

**EBX5 APIS MDM Add-On**

**Technical Specification**

**Sprint 1.8.0**

Version 0.1

Last update: January 09, 2018

Copyright Orchestra Networks 2000, 2018

Record of changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** | **Affected sections** |
| 01/09/18 | 0.1 | Hai Truong Cong | Create document | All sections |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Table of contents

[Record of changes 2](#_Toc504115874)

[Table of contents 3](#_Toc504115875)

[Table of figures 7](#_Toc504115876)

[1. General 8](#_Toc504115877)

[1.1 Scope 8](#_Toc504115878)

[1.2 Global overview 8](#_Toc504115879)

[1.3 List of requirements 8](#_Toc504115880)

[2. Requirements 11](#_Toc504115881)

[2.1 Req 04 – [Test] Add a table into 'Test execution' group in Test domain to store the check list for monkey test. 11](#_Toc504115882)

[2.1.1 Description 11](#_Toc504115883)

[2.1.2 Analysis 11](#_Toc504115884)

[2.1.3 Solution 18](#_Toc504115885)

[2.1.3.1 Class diagram 18](#_Toc504115886)

[2.1.3.2 Sequence diagram 19](#_Toc504115887)

[2.1.3.3 Miscellaneous 19](#_Toc504115888)

[2.1.4 Limitations 19](#_Toc504115889)

[2.1.5 Technical questions 19](#_Toc504115890)

[2.2 Req 05 – [Test] Add a table into the 'Test execution' group in Test domain to store the result of verified mantis task. 20](#_Toc504115891)

[2.2.1 Description 20](#_Toc504115892)

[2.2.2 Analysis 20](#_Toc504115893)

[2.2.3 Solution 20](#_Toc504115894)

[2.2.3.1 Class diagram 20](#_Toc504115895)

[2.2.3.2 Detail solution 20](#_Toc504115896)

[2.2.3.3 Miscellaneous 20](#_Toc504115897)

[2.2.4 Limitations 20](#_Toc504115898)

[2.2.5 Technical questions 20](#_Toc504115899)

[2.3 Req 06 – Order of Input output context on hierarchy views in Test data should be the same of step in scenario in campaign definition. 20](#_Toc504115900)

[2.3.1 Description 21](#_Toc504115901)

[2.3.2 Analysis 21](#_Toc504115902)

[2.3.3 Solution 25](#_Toc504115903)

[2.3.3.1 Class diagram 25](#_Toc504115904)

[2.3.3.2 Detail solution 26](#_Toc504115905)

[2.3.3.3 Miscellaneous 26](#_Toc504115906)

[2.3.4 Limitations 26](#_Toc504115907)

[2.3.5 Technical questions 26](#_Toc504115908)

[2.4 Req 07 – Remain filtering upon creating new record 27](#_Toc504115909)

[2.4.1 Description 27](#_Toc504115910)

[2.4.2 Analysis 27](#_Toc504115911)

[2.4.3 Solution 27](#_Toc504115912)

[2.4.3.1 Class diagram 27](#_Toc504115913)

[2.4.3.2 Detail solution 27](#_Toc504115914)

[2.4.3.3 Miscellaneous 27](#_Toc504115915)

[2.4.4 Limitations 27](#_Toc504115916)

[2.4.5 Technical questions 27](#_Toc504115917)

[2.5 Req 09 – It should be possible to search for data on all node levels in a data hierarchy view. 28](#_Toc504115918)

[2.5.1 Description 28](#_Toc504115919)

[2.5.2 Analysis 28](#_Toc504115920)

[2.5.3 Solution 28](#_Toc504115921)

[2.5.3.1 Class diagram 28](#_Toc504115922)

[2.5.3.2 Sequence diagram 29](#_Toc504115923)

[2.5.3.3 Miscellaneous 29](#_Toc504115924)

[2.5.4 Limitations 29](#_Toc504115925)

[2.5.5 Technical questions 30](#_Toc504115926)

[2.6 Req 10 – The ability to support the agile process. 30](#_Toc504115927)

[2.6.1 Description 30](#_Toc504115928)

[2.6.2 Analysis 30](#_Toc504115929)

[2.6.3 Solution 30](#_Toc504115930)

[2.6.3.1 Class diagram 30](#_Toc504115931)

[2.6.3.2 Detail solution 30](#_Toc504115932)

[2.6.3.3 Miscellaneous 30](#_Toc504115933)

[2.6.4 Limitations 30](#_Toc504115934)

[2.6.5 Technical questions 30](#_Toc504115935)

[2.7 Req 11 – [Test] Add a new service to create report by 'Test result' in the 'Verify resolved mantis' table. 31](#_Toc504115936)

[2.7.1 Description 31](#_Toc504115937)

[2.7.2 Analysis 31](#_Toc504115938)

[2.7.3 Solution 31](#_Toc504115939)

[2.7.3.1 Class diagram 31](#_Toc504115940)

[2.7.3.2 Detail solution 31](#_Toc504115941)

[2.7.3.3 Miscellaneous 31](#_Toc504115942)

[2.7.4 Limitations 31](#_Toc504115943)

[2.7.5 Technical questions 31](#_Toc504115944)

[3. Logical data model 33](#_Toc504115945)

[4. API Diagram 34](#_Toc504115946)

[5. JUnit test 35](#_Toc504115947)

[5.1 Req 04: [Test] Add a table into 'Test execution' group in Test domain to store the check list for monkey test. 35](#_Toc504115948)

[6. Performance test 36](#_Toc504115949)

[7. Project plan 37](#_Toc504115950)

[7.1 Milestones 37](#_Toc504115951)

[7.2 Detailed planning 37](#_Toc504115952)

Table of figures

[Figure 21: The ‘**Free form case**’ table and relationship with related tables. 12](#_Toc504115987)

[Figure 22: The detail structure of the **Free form case** table. 12](#_Toc504115988)

[Figure 23: Detailed the list of value for the Test result field. 16](#_Toc504115989)

[Figure 24: Class diagram for the "Import free form case" service. 19](#_Toc504115990)

[Figure 25: Sequence diagram for the "Import free form case" service. 19](#_Toc504115991)

[Figure 231: The order of the Step in the Scenario. 21](#_Toc504115992)

[Figure 232: The order of the Scenario in the Campaign definition. 22](#_Toc504115993)

[Figure 233: The data model detail of the Order field in the **Input output context** table. 23](#_Toc504115994)

[Figure 234: The data model for the Input output context table and related talbes. 23](#_Toc504115995)

[Figure 235: The detail of the **Order** field in the **Input output context** table. 24](#_Toc504115996)

[Figure 236: The list of hierarchy view in the **Input output context** table modify according to the Order field. 25](#_Toc504115997)

[Figure 237: The list of hierarchy view in the **Input output data** table modify according to the Order field. 25](#_Toc504115998)

[Figure 238: Changed by the order in the ordering node combo-box. 25](#_Toc504115999)

[Figure 239: Class diagram for migration the order field. 26](#_Toc504116000)

[Figure 251: Class diagram 28](#_Toc504116001)

[Figure 252: Sequence diagram 29](#_Toc504116002)

# General

Apis is an application for managing all data related to life cycle of an add-on*.*

## Scope

This document is used to describe the technical specification of Apis Add-on Sprint 1.8.0.

## Global overview

The main content of this Sprint is:

* Support the agile process
* Manager the check list for money test

## List of requirements

The Sprint content includes the following services and features:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Name** | **Mantis** | **Description** | **Has detail** |
|  | Req 01: The ability to redefine folder location for Apis driver by using a Java Property | 30353 | To improve the exploitability of the DAM add-on, it should be possible to redefine this location folder by a Java property. | No |
|  | Req 02: Using an enumeration for the label field in the Role wrapper table. | 30342 | In the 'apis-wrapper-forSdg' dataspace, 'Apis directory wrapper for SDG' dataset, an enumeration is used for the label field in the Role wrapper table. | No |
|  | Req 03: It is necessary to remove some fields in the 'Apis internal value' table. | 30198 | Currently, we can use the cache to store the number of records on the table. Therefore, it is necessary to remove some field in the 'Apis internal value' table. | No |
|  | Req 04: [Test] Add a table into 'Test execution' group in Test domain to store the check list for monkey test. | 29081 | A new table named 'Check list for monkey test' should be added to the 'Test execution' group in Test domain to store the check list for monkey test. | Yes |
|  | Req 05: [Test] Add a table into the 'Test execution' group in Test domain to store the result of verified mantis task. | 29079 | A new table named 'Verify resolved mantis' should be added to the 'Test execution' group in Test domain to store the result of verified mantis task. | Yes |
|  | Req 06: Order of Input output context on hierarchy views in Test data should be the same of step in scenario in campaign definition. | 29078 | The order of Input output context on hierarchy views in 'Input output context'/'Input output data' table should be the same as that of step in scenario in campaign definition. | Yes |
|  | Req 07: Remain filtering upon creating new record | 16807 | Eg: create new EBX within a sprint in delivery | Yes |
|  | Req 08: [Reporting] Skip invalid data when generating chart instead of displaying an ajax error message. | 12782 | When related data is invalid, skip it. | No |
|  | Req 09: It should be possible to search for data on all node levels in a data hierarchy view. | 10426 | In a data hierarchy view, user should be able to search for data on all node level. | Yes |
|  | Req 10: The ability to support the agile process. | 30366 | In the delivery domain. we needs to design the tables to support the agile process. | Yes |
|  | Req 11: [Test] Add a new service to create report by 'Test result' in the 'Verify resolved mantis' table. | 30405 | There should be a new service named 'Create report' to generate and export report by the 'Test result' in the 'Verify resolved mantis' table. | Yes |

Table 1: List of requirements

# Requirements

## Req 04 – [Test] Add a table into 'Test execution' group in Test domain to store the check list for monkey test.

### Description

The objective of this requirement is using the Apis to manager the check list for monkey test in the Test execution under the Test domain.

