1. Automate the below API endpoints using Rest-Assured
    - Retrieve the list of all products in the store.
    - Retrieve the list of all registered users.
    - Add the product.
    - Delete the product.
    - Update the product.

```
package com.sportyshoe.restAssuredScripts;

import org.testng.annotations.Test;

import io.restassured.RestAssured;


public class GetAllShoes {
	@Test (priority='1')

	public void get_all_shoes()
	{

		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/get-shoes")
		.when()
		.get()
		.then()
		.statusCode(200)
		.log()
		.all();

	}


	@Test(priority='2')

	public void get_all_users()
	{

		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/get-users")
		.when()
		.get()
		.then()
		.statusCode(200)
		.log()
		.all();

	}


}
```

package com.sportyshoe.restAssuredScripts;

import org.testng.annotations.Test;

```
import io.restassured.RestAssured;

public class PostandPutnewshoe {
	@Test(priority='1')
	public void add_new_product()
	{

		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/add-shoe")
		.queryParam("id","1020")
		.queryParam("image", "www.imge.com")
		.queryParam("name","Nike")
		.queryParam("category", "Running")
		.queryParam("sizes","5,6,7")
		.queryParam("price", "2000")
		.when()
		.post()
		.then()
		.log().all();

}


	@Test(priority='2')
	public void update_a_product()
	{

		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/update-shoe")
		.queryParam("id","1020")
		.queryParam("image", "www.imge123.com")
		.queryParam("name","Reebok")
		.queryParam("category", "Running")
		.queryParam("sizes","5,6,7")
		.queryParam("price", "2500")
		.when()
		.put()
		.then()
		.log().all();

}

}

package com.sportyshoe.restAssuredScripts;

import org.testng.annotations.Test;
```

```
import io.restassured.RestAssured;

public class Delete {
	@Test(priority='1')
	public void delete_product()
	{

		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/delete-shoe")
		.queryParam("id", "1010")
		.when().delete()
		.then().log().all();

	}

}
```

2. Create Selenium scripts using TestNG to test all the pages in the web app that will automate:
    - Login page
    - Registration Page
    - Add Product to cart page.
    - Place Order Page

Home page

```
package com.sportyshoe_selenium;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;


public class HomePage extends TestBase{
	@FindBy(xpath="//div[@class='container mt-3']/descendant::p[1]")
	WebElement text;

	@FindBy(linkText="New User? Register Here")
	WebElement registerLink;

	public HomePage(WebDriver driver) {
		PageFactory.initElements(driver, this);

	}

	public String getURL_page()
	{
		String URLnew = driver.getCurrentUrl();
		return URLnew;
	}

	public String Validate_Text_On_Page()
	{
```

```
		String pageText = text.getText();
		System.out.println(pageText);
		return pageText;
	}

	public void click_register_link()
	{
		registerLink.click();
	}

}
```

Login page

```
package com.sportyshoe_selenium;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
public class LoginPage {
	@FindBy(xpath="//input[@id='email']")
	WebElement loginEmail;

	@FindBy(xpath="//input[@id='password']")
	WebElement loginpassword;

	@FindBy(xpath="//button[@type='submit']")
	WebElement loginbtn;

	@FindBy(linkText="Cart")
	WebElement clickCart;


	public LoginPage(WebDriver driver) {
		PageFactory.initElements(driver, this);


	}

	public void user_login()
	{
		loginEmail.sendKeys("sonal@gmail.com");
		loginpassword.sendKeys("sonal@123");
		loginbtn.click();
	}


	public void click_cart()
	{
		clickCart.click();
	}

}
```

Register page

```
package com.sportyshoe_selenium;

import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;


public class RegisterPage extends TestBase{

	@FindBy(xpath="//input[@id='name']")

	WebElement registername;


	@FindBy(xpath="//input[@id='email']")

	WebElement registeremail;


	@FindBy(xpath="//input[@id='password']")

	WebElement registerpassword;


	@FindBy(xpath="//button[@type='submit']")

	WebElement registerBtn;


	@FindBy(xpath="//div[@class='mt-4 p-5 bg-primary text-white
rounded']/descendant::p[3]")

	WebElement userText;




	public RegisterPage(WebDriver driver) {

		PageFactory.initElements(driver, this);


	}


	public void register_user()
```

```
	{
		registername.sendKeys("sonal");
		registeremail.sendKeys("sonal@gmail.com");
		registerpassword.sendKeys("sonal@123");
		registerBtn.click();
	}

	public String validate_registration_URL()
	{
		String register_url = driver.getCurrentUrl();
		return register_url;
	}

	public String validate_registration_Text()
	{
		String user_name = userText.getText();
		return user_name;
	}

}
```

Addtocart

```
package com.sportyshoe_selenium;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
```

```
public class AddtoCartPage {

	@FindBy(xpath="//a[@id=\"shoe101\"]")

	WebElement viewShoeBTN;


	@FindBy(xpath = "//a[@id='cart101']")

	WebElement addtocartBTN;


	@FindBy(xpath="//div[@class='alert alert-success']/descendant::p[1]")

	WebElement successText;


	JavascriptExecutor executor;


	public AddtoCartPage(WebDriver driver) {

		PageFactory.initElements(driver, this);

		executor = (JavascriptExecutor) driver;


	}


	public void add_product_to_cart() throws InterruptedException

	{



		executor.executeScript("arguments[0].click();", addtocartBTN);



	}
```

```
	public String validate_success_message()
	{

		String successtext = successText.getText();
		return successtext;

	}

}
```

