



# **Assignment**

**Task 2**

**API Automation: RestAssured**

**Language: Java**

**Maven**

**TestNG**

**Haseeb Saleem**

## Contents

|   |   |
|---|---|
| API Details: .....  | 3 |
| Test Cases .....  | 3 |
| Test Case 1: verifySuccessForValidRequest .....             | 3 |
| Test Case 2: verifyBadRequestForNullParameters .....        | 3 |
| Test Case 3: verifyBadRequestForInvalidSizeParameters ..... | 4 |
| Test Case 4: verifyObjectCountInResultsArray .....          | 4 |
| Test Case 5: verifyTotalAndPages .....                      | 4 |
| Test Case 6: verifyLocalizationUsingDifferentHeader .....   | 5 |
| Test Case 7: verifyPutApiMethodNotAllowedInGetContext ..... | 5 |

**This documentation provides a clear understanding of the purpose and functionality of each test case in your productApiTests class.**

## API Details:

- **Method:** GET
- **URL:** <https://api.qa.soum.sa/api-v2/rest/api/v1/product/?page%30number=2&size=2>
- **Headers:** Accept-Language : en

## Test Cases

### Test Case 1: verifySuccessForValidRequest

**Purpose:** This test case verifies the success response for a valid request to the product API.

#### Functionality:

- Sends a GET request to the product API with query parameters "page number" set to 1 and "size" set to the configured size of the product.
- Asserts that the response status code is 200.
- Validates the response body:
  - a. Checks that the "status" field is "success".
  - b. Ensures the "message" field is "Get Products successfully".
  - c. Verifies the existence of "total" and "pages" keys in "responseData.info".
  - d. Checks that the "results" array has more than 0 elements.
  - e. Validates the presence of product images and non-empty description for the first product in the response.

### Test Case 2: verifyBadRequestForNullParameters

**Purpose:** This test case verifies a bad request response when sending null parameters to the product API.

#### Functionality:

- Sends a GET request to the product API with empty "page number" and "size" parameters.
- Asserts that the response status code is 400.
- Validates the response body:
  - Checks that the "status" field is "fail".
  - Ensures the first violation has a code of 400 and a message of "Fail to get product with pagination".

### Test Case 3: `verifyBadRequestForInvalidSizeParameters`

**Purpose:** This test case verifies a bad request response when sending an invalid "size" parameter to the product API.

**Functionality:**

- Sends a GET request to the product API with a numeric "page number" and a non-numeric "size" parameter.
- Asserts that the response status code is 400.
- Validates the response body:
  - Checks that the "status" field is "fail".
  - Ensures the first violation has a code of 400 and a message of "Fail to get product with pagination".

### Test Case 4: `verifyObjectCountInResultsArray`

**Purpose:** This test case verifies that the count of objects in the "results" array matches the configured size of the product.

**Functionality:**

- Sends a GET request to the product API with "page number" set to 1 and "size" set to the configured size of the product.
- Asserts that the response status code is 200.
- Validates the response body:
  - a. Checks that the "message" field is "Get Products successfully".
  - b. Compares the size of the "responseData.results" array with the configured size of the product.

### Test Case 5: `verifyTotalAndPages`

**Purpose:** This test case verifies the accuracy of "total" and "pages" values in the response.

**Functionality:**

- Sends a GET request to the product API with "page number" set to 1 and "size" set to the configured size of the product.
- Extracts the response and retrieves "total" and "pages" values from "responseData.info".
- Calculates the expected "finalValue" based on the total and configured size of the product.
- Asserts that the "pages" value in the response matches the calculated "finalValue".

### Test Case 6: [verifyLocalizationUsingDifferentHeader](#)

**Purpose:** This test case verifies the localization of the response by using a different "Accept-Language" header.

**Functionality:**

- Sends a GET request to the product API with the "Accept-Language" header set to "ar" (Arabic).
- Asserts that the response status code is 200.
- Validates the response body:
  - Checks that the Arabic localized fields for the first product, "answer\_to\_questions\_ar" and "grade\_ar", are not empty.

### Test Case 7: [verifyPutApiMethodNotAllowedInGetContext](#)

**Purpose:** This test case verifies the handling of a PUT request in the context of the product API, expecting a failure response.

**Functionality:**

- Sends a PUT request to the product API with query parameters "page number" set to 1 and "size" set to the configured size of the product.
- Sets the "Accept-Language" header to "ar" to simulate a request in Arabic language.
- Asserts that the response status code is 400, indicating a Bad Request.
- Validates that the response body indicates a failure:
  - Checks that the "status" field is not "success" (indicating failure).