

CAPSTONE PROJECT

Project Details: -

Automation Testing using different Tools –

1. API with Rest Assured
2. Selenium using TestNg
3. Postman
4. JMeter

Create a Testing Framework for Sporty Shoes Website

- The above application is already functional. What is needed now is to add a testing layer that will ensure that everything is passed through QA.

Testing Process: -

Selenium using Testng:

Automate the Sporty Shoes Website and buy shoes online: -

1. Create Maven project
2. Update dependencies in pom.xml with selenium and testng dependencies.
3. According to the Project statement, Create pages and test script using page object model.
4. Import the Pages to Test Script
5. Run the Test Script.
6. Verify from console Output.

Postman:

For Get Request: -

1. Create a request. ...
2. Enter the URL in the address bar. ...
3. Now, select the GET request from the list of request methods.
4. Now press send.

5. Once you press the send button, you will get the response from the server

For Post Request: -

1. Click on the New menu from the Postman application.
2. SAVE REQUEST in Collection.
3. The Request name gets reflected on the Request tab.
4. Move to the Body tab below the address bar and select the option raw.
5. Then, choose JSON from the Text dropdown.
6. Copy and paste the below information in the Postman Body tab.
7. Click on the Send button.

API with Rest Assured: -

1. Create a Maven project.
2. Add appropriate dependencies in pom.xml
3. Create test scripts for every API end points
4. For GET request, Set status Code to 200.
5. For POST request, set status Code to 200, if they didn't give any body, or else status code set to 201.
6. Run the test scripts, watch the result in console output.

JMeter: -

Recording Script using JMeter – record the script using Blazemeter plugin, and then upload the script on JMeter.

View result using view result tree.

<https://github.com/PusaBhargavi01/Capstoneproject>

```

<?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>in.restassured</groupId>
  <artifactId>ResrAssured</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>ResrAssured</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.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <!--
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java --
>
    <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-java</artifactId>
      <version>4.11.0</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.testng/testng -->
    <dependency>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
      <version>7.8.0</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
    <dependency>
      <groupId>org.apache.poi</groupId>
      <artifactId>poi</artifactId>
      <version>5.2.3</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
    <dependency>
      <groupId>org.apache.poi</groupId>
      <artifactId>poi-ooxml</artifactId>
      <version>5.2.3</version>
    </dependency>

    <dependency>
      <groupId>in.amazon</groupId>
      <artifactId>Amazon</artifactId>
      <version>0.0.1-SNAPSHOT</version>
    </dependency>
  </dependencies>

```

```

<build>
  <pluginManagement><!-- lock down plugins versions to avoid using Maven
defaults (may be moved to parent pom) -->
    <plugins>
      <!-- clean lifecycle, see
https://maven.apache.org/ref/current/maven-
core/lifecycles.html#clean\_Lifecycle -->
        <plugin>
          <artifactId>maven-clean-plugin</artifactId>
          <version>3.1.0</version>
        </plugin>
      <!-- default lifecycle, jar packaging: see
https://maven.apache.org/ref/current/maven-core/default-
bindings.html#Plugin\_bindings\_for\_jar\_packaging -->
        <plugin>
          <artifactId>maven-resources-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
        </plugin>
        <plugin>
          <artifactId>maven-surefire-plugin</artifactId>
          <version>2.22.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-jar-plugin</artifactId>
          <version>3.0.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-install-plugin</artifactId>
          <version>2.5.2</version>
        </plugin>
        <plugin>
          <artifactId>maven-deploy-plugin</artifactId>
          <version>2.8.2</version>
        </plugin>
      <!-- site lifecycle, see
https://maven.apache.org/ref/current/maven-
core/lifecycles.html#site\_Lifecycle -->
        <plugin>
          <artifactId>maven-site-plugin</artifactId>
          <version>3.7.1</version>
        </plugin>
        <plugin>
          <artifactId>maven-project-info-reports-plugin</artifactId>
          <version>3.0.0</version>
        </plugin>
    </plugins>
  </pluginManagement>
</build>
</project>

