

CAPSTONE PROJECT

By using <http://localhost:9010/> this website I can automate to buy a product by using POM(Page Object Model). Here I use Selenium and Testng.

Source code for Landing Page:

```
package in.sportyshoes.pages;
import org.openqa.selenium.WebDriver;
public class LandingPage {
    private WebDriver driver;
    public LandingPage(WebDriver driver) {
        this.driver = driver;
    }
    public void goToHomePage() {
        driver.get("http://localhost:9010/");
    }
}
```

Source code for Register Page:

```
package in.sportyshoes.pages;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
public class RegisterPage {
    private WebDriver driver;
    public RegisterPage(WebDriver driver) {
        this.driver = driver;
    }
    public void fillRegistrationForm(String name, String email, String password){
        driver.findElement(By.xpath("/html/body/div[2]/form/a")).click();
        driver.findElement(By.id("name")).sendKeys(name);
        driver.findElement(By.id("email")).sendKeys(email);
        driver.findElement(By.id("password")).sendKeys(password);
    }
    public void submitRegistrationForm() {
        driver.findElement(By.xpath("//button[contains(@type,'submit')]")).click();
    }
}
```

Source code for Add Product To Cart page:

```
package in.sportyshoes.pages;
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
public class AddProductToCartPage {
    private WebDriver driver;
    public AddProductToCartPage(WebDriver driver) {
        this.driver = driver;
    }
    public void clickProductButton() throws InterruptedException {
        WebDriverWait wait = new WebDriverWait(driver,Duration.ofSeconds(60));
        WebElement element =
            wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("(//a[contains(@class,'nav-link')])[1]")));
        element.click();
    }
}
```

```

driver.findElement(By.xpath("//a[contains(@class,'nav-
link')])[1]")).click();
Thread.sleep(2000);
WebElement element1 =
driver.findElement(By.xpath("//a[contains(@class,'btn btn-
primary')])[1]"));
((JavascriptExecutor) driver).executeScript("arguments[0].click();",
element1);
}
}

```

Source code for Place Order Page:

```

package in.sportyshoes.pages;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
public class PlaceOrderPage {
private WebDriver driver;
public PlaceOrderPage(WebDriver driver) {
this.driver = driver;
}
public void navigateToCartAndPlaceOrder() throws InterruptedException {
driver.findElement(By.xpath("//a[contains(@class,'nav-
link')])[1]")).click();
Thread.sleep(2000);
driver.findElement(By.xpath("//a[contains(@class,'nav-
link')])[1]")).click();
Thread.sleep(2000);
WebElement element2 =
driver.findElement(By.xpath("//a[contains(@class,'btn btn-
primary')])[1]"));
((JavascriptExecutor) driver).executeScript("arguments[0].click();",
element2);
}
}