Order page

```
package com.sportyshoe_selenium;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class OrderPage {
	@FindBy(linkText="Orders")
	WebElement orderlink;


	public OrderPage(WebDriver driver) {
		PageFactory.initElements(driver, this);
	}

	public void click_orderPage()
	{
```

```
            orderlink.click();



      }



}
```

Test scripts

```
package com.sportyshoe.Tests;

import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import com.sportyshoe_selenium.HomePage;
import com.sportyshoe_selenium.TestBase;

import static org.testng.Assert.assertEquals;

import org.testng.Assert;
import org.testng.Assert.*;


public class HomePageTest extends TestBase{
HomePage hp;

      @BeforeTest
      public void start_browser()
      {
            OpenBrowser("Chrome");
            hp = new HomePage(driver);
      }


      @Test(priority='1')

      public void test_getTitle_page()
      {
            String expected = "http://localhost:9010/";
            String Actual = hp.getURL_page();
            Assert.assertEquals(Actual, expected);

      }

      @Test(priority='2')


      public void Test_Validate_Text_On_Page()
      {
            String expected = "Powered By Simplilearn";
            String actualText = hp.Validate_Text_On_Page();
            Assert.assertEquals(actualText, expected);
      }

      @Test(priority='3')

      public void test_click_register_link() throws InterruptedException
```

```
	{
		Thread.sleep(1500);
		hp.click_register_link();
	}


}
```

Login page test

```
package com.sportyshoe.Tests;

import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import com.sportyshoe_selenium.HomePage;
import com.sportyshoe_selenium.LoginPage;
import com.sportyshoe_selenium.RegisterPage;
import com.sportyshoe_selenium.TestBase;
import static org.testng.Assert.assertEquals;

import org.testng.Assert;
import org.testng.Assert.*;


public class LoginPageTest extends TestBase{
	HomePage hp;
	RegisterPage rp;
	LoginPage lp;

	@BeforeTest
	public void start_browser()
	{
		OpenBrowser("Chrome");
		hp = new HomePage(driver);
		rp = new RegisterPage(driver);
		lp = new LoginPage(driver);
	}


	@Test(priority='1')

	public void test_login()
	{
		lp.user_login();
	}


	@Test(priority='2')

	public void test_getTitle_page()
	{
		String expected = "http://localhost:9010/login";
		String Actual = hp.getURL_page();
		Assert.assertEquals(Actual, expected);

	}

@Test(priority='3')
```

```
	public void Test_validate_registration_Text()
	{
		String expected = "Hello sonal !";
		String actualText = rp.validate_registration_Text();
		Assert.assertEquals(actualText, expected);
	}


@Test(priority='4')


public void test_click_cart()
{
	lp.click_cart();
}


}
```

Register page test

```
package com.sportyshoe.Tests;



import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;



import com.sportyshoe_selenium.HomePage;

import com.sportyshoe_selenium.RegisterPage;

import com.sportyshoe_selenium.TestBase;

import static org.testng.Assert.assertEquals;



import org.testng.Assert;




public class RegisterPageTest extends TestBase{

	HomePage hp;

	RegisterPage rp;



	@BeforeTest
```

```
public void start_browser()
{
	OpenBrowser("Chrome");
	hp = new HomePage(driver);
	rp = new RegisterPage(driver);
}


@Test(priority='1')
public void test_click_register_link() throws InterruptedException
{
	Thread.sleep(1500);
	hp.click_register_link();
}

@Test(priority='2')
public void test_getTitle_page()
{
	String expected = "http://localhost:9010/register";
	String Actual = hp.getURL_page();
	Assert.assertEquals(Actual, expected);

}

@Test(priority='3')
public void Test_register_user()
{
	rp.register_user();
```

```
	}

	@Test(priority='4')
	public void Test_validate_registration_URL()
	{
		String expected = "http://localhost:9010/register-user";
		String Actual = rp.validate_registration_URL();
		assertEquals(Actual, expected);
	}

@Test(priority='5')

	public void Test_validate_registration_Text()
	{
		String expected = "Hello sonal !";
		String actualText = rp.validate_registration_Text();
		Assert.assertEquals(actualText, expected);
	}

}
```

Add to cart test

```
package com.sportyshoe.Tests;

import org.testng.annotations.BeforeTest;
import org.testng.annotations.Test;

import com.sportyshoe_selenium.AddtoCartPage;
```

```
import com.sportyshoe_selenium.HomePage;
import com.sportyshoe_selenium.LoginPage;
import com.sportyshoe_selenium.RegisterPage;
import com.sportyshoe_selenium.TestBase;
import static org.testng.Assert.assertEquals;

import org.testng.Assert;
import org.testng.Assert.*;

public class AddtoCartPageTest extends TestBase{
	HomePage hp;
	RegisterPage rp;
	LoginPage lp;
	AddtoCartPage ac;

	@BeforeTest
	public void start_browser()
	{
		OpenBrowser("Chrome");
		hp = new HomePage(driver);
		rp = new RegisterPage(driver);
		lp = new LoginPage(driver);
		ac = new AddtoCartPage(driver);
	}

	@Test(priority='1')
	public void test_login()
	{
```

```
		lp.user_login();

	}



	@Test(priority='2')

	public void test_add_product_to_cart() throws InterruptedException

	{

		ac.add_product_to_cart();

	}



	@Test(priority='3')

	public void test_validate_success_message()

	{

		String expected = "Message:Shoe BlueWave Running Shoes Added
Successfully to Cart";

	String actualText=	ac.validate_success_message();

	Assert.assertEquals(actualText, expected);

	}



}
```

