1. Configure testing in eclipse

Go to help in eclipse 🡪 Click on Eclipse Market place 🡪 Go to Find 🡪 Type TestNG 🡪 Click Install

1. beforeclass, beforetest, test, dataprovider, afterclass, aftertest, beforesuite, aftersuite, parameters.

package frame;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

public class Annotations {

@Test

public void testCase1() {

System.out.println("This is the Test Case 1");

}

@Test

public void testCase2() {

System.out.println("This is the Test Case 2");

}

@BeforeMethod

public void beforeMethod() {

System.out.println("This will execute before every Method");

}

@AfterMethod

public void afterMethod() {

System.out.println("This will execute after every Method");

}

@BeforeClass

public void beforeClass() {

System.out.println("This will execute before the Class");

}

@AfterClass

public void afterClass() {

System.out.println("This will execute after the Class");

}

@BeforeTest

public void beforeTest() {

System.out.println("This will execute before the Test");

}

@AfterTest

public void afterTest() {

System.out.println("This will execute after the Test");

}

@BeforeSuite

public void beforeSuite() {

System.out.println("This will execute before the Test Suite");

}

@AfterSuite

public void afterSuite() {

System.out.println("This will execute after the Test Suite");

}

}

1. Configure testng.xml for creating test suites, specifying test classes, parameters

**package** frame;

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

**public** **class** AddTestCase {

@Test

**public** **void** addLocationTestCase() {

System.***out***.println("Im in add location test case");

}

@Test

**public** **void** addDepartmentTestCase() {

System.***out***.println("Im in add department test case");

}

@Test

**public** **void** addEmployeeTestCase() {

System.***out***.println("Im in add employee test case");

}

}

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name=*"Sample Test Suite"* verbose=*"1"* >

<test name=*"Method Test Cases"* >

<classes>

<class name=*"frame.AddTestCase"*>

<methods>

<include name=*"addLocationTestCase"* />

<include name=*"addDepartmentTestCase"* />

<exclude name=*"addEmployeeTestCase"* />

</methods>

</class>

</classes>

</test>

</suite>

1. How to use include, exclude and parallel methods

package frame;

import org.testng.annotations.Test;

public class AddTestCase {

@Test

public void addLocationTestCase() {

System.out.println("Im in add location test case");

}

@Test

public void addDepartmentTestCase() {

System.out.println("Im in add department test case");

}

@Test

public void addEmployeeTestCase() {

System.out.println("Im in add employee test case");

}

}

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name="Sample Test Suite" verbose="1" >

<test name="Method Test Cases" >

<classes>

<class name="frame.AddTestCase">

<methods>

<include name="addLocationTestCase" />

<include name="addDepartmentTestCase" />

<exclude name="addEmployeeTestCase" />

</methods>

</class>

</classes>

</test>

</suite>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name="Sample Test Suite" parallel="methods" verbose="1" >

