Red Hat Process Automation Manager 7.5

Getting started with decision services

Red Hat Process Automation Manager 7.5 Getting started with decision services

Red Hat Customer Content Services
brms-docs@redhat.com

Legal Notice

Abstract

This document describes how to create and test an example traffic violation decision service using a Decision Model and Notation (DMN) model in Red Hat Process Automation Manager 7.5. The procedures in this document are based on the Traffic_Violation sample project included in Business Central.

Table of Contents

PREFACE

As a business rules developer, you can use Business Central in Red Hat Process Automation Manager to design a variety of decision services. Red Hat Process Automation Manager provides example projects with example business assets directly in Business Central as a reference. This document describes how to create and test an example traffic violation project based on the **Traffic_Violation** sample project included in Business Central. This sample project uses a Decision Model and Notation (DMN) model to define driver penalty and suspension rules in a traffic violation decision service. You can follow the steps in this document to create the project and the assets it contains, or open and review the existing **Traffic_Violation** sample project.

For more information about the DMN components and implementation in Red Hat Process Automation Manager, see *Designing a decision service using DMN models* .

Prerequisites

- Red Hat JBoss Enterprise Application Platform 7.2 is installed. For installation information, see *Red Hat JBoss Enterprise Application Platform 7.2 Installation Guide* .

- Red Hat Process Automation Manager is installed and configured with Process Server. For more information, see *Installing and configuring Red Hat Process Automation Manager on Red Hat JBoss EAP 7.2*.

- Red Hat Process Automation Manager is running and you can log in to Business Central with the **developer** role. For more information, see *Planning a Red Hat Process Automation Manager installation*.

CHAPTER 1. EXAMPLE PROJECTS AND BUSINESS ASSETS IN BUSINESS CENTRAL

Business Central contains example projects with example business assets that you can use as a reference for the rules, processes, or other assets that you create in your own Red Hat Process Automation Manager projects. Each sample project is designed differently to demonstrate process automation, decision management, or business optimization assets and logic in Red Hat Process Automation Manager.

> **NOTE**
>
> Red Hat does not provide support for the example code included in the Red Hat Process Automation Manager distribution.

The following example projects are available in Business Central:

- **Evaluation_Process**: (Process automation) Example evaluation process using business process assets. Evaluates employees based on performance.

- **Mortgage_Process**: (Process automation) Example loan approval process using business process and decision assets. Determines loan eligibility based on applicant data and qualifications.

- **IT_Orders**: (Process automation and case management) Example ordering case using business process and case management assets. Places an IT hardware order based on needs and approvals.

- **Traffic_Violation**: (Decision management with DMN) Example traffic violation decision service using a Decision Model and Notation (DMN) model. Determines driver penalty and suspension based on traffic violations.

- **Mortgages**: (Decision management with rules) Example loan approval process using rule-based decision assets. Determines loan eligibility based on applicant data and qualifications.

- **Employee_Rostering**: (Business optimization) Example employee rostering optimization using decision and solver assets. Assigns employees to shifts based on skills.

- **OptaCloud**: (Business optimization) Example resource allocation optimization using decision and solver assets. Assigns processes to computers with limited resources.

- **Course_Scheduling**: (Business optimization) Example course scheduling and curriculum decision process. Assigns lectures to rooms and determines a student's curriculum based on factors, such as course conflicts and class room capacity.

- **Dinner_Party**: (Business optimization) Guest seating optimization using guided decision tables. Assigns guest seating based on each guest's job type, political beliefs, and known relationships.

1.1. ACCESSING EXAMPLE PROJECTS AND BUSINESS ASSETS IN BUSINESS CENTRAL

You can use the example projects in Business Central to explore example business assets as a reference for the rules or other assets that you create in your own Red Hat Process Automation Manager projects.

Prerequisites

- Business Central is installed and running. For installation options, see *Planning a Red Hat Process Automation Manager installation*.

Procedure

1. In Business Central, go to **Menu** → **Design** → **Projects** and click **Try Samples**.
   If a project already exists, click the three vertical dots in the upper-right corner of the **Projects** page and click **Try Samples**.

2. Review the descriptions for each sample project to determine which project you want to explore. Each sample project is designed differently to demonstrate process automation, decision management, or business optimization assets and logic in Red Hat Process Automation Manager.

3. Select one or more sample projects and click **Ok** to add the projects to your space.

4. In the **Projects** page of your space, select one of the new example projects to view the example assets for that project.

5. Select each example asset to explore how the project is designed to achieve the specified goal or workflow.

6. In the upper-right corner of the project **Assets** page, click **Build** to build the sample project or **Deploy** to build the project and then deploy it to Process Server.

   > **NOTE**
   >
   > You can also select the **Build & Install** option to build the project and publish the KJAR file to the configured Maven repository without deploying to a Process Server. In a development environment, you can click **Deploy** to deploy the built KJAR file to a Process Server without stopping any running instances (if applicable), or click **Redeploy** to deploy the built KJAR file and replace all instances. The next time you deploy or redeploy the built KJAR, the previous deployment unit (KIE container) is automatically updated in the same target Process Server. In a production environment, the **Redeploy** option is disabled and you can click **Deploy** only to deploy the built KJAR file to a new deployment unit (KIE container) on a Process Server.
   >
   > To configure the Process Server environment mode, set the **org.kie.server.mode** system property to **org.kie.server.mode=development** or **org.kie.server.mode=production**. To configure the deployment behavior for a corresponding project in Business Central, go to project **Settings** → **General Settings** → **Version** and toggle the **Development Mode** option. By default, Process Server and all new projects in Business Central are in development mode. You cannot deploy a project with **Development Mode** turned on or with a manually added **SNAPSHOT** version suffix to a Process Server that is in production mode.

   To review project deployment details (if applicable), go to **Menu** → **Deploy** → **Execution Servers**.

CHAPTER 2. CREATING THE TRAFFIC VIOLATIONS PROJECT

For this example, create a new project called **traffic-violation**. A project is a container for assets such as data objects, DMN assets, and test scenarios. This example project that you are creating is similar to the existing **Traffic_Violation** sample project in Business Central.