### Analysis

In the Test execution group, create a new table named **'Free form case**' to archive the check list for monkey test.

The detail of the **Free form case** table as following:

|  |  |
| --- | --- |
| **Type** | Table |
| **Name** | FreeFormCase |
| **Label** | **Free form case** |
| **Description** | This table contains the check list for monkey test. |
| **Path** | /root/Apis/TestDriving/TestExecution/FreeFormCase |

The **Free form case** table will have some columns as below:

* Oid
* Code
* Product build
* Feature
* Detail feature
* Literal specification
* Campaign definition
* Scenario
* Step
* Free form case
* Tester
* Execution date
* Test result
* Mantis identifier
* Note

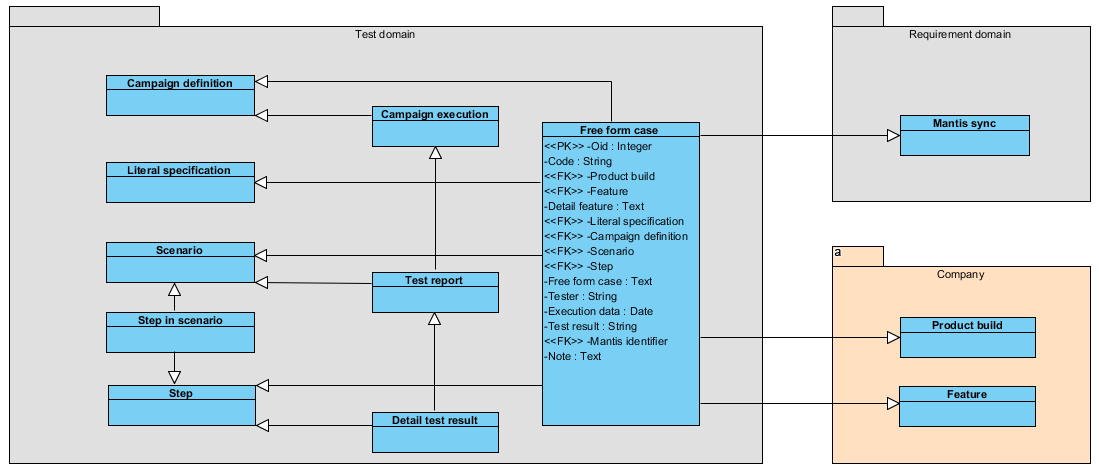


Figure 21: The ‘**Free form case**’ table and relationship with related tables.

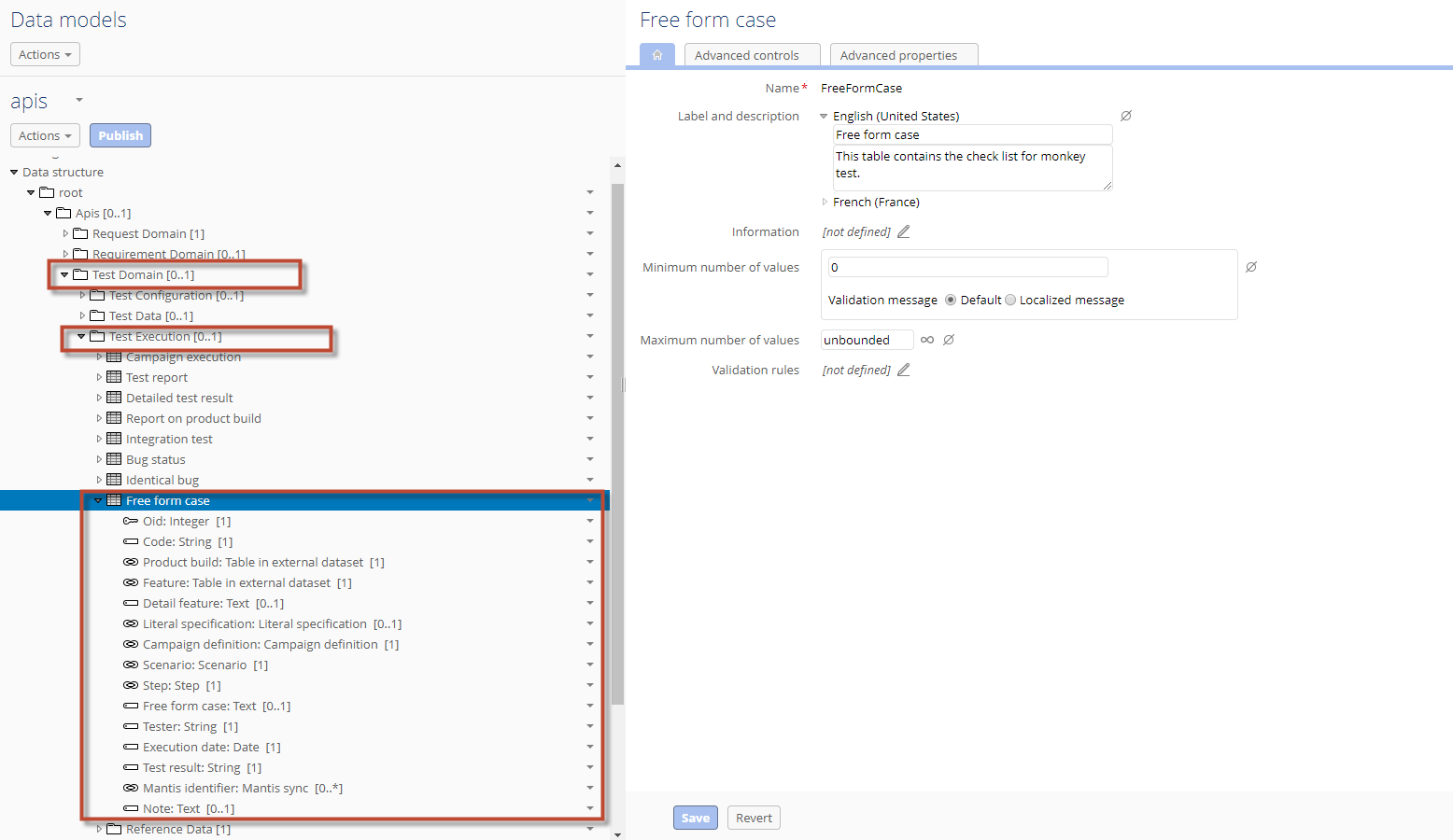


Figure 22: The detail structure of the **Free form case** table.

* The detail of the **Oid** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | oid |
| Label | oid |
| Type | Integer |
| Mandatory | Yes |
| Description | Primary Key.  Hidden field.  Auto-increment. |

* The detail of the **Code** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | code |
| Label | Code |
| Type | String |
| Mandatory | Yes |
| Description | Unique.  The code must start with prefix FF. |

Create a class have named **CampaignDefinitionCodePrefixConstraint** in **com.orchestranetworks.addon.apis.rule.test** package to apply for the **Code** field.

|  |  |
| --- | --- |
| Class | Description |
| CampaignDefinitionCodePrefixConstraint.java | This class will force the code must start with FF prefix.  Extend **CodePrefixConstraint** class. |

* The detail of the **Product build** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | fkProductBuild |
| Label | Product build |
| Type | String |
| Mandatory | Yes |
| Description | Foreign key.  Refer: company: /root/Reality/Product/ProductBuilt |

* The detail of the **Feature** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | fkFeature |
| Label | Feature |
| Type | String |
| Mandatory | No |
| Description | Foreign key.  Refer: company: /root/Reality/Product/Feature |

* The detail of the **Detail feature** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | detailFeature |
| Label | Detail feature |
| Type | Text |
| Mandatory | No |
| Description | Detail for feature |

* The detail of the **Literal specification** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | fkLiteralSpecification |
| Label | Literal specification |
| Type | String |
| Mandatory | No |
| Description | Foreign key.  Refer: /root/Apis/TestDriving/TestConfiguration/LiteralSpecification |

* The detail of the **Campaign definition** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | fkCampaignDefinition |
| Label | Campaign definition |
| Type | String |
| Mandatory | Yes |
| Description | Foreign key.  Refer: /root/Apis/TestDriving/TestConfiguration/CampaignDefinition |

* The detail of the **Scenario** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | fkScenario |
| Label | Scenario |
| Type | String |
| Mandatory | Yes |
| Description | Foreign key.  Refer: /root/Apis/TestDriving/TestConfiguration/Scenario |

* The detail of the **Step** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | fkStep |
| Label | Step |
| Type | String |
| Mandatory | Yes |
| Description | Foreign key.  Refer: /root/Apis/TestDriving/TestConfiguration/Step |

* The detail of the **Free form case** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | freeFormCase |
| Label | Free form case |
| Type | Text |
| Mandatory | No |
| Description | Description for free form case. |

* The detail of the **Tester** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | tester |
| Label | Tester |
| Type | String |
| Mandatory | Yes |
| Description | The tester who are responsible for testing. |

Use the **TestUsersEnumeration** class in the **com.orchestranetworks.addon.apis.rule.test** package for the **Tester** field to filter list of test user role.