<test name="Method Test Cases" >

<classes>

<class name="frame.AddTestCase">

</class>

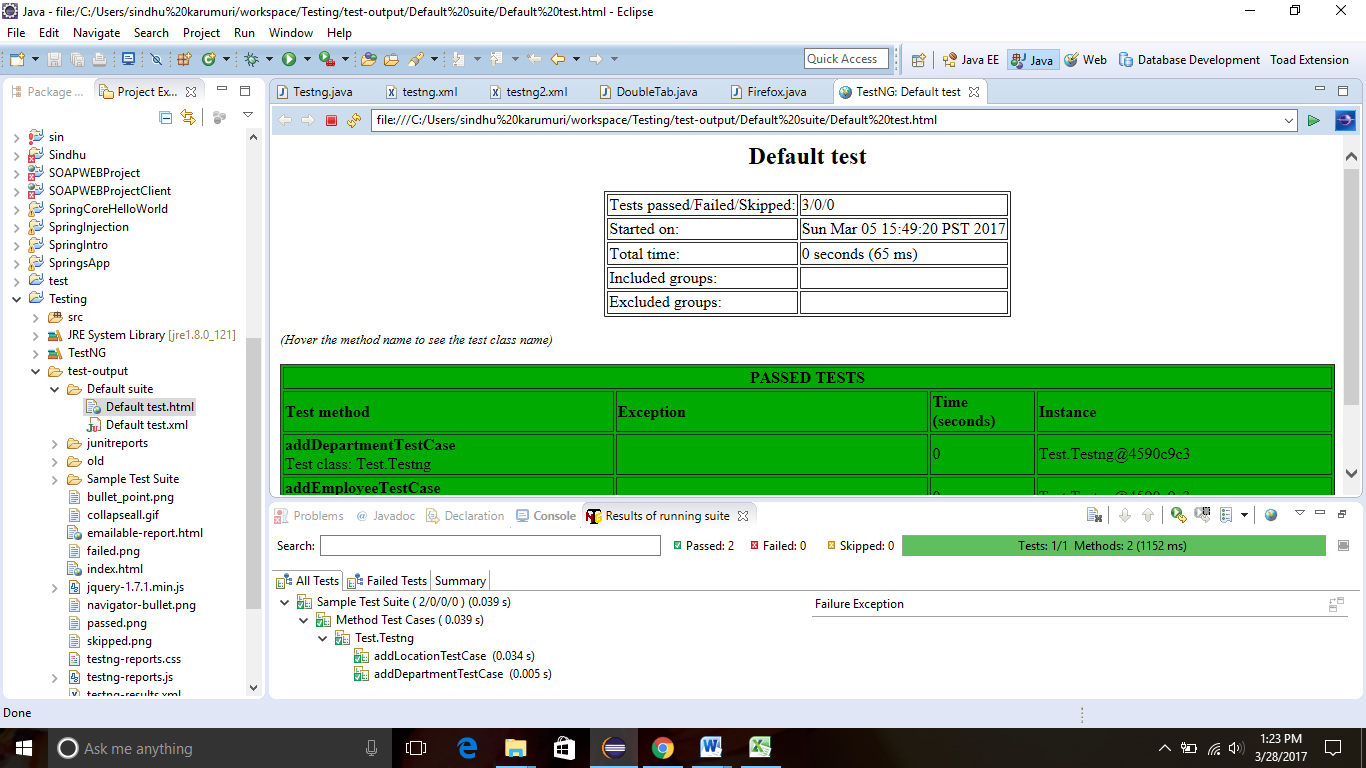
</classes>

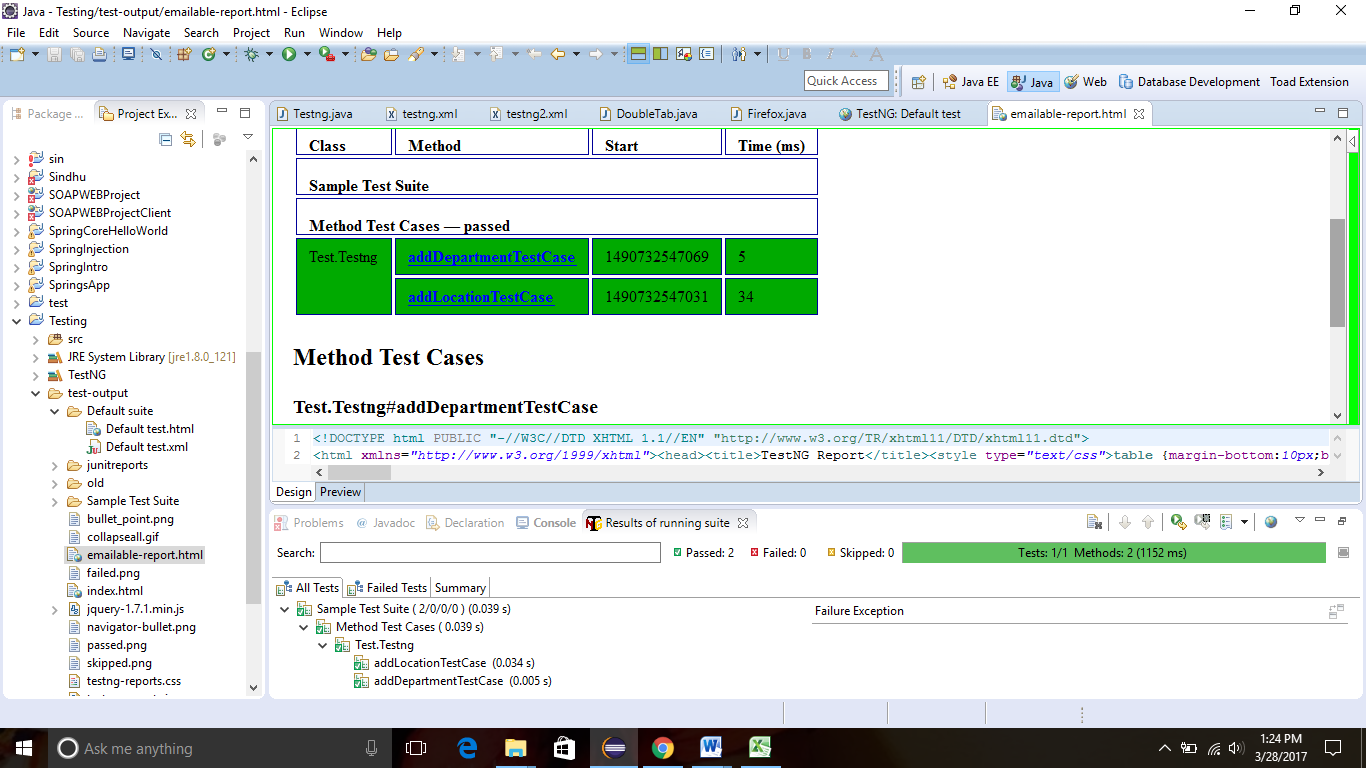
