

Source Code- Cucumber-Feature-Project:

CreateShoe.feature

@tag

Feature: Create a new shoe in the store

I want to test the functionality of creating a new shoe

@tag1

Scenario: Create a new shoe on the website

Given Open the API base URL

And Enter the shoe details

When Hit the POST request to create a new shoe

Then The response code should be 200

And Print the result of adding new shoe

DeleteProduct.feature

@tag

Feature: Delete a shoe from the store

I want to test the functionality of deleting a shoe from the store

@tag1

Scenario: Delete a shoe from the website

Given Open the API base URL

And there is an existing shoe with ID 101

When hit the DELETE request for the shoe with ID 101

Then the response code should be 200

And the response should indicate successful deletion

GetProducts.feature

Feature: Retrieve all products from the sports shoe store

As a user,

I want to retrieve the list of all available shoes from the sports shoe store,

So that I can browse and choose the desired products.

@tag @GetProducts

Scenario: Retrieve all products successfully

Given I am interested in viewing the available shoes

```
When I send a request to retrieve the list of products
Then I should receive a successful response with status code 200
And the response should contain a valid list of sports shoes
```

GetUsers.feature

```
Feature: Retrieve all registered users
  I want to retrieve the list of all available register users

  @tag @GetProducts
  Scenario: Retrieve all registered users
    Given I am interested in viewing the registered users
    When I send a request to retrieve the users
    Then I want to receive a successful response with status code 200
    And the response should contain a valid list of users
```

UpdateProduct.feature

```
@tag
Feature: Update a shoe in the store
  I want to test the functionality of updating a shoe in the store

  @tag1 @UpdateRequest
  Scenario: Update a shoe on the website
    Given open the update product base url
    When hit the PUT request to update the shoe with ID 101
    Then the response code should be equal to 200
    And the response should indicate successful update
```

DeleteProductSteps.java

```
package com.simplilearn.cucumber.stepdefinitions;

import static org.hamcrest.CoreMatchers.containsString;

import io.cucumber.java.en.Given;
```

```
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import io.restassured.RestAssured;
import io.restassured.response.Response;
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;
```

```
public class DeleteProductSteps {

    private Response response;

    private ValidatableResponse json;

    private RequestSpecification request;

    private String BASE_URL = "http://localhost:9010";

    @Given("there is an existing shoe with ID {int}")
    public void there_is_an_existing_shoe_with_id(Integer int1) {
        request=RestAssured.given().baseUrl(BASE_URL);
    }

    @When("hit the DELETE request for the shoe with ID {int}")
    public void hit_the_delete_request_for_the_shoe_with_id(Integer shoeld) {
        response=request.delete("/delete-shoe?id="+shoeld);
    }

    @Then("the response code should be {int}")
    public void the_response_code_should_be(Integer code) {
        json=response.then().statusCode(code);
    }
}
```

```
@Then("the response should indicate successful deletion")
public void the_response_should_indicate_successful_deletion() {
    response.then().body("message", containsString("Shoe with ID 101 Deleted Successfully"));
}

}
```

GetRequestSteps.java

```
package com.simplilearn.cucumber.stepdefinitions;

import org.hamcrest.Matchers;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import io.restassured.RestAssured;
import io.restassured.response.Response;
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;

public class GetRequestSteps {

    private Response response;
    private ValidatableResponse json;
```

```
private RequestSpecification request;

private String BASE_URL = "http://localhost:9010";

@Given("I am interested in viewing the available shoes")
public void i_am_interested_in_viewing_the_available_shoes() {
    request=RestAssured.given().baseUrl(BASE_URL);
}

@When("I send a request to retrieve the list of products")
public void i_send_a_request_to_retrieve_the_list_of_products() {
    response=request.when().get("/get-shoes");
}

@Then("I should receive a successful response with status code {int}")
public void i_should_receive_a_successful_response_with_status_code(Integer
code) {
    json=response.then().statusCode(code);
    System.out.println(json.toString());
}

@Then("the response should contain a valid list of sports shoes")
public void the_response_should_contain_a_valid_list_of_sports_shoes() {
    response.then().body("shoes", Matchers.notNullValue());
}

}
```

GetUserSteps.java

```
package com.simplilearn.cucumber.stepdefinitions;

import org.hamcrest.Matchers;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import io.restassured.RestAssured;
import io.restassured.response.Response;
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;

public class GetUserSteps {

    private Response response;
    private ValidatableResponse json;
    private RequestSpecification request;

    private String BASE_URL = "http://localhost:9010";

    @Given("I am interested in viewing the registered users")
    public void i_am_interested_in_viewing_the_registered_users() {
        request=RestAssured.given().baseUrl(BASE_URL);
    }

    @When("I send a request to retrieve the users")
```

```

    public void i_send_a_request_to_retrieve_the_users() {
        response=request.when().get("/get-users");
    }

    @Then("I want to receive a successful response with status code {int}")
    public void i_want_to_receive_a_successful_response_with_status_code(Integer
code) {
        json=response.then().statusCode(code);
        System.out.println(json.toString());
    }

    @Then("the response should contain a valid list of users")
    public void the_response_should_contain_a_valid_list_of_users() {
        response.then().body("users", Matchers.notNullValue());
    }
}

```

PostRequestSteps.java

```

package com.simplilearn.cucumber.stepdefinitions;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import io.restassured.RestAssured;
import io.restassured.response.Response;

