Sumit Vadaviya

Camunda BPM Expert, Corporate Trainer

Email : sumit.vadaviya@trainosoft.com

I

n this session we will look at what is service task and how it functions ?.

****

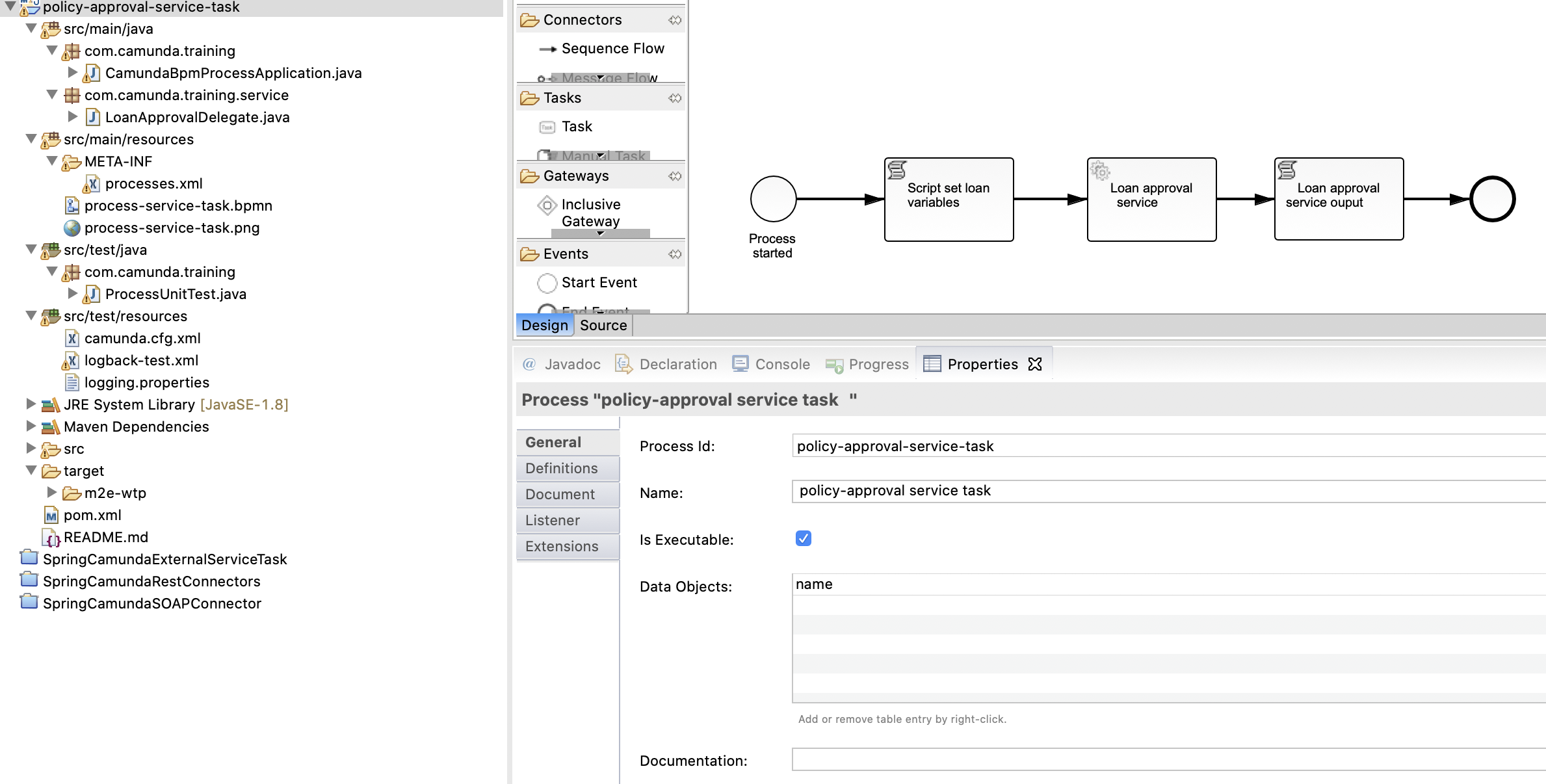
Service Tasks are used to perform some form of "technical" task. Whenever you want to perform calculations, trigger some action within your application’s back end or integrate with another system, it often involves Service Tasks.

