Red Hat Process Automation Manager 7.1

Testing a decision service using test scenarios

Last Updated: 2018-10-10

Red Hat Process Automation Manager 7.1 Testing a decision service using test scenarios

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

Legal Notice

Abstract

This document describes how to test a decision service using test scenarios in Red Hat Process Automation Manager 7.1.

Table of Contents

PREFACE

As a business analyst or business rules developer, you can use test scenarios in Business Central to test a decision service before a project is deployed. Testing a decision service ensures that rule assets in the project function properly and as expected. You can test a decision service at any time during project development.

Prerequisites

- The team and project for the decision service have been created in Business Central. For details, see *Getting started with decision services*.
- Business rules and their associated data objects have been defined for the decision service. For example, see *Designing a decision service using guided decision tables*.

NOTE

Having defined business rules is not a technical prerequisite for test scenarios, because the scenarios can test the defined data that constitutes the business rules. However, creating the rules first is helpful so that you can also test entire rules in test scenarios and so that the scenarios more closely match the intended decision service.

CHAPTER 1. TEST SCENARIOS

Test scenarios in Red Hat Process Automation Manager enable you to validate the functionality of business rules and business rule data 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.

Test scenarios can be executed one at a time or as a group. The group execution contains all the scenarios from one package. Test scenarios are independent, so one scenario cannot affect or modify the other. 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.

All data objects related to a test scenario must be in 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**.

CHAPTER 2. DATA OBJECTS

Data objects are the building blocks for the rule assets that you create. Data objects are custom data types implemented as Java objects in specified packages of your project. For example, you might create a **`Person`** object with data fields **`Name`**, **`Address`**, and **`DateOfBirth`** to specify personal details for loan application rules. These custom data types determine what data your assets and your decision service are based on.

2.1. CREATING DATA OBJECTS

The following procedure is a generic overview of creating data objects. It is not specific to a particular business process.

Procedure

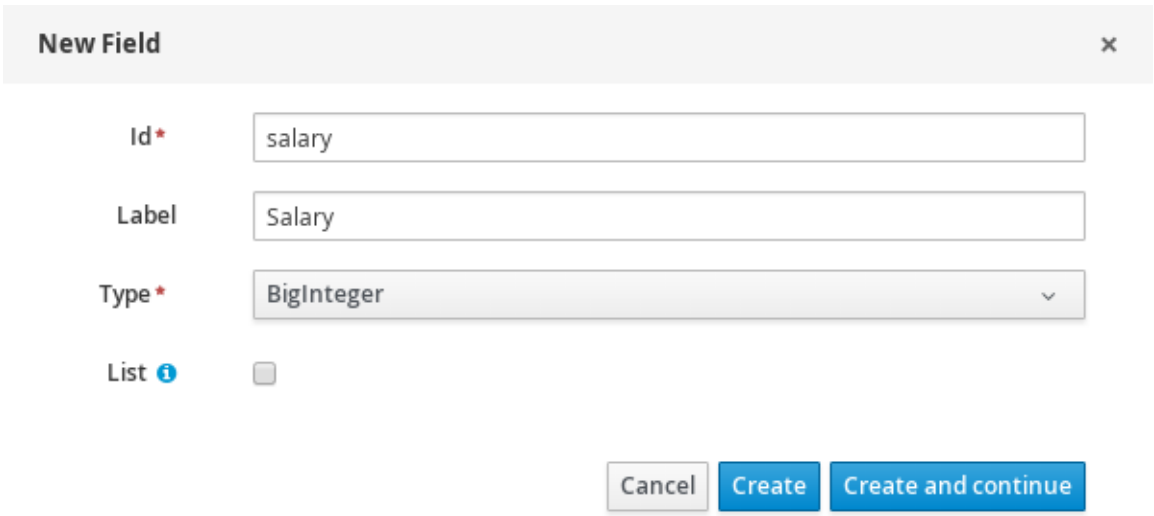
1. In Business Central, go to **Menu** → **Design** → **Projects** and click the project name.
2. Click **Add Asset** → **Data Object**.
3. Enter a unique **Data Object** name and select the **Package** where you want the data object to be available for other rule assets. Data objects with the same name cannot exist in the same package. The package that you specify must be the same package where the rule assets that require those data objects have been assigned or will be assigned.