```

Source code for SportyShoeOrder:

```

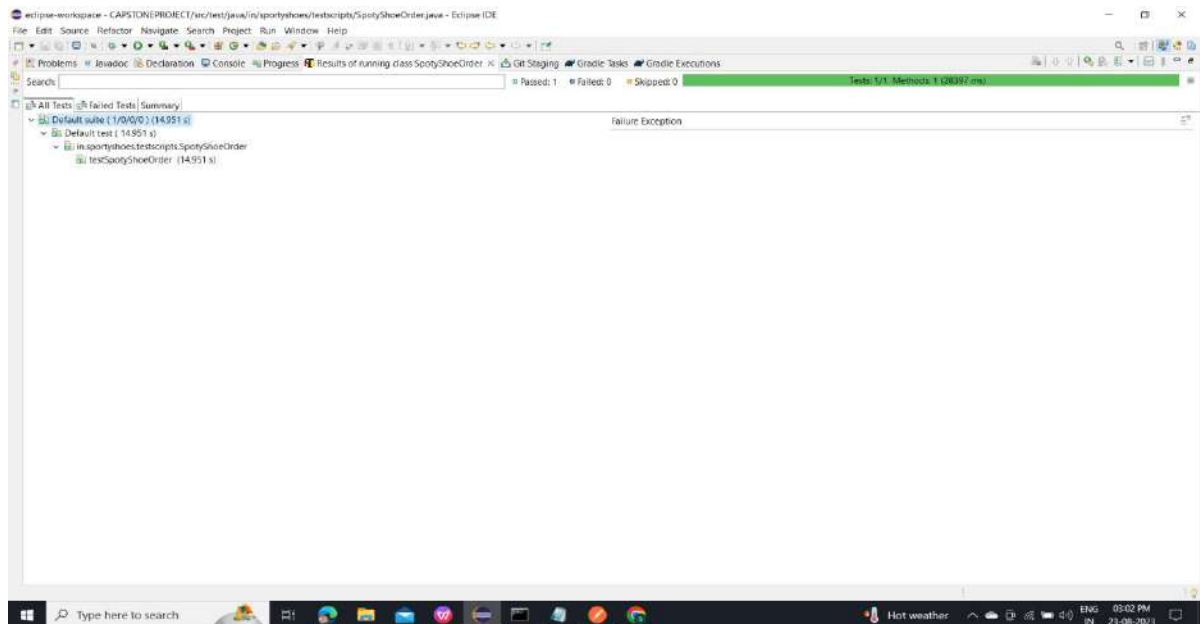
package in.sportyshoes.testscripts;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import in.sportyshoes.pages.LandingPage;
import in.sportyshoes.pages.PlaceOrderPage;
import in.sportyshoes.pages.AddProductToCartPage;
import in.sportyshoes.pages.RegisterPage;
public class SportyShoeOrder {
private WebDriver driver;
private LandingPage landingPage;
private RegisterPage registerPage;
private AddProductToCartPage addproducttocartPage;
private PlaceOrderPage placeorderPage;
@BeforeMethod
public void setUp() {
driver = new ChromeDriver();
driver.manage().window().maximize();
landingPage = new LandingPage(driver);