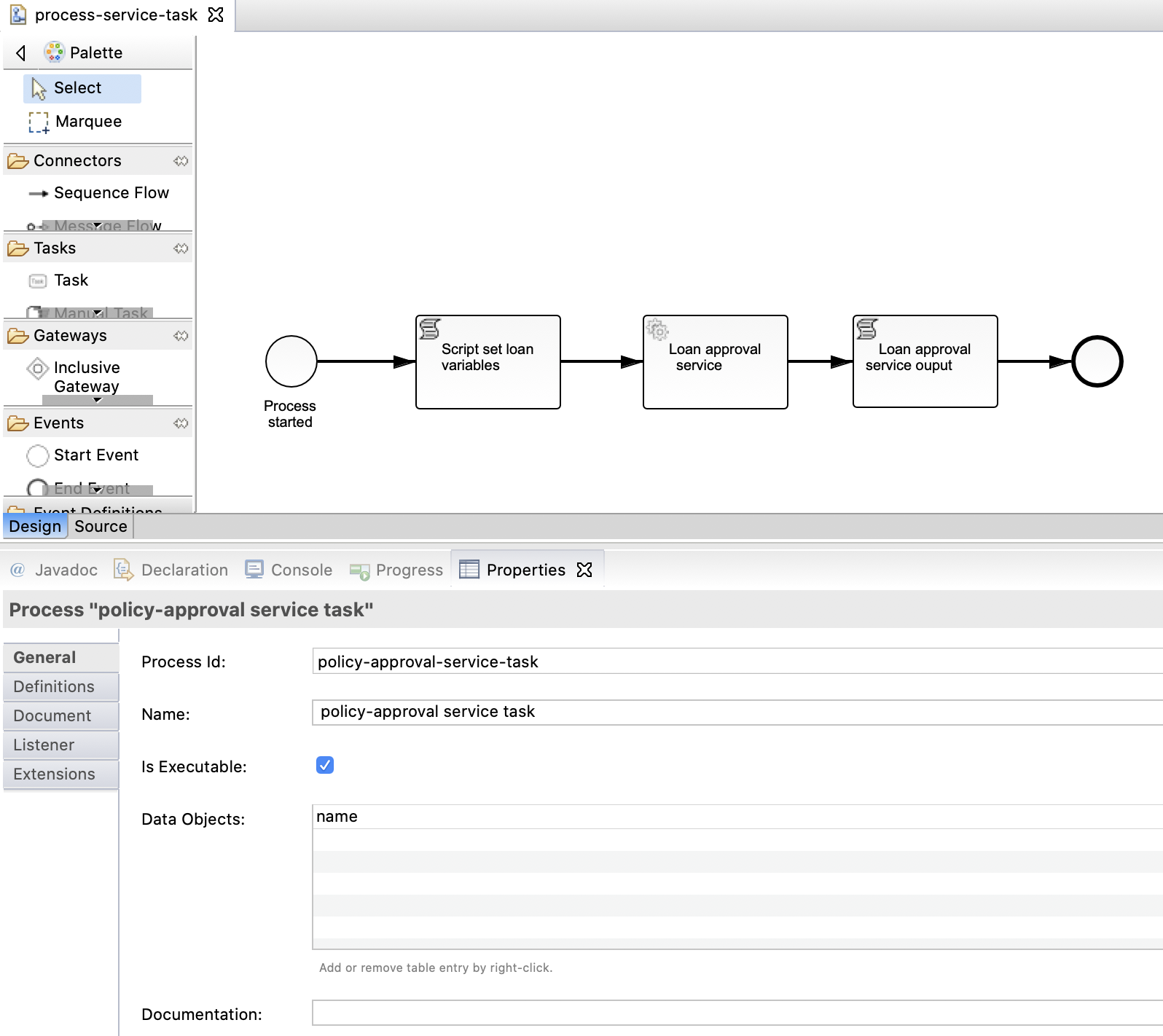
In many cases you want to store the result of an operation carried out by a Service Task in a variable. The resulting variable name is specified in the Result variable attribute.

