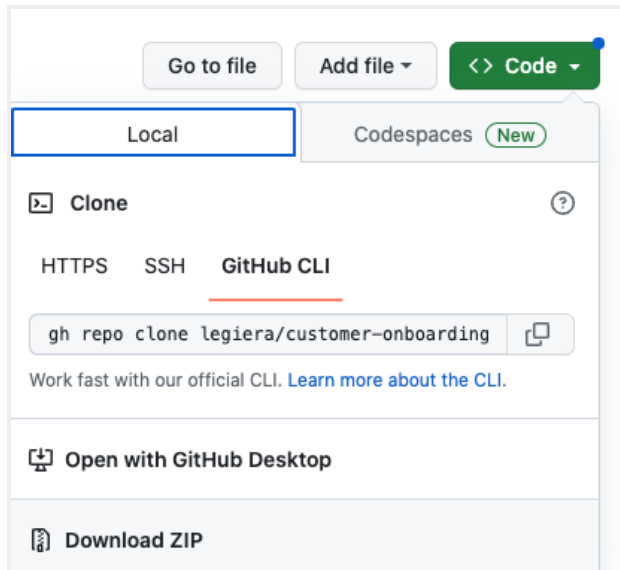


Camunda workshop instruction

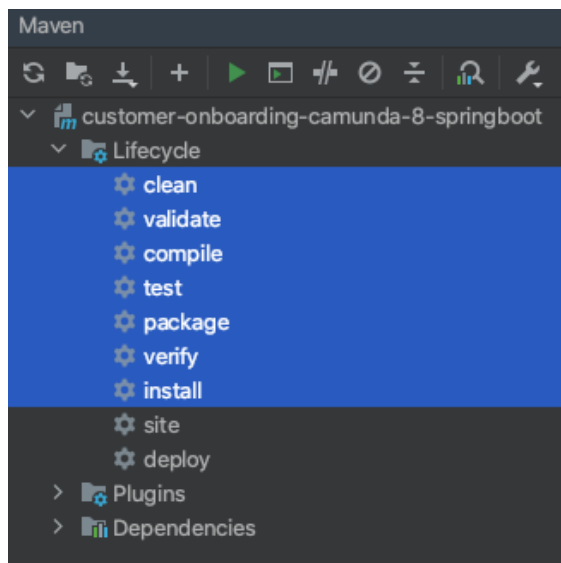
I Clone and build project

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.



2. Open project in IntelliJ. Projects> Open> select folder with downloaded project.
3. 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**:



II Camunda Cloud

1. Create a free account on Camunda Cloud.

- Go to <https://camunda.com/platform/> and click
- Provide your details:

Try Free

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.

Sign Up

or sign up with:

 Google

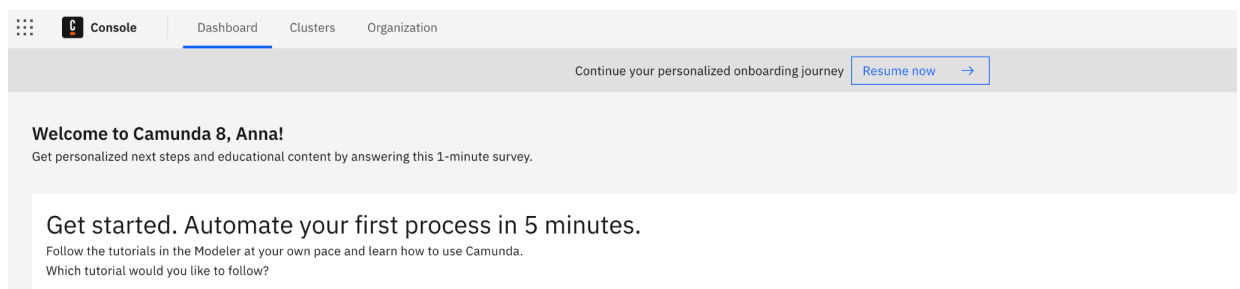
 Github

Want to host Camunda Platform on your own infrastructure?
[Download Self-Managed](#)

