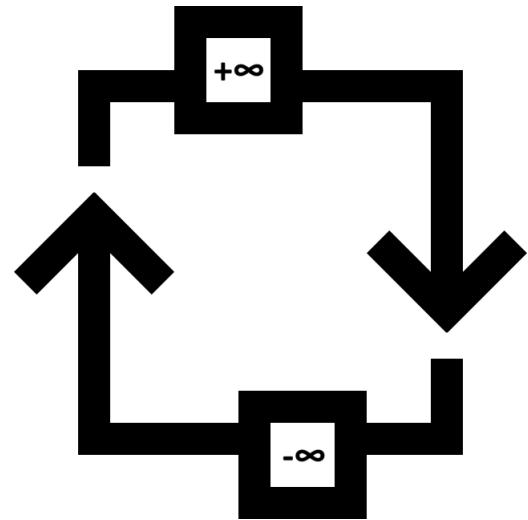


CONVERGENT ANALYTICS

AI ORCHESTRATION PLATFORM

VERSION: 1



TRACKING

EDITS

Edit	Date	Author(s)	Role(s)
Initial edit	2025-03-10	Sergey Lukyanchikov	Project Owner

APPROVALS

Approval	Date	Approver(s)	Signature(s)
Initial approval	2025-03-11	Sergey Lukyanchikov	NA

DISCLAIMER

This document provides information required for installation and use of AI Orchestration Platform. The functionality of AI Orchestration Platform is a template to be adjusted to the needs of the user. The user installs and applies AI Orchestration Platform at their own discretion and risk.

TABLE OF CONTENTS

General	4
Components installation.....	5
OpenJDK.....	5
PostgreSQL.....	5
Python.....	5
Jep	5
Eclipse	6
jBPM.....	8
BIRT (optional)	9
PATH	9
AI Orchestration Platform download, import and setup	10
Download.....	10
Import	12
Setup	15
AI Orchestration Platform customizing and testing.....	30
Customizing.....	30
Testing	31

GENERAL

[AI Orchestration Platform](#) is a bundle of open-source components configured to automate AI-centric client-server computations. AI Orchestration Platform is currently tested under the following components (a non-exhaustive list of the most important items):

- [jBPM \(jbpm-server-7.74.1.Final-dist\)](#) configured for PostgreSQL persistence
- [PostgreSQL \(postgresql-15.1-1-windows-x64\)](#)
- [Python \(python-3.6.7-amd64\)](#)
- [Jep \(jep-4.2.0.jar\)](#)
- [KIE API \(kie-api-7.74.1.Final.jar\)](#)
- [OpenJDK \(OpenJDK11U-jdk_x64_windows_hotspot_11.0.9.1_1\)](#)

IMPORTANT: the above links are indicative, the components available under those links are not controlled by us. Download at your discretion and risk.

COMPONENTS INSTALLATION

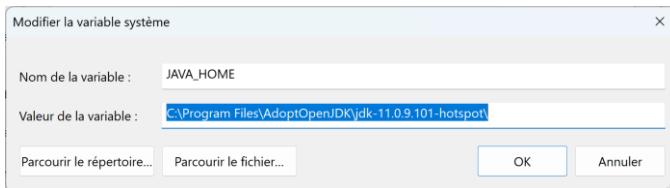
Before downloading and configuring AI Orchestration Platform, we must download and install open-source components on which AI Orchestration Platform depends:

OPENJDK

Download and installation instructions: [OpenJDK \(OpenJDK11U-jdk_x64_windows_hotspot_11.0.9.1_1\)](#)

Check that the following environment variables are defined (below are examples):

- JAVA_HOME



POSTGRESQL

Download server installer: [PostgreSQL \(postgresql-15.1-1-windows-x64\)](#)

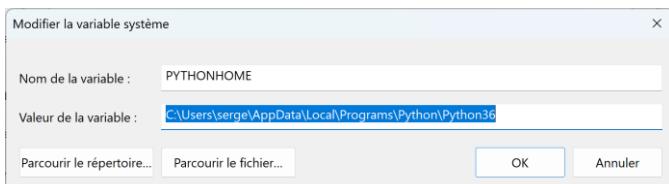
Download management console installer: [pgAdmin \(pgadmin4-8.12-x64\)](#)

PYTHON

Download installer: [PostgreSQL \(postgresql-15.1-1-windows-x64\)](#)

Check that the following environment variables are defined (below are examples):

- PYTHONHOME

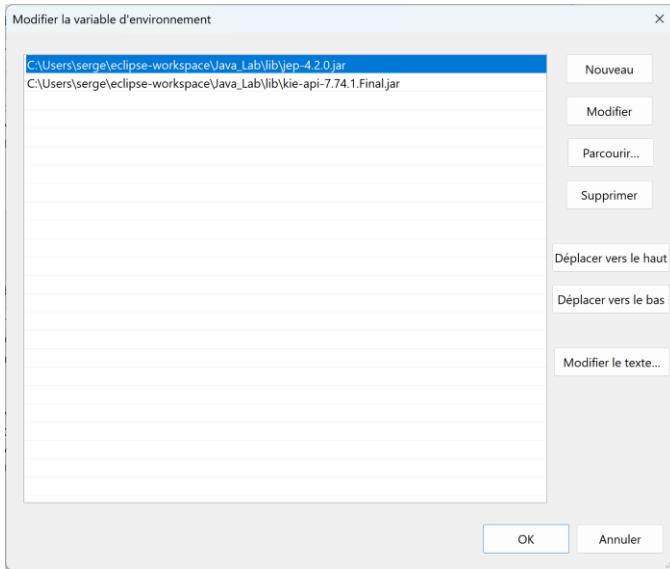


JEP

Installation instructions and (download): [Jep \(Jep 4.2.0\)](#)

Check that the following environment variables are defined (below are just examples):

- CLASSPATH



ECLIPSE

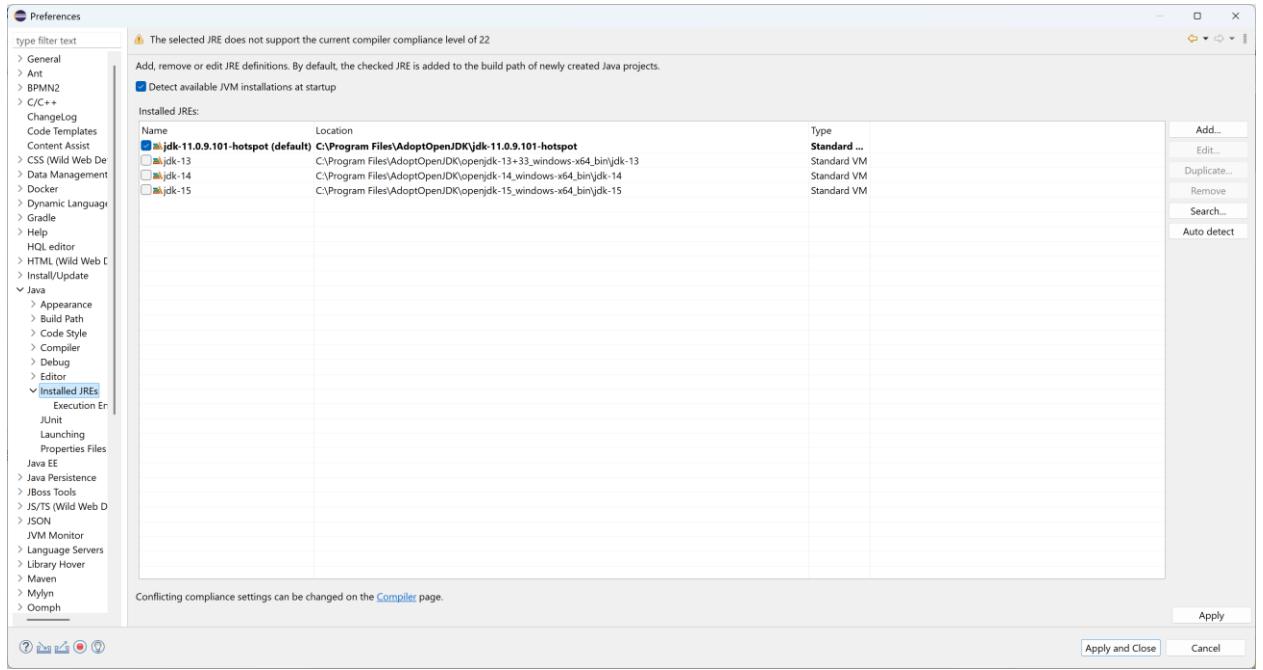
Download Eclipse installer: [Eclipse \(eclipse-inst-jre-win64 2024-09\)](#), choose Eclipse IDE for C/C++

Add the following extensions via Eclipse Marketplace:

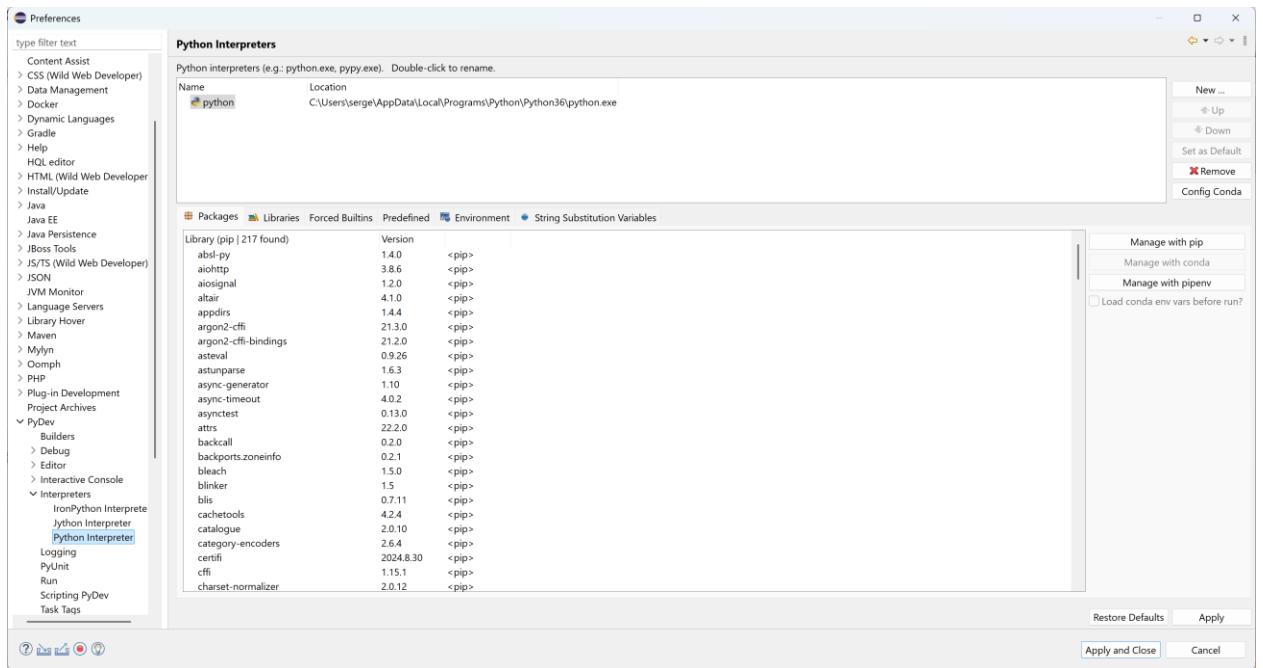
- Eclipse Java Development Tools
- QuickShell
- JBoss Tools
- Eclipse BPMN2 Modeler
- PyDev

Check that the following settings are defined in Eclipse (below are just examples):

