



SOLAR VILLAGE

POC

Advanced Process Development with Red Hat
Process Automation Manager Assignment

ABSTRACT

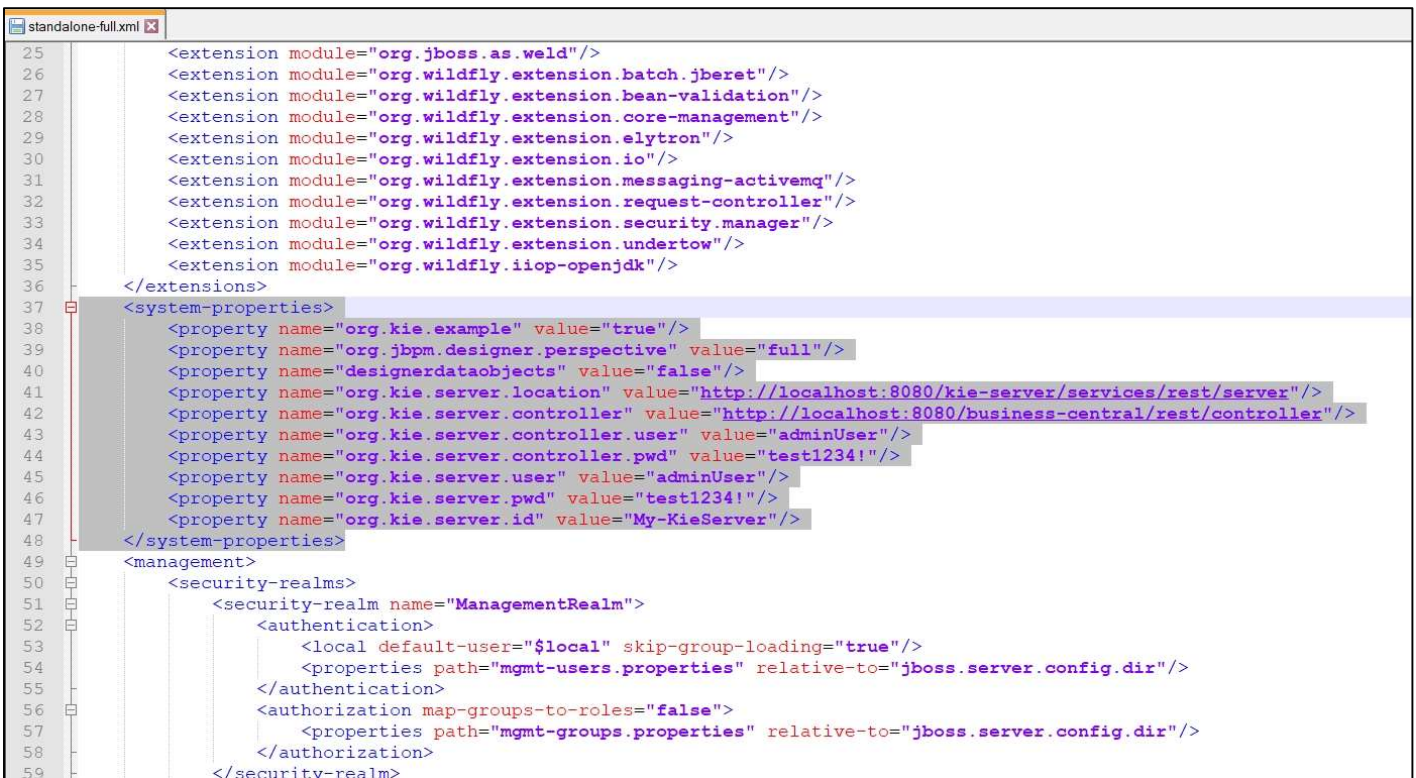
A RedHat Process Automation Manager solution
for automation of New Order processing.

AUTHOR

SAI VAMSEE BONTA

1. Managed Kie Server Configuration

- We are using a managed kie-server configuration in this project.
- To configure navigate to the below path:
 >%EAP_HOME%\standalone\configuration\standalone-full.xml
- And check that the below highlighted system properties are configured properly.