Order page test

```
package com.sportyshoe.Tests;



import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;



import com.sportyshoe_selenium.HomePage;

import com.sportyshoe_selenium.LoginPage;
```

```
import com.sportyshoe_selenium.OrderPage;
import com.sportyshoe_selenium.RegisterPage;
import com.sportyshoe_selenium.TestBase;
import static org.testng.Assert.assertEquals;

import org.testng.Assert;
import org.testng.Assert.*;

public class OrderpageTest extends TestBase {
	HomePage hp;
	RegisterPage rp;
	LoginPage lp;
	OrderPage op;

	@BeforeTest
	public void start_browser()
	{
		OpenBrowser("Chrome");
		hp = new HomePage(driver);
		rp = new RegisterPage(driver);
		lp = new LoginPage(driver);
		op = new OrderPage(driver);
	}

	@Test(priority='1')
	public void test_login()
	{
```

```
		lp.user_login();

	}


	@Test(priority='2')


	public void test_click_orders()

	{

		op.click_orderPage();

	}




	@Test(priority='3')

	public void test_getTitle_page()

	{

		String expected = "http://localhost:9010/orders";

		String Actual = hp.getURL_page();

		Assert.assertEquals(Actual, expected);



	}


}
```

Cucumber

```
Feature: Testing API endpoints on sportyshoe page

 Scenario: get the details from sportyshoe page

  When I get all shoes from sportyshoe page
  Then  I list the  all registered users
  Then  I Add the product to cart
  And I Update the product
  Then I Delete the product
```