- You should receive confirmation on your email. Check your email and verify your account.
- Login on Camunda Cloud: <https://camunda.com/platform/>

2. Create Cluster on Camunda Cloud

- In the menu on the top **select Clusters** and then **Create new cluster**:



- Then click
- Provide all details for your cluster as below and then click

Create cluster

Clusters /

Create a cluster

Cluster name

workshop

Select a cluster type

Trial Cluster

Select region

☐ Iowa, North America (us-central1)
 ☐ Toronto, North America (northamerica-northeast2)
 ☐ Sydney, Australia (australia-southeast1)
 ☒ South Carolina, North America (us-east1)
 ☒ Belgium, Europe (europe-west1)

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

Channel

Alpha

Stable

The Alpha channel provides preview releases in preparation for the next stable release. They provide a short-term-stable point to test new features and give feedback before they are released to the stable channel. Try these to make sure the upcoming release works well in your infrastructure. These releases cannot be updated to a newer release and therefore are not meant to be used in production.

Generation

☒ Zeebe 8.3.0-alpha 1

Summary

1 x Trial Cluster
 Region
 Belgium, Europe (europe-west1)
 Channel
 Alpha
 Generation
 Zeebe 8.3.0-alpha 1

Create cluster

Cancel

Need help? [Talk to an expert](#)

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

Clusters

Name	Region	Generation	Type	Status
workshop	Belgium, Europe (europe-west1)	Zeebe 8.3.0-alpha 1	Trial Cluster	● Creating

3. Create Client Credential

a. Go to the Clusters tab and click the name of your cluster. You should see below details:

workshop

Overview

API

Alerts

Activity

Connector Secrets

Applications

Zeebe

The engine that powers Camunda 8

● Creating

Operate

Monitor and fix running processes

● Creating

Tasklist

Assign & execute user tasks

● Creating

Optimize

Get insights to improve your processes

● Creating

Check out our [User Guide](#) for more information.

Cluster Details

Cluster Id

88a8b59d-afa1-47c5-81ee-1423d8ecc911

Created At

May 18, 2023 at 02:30 AM

Region

Belgium, Europe (europe-west1)

Type

Trial Cluster

Generation

Zeebe 8.3.0-alpha 1

- b. Then select **API tab** and click: Create your first Client +
- 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

Cancel Create

- 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.

Spring Boot Spring Boot YAML Env Vars Desktop Modeler

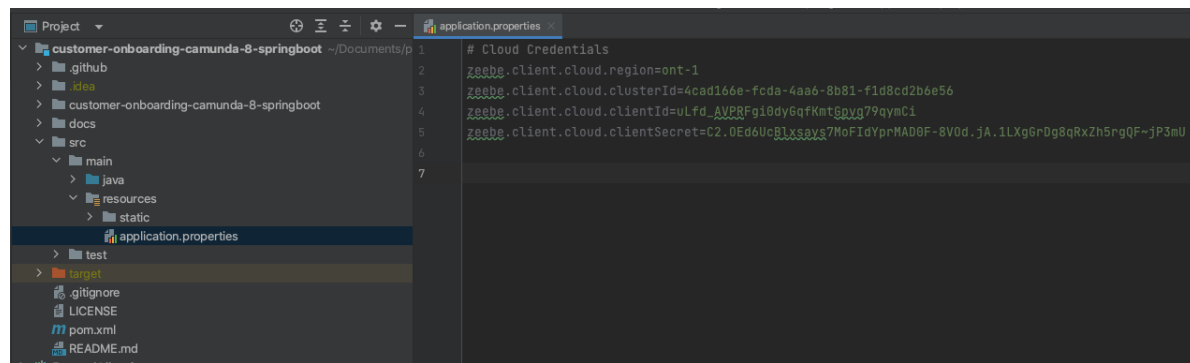
This format is designed to be used with the [Spring Zeebe](#) as part of your Java Spring Boot application or our Microservice Orchestration tutorial.

```
zeebe.client.cloud.region=bru-2
zeebe.client.cloud.clusterId=88a8b59d-afa1-47c5-81ee-1423d8ecc911
zeebe.client.cloud.clientId=qLVkPuPl2MdkGBWEd1Id13JC1if47J5M
zeebe.client.cloud.clientSecret=3rySKBfmRtdG0PopDmsgxkLKStAjW_CF4H11ew.LyryZK1o58JUxLveMZkutv9f3
```