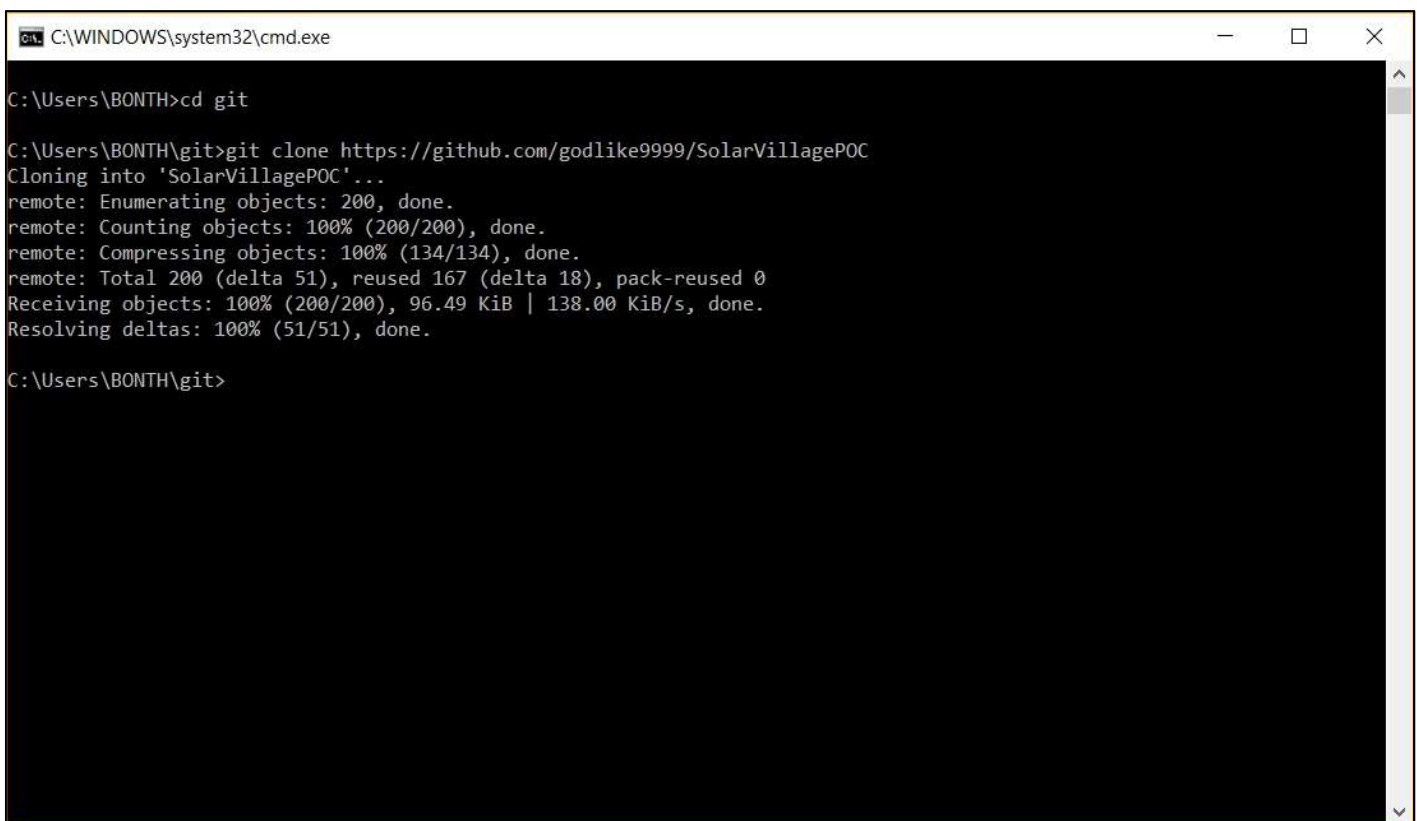
```

25 <extension module="org.jboss.as.weld"/>
26 <extension module="org.wildfly.extension.batch.jberet"/>
27 <extension module="org.wildfly.extension.bean-validation"/>
28 <extension module="org.wildfly.extension.core-management"/>
29 <extension module="org.wildfly.extension.elytron"/>
30 <extension module="org.wildfly.extension.io"/>
31 <extension module="org.wildfly.extension.messaging-activemq"/>
32 <extension module="org.wildfly.extension.request-controller"/>
33 <extension module="org.wildfly.extension.security.manager"/>
34 <extension module="org.wildfly.extension.undertow"/>
35 <extension module="org.wildfly.iioption-openjdk"/>
36 </extensions>
37 <system-properties>
38 <property name="org.kie.example" value="true"/>
39 <property name="org.jbpm.designer.perspective" value="full"/>
40 <property name="designerdataobjects" value="false"/>
41 <property name="org.kie.server.location" value="http://localhost:8080/kie-server/services/rest/server"/>
42 <property name="org.kie.server.controller" value="http://localhost:8080/business-central/rest/controller"/>
43 <property name="org.kie.server.controller.user" value="adminUser"/>
44 <property name="org.kie.server.controller.pwd" value="test1234!"/>
45 <property name="org.kie.server.user" value="adminUser"/>
46 <property name="org.kie.server.pwd" value="test1234!"/>
47 <property name="org.kie.server.id" value="My-KieServer"/>
48 </system-properties>
49 <management>
50 <security-realms>
51 <security-realm name="ManagementRealm">
52 <authentication>
53 <local default-user="$local" skip-group-loading="true"/>
54 <properties path="mgmt-users.properties" relative-to="jboss.server.config.dir"/>
55 </authentication>
56 <authorization map-groups-to-roles="false">
57 <properties path="mgmt-groups.properties" relative-to="jboss.server.config.dir"/>
58 </authorization>
59 </security-realm>

```

2. Clone and Build Project

- The project contains four maven modules:
 1. domainModel
 2. govtPermit
 3. customWorkItemHandlers
 4. NewOrders
 - The **domainModel** contains all the POJOs and controllers.
 - The **govtPermit** provides a mock REST service for govt. Permit process
 - The **customWorkItemHandlers** are implemented to communicate with the REST API.
 - The **NewOrders** contains business processes.
- ✓ Clone the git project as shown below.



```
C:\WINDOWS\system32\cmd.exe

C:\Users\BONTH>cd git

C:\Users\BONTH\git>git clone https://github.com/godlike9999/SolarVillagePOC
Cloning into 'SolarVillagePOC'...
remote: Enumerating objects: 200, done.
remote: Counting objects: 100% (200/200), done.
remote: Compressing objects: 100% (134/134), done.
remote: Total 200 (delta 51), reused 167 (delta 18), pack-reused 0
Receiving objects: 100% (200/200), 96.49 KiB | 138.00 KiB/s, done.
Resolving deltas: 100% (51/51), done.

C:\Users\BONTH\git>
```

- ✓ Build the project using Maven as shown below.

```
C:\WINDOWS\system32\cmd.exe - mvn clean install

C:\Users\BONTH\git>cd SolarVillagePOC

C:\Users\BONTH\git\SolarVillagePOC>mvn clean install
[INFO] Scanning for projects...
[INFO] -----
[INFO] Reactor Build Order:
[INFO]
[INFO] SolarVillagePOC [pom]
[INFO] domainModel [jar]
[INFO] govtPermit [war]
[INFO] customWorkItemHandlers [jar]
[INFO] NewOrders [kjar]
[INFO]
[INFO] -----< org.solarVillage:SolarVillagePOC >-----
[INFO] Building SolarVillagePOC 0.0.1-SNAPSHOT [1/5]
[INFO] -----[ pom ]-----
```

```
C:\WINDOWS\system32\cmd.exe

[INFO] --- maven-compiler-plugin:3.8.0-jboss-2:testCompile (default-testCompile) @ NewOrders ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:3.0.0-M3:test (default-test) @ NewOrders ---
[INFO] --- maven-jar-plugin:3.1.1:jar (default-jar) @ NewOrders ---
[INFO] Building jar: C:\Users\BONTH\git\SolarVillagePOC\NewOrders\target\NewOrders-1.0.0.jar
[INFO] --- maven-install-plugin:3.0.0-M1:install (default-install) @ NewOrders ---
[INFO] Installing C:\Users\BONTH\git\SolarVillagePOC\NewOrders\target\NewOrders-1.0.0.jar to C:\Users\BONTH\.m2\repository\org\solarVillage\NewOrders\1.0.0\NewOrders-1.0.0.jar
[INFO] Installing C:\Users\BONTH\git\SolarVillagePOC\NewOrders\pom.xml to C:\Users\BONTH\.m2\repository\org\solarVillage\NewOrders\1.0.0\NewOrders-1.0.0.pom
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] SolarVillagePOC 0.0.1-SNAPSHOT ..... SUCCESS [ 1.853 s]
[INFO] domainModel 0.0.1-SNAPSHOT ..... SUCCESS [ 8.194 s]
[INFO] govtPermit 0.0.1-SNAPSHOT ..... SUCCESS [ 5.799 s]
[INFO] customWorkItemHandlers 0.0.1-SNAPSHOT ..... SUCCESS [ 1.049 s]
[INFO] NewOrders 1.0.0 ..... SUCCESS [ 15.409 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 50.970 s
[INFO] Finished at: 2019-01-04T01:06:36+05:30
[INFO] -----

C:\Users\BONTH\git\SolarVillagePOC>
```