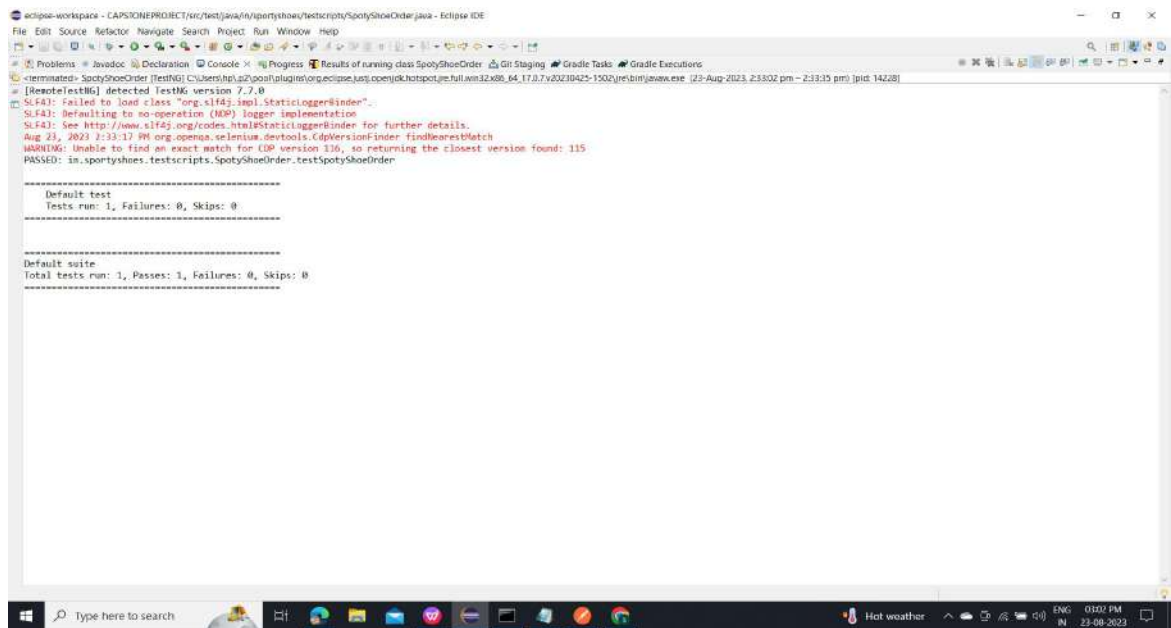
```

```

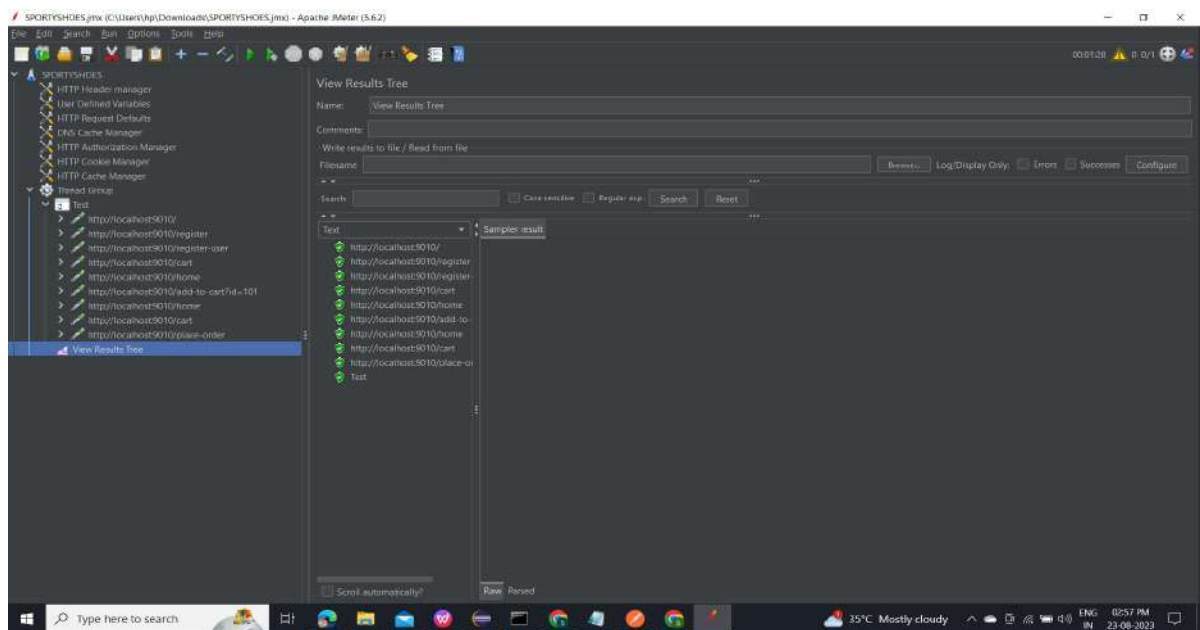
registerPage = new RegisterPage(driver);
addproducttocartPage = new AddProductToCartPage(driver);
placeorderPage = new PlaceOrderPage(driver);
landingPage.goToHomePage();
}
@Test
public void testSpotyShoeOrder() throws InterruptedException {
    registerPage.fillRegistrationForm("john", "abcdef@gmail.com", "123456");
    Thread.sleep(2000);
    registerPage.submitRegistrationForm();
    Thread.sleep(2000);
    addproducttocartPage.clickProductButton();
    Thread.sleep(2000);
    placeorderPage.navigateToCartAndPlaceOrder();
}
@AfterMethod
public void tearDown() {
    driver.quit();
}
}

