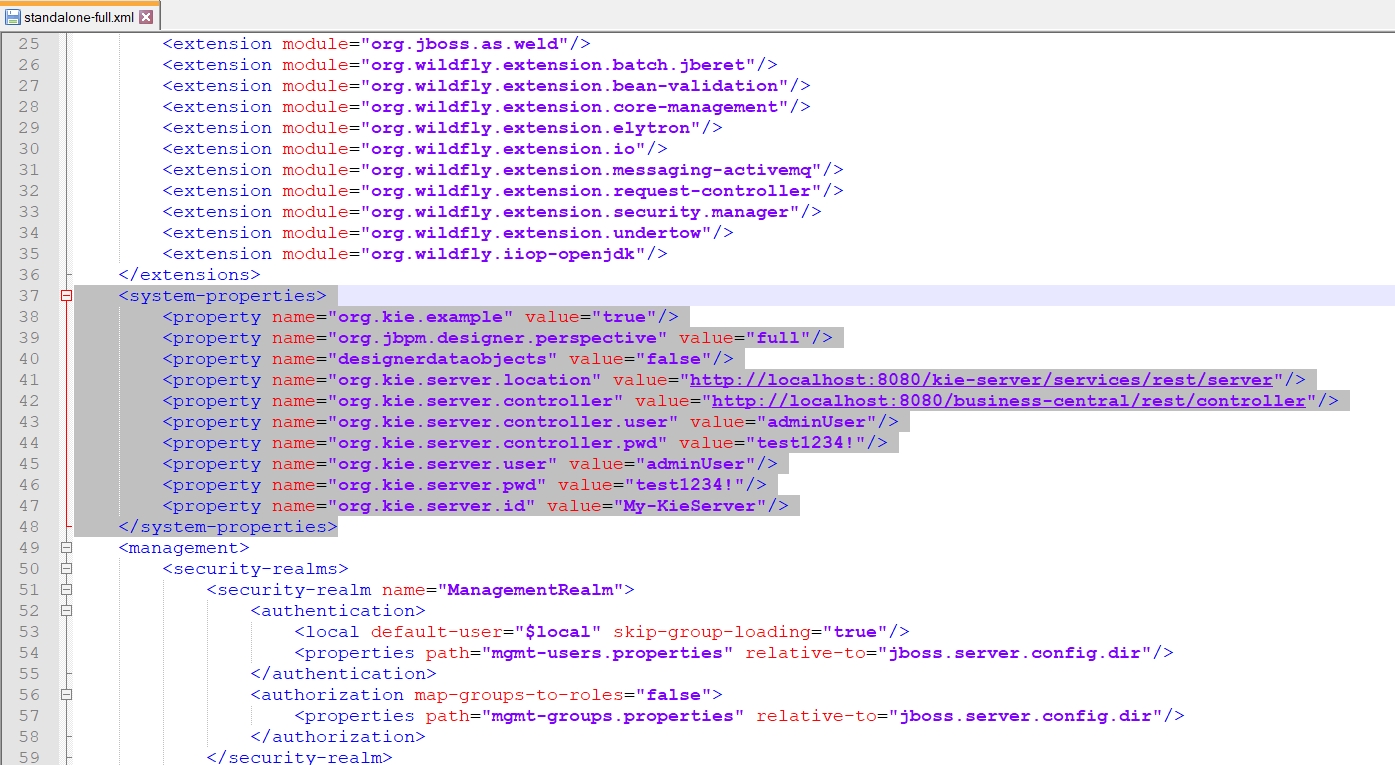
|  |  |
| --- | --- |
| case-tf9-001-finkenstein-solar-thermaflex-polybutylene-polybutene-pb1.jpg  SOLAR VILLAGE POC  **Advanced Process Development with Red Hat Process Automation Manager Assignment** | Abstract  A RedHat Process Automation Manager solution for automation of New Order processing.  Author  SAI VAMSEE BONTHA |

1. **Managed Kie Server Configuration**

* We are using a managed kie-server configuration in this project.
* To configure navigate to the below path:

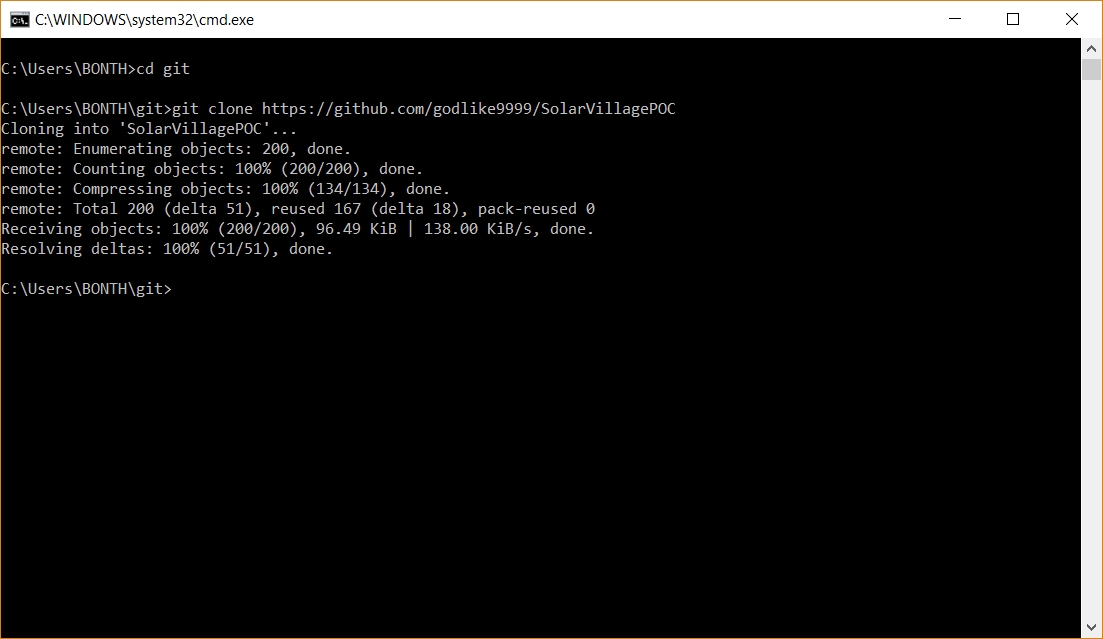
*>%EAP\_HOME%\standalone\configuration\standalone-full.xml*

* And check that the below highlighted system properties are configured properly.

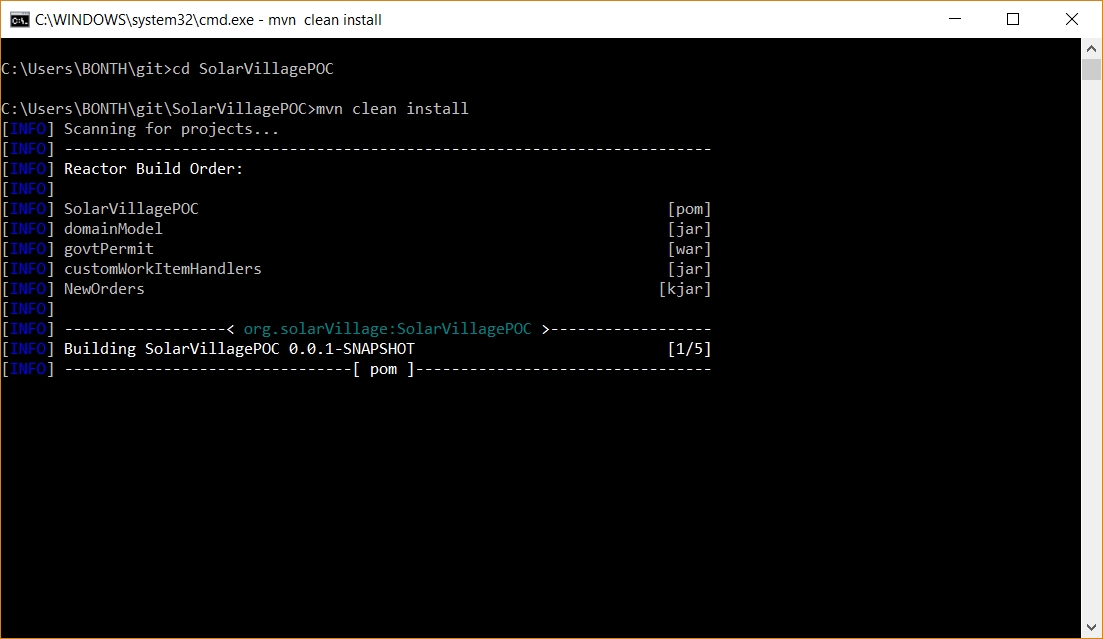
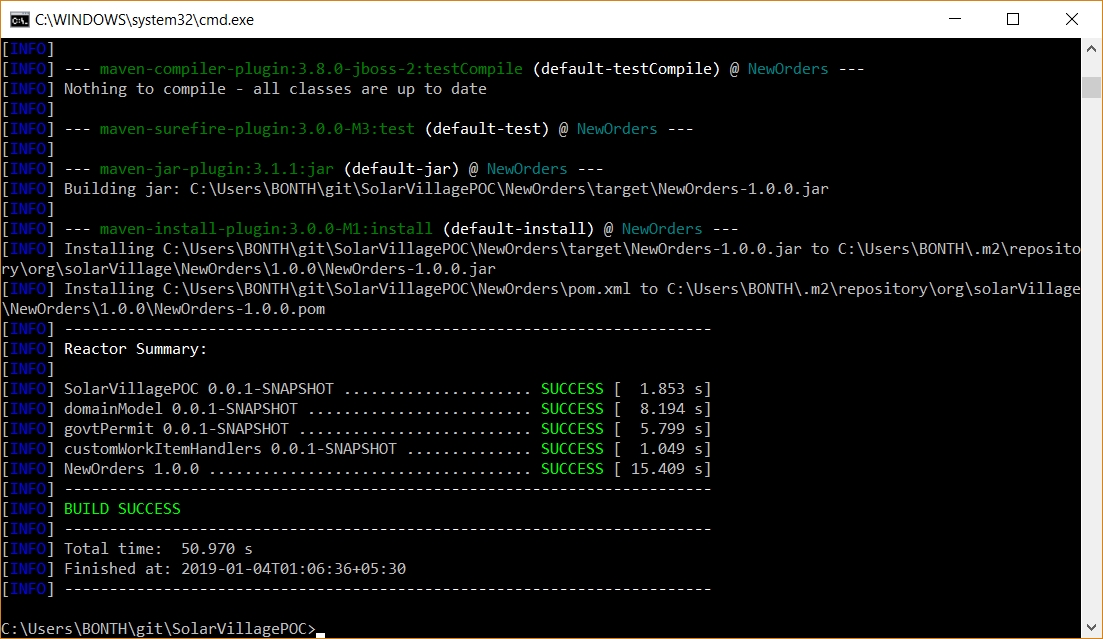