IMPORTING DATA OBJECTS FROM OTHER PACKAGES

You can also import an existing data object from another package into the package of the rule asset. Select the relevant rule asset within the project and in the asset designer, go to **Data Objects** → **New item** to select the object to be imported.

4. To make your data object persistable, select the **Persistable** checkbox. Persistable data objects are able to be stored in a database according to the JPA specification. The default JPA is Hibernate.
5. Click **Ok**.
6. In the data object designer, click **add field** to add a field to the object with the attributes **Id**, **Label**, and **Type**. Required attributes are marked with an asterisk (*).
   - **Id:** Enter the unique ID of the field.
   - **Label:** (Optional) Enter a label for the field.
   - **Type:** Enter the data type of the field.
   - **List:** Select this check box to enable the field to hold multiple items for the specified type.

Figure 2.1. Add data fields to a data object



7. Click **Create** to add the new field, or click **Create and continue** to add the new field and continue adding other fields.

NOTE

To edit a field, select the field row and use the **general properties** on the right side of the screen.

CHAPTER 3. CREATING AND RUNNING A TEST SCENARIO

You can create test scenarios in Business Central to test the functionality of business rule data before deployment. A basic test scenario must have at least the following data:

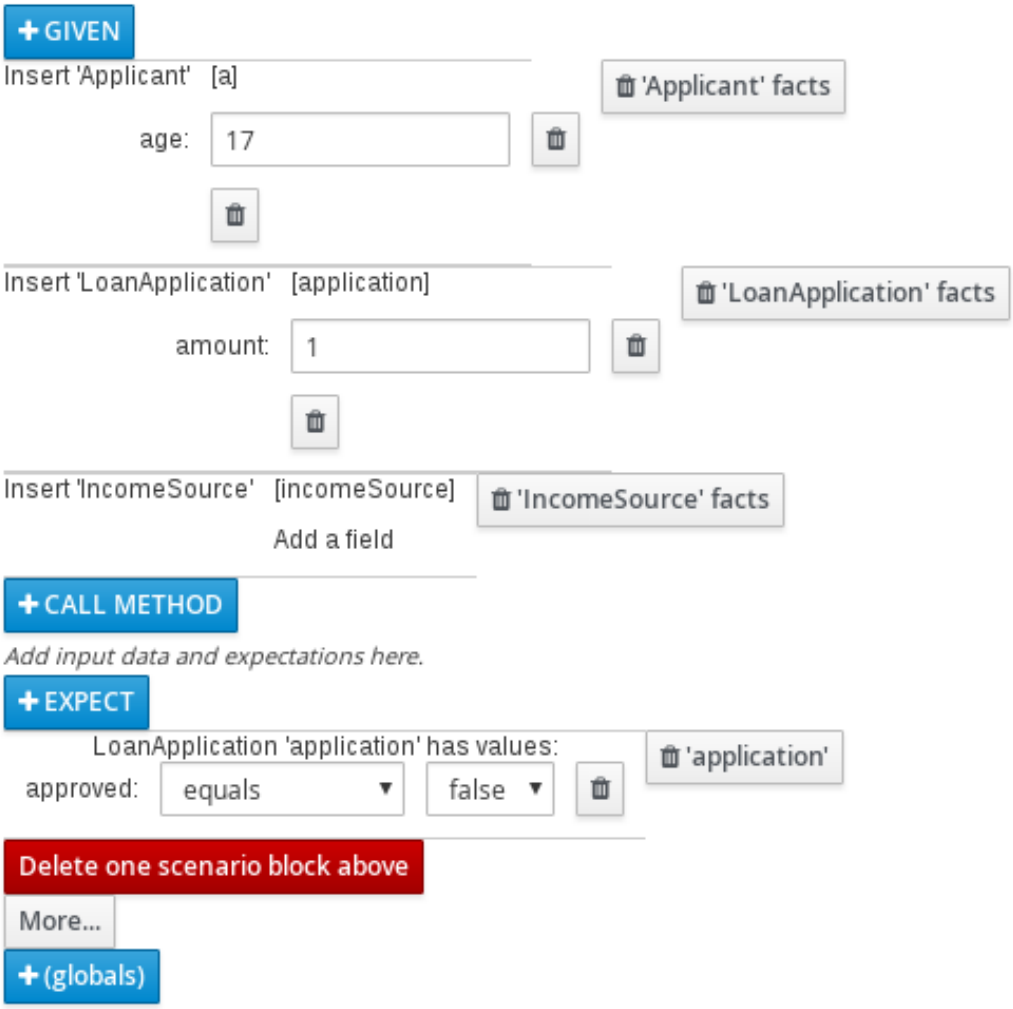
- Related data objects
- **GIVEN** facts
- **EXPECT** results