```

```
package in.sportyshoes.Pages;
```

```
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(id = "email")
    private WebElement email;

    @FindBy(id = "password")
    private WebElement password;
    @FindBy(xpath = "//button[contains(@class,'btn')]")
    private WebElement LoginBtn;

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

    public void enteremail(String EMail) {
        email.sendKeys(EMail);
    }

    public void enterPass(String pass) {
        password.sendKeys(pass);
    }

    public void clicklogin() {
        LoginBtn.click();
    }
}

```

```

package in.sportyshoes.Pages;

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 {
    @FindBy(linkText = "New User? Register Here")
    private WebElement register;

    @FindBy(id = "name")
    private WebElement name;

    @FindBy(id = "email")
    private WebElement email;

    @FindBy(id = "password")
    private WebElement password;

    @FindBy(xpath = "//button[contains(@class,'btn')]")
    private WebElement RegisterBtn;

    @FindBy(linkText = "Logout")
    private WebElement logout;
}

```

```

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

    public void clickRegister() {
        register.click();
    }

    public void enterName(String Name) {
        name.sendKeys(Name);
    }

    public void enteremail(String EMail) {
        email.sendKeys(EMail);
    }

    public void enterPass(String pass) {
        password.sendKeys(pass);
    }

    public void clickregister() {
        RegisterBtn.click();
    }

    public void Logout() {
        logout.click();
    }
}

```

```

package in.sportyshoes.Pages;

```

```

import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class ProductsPage {
    private WebDriver driver;
    private JavascriptExecutor jsExecutor;
    Actions actions;

    //@FindBy(xpath = "//a[contains(@class,'btn')][1]")
    @FindBy(linkText = "Add to Cart")
    private WebElement addtocart;
    @FindBy(linkText = "Place Order")
    private WebElement placeorder;

    //@FindBy(xpath = "//div[contains(@class,'alert')]")
    @FindBy(linkText = "Home")
    private WebElement home;

    @FindBy(linkText = "Cart")
    private WebElement cart;

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

```



```

        this.driver =driver;
        this.jsExecutor= (JavascriptExecutor)driver;
    }

    public void clickaddtocart() {
        JavascriptExecutor jsExecutor = (JavascriptExecutor)driver;

        jsExecutor.executeScript("arguments[0].click();", addtocart);
    }

    public void clickHome() {
        home.click();
    }
    public void clickcart() {
        cart.click();
    }

    public void clickplaceOrder() {
        JavascriptExecutor jsExecutor = (JavascriptExecutor)driver;

        jsExecutor.executeScript("arguments[0].click();", placeorder);
    }

}

package in.sportyshoes.TestScripts;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import in.sportyshoes.Pages.LoginPage;
import in.sportyshoes.Pages.ProductsPage;
import in.sportyshoes.Pages.RegisterPage;

public class SportyShoesBuyOnline {
    public static void main(String[] args) throws InterruptedException{

        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("http://localhost:9010/");

        RegisterPage registerPage = new RegisterPage(driver);
        LoginPage loginPage = new LoginPage(driver);
        ProductsPage productsPage = new ProductsPage(driver);
        registerPage.clickRegister();
        registerPage.enterName("Bhargavi");
        registerPage.enteremail("abcde@gmail.com");
        registerPage.enterPass("Bhar@01");
        registerPage.clickregister();
        registerPage.Logout();
        loginPage.enteremail("abcde@gmail.com");
        loginPage.enterPass("Bhar@01");
        loginPage.clicklogin();
        productsPage.clickaddtocart();
        productsPage.clickHome();
    }
}

