

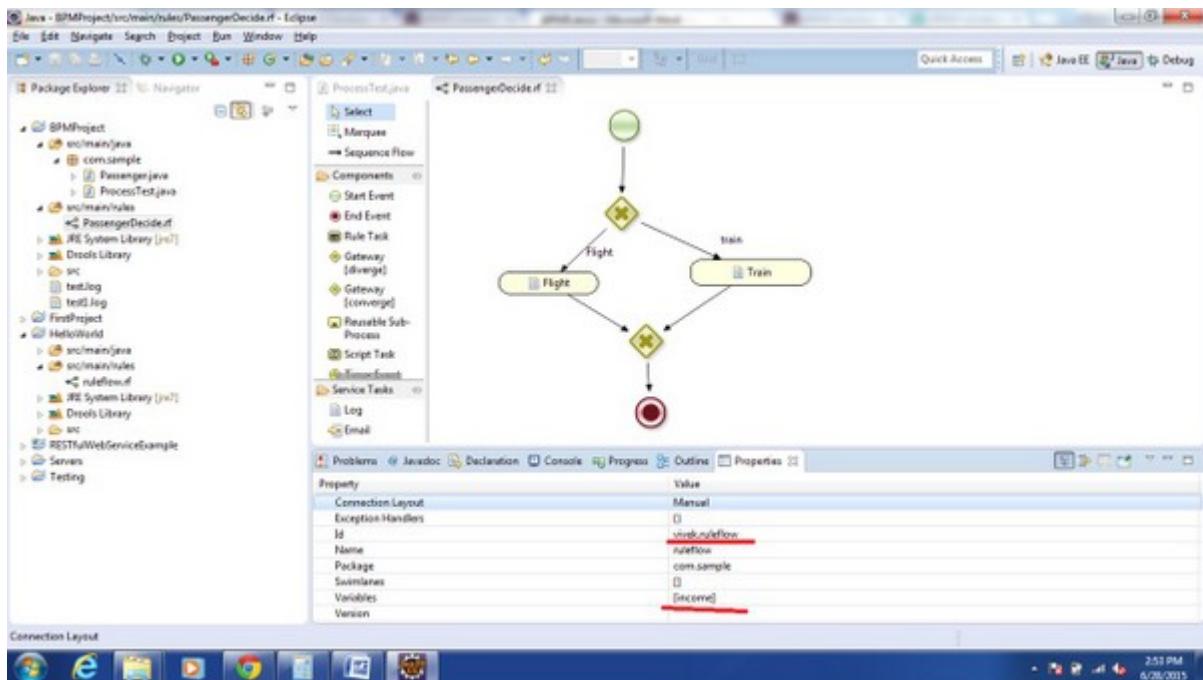
JBPM5 - EXAMPLE

http://www.tutorialspoint.com/jbpm5/jbpm5_example.htm

Copyright © tutorialspoint.com

We will take an example in this chapter to explain how to put jBPM into practice. The task at hand is to use jBPM to decide whether a passenger will board a flight or a train, depending upon his income.

Steps to create the project remains the same as that of "Hello World" project. Open the .rf file in the editor, and see the changes as marked in red –

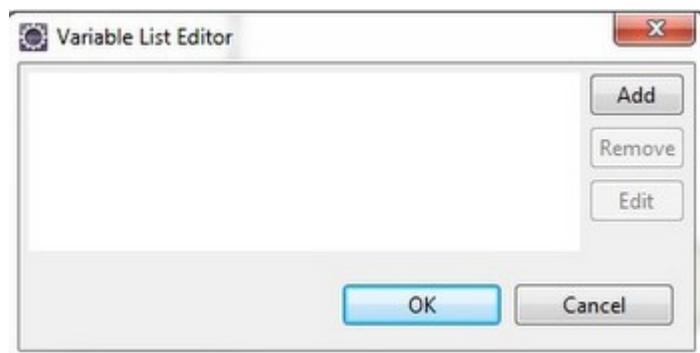


Here, we have changed the ruleflow id *unique* and added a variable income, as we will be deciding the route based on the income of the passenger.

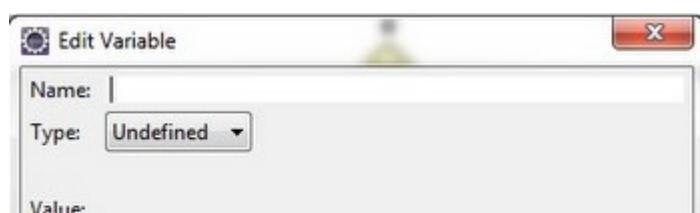
How to set variable – Select the edit button marked in red.