1. **Clone and Build Project**

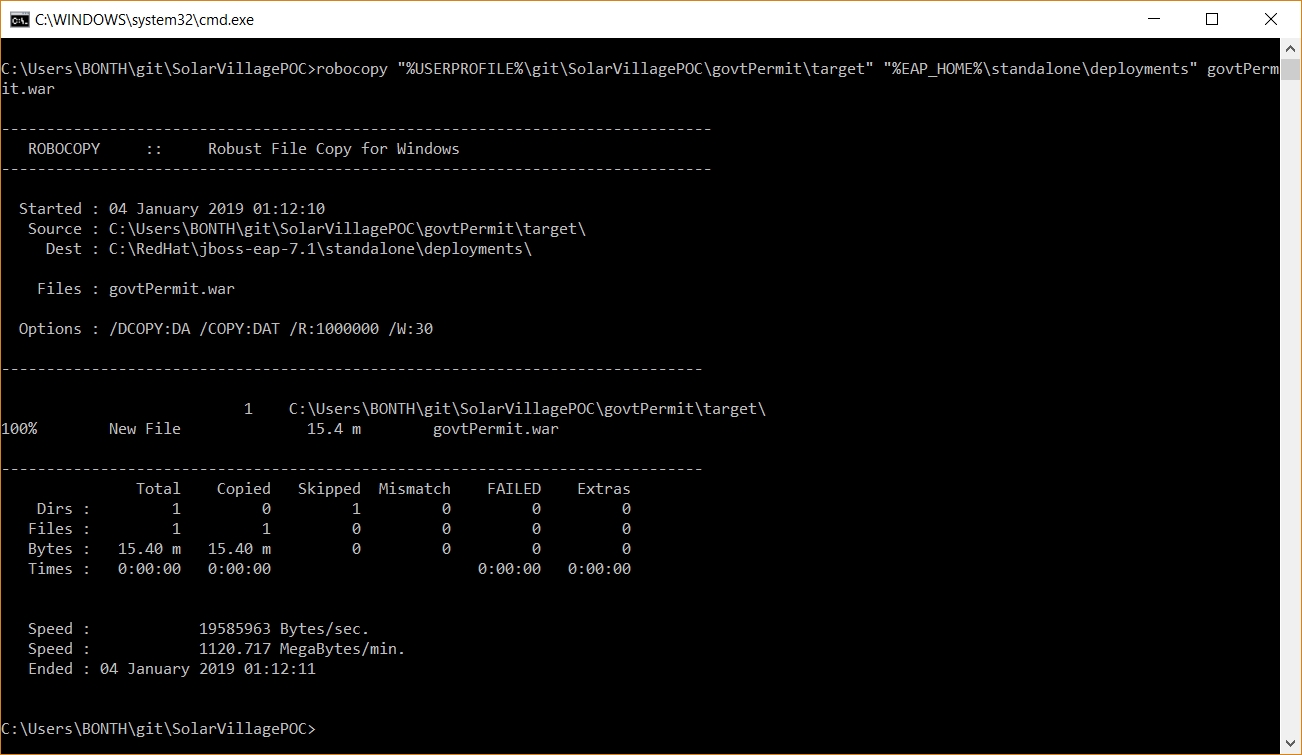
* The project contains four maven modules:
  1. domainModel
  2. govtPermit
  3. customWorkItemHandlers
  4. NewOrders
* The **domainModel** contains all the POJOs and controllers.
* The **govtPermit** provides a mock REST service for govt. Permit process
* The **customWorkItemHandlers** are implemented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

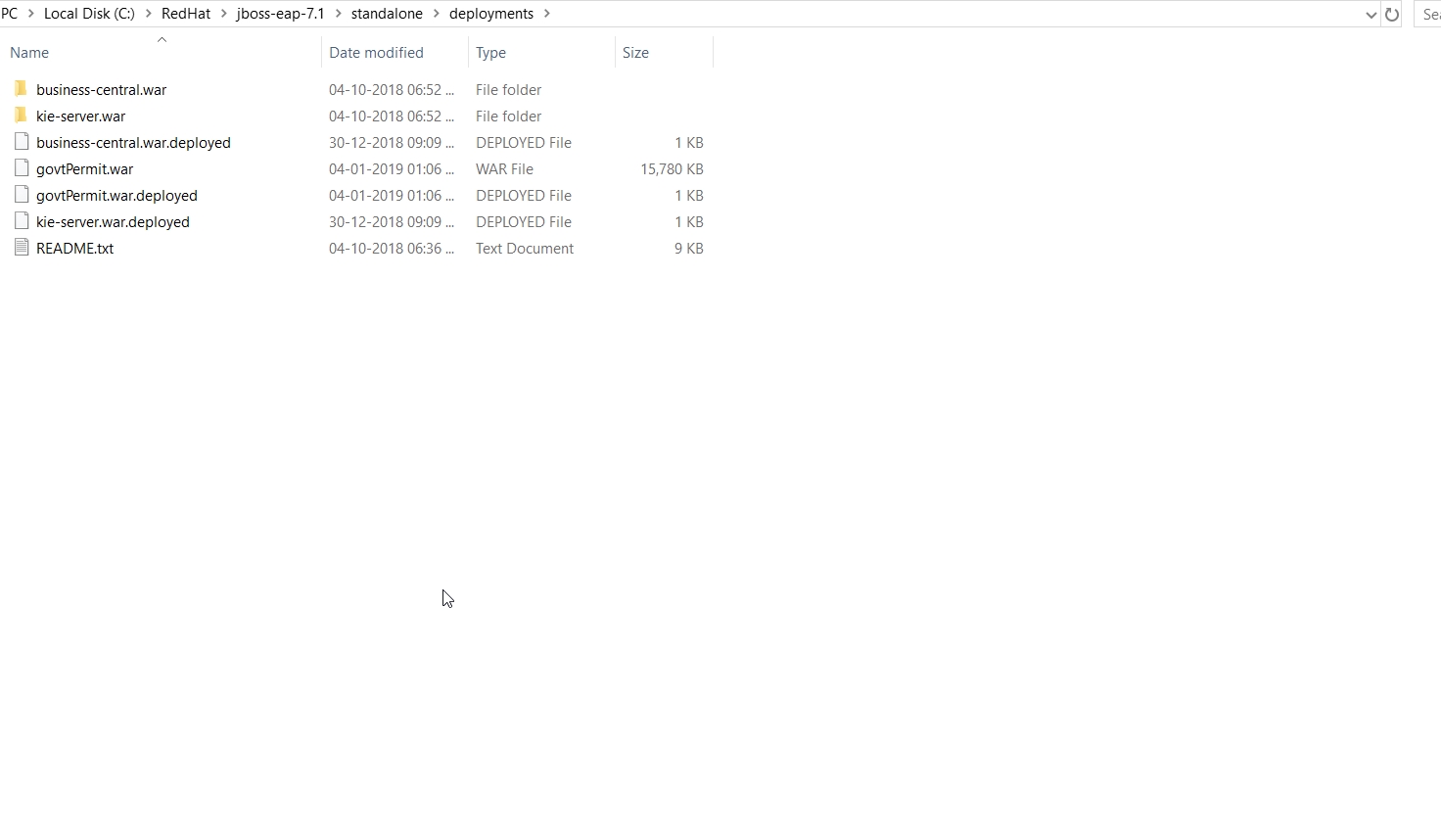


* Build the project using Maven as shown below.

