1. What are the main file components of the Automation framework built for Selenium?

Framework Built in TestNG

1. **TestNG file (.xml extension)**
   - Contains name of the test folder, classes , methods.
   - Executes of test cases in groups.
   - Allows parallel execution.
   - Include/exclude test methods during execution.
   - Allows execution of multiple test cases from more than one classes.

Framework Built in Cucumber

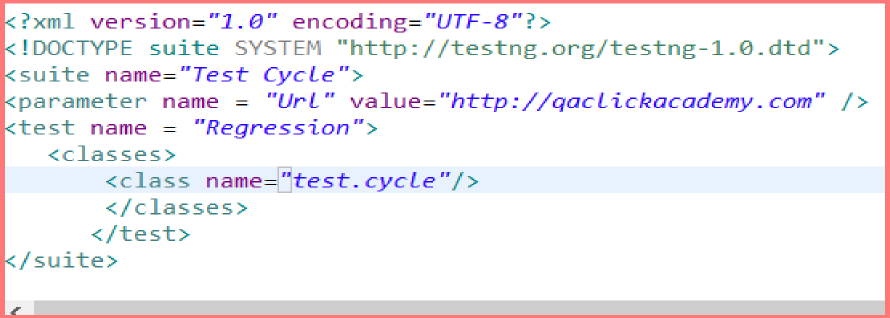
1. **Feature file (.feature extension)**
   - Contains on or many test scenarios in plain text.
   - All tests are in form of Given, When, Then, And , Feature, Background, Scenario and Scenario Outline, But.
2. **Step Definition file (.java extension)**
   - Mapping of every step of test scenarios in Feature file to the automation code.
3. **Test Runner file (.java extension)**
   - Interlink between Feature file and Step Definition file.
   - Provides option to run a single or multiple feature files.
   - Contains the location of the Feature and Step Definition file.

2.

How do you handle single data parameterization in Automation Frameworks ?

Framework Built in TestNG

testNG File



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Test Cycle">
<parameter name = "Url" value="http://qaclickacademy.com" />
<test name = "Regression">
   <classes>
       <class name="test.cycle"/>
       </classes>
      </test>
</suite>
```

Here the testNG xml file contains the site Url

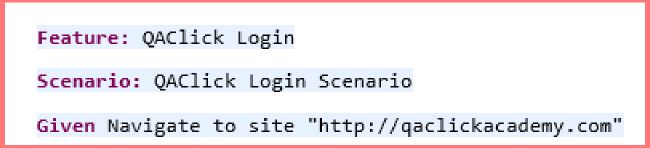
Class File

```
  @Parameter({“Url”})
     @Test
       public void Login(String url){
          driver.get(url);
      }
```

The corresponding test method in the class file

Framework Built in Cucumber

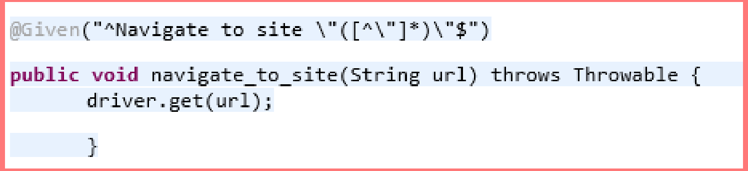
Feature File



```
Feature: QAClick Login

Scenario: QAClick Login Scenario

Given Navigate to site "http://qaclickacademy.com"
```

Here the Feature file contains the site Url.

Step Definition File



```
@Given("^Navigate to site \"([^\"]*)\"$")

public void navigate_to_site(String url) throws Throwable {
      driver.get(url);

      }
```

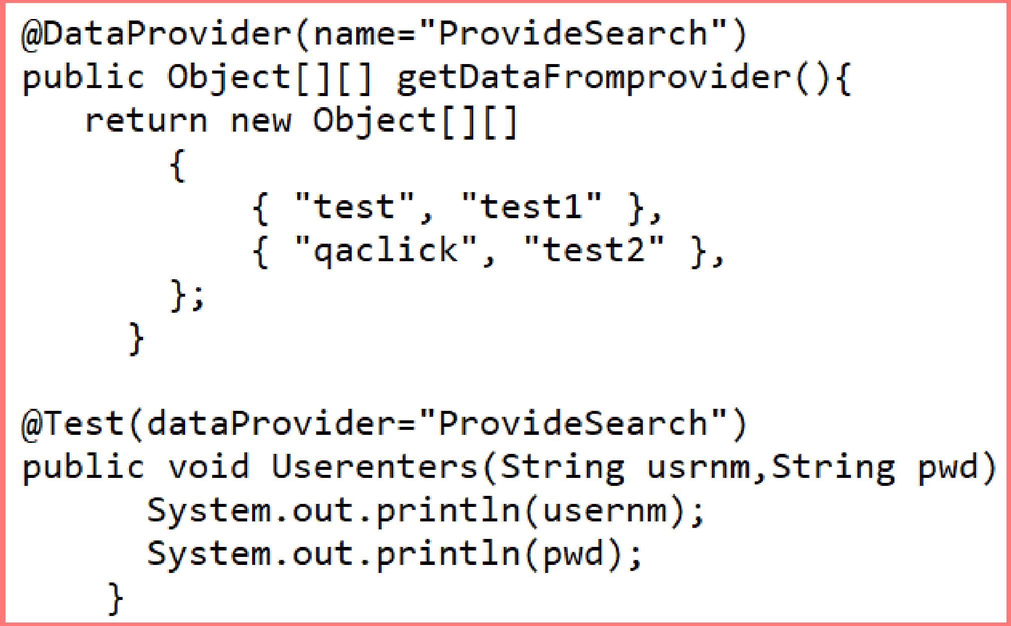
The corresponding mapping of the feature file to the step definition file.

3.

How do you handle multiple data parametrizations in Automation Frameworks ?

Framework Built in TestNG

Class File



```
@DataProvider(name="ProvideSearch")
public Object[][] getDataFromprovider(){
   return new Object[][]
       {
           { "test", "test1" },
           { "qaclick", "test2" },
       };
     }

