# **Web Application Test Scenarios**

## **positive tests**

Execute all test case using valid test data and parameter.

**Register new user account:**

1. Navigation the store home page by call (http://automationpractice.com/index.php).
2. Click on ‘sign in’ link from menu.
3. Fill Email address field with valid email address ([ahmed30@gmail.com](mailto:ahmed30@gmail.com))
4. Click on create an account button.
5. Fill ‘CREATE AN ACCOUNT’ form field with valid data.
6. Click on register button.
7. Should:
   1. redirect to ‘MY ACCOUNT’ page
   2. With display first and last name for registered user.

## **Negative tests**

Execute all test case using invalid test data or parameter

**Register new user account with invalid email:**

1. Navigation the store home page by call (http://automationpractice.com/index.php).
2. Click on sign in link from menu.
3. Fill Email address field with invalid email address ([ahmed30](mailto:ahmed30@gmail.com))
4. Click on create an account button.
5. Should:
   1. Display error message.

**Submit Empty user registration form:**

1. Navigation the store home page by call (http://automationpractice.com/index.php).
2. Click on sign in link from menu.
3. Fill Email address field with valid email address ([ahmed30@gmail.com](mailto:ahmed30@gmail.com))
4. Click on create an account button.
5. Click on register button with empty form field.
6. Should:
   1. Display error message.

**Register new user account with valid email and invalid user data:**

1. Navigation the store home page by call (http://automationpractice.com/index.php).
2. Click on sign in link from menu.
3. Fill Email address field with valid email address ([ahmed30@gmail.com](mailto:ahmed30@gmail.com))
4. Click on create an account button.
5. Fill CREATE AN ACCOUNT form field with invalid data.
6. Click on register button.
7. Should:
   1. Display error message.

# **API Test Scenarios**

## **positive tests**

Execute all test case by passing valid test data and parameter.

**Get authors list:**

* Call (/api/v1/Authors) Endpoint using get method.
* Check the response status code equal 200.

**Create new author:**

* Add request body as:

{

  "id": 1,

  "idBook": 0,

  "firstName": "ali",

  "lastName": "ahmed"

}

* Call ( /api/v1/Authors ) Endpoint using post method
* Check the response status code equal 200.
* Check the response body should be the same request body.

**Get author by author id:**

* Add author id as a parameter.
* Call (/api/v1/Authors/{id}) Endpoint using get method.
* Check the response status code equal 200.
* Check the response body should contain id, idBook, firstName, lastName properties.

**Update the author by author id:**

* Add request body as:

{

  "id": 1,

  "idBook": 0,

  "firstName": "ali",

  "lastName": "ahmed"

}

* Add author id as a parameter
* Call (/api/v1/Authors/{id}) Endpoint using PUT method
* Check the response status code equal 200.
* Check the response body should be the same request body.

**Delete the author by author id:**

* Add author id as a parameter
* Call ( /api/v1/Authors/{id}) Endpoint using PUT method
* Check the response status code equal 200.
* Check the response body should be the same request body.

**Get the author books by author id:**

* Add author id as a parameter
* Call (/api/v1/Authors/authors/books/{idBook}) Endpoint using Get method
* Check the response status code equal 200.
* Check the response body should have list of books.

## **Negative testing**

**Negative - Create new author with invalid data type of (**firstName and lastName)**:**

* Add request body as:

{

  "id": 1,

  "idBook": 0,

  "firstName": 1,

  "lastName": 1

}

* Call ( /api/v1/Authors ) Endpoint using post method
* Check the response status code equal 400.
* Check the response body should be the same request body.

**Negative - Get author by invalid type of author properties:**

* Add author id as a parameter has string type.
* Call (/api/v1/Authors/{id}) Endpoint using get method.
* Check the response status code equal 400.
* Check the response body should contain error properties.

**Negative - Update the author invalid type of author properties (**firstName and lastName)**:**

* Add request body as:

{

  "id": 1,

  "idBook": 0,

  "firstName": 1,

  "lastName": 1

}

* Add author id as a parameter
* Call ( /api/v1/Authors/10 ) Endpoint using PUT method
* Check the response:
  + Check the response status code equal 400.
  + Check the response body should contain error properties.

**Negative - Delete the author by invalid type of author properties (id):**

* Add author id as a parameter has string type
* Call ( /api/v1/Authors/10 ) Endpoint using PUT method
* Check the response status code equal 400.
* Check the response body should contain error properties.

**Negative - Get the author books by invalid type of author properties (id):**

* Add author id as a parameter has string type
* Call (/api/v1/Authors/authors/books/1) Endpoint using Get method
* Check the response status code equal 400.
* Check the response body should contain error properties.