```

```
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;

public class PostRequestSteps {

    private Response response;
    private ValidatableResponse json;
    private RequestSpecification request;
    private String requestBody;

    private String BASE_URL = "http://localhost:9010";

    @Given("Open the API base URL")
    public void open_the_api_base_url() {
        request=RestAssured.given().baseUrl(BASE_URL);
    }

    @Given("Enter the shoe details")
    public void enter_the_shoe_details() {
        requestBody = "id=" + 101 + "&image=image_url&name=Sample  
Shoes&category=Sports Shoe&sizes=9&price=1999";
    }

    @When("Hit the POST request to create a new shoe")
    public void hit_the_post_request_to_create_a_new_shoe() {
        int id = 101;
        String image_url = "image_url";
        String name = "sampleShoe";
        String category = "Runnings";
    }
}
```



```

int sizes = 9;
int price = 1000;

    response=request.queryParam("id",id).queryParam("image",image_url)
    .queryParam("name",name).queryParam("category",category)
    .queryParam("sizes",sizes).queryParam("price",price).when().post("/add-shoe");
}

@Then("The response code should be {int}")
public void the_response_code_should_be(Integer code) {
    json=response.then().statusCode(code);
    System.out.println(json);
}

@Then("Print the result of adding new shoe")
public void print_the_result_of_adding_new_shoe() {
    System.out.println(response.getBody().asString());
}
}

```

UpdateshoeSteps.java

```
package com.simplilearn.cucumber.stepdefinitions;
```

```
import org.hamcrest.Matchers;
```

```
import io.cucumber.java.en.Given;
```

```
import io.cucumber.java.en.Then;
```

```
import io.cucumber.java.en.When;
```

```

import io.restassured.RestAssured;
import io.restassured.response.Response;
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;

public class UpdateshoeSteps {

    private Response response;
    private ValidatableResponse json;
    private RequestSpecification request;

    private String BASE_URL = "http://localhost:9010";

    @Given("open the update product base url")
    public void open_the_update_product_base_url() {
        request=RestAssured.given().baseUrl(BASE_URL);
    }

    @When("hit the PUT request to update the shoe with ID {int}")
    public void hit_the_put_request_to_update_the_shoe_with_id(Integer shoeld) {

        int id = shoeld;

        String image= "update image";
        String name = "updated Shoe";
        String category = " updated Runnings";
        int sizes = 12;
        int price = 1099;
    }

```

```

        response =request
            .queryParams("id",id).queryParams("image",image)
            .queryParams("name",name).queryParams("category",category)
            .queryParams("sizes",sizes).queryParams("price",price)
            .when().put( "/update-shoe?id=" );
    }

    @Then("the response code should be equal to {int}")
    public void the_response_code_should_be_equal_to(Integer expectedStatusCode)
    {
        json=response.then().statusCode(expectedStatusCode);
    }

    @Then("the response should indicate successful update")
    public void the_response_should_indicate_successful_update() {
        System.out.println(json.toString());
    }
}

```

TestRunner.java

```

package com.simplilearn.cucumber.testrunner;

import org.junit.runner.RunWith;

import io.cucumber.junit.Cucumber;
import io.cucumber.junit.CucumberOptions;

@RunWith(Cucumber.class)

```

```
@CucumberOptions(features = "src/test/resources/Features", glue =
{"com/simplilearn/cucumber/stepdefinitions" })
```

```
public class TestRunner {
```

```
}
```

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.simplilearn.cucumber</groupId>
  <artifactId>PhaseEnd_Cucumber_Restassured_Project</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>PhaseEnd_Cucumber_Restassured_Project</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.target>8</maven.compiler.target>
    <maven.compiler.source>8</maven.compiler.source>
    <rest.assured.version>5.4.0</rest.assured.version>
    <junit.version>4.13.2</junit.version>
    <cucumber.version>7.15.0</cucumber.version>
  </properties>

  <dependencies>
    <!-- rest-assured -->
    <dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>rest-assured</artifactId>
      <version>${rest.assured.version}</version>
      <scope>test</scope>
    </dependency>

    <!-- rest-assured/json-path -->
    <dependency>
      <groupId>io.rest-assured</groupId>
      <artifactId>json-path</artifactId>
      <version>${rest.assured.version}</version>
      <scope>test</scope>
    </dependency>

    <!-- rest-assured/json-schema-validator -->
    <dependency>
```

```

        <groupId>io.rest-assured</groupId>
        <artifactId>json-schema-validator</artifactId>
        <version>${rest.assured.version}</version>
    </dependency>

    <!-- io.rest-assured/xml-path -->
    <dependency>
        <groupId>io.rest-assured</groupId>
        <artifactId>xml-path</artifactId>
        <version>${rest.assured.version}</version>
    </dependency>

    <!-- cucumber-java -->
    <dependency>
        <groupId>io.cucumber</groupId>
        <artifactId>cucumber-java</artifactId>
        <version>${cucumber.version}</version>
    </dependency>

    <!-- cucumber-junit :: junit4-->
    <dependency>
        <groupId>io.cucumber</groupId>
        <artifactId>cucumber-junit</artifactId>
        <version>${cucumber.version}</version>
        <scope>test</scope>
    </dependency>

    <!-- junit4 -->
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>${junit.version}</version>
        <scope>test</scope>
    </dependency>

</dependencies>
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-surefire-plugin</artifactId>
            <version>3.2.3</version>
        </plugin>
    </plugins>
</build>
</project>

```