3. Deploy govtPermit war

- Deploy the govtPermit war file into the EAP Server as Shown below.

```

C:\WINDOWS\system32\cmd.exe

C:\Users\BONTH\git\SolarVillagePOC>robocopy "%USERPROFILE%\git\SolarVillagePOC\govtPermit\target" "%EAP_HOME%\standalone\deployments" govtPermit.war

-----
ROBOCOPY    ::    Robust File Copy for Windows
-----

Started : 04 January 2019 01:12:10
Source  : C:\Users\BONTH\git\SolarVillagePOC\govtPermit\target\
Dest    : C:\RedHat\jboss-eap-7.1\standalone\deployments\

Files : govtPermit.war

Options : /DCOPY:DA /COPY:DAT /R:1000000 /W:30

-----
100%      New File           1    C:\Users\BONTH\git\SolarVillagePOC\govtPermit\target\
                               15.4 m      govtPermit.war
-----

      Total    Copied    Skipped    Mismatch    FAILED    Extras
Dirs  :        1         0         1         0         0         0
Files :        1         1         0         0         0         0
Bytes :   15.40 m   15.40 m         0         0         0         0
Times :    0:00:00    0:00:00             0:00:00  0:00:00

Speed :           19585963 Bytes/sec.
Speed :           1120.717 MegaBytes/min.
Ended : 04 January 2019 01:12:11

C:\Users\BONTH\git\SolarVillagePOC>

```

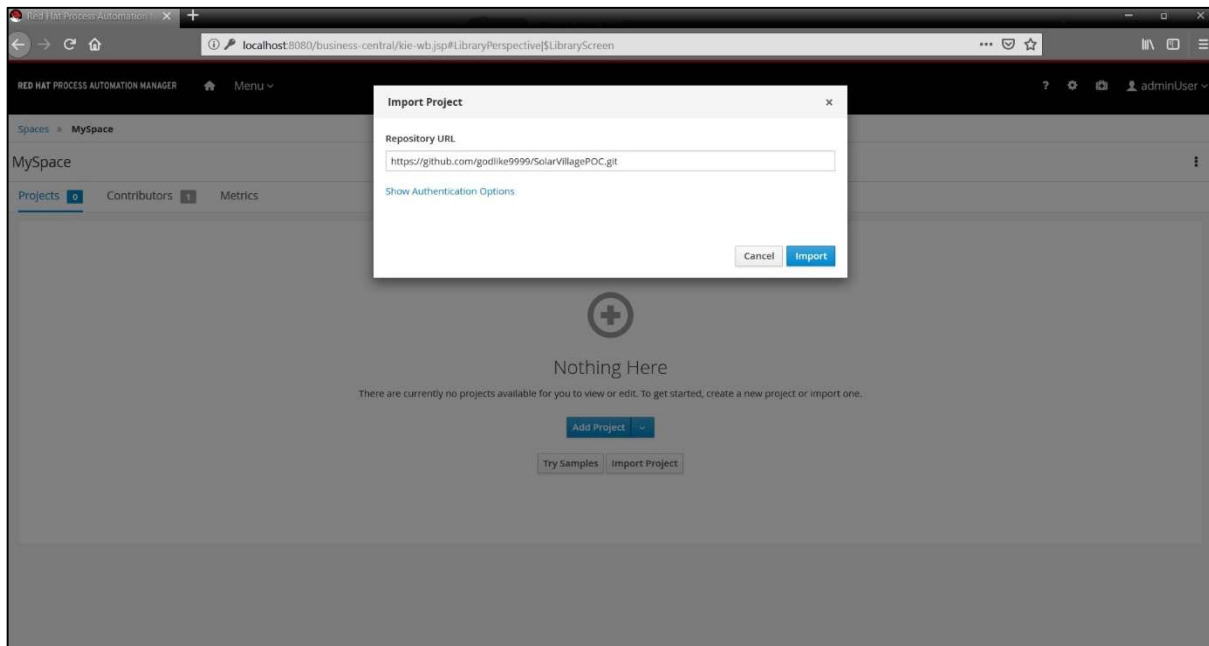
- After deployment, a file will be created with the extension **.deployed** for the govtPermit.war as shown.

