

Test Framework Document

(By: Nimesha Kumarapperuma)

1. Introduction

An overview of the Selenium-based test automation framework used for the Guru99 demo application is given in this document. It describes how to set up, organize, and run tests as well as recommended practices for utilizing and maintaining the framework.

2. Framework Overview

The framework is built using Java, Selenium WebDriver, and TestNG. It supports cross-browser testing and includes modular components for configuration, test execution, reporting, and utility functions.

Key Features

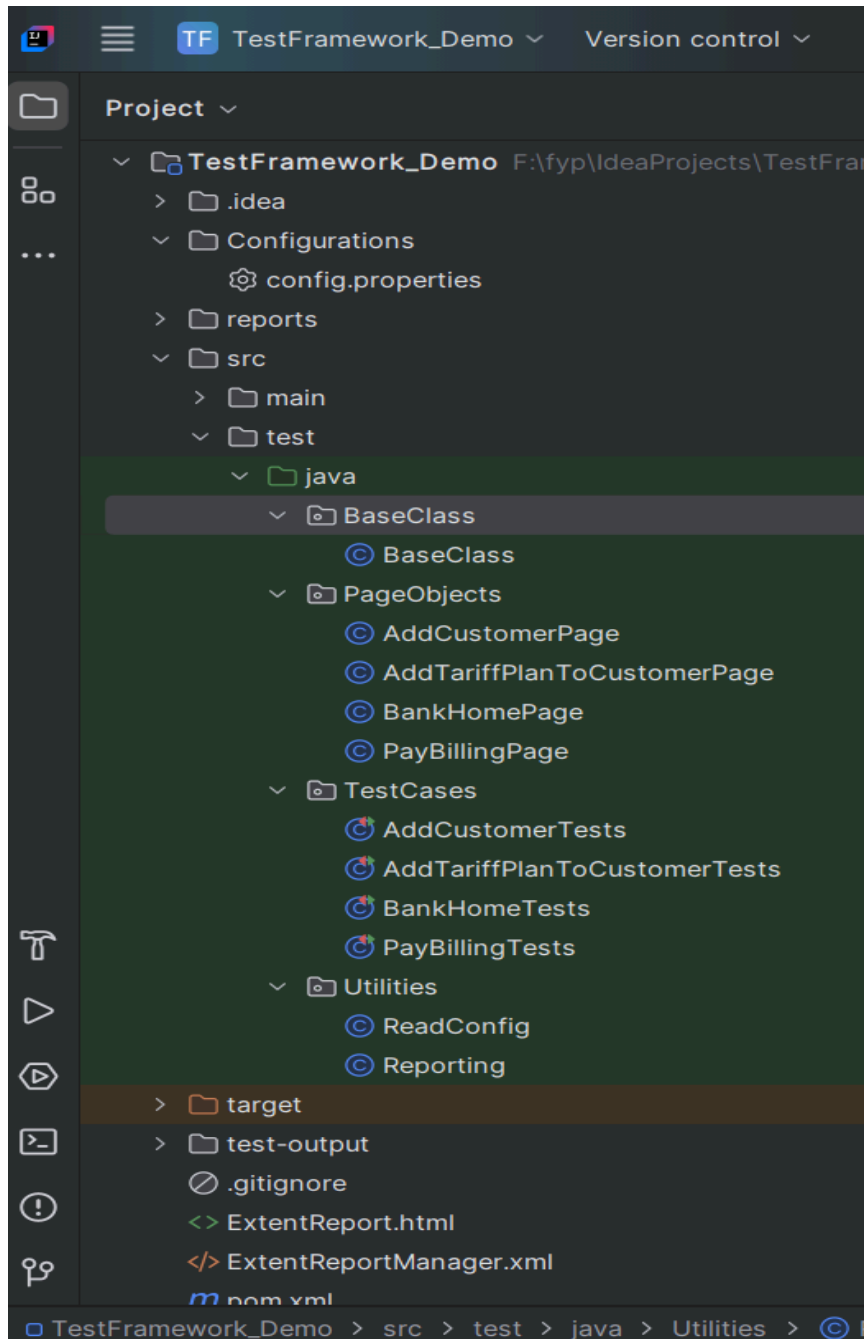
- Cross-browser support: Chrome and Firefox.
- Modular design using Page Object Model (POM)

```
<!--  
https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->  
  
<dependency>  
  
    <groupId>org.seleniumhq.selenium</groupId>  
  
    <artifactId>selenium-java</artifactId>  
  
    <version>4.25.0</version>  
  
</dependency>  
  
<!-- https://mvnrepository.com/artifact/org.testng/testng -->  
  
<dependency>  
  
    <groupId>org.testng</groupId>  
  
    <artifactId>testng</artifactId>  
  
    <version>7.10.2</version>  
  
    <scope>test</scope>  
  
</dependency>
```

