Guided Exercise: Executing Rules Using KIE API

In this exercise, you will execute a Red Hat Decision Manager application by using KIE Server Java client API.

Outcomes

You should be able to use the KIE Server Java client API to execute a project deployed on the KIE Server.

In this exercise, you will use one of the projects you implemented previously. You will import the project into Red Hat Decision Manager and deploy it to the KIE Server.

Then, you will use the KIE Server Java client API to authenticate your API calls and execute the rules by using given parameters. After the Decision Manager application finishes, you will display the results.

To perform this exercise, ensure you have access to:

- A configured and running instance of Red Hat Decision Manager.
- Access to the Business Central web interface.

Ensure you are familiar with section called "Authoring Assets in Business Central" in the appendix. In this exercise, you will use the following sections of the appendix:

- Logging into Business Central
- Importing a project

You can explore the finished project at `/home/student/AD364/solutions/integrate-kie/integrate-kie-project`. (THE FILE THAT CONTAINS THIS DIRECTORY IS HERE)

Procedure 4.1. Instructions

1. Build and deploy the Decision Manager application.
    1. Log into Business Central at `http://localhost:8080/decision-central` as the `student` user with the `student1!` password.
    2. Import the starter project into the `AD364` space by using the following URL:

    ```
    /home/student/AD364/labs/integrate-kie/import
    ```

    3. Explore the **Laptop Recommendation Rules** Guided Decision Table, which contains rules that will fire based on the `OperatingSystem` and `TravelProfile` objects in the working memory.

    4. Build and deploy the project to the KIE Server.

2. Open the starter project:

```
[student@workstation ~]$ codium \
/home/student/AD364/labs/integrate-kie/integrate-kie-project
```

3. Review the application source code.

    1. Review the `pom.xml` file. Note the `kie-server-client` dependency, which contains the KIE Server Java client API.

    2. In the **Explorer** view, open and explore the `src/main/java/com/ad364/bc_tables` package. This package contains the data model classes used in the Decision Manager project.

    3. In the **Explorer** view, open the `src/main/java/com/ad364/integrate_kie/Main.java` class. The `Main` class contains a stub of the KIE Server Java client API project.

4. Locate the `TODO: Credentials` comment and set the class fields following the comment. Use the `kieserver` user with the `kieserver1!` password to authenticate against the Decision Manager API.

    The `kieserver` user is already present in your Decision Manager deployment and contains permissions to use the KIE API only.

    Use the `integrate-kie` container alias.

    1. Change the `USERNAME` value to `kieserver`.

    2. Change the `PASSWORD` value to `kieserver1!`.

    3. Change the `CONTAINER_ID` value to `integrate-kie`.

5. Locate the `TODO: Configure the KieServicesConfig object` comment and implement the `KieServiceConfig` configuration.

    1. Use the credentials to instantiate a new `CredentialsProvider`:

```
CredentialsProvider credentialsProvider= new EnteredCredentialsProvider(USERNAME, PASSWORD);
```

   2. Use the `credentialsProvider` and `KIE_SERVER_URL` objects to instantiate a new `KieServicesConfiguration`:

```
KieServicesConfiguration kieServicesConfig = KieServicesFactory.newRestConfiguration(KIE_SERVER_URL, credentialsProvider);
```

   3. Set the `kieServicesConfig` marshalling format to JSON:

      ```
      kieServicesConfig.setMarshallingFormat(MarshallingFormat.JSON);
      ```

6. Locate the `TODO: Create a RuleServicesClient client` comment and use the `kieServicesClient` object to instantiate a new `RuleServicesClient` object.:

```
KieServicesClient kieServicesClient = KieServicesFactory.newKieServicesClient(kieServicesConfig);

RuleServicesClient rulesClient = kieServicesClient.getServicesClient(RuleServicesClient.class);
```

7. Locate the `TODO commands` comment and prepare commands that will execute in the Decision Manager container. Add each of the commands into the `commands` list below the `TODO commands` comment section.