```



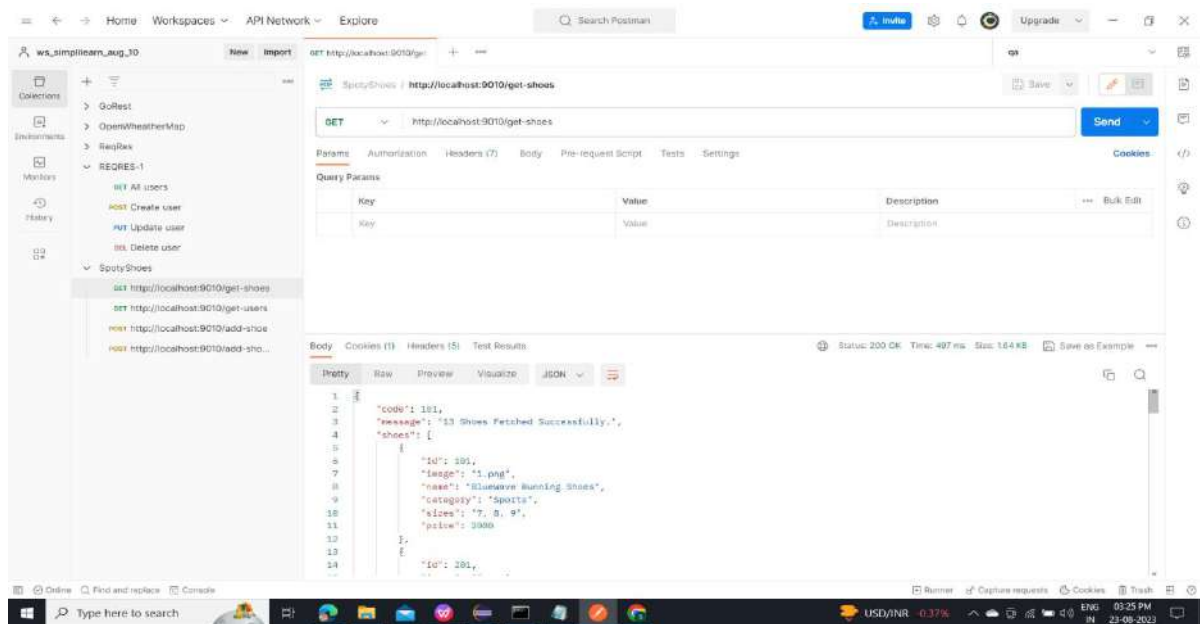


By using jmeter load scripts:

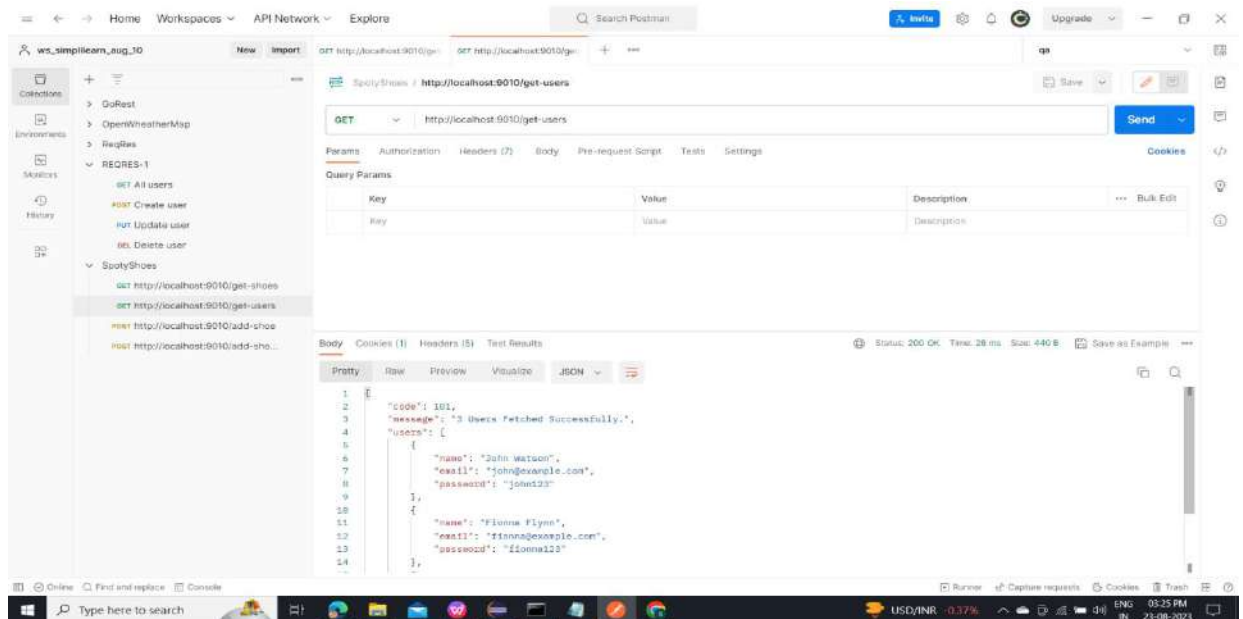


By using postman:

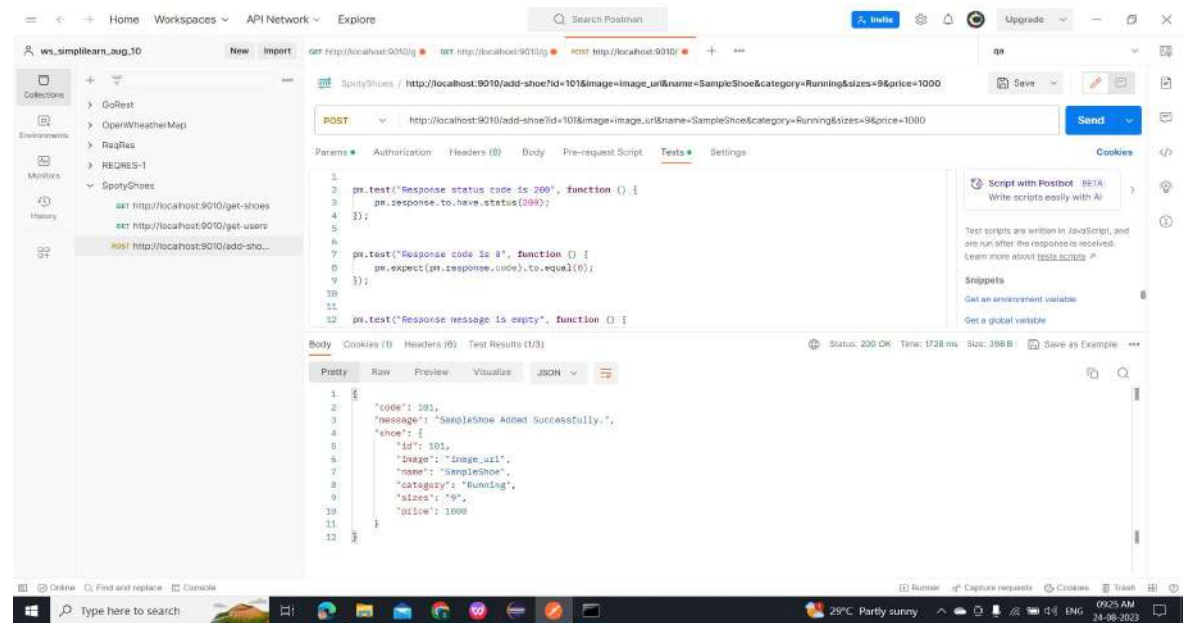
Using GET Request for get-shoes:



Using GET Request for get-users:



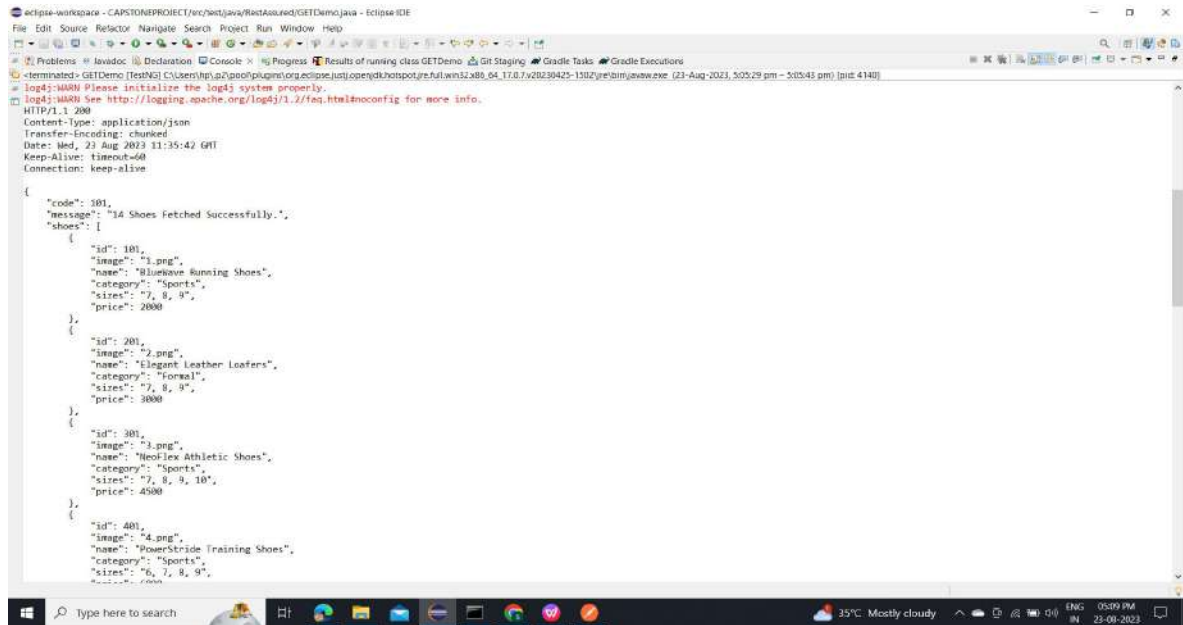
Using POST Request add-shoe:



Rest Assured Using Requests:

GET Request for shoes:

```
package RestAssured;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class GETDemo {
    @Test
    public void getServerResponse() {
        RestAssured
        .when()
        .get("http://localhost:9010/get-shoes")
        .then()
        .assertThat()
        .statusCode(200)
        .log().all();
    }
}
```

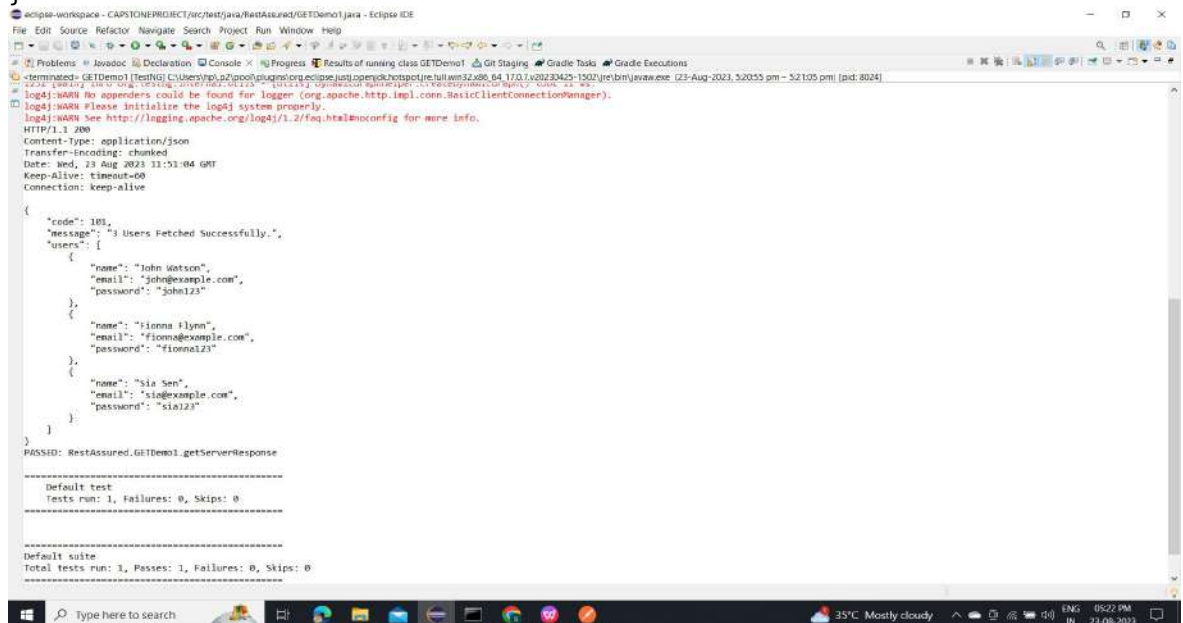


GET Request for users:

```

package RestAssured;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
public class GETDemo1 {
@Test
public void getServerResponse() {
RestAssured
.given()
.when()
.get("http://localhost:9010/get-users")
.then()
.assertThat()
.log().all()
.statusCode(200);
}
}

