SourceCode-Cucumber-RestAssured-Project

AddProductSteps.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.http.ContentType;
import io.restassured.response.Response;
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;

public class AddProductSteps {


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

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

	@Given("the user is trying to create a product")
	public void the_user_is_trying_to_create_a_product() {
	   request=RestAssured.given().baseUri(BASE_URL);
	}

	@When("the user  is hit the create product url")
	public void the_user_is_hit_the_create_product_url() {
```

```
		String requestBody="{\r\n"
				+ "	\"id\": 999,\r\n"
				+ "	\"image\": \"1.png\",\r\n"
				+ "	\"name\": \"Disprin\",\r\n"
				+ "	\"category\": \"medicine\",\r\n"
				+ "	\"brand\": \"BZ Medico\",\r\n"
				+ "	\"status\": 1,\r\n"
				+ "	\"price\": 100\r\n"
				+ "}\r\n";

response=request.when().contentType(ContentType.JSON).body(requestBody).post("/add-product");
	}

	@Then("the system should confirm the successful creation of the medicine product")
	public void the_system_should_confirm_the_successful_creation_of_the_medicine_product() {
	   json=response.then().assertThat().statusCode(200);
	}

	@Then("check additional outcomes related to product creation")
	public void check_additional_outcomes_related_to_product_creation() {
	   System.out.println(response.getBody().asString());
	}
}
```

DeleteMedicineSteps.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 DeleteMedicineSteps {


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

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

	@Given("the user is logged into the product management system")
	public void the_user_is_logged_into_the_product_management_system() {
	   request=RestAssured.given().baseUri(BASE_URL);
	}

	@When("there is an existing medicine product to be deleted")
	public void there_is_an_existing_medicine_product_to_be_deleted() {
	   response=request.delete("/delete-product?id=999");
```

```
    }

    @Then("the system should confirm the successful deletion of the medicine product")
    public void the_system_should_confirm_the_successful_deletion_of_the_medicine_product() {
        json=response.then().assertThat().statusCode(200);
    }

    @Then("the user can verify that the product is no longer available")
    public void the_user_can_verify_that_the_product_is_no_longer_available() {
        System.out.println(response.getBody().asString());
    }

}
```

RetrieveMedicineSteps.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 RetrieveMedicineSteps {
```

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

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

    @Given("the user is on the Medicare products page")
    public void the_user_is_on_the_medicare_products_page() {
       request=RestAssured.given().baseUri(BASE_URL);
    }

    @When("the user clicks on the {string} button")
    public void the_user_clicks_on_the_button(String string) {
       response=request.when().get("/get-products");
    }

    @Then("all medicines should be displayed")
    public void all_medicines_should_be_displayed() {
  json=response.then().statusCode(200);
    }

    @Then("the user can check more details about each medicine")
    public void the_user_can_check_more_details_about_each_medicine() {
      System.out.println(response.getBody().asString());
    }

}
```

RetrieveUsersSteps.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 RetrieveUserSteps {

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

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

	@Given("the user has logged into the Medicare system")
	public void the_user_has_logged_into_the_medicare_system() {
	   request=RestAssured.given().baseUri(BASE_URL);
	}

	@When("the user initiates the {string} action")
	public void the_user_initiates_the_action(String string) {
	   response=request.get("/get-users");
```

```
    }

    @Then("the system should return a list of all users")
    public void the_system_should_return_a_list_of_all_users() {
       json=response.then().statusCode(200);
    }

    @Then("the user can validate the details of each retrieved user")
    public void the_user_can_validate_the_details_of_each_retrieved_user() {
       System.out.println(response.getBody().asPrettyString());
    }

}
```

UpdateMedicineSteps.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.http.ContentType;
import io.restassured.response.Response;
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;

public class UpdateMedicineSteps {
```

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

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

	@Given("the user is on the update product page")
	public void the_user_is_on_the_update_product_page() {
	   request=RestAssured.given().baseUri(BASE_URL);
	}

	@When("modifies the details of the medicine product")
	public void modifies_the_details_of_the_medicine_product() {
		String requestBody="{\r\n"
				+ "	\"id\": 999,\r\n"
				+ "	\"image\": \"2.png\",\r\n"
				+ "	\"name\": \"Disprin+\",\r\n"
				+ "	\"category\": \"medicine\",\r\n"
				+ "	\"brand\": \"BZ Medico\",\r\n"
				+ "	\"status\": 1,\r\n"
				+ "	\"price\": 120\r\n"
				+ "}\r\n"
				+ "";

response=request.contentType(ContentType.JSON).body(requestBody).when().put("/updat
e-product");
	}
```