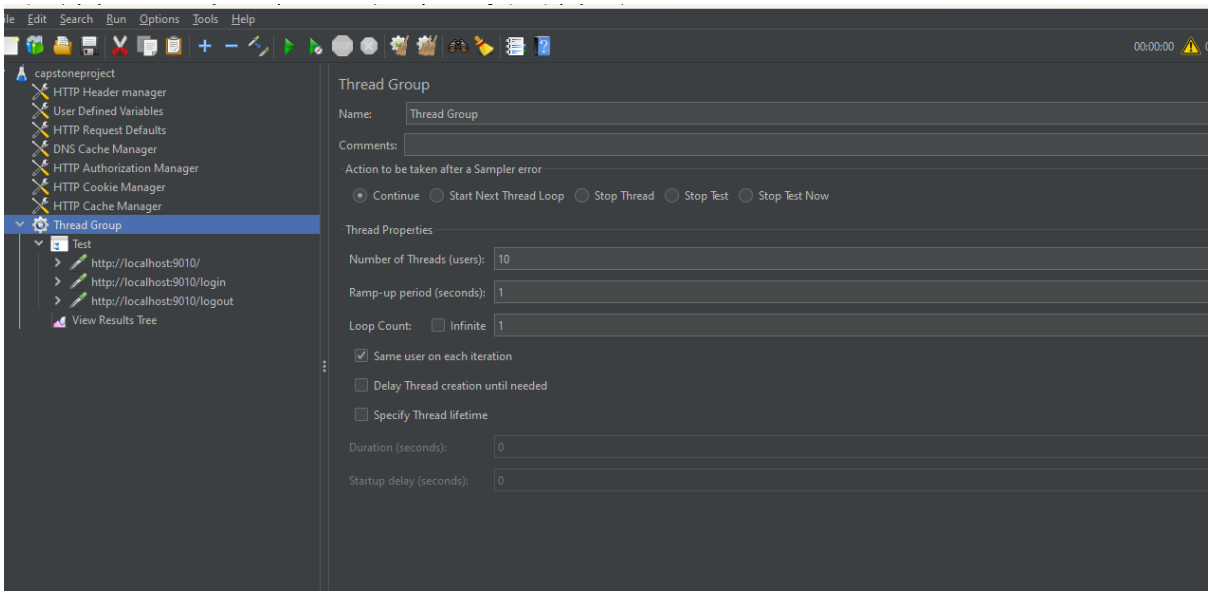
Steps

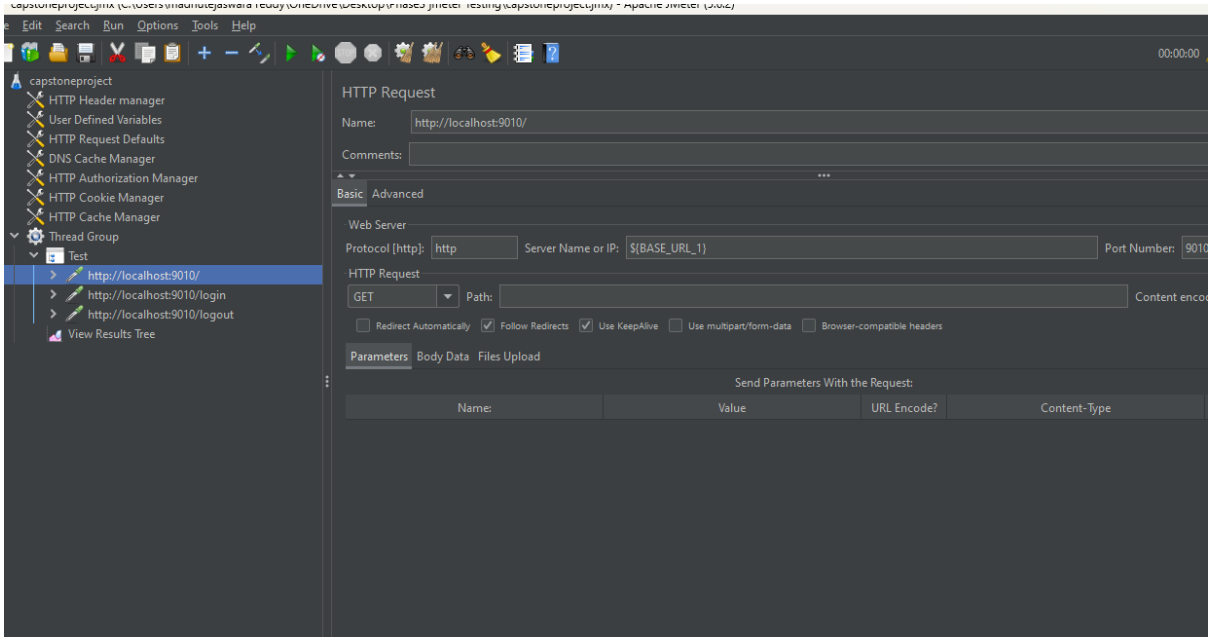
```
package steps;
```

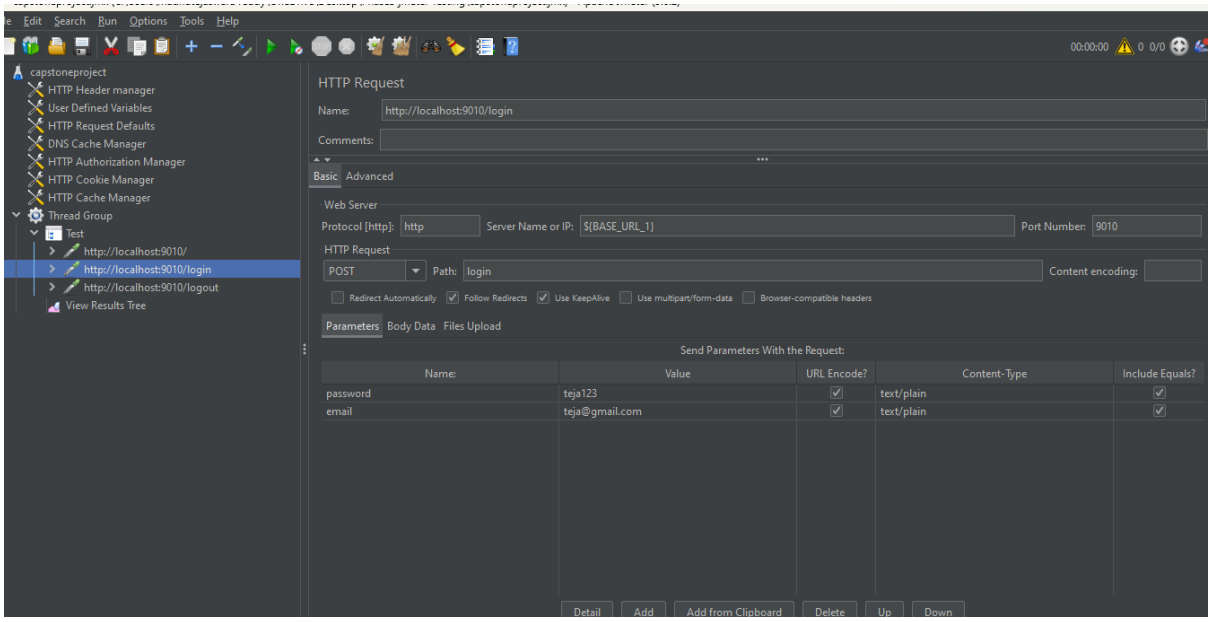
```java
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import io.restassured.RestAssured;


public class Loginpage {
	@When("I get all shoes from sportyshoe page")
	public void i_get_all_shoes_from_sportyshoe_page() {
		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/get-shoes")
		.when()
		.get()
		.then()
		.statusCode(200)
		.log()
		.all();

	}

	@Then("I list the  all registered users")
	public void i_list_the_all_registered_users() {
		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/get-users")
		.when()
		.get()
		.then()
		.statusCode(200)
		.log()
		.all();

	}

	@Then("I Add the product to cart")
	public void i_add_the_product_to_cart() {
		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/add-shoe")
		.queryParam("id","101")
		.queryParam("image", "www.image.com")
		.queryParam("name","Nike")
		.queryParam("category", "Running")
		.queryParam("sizes","9")
		.queryParam("price", "2000")
		.when()
		.post()
		.then()
		.log().all();

	}

	@Then("I Update the product")
	public void i_update_the_product() {
		RestAssured.given()
		.baseUri("http://localhost:9010")
		.basePath("/update-shoe")
		.queryParam("id","101")
		.queryParam("image", "www.imge123.com")
		.queryParam("name","Reebok")
		.queryParam("category", "walking")
```