Procedure
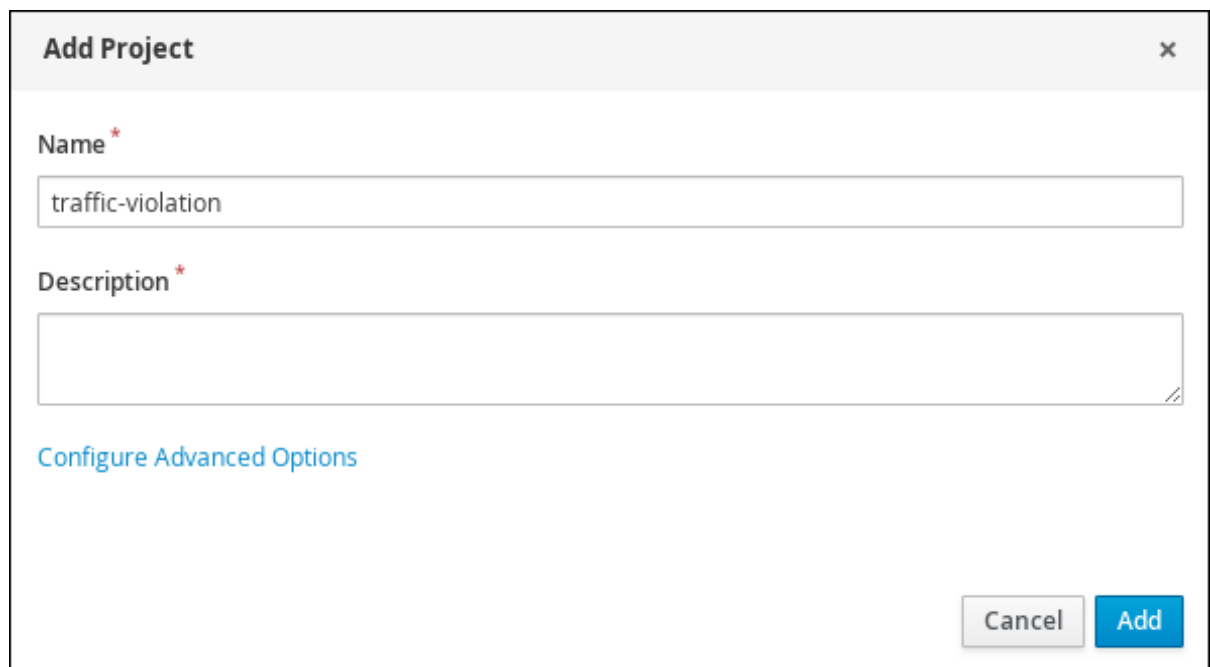
1. Log in to Business Central.

2. Go to **Menu → Design → Projects**.
   Red Hat Process Automation Manager provides a default space called **MySpace**, as shown in the following image. You can use the default space to create and test example projects.

   Figure 2.1. Default space

   Spaces » **MySpace**

3. Click **Add Project**.

4. Enter **traffic-violation** in the Name field.

5. Click **Add**.

   Figure 2.2. Add Project window

   Add Project ✕

   Name*
   traffic-violation

   Description*

   Configure Advanced Options

   Cancel   Add

   The **Assets** view of the project opens.

CHAPTER 3. DECISION MODEL AND NOTATION (DMN)

Decision Model and Notation (DMN) is a standard established by the Object Management Group (OMG) for describing and modeling operational decisions. DMN defines an XML schema that enables DMN models to be shared between DMN-compliant platforms and across organizations so that business analysts and business rules developers can collaborate in designing and implementing DMN decision services. The DMN standard is similar to and can be used together with the Business Process Model and Notation (BPMN) standard for designing and modeling business processes.

For more information about the background and applications of DMN, see the OMG Decision Model and Notation specification.

3.1. CREATING THE TRAFFIC VIOLATIONS DMN DECISION REQUIREMENTS DIAGRAM (DRD)

A decision requirements diagram (DRD) is a visual representation of your DMN model. Use the DMN designer in Business Central to design the DRD for the traffic violations project and to define the decision logic of the DRD components.

Figure 3.1. DRD for the Traffic Violations example
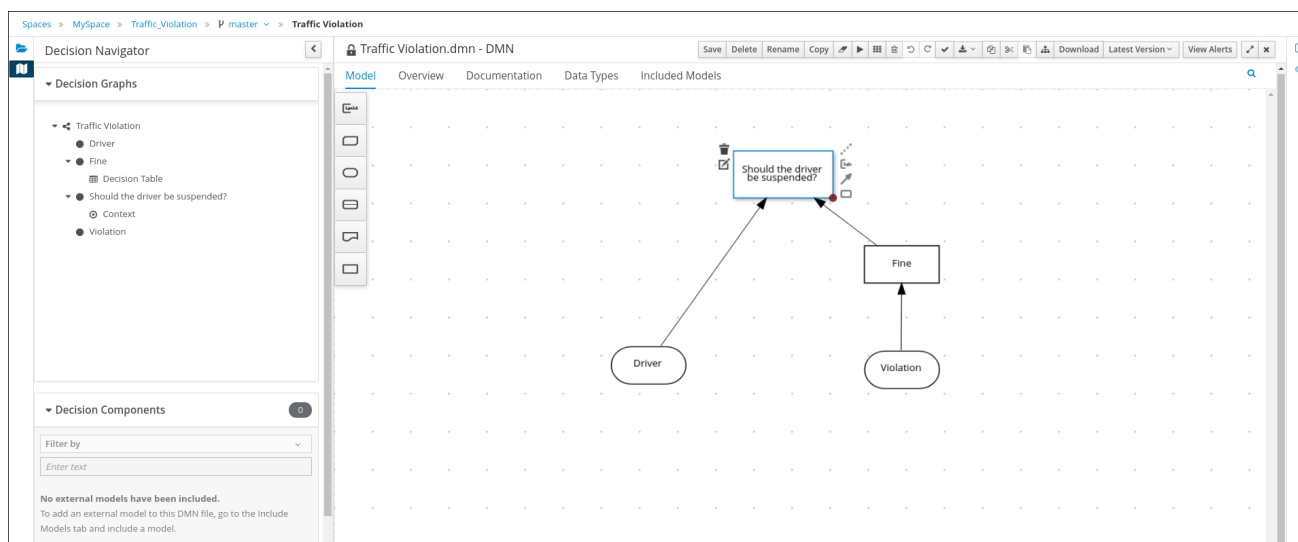
Prerequisites

- You have created the traffic violations project in Business Central.

Procedure

1. On the **traffic-violation** project's home page, click **Add Asset**.

2. On the **Add Asset** page, click **DMN**. The **Create new DMN** window is opened.

   a. In the **Create new DMN** window, enter **Traffic Violation** in the **DMN** name field.

   b. From the **Package** list, select **com.myspace.traffic_violation**.

   c. Click **Ok**. The DMN asset in the DMN designer is opened.

3. In the DMN designer canvas, drag two **DMN Input Data** input nodes on to the canvas.

4. Double-click each input node and rename one to **Driver** and the other to **Violation**.

5. Drag a **DMN Decision** decision node on to the canvas.

6. Double-click the decision node and rename it to **Fine**.

7. Click the **Violation** input node, select the **Create DMN Information Requirement** icon and click the **Fine** decision node to link the two nodes.

8. Drag a **DMN Decision** decision node on to the canvas.

9. Double-click the decision node and rename it to **Should the driver be suspended?**.

10. Click the **Driver** input node, select the **Create DMN Information Requirement** icon and click the **Should the driver be suspended?** decision node to link the two nodes.

11. Click the **Fine** decision node, select the **Create DMN Information Requirement** icon, and select the **Should the driver be suspended?** decision node.

12. Click **Save** and click **Save** again to confirm.

3.2. CREATING THE TRAFFIC VIOLATIONS DMN CUSTOM DATA TYPES

DMN data types determine the structure of the data that you use within a table, column, or field in a DMN boxed expression for defining decision logic. You can use default DMN data types (such as string, number, or boolean) or you can create custom data types to specify additional fields and constraints that you want to implement for the boxed expression values. Use the DMN designer's **Data Types** tab in Business Central to define the custom data types for the traffic violations project.