Download 

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

4. Go to your project in IntelliJ and update application.properties with your details:



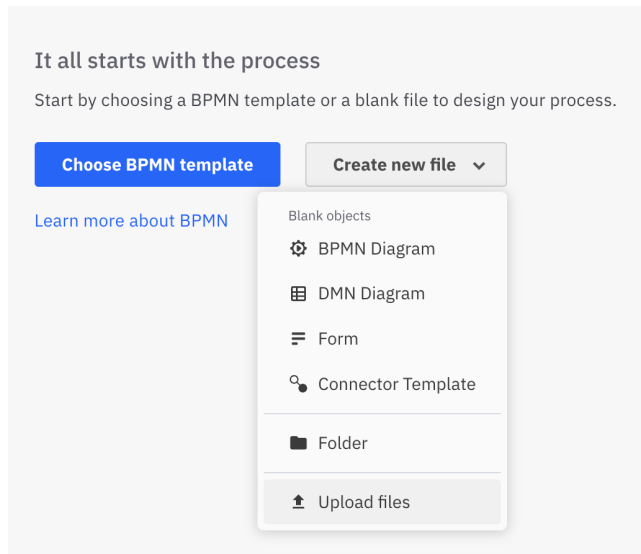
The screenshot shows the IntelliJ IDEA interface. On the left, the Project tool window displays the file structure of a project named 'customer-onboarding-camunda-8-springboot'. The 'src/main/resources' directory is expanded, and 'application.properties' is selected. The main editor area on the right shows the content of 'application.properties' with the following text:

```
1 # Cloud Credentials
2 zeebe.client.cloud.region=ont-1
3 zeebe.client.cloud.clusterId=4cad166e-fcda-4aa6-8b81-f1d8cd2b6e56
4 zeebe.client.cloud.clientId=uLfd_AVP8Fg18dy6qfKmt6pyg79qymC1
5 zeebe.client.cloud.clientSecret=C2.0Ed6UcB1xsays7MoFIdYprMAD0F-8V0d.JA.1LXgGrDg8qRxZh5rgQF~jP3mU
6
7
```

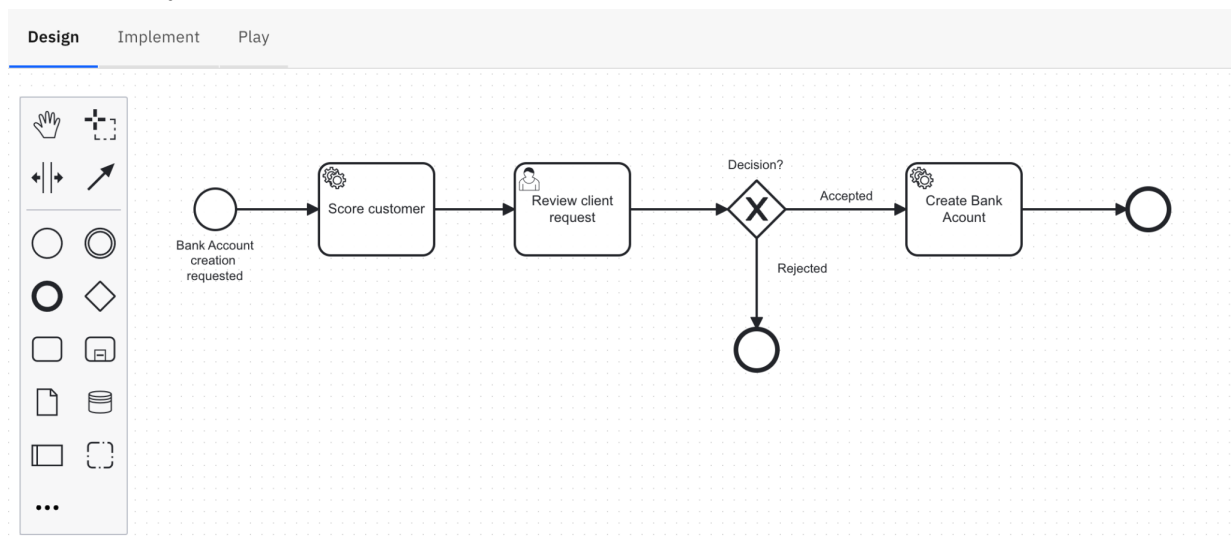
III First Exercise - Simple Bank Account System