@Test(dataProvider="ProvideSearch")
public void Userenters(String usrnm,String pwd)
      System.out.println(usernm);
      System.out.println(pwd);
    }
```

DataProvider attribute helper used for parameterizing.

Framework Built in Cucumber

Feature File

_Feature: Login Feature

Scenario Outline: Login Test

Given User is on Login Page

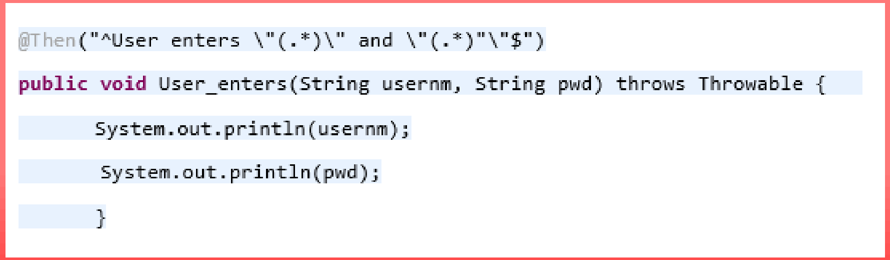
When title of login page is QAClick

Then User enters "<username>" and "<password>“

Examples:

username	password
qaclick	test#123
test	t123

Step Definition File



```
@Then("^User enters \"(.*)\" and \"(.*)"\"$")
public void User_enters(String usernm, String pwd) throws Throwable {
        System.out.println(usernm);
         System.out.println(pwd);
        }
```

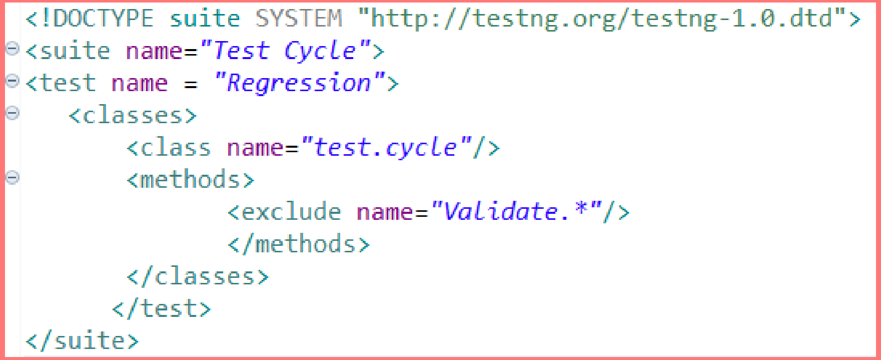
The corresponding mapping of the parametrized **Then** in feature file .

4.

How to make use of regular expressions in Automation Frameworks ?

Framework Built in TestNG

testNG File



```
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Test Cycle">
<test name = "Regression">
   <classes>
       <class name="test.cycle"/>
       <methods>
              <exclude name="Validate.*"/>
              </methods>
       </classes>
      </test>
</suite>
```

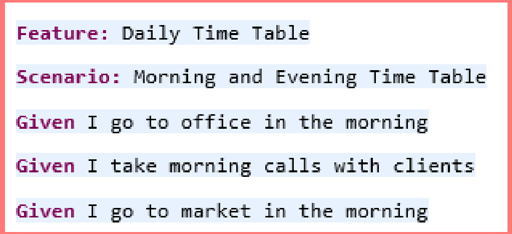
Class File

```
@Test
public void ValidateLoan(){
   System.out.println("ValidateLoan");
}
@Test
public void VerifyLoan(){
   System.out.println("VerifyLoan");
}
@Test
public void ValidateLease(){
   System.out.println("ValidateLease");
}
```

Out of the three test methods, only **VerifyLoan()** will be executed, since we have excluded all the methods starting with name **Validate** using **regular expression**.

Framework Built in Cucumber

Feature File



```
Feature: Daily Time Table

Scenario: Morning and Evening Time Table

Given I go to office in the morning

Given I take morning calls with clients

Given I go to market in the morning
```

Step Definition File

```
@Given("^I go to ([^\"]*) in the morning$")
public void goTo(String place){
   if(place.equals("office"))
        {
          System.out.println("I go to office in the morning");
        }else
        {
          System.out.println("I go to market in the morning");
        }
     }
@Given("^I take morning calls with clients$")
public void morningCalls(){
     {
        System.out.println("calls with clients");
     }
```

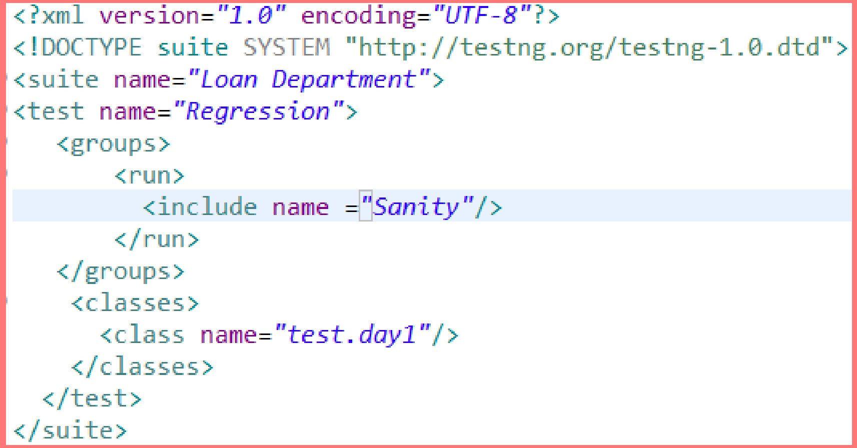
Here, **@Given("^I go to ([^\"]*) in the morning$")** maps two Given statements in Feature file using regular expression.

5.

How to run selected tests from the set of test cases in Automation Frameworks ?

Framework Built in TestNG

testNG File



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Loan Department">
<test name="Regression">
   <groups>
       <run>
         <include name ="Sanity"/>
       </run>
   </groups>
    <classes>
      <class name="test.day1"/>
    </classes>
  </test>
</suite>
```

Here the testNG xml contains **group** Sanity to be **included** in run.

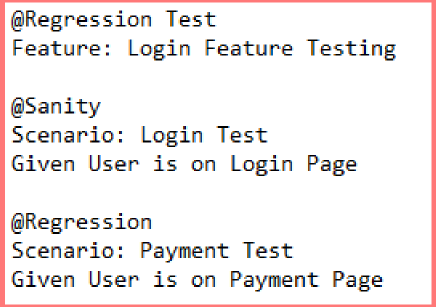
Class File

```
@Test(groups={"Sanity"})
  public void Login(){
     System.out.println("Login  is successful");
 }
```

Only the test methods having groups as **Sanity** will be executed, out of the full regression suite.

Framework Built in Cucumber

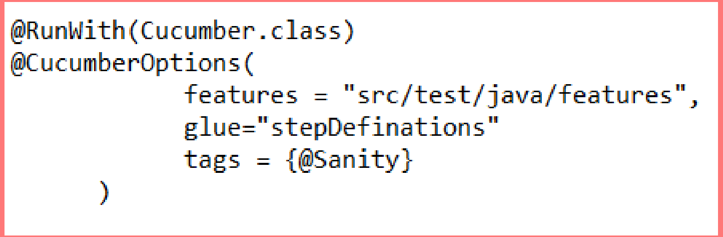
Feature File



```
@Regression Test
Feature: Login Feature Testing

@Sanity
Scenario: Login Test
Given User is on Login Page

@Regression
Scenario: Payment Test
Given User is on Payment Page
```

Here the Feature file contains the scenarios with tags Sanity and Regression.