Figure 3.2. The custom data types tab
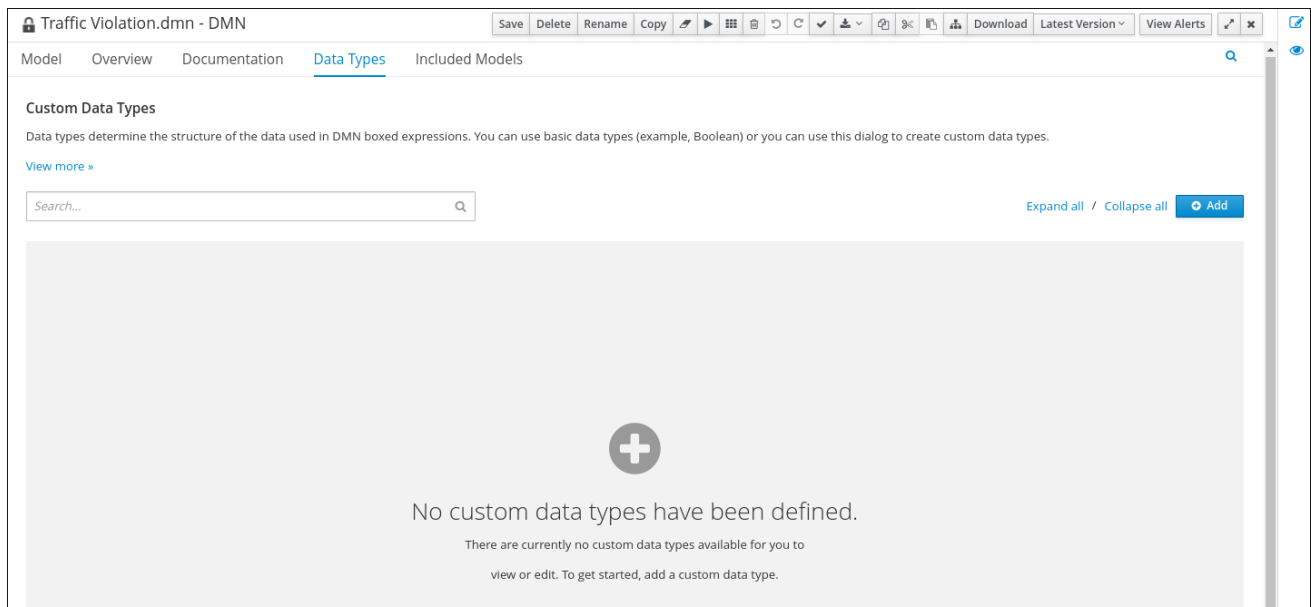
Alternately, you can also access the **Data Types** tab from the **Diagram properties** tab on the right. From the **Diagram properties** tab, click **Manage** from **Data type** under **Information item**.

Figure 3.3. Diagram properties tab

The following tables list the **Violation**, **Driver**, and **Fine** custom data types that you will create for this project.

Table 3.1. Driver custom data types

Name	Type
tDriver	Structure

Name	Type
Name	string
Age	number
State	string
City	string
Points	number

Table 3.2. Violation custom data types

Name	Type
tViolation	Structure
Code	string
Date	date
Type	string
Speed Limit	number
Actual Speed	number

Table 3.3. Fine custom data types

Name	Type
tFine	Structure
Amount	number
Points	number

Prerequisites

- You created the traffic violations DMN decision requirements diagram (DRDs) in Business Central.

Procedure

1. To create the **tDriver** custom data type, click **Add** on the **Data Types** tab, enter **tDriver** in the **Name** field, select **Structure** from the **Type** list, and click **Save**.

Figure 3.4. The tDriver custom data type
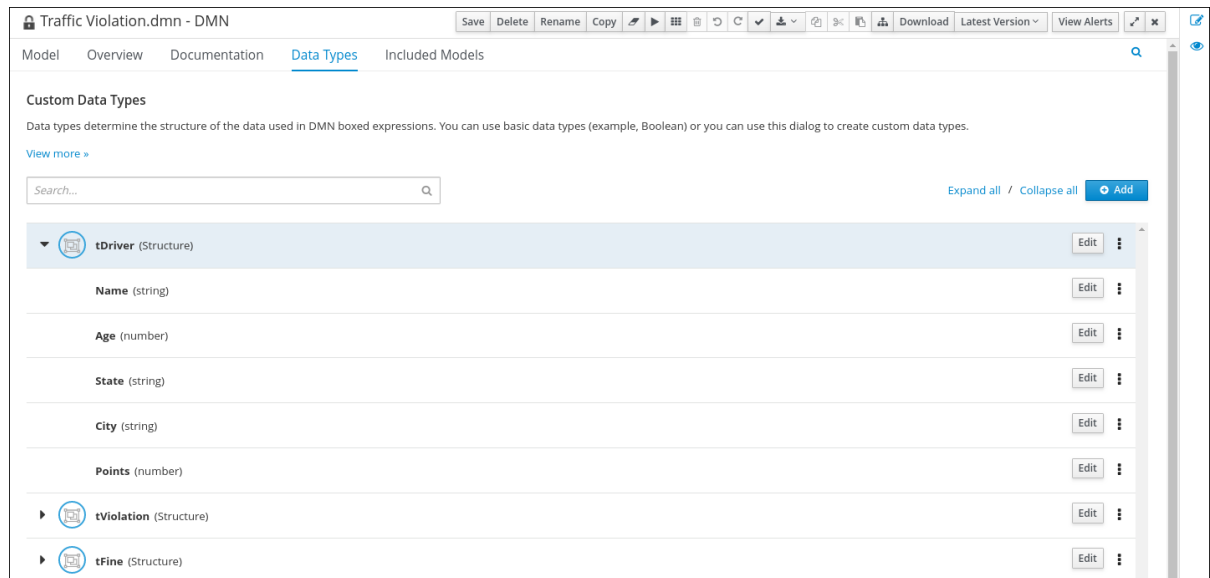
2. To create the first nested data type, enter **Name** in the **Name** field of the nested data type, select **String** from the **Type** list and click **Save**.

3. For each of the remaining nested data types, click the three vertical dots next to the **tDriver** data type, select **Insert nested field**, and add the following nested data types. Click **Save** for each nested data type you add.

   - **Name** (string)

   - **Age** (number)

   - **State** (string)

   - **City** (string)

   - **Points** (number)

4. To create the **tViolation** custom data type, click **Add** on the **Data Types** tab, enter **tViolation** in the **Name** field, select **Structure** from the **Type** list, and click **Save**.

Figure 3.5. The tViolation custom data type

5. To create the first nested data type, enter **Code** in the **Name** field of the nested data type, select **String** from the **Type** list and click **Save**.

6. For each of the remaining nested data types, click the three vertical dots next to the **tViolation** data type, select **Insert nested field**, and add the following nested data types. Click **Save** for each nested data type that you add.

   - **Code** (string)

- **Date** (date)

- **Type** (string)

- **Speed Limit** (number)

- **Actual Speed** (number)

7. To add the following constraints to the **Type** nested data type, click **Edit → Constraints →
   Enumeration** and add the following constraints. Click the check icon ![check] to save each
   enumeration constraint that you add.