Camunda Service Task

There are four ways of declaring how to invoke Java logic:

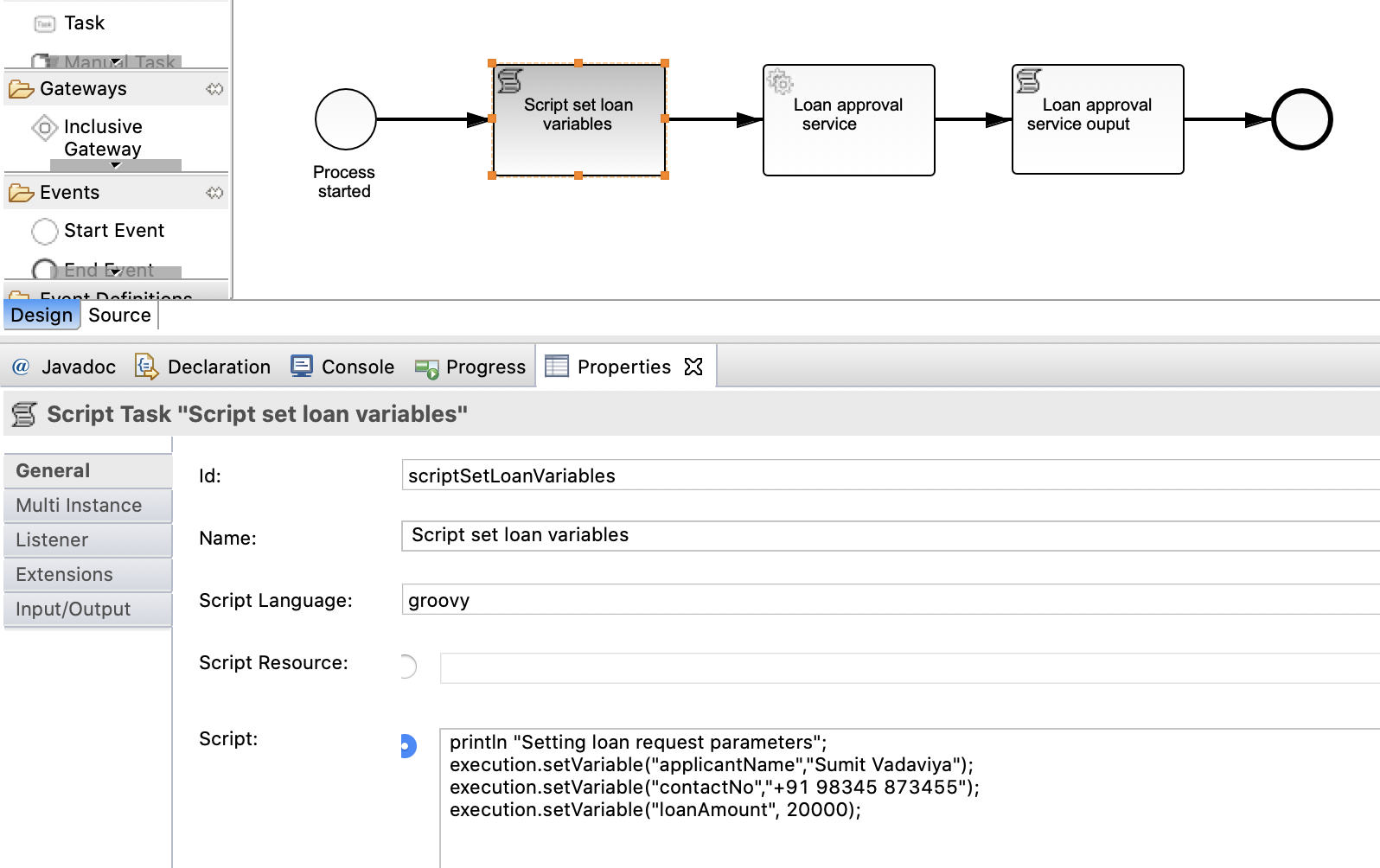
1. Specifying a class that implements a JavaDelegate or ActivityBehavior
2. Evaluating an expression that resolves to a delegation object
3. Invoking a method expression
4. Evaluating a value expression
5. **We will use process application as we had created on day 1 to demonstrate service task as a class that implements a JavaDelegate.**