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. In the project find **bank-account-simple.bpmn**
2. In Camunda cloud go to Modeler and create new project and select Create new file and upload bank-account-simple.bpmn:



3. Now you see our model in Modeler:



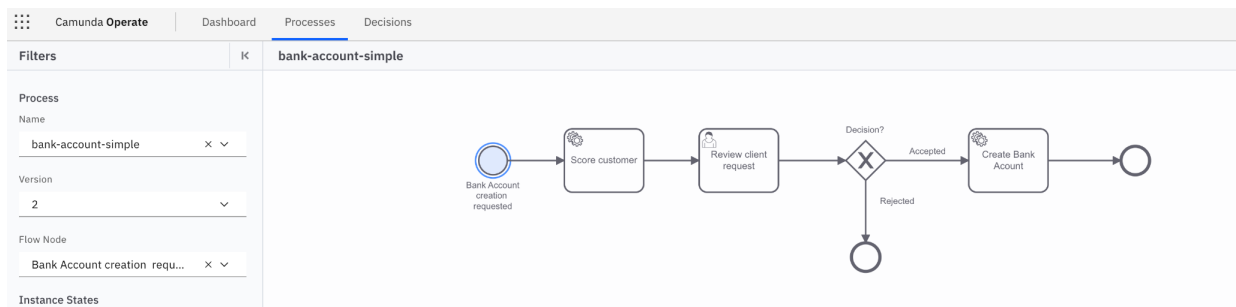
4. Then click Implement tab and then  Deploy
5. Then go to the Operate tab. You should see Instance of your process:

0 bank-account-simple – 0 Instances in 2 Versions 0

6. Run application. Right click on **BankSpringbootApplication** and then run. You should see below logs in the console:

```
2023-05-18T02:49:11.747+02:00 INFO 13438 --- [main] i.c.z.s.c.c.OutboundConnectorManager : Reading environment variables or parsing SPI to find connectors that are not Spring beans
2023-05-18T02:49:11.748+02:00 INFO 13438 --- [main] .u.o.OutboundConnectorRegistrationHelper : Parsing SPI to find connectors that are not Spring beans
2023-05-18T02:49:11.749+02:00 WARN 13438 --- [main] i.c.z.s.c.c.OutboundConnectorManager : No outbound connectors configured or found in classpath
2023-05-18T02:49:11.773+02:00 INFO 13438 --- [main] i.c.z.s.c.c.jobhandling.JobWorkerManager : . Starting Zeebe worker: ZeebeWorkerValue{type='createNewBankAccount', name='customerOnboarding6LueCode#createNewBankAccount', timeout=-1, maxJ
2023-05-18T02:49:11.774+02:00 INFO 13438 --- [main] i.c.z.s.c.c.jobhandling.JobWorkerManager : . Starting Zeebe worker: ZeebeWorkerValue{type='scoreCustomer', name='scoringAdapter#scoreCustomer', timeout=-1, maxJ
2023-05-18T02:49:11.817+02:00 INFO 13438 --- [main] o.s.b.w.e.n.n.NettyWebServer : Netty started on port 8080
2023-05-18T02:49:11.826+02:00 INFO 13438 --- [main] i.b.o.c.BankSpringbootApplication : Started BankSpringbootApplication in 2.364 seconds (process running for 2.679)
```