* The detail of the **Execution date** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | executionDate |
| Label | Execution date |
| Type | Date |
| Mandatory | Yes |
| Description | The date for execution. |

* The detail of the **Test result** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | testResult |
| Label | Test result |
| Type | String |
| Mandatory | Yes |
| Description | The result of testing. |

Defines a list of predefined possible values for the **Test result** field as below:

|  |  |  |
| --- | --- | --- |
| No | Value | Label |
| 1 | Pass | Pass |
| 2 | Failed | Failed |

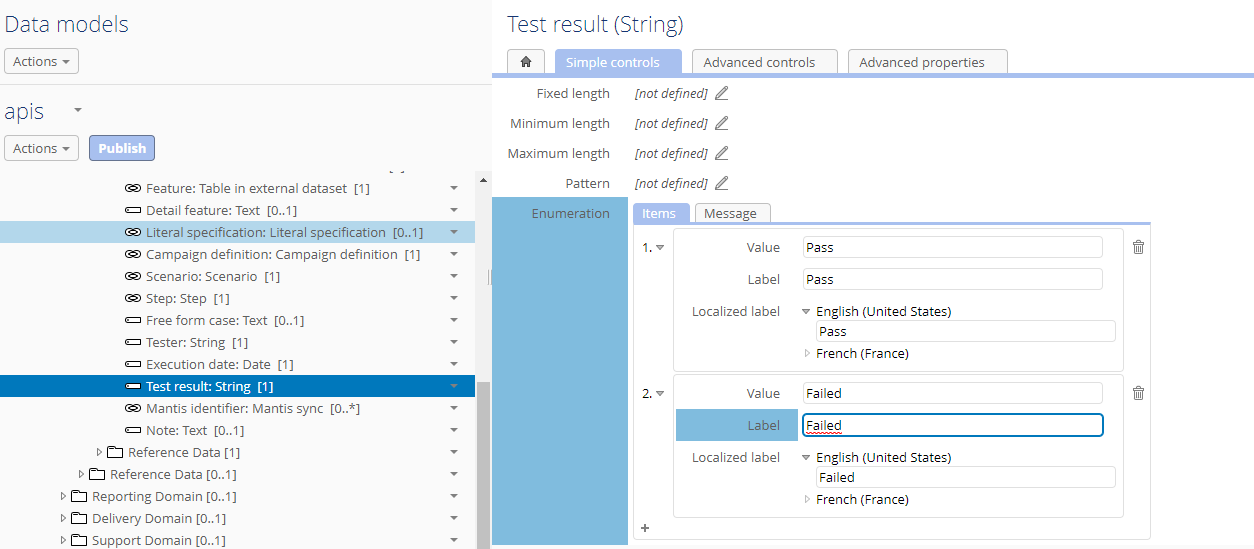


Figure 23: Detailed the list of value for the **Test result** field.

* The detail of the **Mantis identifier** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | fkMantisIdentitier |
| Label | Mantis identifier |
| Type | String |
| Mandatory | No |
| Description | Foreign Key. Multi-value.  Refer: /root/Apis/RequestDriving/MantisSynchronization. |

* The detail of the **Note** field in the **Free from case** table as following:

|  |  |
| --- | --- |
| Name | note |
| Label | Note |
| Type | Text |
| Mandatory | No |
| Description | Note for free form case. |

The data in the **Free form case** table and related tables will be imported from excel file via a new service named ‘**Import free form case’** on the Apis dataset.

This service has main functions as the following:

* To import the check list for monkey test into the **Free form case** table.
* Create new record on the **Step** table.
* To import the associated links between the **Scenario** table and the **step** table into the **Steps in scenarios** table.
* Create the **Campaign execution**, **Test report**, **Detailed test result** and **Failed steps and bugs** tables based on data on record import in the **Free form case** table.

User will select imported file from local to execute importing data and there are three mode of importing as following:

* Update or insert.
* Insert only (default).
* Update only.

Users have to use a file template to input data for the imported file. The template of imported file is defined as below:

- The file must have type is .xls or .xlsx.

- The file template has one sheet: “Free form case” sheet includes column as following:

|  |  |  |
| --- | --- | --- |
| **Column** | **Mandatory** | **Comment** |
| Code | Yes | The code must start with FF. Example: FF-Apis01, FF01… |
| Product build | Yes | Label in the ‘Product build’ table as format:  ${./FKProductVersion} ${./identifier} ${./FKBuiltType}${./order} |
| Feature | No | Label in the ‘Feature’ table as format:  ${./FKDomain} - ${./name} |
| Detail feature | No | Description of feature |
| Literal code | No | Value code field in ‘Literal specification’ table. |
| Campaign code | Yes | Value Campaign code field in ‘Campaign definition’ table. |
| Scenario code | Yes | Value Scenario code field in ‘Scenario’ table. |
| Step code | Yes | Value Code field in ‘Step’ table. |
| Free form case | No | Detail of free form case. |
| Tester | Yes | UserID in SDG. |
| Execution date | Yes | Format: MM/DD/YYYY |
| Test result | Yes | The enumeration: Pass, Failed |
| Mantis identifier | No | List of related mantis. Separated by commas below or break lines. Ex: 30342, 2459  30405 |
| Note | No | Note for free form case. |

Permission:

* User who has ‘tester’ role or ‘addon-test-leader’ role or ‘test manager’ or ‘manager’, can run this service.
* Other users do not see this service.

### Solution

#### Class diagram

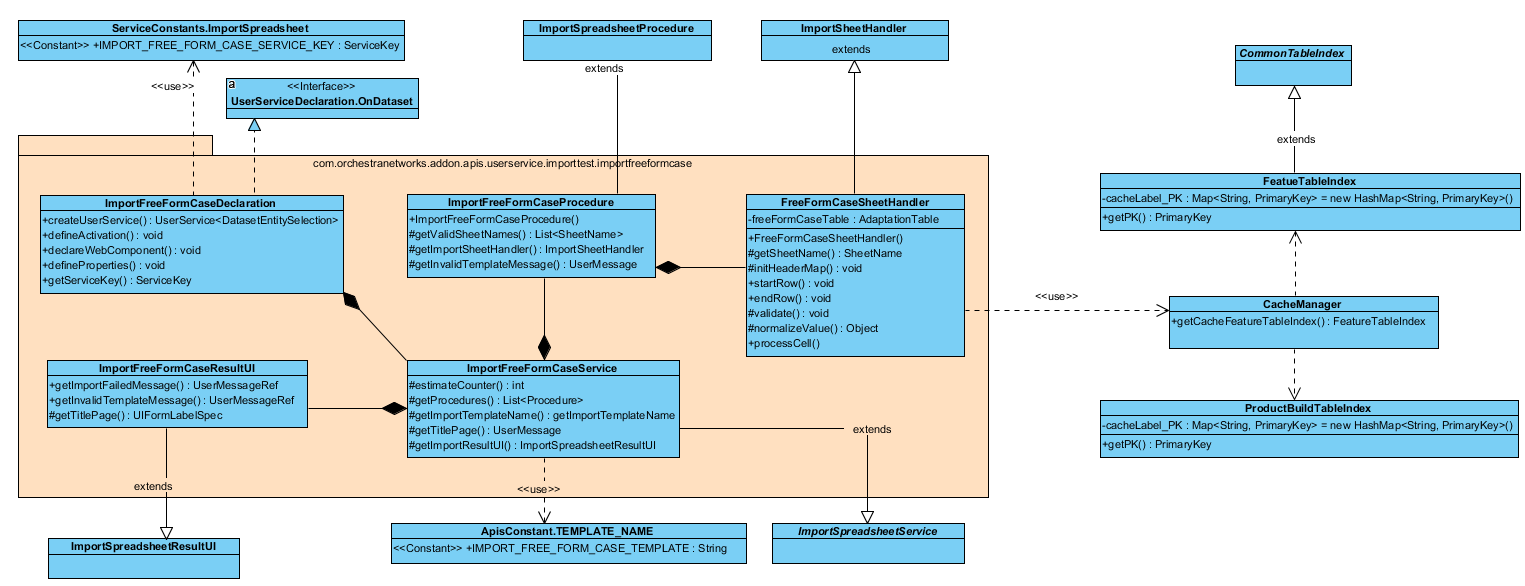


Figure 24: Class diagram for the **Import free form case** service.