****

**2. Process:** policy-approval service task

**Process Id:** policy-approval-service-task

**Name:** policy-approval service task

**3. Script task:** scriptSetLoanVariables****

**Id:** scriptSetLoanVariables

**Name:** Script set loan variables

**Script language:** groovy

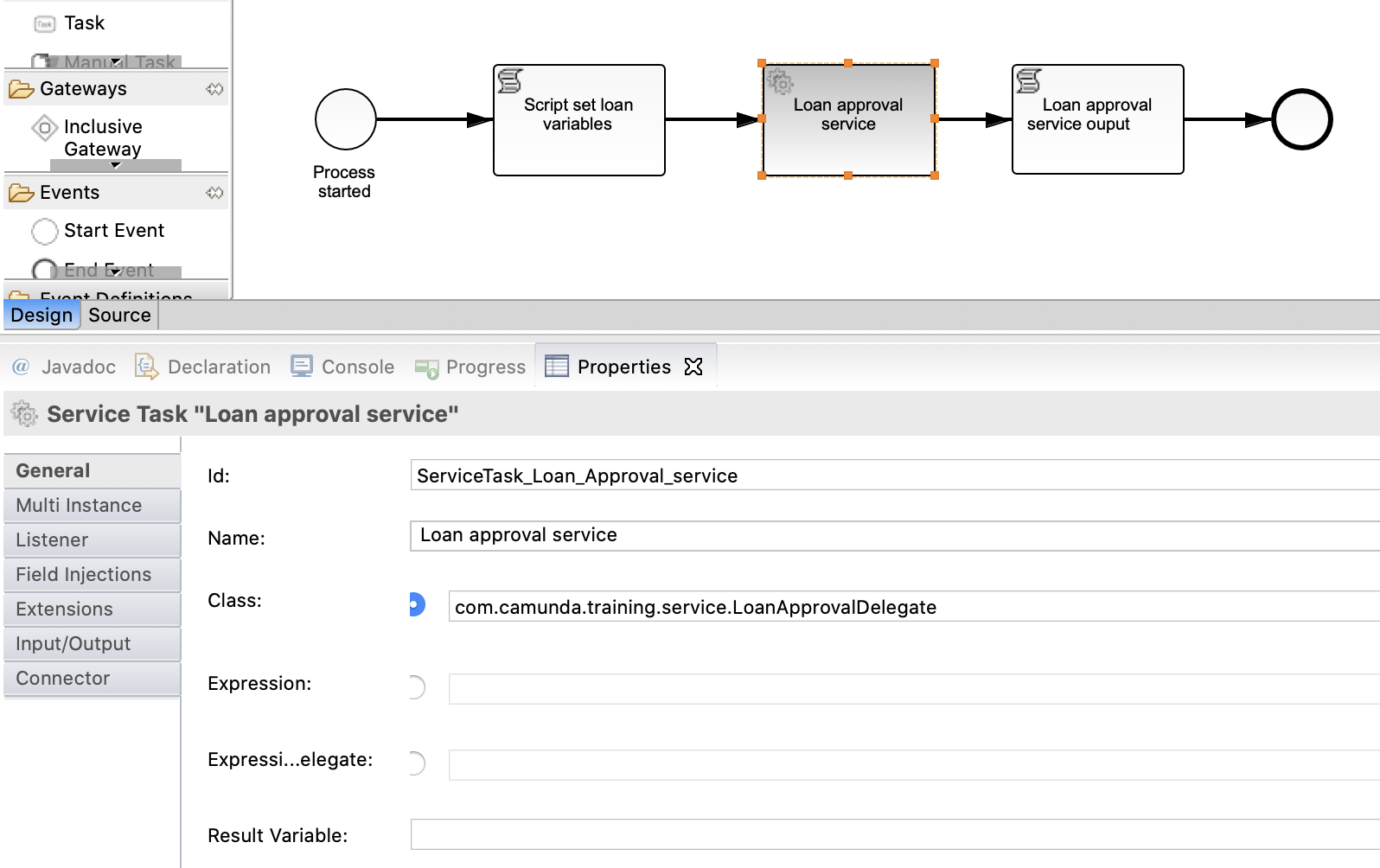
**Script:**

println "Setting loan request parameters";

execution.setVariable("applicantName","Sumit Vadaviya");

execution.setVariable("contactNo","+91 98345 873455");

