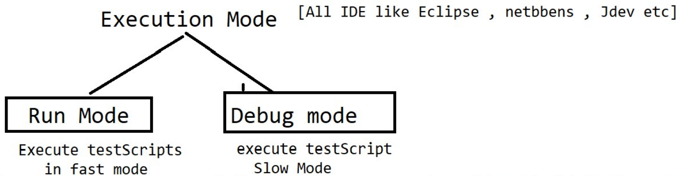
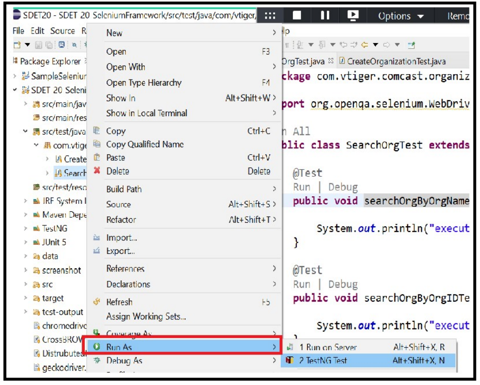
What is Debugging ?

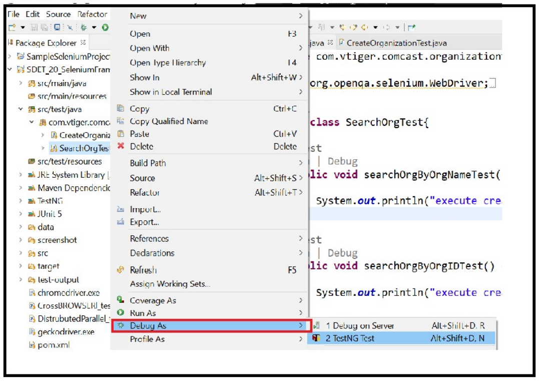
It’s one of the Execution mode in all IDE(Eclipse) , used to run the test scripts in Slow Mode& wait for user instruction

Note: Break point is mandatory to run the testscript in Debug mode


Execution Mode
[All IDE like Eclipse , netbbens , Jdev etc]
Run Mode
Debug mode
Execute testScripts in fast mode
execute testScript Slow Mode



```
ckage com.vtiger.comcast.organiz

port org.openqa.selenium.WebDriv

n All
blic class SearchOrgTest extends

  @Test
  Run | Debug
  public void searchOrgByOrgName

      System.out.println("execut
  }

  @Test
  Run | Debug
  public void searchOrgByOrgIDTe

      System.out.println("execut
```



```
 com.vtiger.comcast.organization

org.openqa.selenium.WebDriver;

class SearchOrgTest{

st
 | Debug
lic void searchOrgByOrgNameTest(

  System.out.println("execute cre

st
 | Debug
lic void searchOrgByOrgIDTest()

  System.out.println("execute cre
```

Why Debugging mode is required ?

➔ When Jenkins execute 1000 of test scripts in a batch , there are few testscripts got failed due to some issue , in order to find root cause (exact reason) of the issue we go for debug mode

➔ In Test Scripts development also will use debug mode to the find the reason of the issue

, because selenium all ways run test scripts from the beginning

➔We can understand the framework execution flow, even though is verify complicated

➔We can execute piece of Program in “expression -Winodw”

=➔ In debug mode , we can see the variable & object values in runtime using “variables=window”

List out theEclipse Options in Debug mode ?

BreakPoint(Ctrl + Swift + B) : used to pause the execution in runtime & wait for user instruction

Step Into (F5) : control get inside the calling function from called Function , & pause the execution inside the first line of the calling function

Step Over (F6) : execute line by line

Step Return (F7) : control get out of the calling function by executing remaining steps in fast mode & control pause the execution where its being called