```

```
productsPage.clickcart();  
Thread.sleep(3000);  
  
productsPage.clickplaceOrder();  
}  
  
}
```

```

<?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>in.restassured</groupId>
  <artifactId>REStAssured</artifactId>
  <version>0.0.1-SNAPSHOT</version>

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

  <properties>
    <serenity.version>1.5.8</serenity.version>
    <serenity.maven.version>1.5.8</serenity.maven.version>
    <slf4j.version>1.6.1</slf4j.version>

    <maven.failsafe.plugin.version>2.18</maven.failsafe.plugin.version>

    <maven.compiler.plugin.version>3.2</maven.compiler.plugin.version>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>net.serenity-bdd</groupId>
      <artifactId>serenity-core</artifactId>
      <version>${serenity.version}</version>
    </dependency>
    <dependency>
      <groupId>net.serenity-bdd</groupId>
      <artifactId>serenity-rest-assured</artifactId>
      <version>${serenity.version}</version>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-simple</artifactId>
      <version>${slf4j.version}</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/org.testng/testng -->
    <dependency>
      <groupId>org.testng</groupId>
      <artifactId>testng</artifactId>
      <version>7.1.0</version>
      <scope>test</scope>
    </dependency>

    <!-- API, java.xml.bind module -->
    <dependency>
      <groupId>jakarta.xml.bind</groupId>
      <artifactId>jakarta.xml.bind-api</artifactId>
      <version>2.3.2</version>
    </dependency>

    <!-- Runtime, com.sun.xml.bind module -->
    <dependency>

```

```

        <groupId>org.glassfish.jaxb</groupId>
        <artifactId>jaxb-runtime</artifactId>
        <version>2.3.2</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/log4j/log4j -->
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
</dependencies>

<build>
    <pluginManagement><!-- lock down plugins versions to avoid using Maven
defaults (may be moved to parent pom) -->
        <plugins>
            <!-- clean lifecycle, see
https://maven.apache.org/ref/current/maven-
core/lifecycles.html#clean_Lifecycle -->
            <plugin>
                <artifactId>maven-clean-plugin</artifactId>
                <version>3.1.0</version>
            </plugin>
            <!-- default lifecycle, jar packaging: see
https://maven.apache.org/ref/current/maven-core/default-
bindings.html#Plugin_bindings_for_jar_packaging -->
            <plugin>
                <artifactId>maven-resources-plugin</artifactId>
                <version>3.0.2</version>
            </plugin>
            <plugin>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.0</version>
            </plugin>
            <plugin>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>2.22.1</version>
            </plugin>
            <plugin>
                <artifactId>maven-jar-plugin</artifactId>
                <version>3.0.2</version>
            </plugin>
            <plugin>
                <artifactId>maven-install-plugin</artifactId>
                <version>2.5.2</version>
            </plugin>
            <plugin>
                <artifactId>maven-deploy-plugin</artifactId>
                <version>2.8.2</version>
            </plugin>
            <!-- site lifecycle, see
https://maven.apache.org/ref/current/maven-
core/lifecycles.html#site_Lifecycle -->
            <plugin>
                <artifactId>maven-site-plugin</artifactId>
                <version>3.7.1</version>
            </plugin>
            <plugin>
                <artifactId>maven-project-info-reports-plugin</artifactId>
                <version>3.0.0</version>

```

```
        </plugin>
    </plugins>
</pluginManagement>
</build>
</project>
```

```
package in.restassured.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();
    }
}
```

```
package in.restassured.RESTassured;
```

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

```
import io.restassured.RestAssured;
```

```
public class GetDemoUsers {

    @Test

    public void getServerResponse() {

        RestAssured

            .when()

                .get("http://localhost:9010/get-users")
```

```

        .then()

        .assertThat()

        .statusCode(200)

        .log().all();
    }

}

package in.restassured.RESTassured;

import org.testng.annotations.Test;

import io.restassured.RestAssured;

public class postDemoshoe {

    @Test

    public void getServerResponse () {
        RestAssured

            .when()
                .post ("http://localhost:9010/add-
shoe?id=101&image=image_url&name=SampleShoe&category=Running&sizes=9&price=
1000")
            .then()
                .assertThat ()
                    .statusCode (200)
                .log () .all ();
    }

}

```