

Data Driven Automation Framework



Library Package

```
public class Excel {  
    public static String readExcelData(String path,String sheet,int row, int cell) {  
        String data="";  
        try {  
            FileInputStream fis =new FileInputStream(path);  
            XSSFWorkbook workbook=new XSSFWorkbook(fis);  
            data=workbook.getSheet(sheet).getRow(row).getCell(cell).toString();  
        } catch (Exception e) {  
            System.out.println("File Not Found: "+e.getStackTrace());  
        }  
        return data;  
    }  
}
```

```
public interface Constants {  
    String appURL="https://www.google.com/";  
    String excelPath="./excel/NameData.xlsx";  
    String sheetName="name";  
    String instagram=Excel.readExcelData(excelPath,sheetName,0,0);  
    String twitter=Excel.readExcelData(excelPath,sheetName,1,0);  
    String linkedin=Excel.readExcelData(excelPath,sheetName,2,0);  
}
```

```
public class BasePage {  
    public BasePage(WebDriver driver) {  
        PageFactory.initElements(driver, this);  
    }  
}
```

```
public class BaseTest implements Constants{
    public WebDriver driver;
    @BeforeMethod
    public void launchBrowser() {
        WebDriverManager.chromedriver().setup();
        driver=new ChromeDriver();
        driver.manage().window().maximize();
        driver.get(appURL);
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.SECONDS);
    }
    @AfterMethod
    public void tearDown() {
        driver.manage().deleteAllCookies();
        driver.close();
    }
}
```

PageObject Package

```
public class GoogleSearchPagePO extends BasePage{  
    public GoogleSearchPagePO(WebDriver driver) {  
        super(driver);  
    }  
    @FindBy(name="q")  
    private WebElement searchTextBox;  
    public WebElement searchTextBox() {  
        return searchTextBox;  
    }  
}
```

```
public class SearchResultPagePO extends BasePage{
    public SearchResultPagePO(WebDriver driver) {
        super(driver);
    }
    @FindBy(xpath="//a[@href='https://twitter.com/']")
    private WebElement twitterLink;
    public WebElement twitterLink() {
        return twitterLink;
    }
    @FindBy(xpath="//a[@href='https://www.instagram.com/']")
    private WebElement instagramLink;
    public WebElement instagramLink() {
        return instagramLink;
    }
    @FindBy(xpath="//a[@href='https://www.linkedin.com/']")
    private WebElement linkedinLink;
    public WebElement linkedinLink() {
        return linkedinLink;
    }
}
```

TestScripts Package

```
public class TestInstagram extends BaseTest{
    @Test
    public void instagram() {
        try {
            GoogleSearchPagePO gsp=new GoogleSearchPagePO(driver);
            gsp.searchTextBox().sendKeys(instagram);
            gsp.searchTextBox().sendKeys(Keys.ENTER);

            SearchResultPagePO srp=new SearchResultPagePO(driver);
            srp.instagramLink().click();

            String expTitle="Instagram";
            String actTitle = driver.getTitle();

            Assert.assertEquals(actTitle, expTitle);
            Reporter.log("Instagram Title is matching", true);
        } catch (Exception e) {
            Reporter.log("Instagram Title is not matching", true);
            Assert.fail();
        }
    }
}
```

```
public class TestTwitter extends BaseTest{
```

```
    @Test
```

```
    public void twitter() {
```

```
        try {
```

```
            GoogleSearchPagePO gsp=new GoogleSearchPagePO(driver);
```

```
            gsp.searchTextBox().sendKeys(twitter);
```

```
            gsp.searchTextBox().sendKeys(Keys.ENTER);
```

```
            SearchResultPagePO srp=new SearchResultPagePO(driver);
```

```
            srp.twitterLink().click();
```

```
            String expTitle="Twitter. It's what's happening / Twitter";
```

```
            String actTitle = driver.getTitle();
```

```
            AssertJUnit.assertEquals(actTitle, expTitle);
```

```
            Reporter.log("Twitter Title is matching", true);
```

```
        } catch (Exception e) {
```

```
            Reporter.log("Twitter Title is not matching", true);
```

```
            AssertJUnit.fail();
```

```
        }
```

```
    }
```

```
}
```



```
public class TestLinkedin extends BaseTest{

    @Test
    public void linkedin() {
        try {
            GoogleSearchPagePO gsp=new GoogleSearchPagePO(driver);
            gsp.searchTextBox().sendKeys(linkedin);
            gsp.searchTextBox().sendKeys(Keys.ENTER);

            SearchResultPagePO srp=new SearchResultPagePO(driver);
            srp.linkedinLink().click();

            String expTitle="LinkedIn: Log In or Sign Up";
            String actTitle = driver.getTitle();

            Assert.assertEquals(actTitle, expTitle);
            Reporter.log("LinkedIn Title is matching", true);
        } catch (Exception e) {
            Reporter.log("LinkedIn Title is not matching", true);
            Assert.fail();
        }
    }
}
```