Test Runner File



```
@RunWith(Cucumber.class)
@CucumberOptions(
            features = "src/test/java/features",
            glue="stepDefinations"
            tags = {@Sanity}
      )
```

Test Runner File contains the tag **@Sanity** hence only the scenarios with Sanity tag will executed.

6.

How to skip selected test method from the set of test cases in Automation Frameworks ?

Framework Built in TestNG

Class File

```
@Test(enabled=false)
        public void validateDate()
        {
            System.out.println("Validate date is successful);
        }
```

Here the helper attribute **enabled set to false** is used to skip a particular test method from execution.

Framework Built in Cucumber

Feature File

```
@Regression Test
Feature: Login Feature Testing

@Sanity
Scenario: Login Test
Given User is on Login Page

@Regression
Scenario: Payment Test
Given User is on Payment Page
```

Here the Feature file contains the scenarios with tags Sanity and Regression.

Test Runner File

```
@RunWith(Cucumber.class)
@CucumberOptions(
            features = "src/test/java/features",
            glue="stepDefinations"
            tags = {"~@Sanity"}
      )
```

In order to **skip the scenarios with @Sanity** , **~** is placed before @Sanity tag.

7. How to include and exclude test methods from set of test cases in Automation Framework ?

Framework Built in TestNG

testNG File

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Loan Department">
<test name="Cycle">
   <groups>
       <run>
         <include name ="Sanity"/>
         <exclude name = "Regression"/>
       </run>
   </groups>
    <classes>
      <class name="test.day1"/>
    </classes>
  </test>
</suite>
```

Here the testNG xml contains group **Sanity to be included and Regression to be excluded** in run.

Class File

```java
@Test(groups={"Sanity"})
public void Login()
       {
       System.out.println("Login successful");
       }
@Test(groups={"Regression"})
public void verfiyPayment()
       {
       System.out.println("Payment successful");
       }
```

Framework Built in Cucumber

Feature File

```
@Regression Test
Feature: Login Feature Testing

@Sanity
Scenario: Login Test
Given User is on Login Page

@Regression
Scenario: Payment Test
Given User is on Payment Page
```

Here the Feature file contains the scenarios with tags Sanity and Regression.

Test Runner File

```java
@RunWith(Cucumber.class)
@CucumberOptions(
            features = "src/test/java/features",
            glue="stepDefinations"
            tags = {"@Sanity" , "~@Regression"}
     )
```

Here the test runner file contains tag **Sanity to be included** and **Regression to be excluded** in run.

8. How to execute pre conditions and post conditions for each test method in Automation Framework?

Framework Built in TestNG

Class File

```
@BeforeMethod
public void beforeevery()
{
  System.out.println("I will execute before every test method");
}

@AfterMethod
public void afterevery()
{
  System.out.println("I will execute after  every test method");
}
@Test
public void Login()
{
  System.out.println("Login successful");
}
```

First the method with tags **@BeforeMethod** will be executed (pre condition), followed by the test method (Login()) and finally the **@AfterMethod** will be executed(post condition).

Framework Built in Cucumber

Feature File

```
Feature: Daily Time Table

Scenario: Morning and Evening Time Table

Given I go to office in the morning
```

Feature file contains the above scenario.

Step Definition File

```
@Before
public void prerequiste()
        {
        System.out.println("I will execute first");
        }
@After
public void postcondition()
        {
        System.out.println("I will execute last");
        }
@Given("^I go to office in the morning$")
public void goTo()
        {
        System.out.println("This is the scenario");
        }
```

First the method with hook **@Before** will be executed (pre condition), followed by the test method (goTo() method) and finally the **@After** will be executed(post condition).

9. How to execute pre conditions for a particular scenario in Automation Framework ?

Framework Built in TestNG

Class File

```
@Test(dependsOnMethods={"Login"})
public void ValidatePayment()
{
   System.out.println("Payment Successful");
}
@Test
public void Login()
{
   System.out.println("Login Successful");
}
 @Test
public void ValidateHistory()
{
   System.out.println("History Validated");
}
```

Here **dependsOnMethods helper attribute** is used. **ValidatePayment()** will only be executed after Login() precondition is ran successfully . But ValidateHistory() runs independently without having a pre condition test method to be executed.

Framework Built in Cucumber

Feature File

```
@FirstScenario
Scenario: This is Login test
        Given Navigate to home page
        When Enter credentials

Scenario: This is Payment test
        Given Navigate to payment page
        When Enter amount
```

Step Definition File

```
@Before("@FirstScenario")
ublic void prerequiste()
        {
        System.out.println("prerequisite");
        }

@Given("^Navigate to home page$")
ublic void nav_homepage()
        {
        System.out.println("Login Successful");
        }

@Given("^Navigate to payment page$")
ublic void nav_paymentpage()
        {
        System.out.println("Payment Successful");
        }
```

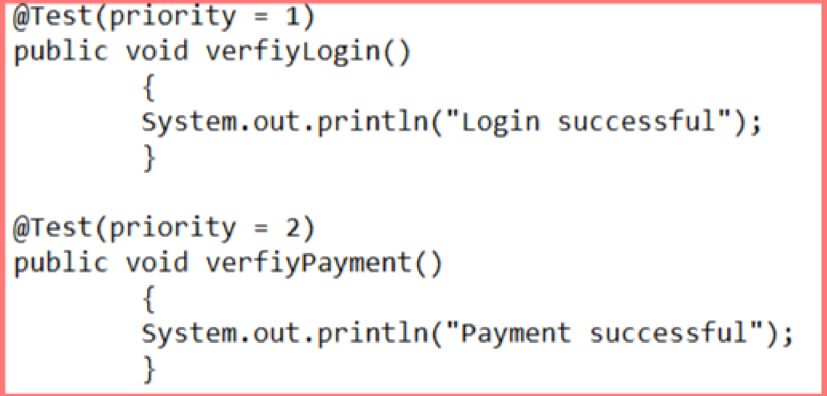
Here precondition prerequisite() has to be run successfully before nav_homepage() but nav_payment() runs independently.

10.

How to set priority for execution in Automation Framework ?

Framework Built in TestNG

Class File

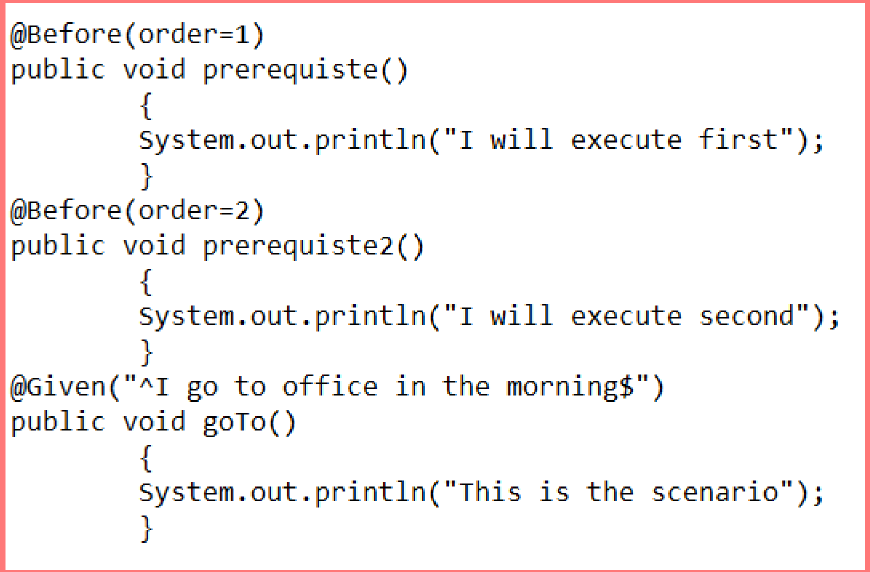