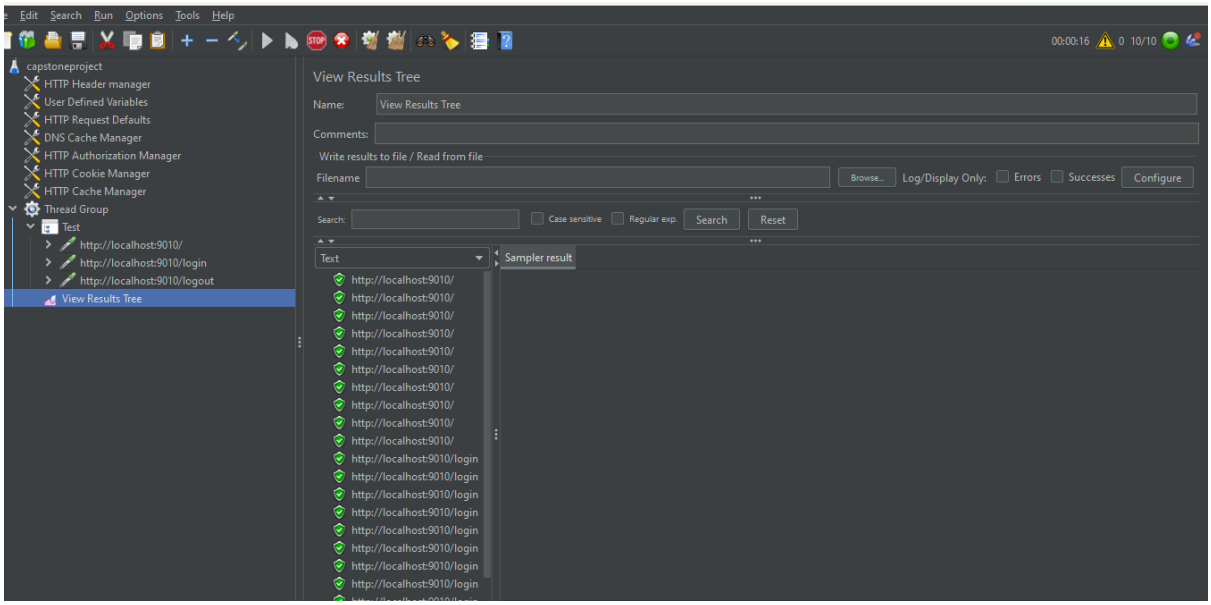
```
                .queryParam("sizes","8,9,10")
                .queryParam("price", "1500")
                .when()
                .put()
                .then()

                .log().all()
                ;

        }

        @Then("I Delete the product")
        public void i_delete_the_product() {
                RestAssured.given()
                .baseUri("http://localhost:9010")
                .basePath("/delete-shoe")
                .queryParam("id", "101")
                .when().delete()
                .then().log().all();


        }
}
```

Jmeter



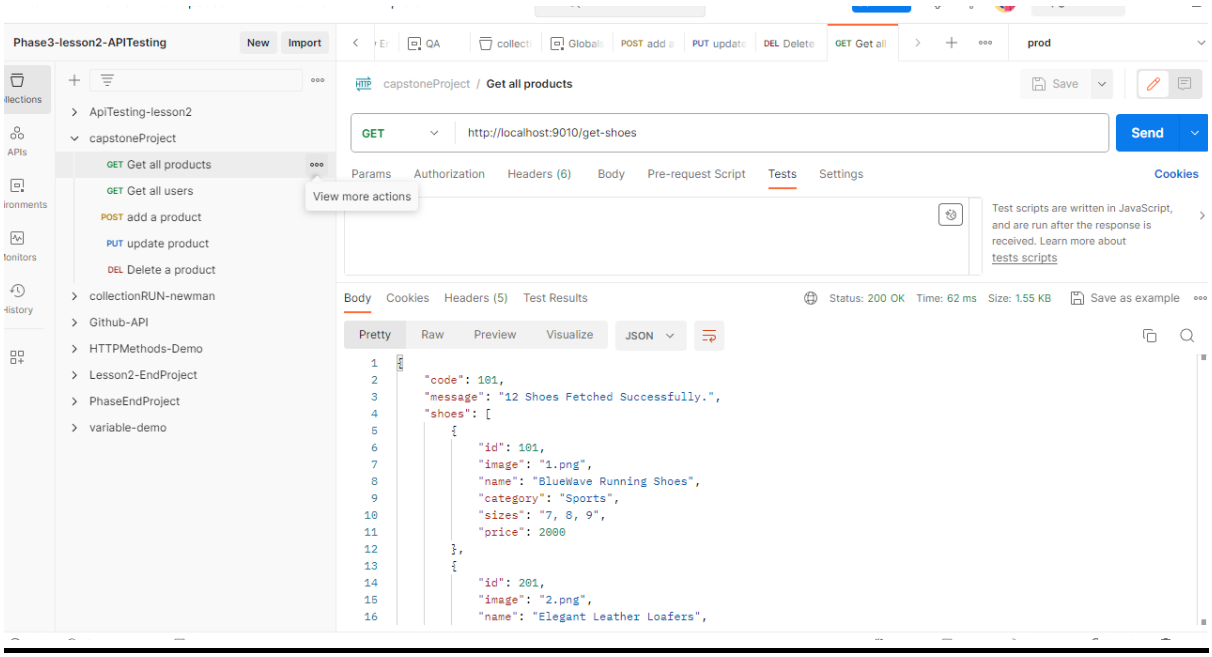

Edit Search Run Options Tools Help
00:00:00
capstoneproject
HTTP Header manager
User Defined Variables
HTTP Request Defaults
DNS Cache Manager
HTTP Authorization Manager
HTTP Cookie Manager
HTTP Cache Manager
Thread Group
Test
http://localhost:9010/
http://localhost:9010/login
http://localhost:9010/logout
View Results Tree
HTTP Request
Name: http://localhost:9010/
Comments:
Basic Advanced
Web Server
Protocol [http]: http
Server Name or IP: ${BASE_URL_1}
Port Number:
HTTP Request
GET
Path:
Content encoc
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters Body Data Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?
Content-Type