7. Click on it. Instance of our process should be the same as in the Modeler:



8. In the second tab go to our model in Modeler.

9. Select **Implement Tab** and click  **Run** 



10. To start instance provide variable 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

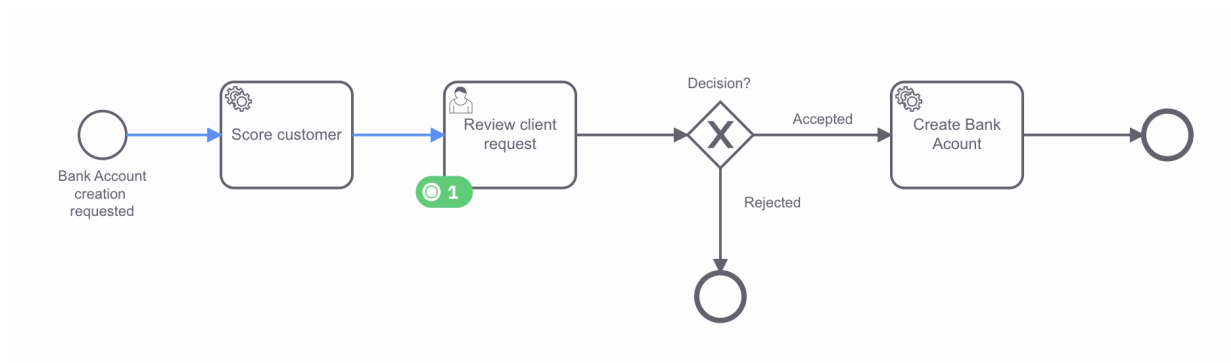
in JSON format e.g., {"orderNumber": "A12BH98", "date": "2020-10-15", "amount": 185.34}

`{"automaticProcess": "true"}`

Cancel

Run

11. Now go again to your Instance. Now you should see new event that we created:



12. Go to Tasklist in the main menu. You should see the Review **client request** task.

Filter options
All open
Review client request
bank-account-simple
Unassigned
Created
18 May 2023 - 03:10 AM

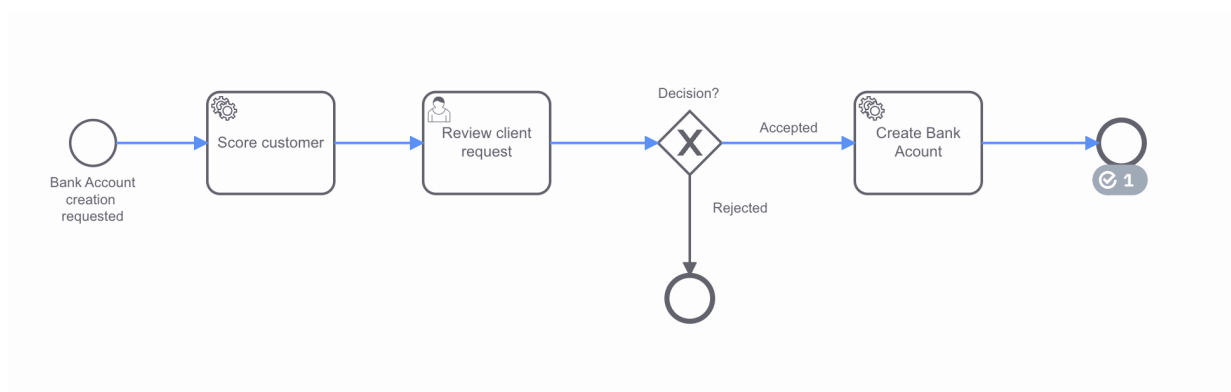
Review client request
bank-account-simple
☐ Should we create a bank account for this Client?
Can this request be processed automatically without further manual interaction?

