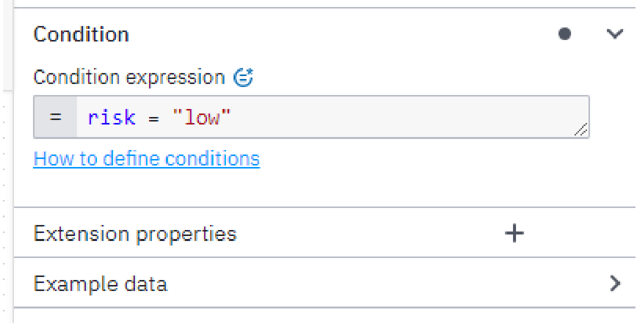
How to configure a business rule task

Result variable: risk

7. Next modify the exclusive gateway to have three outputs with Medium, High and Low risk. If risk is **Low** we want to send the account creation request directly to the microservice responsible for account creation. If the risk is **Medium**, we want the employee to review the request and then send it to the microservice responsible for account creation. If risk is **High** the account creation request should be rejected. Try to use the knowledge you gained up to this point and create the diagram as below:



8. You need to configure the output flows from the exclusive gateway. Use the following settings:



Condition

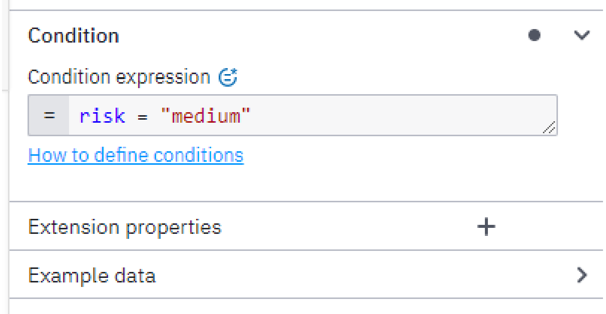
Condition expression

`= risk = "low"`

How to define conditions

Extension properties +

Example data >



Condition

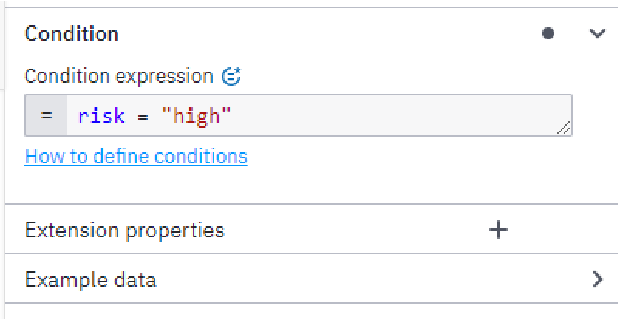
Condition expression

`= risk = "medium"`

How to define conditions

Extension properties +

Example data >



Condition

Condition expression

`= risk = "high"`

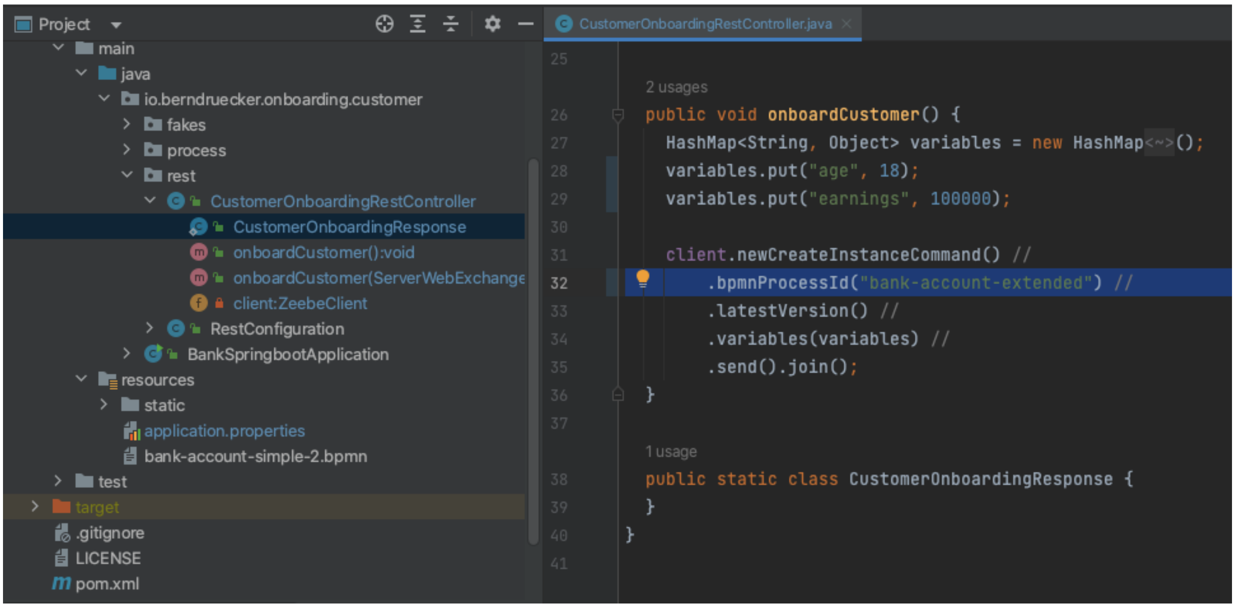
How to define conditions

Extension properties +

Example data >

9. Deploy BPMN diagram.