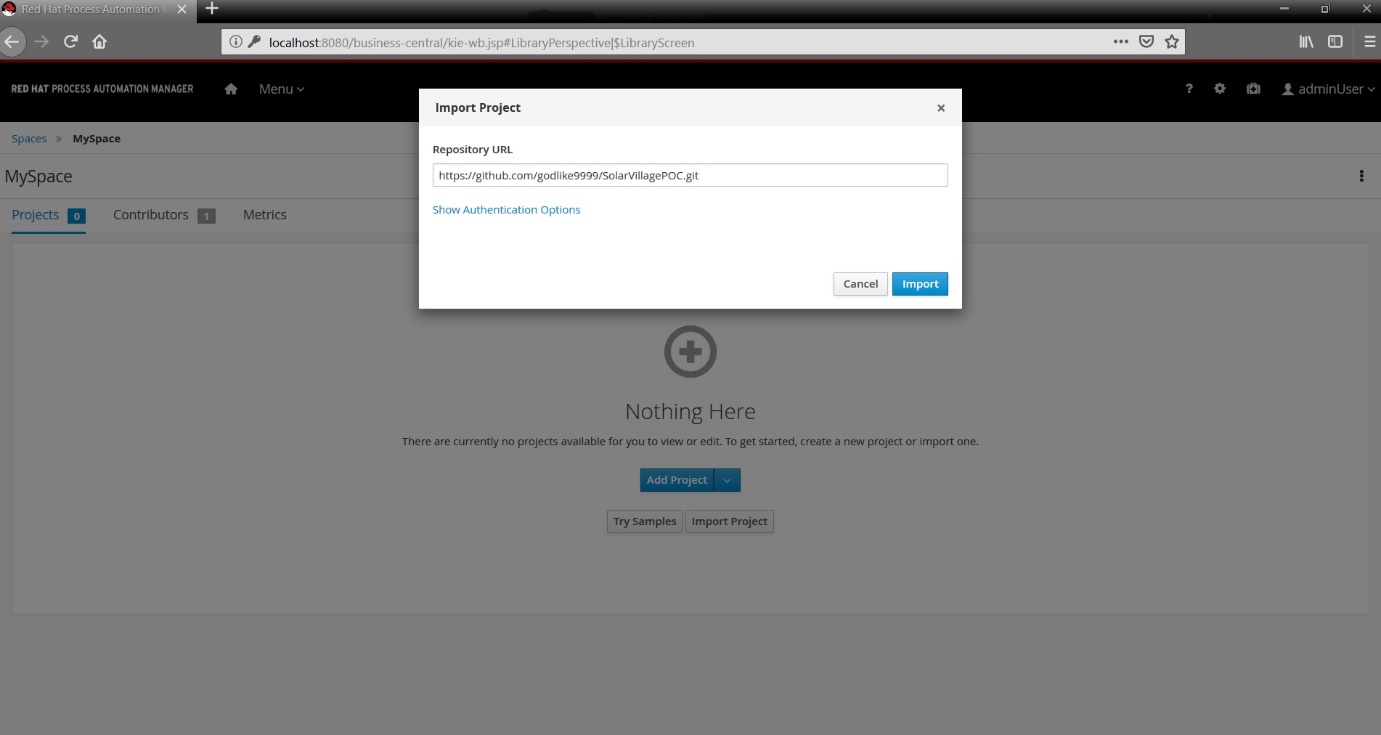


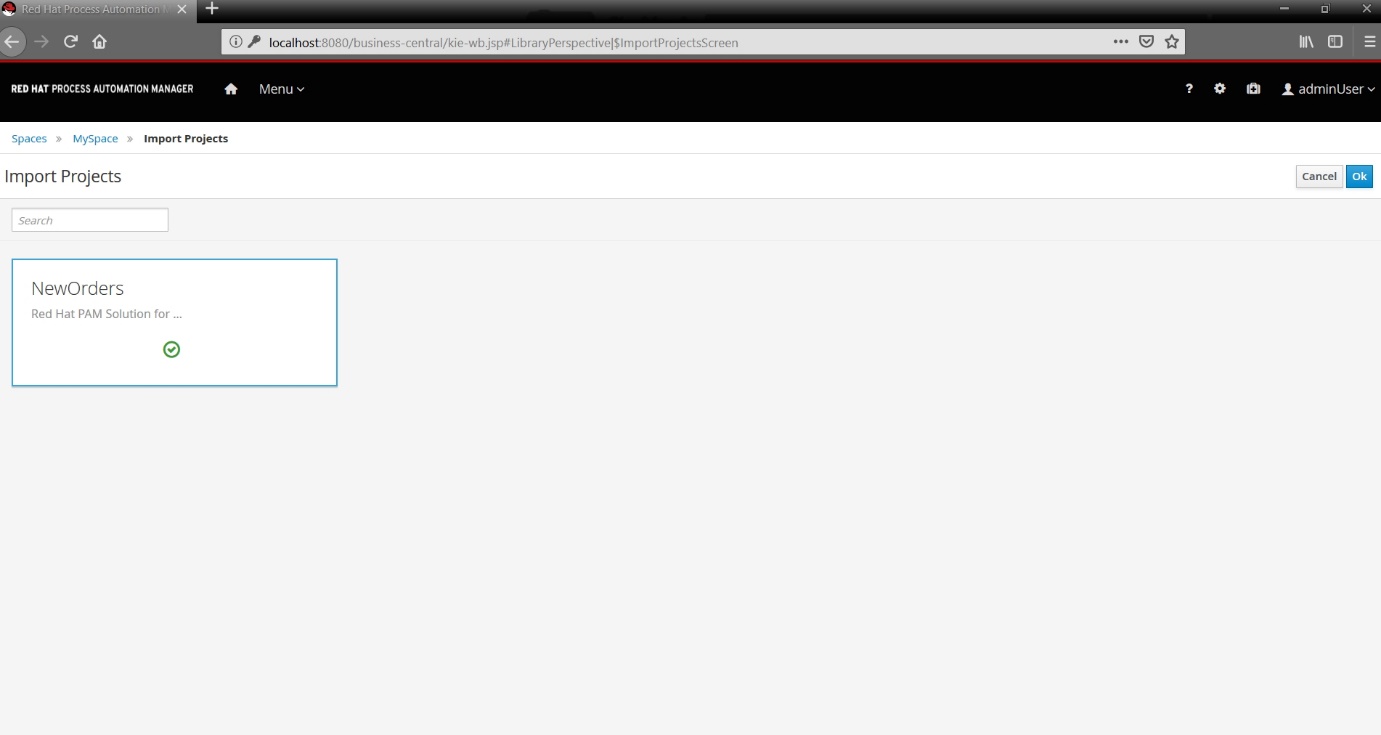
1. **Deploy govtPermit war**

* Deploy the govtPermit war file into the EAP Server as Shown below.
* After deployment, a file will be created with the extension **.deployed** for the govtPermit.war as shown.