#### Sequence diagram

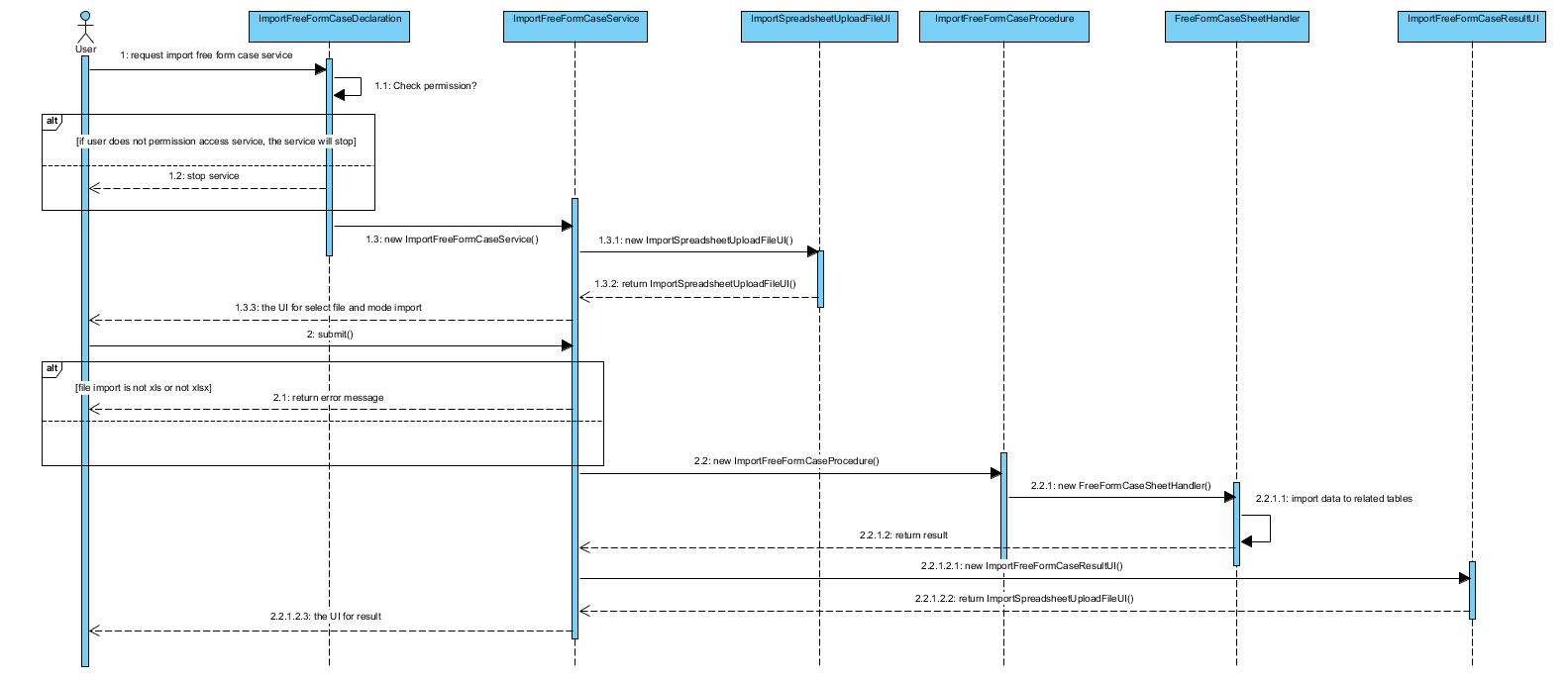


Figure 25: Sequence diagram for the **Import free form case** service.

#### Miscellaneous

### Limitations

### Technical questions

* **Extension library**
* **Technical point**
* User UserService API to create the ‘Import free form case’ service

## Req 05 – [Test] Add a table into the 'Test execution' group in Test domain to store the result of verified mantis task.

### Description

### Analysis

### Solution

#### Class diagram

#### Detail solution

#### Miscellaneous

### Limitations

* **Impacted bug in EBX**

**N/A**

* **Impacted evolution in EBX**

**N/A**

* **Functional limitation**

**N/A**

### Technical questions

* **Extension library**

**N/A**

* **Technical point**

**N/A**

## Req 06 – Order of Input output context on hierarchy views in Test data should be the same of step in scenario in campaign definition.

### Description

The objective of this requirement is the order of **Input output context** on hierarchy views in 'Input output context'/'Input output data' table should be the same as that of **step in scenario in campaign definition**.

### Analysis

In the Step table, open **Steps in campaign definition** hierarchy view, the order of the Step in the Scenario is based on the **order** field in the **Step in scenario** table.

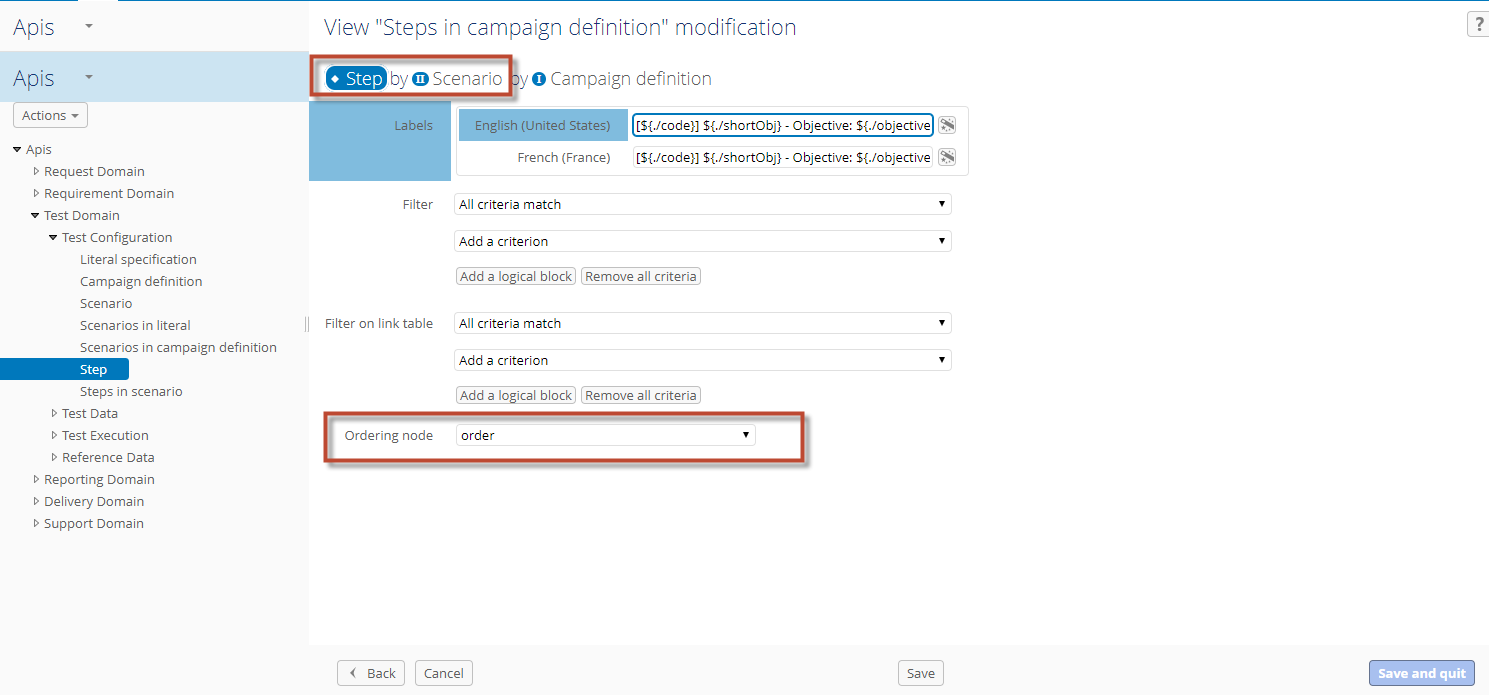


Figure 231: The order of the Step in the Scenario.