   - **"speed"**

   - **"parking"**

   - **"driving under the influence"**

8. To create the **tFine** custom data type, click **Add** on the **Data Types** tab, enter **tFine** in the
   Name field, select **Structure** from the **Type** list, and click **Save**.

   Figure 3.6. The tFine custom data type
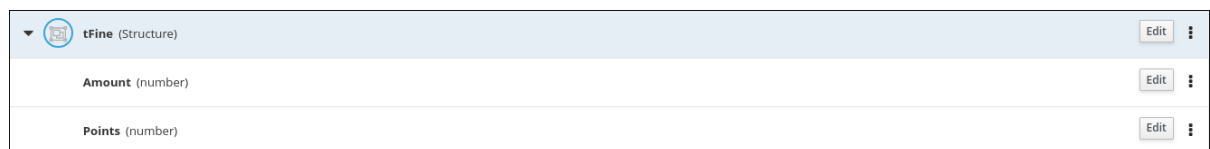
9. To create the first nested data type, enter **Amount** in the **Name** field of the nested data type,
   select **Number** from the **Type** list and click **Save**.

10. For each of the remaining nested data types, click the three vertical dots next to the **tFine** data
    type, select **Insert nested field**, and add the following nested data types. Click **Save** for each
    nested data type that you add.

    - **Amount** (number)

    - **Points** (number)

11. Once all the three custom data types are created, click **Save** to open the **Confirm Save** dialog
    box and click **Save** again.

3.3. ASSIGNING CUSTOM DATA TYPES TO THE DRD INPUT AND DECISION NODES

After you create the DMN custom data types, assign them to the appropriate **DMN Input Data** and
DMN Decision nodes in the traffic violations DRD.

Prerequisites

- You have created the traffic violations DMN custom data types in Business Central.

Procedure

1. Click the **Model** tab on the DMN designer and click **Diagram properties** in the top-right corner of the DMN designer to expose the DRD properties.

2. In the DRD, select the **Driver** input data node and in the **Diagram properties** panel, set **Data type** to **tDriver**.

3. Select the **Violation** input data node and set the **Data type** to **tViolation**.

4. Select the **Fine** input data node and set the **Data type** to **tFine**.

5. Select the **Should the driver be suspended?** decision node and set the following properties:

   - Data type: **string**

   - Question: **Should the driver be suspended due to points on his driver license?**

   - Allowed Answers: **"Yes","No"**

6. Click **Save** to open the **Confirm Save** dialog box and click **Save** again.

You have assigned the custom data types to your DRD's input and decision nodes.

3.4. DEFINING THE TRAFFIC VIOLATIONS DMN DECISION LOGIC

To calculate the fine and to decide whether the driver is to be suspended or not, you can define the traffic violations DMN decision logic using a DMN decision table and context boxed expression.

Figure 3.7. Fine decision table

Figure 3.8. Should the driver be suspended?

Prerequisites

- You have assigned the DMN custom data types to the appropriate decision and input nodes in the traffic violations DRD in Business Central.

Procedure

1. To calculate fine, in the DMN designer canvas, select the **Fine** decision node and click the **Edit** icon to open the DMN boxed expression designer.

2. Click **Select expression → Decision Table**.

3. For the **Violation.Date**, **Violation.Code**, and **Violation.Speed Limit** parameter fields, right-click and select **Delete** for each field.

4. Click the **Violation.Actual Speed** column header and enter the expression **Violation.Actual Speed - Violation.Speed Limit** in the **Name** field.

5. Right-click the **Fine** parameter field and select either **Insert left** or **Insert right**.

6. Click the **output-2** column sub-header, enter **Amount** in the **Name** field, and select **number** from the **Data Type** field.

7. Similarly, click the **output-1** column sub-header, enter **Points** in the **Name** field, and select **number** from the **Data Type** field.

8. Next, enter the following values in the first row of the decision table:

   - Violation.Type: **"speed"**

   - Violation.Actual Speed – Violation.Speed Limit **[10..30)**

   - Amount: **500**

- Points: **3**
  Right-click the first row and select **Insert below** to add another row.

9. Enter the following values in the second row of the decision table:

    - Violation.Type: **"speed"**

    - Violation.Actual Speed – Violation.Speed Limit **>= 30**

    - Amount: **1000**

    - Points: **7**
      Right-click the second row and select **Insert below** to add another row.

10. Enter the following values in the third row of the decision table:

    - Violation.Type: **"parking"**

    - Violation.Actual Speed – Violation.Speed Limit **-**

    - Amount: **100**

    - Points: **1**
      Right-click the third row and select **Insert below** to add another row.

11. Enter the following values in the fourth row of the decision table:

    - Violation.Type: **"driving under the influence"**

    - Violation.Actual Speed – Violation.Speed Limit **-**

    - Amount: **1000**

    - Points: **5**

12. Click **Save** to open the **Confirm Save** dialog box and click **Save** again.

13. To define the driver suspension rule, return to the DMN designer canvas, select the **Should the driver be suspended?** decision node, and click the **Edit** icon to open the DMN boxed expression designer.

14. Click **Select expression → Context**.

15. Click **ContextEntry-1**, enter **Total Points** as the **Name**, and select **number** as the **Data Type**.

16. Click the cell next to **Total Points**, select **Literal Expression** from the context menu, and enter **Driver.Points + Fine.Points** as the expression.

17. In the cell below **Driver.Points + Fine.Points**, select **Literal Expression** from the context menu, and enter **if Total Points >= 20 then "Yes" else "No"**.

18. Click **Save** to open the **Confirm Save** dialog box and click **Save** again.
    You have defined how to calculate the fine and the context for deciding when to suspend the driver. You can navigate to the **traffic-violation** project page and click **Build** to build the example project and address any errors noted in the **Alerts** panel.

CHAPTER 4. TEST SCENARIOS

Test scenarios in Red Hat Process Automation Manager enable you to validate the functionality of business rules and business rule data (for rules-based test scenarios) or of DMN models (for DMN-based test scenarios) before deploying them into a production environment. With a test scenario, you use data from your project to set given conditions and expected results based on one or more defined business rules. When you run the scenario, the expected results and actual results of the rule instance are compared. If the expected results match the actual results, the test is successful. If the expected results do not match the actual results, then the test fails.

Red Hat Process Automation Manager currently supports both the new **Test Scenarios** designer and the former **Test Scenarios (Legacy)** designer. The default designer is the new test scenarios designer, which supports testing of both rules and DMN models and provides an enhanced overall user experience with test scenarios. If required, you can continue to use the legacy test scenarios designer, which supports rule-based test scenarios only.

You can run the defined test scenarios in a number of ways, for example, you can run available test scenarios at the project level or inside a specific test scenario asset. Test scenarios are independent and cannot affect or modify other test scenarios. You can run test scenarios at any time during project development in Business Central. You do not have to compile or deploy your decision service to run test scenarios.

You can import data objects from different packages to the same project package as the test scenario. Assets in the same package are imported by default. After you create the necessary data objects and the test scenario, you can use the **Data Objects** tab of the test scenarios designer to verify that all required data objects are listed or to import other existing data objects by adding a **New item**.
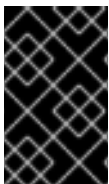
> **IMPORTANT**
>
> Throughout the test scenarios documentation, all references to *test scenarios* and the *test scenarios designer* are for the new version, unless explicitly noted as the legacy version.