With this data, the test scenario can validate the expected and actual results for that rule instance based on the defined facts. You can also add a **CALL METHOD** and any available **globals** to a test scenario, but these scenario settings are optional.

Procedure

1. In Business Central, go to **Menu** → **Design** → **Projects** and click the project name.
2. Click **Add Asset** → **Test Scenario**.
3. Enter an informative **Test Scenario** name and select the appropriate **Package**. The package that you specify must be the same package where the required data objects and rule assets have been assigned or will be assigned.
4. Click **Ok** to create the test scenario.
   The new test scenario is now listed in the **Test Scenarios** panel of the **Project Explorer**,
5. Click the **Data Objects** tab to verify that all data objects required for the rules that you want to test are listed. If not, click **New item** to import the needed data objects from other packages, or create data objects within your package.
6. After all data objects are in place, return to the **Model** tab of the test scenarios designer and define the **GIVEN** and **EXPECT** data for the scenario, based on the available data objects.

Figure 3.1. The test scenarios designer



The **GIVEN** section defines the input facts for the test. For example, if an **`Underage`** rule in the project declines loan applications for applicants under the age of 21, then the **GIVEN** facts in the test scenario could be **`Applicant`** with **`age`** set to some integer less than 21.

The **EXPECT** section defines the expected results based on the **GIVEN** input facts. That is, **GIVEN** the input facts, **EXPECT** these other facts to be valid or entire rules to be activated. For example, with the given facts of an applicant under the age of 21 in the scenario, the **EXPECT** results could be **`LoanApplication`** with **`approved`** set to **`false`** (as a result of the underage applicant), or could be the activation of the **`Underage`** rule as a whole.

7. Optionally, add a **CALL METHOD** and any **globals** to the test scenario:

   - **CALL METHOD:** Use this to invoke a method from another fact when the rule execution is initiated. Click **CALL METHOD**, select a fact, and click to select the method to invoke. You can invoke any Java class methods (such as methods from an ArrayList) from the Java library or from a JAR that was imported for the project (if applicable).

   - **globals:** Use this to add any global variables in the project that you want to validate in the test scenario. Click **globals** to select the variable to be validated, and then in the test scenarios designer, click the global name and define field values to be applied to the global variable. If no global variables are available, then they must be created as new assets in Business Central. Global variables are named objects that are visible to the process engine but are different from the objects for facts. Changes in the object of a global do not trigger the re-evaluation of rules.

8. Click **More** at the bottom of the test scenarios designer to add other data blocks to the same scenario file as needed.

9. After you have defined all **GIVEN**, **EXPECT**, and other data for the scenario, click **Save** in the test scenarios designer to save your work.

10. Click **Run scenario** in the upper-right corner to run this `.scenario` file, or click **Run all scenarios** to run all saved `.scenario` files in the project package (if there are multiple). Although the **Run scenario** option does not require the individual `.scenario` file to be saved, the **Run all scenarios** option does require all `.scenario` files to be saved.
    If the test fails, address any problems described in the **Reporting** message at the bottom of the window, review all components in the scenario, and try again to validate the scenario until the scenario passes.