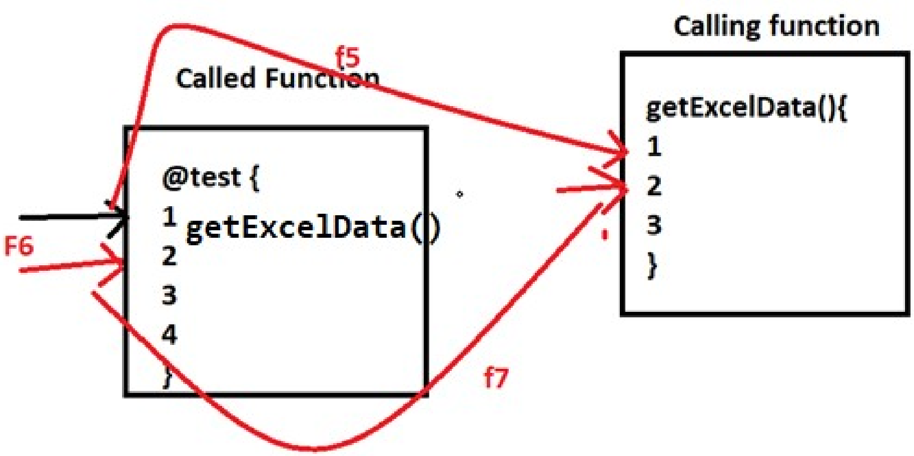
Resume (F8) : execute test in fast mode from one break point to another breakpoint

Ctl+ F2 : terminate the execution

Two Debugging Window

Expression Window : we can execute piece of program in expressions window by modifying the Code in runtime

Variable Window : where we can see all the variable / object value in run time



When Ever we 1000 of test Scripts few test got failed, what is your approach towards the failures?

1. Look at the html Report which sent by Jenkins to check which test scripts is getting failed
2. Look the exception & line number of the failure
3. Look the screenshot of the failure
4. Put break point the specific step, where we you have suspect
5. Run the test scripts in debug mode
6. Find the root cause of the issue
7. If issue with respect to product, raise defect in Jira or, if issue with respect to test scripts modify the scripts

Why ScreenShot in Automation ?

In Automation is help us to understand the root cause of the test script issue or product issue partially

What is the major reason of the test Scripts failure in automation ?

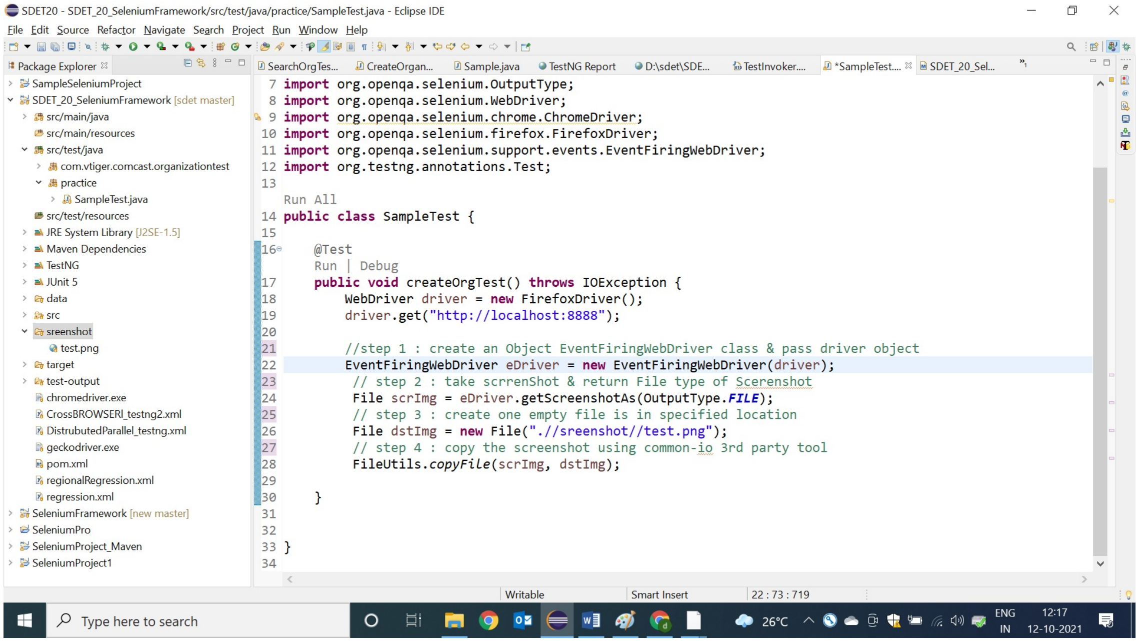
There are 3 reason for the test scripts failure

1. Xpath or locator issue
2. Defect in application
3. Synchronization issue

In real time when ever testscript got failed , what is your approach to find the issue?