10. Update code in the Intellij to point to extended model (you changed the ID in point 1d).
    a. Open *<your_base_path>\customer-onboarding-camunda-8-springboot-master\src\main\java\io\camunda\workshop\customer\rest\CustomerOnboardingRestController.java*
    b. Update the line highlighted below:



```
main
  java
    io.berndruecker.onboarding.customer
      fakes
      process
      rest
        CustomerOnboardingRestController
          CustomerOnboardingResponse
          onboardCustomer():void
          onboardCustomer(ServerWebExchange
          client:ZeebeClient
        RestConfiguration
      BankSpringbootApplication
  resources
    static
    application.properties
    bank-account-simple-2.bpmn
test
target
.gitignore
LICENSE
pom.xml
```

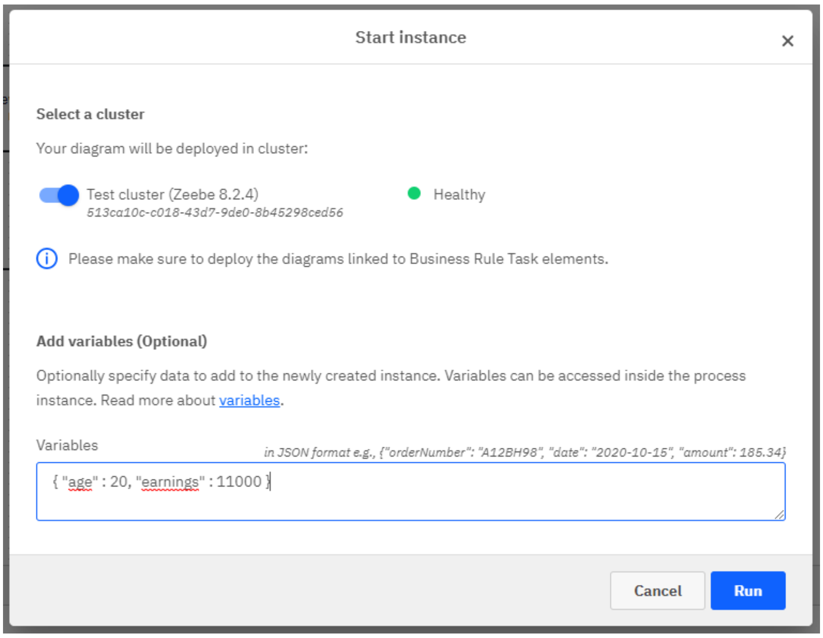
```
2 usages
public void onboardCustomer() {
  HashMap<String, Object> variables = new HashMap<~>();
  variables.put("age", 18);
  variables.put("earnings", 100000);

  client.newCreateInstanceCommand() //
      .bpmnProcessId("bank-account-extended") //
      .latestVersion() //
      .variables(variables) //
      .send().join();
}

1 usage
public static class CustomerOnboardingResponse {
}
}
```

.