4.1. TESTING THE TRAFFIC VIOLATIONS USING TEST SCENARIOS

Use the test scenarios designer in Business Central to test the DMN decision requirements diagrams (DRDs) and define decision logic for the traffic violations project.

Figure 4.1. Test scenario for the traffic violations example

Prerequisites

- You have successfully built the traffic violations project in Business Central.

Procedure

1. On the **traffic-violation** project's home screen, click **Add Asset** to open the **Add Asset** screen.

2. Click **Test Scenario** to open the **Create new Test Scenario** dialog.

   a. Enter **Violation Scenarios** in the **Test Scenario** field.

   b. From the **Package** list, select **com.myspace.traffic_violation**.

   c. Select **DMN** as the **Source type**.

   d. From the **Choose a DMN asset** list, select the path to the DMN asset.

   e. Click **Ok** to open the **Violation Scenarios** test scenario in the **Test Scenarios** designer.

3. Under **Driver** column sub-header, right-click the **State**, **City**, **Age**, and **Name** value cells and select **Delete column** from the context menu options to remove them.

4. Under **Violation** column sub-header, right-click the **Date** and **Code** value cells and select **Delete column** to remove them.

5. Enter the following information in the first row of the test scenarios:

   - Scenario description: **Above speed limit: 10km/h and 30 km/h**

   - Points (under **Given** column header): **10**

   - Type: **"speed"**

   - Speed Limit: **100**

   - Actual Speed: **120**

   - Points (under **Expect** column header): **3**

   - Amount: **500**

   - **Should the driver be suspended? "No"**
     Right-click the first row and select **Insert row below** to add another row.

6. Enter the following information in the second row of the test scenarios:

   - Scenario description: **Above speed limit: more than 30 km/h**

   - Points (under **Given** column header): **10**

   - Type: **"speed"**

   - Speed Limit: **100**

   - Actual Speed: **150**

   - Points (under **Expect** column header): **7**

   - Amount: **1000**

   - **Should the driver be suspended? "No"**
     Right-click the second row and select **Insert row below** to add another row.

7. Enter the following information in the third row of the test scenarios:

   - Scenario description: **Parking violation**

- **Points** (under **Given** column header): **10**

- **Type**: **"parking"**

- **Speed Limit**: leave blank

- **Actual Speed**: leave blank

- **Points** (under **Expect** column header): **1**

- **Amount**: **100**

- **Should the driver be suspended? "No"**
  Right-click the third row and select **Insert row below** to add another row.

8. Enter the following information in the fourth row of the test scenarios:

   - **Scenario description**: **DUI violation**

   - **Points** (under **Given** column header): **10**

   - **Type**: **"driving under the influence"**

   - **Speed Limit**: leave blank

   - **Actual Speed**: leave blank

   - **Points** (under **Expect** column header): **5**

   - **Amount**: **1000**

   - **Should the driver be suspended? "No"**
     Right-click the fourth row and select **Insert row below** to add another row.

9. Enter the following information in the fifth row of the test scenarios:

   - **Scenario description**: **Driver suspended**

   - **Points** (under **Given** column header): **15**

   - **Type**: **"speed"**

   - **Speed Limit**: **100**

   - **Actual Speed**: **140**

   - **Points** (under **Expect** column header): **7**

   - **Amount**: **1000**

   - **Should the driver be suspended? "Yes"**

10. Click **Save** to open the **Confirm Save** dialog box and click **Save** again.

11. Click the play button to check whether the test scenarios pass or fail.

Figure 4.2. Test scenario execution result for the traffic violations example

In case of failure, correct the errors and run the test scenarios again.

CHAPTER 5. DMN MODEL EXECUTION

You can create or import DMN files in your Red Hat Process Automation Manager project using Business Central or package the DMN files as part of your project knowledge JAR (KJAR) file without Business Central. After you implement your DMN files in your Red Hat Process Automation Manager project, you can execute the DMN decision service by deploying the KIE container that contains it to Process Server for remote access and interacting with the container using the Process Server REST API.

For information about including external DMN assets with your project packaging and deployment method, see *Packaging and deploying a Red Hat Process Automation Manager project* .

5.1. EXECUTING A DMN SERVICE USING THE PROCESS SERVER REST API

Directly interacting with the REST endpoints of Process Server provides the most separation between the calling code and the decision logic definition. The calling code is completely free of direct dependencies, and you can implement it in an entirely different development platform such as **Node.js** or **.NET**. The examples in this section demonstrate Nix-style curl commands but provide relevant information to adapt to any REST client.

For more information about the Process Server REST API, see *Interacting with Red Hat Process Automation Manager using KIE APIs*.

Prerequisites

- Process Server is installed and configured, including a known user name and credentials for a user with the **kie-server** role. For installation options, see *Planning a Red Hat Process Automation Manager installation*.

- You have built the DMN project as a KJAR artifact and deployed it to Process Server. Ideally, you have built the DMN project as an executable model for more efficient execution:

  ```
  mvn clean install -DgenerateDMNModel=yes
  ```

  For more information about project packaging and deployment and executable models, see *Packaging and deploying a Red Hat Process Automation Manager project* .

- You have the ID of the KIE container containing the DMN model. If more than one model is present, you must also know the model namespace and model name of the relevant model.

Procedure

1. Determine the base URL for accessing the Process Server REST API endpoints. This requires knowing the following values (with the default local deployment values as an example):

   - Host (**localhost**)

   - Port (**8080**)

   - Root context (**kie-server**)

   - Base REST path (**services/rest/**)

   Example base URL in local deployment for the traffic violations project:

http://localhost:8080/kie-server/services/rest/server/containers/traffic-violation_1.0.0

2. Determine user authentication requirements.
   When users are defined directly in the Process Server configuration, HTTP Basic authentication is used and requires the user name and password. Successful requests require that the user have the **kie-server** role.

   The following example demonstrates how to add credentials to a curl request:

   ```
   curl -u username:password <request>
   ```

   If Process Server is configured with Red Hat Single Sign-On, the request must include a bearer token:

   ```
   curl -H "Authorization: bearer $TOKEN" <request>
   ```

3. Specify the format of the request and response. The REST API endpoints work with both JSON and XML formats and are set using request headers:

   JSON

   ```
   curl -H "accept: application/json" -H "content-type: application/json"
   ```

   XML

   ```
   curl -H "accept: application/xml" -H "content-type: application/xml"
   ```

4. (Optional) Query the container for a list of deployed decision models:
   [GET] server/containers/{containerId}/dmn

   Example curl request:

   ```
   curl -u wbadmin:wbadmin -H "accept: application/xml" -X GET "http://localhost:8080/kie-server/services/rest/server/containers/traffic-violation_1.0.0/dmn"
   ```

   Sample XML output:

   ```xml
   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
   <response type="SUCCESS" msg="Ok models successfully retrieved from container 'Traffic-Violation_1.0.0'">
     <dmn-model-info-list>
       <model>
         <model-namespace>https://github.com/kiegroup/drools/kie-dmn/_60b01f4d-e407-43f7-848e-258723b5fac8</model-namespace>
         <model-name>Traffic Violation</model-name>
         <model-id>_2CD7D1AA-BD84-4B43-AD21-B0342ADE655A</model-id>
         <decisions>
           <dmn-decision-info>
             <decision-id>_23428EE8-DC8B-4067-8E67-9D7C53EC975F</decision-id>
             <decision-name>Fine</decision-name>
           </dmn-decision-info>
           <dmn-decision-info>
             <decision-id>_B5EEE2B1-915C-44DC-BE43-C244DC066FD8</decision-id>
   ```

```
                    <decision-name>Should the driver be suspended?</decision-name>
                </dmn-decision-info>
            </decisions>
            <inputs>
                <dmn-inputdata-info>
                    <inputdata-id>_CEB959CD-3638-4A87-93BA-03CD0FB63AE3</inputdata-id>
                    <inputdata-name>Violation</inputdata-name>
                    <inputdata-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                        <local-part>tViolation</local-part>
                        <prefix></prefix>
                    </inputdata-typeref>
                </dmn-inputdata-info>
                <dmn-inputdata-info>
                    <inputdata-id>_B0E810E6-7596-430A-B5CF-67CE16863B6C</inputdata-id>
                    <inputdata-name>Driver</inputdata-name>
                    <inputdata-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                        <local-part>tDriver</local-part>
                        <prefix></prefix>
                    </inputdata-typeref>
                </dmn-inputdata-info>
            </inputs>
            <itemdefinitions>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_9C758F4A-7D72-4D0F-B63F-
2F5B8405980E</itemdefinition-id>
                    <itemdefinition-name>tViolation</itemdefinition-name>
                    <itemdefinition-itemcomponent>
                        <dmn-itemdefinition-info>
                            <itemdefinition-id>_0B6FF1E2-ACE9-4FB3-876B-
5BB30B88009B</itemdefinition-id>
                            <itemdefinition-name>Code</itemdefinition-name>
                            <itemdefinition-typeref>
                                <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60b01f4d-
e407-43f7-848e-258723b5fac8</namespace-uri>
                                <local-part>string</local-part>
                                <prefix></prefix>
                            </itemdefinition-typeref>
                            <itemdefinition-itemcomponent/>
                            <itemdefinition-iscollection>false</itemdefinition-iscollection>
                        </dmn-itemdefinition-info>
                        <dmn-itemdefinition-info>
                            <itemdefinition-id>_27A5DA18-3CA7-4C06-81B7-
CF7F2F050E29</itemdefinition-id>
                            <itemdefinition-name>date</itemdefinition-name>
                            <itemdefinition-typeref>
                                <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                                <local-part>date</local-part>
                                <prefix></prefix>
                            </itemdefinition-typeref>
                            <itemdefinition-itemcomponent/>
                            <itemdefinition-iscollection>false</itemdefinition-iscollection>
```

```
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_8961969A-8A80-4F12-B568-
346920C0F038</itemdefinition-id>
                    <itemdefinition-name>type</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                        <local-part>string</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_7450F12A-3E95-4D5E-8DCE-
2CB1FAC2BDD4</itemdefinition-id>
                    <itemdefinition-name>speed limit</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60b01f4d-
e407-43f7-848e-258723b5fac8</namespace-uri>
                        <local-part>number</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_0A9A6F26-6C14-414D-A9BF-
765E5850429A</itemdefinition-id>
                    <itemdefinition-name>Actual Speed</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                        <local-part>number</local-part>
                        <prefix></prefix>
                    </itemdefinition-typeref>
                    <itemdefinition-itemcomponent/>
                    <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
            </itemdefinition-itemcomponent>
            <itemdefinition-iscollection>false</itemdefinition-iscollection>
        </dmn-itemdefinition-info>
        <dmn-itemdefinition-info>
            <itemdefinition-id>_13C7EFD8-B85C-43BF-94D3-
14FABE39A4A0</itemdefinition-id>
            <itemdefinition-name>tDriver</itemdefinition-name>
            <itemdefinition-itemcomponent>
                <dmn-itemdefinition-info>
                    <itemdefinition-id>_EC11744C-4160-4549-9610-
2C757F40DFE8</itemdefinition-id>
                    <itemdefinition-name>Name</itemdefinition-name>
                    <itemdefinition-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                        <local-part>string</local-part>
```

```
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_E95BE3DB-4A51-4658-A166-
02493EAAC9D2</itemdefinition-id>
                  <itemdefinition-name>Age</itemdefinition-name>
                  <itemdefinition-typeref>
                    <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                    <local-part>number</local-part>
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_7B3023E2-BC44-4BF3-BF7E-
773C240FB9AD</itemdefinition-id>
                  <itemdefinition-name>State</itemdefinition-name>
                  <itemdefinition-typeref>
                    <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                    <local-part>string</local-part>
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_3D4B49DD-700C-4925-99A7-
3B2B873F7800</itemdefinition-id>
                  <itemdefinition-name>city</itemdefinition-name>
                  <itemdefinition-typeref>
                    <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                    <local-part>string</local-part>
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_B37C49E8-B0D9-4B20-9DC6-
D655BB1CA7B1</itemdefinition-id>
                  <itemdefinition-name>Points</itemdefinition-name>
                  <itemdefinition-typeref>
                    <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                    <local-part>number</local-part>
                    <prefix></prefix>
                  </itemdefinition-typeref>
                  <itemdefinition-itemcomponent/>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
```

```
                    </dmn-itemdefinition-info>
                  </itemdefinition-itemcomponent>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
                <dmn-itemdefinition-info>
                  <itemdefinition-id>_A4077C7E-B57A-4DEE-9C65-
7769636316F3</itemdefinition-id>
                  <itemdefinition-name>tFine</itemdefinition-name>
                  <itemdefinition-itemcomponent>
                    <dmn-itemdefinition-info>
                      <itemdefinition-id>_79B152A8-DE83-4001-B88B-
52DFF0D73B2D</itemdefinition-id>
                      <itemdefinition-name>Amount</itemdefinition-name>
                      <itemdefinition-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                        <local-part>number</local-part>
                        <prefix></prefix>
                      </itemdefinition-typeref>
                      <itemdefinition-itemcomponent/>
                      <itemdefinition-iscollection>false</itemdefinition-iscollection>
                    </dmn-itemdefinition-info>
                    <dmn-itemdefinition-info>
                      <itemdefinition-id>_D7CB5F9C-9D55-48C2-83EE-
D47045EC90D0</itemdefinition-id>
                      <itemdefinition-name>Points</itemdefinition-name>
                      <itemdefinition-typeref>
                        <namespace-uri>https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-
E407-43F7-848E-258723B5FAC8</namespace-uri>
                        <local-part>number</local-part>
                        <prefix></prefix>
                      </itemdefinition-typeref>
                      <itemdefinition-itemcomponent/>
                      <itemdefinition-iscollection>false</itemdefinition-iscollection>
                    </dmn-itemdefinition-info>
                  </itemdefinition-itemcomponent>
                  <itemdefinition-iscollection>false</itemdefinition-iscollection>
                </dmn-itemdefinition-info>
            </itemdefinitions>
            <decisionservices/>
          </model>
      </dmn-model-info-list>
    </response>
```