Edit Search Run Options Tools Help
00:00:00 0 0/0
capstoneproject
HTTP Header manager
User Defined Variables
HTTP Request Defaults
DNS Cache Manager
HTTP Authorization Manager
HTTP Cookie Manager
HTTP Cache Manager
Thread Group
Test
http://localhost:9010/
http://localhost:9010/login
http://localhost:9010/logout
View Results Tree
HTTP Request
Name: http://localhost:9010/login
Comments:
Basic Advanced
Web Server
Protocol [http]: http
Server Name or IP: ${BASE_URL_1}
Port Number: 9010
HTTP Request
POST
Path: login
Content encoding:
Redirect Automatically
Follow Redirects
Use KeepAlive
Use multipart/form-data
Browser-compatible headers
Parameters Body Data Files Upload
Send Parameters With the Request:
Name:
Value
URL Encode?
Content-Type
Include Equals?
password
teja123
text/plain
email
teja@gmail.com
text/plain
Detail
Add
Add from Clipboard
Delete
Up
Down

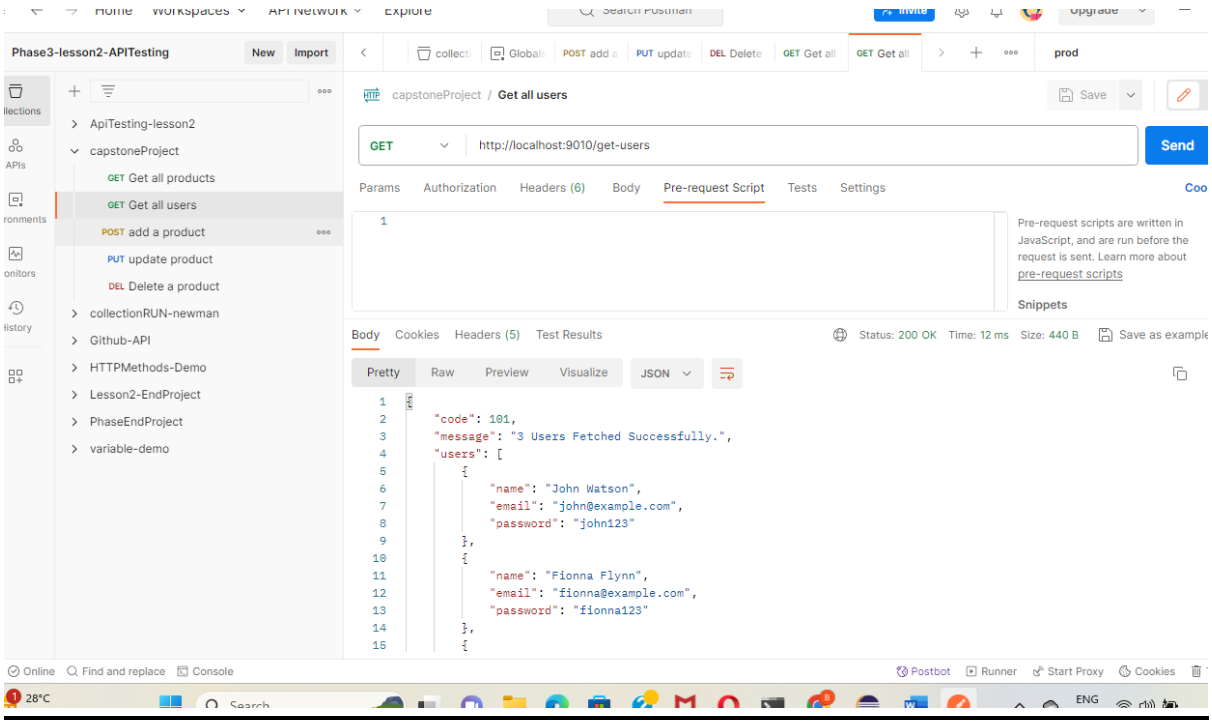


Postman

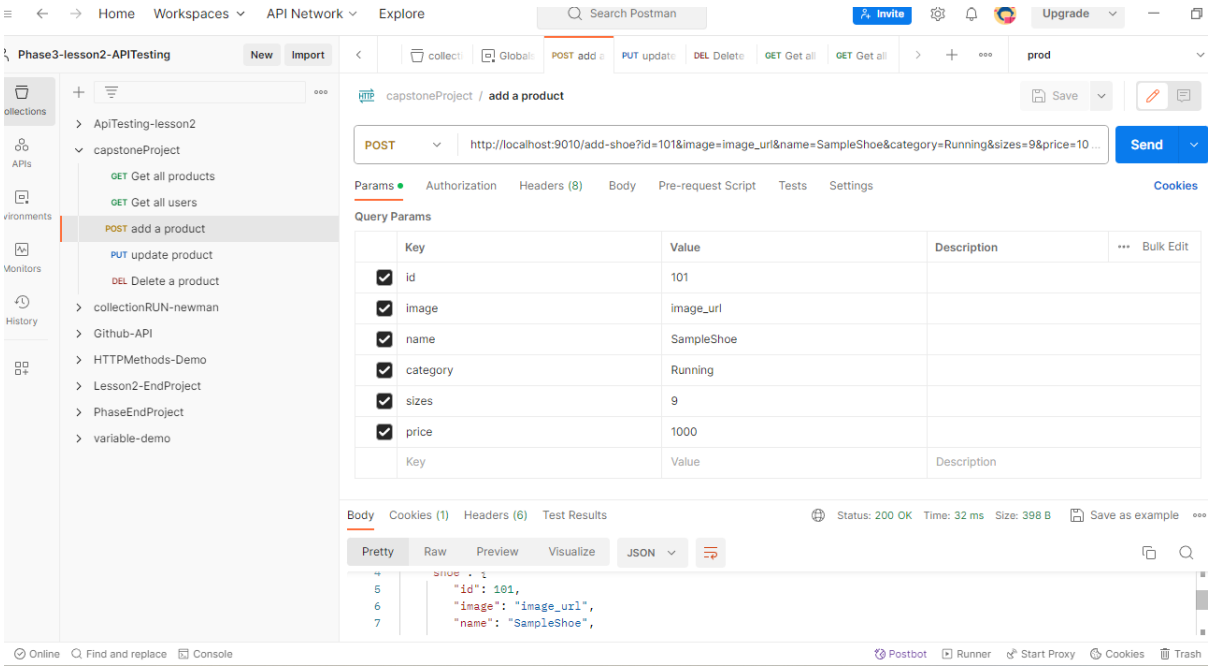
Get all users


Phase3-lesson2-APITesting
New
Import
ApiTesting-lesson2
capstoneProject
GET Get all products
GET Get all users
POST add a product
PUT update product
DEL Delete a product
collectionRUN-newman
Github-API
HTTPMethods-Demo
Lesson2-EndProject
PhaseEndProject
variable-demo
View more actions
capstoneProject / Get all products
Save
GET
http://localhost:9010/get-shoes
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies
Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 62 ms
Size: 1.55 KB
Save as example
Pretty
Raw
Preview
Visualize
JSON
{
"code": 101,
"message": "12 Shoes Fetched Successfully.",
"shoes": [
{
"id": 101,
"image": "1.png",
"name": "BlueWave Running Shoes",
"category": "Sports",
"sizes": "7, 8, 9",
"price": 2000
},
{
"id": 201,
"image": "2.png",
"name": "Elegant Leather Loafers",

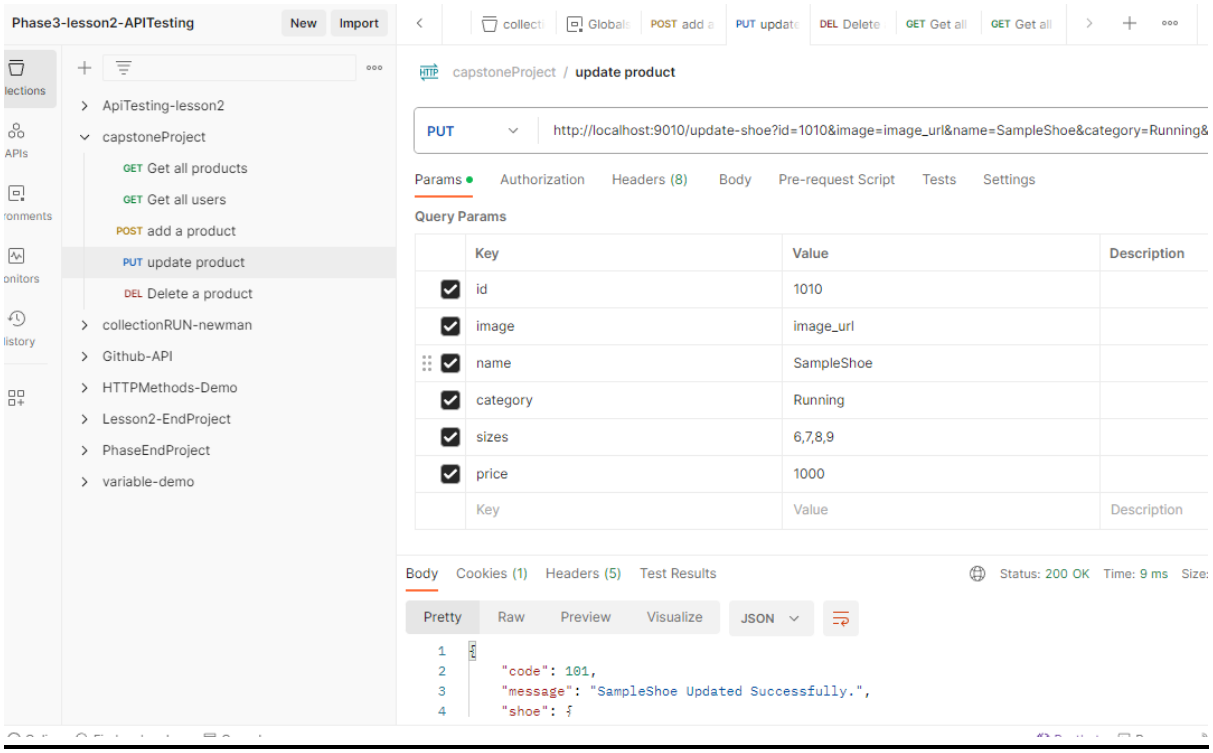
Get


Phase3-lesson2-APITesting
New
Import
ApiTesting-lesson2
capstoneProject
GET Get all products
GET Get all users
POST add a product
PUT update product
DEL Delete a product
collectionRUN-newman
Github-API
HTTPMethods-Demo
Lesson2-EndProject
PhaseEndProject
variable-demo
capstoneProject / Get all users
Save
GET
http://localhost:9010/get-users
Send
Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Pre-request scripts are written in JavaScript, and are run before the request is sent. Learn more about pre-request scripts
Snippets
Body
Cookies
Headers (5)
Test Results
Status: 200 OK
Time: 12 ms
Size: 440 B
Pretty
Raw
Preview
Visualize
JSON
{
"code": 101,
"message": "3 Users Fetched Successfully.",
"users": [
{
"name": "John Watson",
"email": "john@example.com",
"password": "john123"
},
{
"name": "Fionna Flynn",
"email": "fionna@example.com",
"password": "fionna123"
},
{
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
28°C
ENG