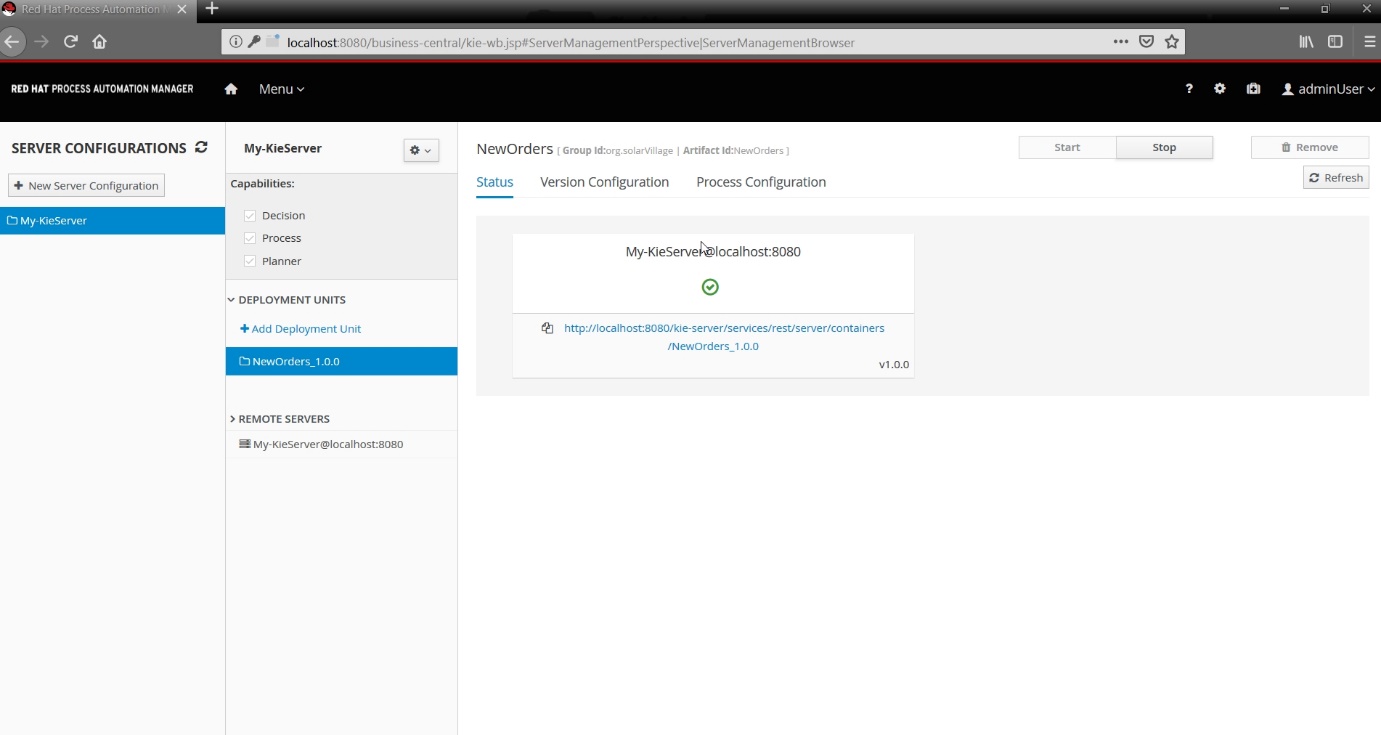
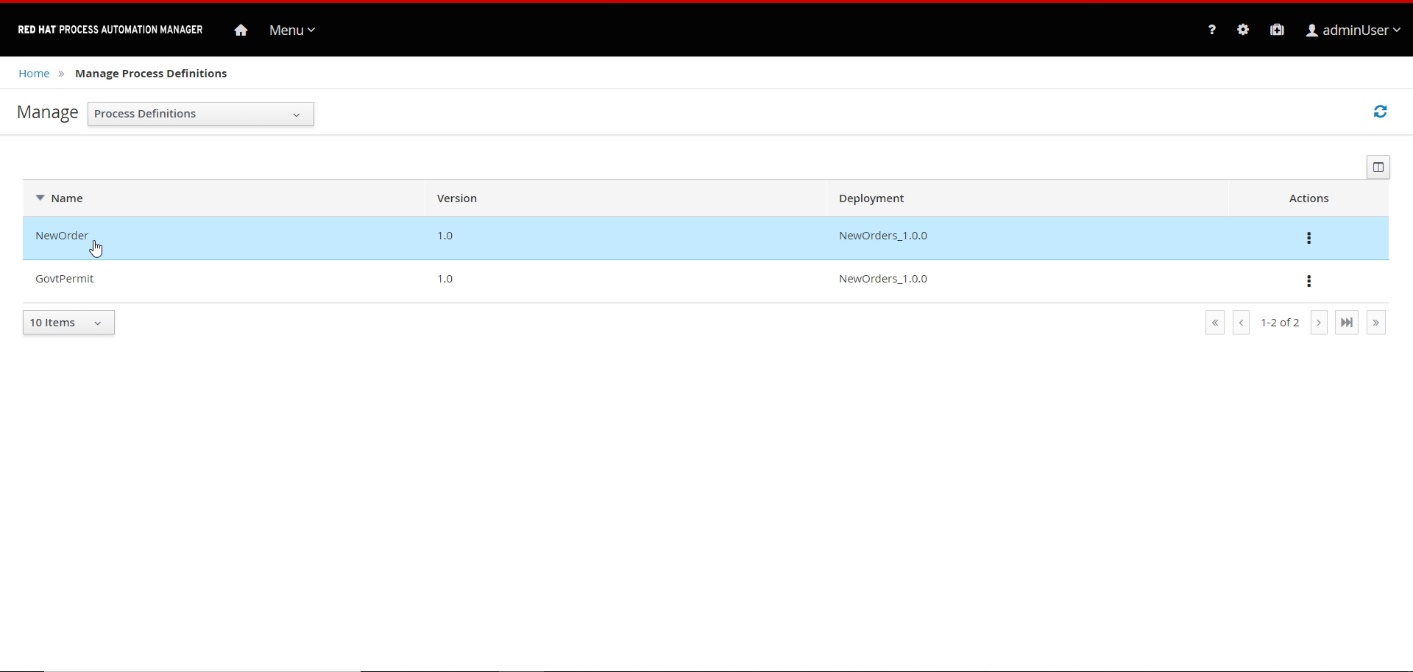