```



POST Request for add shoe:

```
package RestAssured;
import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;

public class POSTDemo {
    @BeforeMethod
    public void setUp() {
        RestAssured.baseURI = "http://localhost:9010"; // Update the base URI
    }
    @Test
    public void createShoeResourceWithParameters() {
        Response response = given()
            .param("id", 101)
            .param("image", "image_url")
            .param("name", "SampleShoe")
            .param("category", "Running")
            .param("sizes", "9")
            .param("price", 1000)
            .when()
            .post("/add-shoe");
        JsonPath jsonPath = response.jsonPath();
        jsonPath.prettyPrint();
        response.then()
            .assertThat().statusCode(200) // Response status code is 200
            .assertThat().body("code", equalTo(101)) // Response code is 101
            .assertThat().body("message", equalTo("SampleShoe Added Successfully.)) //
            Response message is correct
            .assertThat().body("shoe.id", equalTo(101)) // Shoe ID is 101
            .assertThat().body("shoe.image", equalTo("image_url")) // Shoe image is
            correct
            .assertThat().body("shoe.name", equalTo("SampleShoe")) // Shoe name is
            correct
            .assertThat().body("shoe.category", equalTo("Running")) // Shoe category is
            correct
            .assertThat().body("shoe.sizes", equalTo("9")) // Shoe sizes are correct
            .assertThat().body("shoe.price", equalTo(1000)); // Shoe price is correct
    }
}
```

```

1055 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1055 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1058 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1058 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1061 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1061 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1063 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1063 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1065 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1065 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1065 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1065 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1066 [main] INFO org.testing.internal.Utils - [TestNG] Running:
C:\Users\hp\AppData\Local\Temp\testing-eclipse-618812457\testing-customsuite.xml

1130 [main] INFO org.testing.internal.Utils - [Utils] DynamicGraphHelper.createDynamicGraph() took 9 ms.
{
  "code": 101,
  "message": "SampleShoe Added Successfully.",
  "shoe": {
    "image": "image_url",
    "sizes": "9",
    "price": 3000,
    "name": "SampleShoe",
    "id": 101,
    "category": "Running"
  }
}
PASSED: RestAssured.POSTDemo.createShoeResourceWithParameters

Default test
Tests run: 1, Failures: 0, Skips: 0

Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

4679 [main] INFO org.testing.internal.Utils - [TestNG] Time taken by org.testing.internal.ExitCodeListener@3003e508: 0 ms

```

Here we can change manually to get output of the different shoes:

```

1324 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1324 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1327 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1328 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1328 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1330 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1331 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.collectMethodsByGroup() took 0 ms.
1331 [main] INFO org.testing.internal.Utils - [Utils] MethodGroupHelper.sortMethods() took 0 ms.
1337 [main] INFO org.testing.internal.Utils - [TestNG] Running:
C:\Users\hp\AppData\Local\Temp\testing-eclipse-783842376\testing-customsuite.xml

1405 [main] INFO org.testing.internal.Utils - [Utils] DynamicGraphHelper.createDynamicGraph() took 18 ms.
{
  "code": 101,
  "message": "PowerStride Training Shoes Added Successfully.",
  "shoe": {
    "image": "4.png",
    "sizes": "6, 7, 8, 9",
    "price": 6000,
    "name": "PowerStride Training Shoes",
    "id": 401,
    "category": "Sports"
  }
}
PASSED: RestAssured.POSTDemo.createShoeResourceWithParameters

Default test
Tests run: 1, Failures: 0, Skips: 0

Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

6096 [main] INFO org.testing.internal.Utils - [TestNG] Time taken by org.testing.internal.ExitCodeListener@67531e3a: 0 ms

```

All in one code to form GET and POST Requests:

```

package RestAssured;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import io.restassured.RestAssured;
import io.restassured.path.json.JsonPath;
import io.restassured.response.Response;
import static io.restassured.RestAssured.given;
import static org.hamcrest.Matchers.equalTo;
public class GET_GET_POST {
    @BeforeMethod
    public void setUp() {
        RestAssured.baseURI = "http://localhost:9010"; // Update the base URI
    }

```

```

@Test
public void getShoesServerResponse() {
    RestAssured
        .when()
        .get("/get-shoes")
        .then()
        .assertThat()
        .statusCode(200)
        .log().all();
}

@Test
public void getUsersServerResponse() {
    RestAssured
        .given()
        .when()
        .get("/get-users")
        .then()
        .assertThat()
        .log().all()
        .statusCode(200);
}

@Test
public void createShoeResourceWithParameters() {
    Response response = given()
        .param("id", 401)
        .param("image", "4.png")
        .param("name", "PowerStride Training Shoes")
        .param("category", "Sports")
        .param("sizes", "6, 7, 8, 9")
        .param("price", 6000)
        .when()
        .post("/add-shoe");
    JsonPath jsonPath = response.jsonPath();
    jsonPath.prettyPrint();
    response.then()
        .assertThat().statusCode(200)
        .assertThat().body("code", equalTo(101))
        .assertThat().body("message", equalTo("PowerStride Training Shoes Added Successfully."))
        .assertThat().body("shoe.id", equalTo(401))
        .assertThat().body("shoe.image", equalTo("4.png"))
        .assertThat().body("shoe.name", equalTo("PowerStride Training Shoes"))
        .assertThat().body("shoe.category", equalTo("Sports"))
        .assertThat().body("shoe.sizes", equalTo("6, 7, 8, 9"))
        .assertThat().body("shoe.price", equalTo(6000));
}
}