Click the **Add** button –

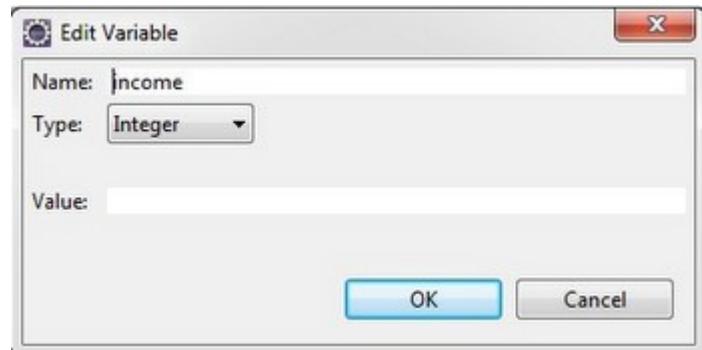


The following screen pops up –

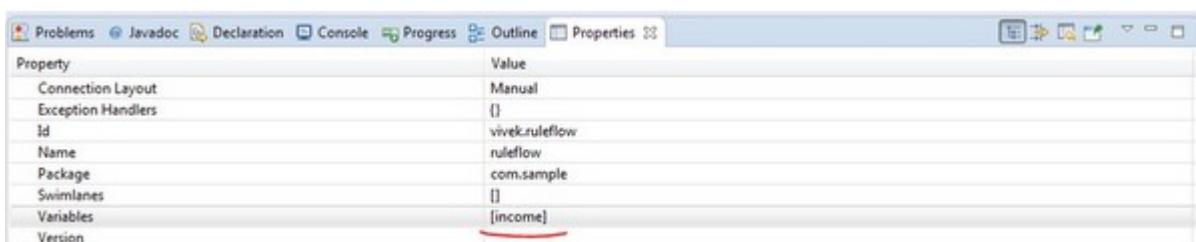




Put name as income and type as Integer.



Click **OK**. You will see the income variable defined now.



Changes to be done in ProcessTest.java class *Class to load and run your process* –

```

28  */
29 public class ProcessTest {
30
31     public static final void main(String[] args) {
32         try {
33             // load up the knowledge base
34             KnowledgeBase kbase = readKnowledgeBase();
35             StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();
36             KnowledgeRuntimeLogger logger = KnowledgeRuntimeLoggerFactory.newFileLogger(ksession, "test");
37             Map<String, Object> params = new HashMap<String, Object>();
38             params.put("income", 1200);
39             // start a new process instance
40             ksession.startProcess("vivek.ruleflow", params);
41             logger.close();
42         } catch (Throwable t) {
43             t.printStackTrace();
44         }
45     }
46 }

```

```

Map<String, Object> params = new HashMap<String, Object>();
params.put("income", 1200);

// start a new process instance
ksession.startProcess("vivek.ruleflow", params);

```

You need to create a MAP and set the value in MAP and pass this value to the startProcess overloaded method. As depicted, we have changed the ruleflow id and hence used the same id *vivek.ruleflow* in the startProcess method.

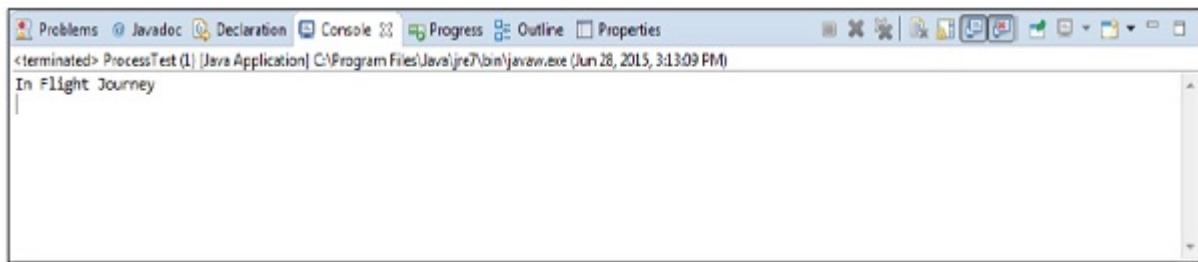
Before running the ProcessTest.java class, create a POJO Passenger.java in the same package as ProcessTest.java

```

public class Passenger {
    private String name;
    private int income;
    Getters and setters here
}

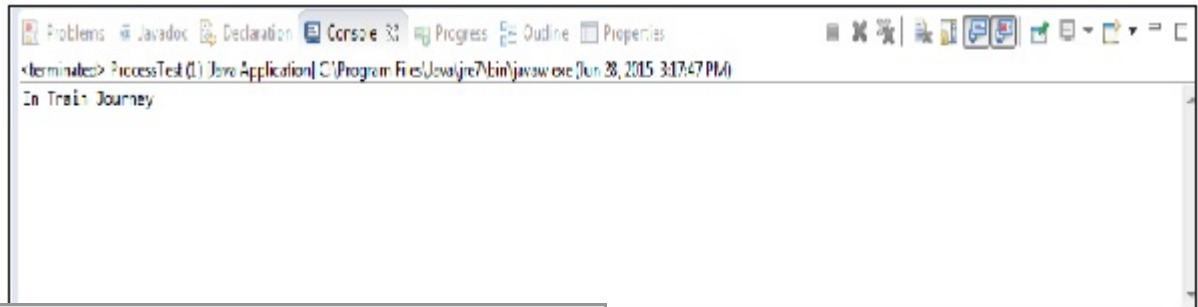
```

Now run the ProcessTest.java class as a Java Application and see the output –



The screenshot shows the Eclipse IDE's Console view. The title bar includes tabs for Problems, Javadoc, Declaration, Console, Progress, Outline, and Properties. The main content area displays the message: <terminated> ProcessTest (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Jun 28, 2015, 3:13:09 PM)
In Flight Journey

Change the value – `params.put "income" , 900;` and re-run the class and see the change in output.



The screenshot shows the Eclipse IDE's Console view. The title bar includes tabs for Problems, Javadoc, Declaration, Console, Progress, Outline, and Properties. The main content area displays the message: <terminated> ProcessTest (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Jun 28, 2015, 3:17:47 PM)
In Train Journey

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js