Sample JSON output:

```
{
  "type" : "SUCCESS",
  "msg" : "OK models successfully retrieved from container 'Traffic-Violation_1.0.0'",
  "result" : {
    "dmn-model-info-list" : {
      "models" : [ {
        "model-namespace" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
        "model-name" : "Traffic Violation",
        "model-id" : "_2CD7D1AA-BD84-4B43-AD21-B0342ADE655A",
```

```
"decisions" : [ {
  "decision-id" : "_23428EE8-DC8B-4067-8E67-9D7C53EC975F",
  "decision-name" : "Fine"
}, {
  "decision-id" : "_B5EEE2B1-915C-44DC-BE43-C244DC066FD8",
  "decision-name" : "Should the driver be suspended?"
} ],
"inputs" : [ {
  "inputdata-id" : "_CEB959CD-3638-4A87-93BA-03CD0FB63AE3",
  "inputdata-name" : "Violation",
  "inputdata-typeRef" : {
    "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-43F7-848E-258723B5FAC8",
    "local-part" : "tViolation",
    "prefix" : ""
  }
}, {
  "inputdata-id" : "_B0E810E6-7596-430A-B5CF-67CE16863B6C",
  "inputdata-name" : "Driver",
  "inputdata-typeRef" : {
    "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-43F7-848E-258723B5FAC8",
    "local-part" : "tDriver",
    "prefix" : ""
  }
} ],
"itemDefinitions" : [ {
  "itemdefinition-id" : "_13C7EFD8-B85C-43BF-94D3-14FABE39A4A0",
  "itemdefinition-name" : "tDriver",
  "itemdefinition-typeRef" : null,
  "itemdefinition-itemComponent" : [ {
    "itemdefinition-id" : "_EC11744C-4160-4549-9610-2C757F40DFE8",
    "itemdefinition-name" : "Name",
    "itemdefinition-typeRef" : {
      "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-43F7-848E-258723B5FAC8",
      "local-part" : "string",
      "prefix" : ""
    },
    "itemdefinition-itemComponent" : [ ],
    "itemdefinition-isCollection" : false
  }, {
    "itemdefinition-id" : "_E95BE3DB-4A51-4658-A166-02493EAAC9D2",
    "itemdefinition-name" : "Age",
    "itemdefinition-typeRef" : {
      "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-43F7-848E-258723B5FAC8",
      "local-part" : "number",
      "prefix" : ""
    },
    "itemdefinition-itemComponent" : [ ],
    "itemdefinition-isCollection" : false
  }, {
    "itemdefinition-id" : "_7B3023E2-BC44-4BF3-BF7E-773C240FB9AD",
    "itemdefinition-name" : "State",
    "itemdefinition-typeRef" : {
```

```
      "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
        "local-part" : "string",
        "prefix" : ""
      },
      "itemdefinition-itemComponent" : [ ],
      "itemdefinition-isCollection" : false
    }, {
      "itemdefinition-id" : "_3D4B49DD-700C-4925-99A7-3B2B873F7800",
      "itemdefinition-name" : "City",
      "itemdefinition-typeRef" : {
        "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
        "local-part" : "string",
        "prefix" : ""
      },
      "itemdefinition-itemComponent" : [ ],
      "itemdefinition-isCollection" : false
    }, {
      "itemdefinition-id" : "_B37C49E8-B0D9-4B20-9DC6-D655BB1CA7B1",
      "itemdefinition-name" : "Points",
      "itemdefinition-typeRef" : {
        "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
        "local-part" : "number",
        "prefix" : ""
      },
      "itemdefinition-itemComponent" : [ ],
      "itemdefinition-isCollection" : false
    } ],
    "itemdefinition-isCollection" : false
  }, {
    "itemdefinition-id" : "_A4077C7E-B57A-4DEE-9C65-7769636316F3",
    "itemdefinition-name" : "tFine",
    "itemdefinition-typeRef" : null,
    "itemdefinition-itemComponent" : [ {
      "itemdefinition-id" : "_79B152A8-DE83-4001-B88B-52DFF0D73B2D",
      "itemdefinition-name" : "Amount",
      "itemdefinition-typeRef" : {
        "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
        "local-part" : "number",
        "prefix" : ""
      },
      "itemdefinition-itemComponent" : [ ],
      "itemdefinition-isCollection" : false
    }, {
      "itemdefinition-id" : "_D7CB5F9C-9D55-48C2-83EE-D47045EC90D0",
      "itemdefinition-name" : "Points",
      "itemdefinition-typeRef" : {
        "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
        "local-part" : "number",
        "prefix" : ""
      },
      "itemdefinition-itemComponent" : [ ],
```

```
        "itemdefinition-isCollection" : false
      } ],
      "itemdefinition-isCollection" : false
    }, {
      "itemdefinition-id" : "_9C758F4A-7D72-4D0F-B63F-2F5B8405980E",
      "itemdefinition-name" : "tViolation",
      "itemdefinition-typeRef" : null,
      "itemdefinition-itemComponent" : [ {
        "itemdefinition-id" : "_0B6FF1E2-ACE9-4FB3-876B-5BB30B88009B",
        "itemdefinition-name" : "Code",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
          "local-part" : "string",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_27A5DA18-3CA7-4C06-81B7-CF7F2F050E29",
        "itemdefinition-name" : "Date",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
          "local-part" : "date",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_8961969A-8A80-4F12-B568-346920C0F038",
        "itemdefinition-name" : "Type",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
          "local-part" : "string",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_7450F12A-3E95-4D5E-8DCE-2CB1FAC2BDD4",
        "itemdefinition-name" : "Speed Limit",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
43F7-848E-258723B5FAC8",
          "local-part" : "number",
          "prefix" : ""
        },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      }, {
        "itemdefinition-id" : "_0A9A6F26-6C14-414D-A9BF-765E5850429A",
        "itemdefinition-name" : "Actual Speed",
        "itemdefinition-typeRef" : {
          "namespace-uri" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-
```

```
43F7-848E-258723B5FAC8",
         "local-part" : "number",
         "prefix" : ""
       },
        "itemdefinition-itemComponent" : [ ],
        "itemdefinition-isCollection" : false
      } ],
       "itemdefinition-isCollection" : false
     } ],
     "decisionServices" : [ ]
   } ]
  }
 }
}
```

5. Execute the model:
   [POST] server/containers/{containerId}/dmn

> **NOTE**
>
> The attribute **model-namespace** is automatically generated and is different for every user. Ensure that the **model-namespace** and **model-name** attributes that you use match those of the deployed model.