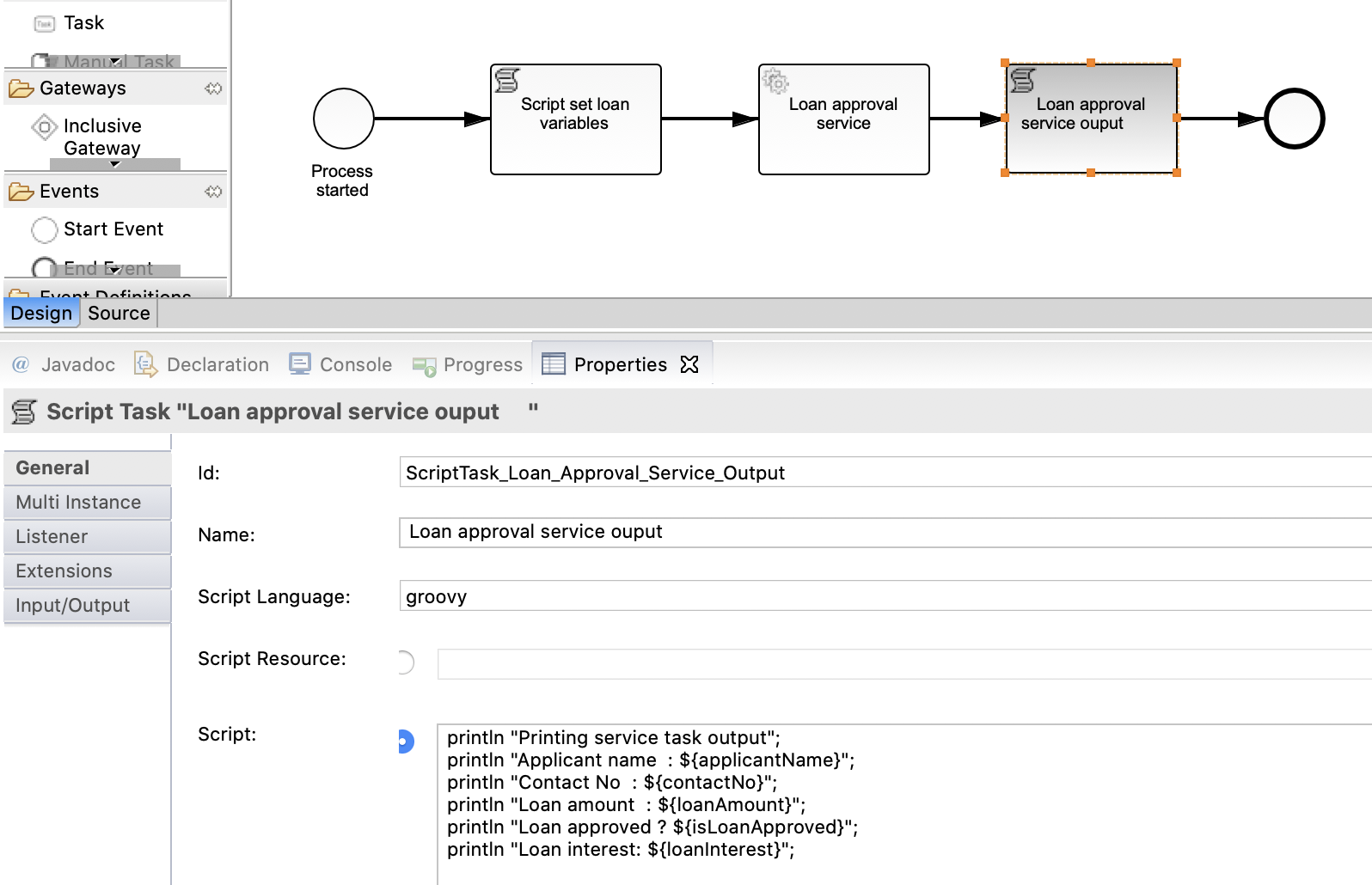
execution.setVariable("loanAmount", 20000);