</test>

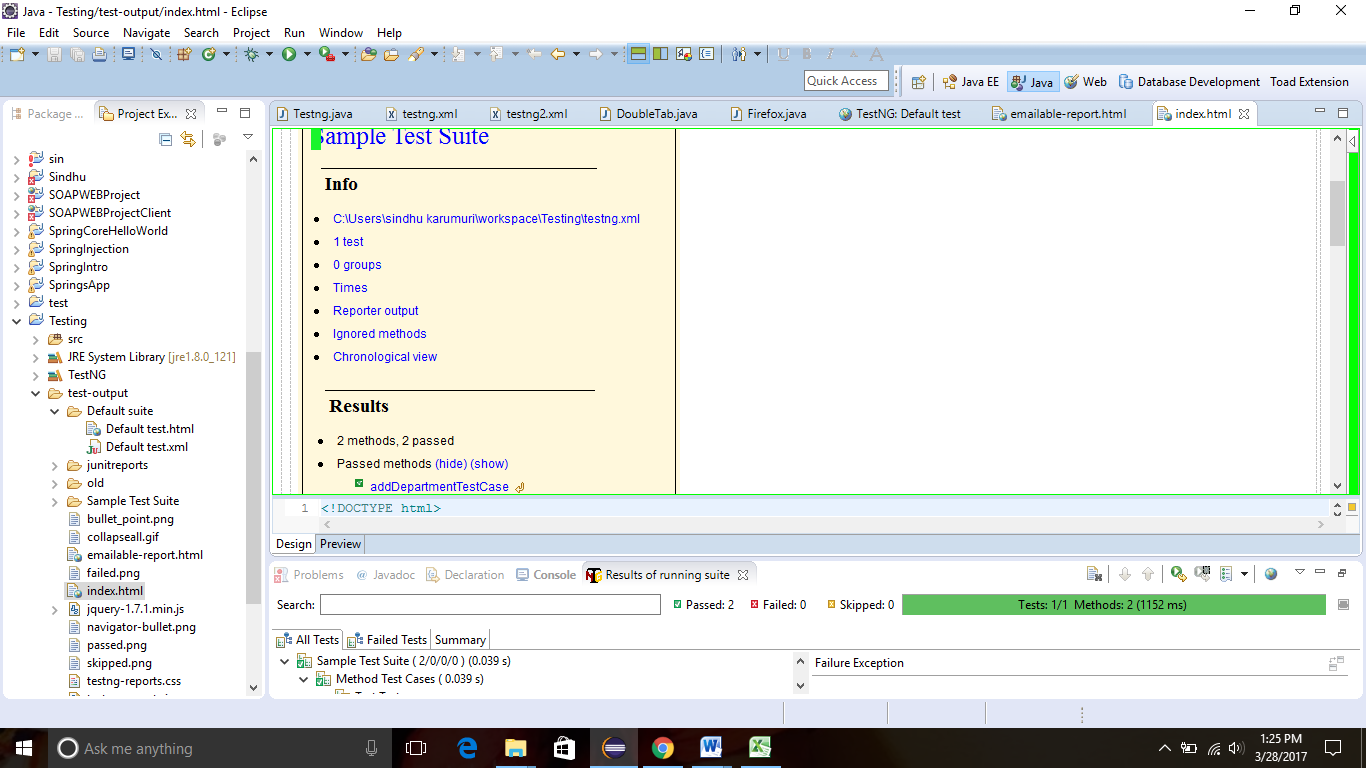
</suite>

1. TestNG report generation.

I have created test reports for the above program. Below are console report and html reports.