1. **Import NewOrders Project in PAM**

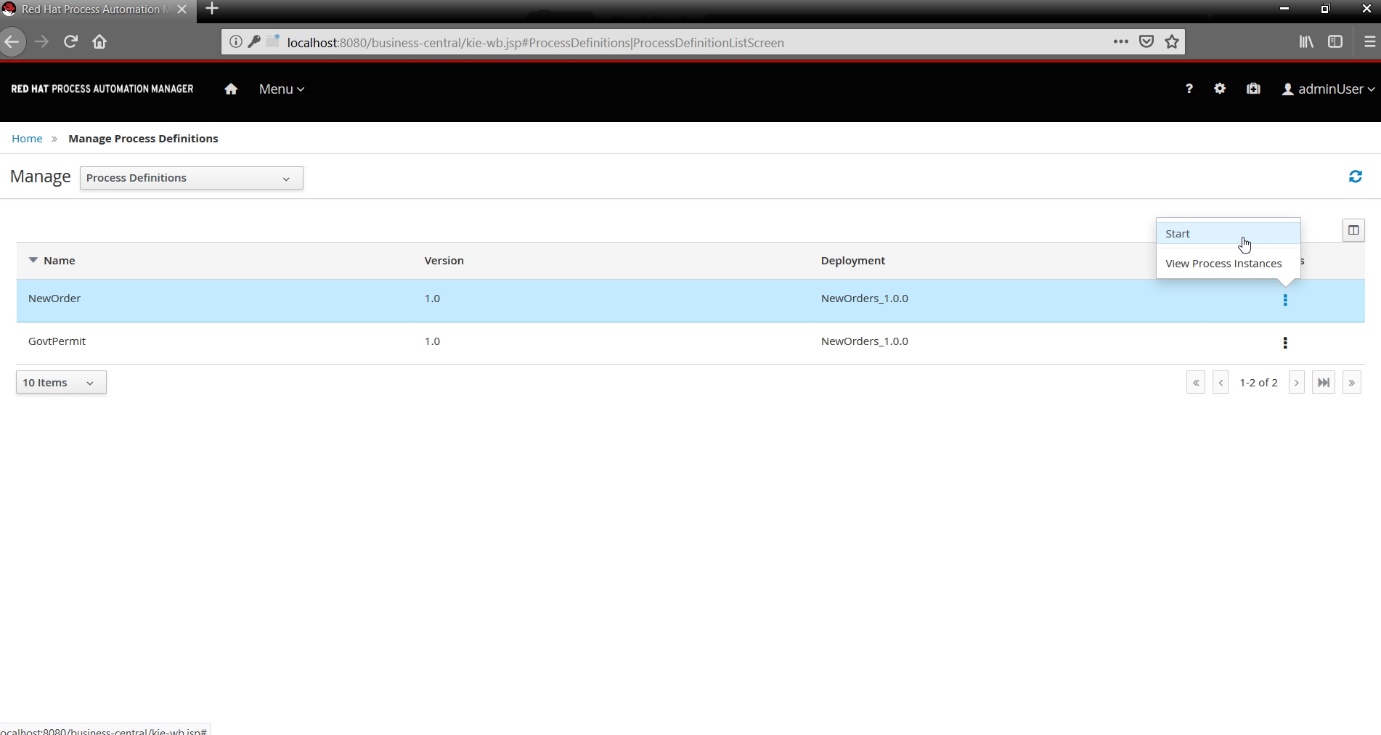
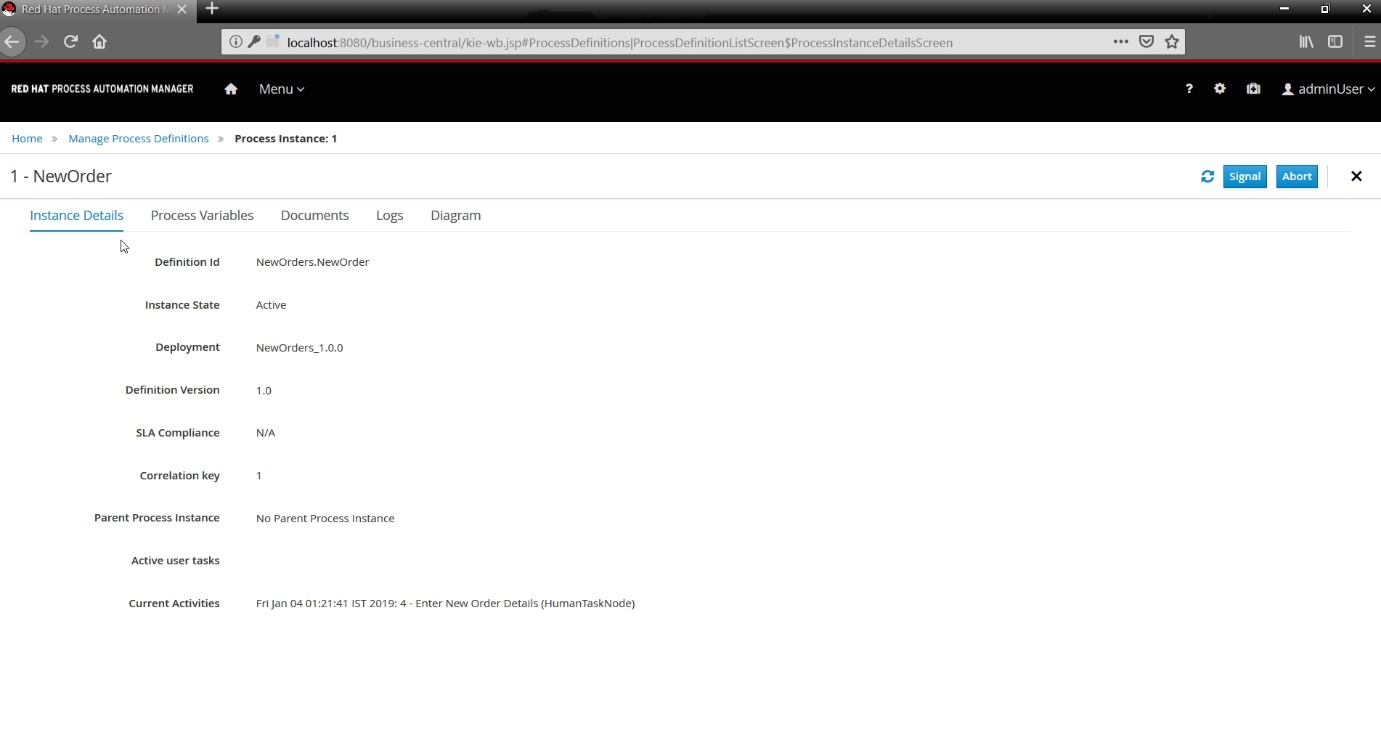
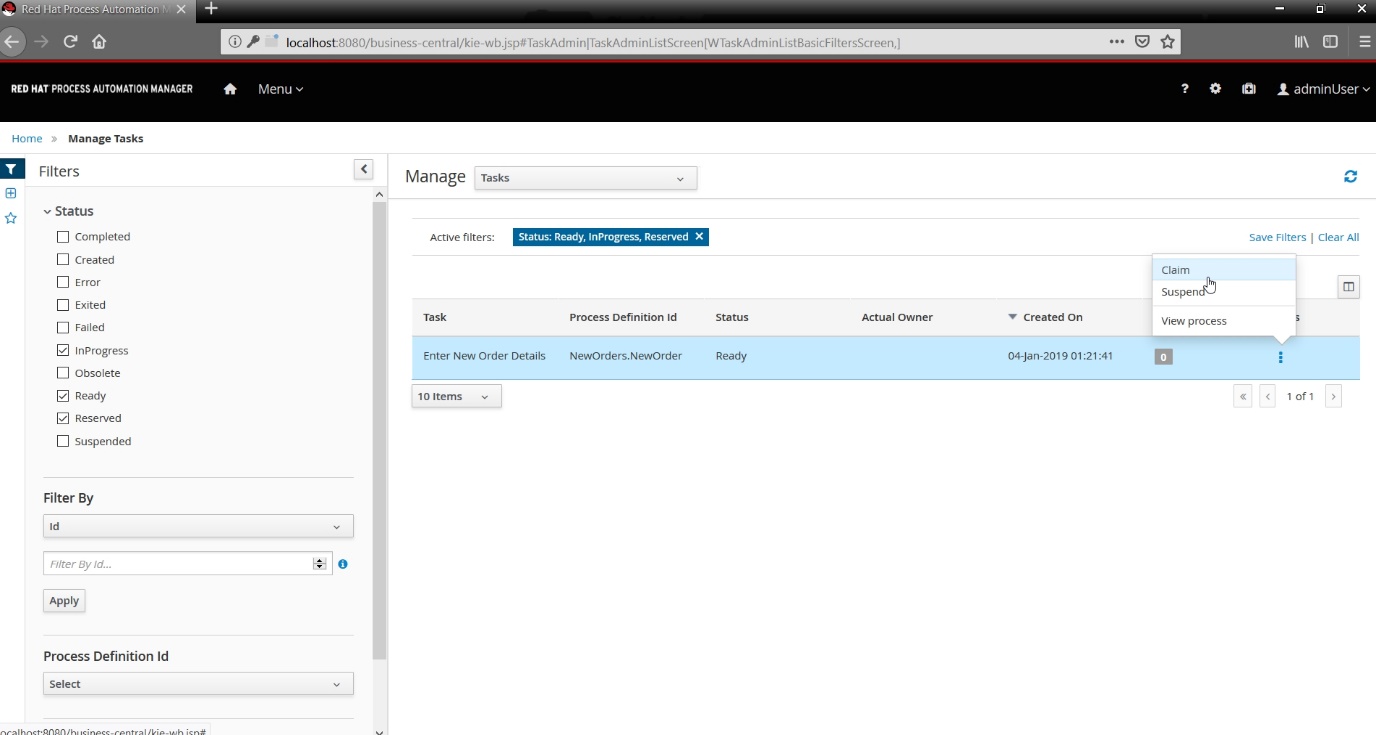
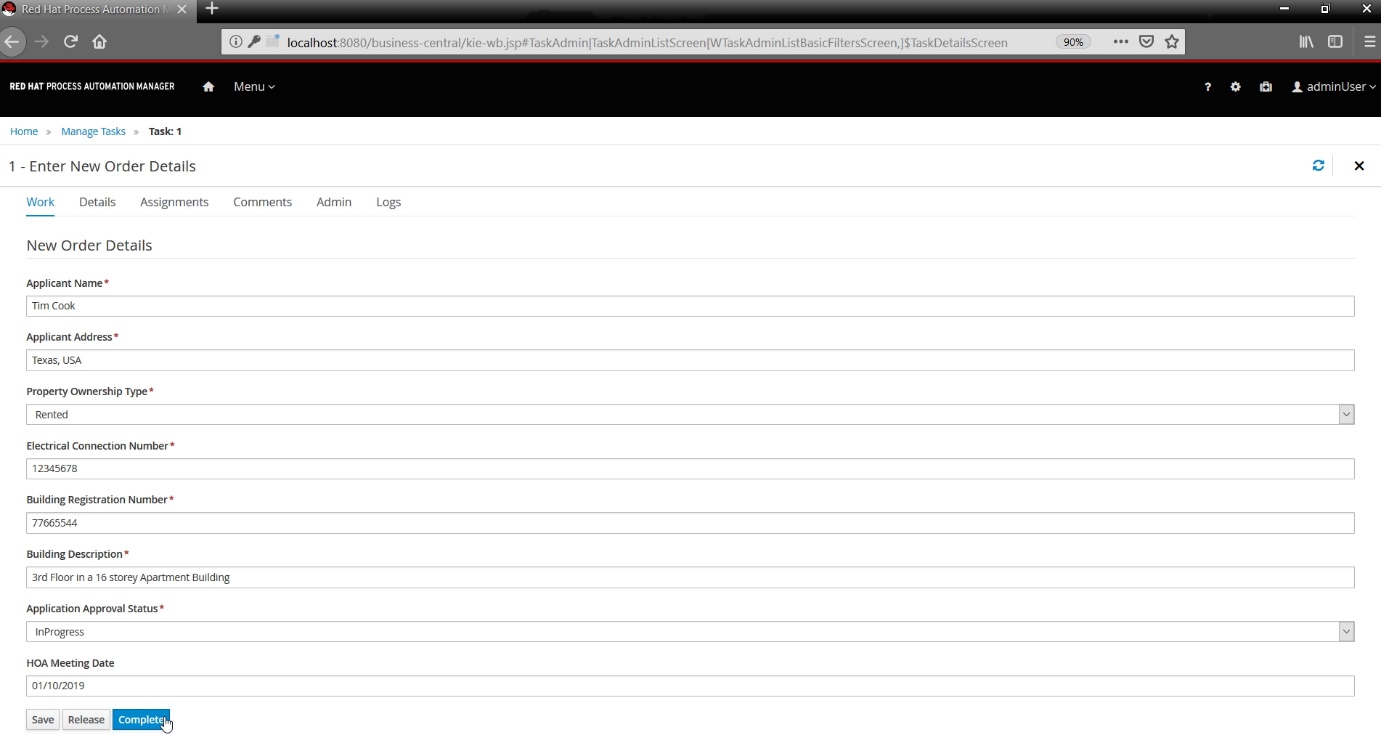
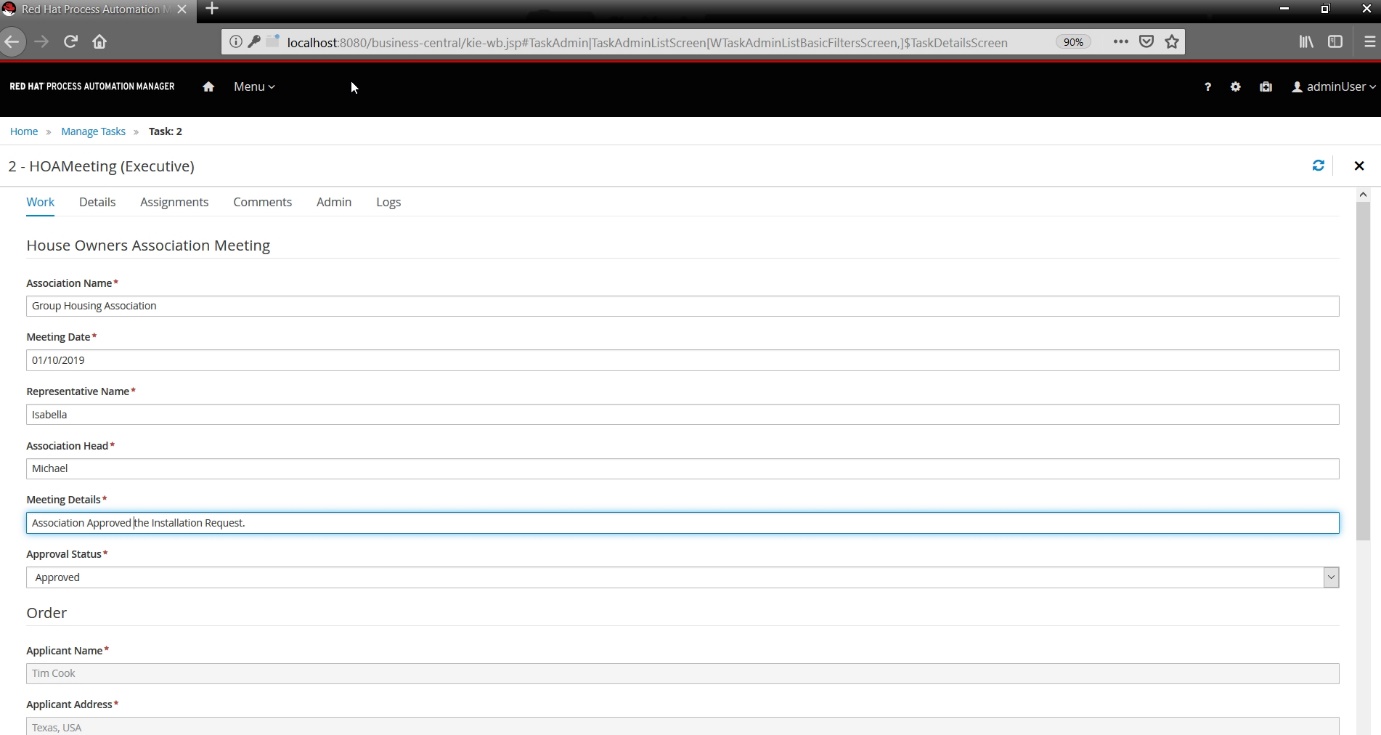
* Open PAM Central page (<http://localhost:8080/business-central/kie-wb.jsp>) and login with your credentials.
* In the PAM Central, navigate to ***Menu ->Design -> Projects.*** Click on ***Import project*** and provide the git url under ***Repository URL.***
* Select the ***NewOrders*** project and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