4**. Service task:** scriptSetLoanVariables

**Id:** ServiceTask\_Loan\_Approval\_service

**Name:** Loan approval service

**Class:** com.camunda.training.service.LoanApprovalDelegate

**5. Script task:** ScriptTask\_Loan\_Approval\_Service\_Output

**Id:** ScriptTask\_Loan\_Approval\_Service\_Output

**Name:** Loan approval service ouput

**Script language:** groovy

**Script:**

println "Printing service task output";

println "Applicant name : ${applicantName}";

println "Contact No : ${contactNo}";

println "Loan amount : ${loanAmount}";

println "Loan approved ? ${isLoanApproved}";

println "Loan interest: ${loanInterest}”;

**Service task class implementation**

**package** com.camunda.training.service;

**import** java.util.logging.Logger;

**import** org.camunda.bpm.engine.delegate.DelegateExecution;

**import** org.camunda.bpm.engine.delegate.JavaDelegate;

/\*\*

\* This is an easy adapter implementation illustrating how a Java Delegate can

\* be used from within a BPMN 2.0 Service Task.

\*/

**public** **class** LoanApprovalDelegate **implements** JavaDelegate {

**private** **final** Logger LOGGER = Logger.*getLogger*(LoanApprovalDelegate.**class**.getName());

**public** **void** execute(DelegateExecution execution) **throws** Exception {

LOGGER.info("\n\n ... LoanApprovalDelegate invoked by " + "processDefinitionId="

+ execution.getProcessDefinitionId() + ", activtyId=" + execution.getCurrentActivityId()

+ ", activtyName='" + execution.getCurrentActivityName() + "'" + ", processInstanceId="

+ execution.getProcessInstanceId() + ", businessKey=" + execution.getProcessBusinessKey()

+ ", executionId=" + execution.getId() + " \n\n");

String applicantName = (String) execution.getVariable("applicantName");

String contactNo = (String) execution.getVariable("contactNo");

Integer loanAmount = (Integer) execution.getVariable("loanAmount");

LOGGER.info("applicantName : " + applicantName);

LOGGER.info("contactNo : " + contactNo);

LOGGER.info("loanAmount : " + loanAmount);

**if** (loanAmount < 10000) {

execution.setVariable("isLoanApproved", **false**);

} **else** **if** (loanAmount >= 10000 && loanAmount < 50000) {

execution.setVariable("loanInterest", 9.0);

execution.setVariable("isLoanApproved", **true**);

} **else** **if** (loanAmount >= 50000 && loanAmount < 100000) {

execution.setVariable("loanInterest", 8.0);

execution.setVariable("isLoanApproved", **true**);