11. Create exemplary instance by clicking on ▶ Run ⌄
    Pass input variables (exemplary below):

```
{ "age" : 20, "earnings" : 11000 }
```

as those will be used for determining the risk by business rule.



Start instance ×

Select a cluster

Your diagram will be deployed in cluster:

Test cluster (Zeebe 8.2.4) 513ca10c-c018-43d7-9de0-8b45298ced56 ● Healthy

ⓘ Please make sure to deploy the diagrams linked to Business Rule Task elements.

Add variables (Optional)

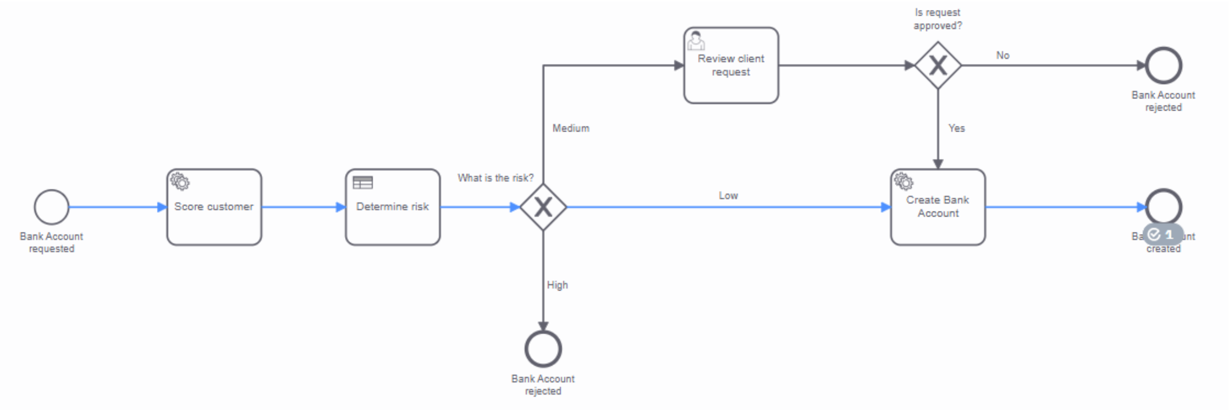
Optionally specify data to add to the newly created instance. Variables can be accessed inside the process instance. Read more about variables.

Variables *in JSON format e.g., {"orderNumber": "A12BH98", "date": "2020-10-15", "amount": 185.34}*

{ "age" : 20, "earnings" : 11000 }

Cancel Run

12. Run your application in Intellij - follow point 5 from section IV.



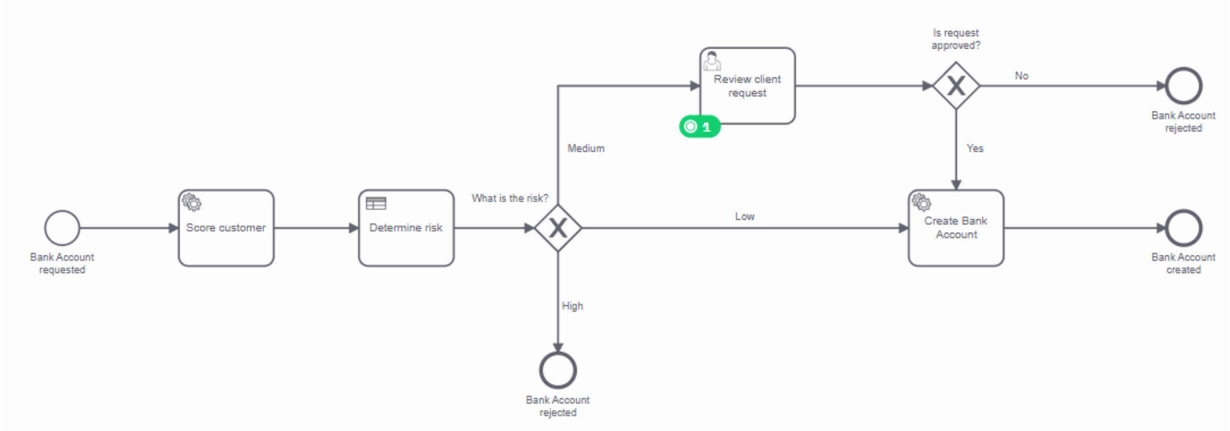
Your instance should be completed successfully with a Bank Account created automatically, without human actors involved.