PC > Local Disk (C:) > RedHat > jboss-eap-7.1 > standalone > deployments >

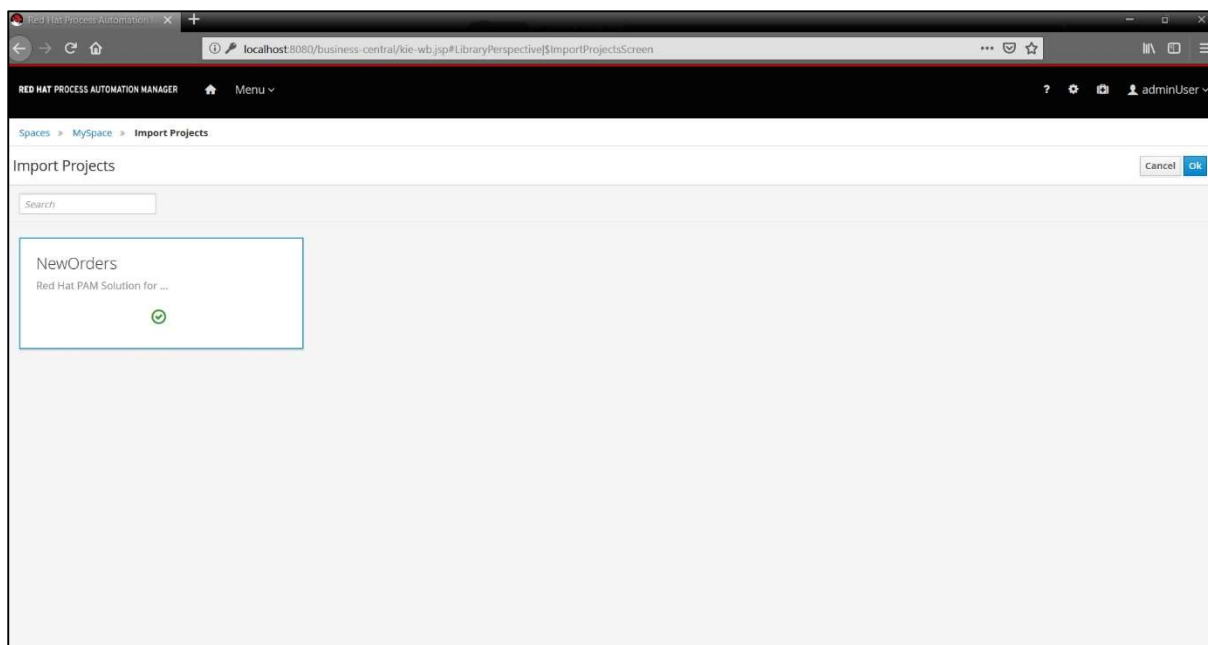
Name	Date modified	Type	Size
business-central.war	04-10-2018 06:52 ...	File folder	
kie-server.war	04-10-2018 06:52 ...	File folder	
business-central.war.deployed	30-12-2018 09:09 ...	DEPLOYED File	1 KB
govtPermit.war	04-01-2019 01:06 ...	WAR File	15,780 KB
govtPermit.war.deployed	04-01-2019 01:06 ...	DEPLOYED File	1 KB
kie-server.war.deployed	30-12-2018 09:09 ...	DEPLOYED File	1 KB
README.txt	04-10-2018 06:36 ...	Text Document	9 KB

4. Import NewOrders Project in PAM

- Open PAM Central page (<http://localhost:8080/business-central/kie-wb.jsp>) and login with your credentials.
- In the PAM Central, navigate to **Menu -> Design -> Projects**. Click on **Import project** and provide the git url under **Repository URL**.

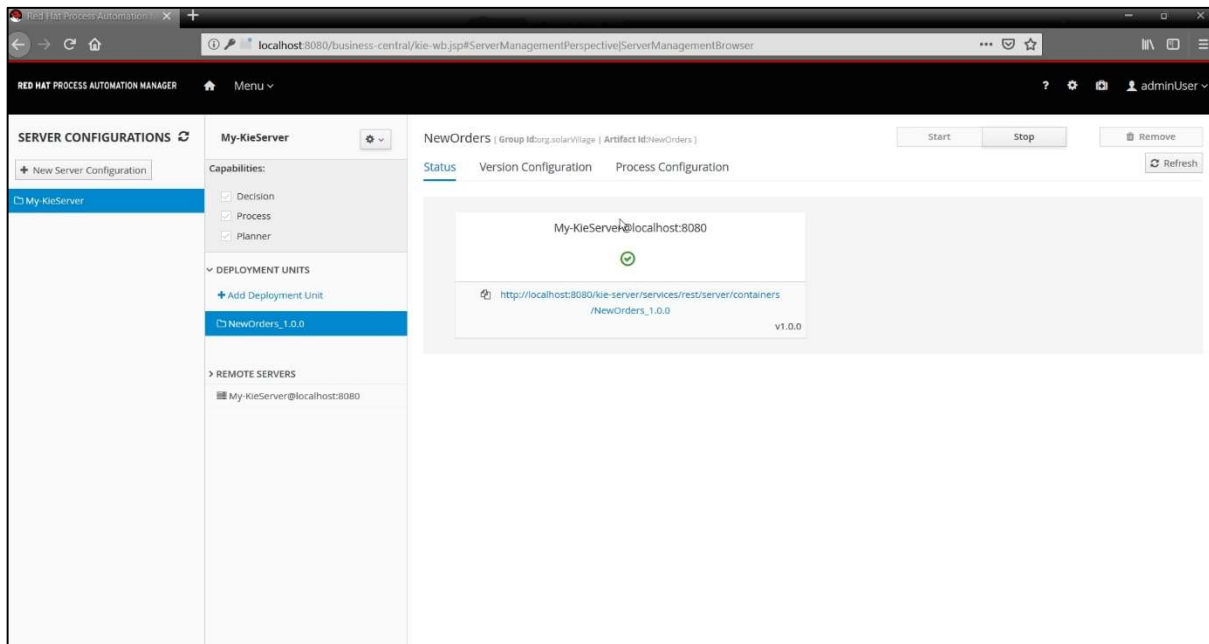


- Select the **NewOrders** project and click **OK** and wait for the import to complete.