The order of the Scenario in the Campaign definition is based on the order field in the **Scenarios in campaign definition** table

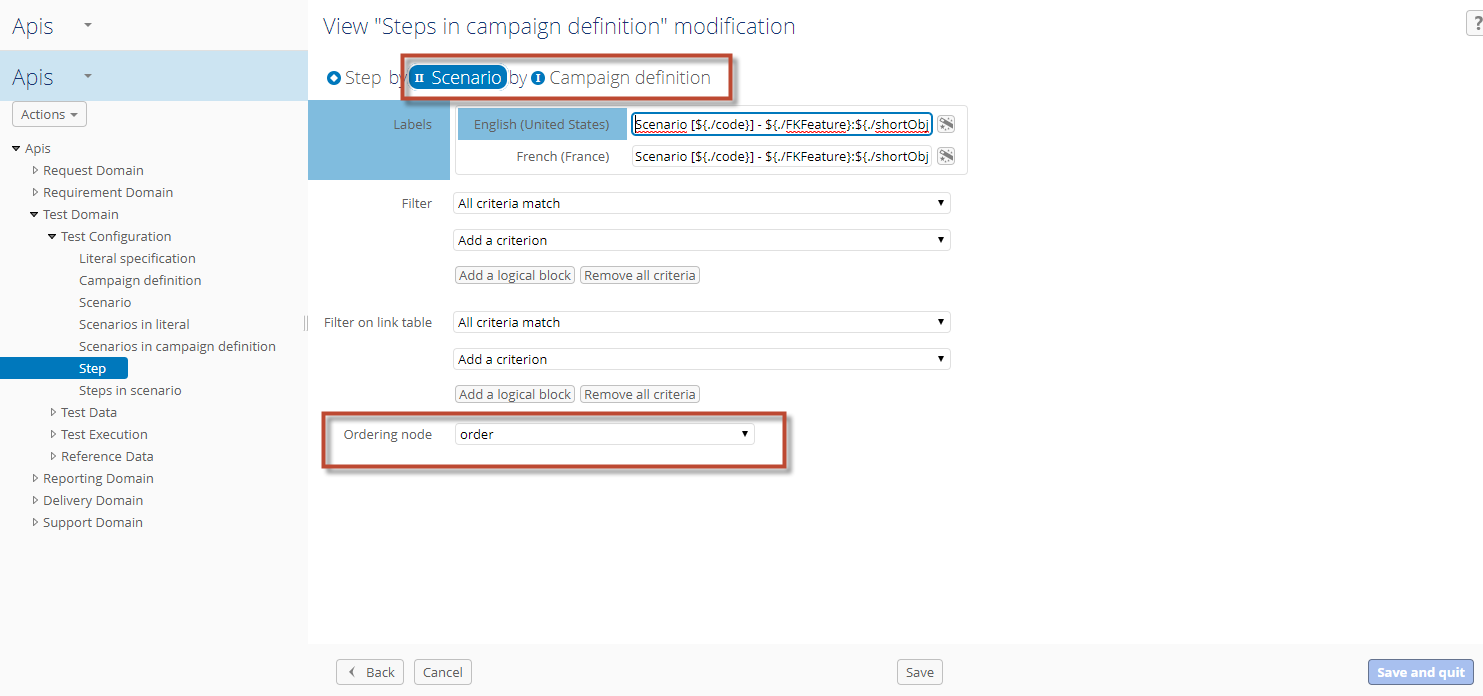


Figure 232: The order of the Scenario in the Campaign definition.

So the order is the same, in the **Input output context** table, we will create a new field named ‘order’. The value of new field is calculated based on the value of the **order** field in the **Step in scenario** table and the value of the order field in the **Scenarios in campaign definition** table.

The details of the **Order** field in the **Input output context** table as below:

|  |  |
| --- | --- |
| Name | order |
| Label | Order |
| Type | Integer |
| Mandatory | No |
| Description | Hidden. |

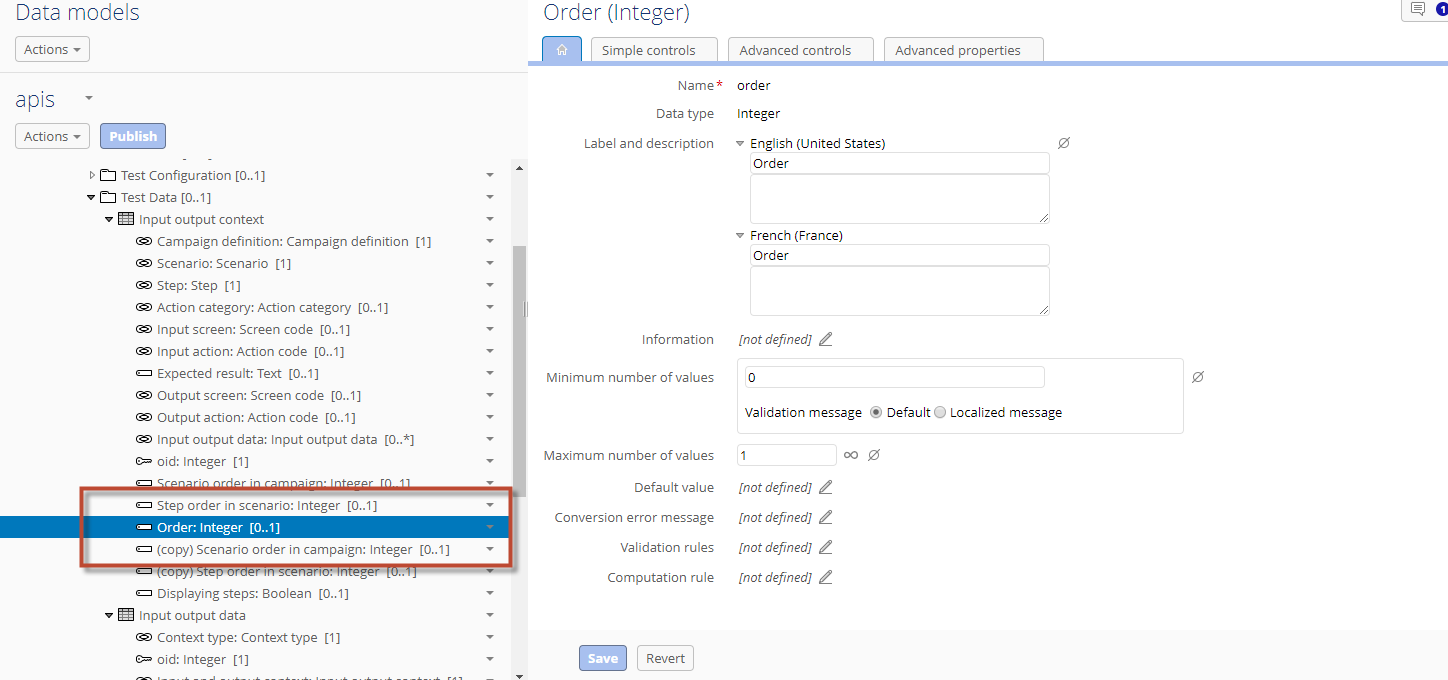


Figure 233: The data model detail of the Order field in the **Input output context** table.

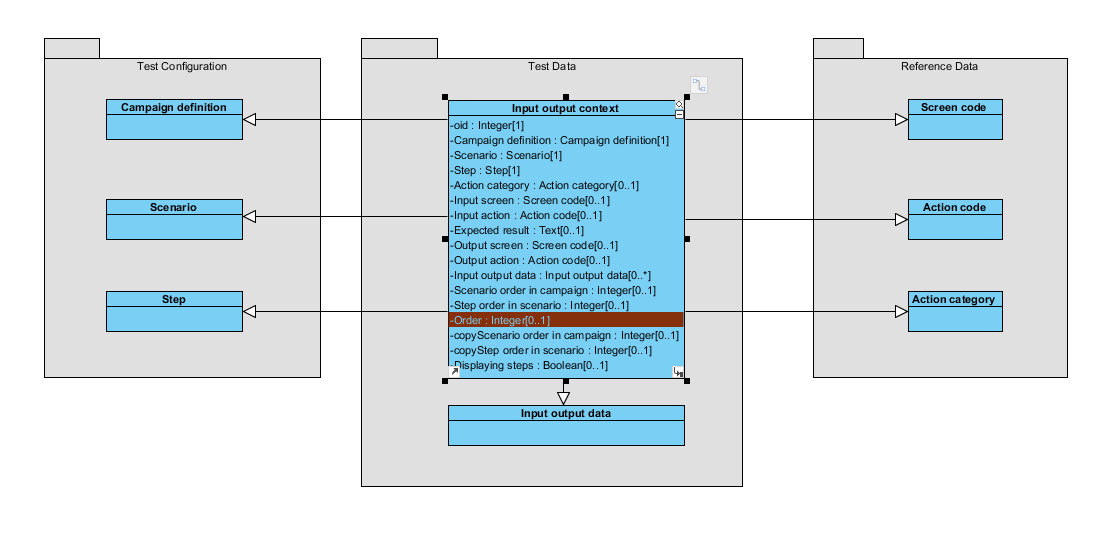


Figure 234: The data model for the Input output context table and related tables.

Currently, the number of steps in a scenario does not exceed 1000 records. Therefore, the value of the **Order** field will be calculated by the following formula:

order = scenarioOrder \* 1000 + stepOrder

In which:

* order: The value of the **order** field in the **Input output context** table.
* scenarioOrder: The value of the **order** field in the **Scenarios in campaign definition** table.
* stepOrder: The value of the **order** field in the **Steps in scenario** table.

