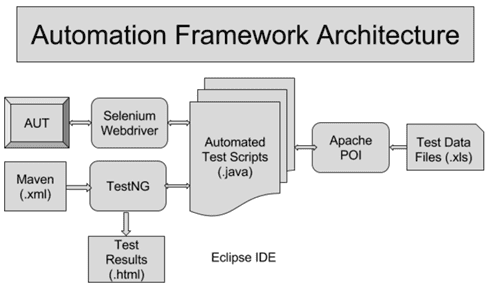
Datadriven framework

Data Driven framework is used to drive the test cases and suite from an external data feed. The data feed can be data sheets like xls, xlsx, and csv files.

## 1) ARCHITECTURE FOR DATA DRIVEN TESTING



### Advantages of using Data Driven Framework

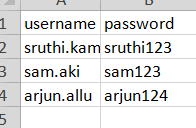
* Improves test coverage.
* Re-usability of code.
* Less maintenance.
* Faster Execution.
* Permits better error handling.

An example which describes the datadriven framework:

Usecase: login to fb in chromedriver with different username and password

1. create an excel sheet and add username and password data
2. Read the excelsheet using apachepoi in java
3. The reading excelsheet which acts as a dataprovider
4. Use the dataprovider in the selinium test class to pass the username and password values to test login at a time with different set of login details.

Loginfb excel sheet in excel workbook:



Read data from excel using apache POI(need to download apache poi jars and add them in java buildpath)

**package** dataDrivenUsingTestng;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** org.apache.poi.openxml4j.exceptions.InvalidFormatException;

**import** org.apache.poi.ss.usermodel.Cell;

**import** org.apache.poi.ss.usermodel.Row;

**import** org.apache.poi.ss.usermodel.Sheet;

**import** org.apache.poi.ss.usermodel.Workbook;

**import** org.apache.poi.xssf.usermodel.XSSFWorkbook;

**public** **class** ReadExcel {

**public** Workbook workbook;

**public** String[][] getCellData(String path, String sheetName) **throws** InvalidFormatException, IOException {

FileInputStream fis = **new** FileInputStream(path);

workbook = **new** XSSFWorkbook(fis);

Sheet sheet = workbook.getSheet(sheetName);

**int** rowcount = sheet.getLastRowNum();

**int** cellcount = sheet.getRow(0).getLastCellNum();

String data[][] = **new** String[rowcount][cellcount];

**for** (**int** i = 1; i <= rowcount; i++) {

Row row = sheet.getRow(i);

**for** (**int** j = 0; j < cellcount; j++) {

Cell c = row.getCell(j);

**try** {

**if** (c.getCellType() == c.getCellType().***STRING***) {

data[i - 1][j] = c.getStringCellValue();

} **else** {

data[i - 1][j] = String.*valueOf*(c.getNumericCellValue());

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

**return** data;

}

}

Selenium class using testng framework:

**package** dataDrivenUsingTestng;

**import** java.io.IOException;

**import** java.util.concurrent.TimeUnit;

**import** org.apache.poi.openxml4j.exceptions.InvalidFormatException;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.testng.annotations.AfterSuite;

**import** org.testng.annotations.BeforeSuite;

**import** org.testng.annotations.DataProvider;

**import** org.testng.annotations.Test;

**public** **class** TestLogin {

WebDriver driver;

@BeforeSuite

**public** **void** setUp() {

System.*setProperty*("webdriver.chrome.driver", "C:\\sruthi-selenium\\chromedriver.exe");

driver = **new** ChromeDriver();

driver.manage().deleteAllCookies();

driver.manage().timeouts().pageLoadTimeout(30, TimeUnit.***SECONDS***);

driver.manage().timeouts().implicitlyWait(30, TimeUnit.***SECONDS***);

driver.get("https://www.facebook.com/");

}

**@Test(dataProvider="getTestData")**

public void logIn(String username, String pswd) {

driver.findElement(By.*id*("email")).sendKeys(username);

driver.findElement(By.*name*("pass")).sendKeys(pswd);

driver.findElement(By.*linkText*("Log In")).click();

}

**@DataProvider**

**public String[][] getTestData() throws InvalidFormatException, IOException {**

**ReadExcel data = new ReadExcel();**

**return data.getCellData("C:\\sruthi-selenium\\practiceTest\\TestFacebook\\src\\datadrivenTesting\\Book1.xlsx",**

**"loginFB");**

**}**

@AfterSuite

**public** **void** closePage() {

driver.close();

}

}

### Conclusion

Input/Output from and to a file is a very critical part of software testing process. ****Apache POI**** plays a vital role in making this possible for Selenium Test Automation.

Selenium integrated with ****Apache POI**** facilitates to run the script multiple times with different data sets, with all data maintained at a single location. It saves time and maintenance effort on the test script.