```
@Test(priority = 1)
public void verfiyLogin()
        {
        System.out.println("Login successful");
        }

@Test(priority = 2)
public void verfiyPayment()
        {
        System.out.println("Payment successful");
        }
```

Here each test method is assigned with **priority** . Test method with **lower priority** will be executed first followed by the test method having **higher priority**. verifyLogin() will be executed first then verifyPayment().

Framework Built in Cucumber

Step Definition File



```
@Before(order=1)
public void prerequiste()
        {
        System.out.println("I will execute first");
        }
@Before(order=2)
public void prerequiste2()
        {
        System.out.println("I will execute second");
        }
@Given("^I go to office in the morning$")
public void goTo()
        {
        System.out.println("This is the scenario");
        }
```

Here each test method is assigned with **order**. Test method with **lower order** (prerequisite()) will executed first . Then prerequisite2() test method will be executed which is having a higher order.

Once these pre conditions get executed successfully test method goTo() will be executed.

11.

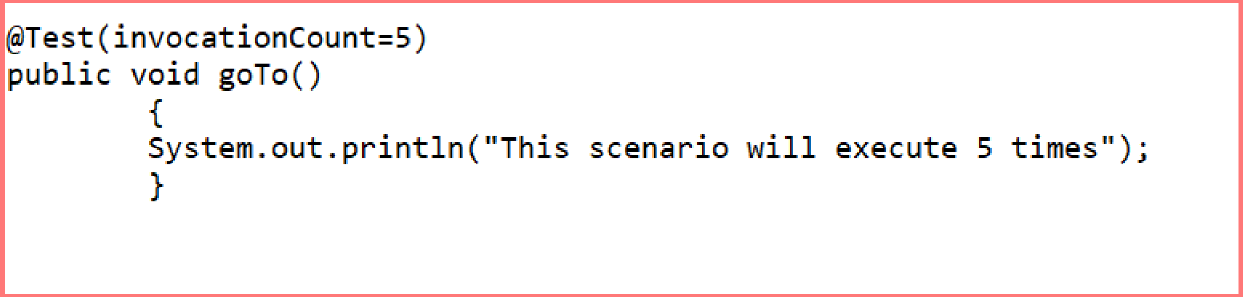
What are the different annotations available in TestNG?

1. @Test
2. @BeforeSuite
3. @AfterSuite
4. @BeforeClass
5. @AfterClass
6. @BeforeTest
7. @AfterTest
8. @BeforeGroups
9. @AfterGrroups
10. @BeforeMethod
11. @AfterMethod

12.

What is invocation count in TestNG?

If we are required to execute a test case **N** number of times , then **invocationCount helper attribute** is used.



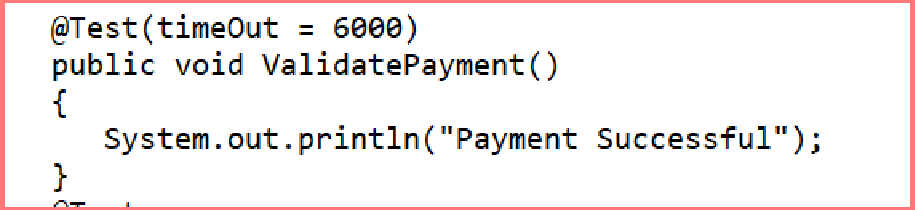
```
@Test(invocationCount=5)
public void goTo()
        {
        System.out.println("This scenario will execute 5 times");
        }
```

In the above example , **This scenario will execute 5 times** will print on the console five times.

13.

What is timeOut in TestNG?

If there any test method that consumes a **lot of time for execution** ,we can **terminate** that method with the help of **timeOut** helper attribute in TestNG.



```
@Test(timeOut = 6000)
public void ValidatePayment()
{
   System.out.println("Payment Successful");
}
```

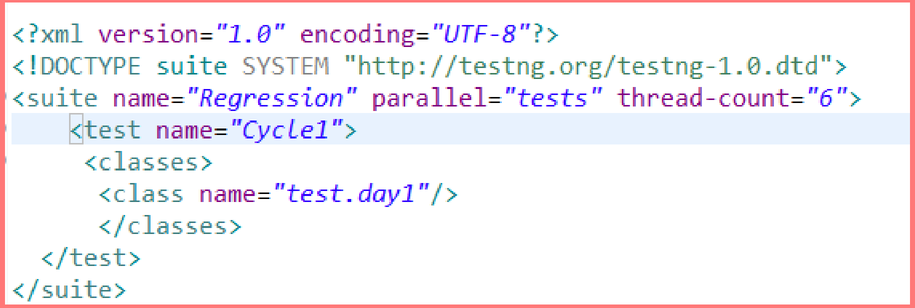
After 6000ms , the test method will be terminated and will be marked as **Failed**.

14.

How to achieve parallel execution in TestNG?

We use **parallel** attribute in testng.xml to achieve parallel execution in TestNG. It also has a parameter called as **thread-count .**The parallel can have the following values:

1. Tests
2. Classes
3. Methods
4. Instances



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Regression" parallel="tests" thread-count="6">
    <test name="Cycle1">
     <classes>
      <class name="test.day1"/>
      </classes>
  </test>
</suite>
```

In the above example , execution will happen in parallel mode for **tests ,** having **thread- count** of 6.