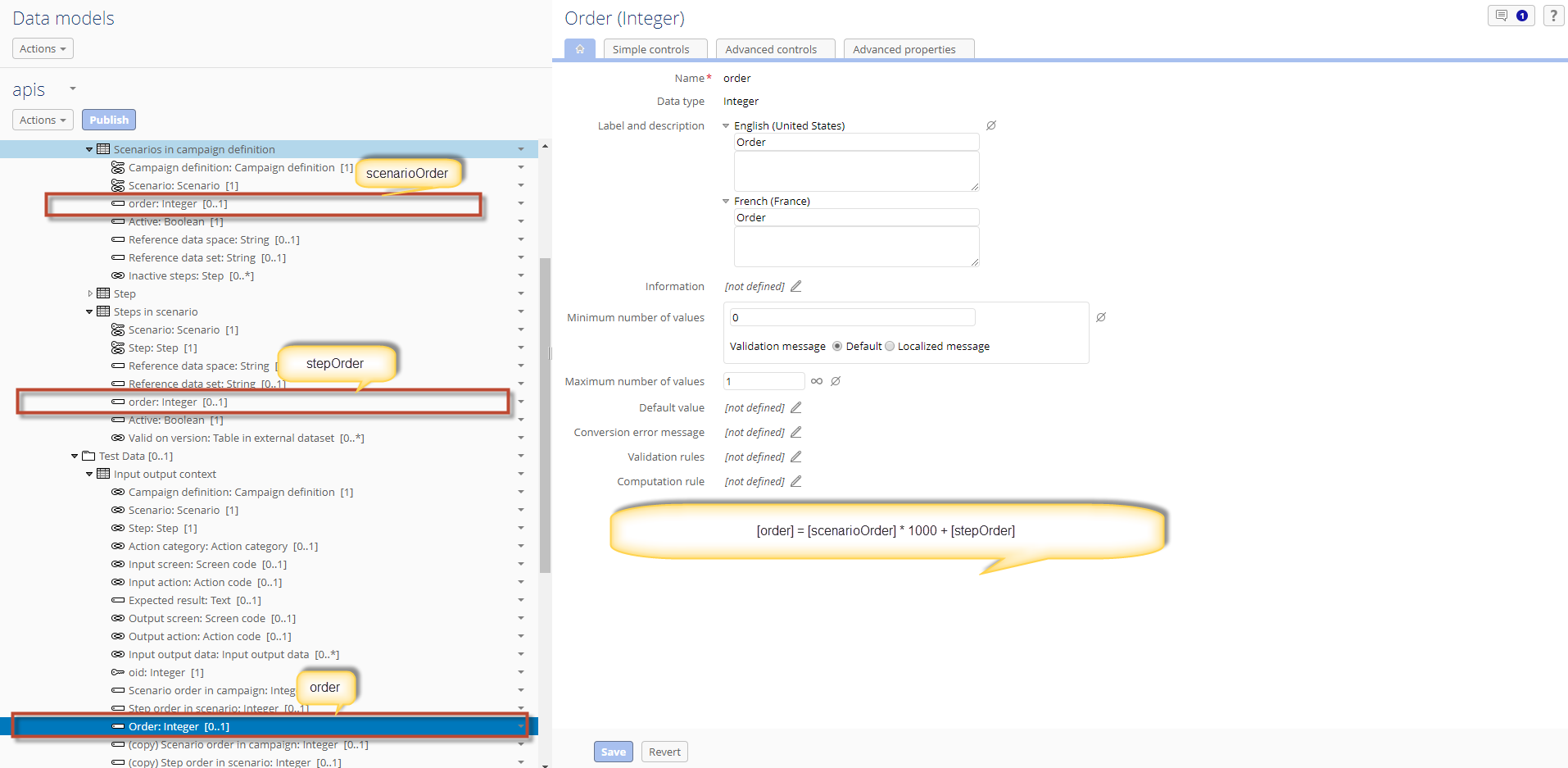


Figure 235: The detail of the **Order** field in the **Input output context** table.

The value of Order field will be automatically calculated by the formula when creating new record.

For those old records, the value of Order field will be calculated at the start of the server by create new class MigrationInputOutputContextProcedure java implements Procedure.

Modify the hierarchy views in the Input output context and Input output data tables according to the Order field.

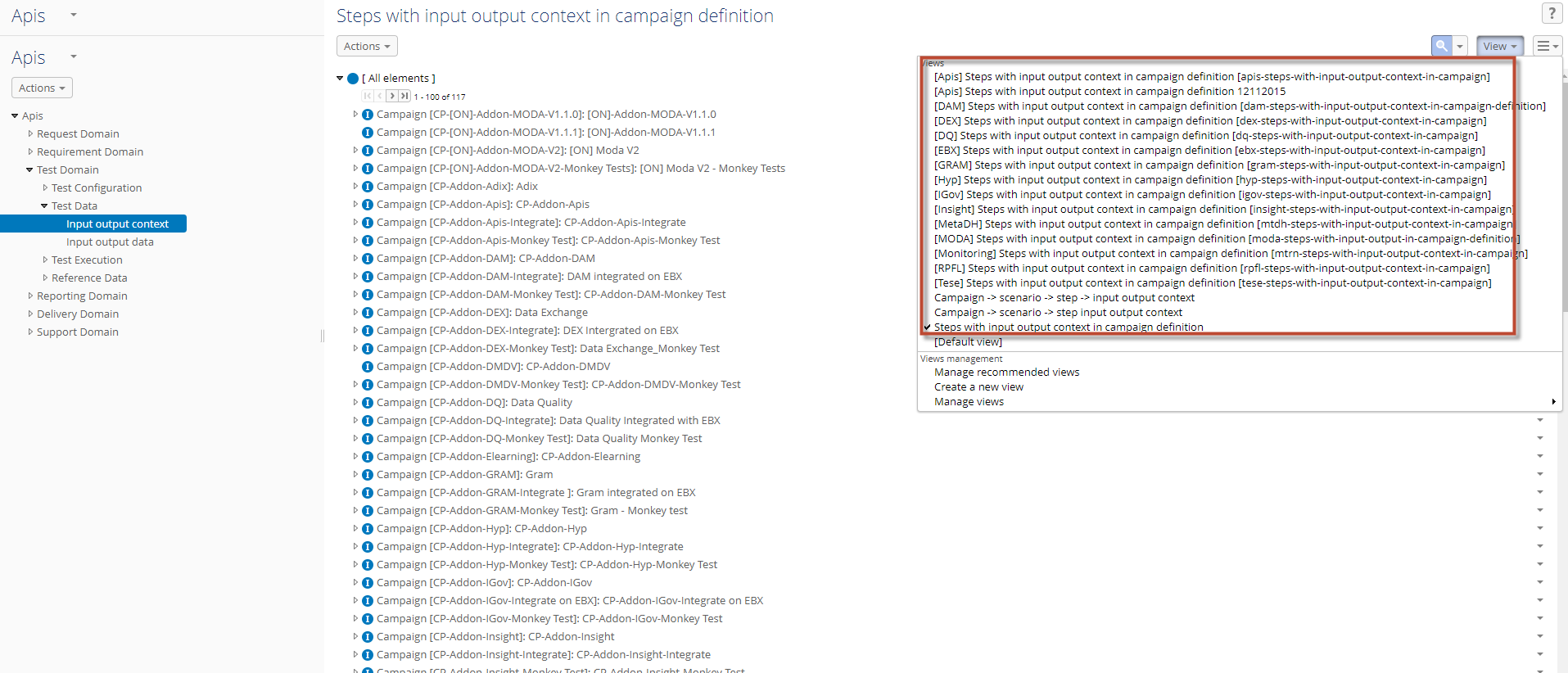


Figure 236: The list of hierarchy view in the **Input output context** table modify according to the Order field.

