**Documentation For Automated Testing**

This module covers Automated Unit Testing of CRUD operation on Product and Order Entity, Importing the data using API REST Services, Exporting/Downloading the Product data to excel sheet and MX Reflection.

**Prerequisites**

* Mendix Studio Pro 9.12.4.
* Unit Testing Module from Market Place.
* Create domain models (see [Creating a Basic Data Layer](https://docs.mendix.com/refguide/create-a-basic-data-layer/))
* Create overview and detail pages (see [How to Create Your First Two Overview and Detail Pages](https://docs.mendix.com/howto/front-end/create-your-first-two-overview-and-detail-pages/))

1. **CRUD Operation**

On the Product Overview Page, Admin can perform CRUD operation on the Product and Order Entity. All the products are displayed on the product Page and Orders on the Order Page using data grid.

1. **Data Import Using API-Implementation**

API for testing and prototyping, to make a REST call to retrieve data of users in a JSON format and display it in the application.

* 1. In your module, create a microflow to retrieve data from a link using API.
  2. For the microflow that will have the logic to make the REST call to retrieve the data is created, we need to create a JSON structure that will indicate how we are going to be receiving the data from the API.
  3. Now, in [JSON Placeholder page](https://jsonplaceholder.typicode.com/), look for /users in resources and open it in a new tab in your browser. Copy the JSON that is displayed and then paste it in the JSON structure content and Format the content.
  4. Now in the created microflow add activity to call Rest services and set the link of data to the location. In the Response tab, choose Import from mapping and provide the variable.
  5. The final microflow will look like this.

**Graphical user interface, text, application, chat or text message

Description automatically generated**

* 1. Go to “Home\_Web” page, add a Data Grid widget in the Layout grid.
  2. Click on the Data Grid and go to Data Source and select microflow, then select the microflow that we created.
  3. Select Yes to automatically fill data and for the server-side paging.
  4. Save files and run locally.
  5. The final page will look like this.

**A screenshot of a computer

Description automatically generated with medium confidence**

1. **Automated Testing-**

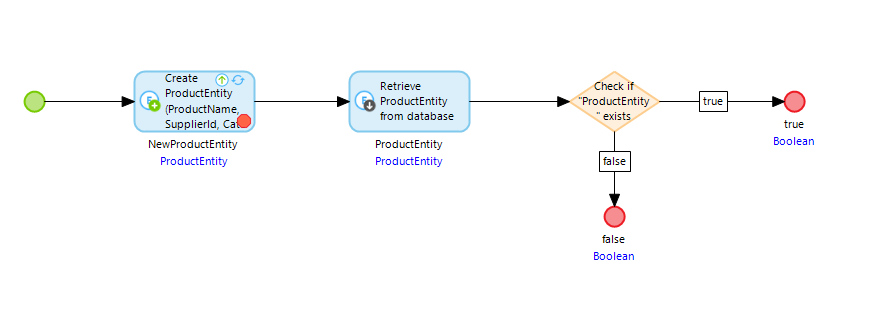
Import the Unit Testing module into your app. For more information, see [Use Marketplace Content in Studio Pro](https://docs.mendix.com/appstore/general/app-store-content/).

Add the **UnitTestOverview** microflow to your navigation structure, or include the **UnitTestOverview** snippet on a custom page.

To create a new microflow test in a module, just add a microflow with a name that starts with **TEST\_** or **UT\_** (case-insensitive). A test microflow should have no input arguments and either no result type, a Boolean result type, or a string result type.

* 1. **Test Create**-

Here we are creating new data and then checking it in the database if it is present or not. If created data is available, test case will be successful and fail otherwise.



**3.2) Test Delete** –

Here we must retrieve data (for example – through ProductId) from database and perform delete operation and then again checking the same ProductId is available in the database. If we are providing the same data from and after deleted in retrieve condition , it test case will pass and fail otherwise.

Diagram

Description automatically generated

* 1. **Test Update**-

Here we must make changes to the available data and then check if edited data is present or not in the table.

Diagram

Description automatically generated

* 1. **Test Validate-**

In this, we must check if all the required fields are filled by the user or not. If any mandatory field is empty, the test case will fail, and pass otherwise. Here we have made ProductId and ProductName as required.

Diagram

Description automatically generated

* 1. **Test Attribute Length**-

Here we need to create a Data Driven Test Case to Check if the given attribute length and the length of the data in each attribute retrieves from REST service is same. For example, we are checking here for Product Id, ProductName, Supplier Id and Category Id. If length of data is same, it will pass the test case otherwise fails.