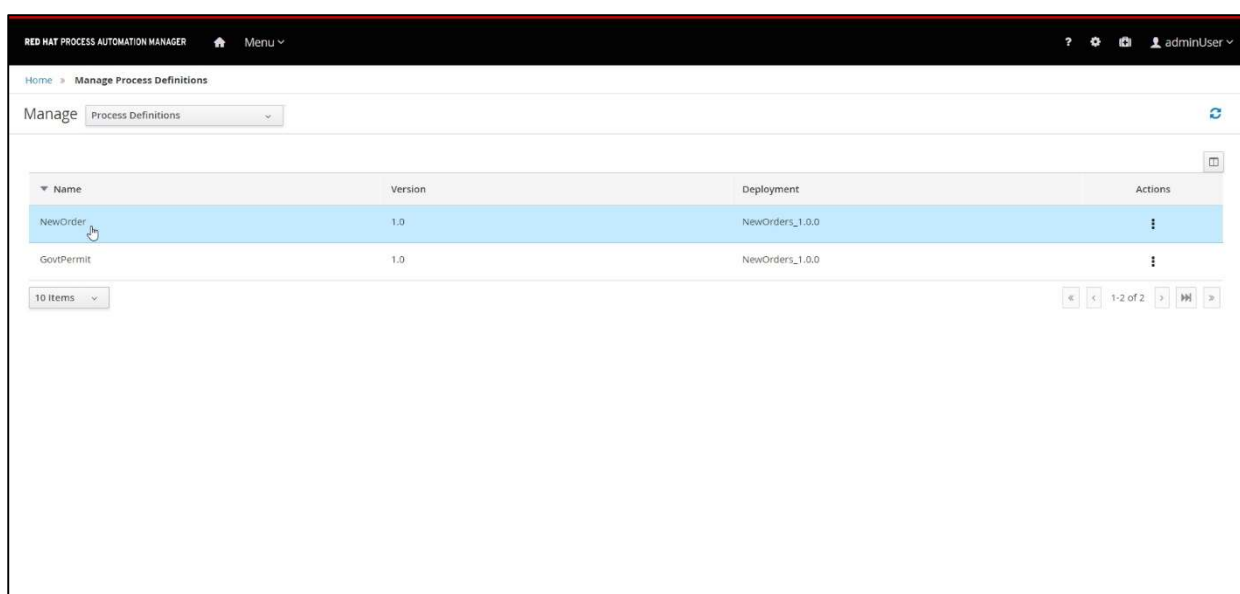


5. Build and Deploy Project in PAM

- Open the **NewOrders** project and click **Deploy** button present on top right.
- You should see a green message box saying Build and Deploy Successful.
- To verify deployment, navigate to **Menu -> Deploy -> Execution Servers** and you should be able to see a deployment unit named **NewOrders_1.0.0**.

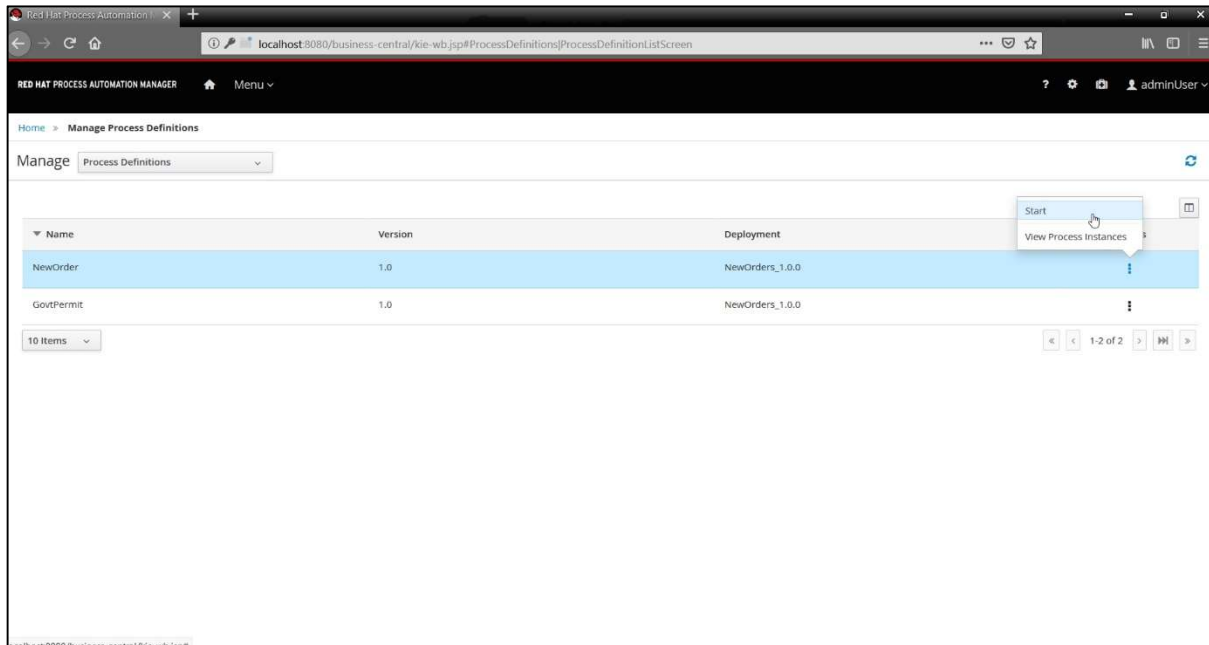


- Also, we can see that process definitions are generated for NewOrder and GovtPermit processes under **Menu -> Manage -> Process Definitions**.

