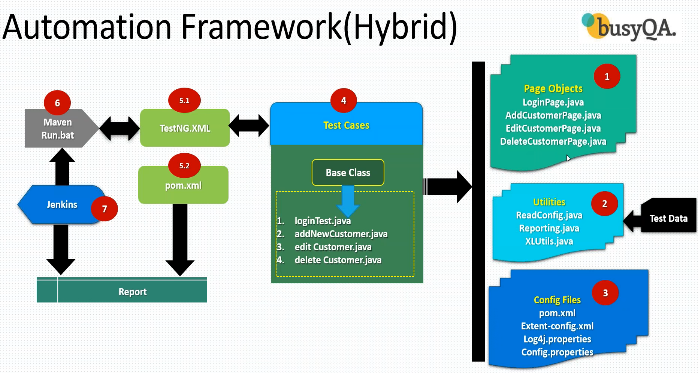
HYBRID FRAMEWORK

1. What is framework in selenium web driver
2. What is the problem here?  
   🡪All the tests will be written in different ways. That will cause a problem in running them together.  
   🡪All the tests will have custom way of logging, that will bring in problem in collecting logs at the end of test run.
3. What is a framework?  
   🡪The initial requirement of having a standard way of writing tests is what will evolve into a framework.  
   🡪Every company or teams can have a different frameworks but the whole idea will be to have a common set of rules.  
   🡪Framework development takes time.  
   🡪A framework is a set of assumptions, concepts and practices that needs to be followed.
4. Components of framework  
   🡪Test case standardization.  
   🡪Logging standardization.  
   🡪Test data and configuration utility.  
   🡪Helper / Utility library.  
   🡪Test execution engine.  
   🡪Reporting utility.
5. Hybrid (POM + Data Driven)  
   Language : JAVA  
   Automation tool : Selenium  
   Build management tool : Maven  
   Unit Test : TestNG

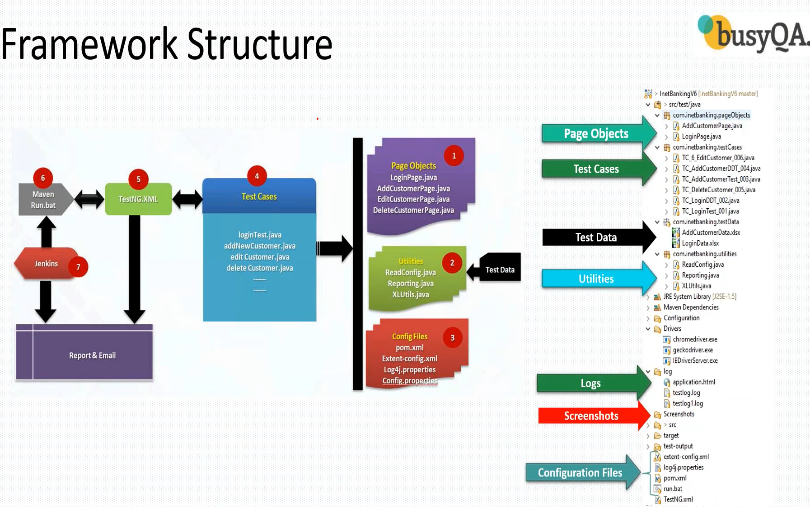
Excel : Apache POI  
Logs : Log4j  
HTML report : Extent report

Application : Any Website E.g. OrangeHRM

1. Folder structure:  
   src/main/java  
   \*config

Automation Framework (Hybrid)  
  


🡪Page Objects, Utilities and Config files are reusable. These three are resources.  
🡪1, 2, 3, 4 are implementation / development part.  
🡪5.1 and 5.2 are execution part.  
🡪6 and 7 are continuous integration part.  
  
These entire framework is divided into multiple phases  
Phase-1: Implementation  
Phase-2: Execution  
Phase-3: Maintenance



Phase-1: Implementation  
🡪Create maven project  
🡪Update pom.xml  
🡪Create page objects  
🡪Create basic test case  
🡪Add logs to test case  
🡪Read common values from prosperities file  
🡪Run test case on desired browser  
🡪Add extent report  
🡪Create data driven test case  
🡪Adding new test case.

Phase-2: Execution  
🡪Run test cases with maven pom.xml  
🡪Run test cases through maven CLI (Command Line Interface)  
🡪Run test cases using run bat  
🡪Run test cases using Jenkins (using bat file)

Phase-3: Maintenance  
🡪Creating repository on GITHUB  
🡪Commit the project code in local repository  
🡪Push the project code to GITHUB remote repository from local GIT repository

STEPS:

🡪Create maven project.  
🡪Under src/test/java, create packages -   
com.inetbanking.pageobjects  
com.inetbanking.testcases  
com.inetbanking.testdata  
com.inetbanking.utilities  
🡪create folders   
Drivers  
Configuration  
Screenshots

🡪Now open pom.xml and add dependencies tag  
Go to maven repository and add dependencies:  
Junit  
Selenium  
TestNG  
Apache POI  
Extent reports  
Log4j API and core

🡪Add different drivers to the Drivers folder.

🡪Now create a class ‘LoginPage’ under pageobjects package.  
**package** com.inetbanking.pageobjects;

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

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.support.CacheLookup;

**import** org.openqa.selenium.support.FindBy;

**import** org.openqa.selenium.support.PageFactory;

**public** **class** LoginPage {

WebDriver ldriver;

**public** LoginPage(WebDriver rdriver) {

ldriver = rdriver;

PageFactory.*initElements*(rdriver, **this**);

}

@FindBy(name="uid")

@CacheLookup

WebElement txtUserName;

@FindBy(name="passsword")

@CacheLookup

WebElement txtPassword;

@FindBy(name="btnLogin")

@CacheLookup

WebElement btnLogin;

**public** **void** setUserName(String uname) {

txtUserName.sendKeys(uname);

}

**public** **void** setPassword(String pwd) {

txtPassword.sendKeys(pwd);

}

**public** **void** clickSubmit() {

btnLogin.click();

}

}

🡪Now create a class ‘BaseClass’ under testcases package.

package com.inetbanking.testcases;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

public class BaseClass {

public String baseURL = "https://demo.guru99.com/V4/";

public String username = "mngr464075";

public String password = "sUqypuh";

public static WebDriver driver;

@BeforeClass

public void setup() {

WebDriver driver = new ChromeDriver();

}

@AfterClass

public void tearDown() {

driver.quit();

}

}

🡪Now create another class ‘TC\_LoginTest\_001’ under testcases package.  
**package** com.inetbanking.testcases;

**import** org.junit.Assert;

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

**import** com.inetbanking.pageobjects.LoginPage;

**public** **class** TC\_LoginTest\_001 **extends** BaseClass {

@Test

**public** **void** loginTest() {

*driver*.get(baseURL);

LoginPage lp = **new** LoginPage(*driver*);

lp.setUserName(username);

lp.setPassword(password);

lp.clickSubmit();

**if** (*driver*.getTitle().equals("Guru 99 Bank Manager HomePage")) {

Assert.*assertTrue*(**true**);

} **else** {

Assert.*assertTrue*(**false**);

}

}

}

🡪Run the class ‘TC\_LoginTest\_001’.