13. Click on that new task and choose:

14. Select box and Complete Task:

Review client request
bank-account-simple
☒ Should we create a bank account for this Client?
Can this request be processed automatically without further manual interaction?

15. Go back to your instance and now you should see that token is processed and account is created:



16. Check also logs in the IntelliJ. What you can see? What happened?

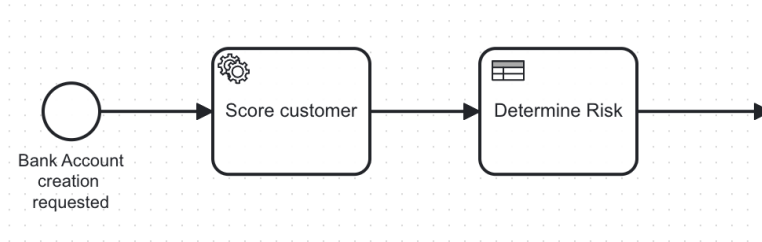
17. Now lets create event from our machine: **curl -X PUT http://localhost:8080/customer**

18. We should see the same effect as before.

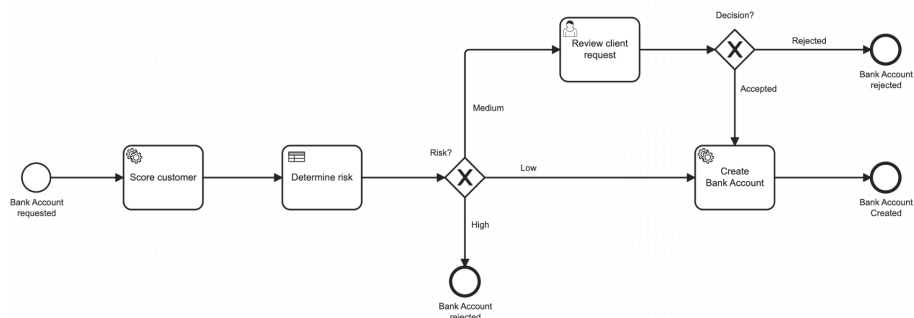
IV Second exercise - extended Bank System

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.** If not, then 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 bank account creation request is sent to Create Bank Account service. Otherwise, the request is rejected and the bank account is not created.

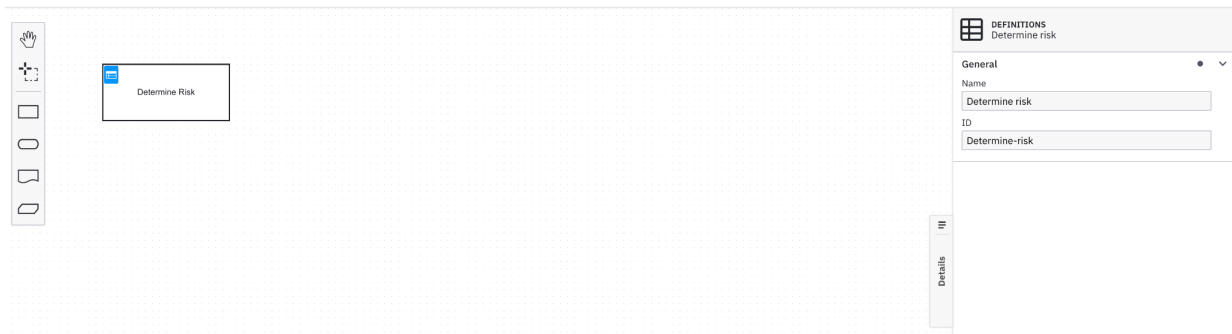
1. Let's start with the BPMN model. Let's modify our existing model.
2. Go to Modeler, find the bank-account-simple model and duplicate it and name it bank-account-extended.
3. Open the model in Modeler, then click **...** in the edit menu on the left and search for **Business Rule Task**. Add it to your model behind **Score Customer Service Task** with name **Determine risk and ID TaskValidateCustomerApplication**




4. 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.




