**Automation Exercise API Test Cases**

**Test Case 1: Validate successful retrieval of all products**

**Given**

* The API URL is <https://automationexercise.com/api/productsList>
* The request method is GET

**When**

* The request is sent to fetch all products

**Then**

* The response code should be 200
* The response should contain a list of all products in JSON format
* The product list should not be empty
* The JSON structure should include fields like id, name, price, etc.

**Test Case 2: Validate response when using an unsupported POST method**

**Given**

* The API URL is <https://automationexercise.com/api/productsList>
* The request method is POST

**When**

* The request is sent to add a product to the products list

**Then**

* The response code should be 405
* The response message should be "This request method is not supported.”

**Test Case 3: Validate successful retrieval of all brands**

**Given**

* The API URL is <https://automationexercise.com/api/brandsList>
* The request method is GET

**When**

* The request is sent to fetch all brands

**Then**

* The response code should be 200
* The response should contain a list of all brands in JSON format
* The brands list should not be empty
* The JSON structure should include fields like id, brand-named

**Test Case 4: Validate response when using an unsupported PUT method**

**Given**

* The API URL is <https://automationexercise.com/api/brandsList>
* The request method is PUT

**When**

* The request is sent to update the brands list

**Then**

* The response code should be 405
* The response message should be "This request method is not supported."

**Test Case 5: Validate successful search for a product**

**Given**

* The API URL is <https://automationexercise.com/api/searchProduct>
* The request method is POST
* The request parameter search\_product is set to a valid product name (e.g., tshirt)

**When**

* The request is sent to search for the product tshirt

**Then**

* The response code should be 200
* The response should return a JSON list of products matching the search term tshirt
* The JSON structure should include fields like id, name, price, and category
* The product list should not be empty

**Test Case 6: Validate response when search\_product parameter is missing**

**Given**

* The API URL is <https://automationexercise.com/api/searchProduct>
* The request method is POST
* The request does not include the search\_product parameter

**When**

* The request is sent to search for a product

**Then**

* The response code should be 400
* The response message should be "Bad request, search\_product parameter is missing in POST request."
* The request should not proceed without the search\_product parameter

**Test Case 7: Validate successful login with valid credentials**

**Given**

* The API URL is <https://automationexercise.com/api/verifyLogin>
* The request method is POST
* The request parameters are valid:
* email: [testTesT123@gmail.com](mailto:testTesT123@gmail.com)
* password: 12345678

**When**

* The request is sent to verify the login

**Then**

* The response code should be 200
* The response message should be "User exists!"
* The response should include user details (if applicable), confirming the user has been authenticated

**Test Case 8: Validate response when email parameter is missing**

**Given**

* The API URL is <https://automationexercise.com/api/verifyLogin>
* The request method is POST
* The request parameters include only password:
* Password: 12345678

**When**

* The request is sent to verify the login

**Then**

* The response code should be 400
* The response message should be "Bad request, email or password parameter is missing in POST request."

**Test Case 9: Validate response for unsupported DELETE request**

**Given**

* The API URL is <https://automationexercise.com/api/verifyLogin>
* The request method is DELETE

**When**

* The request is sent to verify the login

**Then**

* The response code should be 405
* The response message should be "This request method is not supported."

**Test Case 10: Validate successful user account creation with valid parameters**

**Given**

* The API URL is <https://automationexercise.com/api/createAccount>
* The request method is POST
* The request parameters are:
* name: Mariam
* email: testTesT123@gmail.com
* password: 12345678
* birth\_date: 4
* birth\_month: October
* birth\_year: 1986
* firstname: Mariam
* lastname: Beshara
* company: DEPI
* address1: Alex
* address2: Alex
* country: Canda
* zipcode: 1234569
* state: Alex
* city: Alex
* mobile\_number: 123654855

**When**

* The request is sent to create a user account

**Then**

* The response code should be 201
* The response message should be "User created"

**Test Case 11: Validate response when trying to create a user account with an existing email**

**Given**

* The API URL is <https://automationexercise.com/api/createAccount>
* The request method is POST
* The request parameters include an email that already exists in the system:
* name: Mariam
* email: testTesT123@gmail.com
* password: 12345678
* birth\_date: 4
* birth\_month: October
* birth\_year: 1986
* firstname: Mariam
* lastname: Beshara
* company: DEPI
* address1: Alex
* address2: Alex
* country: Canda
* zipcode: 1234569
* state: Alex
* city: Alex
* mobile\_number: 123654855

**When**

* The request is sent to create a user account

**Then**

* The response code should be 400 *(Assuming a conflict response for existing user)*
* The response message should indicate that the email is already in use, "Email already exists."

**Test Case 12: Validate response when trying to verify login with valid details**

**Given**

* The API URL is <https://automationexercise.com/api/verifyLogin>
* The request method is POST
* The request parameters include valid login details:

email: testTesT123@gmail.com

password: 12345678

**When**

* The request is sent to verify the login

**Then**

* The response code should be 200
* The response message should be "User exists!"

**Test Case 13: Validate response when updating a user account**

**Given**

* The API URL is <https://automationexercise.com/api/updateAccount>
* The request method is PUT
* The request parameters include the following user account details:
* name: Mariam2
* email: testTesT123@gmail.com
* password: 12345678
* birth\_date: 5
* birth\_month: October
* birth\_year: 1986
* firstname: Mariam
* lastname: Beshara
* company: DEPI
* address1: Alex
* address2: Alex
* country: Canda
* zipcode: 1234569
* state: Alex
* city: Alex
* mobile\_number: 123654855

**When**

* The request is sent to update the user account

**Then**

* The response code should be 200
* The response message should be "User updated!"

**Test Case 14 : Validate response when retrieving user account details by email**

**Given**

* The API URL is [https://automationexercise.com/api/ getUserDetailByEmail?email= testTesT123@gmail.com](https://automationexercise.com/api/%20getUserDetailByEmail?email=%20testTesT123@gmail.com)
* The request method is GET
* The request parameter is the user's email:
* email: testTesT123@gmail.com

**When**

* The request is sent to retrieve the user account details

**Then**

* The response code should be 200
* The response JSON should contain the user's details (such as name, email, etc.)

**Test Case 15: Validate response when deleting a user account**

**Given**

* The API URL is <https://automationexercise.com/api/deleteAccount>
* The request method is DELETE
* The request parameters include:
* email: testTesT123@gmail.com
* password: 12345678

**When**

* The request is sent to delete the user account

**Then**

* The response code should be 200
* The response message should be "Account deleted!"