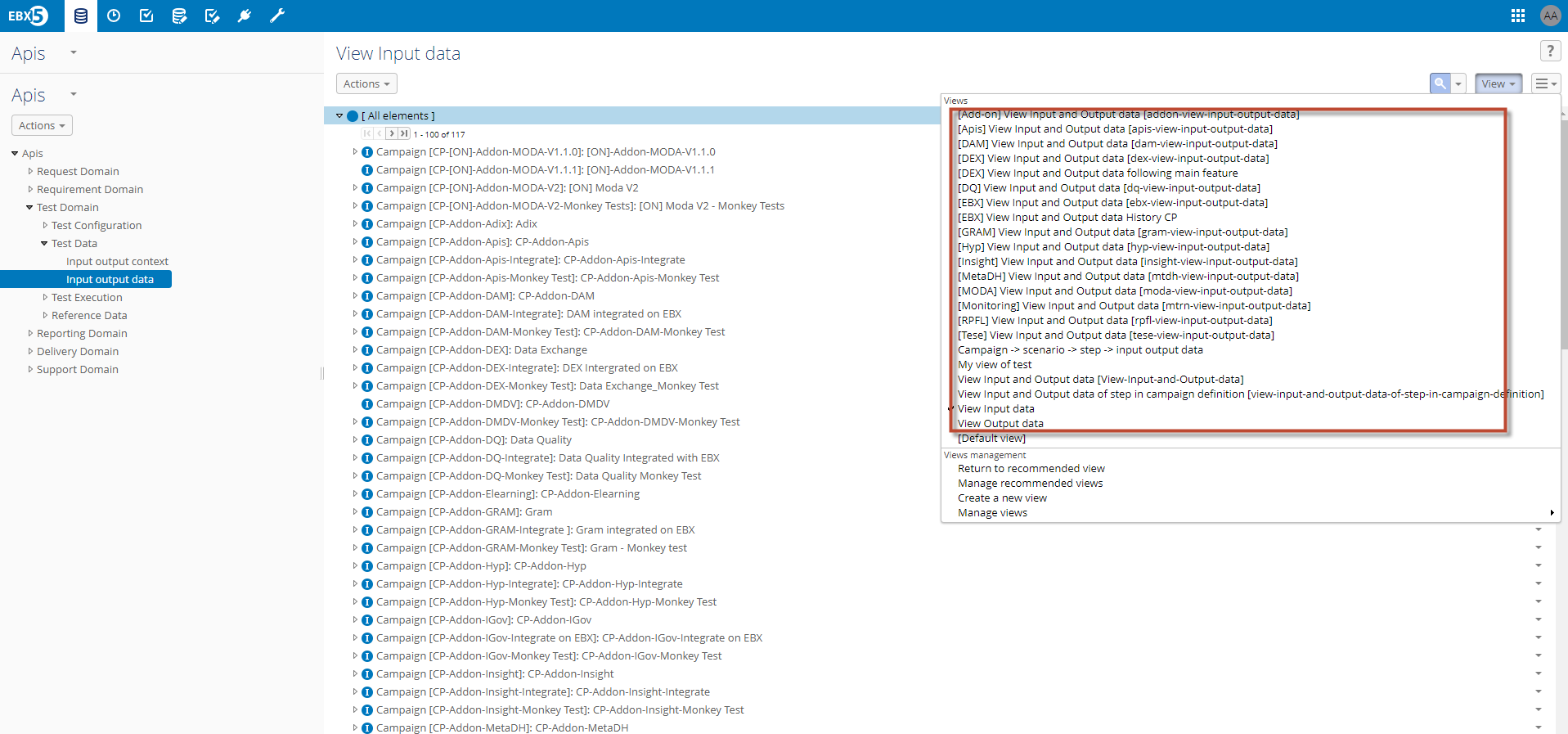


Figure 237: The list of hierarchy view in the **Input output data** table modify according to the Order field.

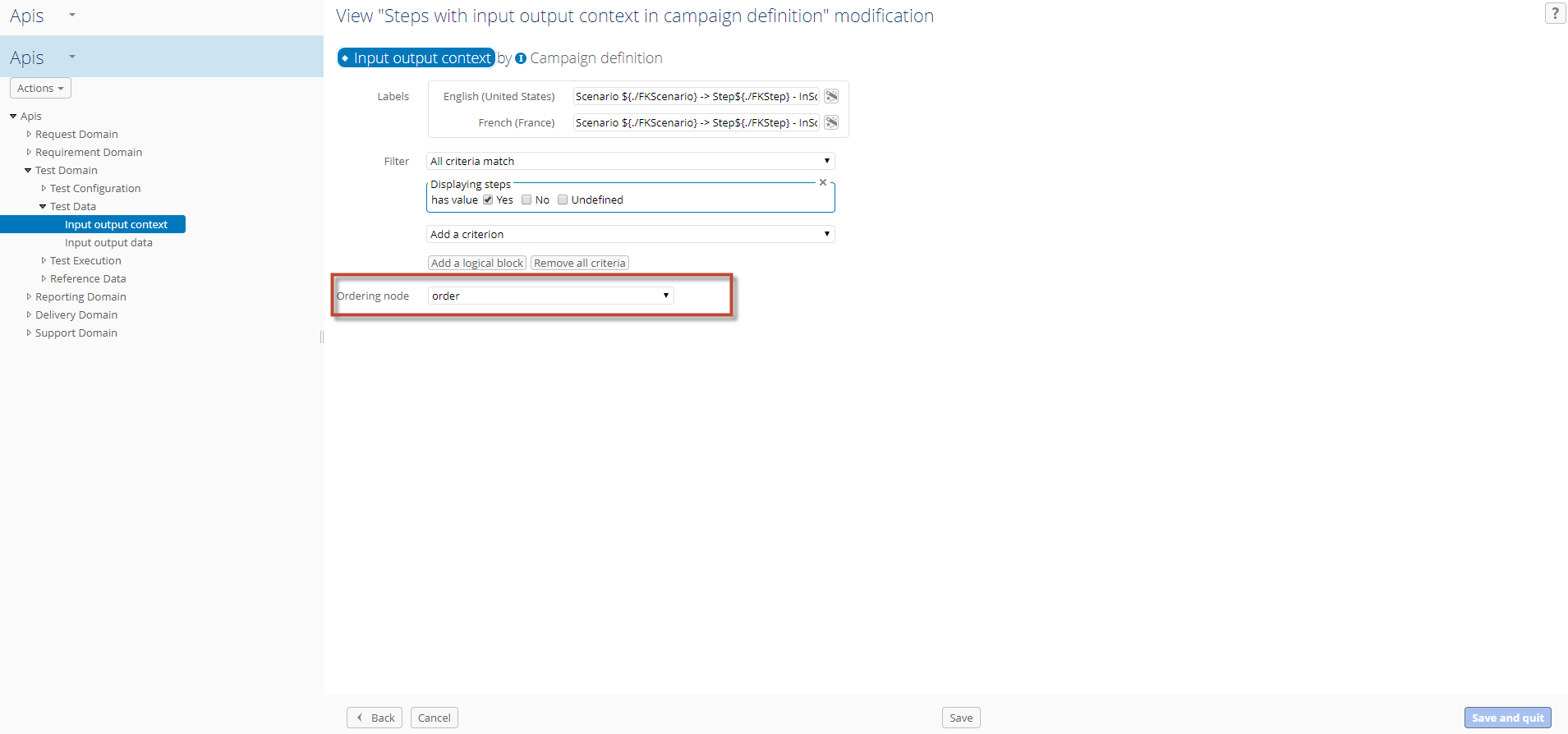


Figure 238: Changed by the order in the ordering node combo-box.

### Solution

#### Class diagram

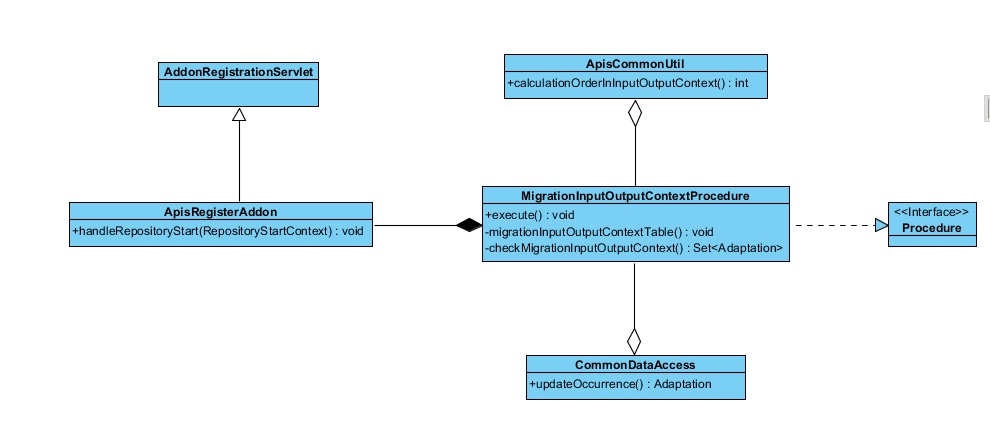


Figure 239: Class diagram for migration the order field.

#### Detail solution

#### Miscellaneous

### Limitations

* **Impacted bug in EBX**

**N/A**

* **Impacted evolution in EBX**

**N/A**

* **Functional limitation**

**N/A**

### Technical questions

* **Extension library**
* **Technical point**

## Req 07 – Remain filtering upon creating new record

### Description

### Analysis

### Solution

#### Class diagram

#### Detail solution

#### Miscellaneous

### Limitations

* **Impacted bug in EBX**

**N/A**

* **Impacted evolution in EBX**

**N/A**

* **Functional limitation**

**N/A**

### Technical questions

* **Extension library**

**N/A**

* **Technical point**

**N/A**

## Req 09 – It should be possible to search for data on all node levels in a data hierarchy view.

### Description

The objective of this requirement is allowing user search the fields on related tables with the **Detailed** **test result** table.

In the **Detailed test result** table, user can search on the field as following:

* Product build (Campaign execution level)
* Campaign definition (Campaign execution level)
* Scenario (Test report level)
* Test run date (Test report level)
* Test type (Test report level)
* Tester (Test report level)

### Analysis

Create new java class as named **DetailTestResultUITableFilter** under **com.orchestranetworks.addon.apis.ui.tablefilter** package to filter the records displayed in the **Detailed test result** table.