- Extensible for new test cases and features

3. Project Structure

The project is organized as follows:



Here I have to create four packages in .test/java folder. Such as;

- **BaseClass**
Here I included the methods that run as Before method and After method and method for capture screenshots.
- **PageObjects**
Here I included the locators and action methods which are corresponding for a particular web page.
- **TestCases**
Here I include the test scripts for the particular web pages.
- **Utilities**
Here I create two separate classes to do the reporting and read the config.properties file.

4. Framework Components

4.1 Configuration

The **config.properties** file stores global settings such as URLs and credentials:

```
Url=https://demo.guru99.com/  
email=nimeshamadhushani100@gmail.com  
customerID=616944
```

Config.properties file is used for storing the values which use redundantly.

4.2 BaseClass

BaseClass.java handles setup and teardown methods with support for cross-browser testing:

```
@Parameters({"browser", "Url"})  
@BeforeMethod  
public void setup(String browser, String Url)  
{  
    if (browser.equals("chrome")) {  
        driver = new ChromeDriver();  
        driver.manage().window().maximize();  
  
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));  
    }  
    else if (browser.equals("firefox")) {  
        driver = new FirefoxDriver();  
    }  
}
```

```

        driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
    }
    driver.get(Url);
}

@AfterMethod
public void tearDown() {
    driver.quit();
}

```

4.3 Page Object Model (POM)

Each page is represented as a separate class with locators and reusable methods:

```

WebDriver driver;

//constructor
public AddCustomerPage(WebDriver driver)
{
    this.driver = driver;
    PageFactory.initElements(driver, this);
}

```

```

@FindBy(xpath = "//label[@id='message']") WebElement MessagesOfFirstName;
@FindBy(id = "message50") WebElement MessagesOfLastName;
@FindBy(id = "message9") WebElement MessagesOfEmail;
@FindBy(id = "message3") WebElement MessagesOfAddress;
@FindBy(id = "message7") WebElement MessagesOfTelephoneNo;

```

```

//Methods
public void clickAddCustomerBtn() {
    AddCustomerBtn.click();
}
public void clickBackgroundCheckBtn() {
    BackgroundCheckBtn.click();
}
public void clickSubmitBtn() {
    SubmitBtn.click();
}
}

```

4.4 Test Cases

Test classes extend the base test class and use TestNG annotations:

```

AddCustomerPage AC;

@Test(priority = 0)
public void addCustomer01() { //Verify that user can navigate to the Add
Customer page

```

```

AC = new AddCustomerPage(driver);
AC.clickAddCustomerBtn();
String expectedTitle = "Guru99 Telecom Add Customer";
String actualTitle = driver.getTitle();
Assert.assertEquals(actualTitle, expectedTitle, "Redirected to an
unexpected page!");
}