5. Create New DMN Diagram in Modeler with name Determine risk and id determine-risk




6. Then click on  to define your logic:



Determine Risk Hit Policy: Unique				
	When	And	Then	Annotations
	age string	earnings string	risk string	
1	>18	>10000	low	
2	<18	<0	high	
3	>18		medium	
4	-	-		
+	-	-		

7. In bank-account-extended Model link Business Rule task to our created DMN. Add below properties:

Implementation


Implementation

DMN decision
 

Called decision



Decision ID ⓘ

determine-risk

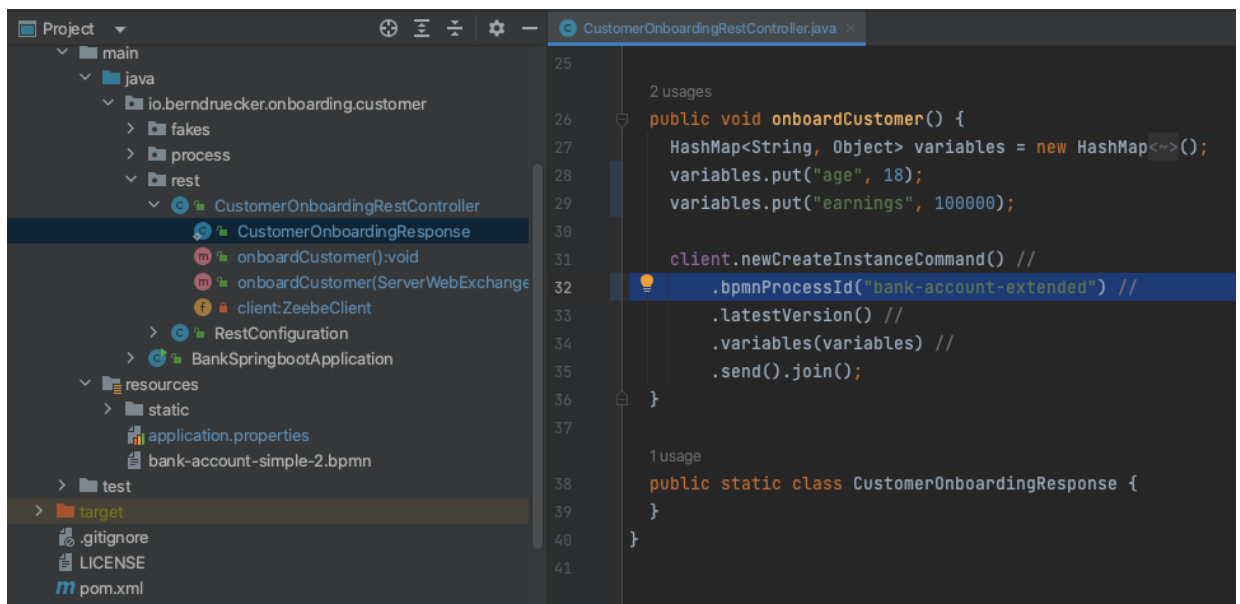
[How to configure a business rule task](#)