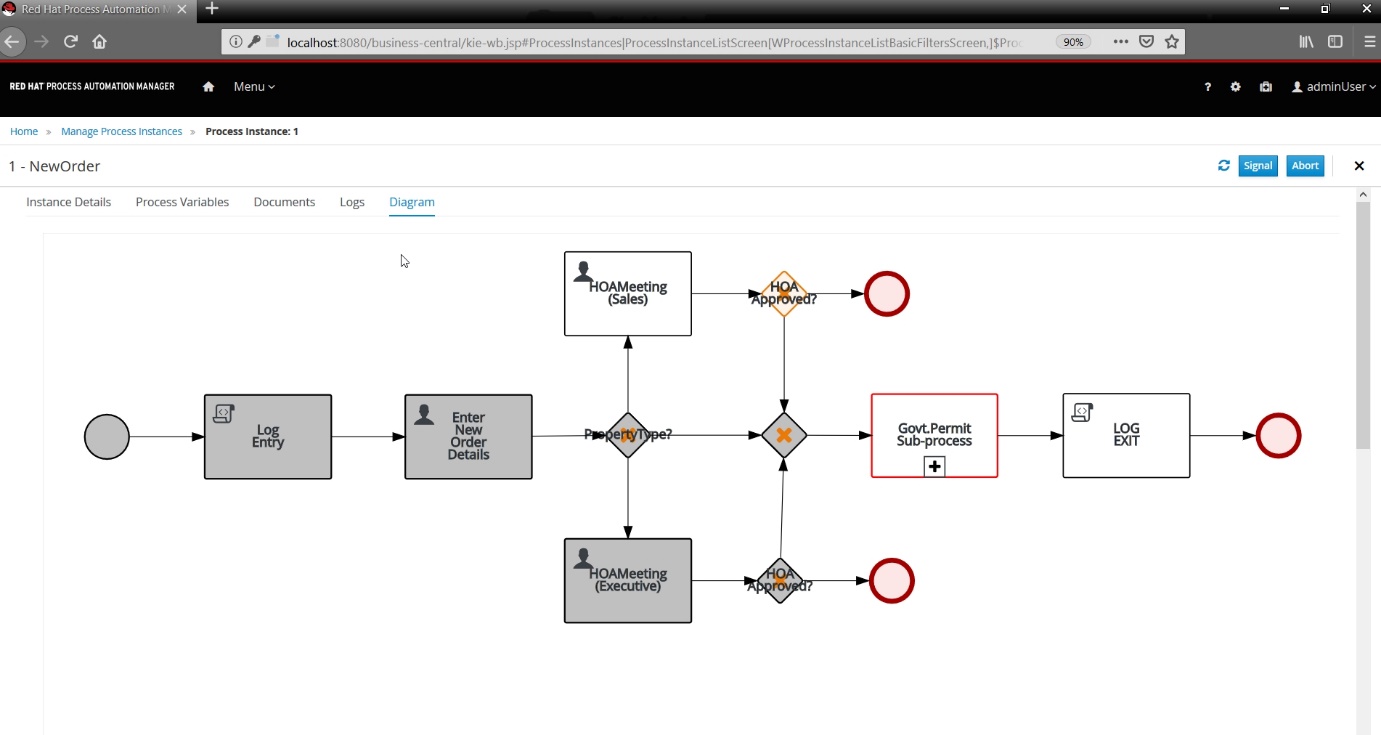


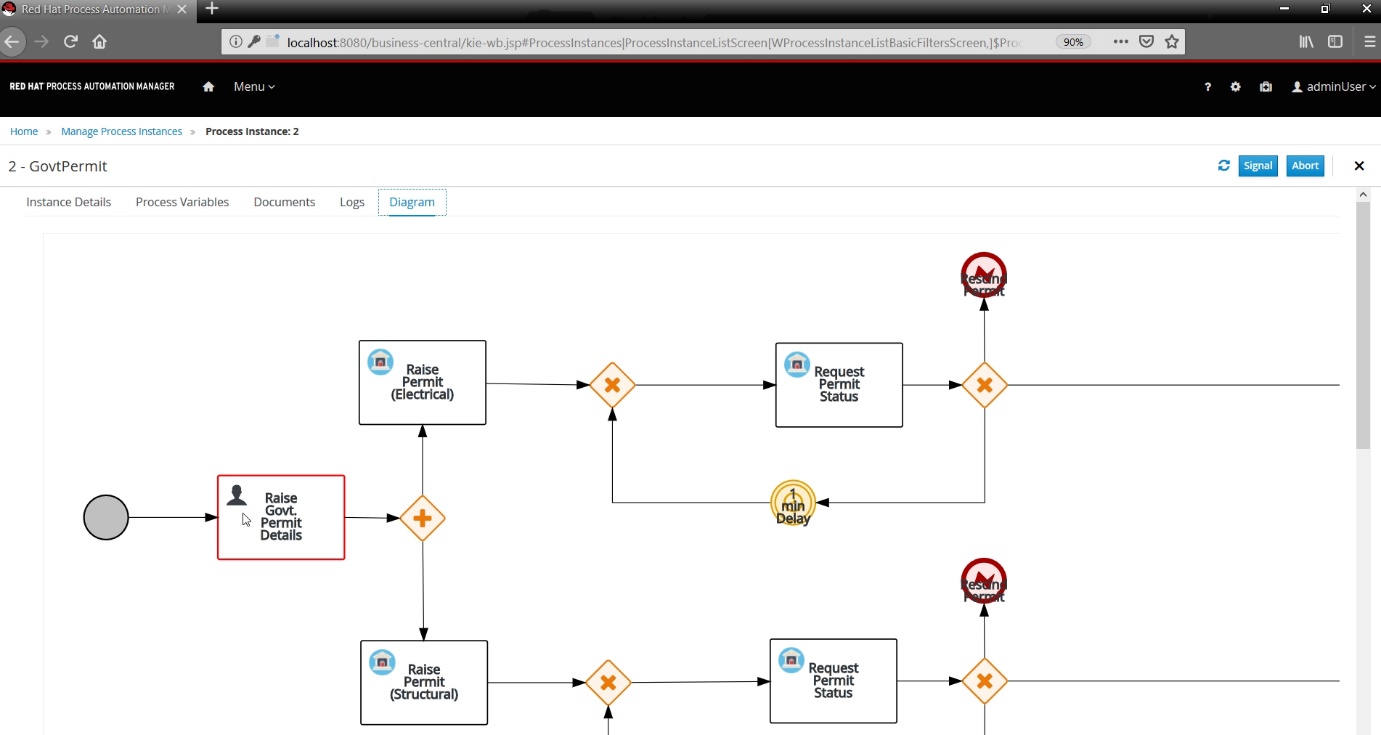
1. **Build and Deploy Project in PAM**

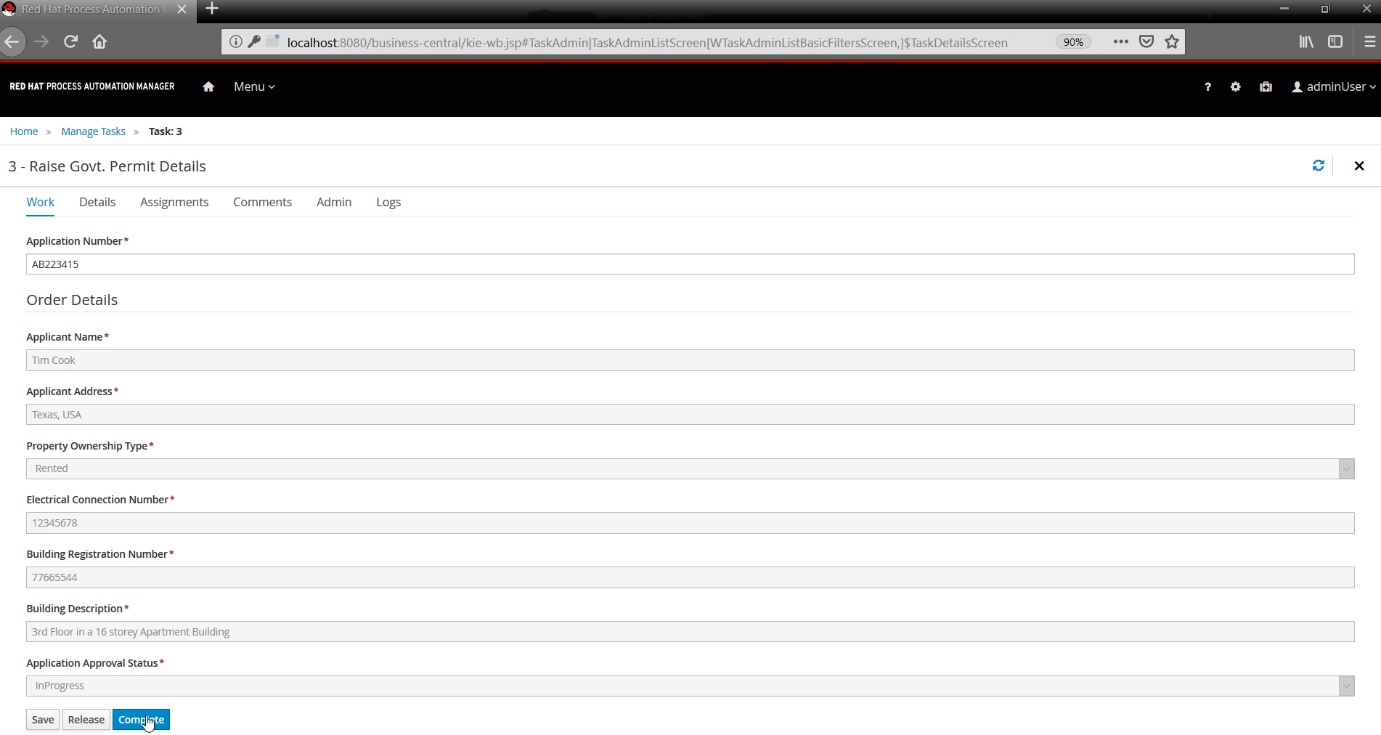
* Open the ***NewOrders*** project and click ***Deploy*** button present on top right.
* You should see a green message box saying Build and Deploy Successful.