```

```

1611 [main] INFO org.testing.internal.Utils - [Utils] DynamicGrapher.createDynamicGraph() took 12 ms.
{
  "code": 101,
  "message": "PowerStride Training Shoes Added Successfully.",
  "shoe": {
    "image": "4.png",
    "sizes": "6, 7, 8, 9",
    "price": 6000,
    "name": "PowerStride Training Shoes",
    "id": 401,
    "category": "Sports"
  }
}
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Thu, 24 Aug 2023 06:35:35 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
  "code": 101,
  "message": "28 Shoes Fetched Successfully.",
  "shoes": [
    {
      "id": 101,
      "image": "1.png",
      "name": "Blonde Running Shoes",
      "category": "Sports",
      "sizes": "7, 8, 9",
      "price": 2000
    },
    {
      "id": 201,
      "image": "2.png",
      "name": "Elegant Leather Loafers",
      "category": "Formal",
      "sizes": "7, 8, 9",
      "price": 3000
    },
    {
      "id": 301,
      "image": "3.png",
      "name": "Casual Sneakers",
      "category": "Sports",
      "sizes": "7, 8, 9",
      "price": 4000
    }
  ]
}

```

```

1611 [main] INFO org.testing.internal.Utils - [Utils] DynamicGrapher.createDynamicGraph() took 12 ms.
{
  "code": 101,
  "message": "PowerStride Training Shoes Added Successfully.",
  "shoe": {
    "image": "4.png",
    "sizes": "6, 7, 8, 9",
    "price": 6000,
    "name": "PowerStride Training Shoes",
    "id": 401,
    "category": "Sports"
  }
}
HTTP/1.1 200
Content-Type: application/json
Transfer-Encoding: chunked
Date: Thu, 24 Aug 2023 06:35:35 GMT
Keep-Alive: timeout=60
Connection: keep-alive

{
  "code": 101,
  "message": "28 Shoes Fetched Successfully.",
  "shoes": [
    {
      "id": 101,
      "image": "1.png",
      "name": "Blonde Running Shoes",
      "category": "Sports",
      "sizes": "7, 8, 9",
      "price": 2000
    },
    {
      "id": 201,
      "image": "2.png",
      "name": "Elegant Leather Loafers",
      "category": "Formal",
      "sizes": "7, 8, 9",
      "price": 3000
    },
    {
      "id": 301,
      "image": "3.png",
      "name": "Casual Sneakers",
      "category": "Sports",
      "sizes": "7, 8, 9",
      "price": 4000
    }
  ]
}
{
  "code": 101,
  "message": "3 Users Fetched Successfully.",
  "users": [
    {
      "name": "John Watson",
      "email": "john@example.com",
      "password": "john123"
    },
    {
      "name": "Fiona Flynn",
      "email": "fiona@example.com",
      "password": "fiona123"
    },
    {
      "name": "Sia Sen",
      "email": "sia@example.com",
      "password": "sia123"
    }
  ]
}
PASSED: RestAssured.GET_GET_POST.getShoesServerResponse
PASSED: RestAssured.GET_GET_POST.createShoeResourceWithParameters
PASSED: RestAssured.GET_GET_POST.getUsersServerResponse

Default test
Tests run: 3, Failures: 0, Skips: 0

Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0

5723 [main] INFO org.testing.internal.Utils - [TestNG] Time taken by org.testing.internal.ExitCodeListener@7e9f2c32: 0 ms

```

TestBase:

```

package RestAssured;
import org.apache.log4j.Logger;
import org.apache.log4j.Level;
import org.apache.log4j.PropertyConfigurator;
import org.testng.annotations.BeforeClass;
import io.restassured.response.Response;
import io.restassured.specification.RequestSpecification;
public class TestBase {
    public static RequestSpecification httpRequest;
    public static Response response;
    public Logger logger;
    @BeforeClass
    public void setUp() {
        logger = Logger.getLogger("LogDemo");
    }
}

```

```
PropertyConfigurator.configure("log4j.properties");  
logger.setLevel(Level.DEBUG);  
}  
}
```

This is my GitHub link where I can push my files:

<https://github.com/Guntikala-Samyuktha-Rani/java-fsd.git>