Result variable

risk

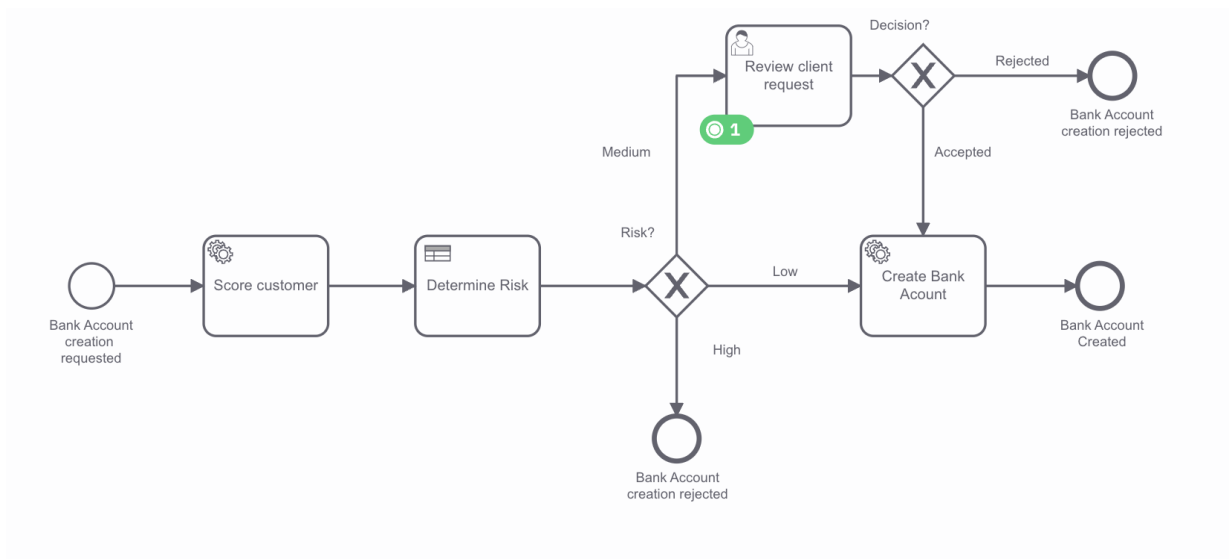
8. Deploy Determine Risk DMN

9. Update code in the IntelliJ



10. Run curl command : **curl -X PUT http://localhost:8080/customer**

11. You should see toke:



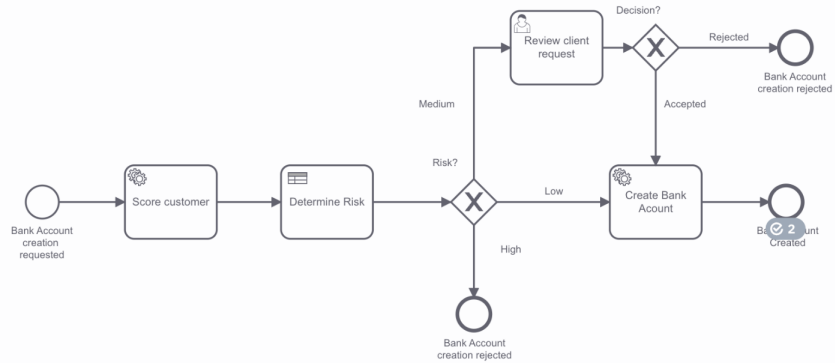
And logs in termina:

```
INFO 15690 --- [pool-6-thread-1] i.b.o.customer.process.ScoringAdapter : Gained new customer!
```

12. Go to the task list and approve the **User Task**.

13. Account see created account in the logs and

```
INFO 15690 --- [pool-6-thread-1] i.b.o.c.p.CustomerOnboardingGlueCode : Create new bank account via REST [{"key":4503599627586411,"type":"createNewBankAccount"},  
INFO 15690 --- [ctor-http-nio-3] .b.o.c.f.CreateBankAccountRestController : REST API called
```



14. Play around, create different events with different variables and check behaviour.

```

{
  "components": [
    {
      "label": "Should we create a bank account for this Client?",
      "type": "checkbox",
      "id": "Field_1n7851c",
      "key": "automaticProcessing",
      "description": "Can this request be processed automatically without further manual interaction?"
    }
  ],
  "type": "default",
  "id": "Form_0w8g0i0",
  "schemaVersion": 4
}

```