* To verify deployment, navigate to ***Menu -> Deploy -> Execution Servers*** and you should be able to see a deployment unit named ***NewOrders\_1.0.0***.
* Also, we can see that process definitions are generated for NewOrder and GovtPermit processes under ***Menu -> Manage -> Process Definitions.***

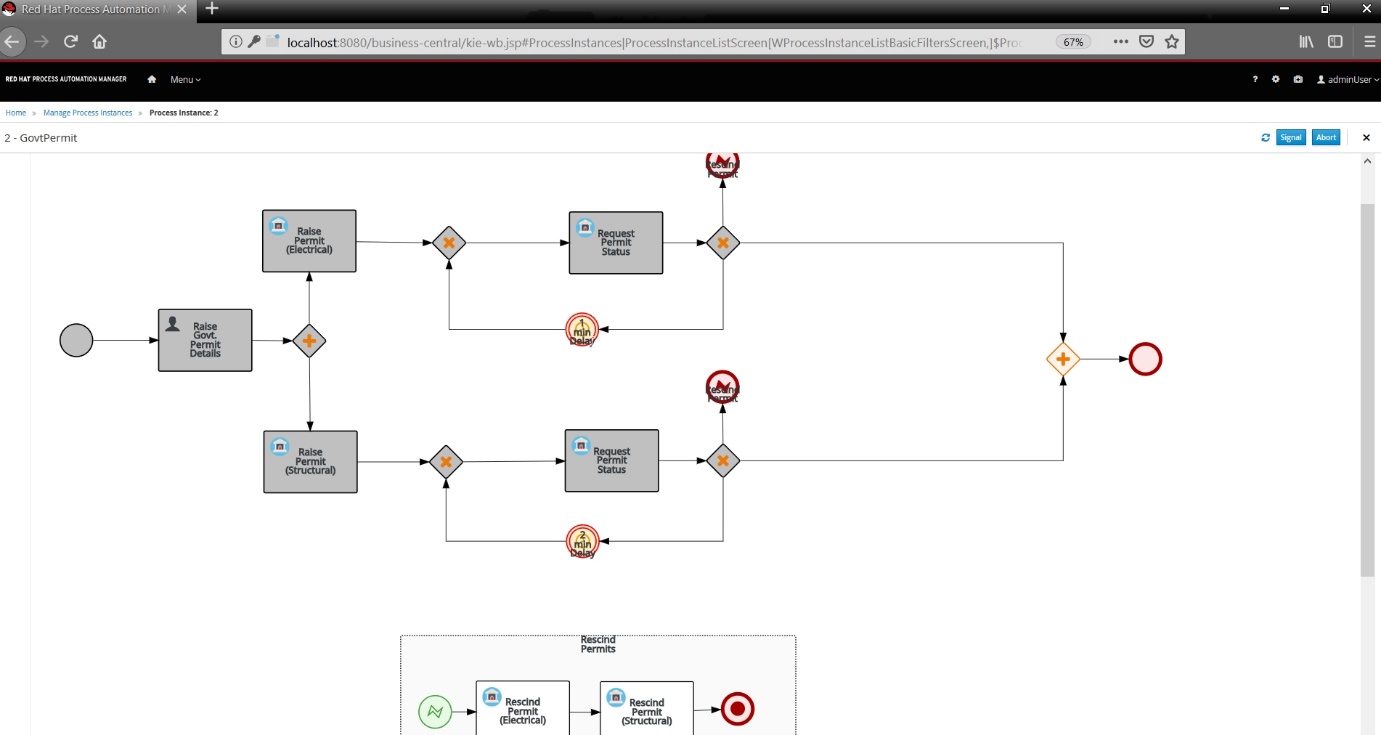
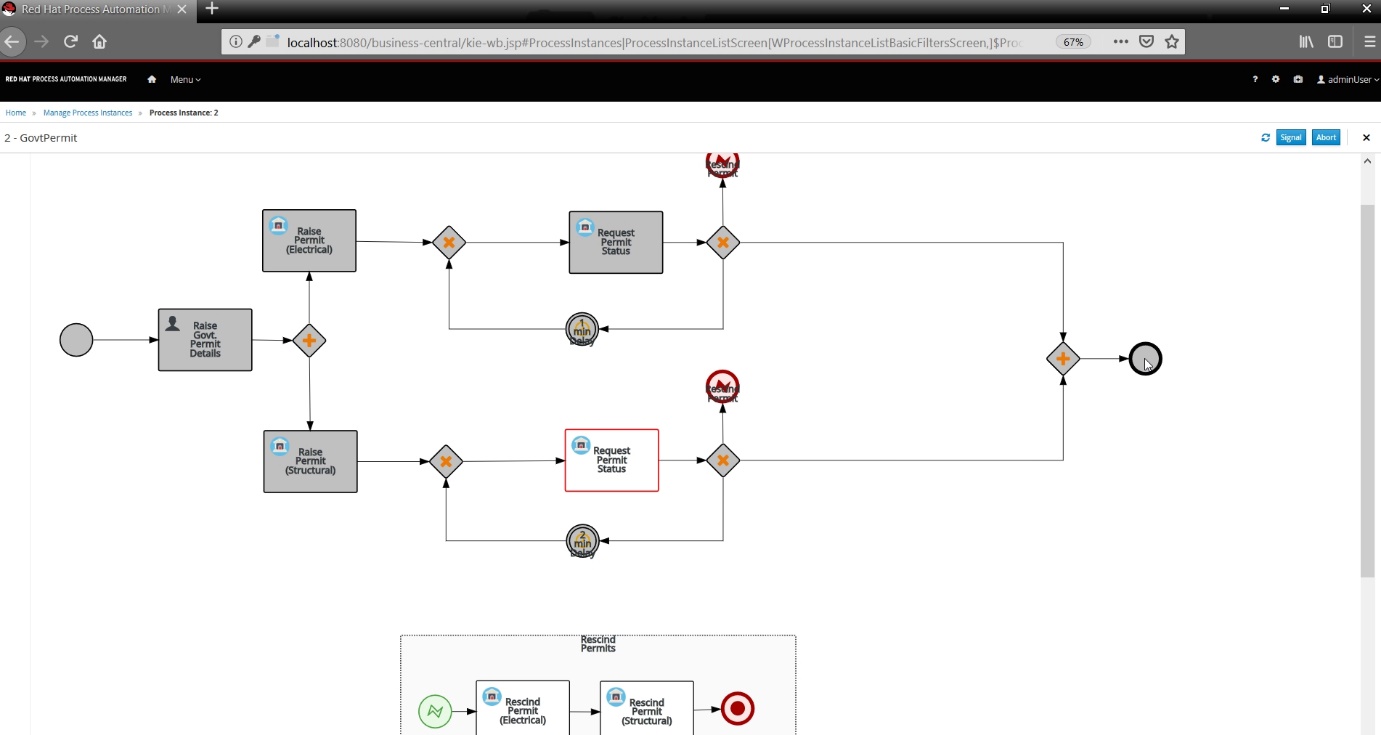
1. **Sample Execution**