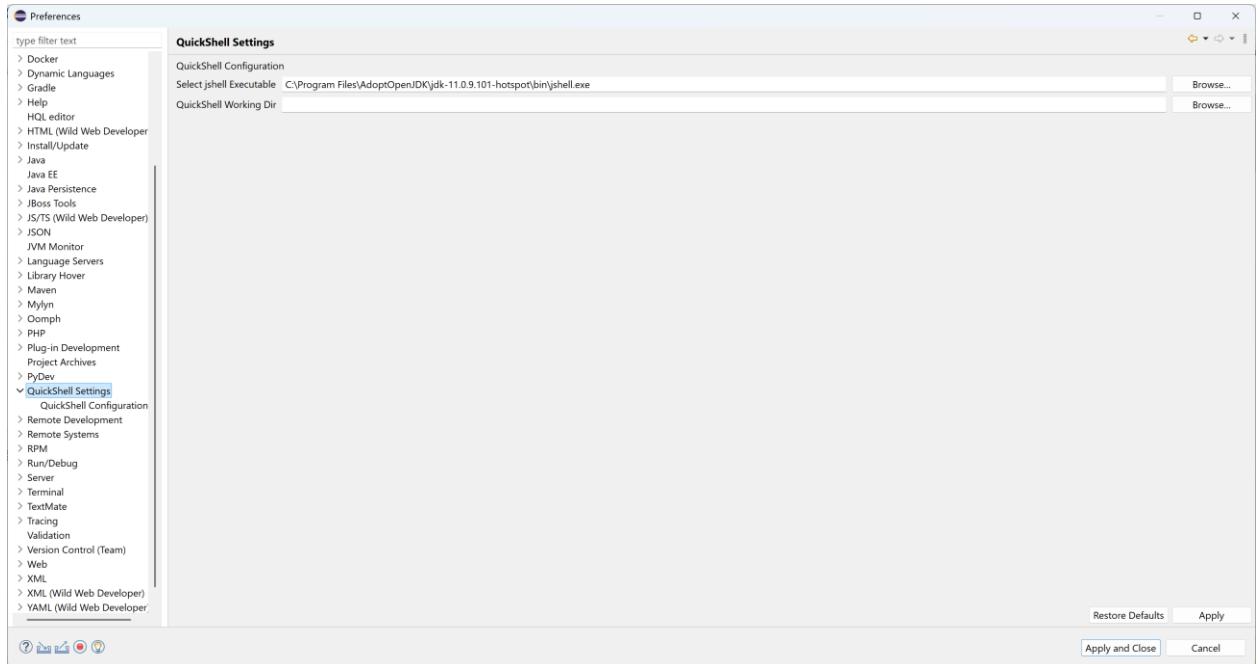
- Installed JREs



- Python Interpreter



- QuickShell Settings



JBPM

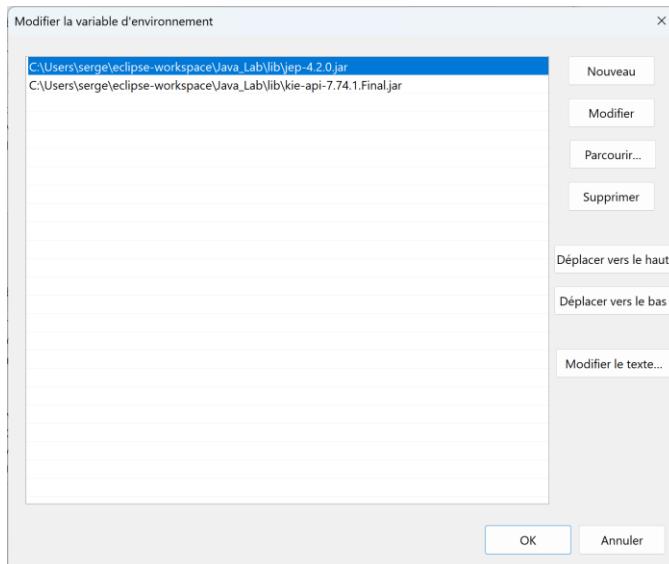
Download and installation instructions: [jbpm-server-7.74.1.Final-dist](#), configure [PostgreSQL persistence](#)

Deploy the following Java archives on the file system:

- [KIE API \(kie-api-7.74.1.Final.jar\)](#)

Check that the following environment variables are defined (below are examples):

- CLASSPATH

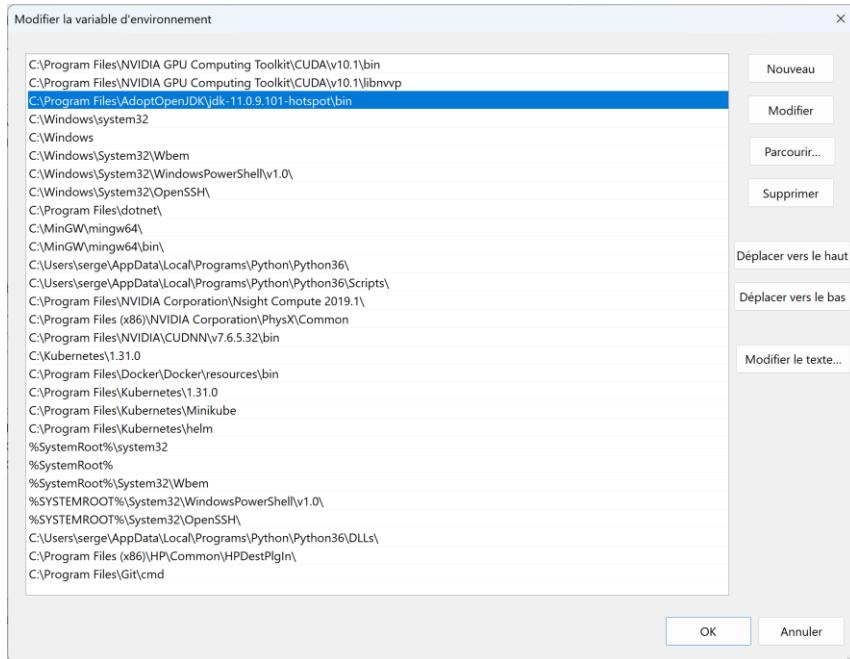


BIRT (OPTIONAL)

Installation instructions and (download): [BIRT \(birt-report-designer-all-in-one-4.17.0-202409160710-win32.win32.x86_64\)](#)

PATH

Check that the system paths to some of the above components are defined (below are examples):



AI ORCHESTRATION PLATFORM DOWNLOAD, IMPORT AND SETUP

DOWNLOAD

IMPORTANT: the below chapters describe a somewhat overkill procedure that is intended to work around some potential clashes between your local package naming and the one implemented in the downloaded classes. It is very likely that in specific cases the same result can be achieved in less steps.

We can now download [AI Orchestration Platform \(AI_Orchestration_Platform_v1.zip\)](#). Extract it from the zip file and move to your Eclipse workspace. The files and folders in the path, e.g.,

C:\Users\serge\eclipse-workspace\jBPM_AIOrchestrationPlatform_discovery\stroke should look like below:



Nom	Modifié le	Type	Taille
.settings	19/03/2025 22:12	Dossier de fichiers	
global	19/03/2025 22:12	Dossier de fichiers	
src	19/03/2025 22:12	Dossier de fichiers	
.classpath	26/12/2024 02:17	Fichier CLASSPATH	1 Ko
.project	26/12/2024 02:17	Fichier PROJECT	1 Ko
package-names-allow-list	26/12/2024 02:17	Fichier	1 Ko
pom.xml	26/12/2024 02:17	Microsoft Edge H...	2 Ko
project.imports	26/12/2024 02:17	Fichier IMPORTS	1 Ko
project.repositories	26/12/2024 02:17	Fichier REPOSITORY...	1 Ko
readme.md	26/12/2024 02:17	Fichier MD	1 Ko

From [Data folder](#) of the same repository, download all the files to a local folder of your choice:

The screenshot shows a GitHub repository page for 'C-NLTX / Open-Source'. The 'Code' tab is selected. On the left, the 'Files' sidebar shows a tree view of files and folders. A folder named 'Data' is expanded, revealing several CSV files: 'column_map_openemr.txt', 'healthcare-dataset-stroke-data...', 'healthcare-dataset-stroke-data....', 'healthcare-dataset-stroke-data_...', and 'Programs'. A dropdown menu above the tree indicates the branch is 'master'. A search bar and a 'Go to file' input field are also present. On the right, a detailed view of a commit titled 'C-NLTX Update and rename healthcare-dataset...' is shown. The commit message includes the URL 'https://github.com/C-NLTX/Open-Source/commit/...'. The commit details show the renamed file 'healthcare-dataset-stroke-data-for-import.csv' and the new file 'healthcare-dataset-stroke-data_cleaned.csv'. The commit was made by 'C-NLTX' on '2023-09-11' at '10:30:00'.

From [Tables folder](#) of the same repository, download all the files to a local folder of your choice:

C-NLTX / Open-Source

<> Code Issues Pull requests Actions Projects Wiki

Files

master

Go to file t

- .settings
- Data
- Programs
- Tables
 - kaggledata.sql
 - kaggletest.sql

Open-Source / Tables / C-NLTX Update and rename kaggle...

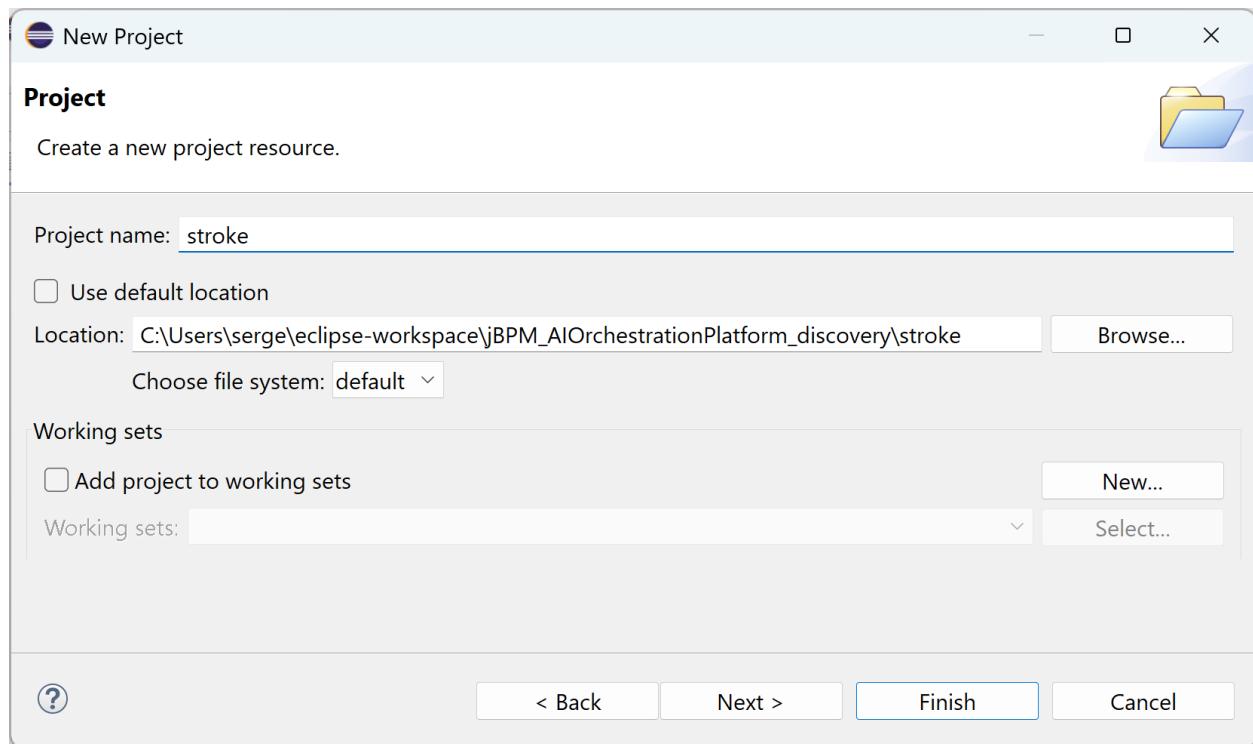
Name ..

kaggledata.sql

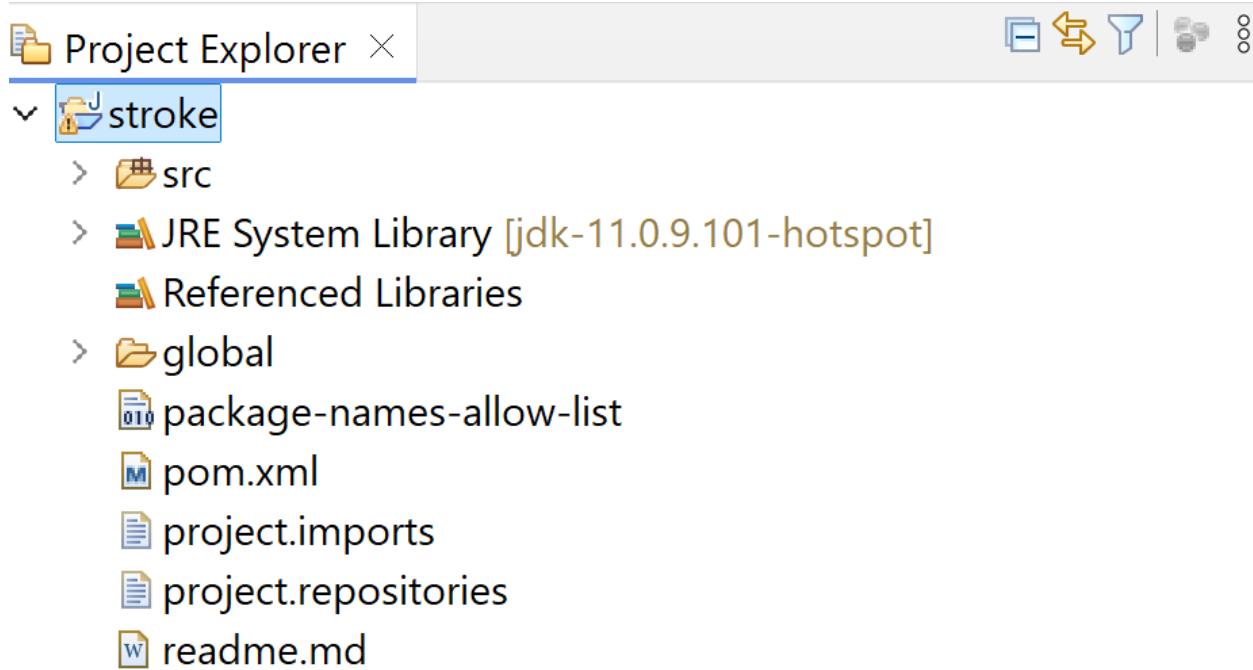
kaggletest.sql

IMPORT

In Eclipse, create a project based on the contents of AI Orchestration Platform's folder:



In Eclipse's Project Explorer, you should be able to find the newly created project:



In pgAdmin, create `stroke` database. In `stroke` database, create `kaggledata` and `kaggletest` tables using the table definitions previously downloaded from [Tables folder](#) of C-NLTX repository. Import in `kaggledata` table the data from `healthcare-dataset-stroke-data_cleaned.csv` file previously downloaded from [Data folder](#) of C-NLTX repository. The data in `kaggledata` table will look like below:

public.kaggledata/stroke/postgres@PostgreSQL 15

No limit Data Output Messages Notifications

	id bigint	gender character varying	age double precision	hypertension integer	heart_disease integer	ever_married character varying	work_type character varying	residence_type character varying	avg_glucose_level double precision	bmi dou
1	9046	Male	67	0	1	Yes	Private	Urban	228.69	
2	51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	
3	31112	Male	80	0	1	Yes	Private	Rural	105.92	
4	60182	Female	49	0	0	Yes	Private	Urban	171.23	
5	1665	Female	79	1	0	Yes	Self-employed	Rural	174.12	
6	56669	Male	81	0	0	Yes	Private	Urban	186.21	
7	53882	Male	74	1	1	Yes	Private	Rural	70.09	
8	10434	Female	69	0	0	No	Private	Urban	94.39	
9	27419	Female	59	0	0	Yes	Private	Rural	76.15	
10	60491	Female	78	0	0	Yes	Private	Urban	58.57	
11	12109	Female	81	1	0	Yes	Private	Rural	80.43	
12	12095	Female	61	0	1	Yes	Govt_job	Rural	120.46	
13	12175	Female	54	0	0	Yes	Private	Urban	104.51	
14	8213	Male	78	0	1	Yes	Private	Urban	219.84	
15	5317	Female	79	0	1	Yes	Private	Urban	214.09	
16	58202	Female	50	1	0	Yes	Self-employed	Rural	167.41	
17	56112	Male	64	0	1	Yes	Private	Urban	191.61	
18	34120	Male	75	1	0	Yes	Private	Urban	221.29	
19	27458	Female	60	0	0	No	Private	Urban	89.22	
20	25226	Male	57	0	1	No	Govt_job	Urban	217.08	

Total rows: 1000 of 5110 Query complete 00:00:00.260

Ln 1, Col 1

The data in kaggletest table (you need to run at least one accuracy test for that data to appear) will look like below:

public.kaggletest/stroke/postgres@PostgreSQL 15

No limit Data Output Messages Notifications

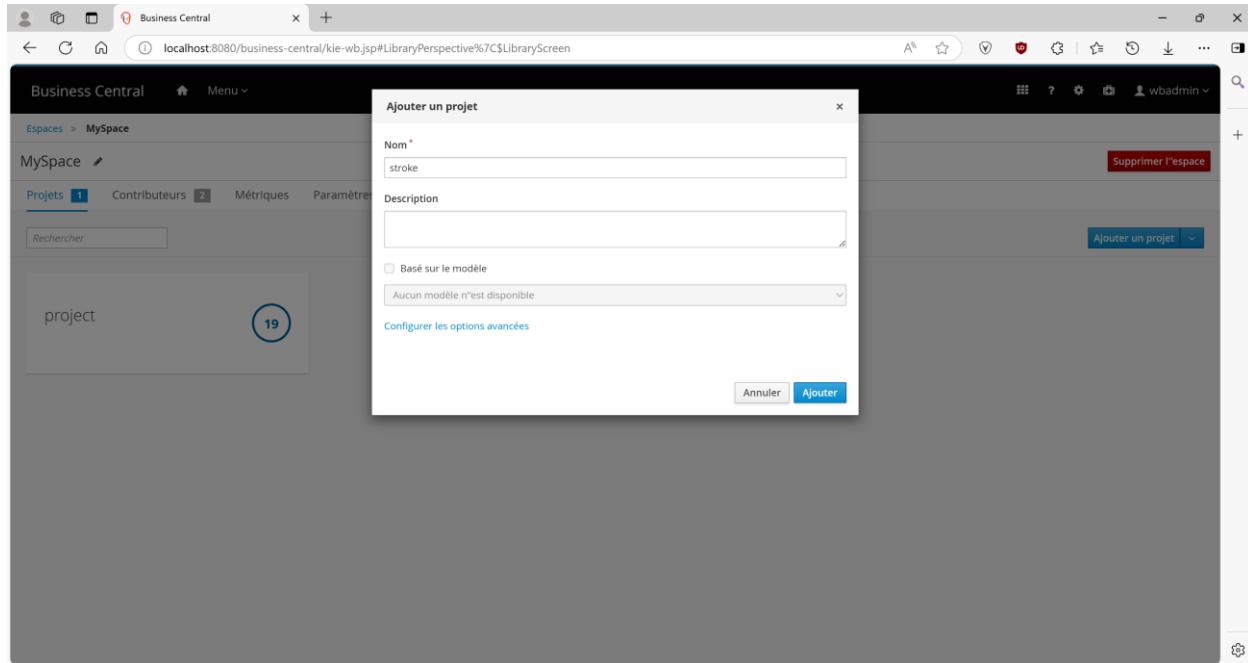
	rt_disease ger	ever_married character varying	work_type character varying	residence_type character varying	avg_glucose_level double precision	bmi double precision	smoking_status character varying	stroke integer	prediction double precision	
1		1	Yes	Private	Urban	228.69	36.6	formerly smoked	1	0.8525082779709623
2		0	Yes	Private	Urban	186.21	29	formerly smoked	1	0.9056102175042715
3		1	No	Govt_job	Urban	217.08	-1	Unknown	1	0.7029105592431985
4		0	Yes	Private	Rural	189.84	-1	Unknown	1	0.8873153280816224
5		0	Yes	Private	Rural	83.41	25.4	Unknown	1	0.0779588688655083
6		0	Yes	Self-employed	Urban	196.92	22.2	never smoked	1	0.7337410587394577
7		1	Yes	Self-employed	Urban	252.72	30.5	formerly smoked	1	0.565860299662291
8		0	No	Govt_job	Urban	84.2	29.7	never smoked	1	0.2625264949536349
9		0	Yes	Private	Urban	60.91	29.9	never smoked	1	0.47950610690377254
10		0	Yes	Private	Urban	71.22	28.5	never smoked	1	0.3661631404759884
11		1	Yes	Private	Rural	127.29	27.7	never smoked	1	0.908463064222811
12		0	Yes	Private	Rural	179.12	28.1	formerly smoked	1	0.8707309913119329
13		0	Yes	Govt_job	Urban	78.92	27.7	formerly smoked	1	0.6735243009713713
14		1	Yes	Private	Urban	68.53	24.2	smokes	1	0.819618628198356
15		0	Yes	Private	Urban	73.54	24	Unknown	1	0.7364536344958862
16		0	Yes	Private	Rural	116.44	23.8	smokes	1	0.5579806148727525
17		0	Yes	Self-employed	Rural	74.02	25	never smoked	1	0.880155833337404
18		0	Yes	Private	Rural	75.02	-1	never smoked	1	0.8792562650000967
19		0	Yes	Self-employed	Rural	91.54	31.4	never smoked	1	0.7880689345374229
20		0	Yes	Self-employed	Rural	91.02	32.9	formerly smoked	1	0.8032666583642609

Total rows: 50 of 50 Query complete 00:00:00.156

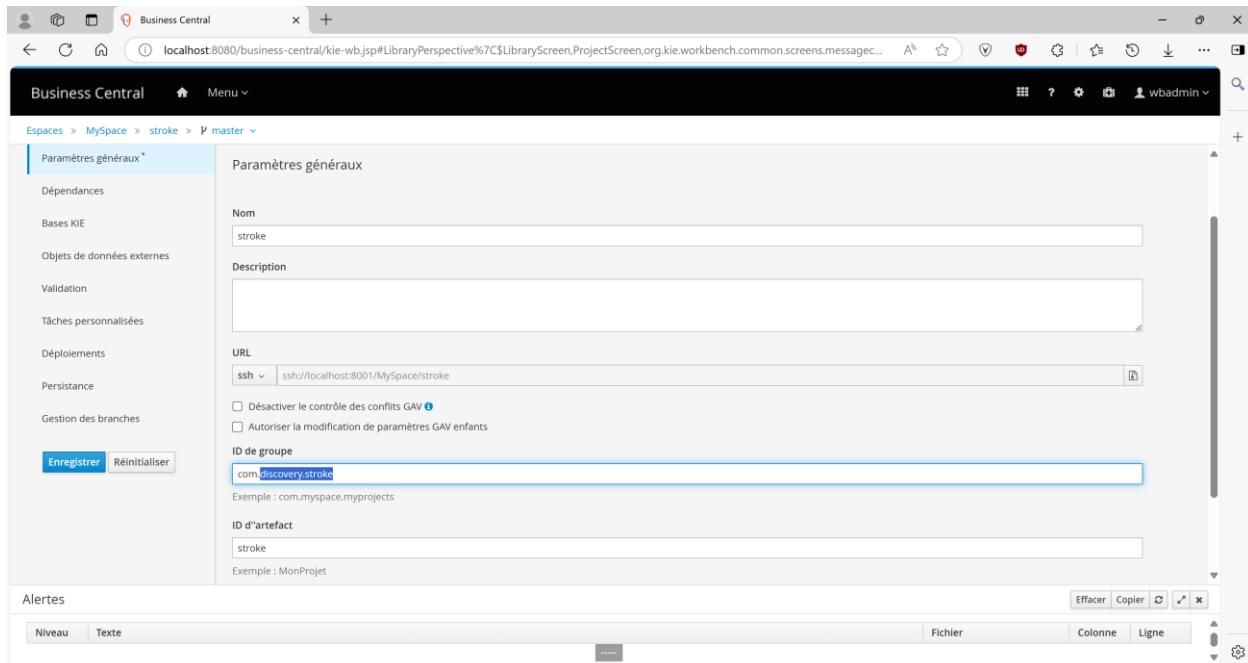
Ln 1, Col 1

SETUP

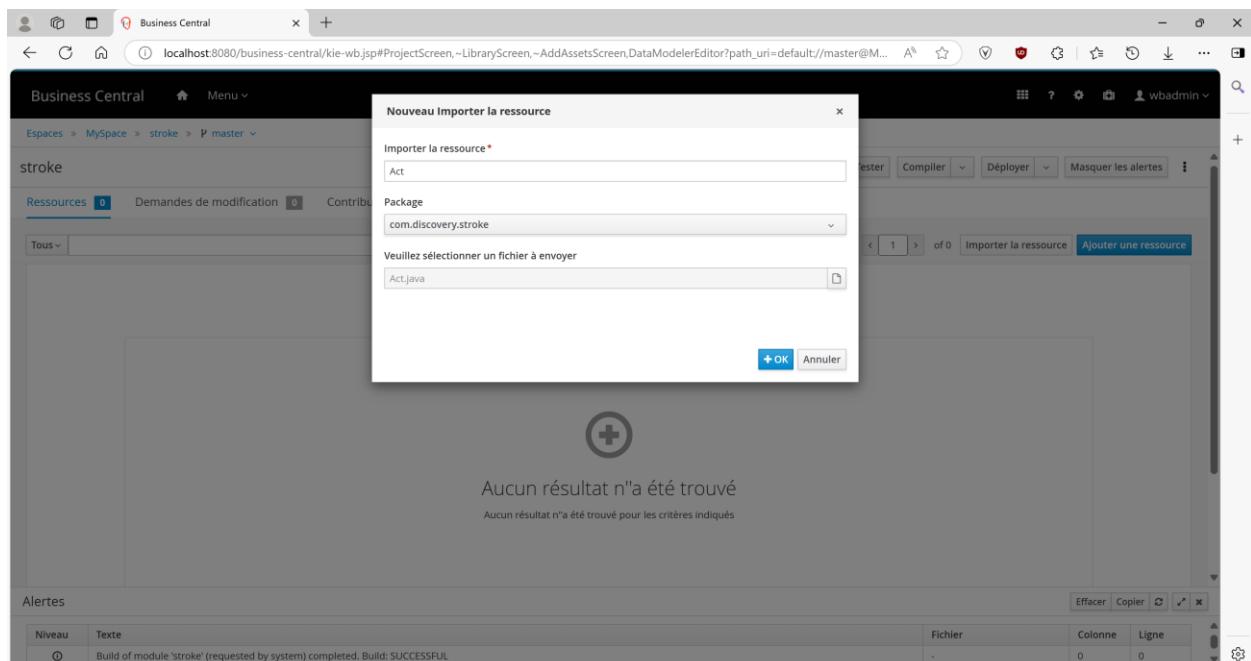
In jBPM's Business Central, create stroke project:



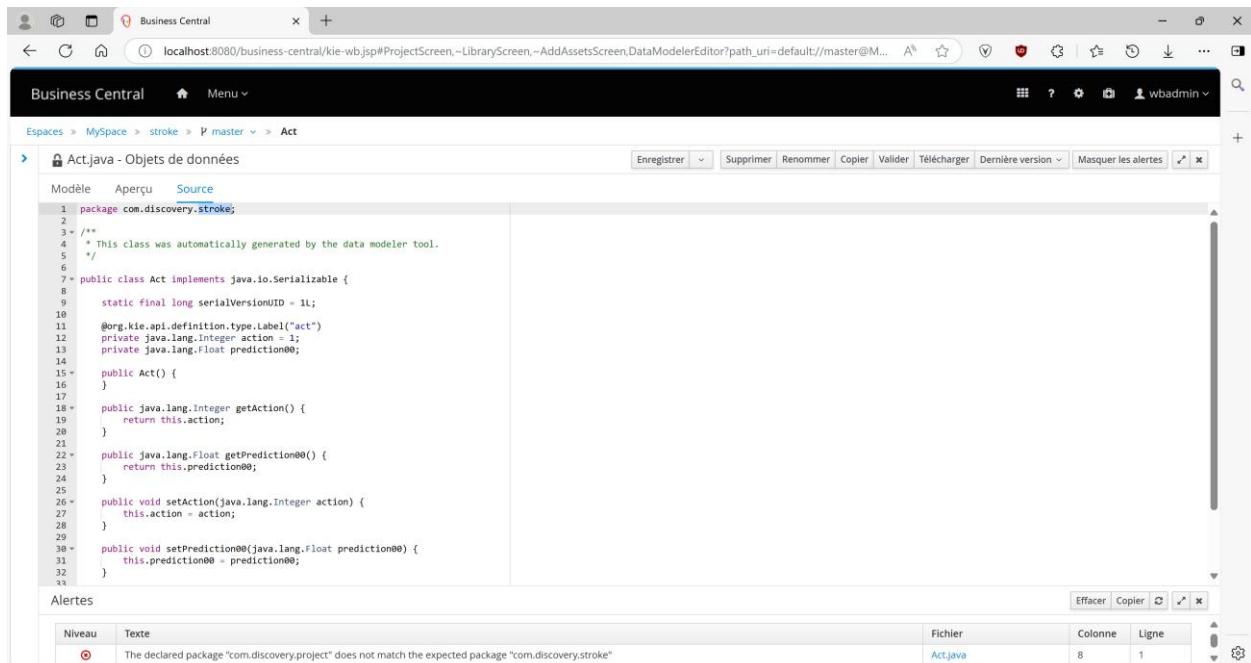
In jBPM's Business Central, in stroke project's parameters, adjust package references:



In jBPM's Business Central, import into server stroke project all the resources from the local stroke project:



On each local data object resource's import, adjust package references:

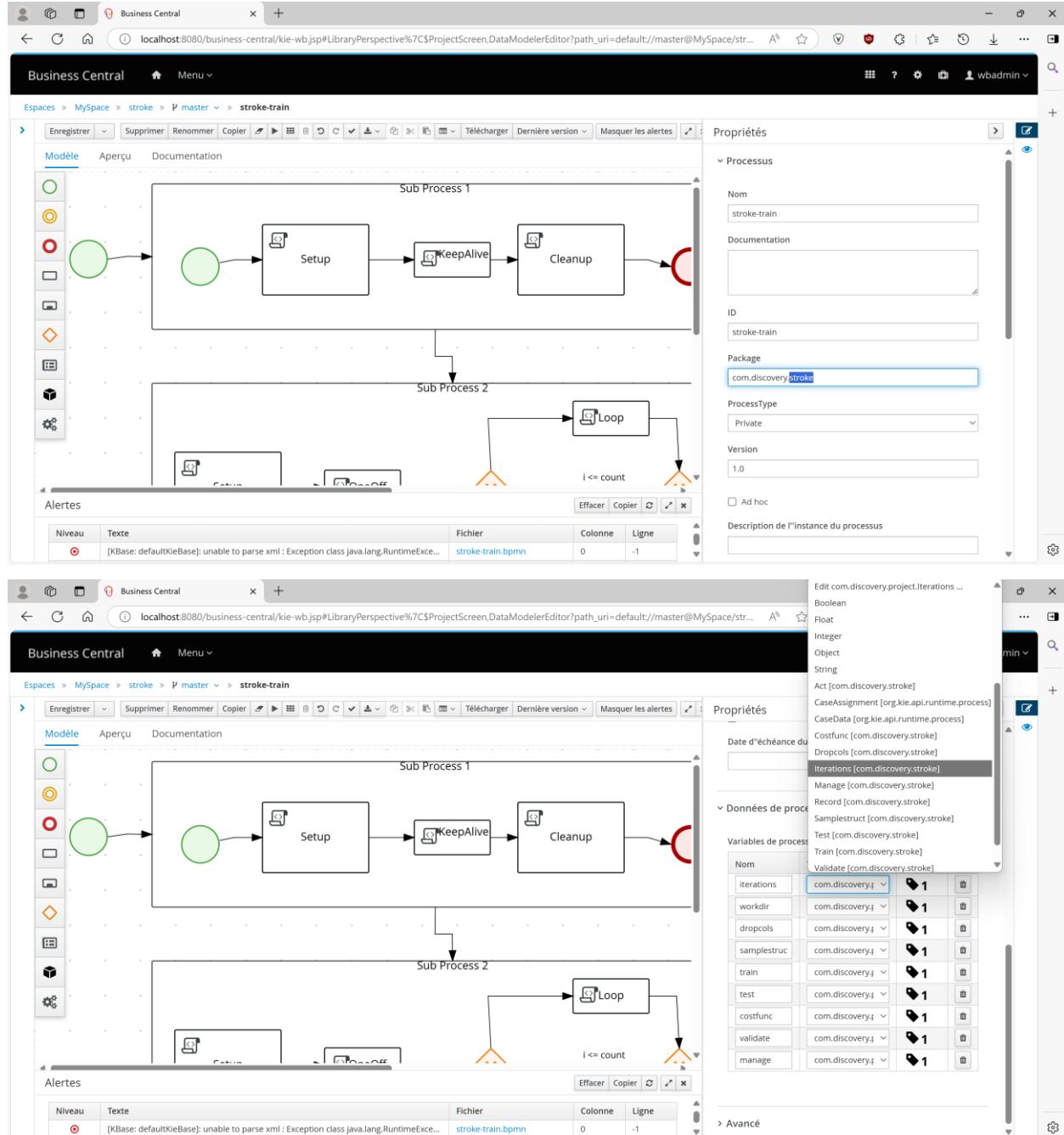


An imported resource will look like below:

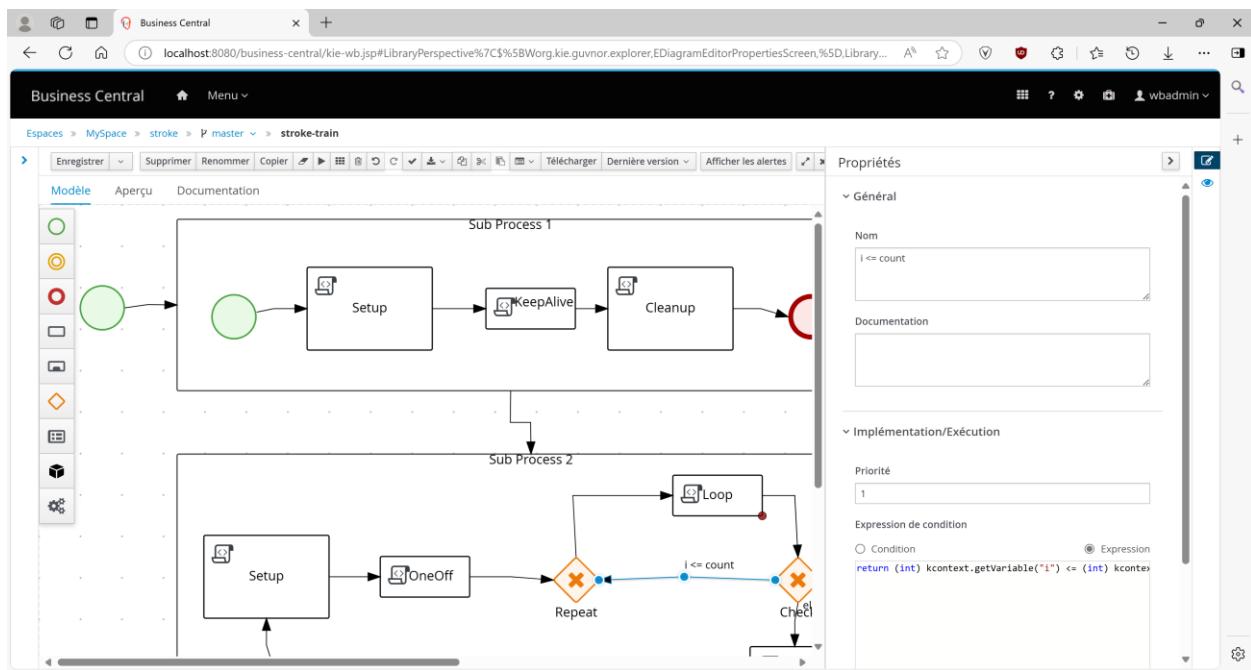
Proceed in the same manner with adjusting importing and adjusting the following data object Java classes:

- Act.java (already done)
- Costfunc.java
- Dropcols.java
- Iterations.java
- Manage.java
- Record.java
- Samplestruct.java
- Test.java
- Train.java
- Validate.java
- Workdir.java

On each local process resource's import, adjust package references:



On each local process resource's import, add casting to Integer in $i \leqslant \text{count}$ arrow:



Proceed in the same manner with adjusting importing and adjusting the following process BPMN classes:

- `stroke-train.bpmn` (already done)
- `stroke-act.bpmn`

Add the dependency for Jep in `stroke` project parameters:

ID de groupe	ID d'artefact	Version	Liste blanche des packages
black.ninja	jep	4.2.0	<input checked="" type="radio"/> All <input type="radio"/> None

Niveau Texte
[KBase: defaultKieBase]: Process Compilation error jep.JepConfig cannot be resolved to a type jep.JepConfig cannot be resolved to a type jep.SubInterpreter cannot be resolved to ...

Compile `stroke` project, you will see the “green pop-up” if it compiled OK:

The screenshot shows the Business Central interface for the 'stroke' project. At the top, there's a navigation bar with 'Business Central' and a user icon. Below it, a breadcrumb trail shows 'Espaces > MySpace > stroke > master'. The main area has tabs for 'Ressources' (14), 'Demandes de modification' (0), 'Contributeurs' (2), 'Métriques', and 'Paramètres'. A search bar and a message 'Compilation réussie' are at the top. Below is a table of resources:

		Objets de données	Dernière modification aujourd'hui	Créé aujourd'hui
	Act	Objets de données	Dernière modification aujourd'hui	Créé aujourd'hui
	Costfunc	Objets de données	Dernière modification aujourd'hui	Créé aujourd'hui
	Dropcols	Objets de données	Dernière modification aujourd'hui	Créé aujourd'hui
	Iterations	Objets de données	Dernière modification aujourd'hui	Créé aujourd'hui

At the bottom, there's an 'Alertes' section with a table:

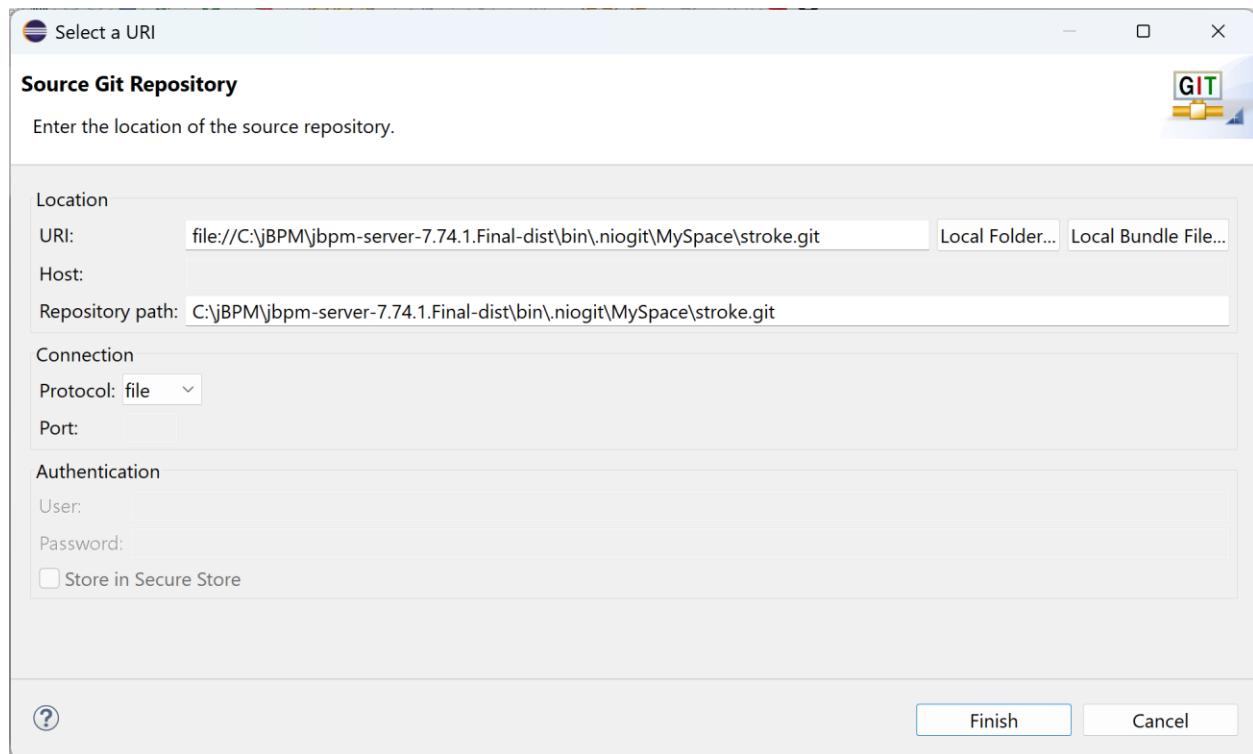
Niveau	Texte	Fichier	Colonne	Ligne
Info	Build of module 'stroke' (requested by wbadmin) completed. Build: SUCCESSFUL	-	0	0

In Eclipse's Git Repositories tab, add a new local Git repository linked to stroke project:

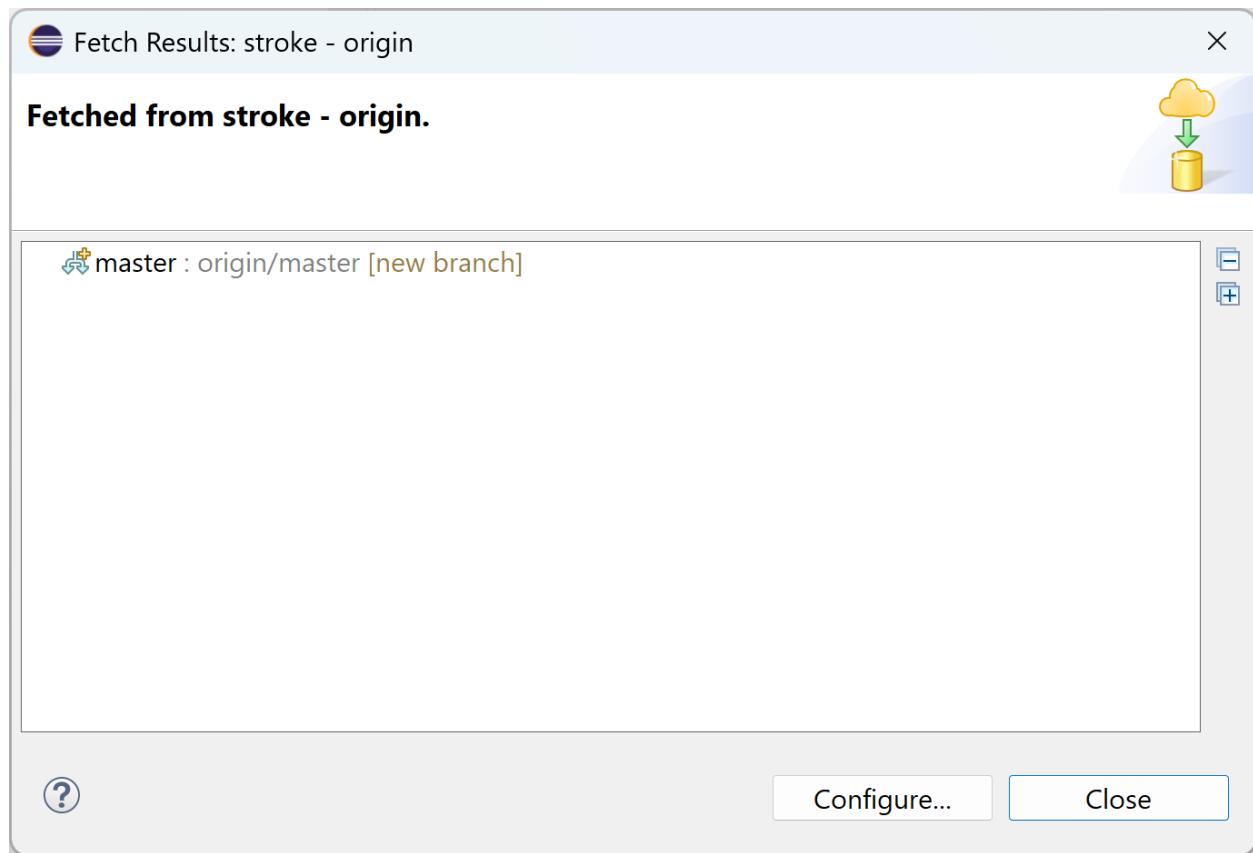
The screenshot shows the Eclipse Git Repositories tab. The 'Git Repositories' tab is selected. The list contains:

- > stroke [master] - C:\Users\serge\eclipse-workspace\jBPM_AIOrchestrationPlatform_discovery\stroke\.git
 - > Branches
 - > Tags
 - > References
 - > Remotes
 - > Working Tree - C:\Users\serge\eclipse-workspace\jBPM_AIOrchestrationPlatform_discovery\stroke
- > SuiteCRM_suitecrm_project [master] - C:\Users\serge\eclipse-workspace\SuiteCRM_suitecrm_project\.git

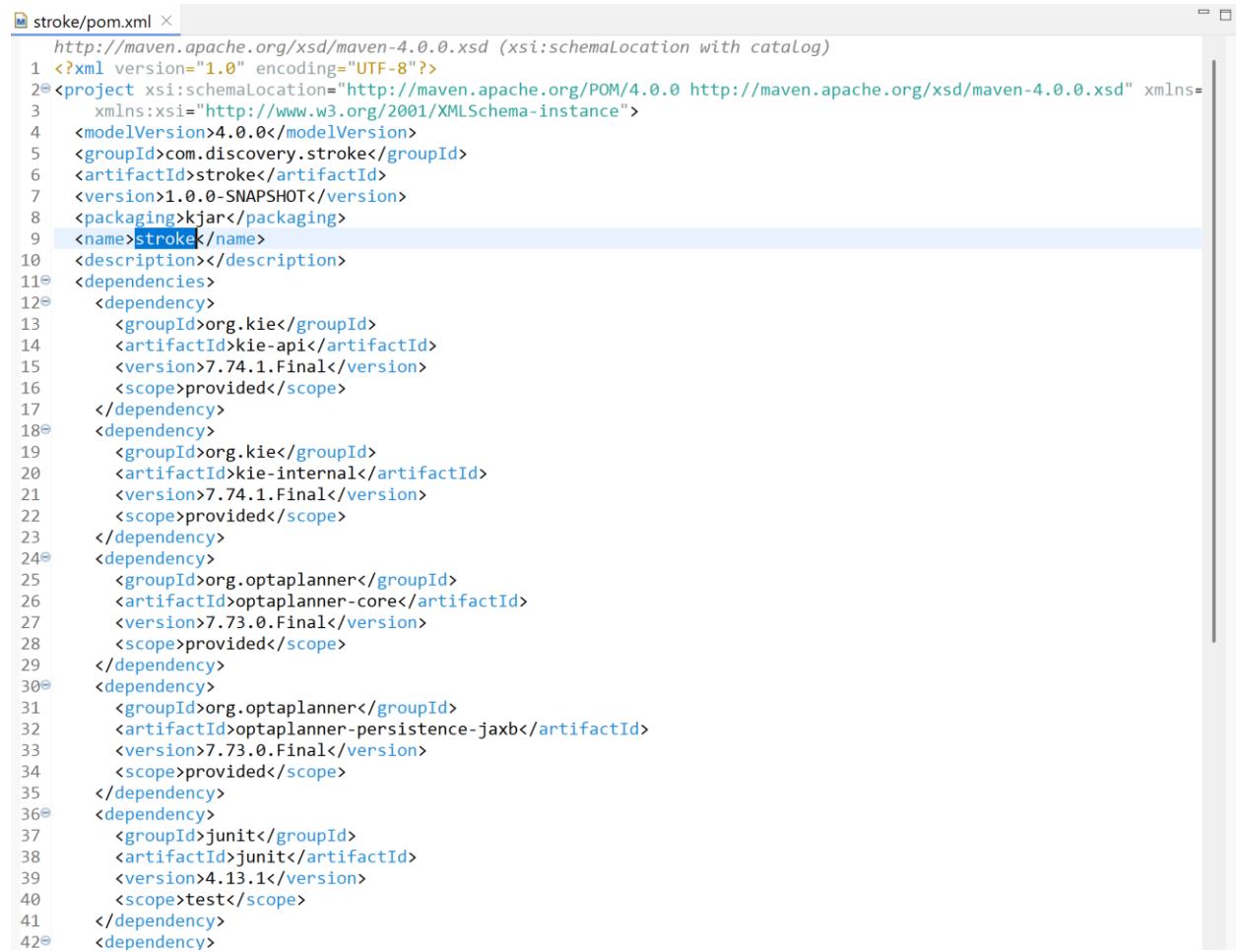
Configure and then Save and Fetch a remote fetch into the local stroke project's Git repository added by you just before:



You will see the following message if your fetch was OK:



Now we will implement similar edits locally in stroke project in Eclipse. In stroke project's parameters, adjust package references:



The screenshot shows the Eclipse IDE interface with the 'stroke/pom.xml' file open in the central editor area. The XML code is displayed, showing Maven project configuration. The 'name' element is highlighted with a blue selection bar. The code lists various dependencies, primarily from org.kie and org.optaplanner groups, with their artifact IDs, versions, and scopes.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>com.discovery.stroke</groupId>
5   <artifactId>stroke</artifactId>
6   <version>1.0.0-SNAPSHOT</version>
7   <packaging>kjar</packaging>
8   <name>stroke</name>
9   <description></description>
10  <dependencies>
11    <dependency>
12      <groupId>org.kie</groupId>
13      <artifactId>kie-api</artifactId>
14      <version>7.74.1.Final</version>
15      <scope>provided</scope>
16    </dependency>
17    <dependency>
18      <groupId>org.kie</groupId>
19      <artifactId>kie-internal</artifactId>
20      <version>7.74.1.Final</version>
21      <scope>provided</scope>
22    </dependency>
23    <dependency>
24      <groupId>org.optaplanner</groupId>
25      <artifactId>optaplanner-core</artifactId>
26      <version>7.73.0.Final</version>
27      <scope>provided</scope>
28    </dependency>
29    <dependency>
30      <groupId>org.optaplanner</groupId>
31      <artifactId>optaplanner-persistence-jaxb</artifactId>
32      <version>7.73.0.Final</version>
33      <scope>provided</scope>
34    </dependency>
35    <dependency>
36      <groupId>junit</groupId>
37      <artifactId>junit</artifactId>
38      <version>4.13.1</version>
39      <scope>test</scope>
40    </dependency>
41    <dependency>
```

Adjust package references in data objects:

```

1 package com.discovery.stroke;
2
3 /**
4  * This class was automatically generated by the data modeler tool.
5 */
6
7 public class Act implements java.io.Serializable {
8
9     static final long serialVersionUID = 1L;
10
11     @org.kie.api.definition.type.Label("act")
12     private java.lang.Integer action = 1;
13     private java.lang.Float prediction00;
14
15     public Act() {
16     }
17
18     public java.lang.Integer getAction() {
19         return this.action;
20     }
21
22     public java.lang.Float getPrediction00() {
23         return this.prediction00;
24     }
25
26     public void setAction(java.lang.Integer action) {
27         this.action = action;
28     }
29
30     public void setPrediction00(java.lang.Float prediction00) {
31         this.prediction00 = prediction00;
32     }
33
34     public Act(java.lang.Integer action, java.lang.Float prediction00) {
35         this.action = action;
36         this.prediction00 = prediction00;
37     }
38
39 }

```

Adjust package reference in persistence.xml:

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:orm="http://java.sun.com/xml/ns/persistence/orm"
3     <persistence-unit name="com.discovery.stroke:1.0.0-SNAPSHOT" transaction-type="JTA">
4         <provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
5         <jta-data-source>java:jboss/datasources/jBPMDS</jta-data-source>
6         <class>org.drools.persistence.info.SessionInfo</class>
7         <class>org.jbpm.persistence.processinstance.ProcessInstanceInfo</class>
8         <class>org.drools.persistence.info.WorkItemInfo</class>
9         <class>org.jbpm.persistence.correlation.CorrelationKeyInfo</class>
10        <class>org.jbpm.persistence.correlation.CorrelationPropertyInfo</class>
11        <class>org.jbpm.runtime.manager.impl.jpa.ContextMappingInfo</class>
12        <class>org.jbpm.process.audit.ProcessInstanceLog</class>
13        <class>org.jbpm.process.audit.NodeInstanceLog</class>
14        <class>org.jbpm.process.audit.VariableInstanceLog</class>
15        <exclude-unlisted-classes>true</exclude-unlisted-classes>
16        <properties>
17            <property name="hibernate.dialect" value="org.hibernate.dialect.PostgreSQLDialect"/>
18            <property name="hibernate.max_fetch_depth" value="3"/>
19            <property name="hibernate.hbm2ddl.auto" value="update"/>
20            <property name="hibernate.show_sql" value="false"/>
21            <property name="hibernate.id.new_generator_mappings" value="false"/>
22            <property name="hibernate.transaction.jta.platform" value="org.hibernate.service.jta.platform.internal
23                </properties>
24            </persistence-unit>
25        </persistence>

```

Adjust package references in both processes:

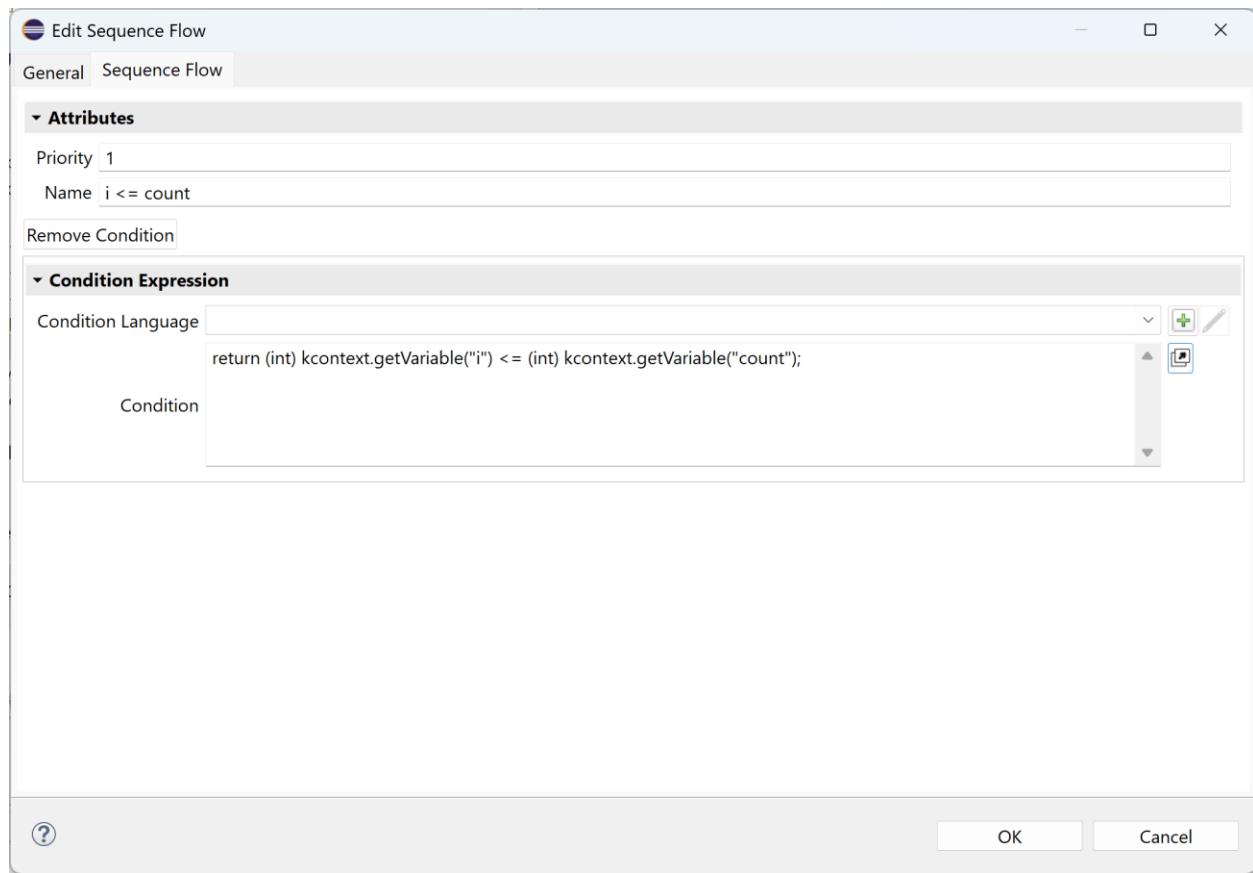
```
stroke-train.bpmn x
  si:schemaLocation) | BPMN20.xsd (xsi:schemaLocation) | drools.xsd (xsi:schemaLocation)
1= "1.0" encoding="UTF-8"?>
2it X=0.0 Y=0.0 -->
3tions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MC
4Definition id="_iterationsItem" isCollection="false" structureRef="com.discovery.stroke.Iterations"/>
5Definition id="_count" isCollection="false" structureRef="Integer"/>
6Definition id="_i" isCollection="false" structureRef="Integer"/>
7Definition id="ItemDefinition_5" isCollection="false" structureRef="java.lang.Integer"/>
8Definition id="ItemDefinition_12" isCollection="false" structureRef="com.discovery.stroke.Workdir"/>
9Definition id="ItemDefinition_1" isCollection="false" structureRef="com.discovery.stroke.Dropcols"/>
10Definition id="ItemDefinition_2" isCollection="false" structureRef="com.discovery.stroke.Samplestruct"/>
11Definition id="ItemDefinition_3" isCollection="false" structureRef="com.discovery.stroke.Train"/>
12Definition id="ItemDefinition_4" isCollection="false" structureRef="com.discovery.stroke.Test"/>
13Definition id="ItemDefinition_6" isCollection="false" structureRef="com.discovery.stroke.Costfunc"/>
14Definition id="ItemDefinition_7" isCollection="false" structureRef="com.discovery.stroke.Validate"/>
15Definition id="ItemDefinition_8" isCollection="false" structureRef="com.discovery.stroke.Manage"/>
16ess id="stroke-train" tns:packageName="com.discovery.stroke" tns:version="1.0" name="stroke-train" isExecutable='
17:tensionElements>
18port name="org.kie.api.runtime.process.CaseAssignment"/>
19port name="org.kie.api.runtime.process.CaseData"/>
20xtensionElements>
21roperty id="iterations" itemSubjectRef="_iterationsItem" name="iterations">
22 extensionElements>
23 metaData name="customTags">
24is:metaValue>iterations</tns:metaValue>
25::metaData>
26!:extensionElements>
27roperty>
28roperty id="workdir" itemSubjectRef="ItemDefinition_12" name="workdir">
29 extensionElements>
30 metaData name="customTags">
31is:metaValue>workdir</tns:metaValue>
32::metaData>
33!:extensionElements>
34roperty>
35roperty id="dropcols" itemSubjectRef="ItemDefinition_1" name="dropcols">
36 extensionElements>
37 metaData name="customTags">
38is:metaValue>dropcols</tns:metaValue>
39::metaData>
40!:extensionElements>
41roperty>
42roperty id="samplestruct" itemSubjectRef="ItemDefinition_2" name="samplestruct">
43 extensionElements>
44 metaData name="customTags">
```

```

stroke.act.bpmn x
  bpsim.xsd (xsi:schemaLocation) | BPMN20.xsd (xsi:schemaLocation) | drools.xsd (xsi:schemaLocation)
1<?xml version="1.0" encoding="UTF-8"?>
2!-- origin at X=0.0 Y=0.0 -->
3<bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:bpmn2="http://www.omg.org/spec/BPMN/2.0" xsi:schemaLocation="http://www.omg.org/spec/BPMN/2.0/bpmn2.xsd http://www.w3.org/2001/XMLSchema-instance">
4  <bpmn2:itemDefinition id="_iterationsItem" isCollection="false" structureRef="com.discovery.stroke.Iterations"/>
5  <bpmn2:itemDefinition id="_count" isCollection="false" structureRef="Integer"/>
6  <bpmn2:itemDefinition id="_i" isCollection="false" structureRef="Integer"/>
7  <bpmn2:itemDefinition id="ItemDefinition_5" isCollection="false" structureRef="java.lang.Integer"/>
8  <bpmn2:itemDefinition id="ItemDefinition_12" isCollection="false" structureRef="com.discovery.stroke.Workdir"/>
9  <bpmn2:itemDefinition id="ItemDefinition_1" isCollection="false" structureRef="com.discovery.stroke.Dropcols"/>
10 <bpmn2:itemDefinition id="ItemDefinition_2" isCollection="false" structureRef="com.discovery.stroke.Samplestruct"/>
11 <bpmn2:itemDefinition id="ItemDefinition_3" isCollection="false" structureRef="com.discovery.stroke.Train"/>
12 <bpmn2:itemDefinition id="ItemDefinition_4" isCollection="false" structureRef="com.discovery.stroke.Test"/>
13 <bpmn2:itemDefinition id="ItemDefinition_8" isCollection="false" structureRef="com.discovery.stroke.Record"/>
14 <bpmn2:itemDefinition id="ItemDefinition_9" isCollection="false" structureRef="com.discovery.stroke.Act"/>
15 <bpmn2:process id="stroke-act" tns:packageName="com.discovery.stroke" tns:version="1.0" name="stroke-act" isExecutable="true">
16   <bpmn2:extensionElements>
17     <tns:import name="org.kie.api.runtime.process.CaseAssignment"/>
18     <tns:import name="org.kie.api.runtime.process.CaseData"/>
19   </bpmn2:extensionElements>
20   <bpmn2:property id="iterations" itemSubjectRef="_iterationsItem" name="iterations">
21     <bpmn2:extensionElements>
22       <tns:metaData name="customTags">
23         <tns:metaValue>iterations</tns:metaValue>
24       </tns:metaData>
25     </bpmn2:extensionElements>
26   </bpmn2:property>
27   <bpmn2:property id="workdir" itemSubjectRef="ItemDefinition_12" name="workdir">
28     <bpmn2:extensionElements>
29       <tns:metaData name="customTags">
30         <tns:metaValue>workdir</tns:metaValue>
31       </tns:metaData>
32     </bpmn2:extensionElements>
33   </bpmn2:property>
34   <bpmn2:property id="dropcols" itemSubjectRef="ItemDefinition_1" name="dropcols">
35     <bpmn2:extensionElements>
36       <tns:metaData name="customTags">
37         <tns:metaValue>dropcols</tns:metaValue>
38       </tns:metaData>
39     </bpmn2:extensionElements>
40   </bpmn2:property>
41   <bpmn2:property id="samplestruct" itemSubjectRef="ItemDefinition_2" name="samplestruct">
42     <bpmn2:extensionElements>
43       <tns:metaData name="customTags">
44         <tns:metaValue>samplestruct</tns:metaValue>

```

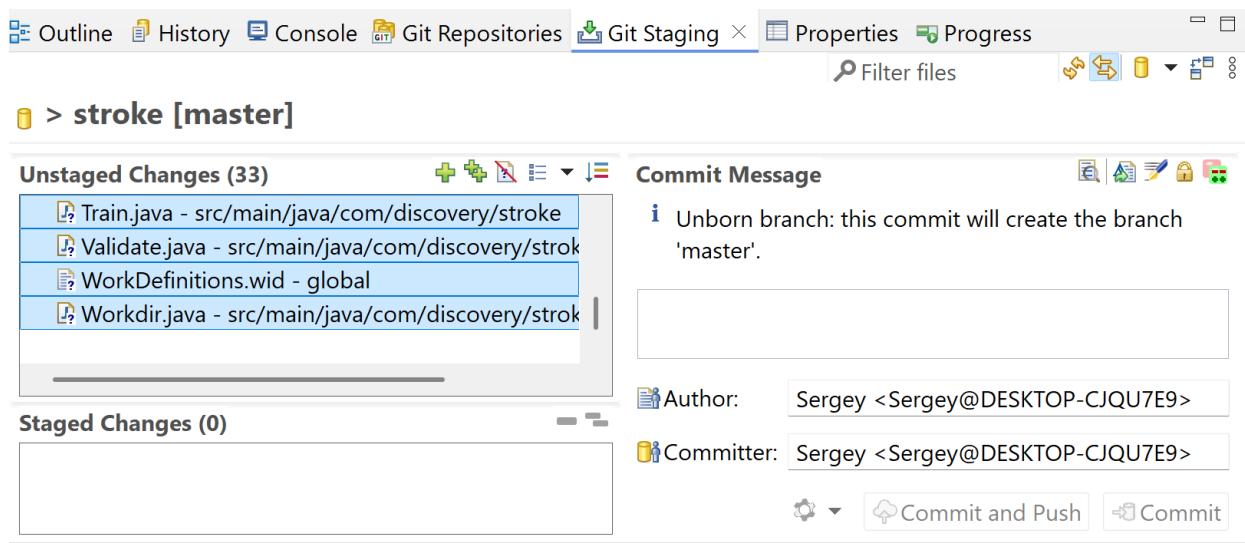
Opening both processes with BPMN2 diagram editor, add casting to Integer in $i \leq count$ arrow:



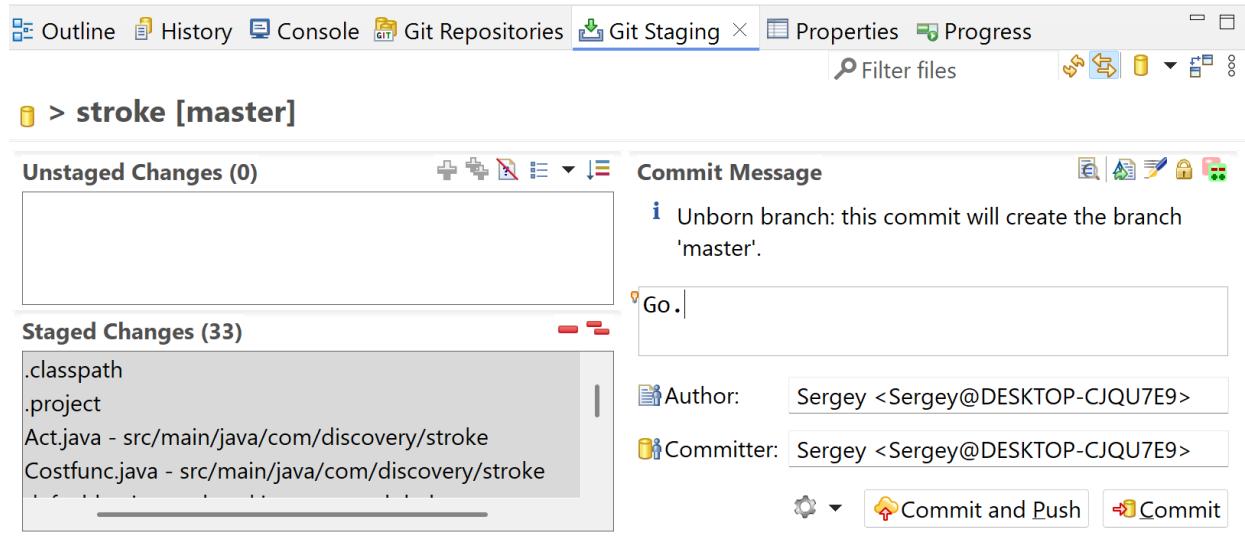
Using Windows Explorer, rename the local Eclipse project subfolders to stroke:



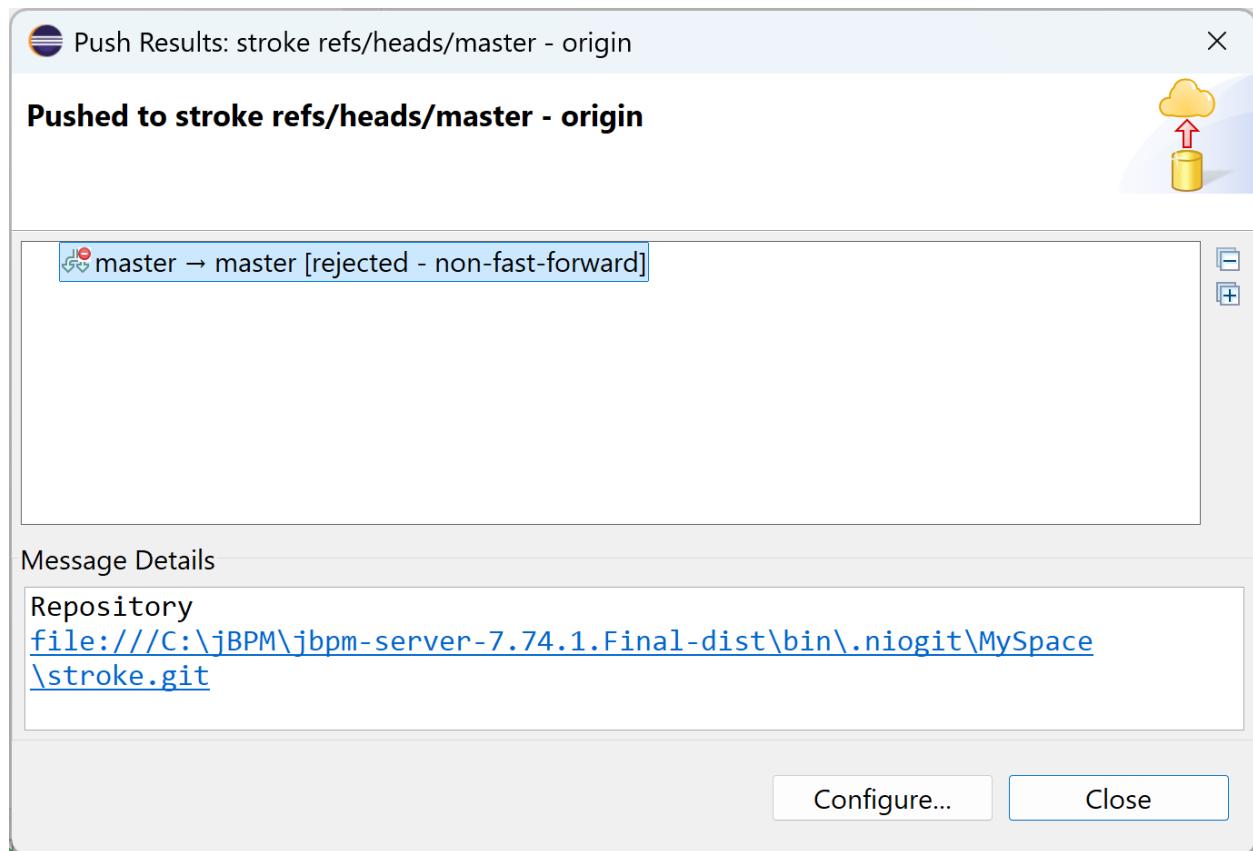
In Eclipse's Git Staging tab, you should be able to see the following unstaged changes:



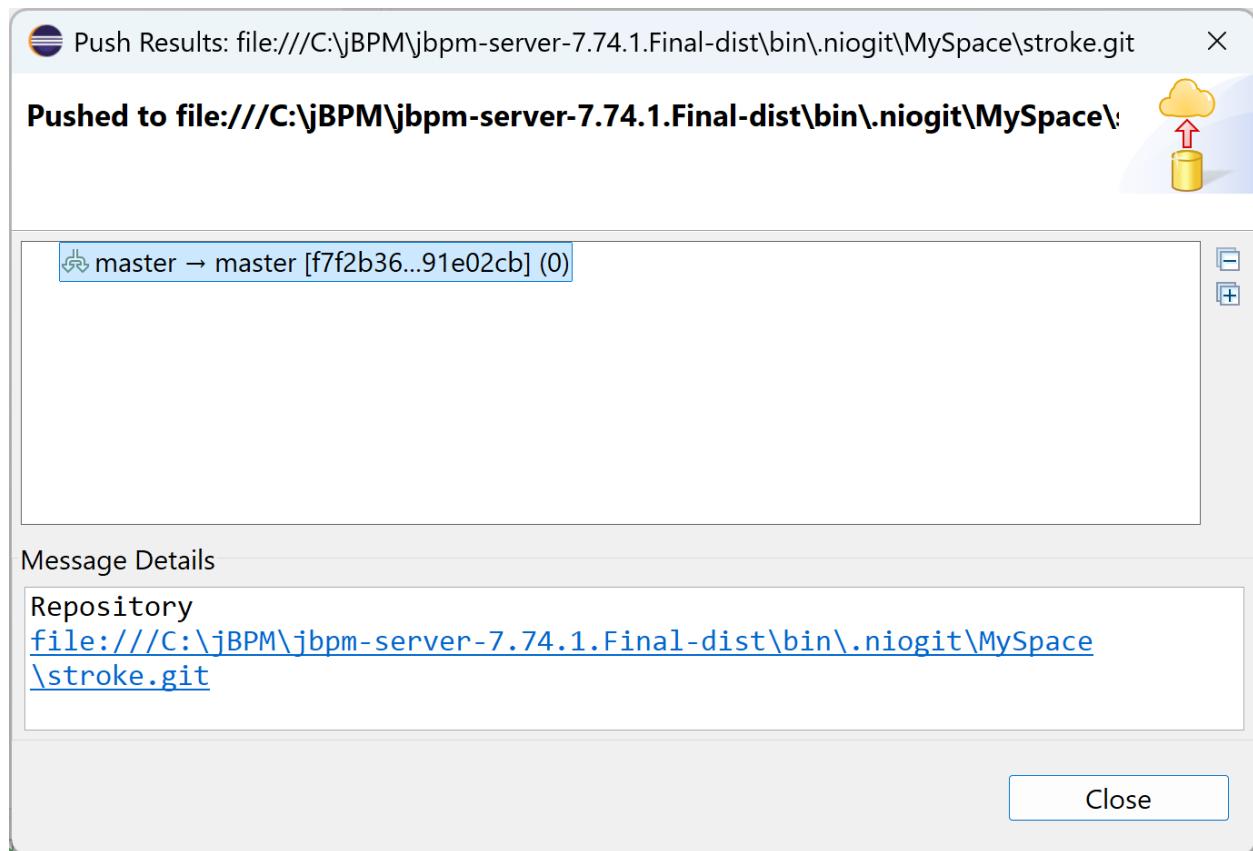
Make them all staged changes. Enter the commit message and click on Commit and Push:



You may receive the following error message:



In that case, close this message, and click on Push HEAD in Git Staging tab. In the dialog window that will appear, check the Force overwrite branch in remote if exists and has diverged box, and click Preview, then Push. You will see the following message if your push was OK:



Compile `stroke` project on the server.

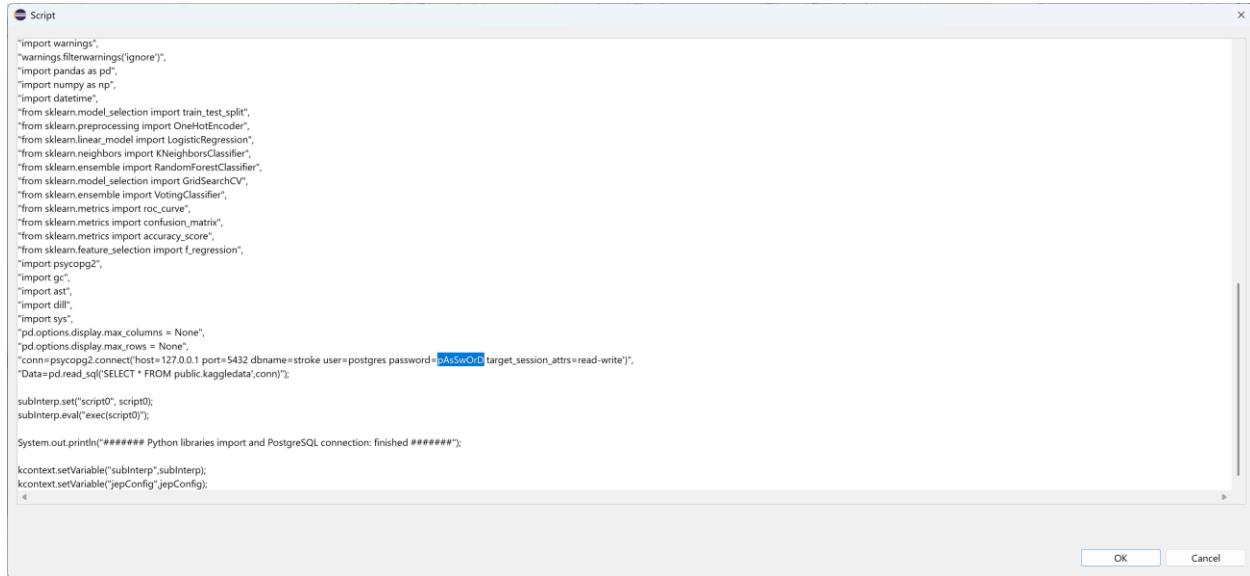
We complete the setup by installing the following Python module versions (Python version 3.6.7 is assumed as specified above in Components installation section):

- pip install pandas==1.1.5
- pip install numpy==1.19.5
- pip install scikit-learn==0.20.3
- pip install scipy==1.5.4
- pip install psycopg2==2.9.6 (dt dec pq3 ext lo64)
- pip install matplotlib==3.3.4
- pip install seaborn==0.11.2

AI ORCHESTRATION PLATFORM CUSTOMIZING AND TESTING

CUSTOMIZING

Before we can test our AI Orchestration Platform, let us customize the content we have just deployed. In stroke-train process (Sub Process 2, Setup), adjust the password to establish a connection to your PostgreSQL server:



The screenshot shows a Java IDE window with a script editor. The code is a Python script that imports various libraries from sklearn, numpy, and pandas. It then connects to a PostgreSQL database using psycopg2, specifying host=127.0.0.1, port=5432, dbname=stroke, user=postgres, and password=bAsSwOrD. The connection is established with target_session_attrs=read-write. After the connection is made, it performs a SELECT * query on the public.kaggledata table. The code concludes with setting subInterp.set("script0", script0) and subInterp.eval("exec(script0);"). A System.out.println statement indicates the completion of the import and connection process. Finally, it sets kcontext.setVariable("subInterp", subInterp) and kcontext.setVariable("jepConfig", jepConfig). The window has standard OS X-style buttons for OK and Cancel.

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import datetime
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import roc_curve
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.feature_selection import f_regression
import psycopg2
import gc
import ast
import dill
import sys
pd.options.display.max_columns = None
pd.options.display.max_rows = None
conn=psycopg2.connect(host='127.0.0.1' port=5432 dbname='stroke' user='postgres' password='bAsSwOrD' target_session_attrs='read-write')
data=pd.read_sql('SELECT * FROM public.kaggledata',conn)
System.out.println("##### Python libraries import and PostgreSQL connection: finished #####")
kcontext.setVariable("subInterp", subInterp);
kcontext.setVariable("jepConfig", jepConfig);

```

Open `Workdir.java` data object's class, find the default path and create this path in your local file system:

```
1 package com.discovery.stroke;
2
3 *
4 // This class was automatically generated by the data modeler tool.
5 /
6
7 public class Workdir implements java.io.Serializable {
8
9     static final long serialVersionUID = 1L;
10
11     @org.kie.api.definition.type.Label("workdir")
12     private java.lang.String path = "C:/xampp/htdocs/OpenEMR-7.0.2/interface/modules/custom_modules/oe-module-jbpm-generic-callout/public";
13
14     public Workdir() {
15     }
16
17     public java.lang.String getPath() {
18         return this.path;
19     }
20
21     public void setPath(java.lang.String path) {
22         this.path = path;
23     }
24
25     public Workdir(java.lang.String path) {
26         this.path = path;
27     }
28
29 }
```

Stage, commit and push to the server any unstaged changes. Compile `stroke` project on the server.

TESTING

We can now test our AI Orchestration Platform by sending train/validate/test and predict requests to Stroke Prediction solution that we implemented on the server. Before anything, deploy Stroke Prediction on the server by clicking Deploy when in stroke project in Business Central:

The screenshot shows the Business Central interface with the 'stroke' project selected. A deployment message at the top states: 'Le déploiement sur la configuration du serveur a réussi et le conteneur a été mis à jour correctement.' Below this, there is a table listing four resources: 'Act', 'Costfunc', 'Dropcols', and 'Iterations'. Each resource is listed as an 'Objets de données' (Data Objects) with a 'Dernière modification il y a 1 jour' (Last modified 1 day ago) and 'Créé il y a 1 jour' (Created 1 day ago). At the bottom, there is a section titled 'Alertes' (Alerts) showing a single entry: 'Build of module 'stroke' (requested by wbadmin) completed. Build: SUCCESSFUL'.

Start KIE Server Documentation portal, go to Process instances section and feed the following elements into /server/containers/{containerId}/processes/{processId}/instances query:

containerid: stroke_1.0.0-SNAPSHOT

processid: stroke-train

body:

```
{  
    "iterations": {  
        "com.discovery.stroke.Iterations": {  
            "number": 1  
        }  
    },  
    "workdir": {  
        "com.discovery.stroke.Workdir": {  
            "path": "C:/xampp/htdocs/OpenEMR-  
7.0.2/interface/modules/custom_modules/oe-module-jbpm-generic-  
callout/public"  
        }  
    }  
}
```