13. Now create an instance from your Java project. Follow the steps from point 11 in section IV and run the following command or open http://localhost:8080/ the link in your browser and click on Open bank account.

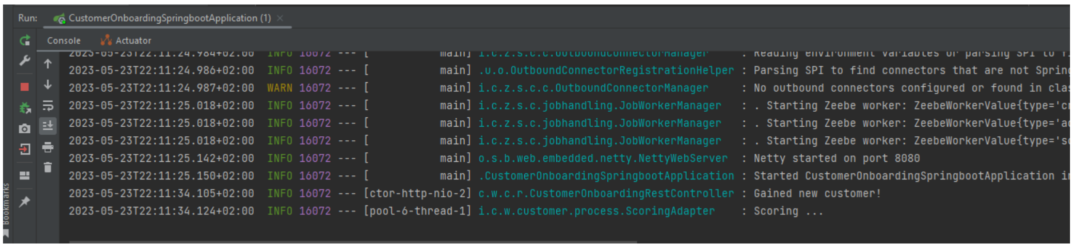
```
curl -X PUT http://localhost:8080/customer
```

14. You should see the token that is waiting for the human actor to review the request as we created instance with the variables:

```
{ "age" : 20, "earnings" : 8000 }
```



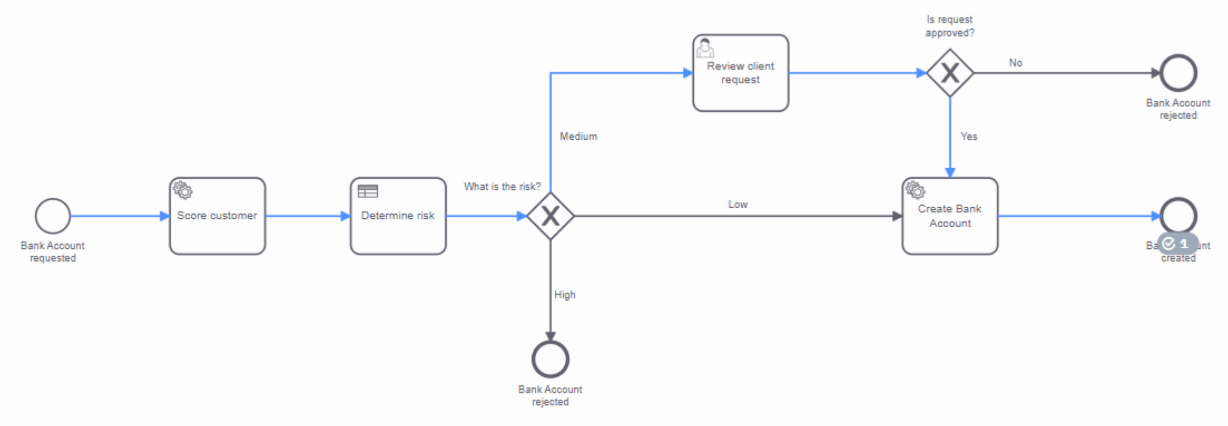
Logs in the terminal will look as follows:



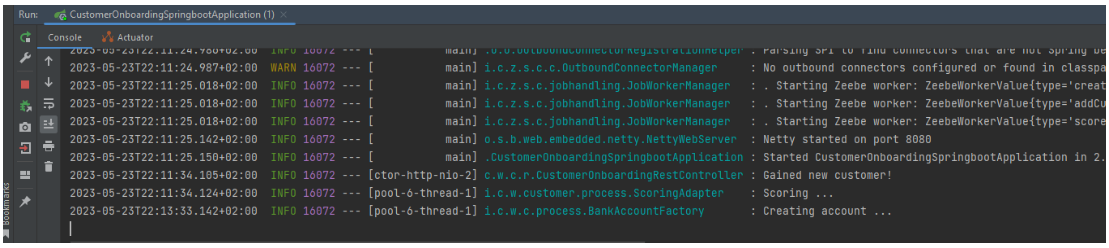
```
2023-05-23T22:11:24.984+02:00  INFO 16072 --- [           main] i.c.z.s.c.c.OutboundConnectorManager     : Reading environment variables or parsing SPI to f
2023-05-23T22:11:24.986+02:00  INFO 16072 --- [           main] .u.o.OutboundConnectorRegistrationHelper : Parsing SPI to find connectors that are not Sprin
2023-05-23T22:11:24.987+02:00  WARN 16072 --- [           main] i.c.z.s.c.c.OutboundConnectorManager     : No outbound connectors configured or found in clas
2023-05-23T22:11:25.018+02:00  INFO 16072 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='cr
2023-05-23T22:11:25.018+02:00  INFO 16072 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='ac
2023-05-23T22:11:25.018+02:00  INFO 16072 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='sc
2023-05-23T22:11:25.142+02:00  INFO 16072 --- [           main] o.s.b.web.embedded.netty.NettyWebServer  : Netty started on port 8080
2023-05-23T22:11:25.150+02:00  INFO 16072 --- [           main] .CustomerOnboardingSpringbootApplication : Started CustomerOnboardingSpringbootApplication in
2023-05-23T22:11:34.105+02:00  INFO 16072 --- [ctor-http-nio-2] c.w.c.r.CustomerOnboardingRestController : Gained new customer!
2023-05-23T22:11:34.124+02:00  INFO 16072 --- [pool-6-thread-1] i.c.w.customer.process.ScoringAdapter    : Scoring ...
```

15. Go to the task list - click on Camunda apps -> *Tasklist* tab and approve the **User Task**.

16. This time your instance will result in creation or rejection of the account. Below is the example where the user task was approved successfully:



You can also see in the Java application logs that account has been created:



```
2023-05-23T22:11:24.986+02:00  INFO 16072 --- [           main] .o.o.OutboundConnectorRegistrationHelper : Parsing SPI to find connectors that are not Spring be
2023-05-23T22:11:24.987+02:00  WARN 16072 --- [           main] i.c.z.s.c.c.OutboundConnectorManager     : No outbound connectors configured or found in classpa
2023-05-23T22:11:25.018+02:00  INFO 16072 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='creat
2023-05-23T22:11:25.018+02:00  INFO 16072 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='addCu
2023-05-23T22:11:25.018+02:00  INFO 16072 --- [           main] i.c.z.s.c.jobhandling.JobWorkerManager   : . Starting Zeebe worker: ZeebeWorkerValue{type='score
2023-05-23T22:11:25.142+02:00  INFO 16072 --- [           main] o.s.b.web.embedded.netty.NettyWebServer  : Netty started on port 8080
2023-05-23T22:11:25.150+02:00  INFO 16072 --- [           main] .CustomerOnboardingSpringbootApplication : Started CustomerOnboardingSpringbootApplication in 2.
2023-05-23T22:11:34.105+02:00  INFO 16072 --- [ctor-http-nio-2] c.w.c.r.CustomerOnboardingRestController : Gained new customer!
2023-05-23T22:11:34.124+02:00  INFO 16072 --- [pool-6-thread-1] i.c.w.customer.process.ScoringAdapter    : Scoring ...
2023-05-23T22:13:33.142+02:00  INFO 16072 --- [pool-6-thread-1] i.c.w.c.process.BankAccountFactory       : Creating account ...
```

17. Play around, create instances with different input variables and check behaviour.