15.

Explain the advantages of Cucumber.

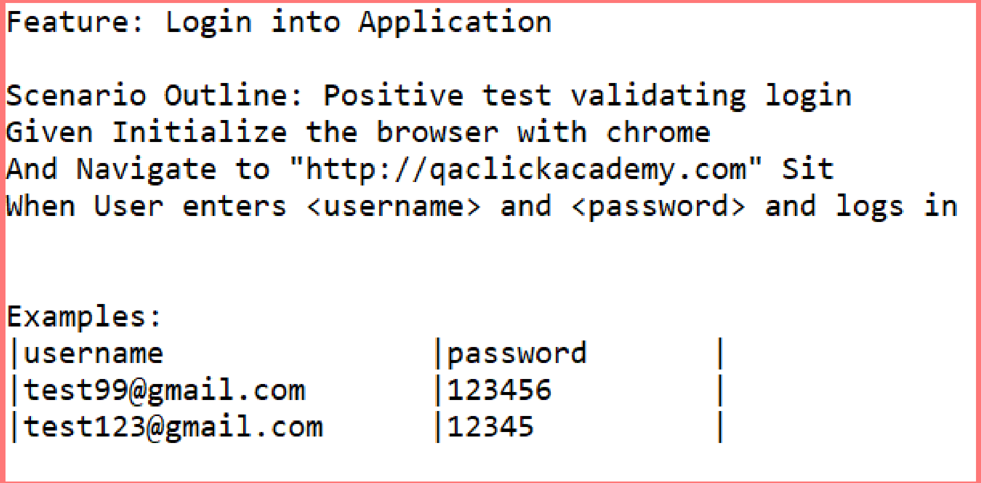
Some of the advantages of cucumber are as the followings:

1. Involves the business stake holders who are not having coding knowledge.
2. Its is an open source tool.
3. Integration with tools like Eclipse is easy.
4. It reduces communication gaps among Business Analyst, Developers and QA in an Agile environment.
5. Plain text representation makes it easy to understand for non technical individuals.
6. It is easy to maintain and work with.

16.

What is Scenario Outline in Cucumber ?

Scenario Outline is the same scenario being executed for multiple data in multiple combinations. The data set is represented in form of table separated by (||). Each row constitute a set of data.



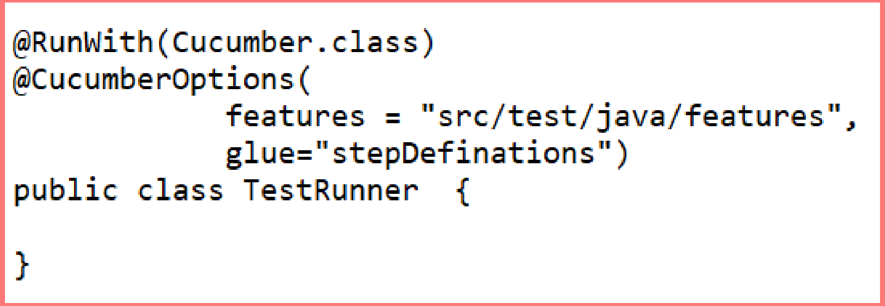
```
Feature: Login into Application

Scenario Outline: Positive test validating login
Given Initialize the browser with chrome
And Navigate to "http://qaclickacademy.com" Sit
When User enters <username> and <password> and logs in


Examples:
|username               |password       |
|test99@gmail.com       |123456         |
|test123@gmail.com      |12345          |
```

In the above example , **When** having <username> and <password> will be executed with three combinations of data defined under **Examples** separated by (||).

17.

What is the use of glue in Cucumber Options tag ?

Cucumber Identifies the **location of the step definition files** from the glue property in Cucumber Options tags.

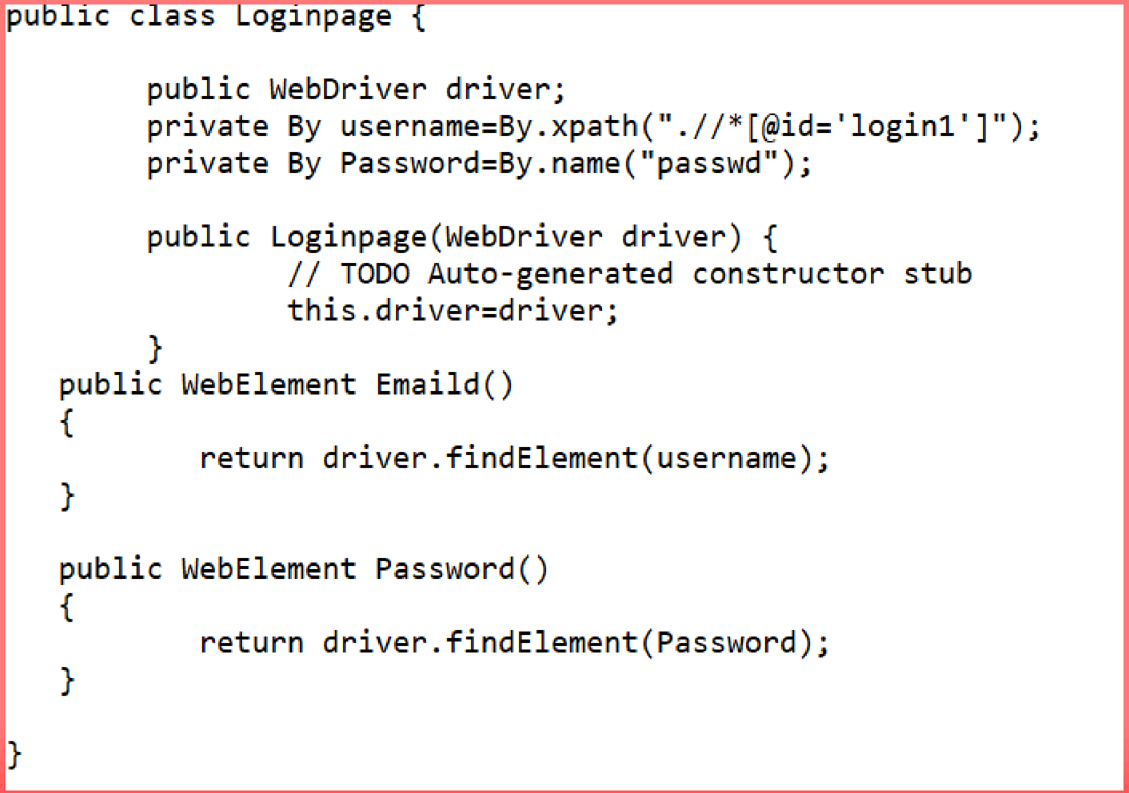


```
@RunWith(Cucumber.class)
@CucumberOptions(
            features = "src/test/java/features",
            glue="stepDefinations")
public class TestRunner  {

}
```

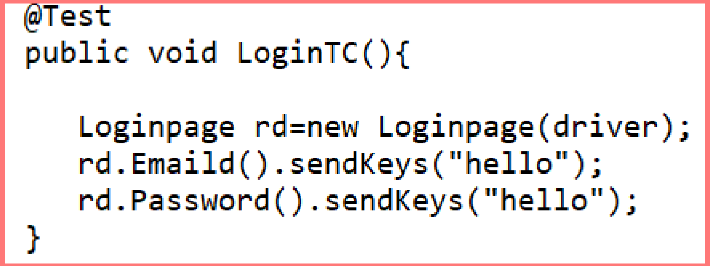
18.