```

```

@Test(priority = 1)
public void addCustomer02(){//Verify that user can submit the customer
details form with filling all the fields
    AC = new AddCustomerPage(driver);
    AC.clickAddCustomerBtn();
    AC.clickBackgroundCheckBtn();
    AC.fillFirstName("Nimesha");
    AC.fillLastName("Perera");
    AC.fillEmail("nimesha@gmail.com");
    AC.fillAddress("No121, Galle Road, Matara");//Defect in Address Field
    AC.fillTelephoneNo("0779687555");
    AC.clickSubmitBtn();
    Alert alert = driver.switchTo().alert();
    alert.accept();
    String actualUrl = driver.getCurrentUrl();
    Assert.assertTrue(actualUrl.contains("/telecom/access.php?"));
}

```

4.5 Reporting

The **Reporting.java** utility integrates ExtentReports for customized test execution reports.

```

public ExtentSparkReporter sparkReporter;//UI of the report

public ExtentReports extent;//Populate common information of the report

public ExtentTest test;//create test case entries in the report and update
status of the test methods

```

```

public void onStart(ITestContext context){

```

```

    String timeStamp = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new
    Date());

```

```

        repName = "Test-Report-"+timeStamp+".html";

```

```

        sparkReporter = new ExtentSparkReporter( "./reports/"+repName);

```

```

sparkReporter.config().setDocumentTitle("Automation Report of Guru99");
sparkReporter.config().setReportName("Functional Testing of Telecom Project");
sparkReporter.config().setTheme(Theme.DARK);
extent = new ExtentReports();
extent.attachReporter(sparkReporter);
extent.setSystemInfo("Application" , "Guru99");
extent.setSystemInfo("Environment" , "QA");
extent.setSystemInfo("User Name" , System.getProperty("user.name"));
extent.setSystemInfo("Tester Name" , "Nimesha");
extent.setSystemInfo("os" , "Windows10");
extent.setSystemInfo("Browser Name" , "Chrome");
}