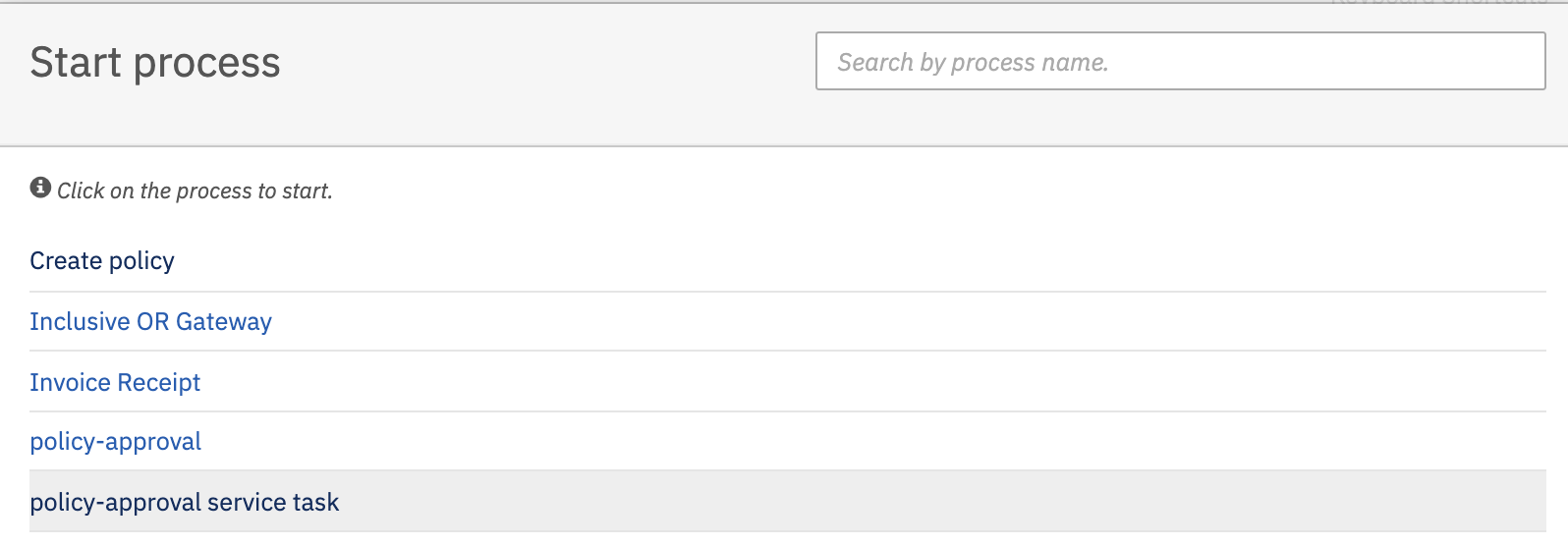
}

LOGGER.info("loanInterest : " + execution.getVariable("loanInterest"));

LOGGER.info("isLoanApproved : " + execution.getVariable("isLoanApproved"));

}

}

**Build the application and deploy**

Hit the Start button to start the process.

**Observe tomcat console output**

31-Aug-2020 09:00:50.965 INFO [Catalina-utility-1] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive [/Volumes/Trainings/Camunda/Softwares/camunda-bpm-tomcat-7.13.0/server/apache-tomcat-9.0.33/webapps/policy-approval-service-task.war] has finished in [3,327] ms

Setting loan request parameters

31-Aug-2020 09:02:09.953 INFO [http-nio-8081-exec-25] com.camunda.training.service.LoanApprovalDelegate.execute

... LoanApprovalDelegate invoked by processDefinitionId=24496e38-eba3-11ea-ba24-026b97bcf0fc, activtyId=ServiceTask\_Loan\_Approval\_service, activtyName='Loan approval service', processInstanceId=535a3a69-eba3-11ea-ba24-026b97bcf0fc, businessKey=null, executionId=535a3a69-eba3-11ea-ba24-026b97bcf0fc

31-Aug-2020 09:02:09.954 INFO [http-nio-8081-exec-25] com.camunda.training.service.LoanApprovalDelegate.execute applicantName : Sumit Vadaviya

31-Aug-2020 09:02:09.954 INFO [http-nio-8081-exec-25] com.camunda.training.service.LoanApprovalDelegate.execute contactNo : +91 98345 873455

31-Aug-2020 09:02:09.954 INFO [http-nio-8081-exec-25] com.camunda.training.service.LoanApprovalDelegate.execute loanAmount : 20000

31-Aug-2020 09:02:09.954 INFO [http-nio-8081-exec-25] com.camunda.training.service.LoanApprovalDelegate.execute loanInterest : 9.0

31-Aug-2020 09:02:09.954 INFO [http-nio-8081-exec-25] com.camunda.training.service.LoanApprovalDelegate.execute isLoanApproved : true

Printing service task output

Applicant name : Sumit Vadaviya

Contact No : +91 98345 873455

Loan amount : 20000

Loan approved ? true

Loan interest: 9.0

You can see script task **Script set loan variables** is executed by setting loan variables, after script task service task **ServiceTask\_Loan\_Approval\_service** is invoked from process and it has perform business logic execution to decide loan interest rate and then again script task **ScriptTask\_Loan\_Approval\_Service\_Output**  is executed to print the service task output.

**Service task implementation through expression**

It is also possible to use an expression that resolves to an object. This object must follow the same rules as objects that are created when the camunda:class attribute is used.

<serviceTask id="beanService"

name="My Bean Service Task"

camunda:delegateExpression="${myDelegateBean}" />

**Service task implementation through expression which calls method**

Or an expression which calls a method or resolves to a value.

<serviceTask id="expressionService"

name="My Expression Service Task"

camunda:expression="${myBean.doWork()}" />

We will demonstrate service task implementation through expression with camunda integration with Spring boot as it requires bean for service task implementation.