11. Click **Save** in the test scenarios designer to save your work after all changes are complete.

For more details about adding GIVEN facts to test scenarios, see Section 3.1, “Adding GIVEN facts in test scenarios”.

For more details about adding EXPECT results to test scenarios, see Section 3.2, “Adding EXPECT results in test scenarios”.

3.1. ADDING GIVEN FACTS IN TEST SCENARIOS

The **GIVEN** section defines input facts for the test. For example, if an `Underage` rule in the project declines loan applications for applicants under the age of 21, then the **GIVEN** facts in the test scenario could be `Applicant` with `age` set to some integer less than 21.

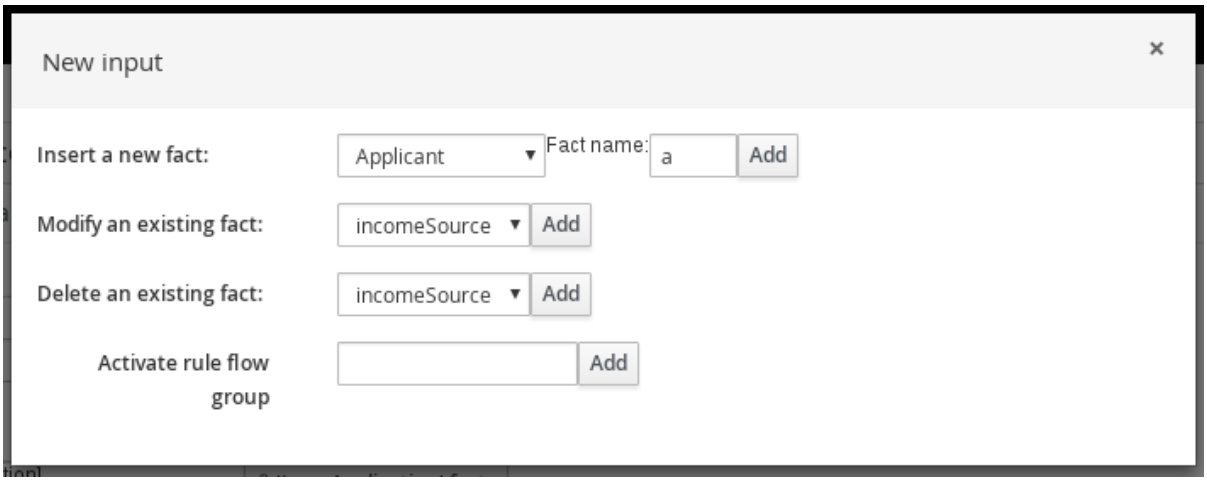
Prerequisite

All data objects required for your test scenario have been created or imported and are listed in the **Data Objects** tab of the test scenarios designer.