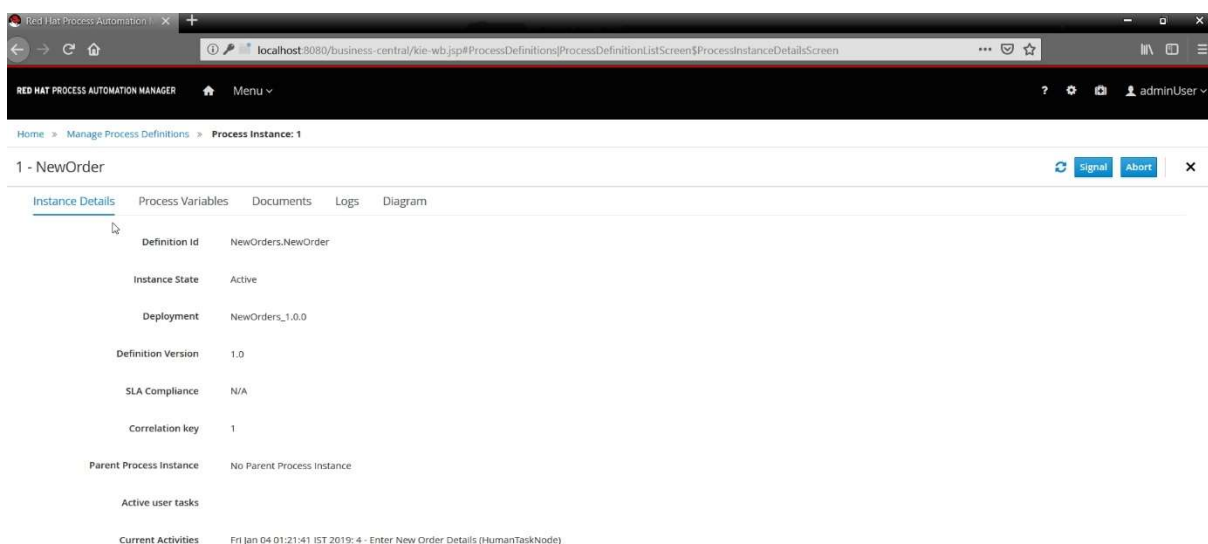


6. Sample Execution

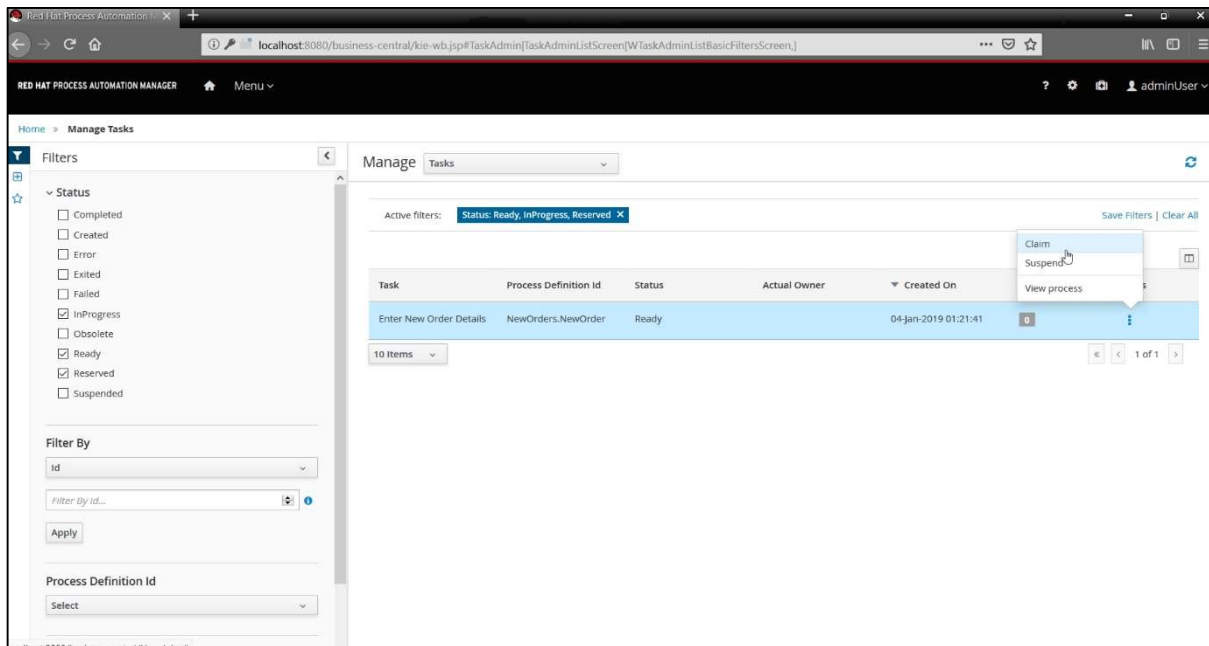
- Start a new process instance for the **NewOrder** process as shown below and click **Submit**.



- Now, you can see a new process instance being created as shown below.



- Go to **Menu -> Manage -> Tasks** and **claim** the Enter New Order Details user task as shown below.



- Now open the task, click **Start** and fill the new order form and click **Complete** once done.

The screenshot displays the '1 - Enter New Order Details' task form. The form has several input fields: 'Applicant Name*' (filled with 'Tim Cook'), 'Applicant Address*' (filled with 'Texas, USA'), 'Property Ownership Type*' (a dropdown menu showing 'Rented'), 'Electrical Connection Number*' (filled with '12345678'), 'Building Registration Number*' (filled with '77665544'), 'Building Description*' (filled with '3rd Floor in a 16 storey Apartment Building'), 'Application Approval Status*' (a dropdown menu showing 'InProgress'), and 'HOA Meeting Date' (filled with '01/10/2019'). At the bottom of the form, there are three buttons: 'Save', 'Release', and 'Complete'. The 'Complete' button is highlighted with a mouse cursor.

- Now claim the HOA Task and fill the HOA Meeting form.

Red Hat Process Automation Manager

Home > Manage Tasks > Task: 2

2 - HOAMeeting (Executive)

Work Details Assignments Comments Admin Logs

House Owners Association Meeting

Association Name *
Group Housing Association

Meeting Date *
01/10/2019

Representative Name *
Isabella

Association Head *
Michael

Meeting Details *
Association Approved the Installation Request.