How will we achieve encapsulation in our Automation Framework ?

Encapsulation means enclosing anything in a container . It is done to hide details and it is achieved by the help of access modifiers like **public , protected and private.**



```
public class Loginpage {

        public WebDriver driver;
        private By username=By.xpath(".//*[@id='login1']");
        private By Password=By.name("passwd");

        public Loginpage(WebDriver driver) {
                // TODO Auto-generated constructor stub
                this.driver=driver;
        }
   public WebElement Emaild()
   {
           return driver.findElement(username);
   }

   public WebElement Password()
   {
           return driver.findElement(Password);
   }

}
```



```
@Test
public void LoginTC(){

   Loginpage rd=new Loginpage(driver);
   rd.Emaild().sendKeys("hello");
   rd.Password().sendKeys("hello");
}
```

Here we have declared the variables as private and methods accessing them as public. Thus , while we are creating the objects , we are not handling them directly but with the help of the methods , thus keeping variables intact.

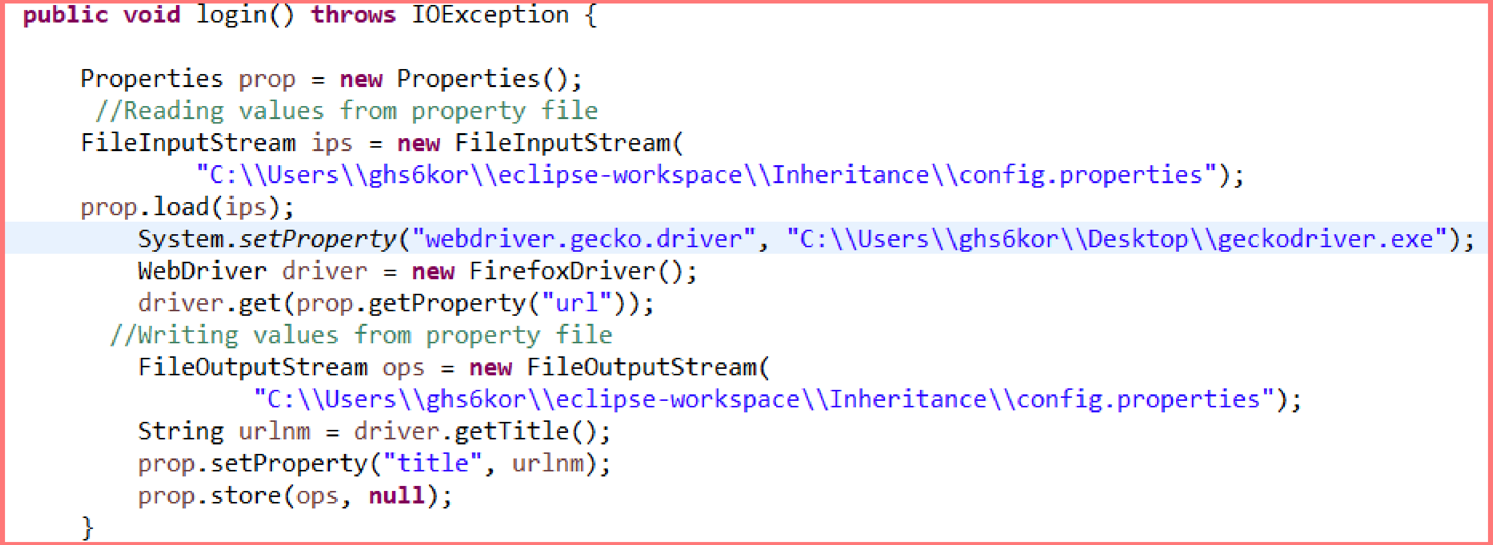
.

19.

How will we declare global variables in Automation Framework ?

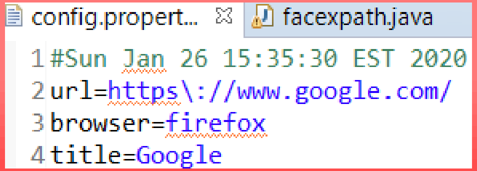
We can declare global variables in Automation Framework with the help of the **Properties** class and a **properties** file . The properties file contains global variables in the form of **key – value pairs**. We can read and write values in the .properties file.

Class File



```
public void login() throws IOException {

    Properties prop = new Properties();
     //Reading values from property file
    FileInputStream ips = new FileInputStream(
            "C:\\Users\\ghs6kor\\eclipse-workspace\\Inheritance\\config.properties");
    prop.load(ips);
        System.setProperty("webdriver.gecko.driver", "C:\\Users\\ghs6kor\\Desktop\\geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get(prop.getProperty("url"));
      //Writing values from property file
        FileOutputStream ops = new FileOutputStream(
                "C:\\Users\\ghs6kor\\eclipse-workspace\\Inheritance\\config.properties");
        String urlnm = driver.getTitle();
        prop.setProperty("title", urlnm);
        prop.store(ops, null);
    }
```