1. Using Listeners for report generation (itestlistener, isuitelistener)

package frame;

import org.testng.IClass;

import org.testng.ITestResult;

import org.testng.TestListenerAdapter;

public class ListenerClass extends TestListenerAdapter {

@Override

public void onTestStart(ITestResult tr) {

log("Test Started....");

}

@Override

public void onTestSuccess(ITestResult tr) {

log("Test '" + tr.getName() + "' PASSED");

log(tr.getTestClass());

log("Priority of this method is " + tr.getMethod().getPriority());

System.out.println(".....");

}

@Override

public void onTestFailure(ITestResult tr) {

log("Test '" + tr.getName() + "' FAILED");

log("Priority of this method is " + tr.getMethod().getPriority());

System.out.println(".....");

}

@Override

public void onTestSkipped(ITestResult tr) {

log("Test '" + tr.getName() + "' SKIPPED");

System.out.println(".....");

}

private void log(String methodName) {

System.out.println(methodName);

}

private void log(IClass testClass) {

System.out.println(testClass);

}

}

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >

<suite name=*"Log Suite Example"* verbose=*"1"*>

<listeners>

<listener class-name=*"frame.ListenerClass"* />

</listeners>

<test name=*"TestNG logs sample"* preserve-order=*"true"*>

<classes>

<class name=*"frame.LoggingClass"*>

<methods>

<include name=*"methodAddingNumbers"* />

<include name=*"dividedByZero"* />

<include name=*"methodSkip"* />

</methods>

</class>

</classes>

</test>

</suite>

1. Page Factory

**package** frame;

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

**public** **class** GoogleSearchPage {

**private** WebElement q;

**public** **void** searchFor(String text) {

q.sendKeys(text);

q.submit();

}

}

package frame;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.PageFactory;

public class UsingGoogleSearchPage {

public static void main(String[] args) {

// Create an instance of a driver

System.setProperty("webdriver.chrome.driver","C:\\Selenium\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

// Navigate to the right place

driver.get("http://www.google.com/");

GoogleSearchPage page = PageFactory.initElements(driver, GoogleSearchPage.class);

// And now do the search.

page.searchFor("Hi");

}

}

1. Create Maven Project

Configuring Maven in Eclipse:

Open Eclipse 🡪 Go to Help 🡪 Click on Install new software 🡪 Available software window gets opened 🡪 Give the path 🡪 Add 🡪 Click on Finish.

Open Eclipse 🡪 Go to File 🡪 Open New🡪 Go to new project 🡪 Go to Other 🡪 Select New Maven Project 🡪 Click on Finish

1. Include class for reading data from excel file under proper package

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.util.concurrent.TimeUnit;

import org.apache.poi.xssf.usermodel.XSSFSheet;

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

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Testcase3 {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver","C:\\Selenium\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

driver.get("http://www.google.com");

WebElement searchbox = driver.findElement(By.name("q"));

try{

FileInputStream file = new FileInputStream(new File("C:\\Users\\User\\Desktop\\input.xlsx"));

XSSFWorkbook workbook = new XSSFWorkbook(file);

XSSFSheet sheet = workbook.getSheetAt(0);

for(int i=0; i<= sheet.getLastRowNum(); i++){

String keyword = sheet.getRow(i).getCell(0).getStringCellValue();

searchbox.sendKeys(keyword);

searchbox.submit();

driver.manage().timeouts().implicitlyWait(10000, TimeUnit.MILLISECONDS);

}

workbook.close();

file.close();

}

catch(FileNotFoundException fnfe){

fnfe.printStackTrace();

}

catch (IOException ioe) {

ioe.printStackTrace();

}

}

}