```
        },
        "dropcols": {
            "com.discovery.stroke.Dropcols": {
                "droplist": "[['id', 'gender', 'ever_married', 'work_type', 'residence_type', 'smoking_status', 'stroke']]"
            }
        },
        "samplestruct": {
            "com.discovery.stroke.Samplestruct": {
                "balancing": 1
            }
        },
        "manage": {
            "com.discovery.stroke.Manage": {
                "publish": 1
            }
        },
        "train": {
            "com.discovery.stroke.Train": {
                "training": 1,
                "folds": 5,
                "bagfrac": 0.99,
                "weights": "",
                "weightsaccval": 1,
                "weightsacctst": 0
            }
        },
        "test": {
            "com.discovery.stroke.Test": {
                "preddraw": 1,
                "predfrac": 0.9
            }
        },
        "costfunc": {
            "com.discovery.stroke.Costfunc": {
                "costtype": "logistic"
            }
        }
    }
}
```

```

        "costoferror": 25.0,
        "costofchecking": 10.0
    }
},
"validate": {
    "com.discovery.stroke.Validate": {
        "testdraws": 1
    }
}
}

```

POST /server/containers/{containerId}/processes/{processId}/instances Starts a new process instance of a specified process.

Parameters

Name	Description
containerId * required	container id where the process definition resides
string (path)	stroke_1.0.0-SNAPSHOT
processId * required	process id that new instance should be created from
string (path)	stroke-train
body	optional map of process variables
(body)	Example Value Model

```
{
    "iterations": {
        "com.discovery.stroke.Iterations": {
            "number": 1
        }
    },
    "workdir": {
        "com.discovery.stroke.Workdir": {
            "path": "C:/xampp/htdocs/OpenEMR-7.0.2/interface/modules/custom_modules/oe-module-jbpm-generic-callout/public"
        }
    },
    "dropcols": {
        "com.discovery.stroke.Dropcols": {
            "droplist": ["id", "gender", "ever_married", "work_type", "residence_type", "smoking_status", "stroke"]
        }
    },
    "samplestruct": {
        "com.discovery.stroke.Samplestruct": {
            "balancing": 1
        }
    }
}
```

If the request ran OK, you will see a similar screen with status 201:

Execute Clear

Responses Response content type application/json

```
Curl
application/json -d "{\"iterations\": { \"com.discovery.stroke.Iterations\": { \"number\": 1 } }, \"workdir\": { \"com.discovery.stroke.Workdir\": { \"path\": \"C:/xampp/htdocs/OpenEMR-7.0.2/interface/modules/custom_modules/oe-module-jbpm-generic-callout/public\" } }, \"dropcols\": { \"com.discovery.stroke.Dropcols\": { \"droplist\": { \"id\", \"gender\", \"ever_married\", \"work_type\", \"residence_type\", \"smoking_status\", \"stroke\" } } }, \"samplestruct\": { \"com.discovery.stroke.Samplestruct\": { \"balancing\": 1 } }, \"manage\": { \"com.discovery.stroke.Manage\": { \"publish\": 1 } }, \"train\": { \"com.discovery.stroke.Train\": { \"training\": 1, \"folds\": 5, \"bagfrac\": 0.99, \"weights\": \"\", \"weightsaccval\": 1, \"weightsacctst\": 0 } }, \"test\": { \"com.discovery.stroke.Test\": { \"predraw\": 1, \"predfrac\": 0.9 } }, \"costfunc\": { \"com.discovery.stroke.Costfunc\": { \"costoferror\": 25.0, \"costofchecking\": 10.0 } }, \"validate\": { \"com.discovery.stroke.Validate\": { \"testdraws\": 1 } } }"
```

Server response

Code	Details
201	<p>Response body</p> <pre>995</pre> <p>Response headers</p> <pre>cache-control: no-cache, no-store, must-revalidate connection: keep-alive content-length: 3 content-type: application/json date: Sat, 22 Mar 2025 08:49:00 GMT expires: pragma: no-cache x-kie-conversationid: %27sample-server%27%3A%27stroke_1.0.0-SNAPSHOT%27%3A%27com.discovery.stroke%3Astroke%3A1.0.0-SNAPSHOT%27%3A%27c19a350b-3b18-4280-829c-ec2cb4a669%27</pre>

Responses

Code	Description
201	<i>Process instance started</i>

In Business Central, in the process instances log, there will be a record created that corresponds to your request:

Business Central

localhost:8080/business-central/kie-wb.jsp#ProcessInstances%5BWProcessInstanceListScreen%5BWProcessInstanceListBasicFiltersScreen%5D\$Pr...

995 - stroke-train

Détails de l'instance Variables de processus Documents Journaux Diagramme

Nom	Valeur	Type	Dernière modification	Actions
ACC	0.76		22-mars-2025 01:48:59	Historique
NPV	0.8		22-mars-2025 01:48:59	Historique
PPV	0.72		22-mars-2025 01:48:59	Historique
TNR	0.7407		22-mars-2025 01:48:59	Historique
TPR	0.7826		22-mars-2025 01:48:59	Historique
bagfrac	0.99		22-mars-2025 01:48:33	Historique
balancing	1		22-mars-2025 01:48:33	Historique
costfunc	com.discovery.stroke.Costfunc@50c47c9c	com.discovery.stroke.Costfunc	22-mars-2025 01:48:27	Historique
costofchecking	10.0		22-mars-2025 01:48:33	Historique
costoferror	25.0		22-mars-2025 01:48:33	Historique

10 Élement ▾ 1-10 de 34

In your file system, under the path specified earlier, you will see an assortment of various outputs generated by stroke-train process:

Nom	Date	Type	Taille	Mots clés
assets		Dossier de fichiers		
DatasetC.pkl	26/02/2025 19:48	Fichier PKL	15 Ko	
globals-sample.php	18/02/2025 21:31	Fichier PHP	4 Ko	
ModelDict.1.0.9.5.0.99.0.all.pkl	26/02/2025 19:48	Fichier PKL	6 745 Ko	
ohe_ever_married.pkl	26/02/2025 19:48	Fichier PKL	1 Ko	
ohe_gender.pkl	26/02/2025 19:48	Fichier PKL	1 Ko	
ohe_residence_type.pkl	26/02/2025 19:48	Fichier PKL	1 Ko	
ohe_smoking_status.pkl	26/02/2025 19:48	Fichier PKL	1 Ko	
ohe_work_type.pkl	26/02/2025 19:48	Fichier PKL	1 Ko	
sample-index.php	19/02/2025 01:04	Fichier PHP	8 Ko	
stroke.txt	18/02/2025 21:17	Document texte	1 Ko	
STROKE011.png	26/02/2025 19:48	Fichier PNG	59 Ko	
STROKE012.png	26/02/2025 19:48	Fichier PNG	27 Ko	
STROKE013.png	26/02/2025 19:48	Fichier PNG	169 Ko	
STROKE021.png	26/02/2025 19:48	Fichier PNG	67 Ko	
STROKE022.png	26/02/2025 19:48	Fichier PNG	27 Ko	
STROKE023.png	26/02/2025 19:48	Fichier PNG	169 Ko	
STROKE031.png	26/02/2025 19:48	Fichier PNG	27 Ko	
STROKE032.png	26/02/2025 19:48	Fichier PNG	23 Ko	
STROKE033.png	26/02/2025 19:48	Fichier PNG	31 Ko	
STROKE034.png	26/02/2025 19:48	Fichier PNG	46 Ko	
STROKERecord.csv	20/03/2025 01:27	Fichier CSV Micro...	1 Ko	

Proceed with the following request (using the same query):

containerid: stroke_1.0.0-SNAPSHOT

processid: stroke-act

body:

{

```

    "iterations": {
        "com.discovery.stroke.Iterations": {
            "number": 1
        }
    },
    "workdir": {
        "com.discovery.stroke.Workdir": {
            "path": "C:/xampp/htdocs/OpenEMR-
7.0.2/interface/modules/custom_modules/oe-module-jbpm-generic-
callout/public"
    }
}
```

```

        }
    },
    "dropcols": {
        "com.discovery.stroke.Dropcols": {
            "droplist":
            "[['id','gender','ever_married','work_type','residence_type','smoking_status','stroke']]"
        }
    },
    "samplestruct": {
        "com.discovery.stroke.Samplestruct": {
            "balancing": 1
        }
    },
    "act": {
        "com.discovery.stroke.Act": {
            "action": 1
        }
    },
    "train": {
        "com.discovery.stroke.Train": {
            "folds": 5,
            "bagfrac": 0.99
        }
    },
    "test": {
        "com.discovery.stroke.Test": {
            "predfrac": 0.9
        }
    },
    "record": {
        "com.discovery.stroke.Record": {
            "id": 9046,
            "gender": "Male",
            "age": 67,
            "hypertension": 0,
            "heart_disease": 0
        }
    }
}

```

```

        "heart_disease": 1,
        "ever_married": "Yes",
        "work_type": "Private",
        "residence_type": "Urban",
        "avg_glucose_level": 228.69,
        "bmi": 36.6,
        "smoking_status": "formerly smoked",
        "stroke": 1
    }
}

}

```

POST /server/containers/{containerId}/processes/{processId}/instances Starts a new process instance of a specified process.

Parameters

Name	Description
containerId <small>required</small>	container id where the process definition resides
string (path)	stroke_1.0.0-SNAPSHOT
processId <small>required</small>	process id that new instance should be created from
string (path)	stroke-act
body	optional map of process variables
(body)	Example Value Model <pre> }, "dropcols": { "com.discovery.stroke.Dropcols": { "droplist": ["'id'", "'gender'", "'ever_married'", "'work_type'", "'residence_type'", "'smoking_status'", "'stroke'"] } }, "samplestruct": { "com.discovery.stroke.Samplestruct": { "balancing": 1 } }, "act": { "com.discovery.stroke.Act": { "action": 1 } }, "train": { "com.discovery.stroke.Train": { "folds": 5, "bagfrac": 0.99 } } } }</pre>

Cancel

If the request ran OK, you will see a similar screen with status 201:

Execute Clear

Responses Response content type application/json

Curl

```
curl -X POST "http://localhost:8080/kie-server/services/rest/server/containers/stroke_1.0.0-SNAPSHOT/processes/stroke-act/instances" -H "accept: application/json" -H "content-type: application/json" -d "{\"iterations\": { \"com.discovery.stroke.Iterations\": { \"number\": 1 } }, \"workdir\": { \"com.discovery.stroke.Workdir\": { \"path\": \"/xampp/htdocs/OpenEMR-7.0.2/interface/modules/custom_modules/oem-module-jbpm-generic-callout/public/V\" } }, \"dropcols\": { \"com.discovery.stroke.Dropcols\": { \"droplist\": [\"id\", \"gender\", \"ever_married\", \"work_type\", \"residence_type\", \"smoking_status\", \"stroke\"] } }, \"samplestruct\": { \"com.discovery.stroke.Samplestruct\": { \"balancing\": 1 } }, \"act\": { \"com.discovery.stroke.Act\": { \"action\": 1 } }, \"train\": { \"com.discovery.stroke.Train\": { \"folds\": 5, \"bagfrac\": 0.99 } }, \"test\": { \"com.discovery.stroke.Test\": { \"predfrac\": 0.9 } }, \"record\": { \"com.discovery.stroke.Record\": { \"id\": 9946, \"gender\": \"Male\", \"age\": 67, \"hypertension\": 0, \"heart_disease\": 1, \"t\" } } }
```

Server response

Code	Details
201	<p>Response body</p> <pre>996</pre> <p>Response headers</p> <pre>cache-control: no-cache, no-store, must-revalidate connection: keep-alive content-length: 3 content-type: application/json date: Sat, 22 Mar 2025 20:39:33 GMT expires: 0 pragma: no-cache x-kie-conversationid: %27sample-server%27%3A%27stroke_1.0.0-SNAPSHOT%27%3A%27com.discovery.stroke%3Astroke%3A1.0.0-SNAPSHOT%27%3A%270b71540d-bc7c-4e72-8c08-2ca4b95a374e%27</pre>

Responses

Code	Description
201	<i>Process instance started</i>

In Business Central, in the process instances log, there will be a record created that corresponds to your request:

Nom	Valeur	Type	Dernière modification	Actions
path	C:/xampp/htdocs/OpenEMR-7.0.2/interface/...		22-mars-2025 21:39:29	Historique
predfrac	0.9		22-mars-2025 21:39:29	Historique
prediction00	0.8934		22-mars-2025 21:39:32	Historique
record	com.discovery.stroke.Record@14ee9324	com.discovery.stroke.Record	22-mars-2025 21:39:26	Historique
residence_type	Urban		22-mars-2025 21:39:29	Historique
samplestruct	com.discovery.stroke.Samplestruct@2dd66e...	com.discovery.stroke.Samplestruct	22-mars-2025 21:39:26	Historique
smoking_status	formerly smoked		22-mars-2025 21:39:29	Historique
stroke	1		22-mars-2025 21:39:29	Historique
subinterp			22-mars-2025 21:39:32	Historique
test	com.discovery.stroke.Test@739b2920	com.discovery.stroke.Test	22-mars-2025 21:39:26	Historique

IMPORTANT: if testing by sending concurrent requests (e.g., from separate browser tabs with KIE Server Documentation portal), make sure in Business Central that only one deployment unit (the one that carries `stroke` project) is running while all the other deployment units are stopped.