```

```

public void onTestFailure(ITestResult result) {

    test = extent.createTest(result.getTestClass().getName());

    test.log(Status.FAIL , "Test case Failed : " + result.getName());

    test.log(Status.INFO , result.getThrowable().getMessage());


    try{

        String imgPath = new BaseClass().captureScreen(result.getName());

        test.addScreenCaptureFromPath(imgPath);

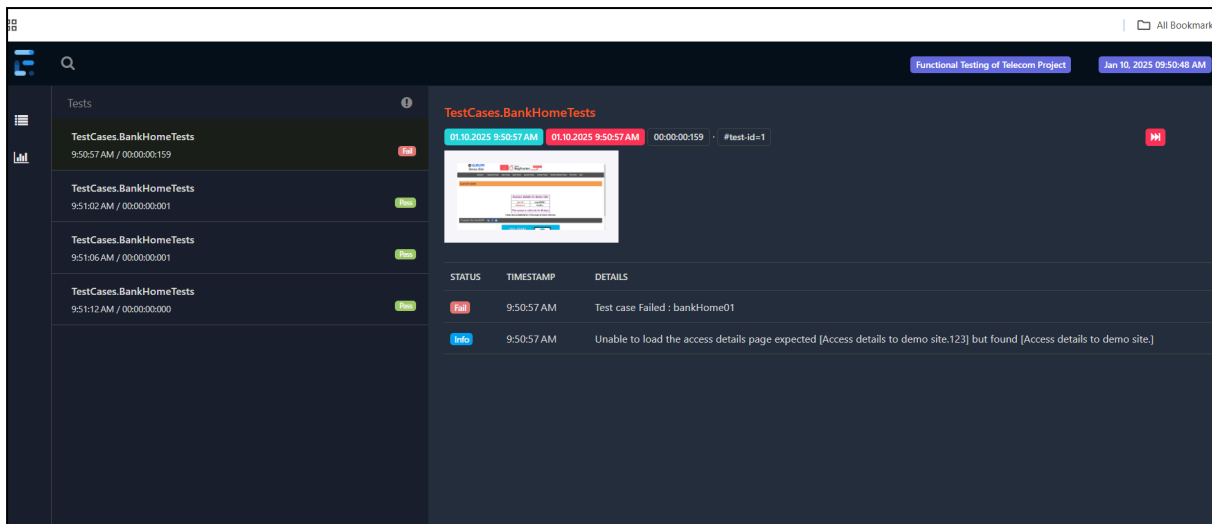
    }catch (IOException e1){

        e1.printStackTrace();

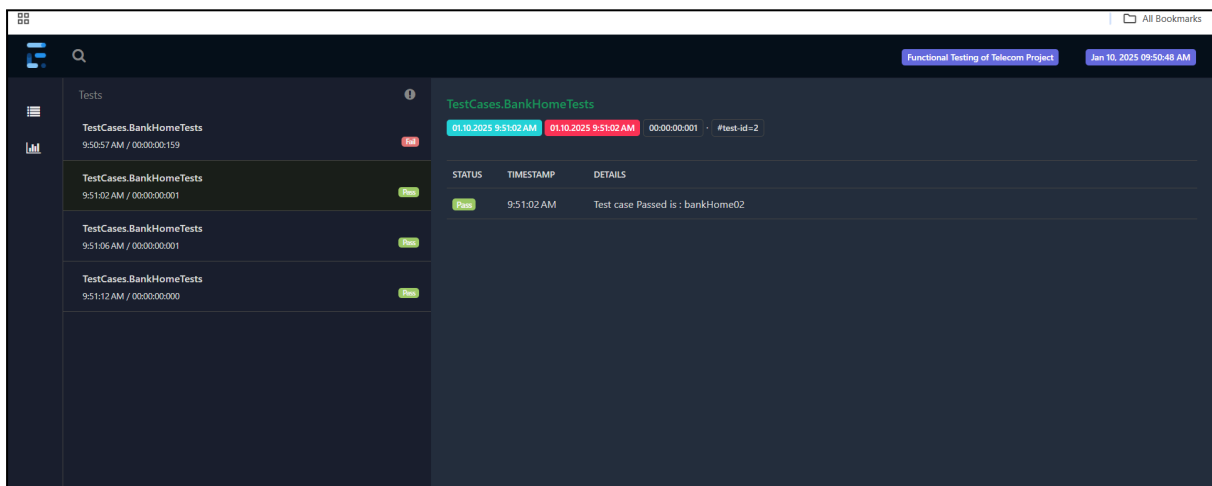
    }

}
}

```



Attach Screen shots when test method get fail



5. Test Execution

5.1 Running Tests

Update the **testng.xml** file to configure browser parameters and test classes:

```
<suite name="Test Suite for Guru99">
  <test name="Bank Home Page">
    <parameter name="browser" value="chrome" />
    <parameter name="Url" value="https://demo.guru99.com/" />
    <classes>
      <class name="TestCases.BankHomeTests"/>
    </classes>
  </test>
</suite>
```

```

<test name="Add Customer Page">

  <parameter name="browser" value="chrome" />

  <parameter name="Url" value="https://demo.guru99.com/telecom/index.html"
/>

  <classes>

    <class name="TestCases.AddCustomerTests"/>

  </classes>

</test>

```

And view reports in the **test-output** folder.

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Login Suite for Guru99							
Guru99	4	0	0	0	24,725		
Class		Method		Start		Time (ms)	
Login Suite for Guru99							
Guru99 — passed							
Cases.BankHomeTests	bankHome01	1736243111390		2876			
	bankHome02	1736243118424		339			
	bankHome03	1736243122894		312			
	bankHome04	1736243127424		2514			
Guru99							
Cases.BankHomeTests#bankHome01							
						back to summary	
Cases.BankHomeTests#bankHome02							
						back to summary	
Cases.BankHomeTests#bankHome03							
						back to summary	
Cases.BankHomeTests#bankHome04							

6. Conclusion

A solid basis for testing the Guru99 example application is offered by this framework. It incorporates crucial features like reporting, and cross-browser testing and is scalable and adaptable. The framework may be expanded to include more situations and applications with the right upkeep.