Procedure

1. In the test scenarios designer, click **GIVEN** to open the **New input** window with the available facts.

Figure 3.2. Add GIVEN input to the test scenario

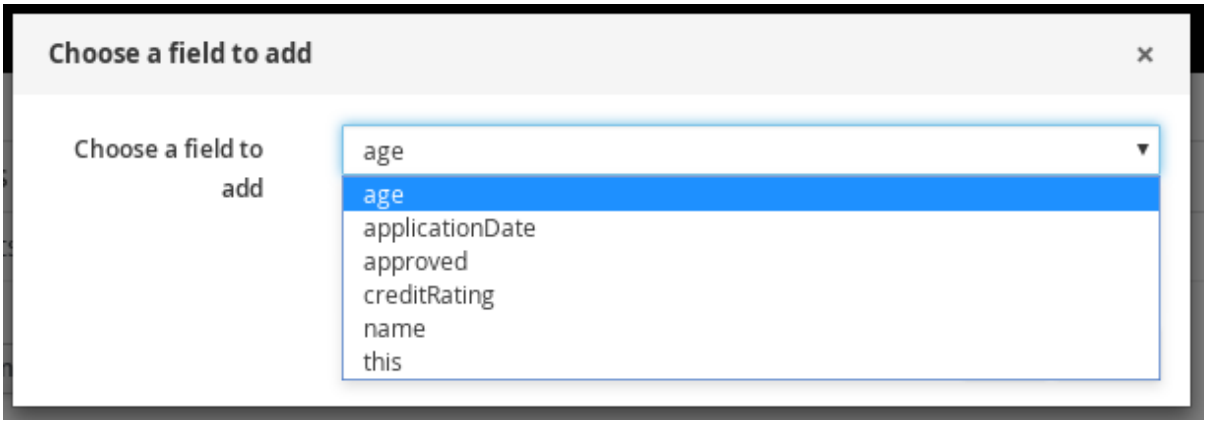


The list includes the following options, depending on the data objects available in the **Data Objects** tab of the test scenarios designer:

- **Insert a new fact:** Use this to add a fact and modify its field values. Enter a variable for the fact as the **Fact name**.
- **Modify an existing fact:** (Appears only after another fact has been added.) Use this to specify a previously inserted fact to be modified in the process engine between executions of the scenario.
- **Delete an existing fact:** (Appears only after another fact has been added.) Use this to specify a previously inserted fact to be deleted from the process engine between executions of the scenario.
- **Activate rule flow group:** Use this to specify a rule flow group to be activated so that all rules within that group can be tested.

2. Choose a fact for the desired input option and click **Add**. For example, set **Insert a new fact:** to **Applicant** and enter **a** or **app** or any other variable for the **Fact name**.
3. Click the fact in the test scenarios designer and select the field to be modified.

Figure 3.3. Modify a fact field



4. Click the edit icon ( ) and select from the following field values:
   - **Literal value:** Creates an open field in which you enter a specific literal value.
   - **Bound variable:** Sets the value of the field to the fact bound to a selected variable. The field type must match the bound variable type.
   - **Create new fact:** Enables you to create a new fact and assign it as a field value of the parent fact. Then you can click the child fact in the test scenarios designer and likewise assign field values or nest other facts similarly.
5. Continue adding any other **GIVEN** input data for the scenario and click **Save** in the test scenarios designer to save your work.

3.2. ADDING EXPECT RESULTS IN TEST SCENARIOS

The **EXPECT** section defines the expected results based on the **GIVEN** input facts. That is, **GIVEN** the input facts, **EXPECT** other specified facts to be valid or entire rules to be activated. For example, with the given facts of an applicant under the age of 21 in the scenario, the **EXPECT** results could be **`LoanApplication`** with **`approved`** set to **`false`** (as a result of the underage applicant), or could be the activation of the **`Underage`** rule as a whole.

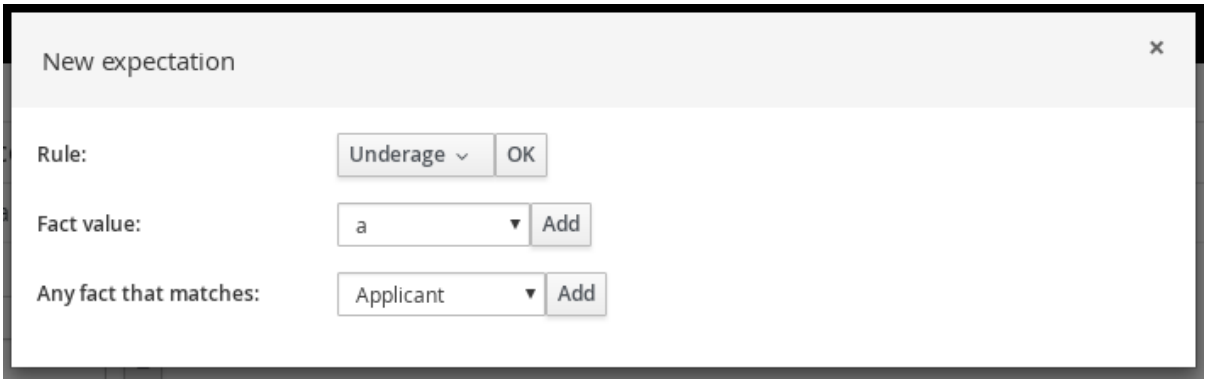
Prerequisite

All data objects required for your test scenario have been created or imported and are listed in the **Data Objects** tab of the test scenarios designer.