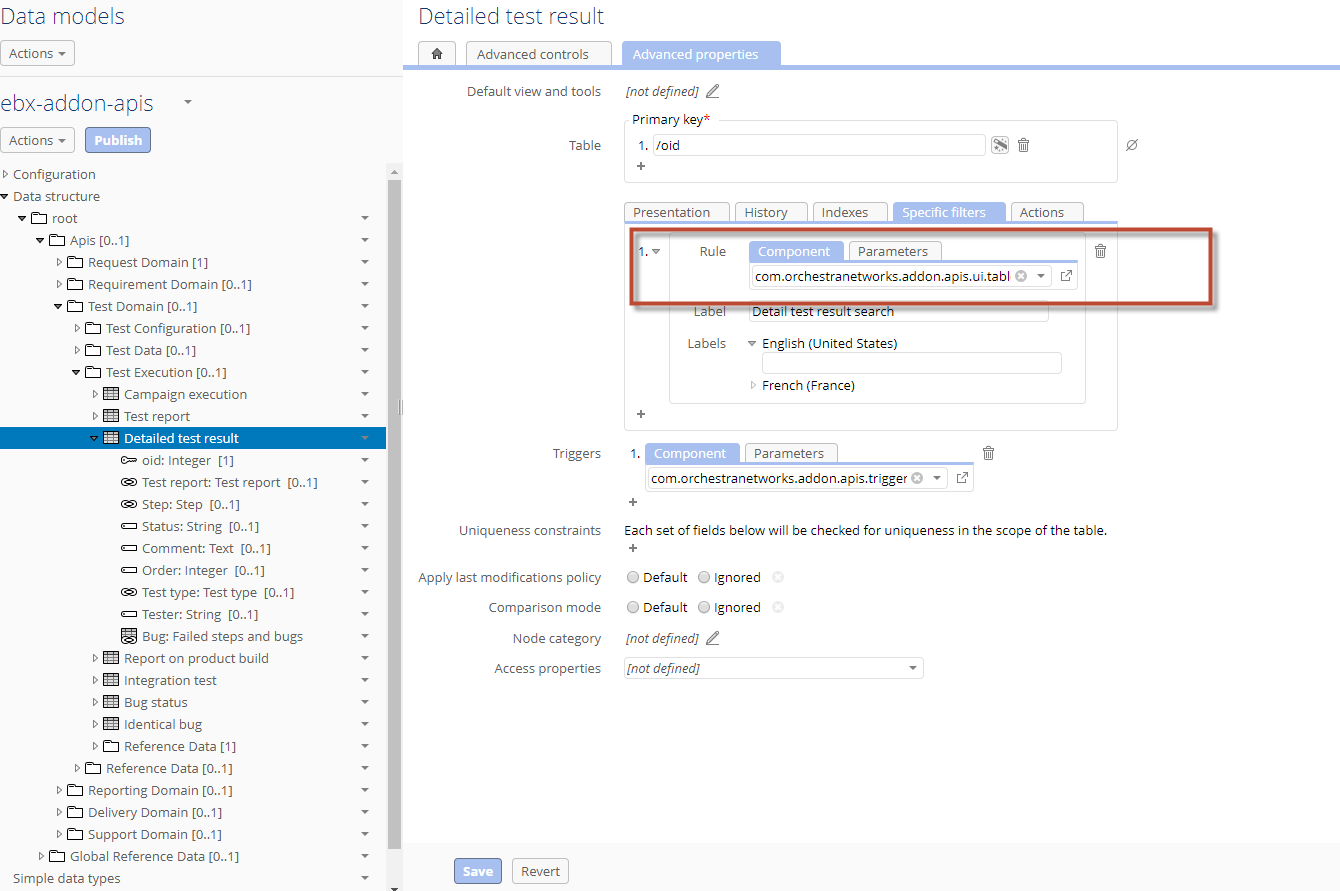


Figure 2.5.1: The **DetailTestResultUITableFilter** Java class to filter the records displayed in the **Detailed test result** table.

### Solution

#### Class diagram

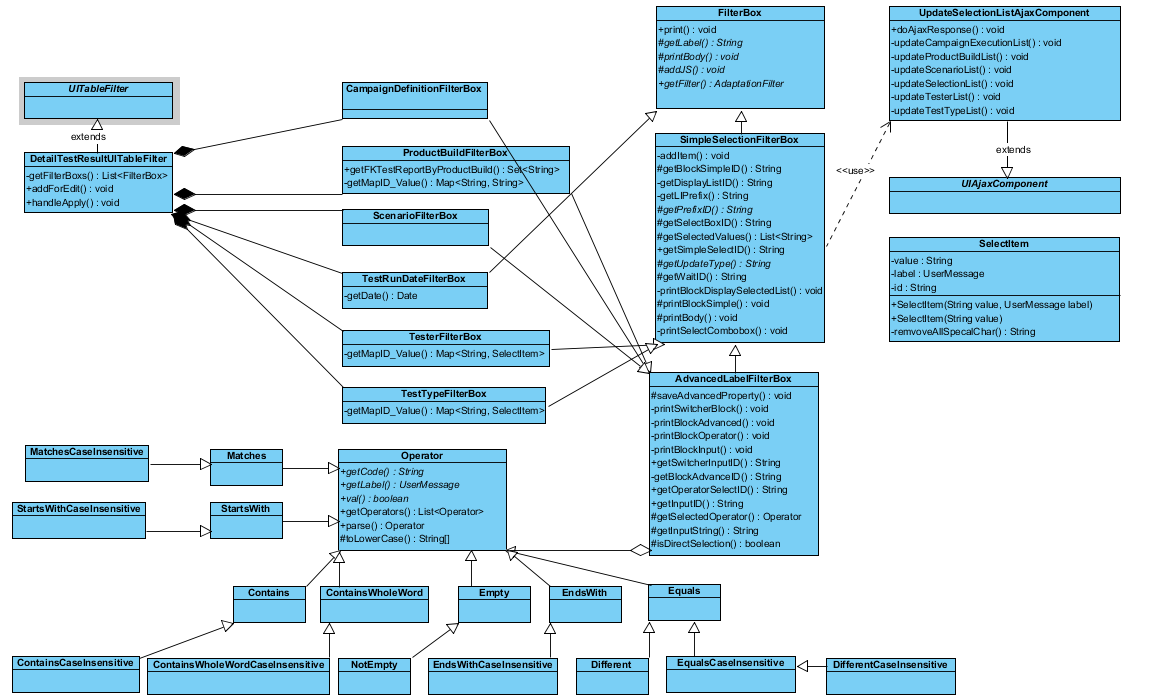


Figure 251: Class diagram for the filter on the Detailed test result table.

#### Sequence diagram

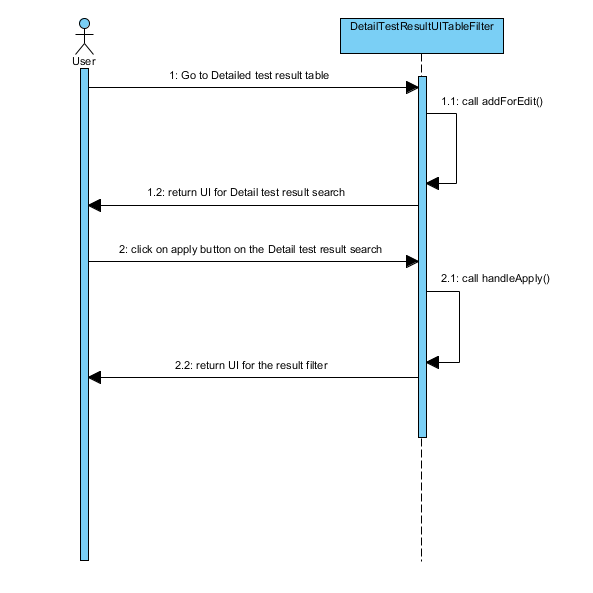


Figure 252: Sequence diagram

#### Miscellaneous

### Limitations

* **Impacted bug in EBX**
* **Impacted evolution in EBX**
* **Functional limitation**

### Technical questions

* **Extension library**
* **Technical point**
* Use UITableFilter API of EBX.

## Req 10 – The ability to support the agile process.

### Description

### Analysis

### Solution

#### Class diagram

#### Detail solution

#### Miscellaneous

### Limitations

* **Impacted bug in EBX**

**N/A**

* **Impacted evolution in EBX**

**N/A**

* **Functional limitation**

**N/A**

### Technical questions

* **Extension library**

**N/A**

* **Technical point**

**N/A**

## Req 11 – [Test] Add a new service to create report by 'Test result' in the 'Verify resolved mantis' table.

### Description

### Analysis

### Solution

#### Class diagram

#### Detail solution

#### Miscellaneous

### Limitations

* **Impacted bug in EBX**

**N/A**

* **Impacted evolution in EBX**

**N/A**

* **Functional limitation**

**N/A**

### Technical questions

* **Extension library**

**N/A**

* **Technical point**

**N/A**

# Logical data model

N/A

# API Diagram

N/A

# JUnit test

## Req 04: [Test] Add a table into 'Test execution' group in Test domain to store the check list for monkey test.

* Import free form case with fully valid values of the columns in ‘Free form case’ sheet.
* Import free form case with values of the columns in ‘Free form case’ sheet is not valid.

# Performance test

N/A

# Project plan

## Milestones

## Detailed planning