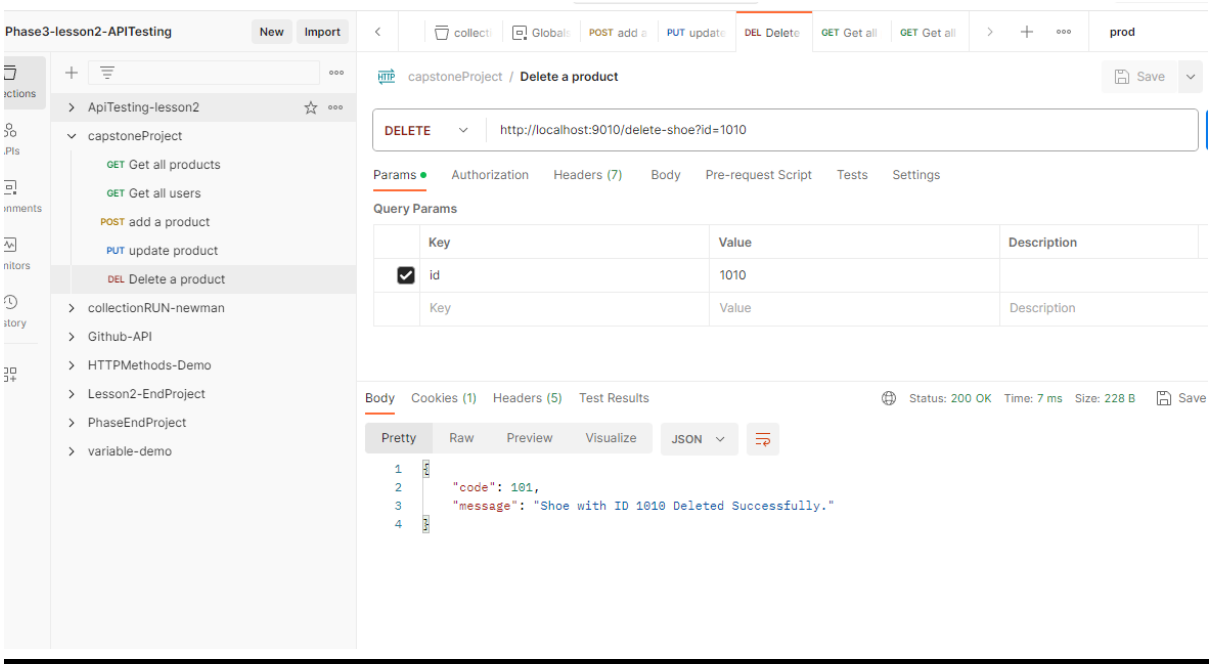
Post


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
Phase3-lesson2-APITesting
New
Import
prod
capstoneProject / add a product
Save
POST
http://localhost:9010/add-shoe?id=101&image=image_url&name=SampleShoe&category=Running&sizes=9&price=10...
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
id
101
image
image_url
name
SampleShoe
category
Running
sizes
9
price
1000
Body
Cookies (1)
Headers (6)
Test Results
Status: 200 OK
Time: 32 ms
Size: 398 B
Save as example
Pretty
Raw
Preview
Visualize
JSON
"id": 101,
"image": "image_url",
"name": "SampleShoe",
ApiTesting-lesson2
capstoneProject
GET Get all products
GET Get all users
POST add a product
PUT update product
DEL Delete a product
collectionRUN-newman
Github-API
HTTPMethods-Demo
Lesson2-EndProject
PhaseEndProject
variable-demo
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash

Put


Phase3-lesson2-APITesting
New
Import
ApiTesting-lesson2
capstoneProject
GET Get all products
GET Get all users
POST add a product
PUT update product
DEL Delete a product
collectionRUN-newman
Github-API
HTTPMethods-Demo
Lesson2-EndProject
PhaseEndProject
variable-demo
capstoneProject / update product
PUT
http://localhost:9010/update-shoe?id=1010&image=image_url&name=SampleShoe&category=Running&
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Query Params
Key
Value
Description
id
1010
image
image_url
name
SampleShoe
category
Running
sizes
6,7,8,9
price
1000
Body
Cookies (1)
Headers (5)
Test Results
Status: 200 OK
Time: 9 ms
Pretty
Raw
Preview
Visualize
JSON
{
"code": 101,
"message": "SampleShoe Updated Successfully.",
"shoe": {

Delete


Phase3-lesson2-APITesting
New
Import
ApiTesting-lesson2
capstoneProject
GET Get all products
GET Get all users
POST add a product
PUT update product
DEL Delete a product
collectionRUN-newman
Github-API
HTTPMethods-Demo
Lesson2-EndProject
PhaseEndProject
variable-demo
prod
capstoneProject / Delete a product
Save
DELETE
http://localhost:9010/delete-shoe?id=1010
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Query Params
Key
Value
Description
id
1010
Key
Value
Description
Body
Cookies (1)
Headers (5)
Test Results
Status: 200 OK
Time: 7 ms
Size: 228 B
Save
Pretty
Raw
Preview
Visualize
JSON
{
"code": 101,
"message": "Shoe with ID 1010 Deleted Successfully."
}