Example curl request:

```
curl -u wbadmin:wbadmin -H "accept: application/json" -H "content-type: application/json" -X
POST "http://localhost:8080/kie-server/services/rest/server/containers/traffic-
violation_1.0.0/dmn" -d "{ \"model-namespace\" : \"https://github.com/kiegroup/drools/kie-
dmn/_60B01F4D-E407-43F7-848E-258723B5FAC8\", \"model-name\" : \"Traffic Violation\",
\"dmn-context\" : {\"Driver\" : {\"Points\" : 15}, \"Violation\" : {\"Type\" : \"speed\", \"Actual
Speed\" : 135, \"Speed Limit\" : 100}}}"
```

Example JSON request:

```
{
  "model-namespace" : "https://github.com/kiegroup/drools/kie-dmn/_60B01F4D-E407-43F7-
848E-258723B5FAC8",
  "model-name" : "Traffic Violation",
  "dmn-context" :
  {
    "Driver" :
    {
      "Points" : 15
    },
    "Violation" :
    {
    "Type" : "speed",
    "Actual Speed" : 135,
    "Speed Limit" : 100
    }
  }
}
```

Example XML request (JAXB format):

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dmn-evaluation-context>
  <dmn-context xsi:type="jaxbListWrapper" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <type>MAP</type>
    <element xsi:type="jaxbStringObjectPair" key="Violation">
      <value xsi:type="jaxbListWrapper">
        <type>MAP</type>
        <element xsi:type="jaxbStringObjectPair" key="Type">
          <value xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema">speed</value>
        </element>
        <element xsi:type="jaxbStringObjectPair" key="Speed Limit">
          <value xsi:type="xs:decimal" xmlns:xs="http://www.w3.org/2001/XMLSchema">100</value>
        </element>
        <element xsi:type="jaxbStringObjectPair" key="Actual Speed">
          <value xsi:type="xs:decimal" xmlns:xs="http://www.w3.org/2001/XMLSchema">135</value>
        </element>
      </value>
    </element>
    <element xsi:type="jaxbStringObjectPair" key="Driver">
      <value xsi:type="jaxbListWrapper">
        <type>MAP</type>
        <element xsi:type="jaxbStringObjectPair" key="Points">
          <value xsi:type="xs:decimal" xmlns:xs="http://www.w3.org/2001/XMLSchema">15</value>
        </element>
      </value>
    </element>
  </dmn-context>
</dmn-evaluation-context>
```

> **NOTE**
>
> Regardless of the request format, the request requires the following elements:
>
> - Model namespace
>
> - Model name
>
> - Context object containing input values

Example JSON response:

```json
{
  "type": "SUCCESS",
  "msg": "OK from container 'Traffic-Violation_1.0.0'",
  "result": {
    "dmn-evaluation-result": {
      "messages": [],
      "model-namespace": "https://github.com/kiegroup/drools/kie-dmn/_7D8116DE-ADF5-
```

```
4560-A116-FE1A2EAFFF48",
      "model-name": "Traffic Violation",
      "decision-name": [],
      "dmn-context": {
        "Violation": {
          "Type": "speed",
          "Speed Limit": 100,
          "Actual Speed": 135
        },
        "Should Driver be Suspended?": "YES",
        "Driver": {
          "Points": 15
        },
        "Fine": {
          "Points": 7,
          "Amount": 1000
        }
      },
    "decision-results": {
      "_E1AF5AC2-E259-455C-96E4-596E30D3BC86": {
        "messages": [],
        "decision-id": "_E1AF5AC2-E259-455C-96E4-596E30D3BC86",
        "decision-name": "Should the Driver be Suspended?",
        "result": "YES",
        "status": "SUCCEEDED"
      },
      "_D7F02CE0-AF50-4505-AB80-C7D6DE257920": {
        "messages": [],
        "decision-id": "_D7F02CE0-AF50-4505-AB80-C7D6DE257920",
        "decision-name": "Fine",
        "result": {
          "Points": 7,
          "Amount": 1000
        },
      "status": "SUCCEEDED"
    }
  }
 }
}
```

Example XML (JAXB format) response:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<response type="SUCCESS" msg="OK from container 'Traffic_1.0.0-SNAPSHOT'">
  <dmn-evaluation-result>
    <model-namespace>https://github.com/kiegroup/drools/kie-dmn/_A4BCA8B8-CF08-433F-93B2-A2598F19ECFF</model-namespace>
    <model-name>Traffic Violation</model-name>
    <dmn-context xsi:type="jaxbListWrapper"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <type>MAP</type>
      <element xsi:type="jaxbStringObjectPair" key="Violation">
        <value xsi:type="jaxbListWrapper">
          <type>MAP</type>
          <element xsi:type="jaxbStringObjectPair" key="Type">
```

```xml
            <value xsi:type="xs:string"
xmlns:xs="http://www.w3.org/2001/XMLSchema">speed</value>
          </element>
          <element xsi:type="jaxbStringObjectPair" key="Speed Limit">
            <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">100</value>
          </element>
          <element xsi:type="jaxbStringObjectPair" key="Actual Speed">
            <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">135</value>
          </element>
        </value>
      </element>
      <element xsi:type="jaxbStringObjectPair" key="Driver">
        <value xsi:type="jaxbListWrapper">
          <type>MAP</type>
          <element xsi:type="jaxbStringObjectPair" key="Points">
            <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">15</value>
          </element>
        </value>
      </element>
      <element xsi:type="jaxbStringObjectPair" key="Fine">
        <value xsi:type="jaxbListWrapper">
          <type>MAP</type>
          <element xsi:type="jaxbStringObjectPair" key="Points">
            <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">7</value>
          </element>
          <element xsi:type="jaxbStringObjectPair" key="Amount">
            <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">1000</value>
          </element>
        </value>
      </element>
      <element xsi:type="jaxbStringObjectPair" key="Should the driver be suspended?">
        <value xsi:type="xs:string"
xmlns:xs="http://www.w3.org/2001/XMLSchema">Yes</value>
      </element>
    </dmn-context>
    <messages/>
    <decisionResults>
      <entry>
        <key>_4055D956-1C47-479C-B3F4-BAEB61F1C929</key>
        <value>
          <decision-id>_4055D956-1C47-479C-B3F4-BAEB61F1C929</decision-id>
          <decision-name>Fine</decision-name>
          <result xsi:type="jaxbListWrapper"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
            <type>MAP</type>
            <element xsi:type="jaxbStringObjectPair" key="Points">
              <value xsi:type="xs:decimal"
xmlns:xs="http://www.w3.org/2001/XMLSchema">7</value>
            </element>
            <element xsi:type="jaxbStringObjectPair" key="Amount">
              <value xsi:type="xs:decimal"
```

```
xmlns:xs="http://www.w3.org/2001/XMLSchema">1000</value>
                    </element>
                </result>
                <messages/>
                <status>SUCCEEDED</status>
            </value>
        </entry>
        <entry>
            <key>_8A408366-D8E9-4626-ABF3-5F69AA01F880</key>
            <value>
                <decision-id>_8A408366-D8E9-4626-ABF3-5F69AA01F880</decision-id>
                <decision-name>Should the driver be suspended?</decision-name>
                <result xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Yes</result>
                <messages/>
                <status>SUCCEEDED</status>
            </value>
        </entry>
    </decisionResults>
  </dmn-evaluation-result>
</response>
```

CHAPTER 6. ADDITIONAL RESOURCES

- *Designing a decision service using DMN models*

- *Testing a decision service using test scenarios*

- *Managing projects in Business Central*

- *Interacting with Red Hat Process Automation Manager using KIE APIs*

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Wednesday, May 5, 2021.