Properties File



config.propert... facexpath.java

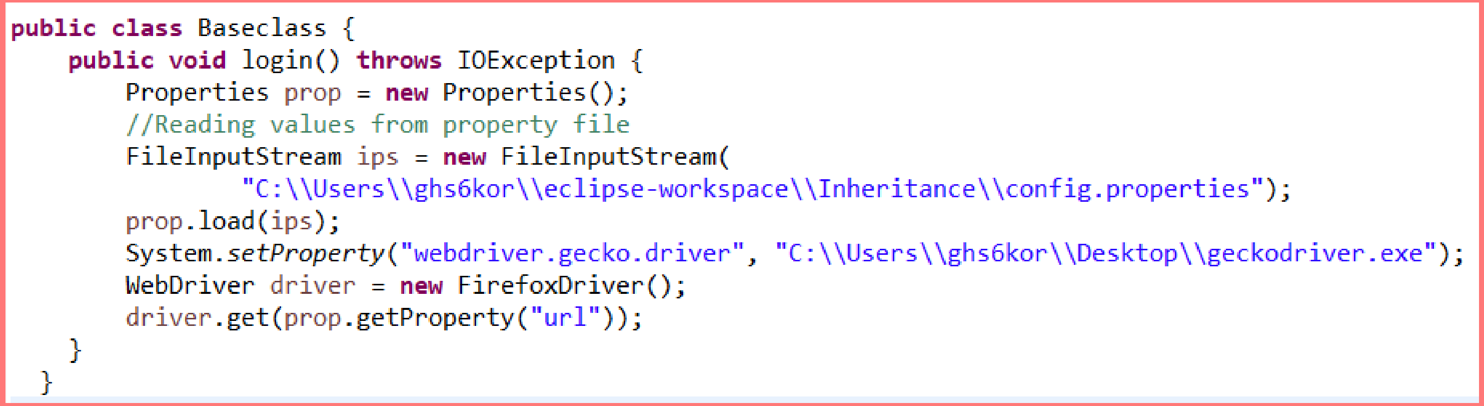
```
#Sun Jan 26 15:35:30 EST 2020
url=https\://www.google.com/
browser=firefox
title=Google
```

The **title** key writes the value **Google** in the properties file. It reads the keys **url**.

20. How do you deal with reusable components in Automation Framework ?

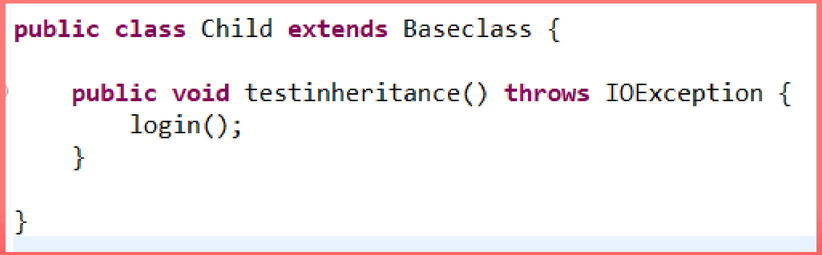
We can deal with the reusable components in Automation Framework with the help of **inheritance** concept . It’s a **parent child relationship** where the child inherits the properties of the parent class.

Parent Class



```
public class Baseclass {
    public void login() throws IOException {
        Properties prop = new Properties();
        //Reading values from property file
        FileInputStream ips = new FileInputStream(
                "C:\\Users\\ghs6kor\\eclipse-workspace\\Inheritance\\config.properties");
        prop.load(ips);
        System.setProperty("webdriver.gecko.driver", "C:\\Users\\ghs6kor\\Desktop\\geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get(prop.getProperty("url"));
    }
  }
```

Child Class



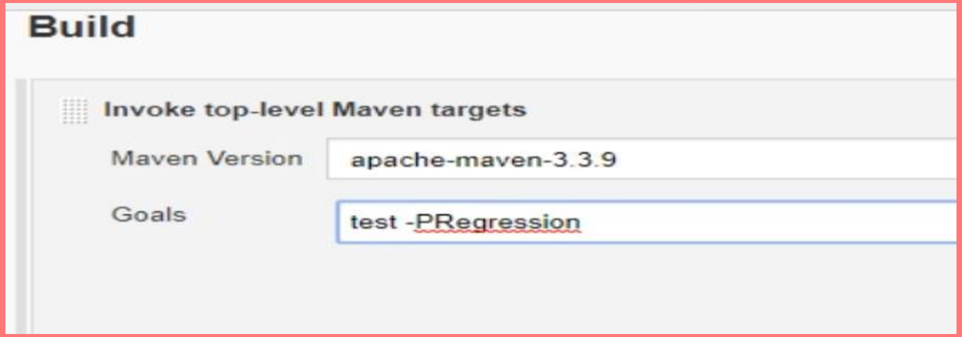
```
public class Child extends Baseclass {

    public void testinheritance() throws IOException {
        login();
    }

}
```

Here the Child class **extends** the Baseclass and calls the login() method from it thereby achieving reusability of code.

21.

How to run Automation Framework from Jenkins ?

We can run Automation Framework from Jenkins with the help of Maven commands. We need to first configure Java and Maven paths in Jenkins. Then while creating the Jenkins jobs , the select **Invoke top-level Maven targets** from the Build section.



Build

Invoke top-level Maven targets

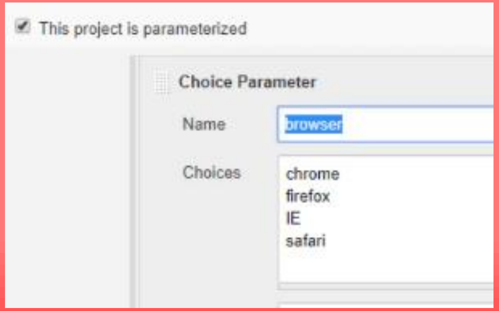
Maven Version: apache-maven-3.3.9

Goals: test -PRegression

Here in the Goals , we need to provided the Maven commands.

test –PRegression is the specific command to trigger build without parameters.

Parameterize Jenkins Build

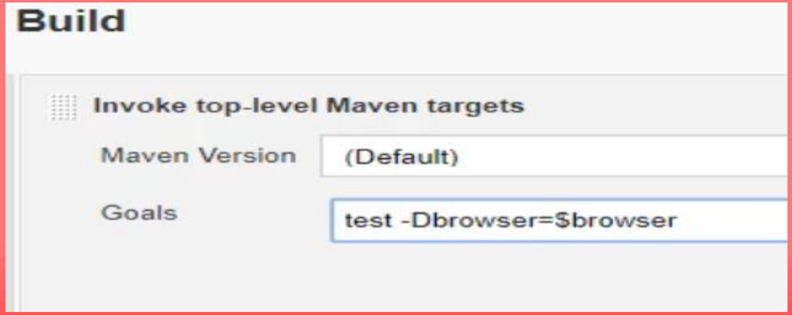


✓ This project is parameterized

Choice Parameter

Name: browser

Choices:
chrome
firefox
IE
safari



Build

Invoke top-level Maven targets

Maven Version: (Default)

Goals: test -Dbrowser=$browser

Here first the parameters are defined in Jenkin jobs. Then the Maven Goals is set with **test –Dbrowser=$browser**, with parameter **browser.**

22.

How do we arrange locators in Automation Framework ?