```
    @Then("the system should confirm the successful update of the medicine product")
    public void
the_system_should_confirm_the_successful_update_of_the_medicine_product() {
       json=response.then().assertThat().statusCode(200);
    }

    @Then("check additional outcomes related to product updating")
    public void check_additional_outcomes_related_to_product_updating() {
       System.out.println(response.getBody().asString());
    }

}
```

UpdateStatusSteps.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.http.ContentType;
import io.restassured.response.Response;
import io.restassured.response.ValidatableResponse;
import io.restassured.specification.RequestSpecification;

public class UpdateStatusSteps {
```

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

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


	@Given("there is an existing medicine product")
	public void there_is_an_existing_medicine_product() {
	   request=RestAssured.given().baseUri(BASE_URL);
	}

	@When("selects a new status for the medicine")
	public void selects_a_new_status_for_the_medicine() {
		String requestBody="{\r\n"
				+ "	\"id\": 999,\r\n"
				+ "	\"image\": \"2.png\",\r\n"
				+ "	\"name\": \"Disprin+\",\r\n"
				+ "	\"category\": \"medicine\",\r\n"
				+ "	\"brand\": \"BZ Medico\",\r\n"
				+ "	\"status\": 0,\r\n"
				+ "	\"price\": 120\r\n"
				+ "}\r\n"
				+ "";

response=request.contentType(ContentType.JSON).body(requestBody).put("/update-product-status");
	}
```

```
	@Then("the system should confirm the successful update of the medicine status")
	public void
the_system_should_confirm_the_successful_update_of_the_medicine_status() {
	  json=response.then().assertThat().statusCode(200);
	}

	@Then("the user can validate the updated status")
	public void the_user_can_validate_the_updated_status() {
	   System.out.println(response.getBody().asString());
	}
}
```

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 {

}
```

AddProduct.feature

```
@tag
Feature: Create a product
  I want to test the functionality of creating a product

  @tag1 @Add
  Scenario: Create a Medicine Product
    Given the user is trying to create a product
    When the user  is hit the create product url
    Then the system should confirm the successful creation of the medicine
product
    And check additional outcomes related to product creation
```

DeleteMedicine.feature

```
@tag
Feature: Delete the medicine
  I want to test the functionality of deleting a medicine

  @tag1 @DeleteMedicine
  Scenario: Delete the product
    Given the user is logged into the product management system
    When there is an existing medicine product to be deleted
    Then the system should confirm the successful deletion of the medicine
product
    And the user can verify that the product is no longer available
```

RetrieveProduct.feature

```
@tag
Feature: Retrieve All products from the Medicare
  I want to test the functionality of retrieving all products

  @tag1  @RetrieveProduct
  Scenario: Retrieve all medicines
    Given the user is on the Medicare products page
    When the user clicks on the "Retrieve All" button
    Then all medicines should be displayed
    And the user can check more details about each medicine
```

RetreiveUsers.feature

```
@tag
Feature: Retrieve all users from Medicare
  I want to test the functionality of getting users.

  @tag1  @RetrieveUsers
  Scenario: Retrieve all users
    Given the user has logged into the Medicare system
    When the user initiates the "Retrieve All Users" action
    Then the system should return a list of all users
    And the user can validate the details of each retrieved user
```

UpdateMedicine.feature

```
@tag
Feature: Update the medicine product
  I want to test the functionality of updating a medicine

  @tag1 @UpdateMedicine
  Scenario: Update medicine in the store
    Given the user is on the update product page
    When modifies the details of the medicine product
    Then the system should confirm the successful update of the medicine
product
    And check additional outcomes related to product updating
```

UpdateStatus.feature

```
@tag
Feature: Update the product status
  I want to test the functionality of updating medicine status

  @tag1
  Scenario: Update the status of medicine
    Given there is an existing medicine product
    When selects a new status for the medicine
    Then the system should confirm the successful update of the medicine
status
    And the user can validate the updated status
```

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>ATE-Capstone_Cucumber_Project</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>ATE-Capstone_Cucumber_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>
```
