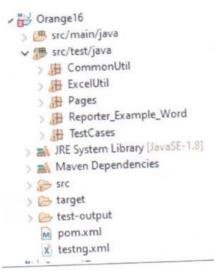# Selenium Sudhakar

Selenium Java CUCUMBER GuideWIre Testing API_Testing SQL Agile_Scrum HR_Round Profiles

WebDriver_arct(Browser)	Selenium Grid	InvocationCount	TestNG Annotations	Waits
Get_vs_Navigate	Selenium Limitations	TestNG_DataProvider	TestNG_Parameters1	Frames
Dropdown	Selenium Featues	TestNG_DataProvider_Lab2	Parallel_Methods_Tests	Multiple_Windows
OR1_applicationProperties	Selenium Challanges	Browser_Compatability_Testing	ParallelClasses	Java_Script_Executer
OR1_Excel	Web Table	ExcelUtil_Apache_POI	Multiple_Suites	Action_class
Css_Selector	Capcha	Proxy Setting	TestNG_Groups	Alerts
xpath_vs_cssSelector	Assertions	Dimesions	TestNG_Listeners	TakeScreenShot
Advanced_Xpaths	Page Object Model	Keyword Driven	TestNG_Priority	Robot_Class
Exceptions	Page Factory Model	Interfaces	dependsOnMethods	ExcelUtil_Apache_POI
Web Element Border	Web Element Color	Classes in Selenium	BrokenLinks	Emailable Report
Locators in Selenium	FindElement vs Find Elements	Number of Hyperlinks	Extent Reports	SSL Certificate
Page Load Strategy	Relative Locators	Selenium Suite	Extent_Reports1	Preserve Order
Maven Life Cycle - Pom.xml	HeadLess Browser	is Displayed - is Enabled - is Selected	Navigate_Get	Quit_Close
Tell me about your self	Selenium_Version	ChromeOptions	GIT HUB Commands	X,Y Coordinates of WebElement
ProjectApproach	Critical_Scenario	Framework_Development	Selenium_jars	SprintWork
Sleep_SetSpeed	Automation_Approach	GSPAN	IncognitoTab	type_typekeys
Collections	GIT_Mergeconflicts	Pending	Pending	Pending

Orange16
> src/main/java
∨ src/test/java
  > CommonUtil
  > ExcelUtil
  > Pages
  > Reporter_Example_Word
  > TestCases
> JRE System Library [JavaSE-1.8]
> Maven Dependencies
> src
> target
> test-output
  pom.xml
  testng.xml

1. We are following Hybrid framework , it is a combination of PageObject Model + Data driven Framework.

2. In our project the main available packages are Pages,TestCases,ReportingUtils,ExcelUtils,ApplicationUtils,Common Utils,Drivers,TestData and OR packages are available

Pages--> for each webpage/entity specific , we will write a seperate java file and corresponding methods will be implemented.

Suppose for a Login WebPage , we can write Login.java , which contains only LoginPage specifc methods will be implemented.

Similarly , for customers WebPage , We can write Customers.java, which contains CustomerPage specific methods will be implemented

For Customer webpage we can implement specific methods as AddCustomer,ModifyCustomers ,SearchCustomer and deleteCustomer etc

Coming to the main TestCase , We can create the Object for every class and we will call the required methods as per the Test Case functional flow

```
LoginPage L1 = new LoginPage();
L1.LoginPage(driver);
L1.Login();

Customer C1= new Customer ();
C1.Customer(driver);
C1.AddCustomer();

HomePage H1= new HomePage();
H1.HomePage(driver);
H1.Logout();
```

We will prapare

TestNG.xml batch file to execute set of test cases in parallel or sequential.

After Executing Testng.xml batch file,default emailable report will be generated in Test Output folder

Coming to the reporting part , we are using Extent Reports.Extent Report.html page will capture all screenshots for every step in the Test Case

To manage Object Repository , We are writing all web Elements xpath/CSSSelectors globally in OR.java file.

```
public static String  UserName="//*[@id='txtUsername']";
public static String  Password="//*[@id='txtPassword']";
public static String  Loginbutton="//*[@id='btnLogin']";
public static String  Admin="//*[@id='menu_admin_viewAdminModule']/b";
public static String  Nationality="//*[@id='menu_admin_nationality']";
```

To read Input data from an Excel files, we are using Excel Util apache POI & TestNG Data Provider.

In the TestNG DataProviders , We have to specify Excel File path & sheet Name , so that all data will be stored in Data Provider Object Array .

Data Provider provides input data to the Test and Test will execute for multiple times for different sets of input data , like that we are following Data driven framework approach.

```java
import ExcelUtil.ExcelApiTest4;

@DataProvider(name = "TC01_Add_Nationality")
public static Object[][] Authentication1() throws Exception {

    ExcelApiTest4 eat = new ExcelApiTest4();
    Object[][] testObjArray = eat.getTableArray("C://HTML Report//OrangeHRM6//TC01_Nationality3.xlsx", "Sheet1");
    System.out.println(testObjArray.length);
    return (testObjArray);
}

@Test(dataProvider="TC01_Add_Nationality")
public void Login_Test(String Browser,String TestURL,String UserName,String Password,String Nationality) throws Exception
{

                if(Browser.equalsIgnoreCase("Chrome"))
                {
                    driver = TestBrowser.OpenChromeBrowser();

                }

                if(Browser.equalsIgnoreCase("FireFox"))
                {
                        driver = TestBrowser.FirefoxBrowser();
                        Thread.sleep(10000);
                }


                if(Browser.equalsIgnoreCase("RemoteWebDriver1"))
                {
                        driver = TestBrowser.RemoteWebDriver();
                        Thread.sleep(10000);
                }
```

```java
public class LoginPage
{
    WebDriver driver;

    public void LoginPage(WebDriver driver)
    {
        this.driver = driver;
    }

    public void Login() throws Exception {

        findElement(By.name("txtUsername")).sendKeys("Admin");
        findElement(By.name("txtPassword")).sendKeys("admin123");
        findElement(By.id("btnLogin")).click();
    }

    public   WebElement findElement(By by) throws Exception
    {

        WebElement elem = driver.findElement(by);
        if (driver instanceof JavascriptExecutor)
        {
         ((JavascriptExecutor)driver).executeScript("arguments[0].style.border='3px solid red'", elem);
        }
        return elem;
    }
}
```

```java
public class TC01_Add_Nationalities
{
    WebDriver driver;

    @BeforeTest
    public void  TestSetup()throws Exception {

        driver = TestBrowser.OpenChromeBrowser();
        String TestURL = "https://opensource-demo.orangehrmlive.com/";
        driver.get(TestURL);
    }

    @Test
    public void Book_OneWay_Flight() throws Exception {

        driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);

        LoginPage L1 = new LoginPage();
        L1.LoginPage(driver);
        L1.Login();

        Nationalities N1= new Nationalities();
        //N1.Nationalities(driver);
        N1.AddNationality();

        HomePage H1= new HomePage();
        H1.HomePage(driver);
        H1.Logout();
    }
```