We can arrange locators in Automation Framework in Page Object Model in the following way:

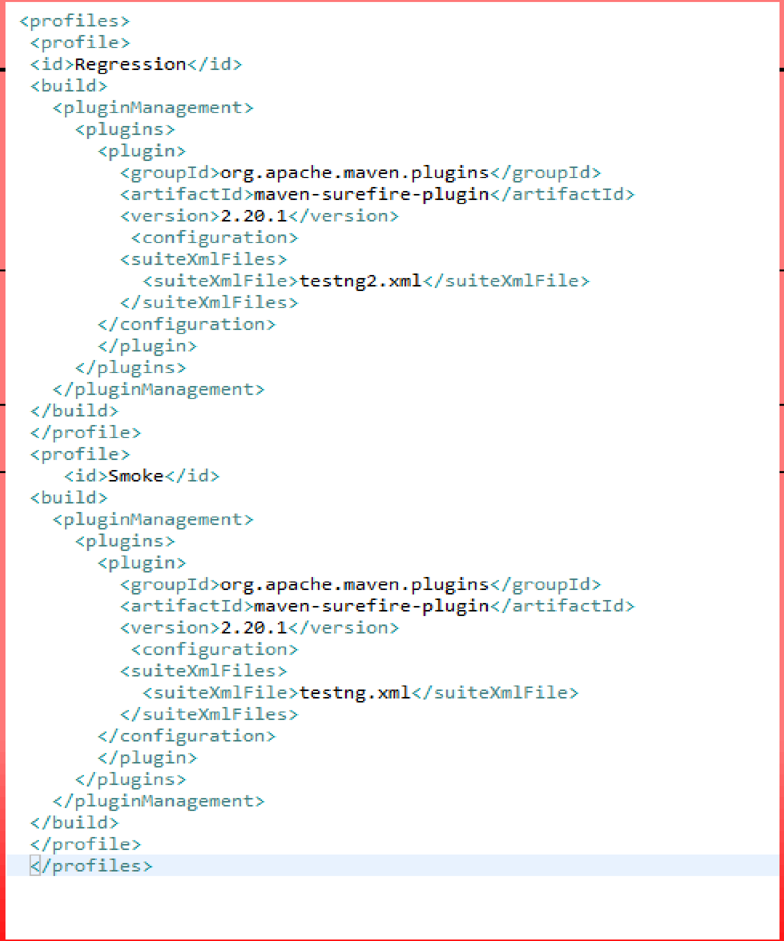
- Segregate locators for a particular screen in separate Java class. For example , LoginPage java class will contain objects with respect to that screen. While the PaymentPage java class will contain objects with respect to that screen.
- The test cases are maintain in different Java classes where we will use the locators by calling the objects from that particular page (for example , LoginPage) to the test case.
- In case of any locator needs to be updated/deleted , we need to modify them to that particular java class for that specific screen and not every where.
- Easy to use and maintain.

23.

Which is the Maven command to achieve profiling?

We can do profiling in Maven using the command : **mvn test –PSmoke**

Pom file



```xml
<profiles>
 <profile>
 <id>Regression</id>
 <build>
   <pluginManagement>
     <plugins>
       <plugin>
         <groupId>org.apache.maven.plugins</groupId>
         <artifactId>maven-surefire-plugin</artifactId>
         <version>2.20.1</version>
          <configuration>
         <suiteXmlFiles>
           <suiteXmlFile>testng2.xml</suiteXmlFile>
         </suiteXmlFiles>
       </configuration>
       </plugin>
     </plugins>
   </pluginManagement>
 </build>
 </profile>
 <profile>
    <id>Smoke</id>
 <build>
   <pluginManagement>
     <plugins>
       <plugin>
         <groupId>org.apache.maven.plugins</groupId>
         <artifactId>maven-surefire-plugin</artifactId>
         <version>2.20.1</version>
          <configuration>
         <suiteXmlFiles>
           <suiteXmlFile>testng.xml</suiteXmlFile>
         </suiteXmlFiles>
       </configuration>
       </plugin>
     </plugins>
   </pluginManagement>
 </build>
 </profile>
</profiles>
```

Here we have defined two profiles with Id **Regression** and **Smoke**. With the command , given above we will be able to trigger the test cases with profile id **Smoke** pointing to the **testng.xml** file. In this way , we can have N number of TestNG xml files driven by a single pom file.

Similarly , to trigger the execution with profile Id Regression , Maven command is : **mvn test –PRegression .** In this case , **testng2.xml** file will be pointed.

24.

How to capture Screenshot automatically on Test Failure in the Framework

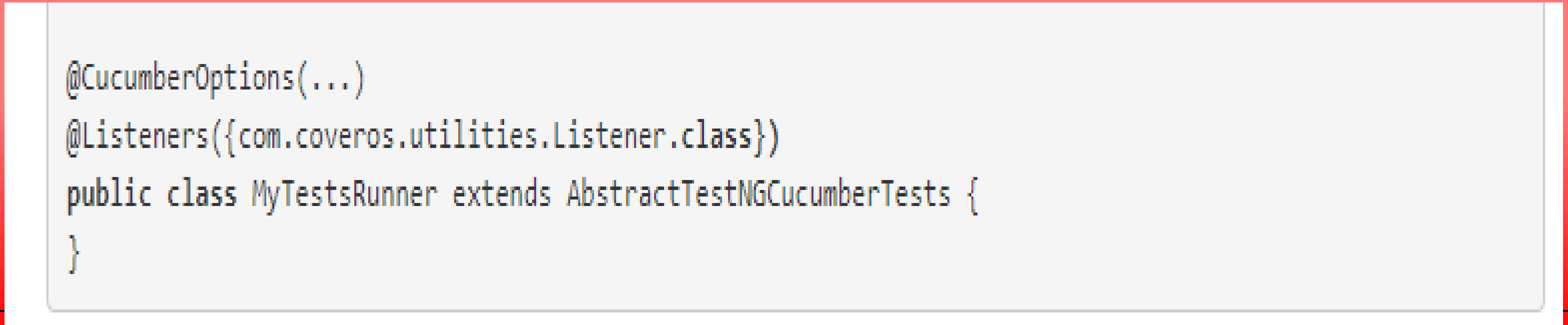
Using TestNG Listeners

- **public void onTestFailure(ITestResult result) {**

// Selenium Screenshot Method

}

Cucumber do not have Listeners by default. We should Integrate it with TestNG to enjoy the features of Cucumber



```
@CucumberOptions(...)
@Listeners({com.coveros.utilities.Listener.class})
public class MyTestsRunner extends AbstractTestNGCucumberTests {
}
```

25.

How do you manage code when multiple people in the team contributing for the Framework

Using Version Control Tools like

GIT

SVN