   1. Use the `getOperatingSystem` and `getTravelProfile` methods to instantiate the `OperatingSystem` and `TravelProfile` objects. Then, create an `insert` command to insert the objects into the rule container. For example:

```
commands.add(commandFactory.newInsert(getOperatingSystem("macOS")));

commands.add(commandFactory.newInsert(getTravelProfile(35, false)));
```

   2. Evaluate all rules by using the `fireAllRules` command:

```
commands.add(commandFactory.newFireAllRules());
```

   3. Inspect all objects in the working memory by using the `getObjects` command:

```
commands.add(commandFactory.newGetObjects("objects"));
```

4. Dispose of all the objects in the working memory by using the dispose command:

```
commands.add(commandFactory.newDispose());
```

8. Execute the project:

```
[student@workstation ~]$ cd /home/student/AD364/labs/integrate-kie/integrate-kie-project

[student@workstation integrate-kie-project]$ mvn clean package exec:java
```

In the output, you should see:

```
[OperatingSystem{osName='macOS'}, TravelProfile{travelPercent=35, noTravel=fa
lse}, Recommendation{make='Apple', model='Macbook Air 13 inch'}]
```

Input other values of travel profile and operating system preferences to explore other rules fire as expected.

Guided Exercise: Executing Projects Using Rest API

In this exercise you will execute an application using the Red Hat Decision Manager REST API.

Outcomes

You should be able to use the RHDM REST API to execute a project deployed on the KIE Server.

You will use a project implemented in a previous exercise. You will import the project into Red Hat Decision Manager and deploy it to the KIE Server.

To perform this exercise, ensure you have access to:

- A configured and running instance of Red Hat Decision Manager.
- Access to the Business Central web interface.

Ensure you are familiar with the section called “Authoring Assets in Business Central” in the appendix. In this exercise, you will use the following sections of the appendix:

- Logging into Business Central
- Importing a project

Procedure 4.2. Instructions

1. Build and deploy the Decision Manager application.

    1. Log into Business Central at `http://localhost:8080/decision-central` as the `student` user with the `student1!` password.

    2. Import the starter project into the `AD364` space using the following URL:

        ```
        /home/student/AD364/labs/integrate-rest
        ```

    3. Explore the **Laptop Recommendation Rules** Guided Decision Table. This table contains rules that will fire based on the `OperatingSystem` and `TravelProfile` objects in the working memory.

    4. Build and deploy the project to the KIE Server. Click **View deployment details** and verify that a container named `integrate-rest_1.0.0-SNAPSHOT` is deployed successfully.

2. Open the KIE Server Swagger web interface to explore the REST API.

    1. Navigate to `localhost:8080/kie-server/docs` in the web browser to open the Swagger web interface.

    2. Expand the `KIE Server and KIE containers` panel, and click **GET** next to the **/server/containers** entry.

    3. Click **Try it out** and then click **Execute**.

        You will be prompted for credentials to access the KIE server. Enter `kieserver` as the username, and `kieserver1!` as the password.

        You will see the following response in the `Responses` section:

        ```
        {
          "type": "SUCCESS",
          "msg": "List of created containers",
          "result": {
            "kie-containers": {
              "kie-container": [
        ...output omitted...
        ```

```
{
          "container-id": "integrate-rest_1.0.0-SNAPSHOT",
          "release-id": {
            "group-id": "com.ad364",
            "artifact-id": "integrate-rest",
            "version": "1.0.0-SNAPSHOT"
...output omitted...

"status": "STARTED",


...output omitted...
```

NOTE

Your output can differ based on other projects and containers that you have deployed on the KIE server.

3. Execute commands on the KIE Server.

    1. Click **POST** in the **KIE session assets** section.

    2. Click **Try it out**.

        Enter `integrate-rest_1.0.0-SNAPSHOT` in the **containerId** field in the `Parameters` section.

        Select `application/json` in the `Parameter content type` field.

    3. Copy the JSON snippet from the `/home/student/AD364/labs/integrate-rest/input1.json` file to the `body` field.

    4. Select `application/json` in the `Response content type` field, and click **Execute**.

    5. If you are prompted for credentials to access the KIE server, enter `kieserver` as the username, and `kieserver1!` as the password.

    6. You should see the following output in the `Responses` section:

```
{
  "type": "SUCCESS",
  "msg": "Container integrate-rest_1.0.0-SNAPSHOT successfully cal
led.",
  "result": {
    "execution-results": {
      "results": [
        {
...output omitted...
            {
              "com.ad364.bc_tables.Recommendation": {
                "make": "Lenovo",
                "model": "Lenovo Thinkcenter M90 Desktop"
...output omitted...
```

4. Execute the container with a different set of input data to trigger other rules.

    1. Replace the JSON content in the `body` section with content from the `/home/student/AD364/labs/integrate-rest/input2.json` file.

    2. Click **Execute**. You should see the following response:

```
...output omitted...
            {
              "com.ad364.bc_tables.Recommendation": {
                "make": "Lenovo",
                "model": "Thinkpad Carbon X1 8th Gen"
              }
            }
...output omitted...
```