Diagram

Description automatically generated

Graphical user interface

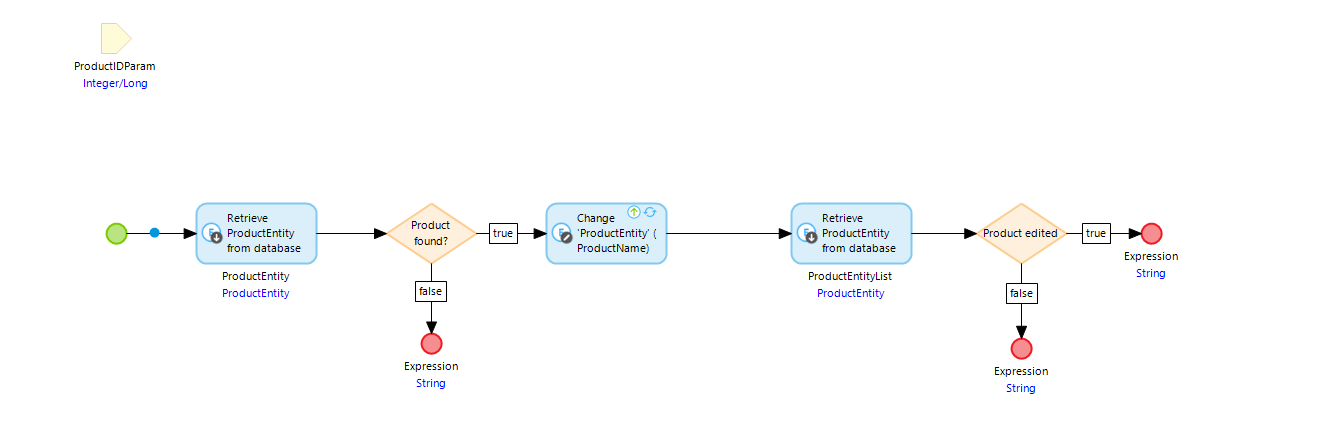
Description automatically generated with low confidence

Diagram

Description automatically generated

* 1. **Test Update Using User Input Parameter-**

In this test, we must take the user input for update. As early we were not taking the input parameter. We need to check if the update is working fine with this input or not. If it fails, it needs to display the error message. For this we are returning string type value for message.

****

**4) Unit Testing**-

To check for all the successful and failed test cases. Matching Values for the successful test case and unmatched value for failed test case.

Let’s take the example of Test Create.

* Double click the first activity of Test Create Microflow and provide new data to check if it will be created or not.

Graphical user interface, table

Description automatically generated

* In the next activity, provide the condition to set the created field. Here we have taken ProductId as required attribute so taking it into account.

Graphical user interface, text, application, email

Description automatically generated

* + Now in the condition, if condition fulfills, the test case will be successful and if condition false, it will return false, the test case will fail.

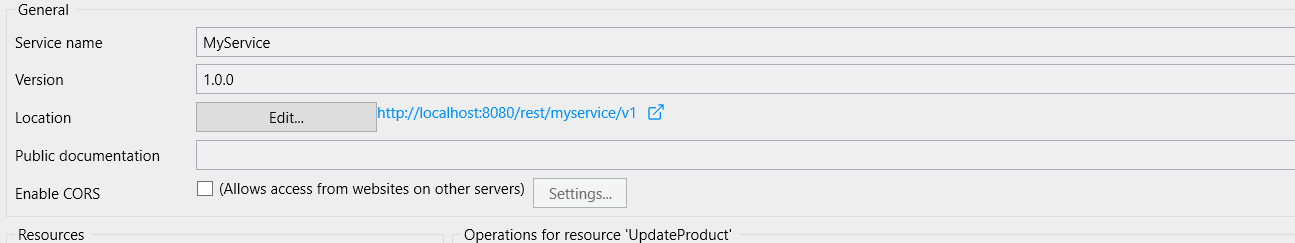
Graphical user interface, text, application

Description automatically generated

* In the same way, other tests can be verified.

Now let’s take one more example of **Update Test Using Input Parameter**.

* Go to My Services of your module. And click the blue link provided here.



* Enter any product Id and Execute.

Graphical user interface, application, Teams

Description automatically generated

* If that Product is available, it will be updated.

Background pattern

Description automatically generated

* Or if the product Id is not available in the database it will show the error message as product doesn’t exist.

Background pattern, rectangle

Description automatically generated