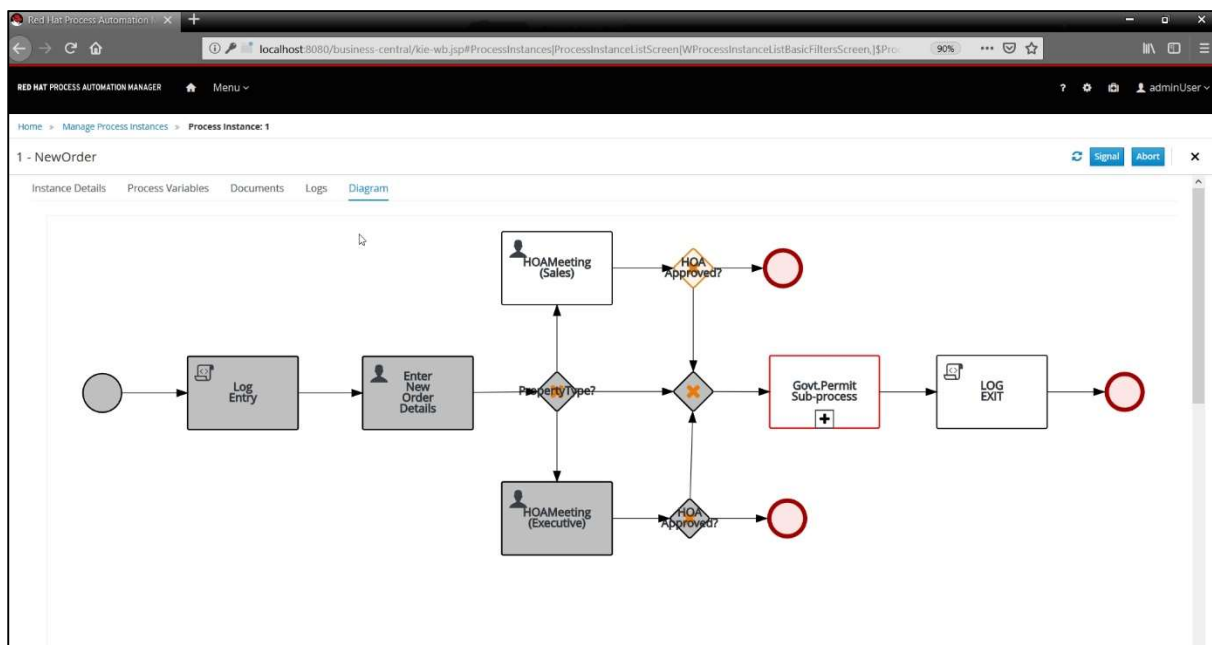
Approval Status *
Approved

Order

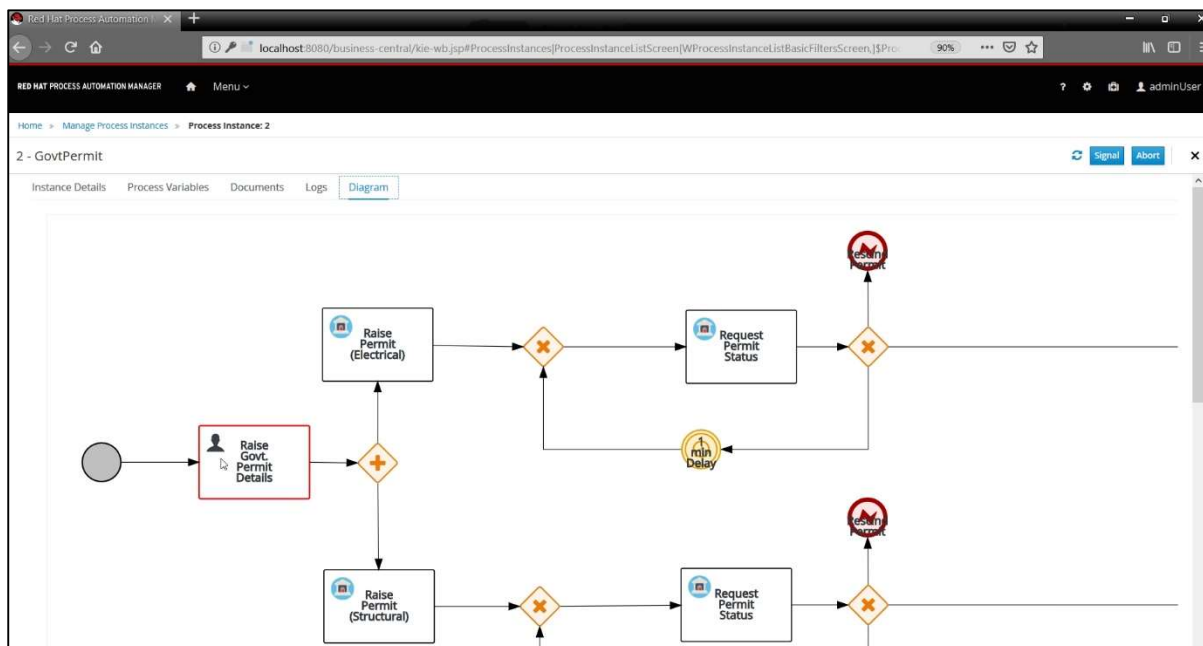
Applicant Name *
Tim Cook

Applicant Address *
Texas, USA

- From the NewOrder process instance we can see where we are in the process flow by clicking on the **Diagram** option.

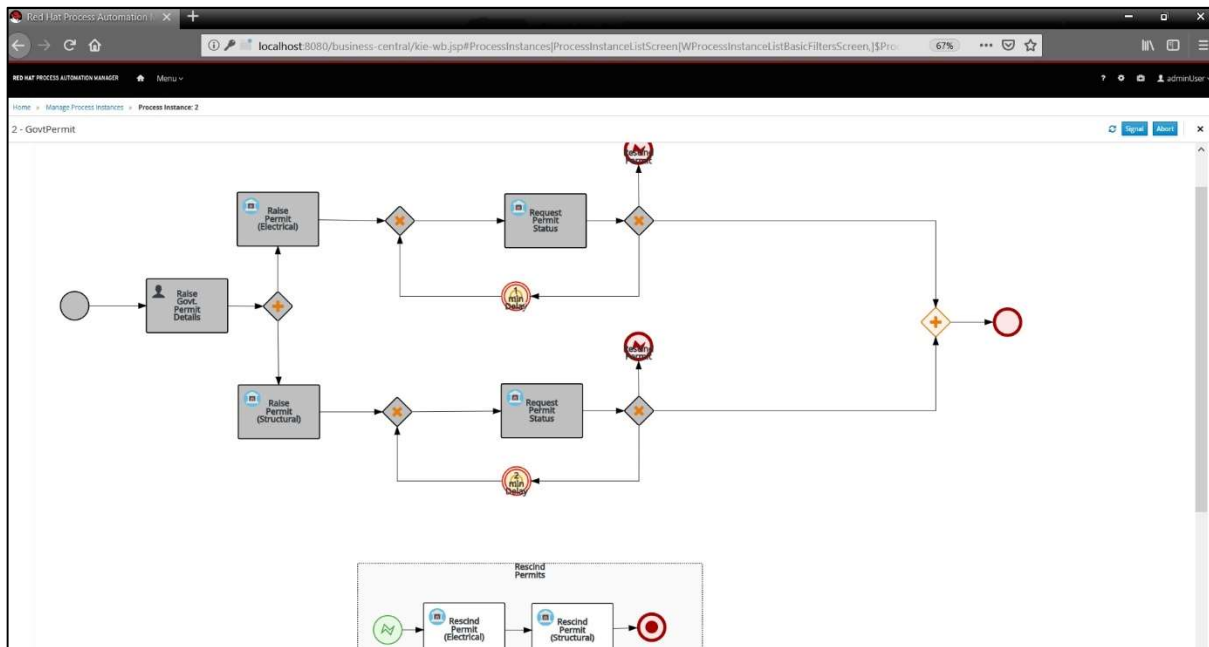


- After we reach the **Govt. Permit SubProcess** in the NewOrder flow, a new process instance for the GovtPermit is created.