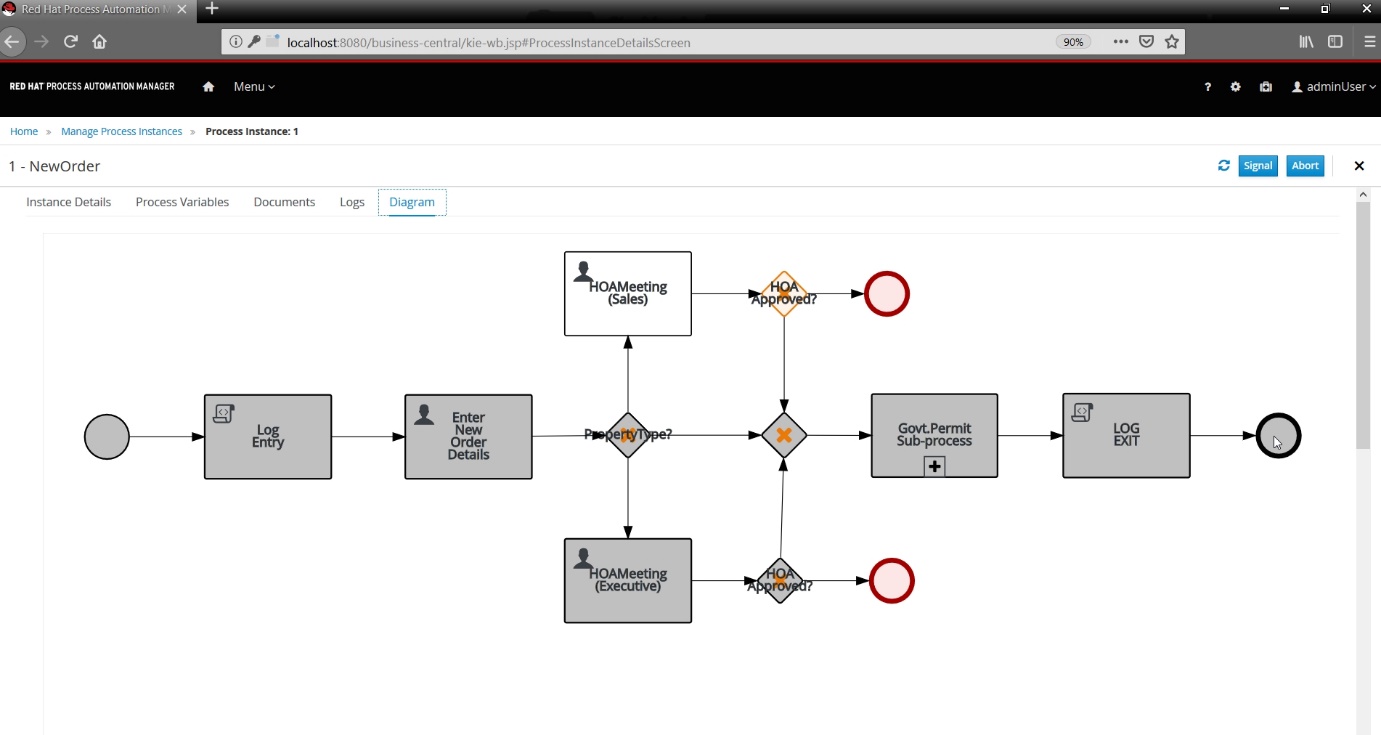
* Start a new process instance for the ***NewOrder*** process as shown below and click ***Submit***.
* Now, you can see a new process instance being created as shown below.
* Go to ***Menu -> Manage -> Tasks*** and ***claim*** the Enter New Order Details user task as shown below.
* Now open the task, click ***Start*** and fill the new order form and click ***Complete*** once done.
* Now claim the HOA Task and fill the HOA Meeting form.
* From the NewOrder process instance we can see where we are in the process flow by clicking on the ***Diagram*** option.

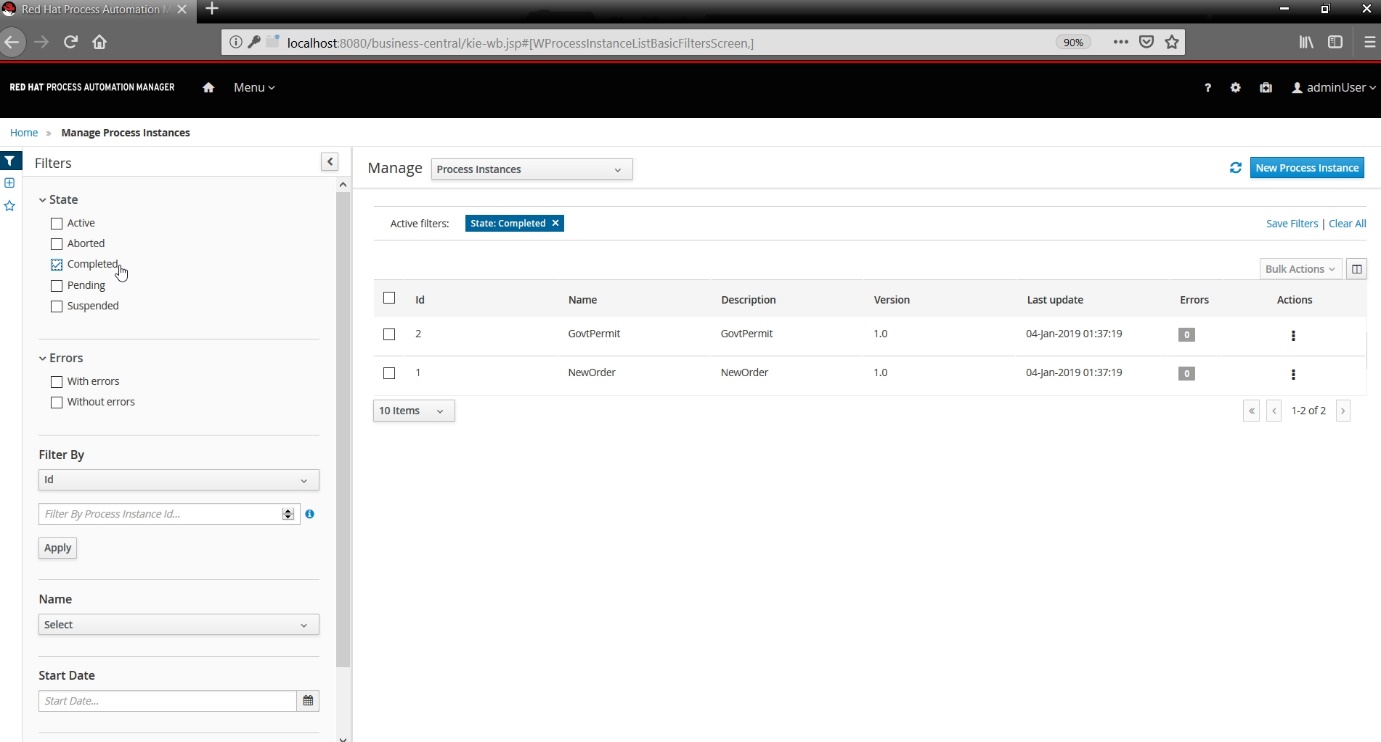


* After we reach the ***Govt.Permit SubProcess*** in the NewOrder flow, a new process instance for the GovtPermit is created.
* Claim the user task ***Raise Govt. Permit Details*** and complete it by giving an application number as shown.



* 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.

1. **Appendix**
2. **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=%7b$ApplicationNumber%7d&status=%7b$Status%7d)

1. **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=%7b$ApplicationNumber%7d&status=%7b$Status%7d)

1. **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>

1. **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>