Procedure

1. In the test scenarios designer, click **EXPECT** to open the **New expectations** window with the available facts.

   Figure 3.4. Add EXPECT results to the test scenario

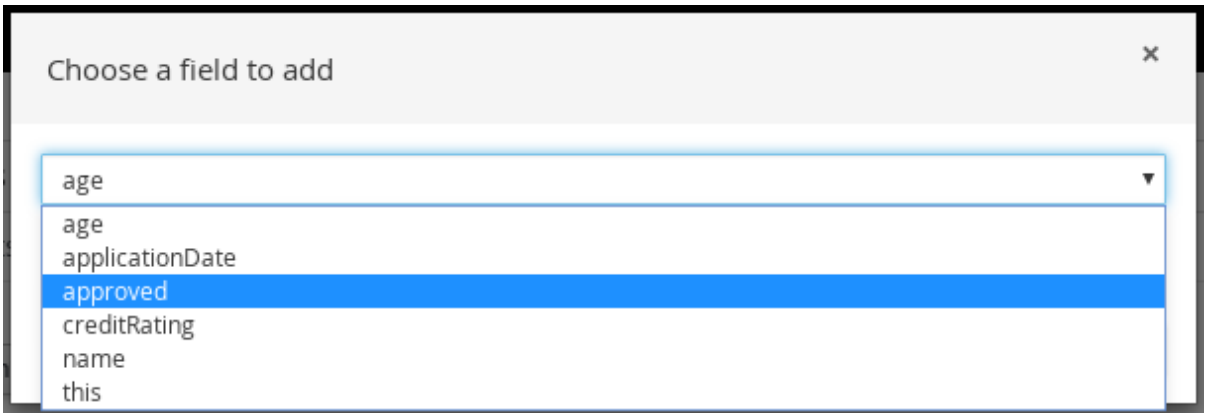
   The list includes the following options, depending on the data in the **GIVEN** section and the data objects available in the **Data Objects** tab of the test scenarios designer:

   - **Rule:** Use this to specify a particular rule in the project that is expected to be activated as a result of the **GIVEN** input. Type the name of a rule that is expected to be activated or select it from the list of rules, and then in the test scenarios designer, specify the number of times the rule should be activated.
   - **Fact value:** Use this to select a fact and define values for it that are expected to be valid as a result of the facts defined in the **GIVEN** section. The facts are listed by the **Fact name** previously defined for the **GIVEN** input.
   - **Any fact that matches:** Use this to validate that at least one fact with the specified values exists as a result of the **GIVEN** input.

2. Choose a fact for the desired expectation (such as **Fact value: `application`**) and click **Add** or **OK**.

3. Click the fact in the test scenarios designer and select the field to be added and modified.

   Figure 3.5. Modify a fact field

4. Set the field values to what is expected to be valid as a result of the **GIVEN** input (such as **`approved`** | **`equals`** | **`false`**).

5. Continue adding any other **EXPECT** input data for the scenario and click **Save** in the test scenarios designer to save your work.

6. After you have defined and saved all **GIVEN**, **EXPECT**, and other data for the scenario, click **Run scenario** in the upper-right corner to run this **`.scenario`** file, or click **Run all scenarios** to run all saved **`.scenario`** files in the project package (if there are multiple). Although the **Run scenario** option does not require the individual **`.scenario`** file to be saved, the **Run all scenarios** option does require all **`.scenario`** files to be saved.
   If the test fails, address any problems described in the **Reporting** message at the bottom of the window, review all components in the scenario, and try again to validate the scenario until the scenario passes.

7. Click **Save** in the test scenarios designer to save your work after all changes are complete.

CHAPTER 4. NEXT STEPS

Packaging and deploying a Red Hat Process Automation Manager project

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Friday, October 12, 2018.