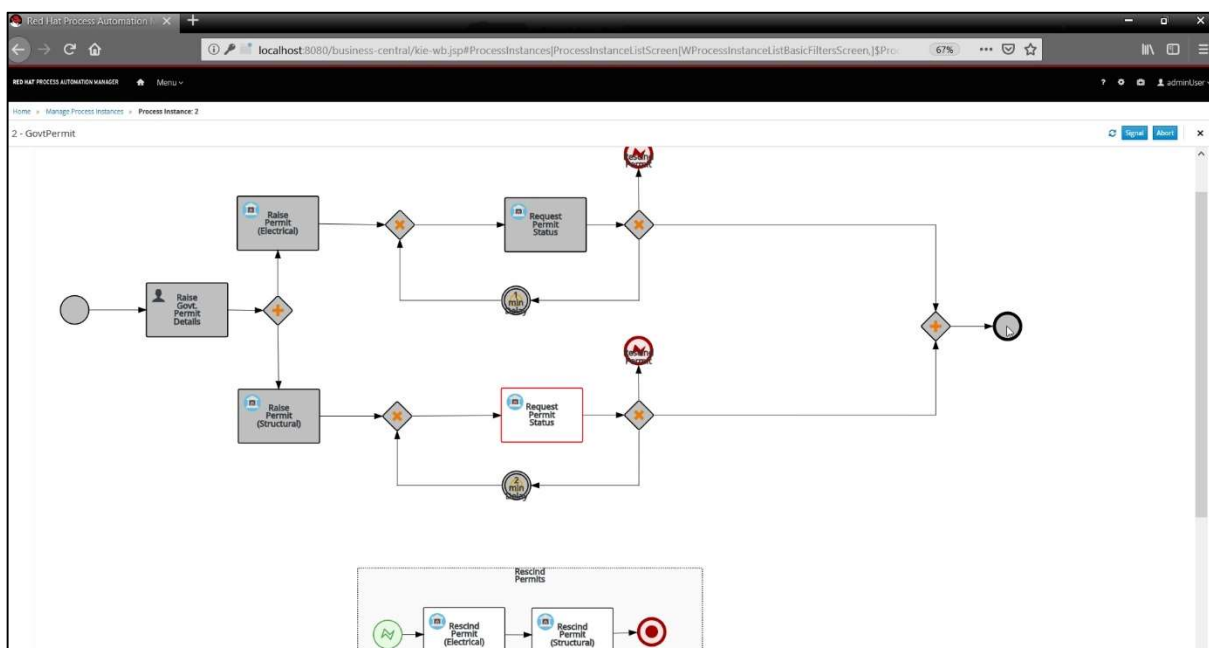


- Claim the user task **Raise Govt. Permit Details** and complete it by giving an application number as shown.

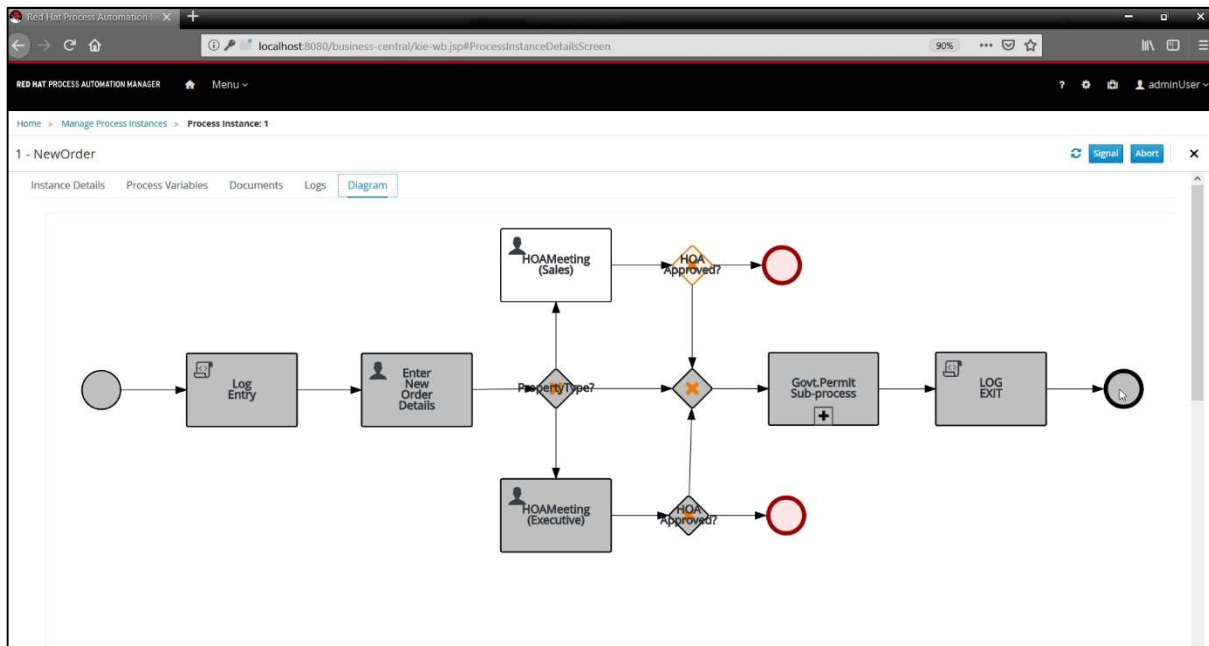
- From the below image we can see that the requests for Electrical and Structural Permits are raised parallelly.
- The status of the requests is checked every 2min for Structural Permit and every 1min for Electrical Permit.



- If any of the permit gets rejected then an interrupt will occur and both the permits get rescinded and the process is aborted.
- If both the permits are successful only, the process gets completed successfully.



- The NewOrder process instance completes successfully after the GovtPermit SubProcess is completed.



- We can also observe that both the process instances are changed to **Completed** state.

Manage Process Instances

Active filters: State: Completed

Id	Name	Description	Version	Last update	Errors	Actions
2	GovtPermit	GovtPermit	1.0	04-Jan-2019 01:37:19	0	
1	NewOrder	NewOrder	1.0	04-Jan-2019 01:37:19	0	

10 Items

7. Appendix

A. Electrical Permit Status Change

The electrical permit status can be changed in Realtime through a rest call.

Request Type: POST

URI :

[http://localhost:8080/govtPermit/rest/solar/changeElectricalStatus?id={\\$ApplicationNumber}&status={\\$Status}](http://localhost:8080/govtPermit/rest/solar/changeElectricalStatus?id={$ApplicationNumber}&status={$Status})

B. Structural Permit Status Change

The structural permit status can be changed in real-time through a rest call.

Request Type: POST

URI :

[http://localhost:8080/govtPermit/rest/solar/changeStructuralStatus?id={\\$ApplicationNumber}&status={\\$Status}](http://localhost:8080/govtPermit/rest/solar/changeStructuralStatus?id={$ApplicationNumber}&status={$Status})

C. Static Map for Electrical Permits

The content of the static hash map containing electrical permits can be seen in real-time through a rest call.

Request Type: GET

URI: <http://localhost:8080/govtPermit/rest/solar/getdataStoreE>

D. Static Map for Structural Permits

The content of the static hash map containing structural permits can be seen in real-time through a rest call.

Request Type: GET

URI: <http://localhost:8080/govtPermit/rest/solar/getdataStoreS>