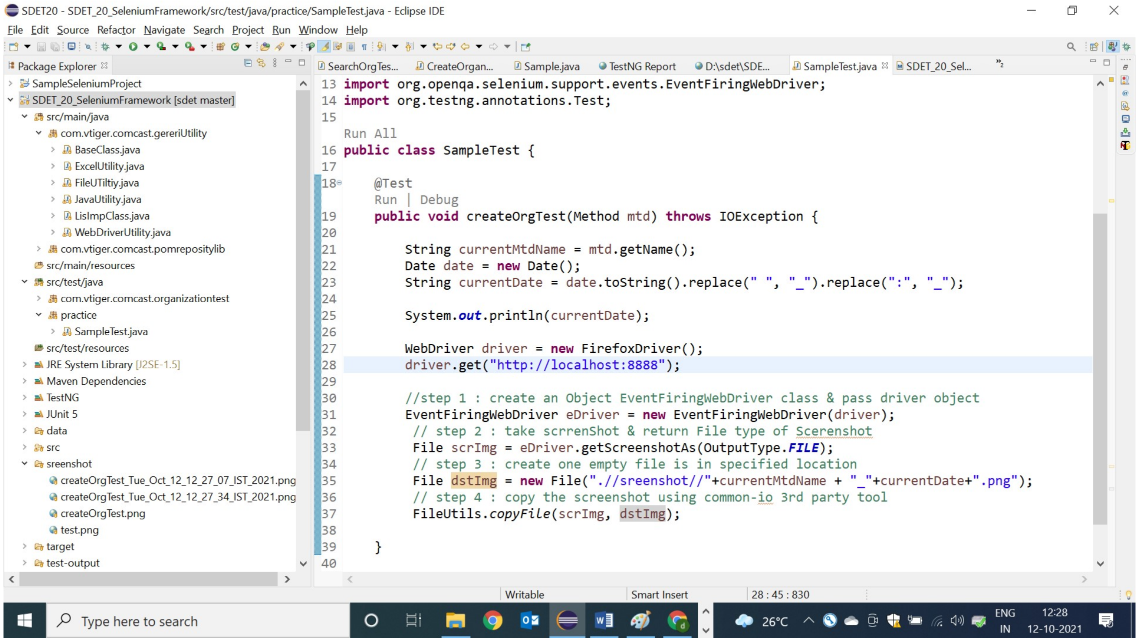
1. Get in to test scripts console logs / Extend-Report logs
2. Look the Exception & line number
3. Look at the screenshot , will get to know reason of the failure partially
4. Get inside test scripts , put break point
5. Run the test script in debug mode
6. Identify the root cause of the issue

How to take a Screenshot?



```java
import org.openqa.selenium.OutputType;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.events.EventFiringWebDriver;
import org.testng.annotations.Test;

Run All
public class SampleTest {

    @Test
    Run | Debug
    public void createOrgTest() throws IOException {
        WebDriver driver = new FirefoxDriver();
        driver.get("http://localhost:8888");

        //step 1 : create an Object EventFiringWebDriver class & pass driver object
        EventFiringWebDriver eDriver = new EventFiringWebDriver(driver);
         // step 2 : take scrrenShot & return File type of Scerenshot
         File scrImg = eDriver.getScreenshotAs(OutputType.FILE);
         // step 3 : create one empty file is in specified location
         File dstImg = new File(".//sreenshot//test.png");
         // step 4 : copy the screenshot using common-io 3rd party tool
         FileUtils.copyFile(scrImg, dstImg);

    }


}
```

How to take a Screenshot with Time Stamp?



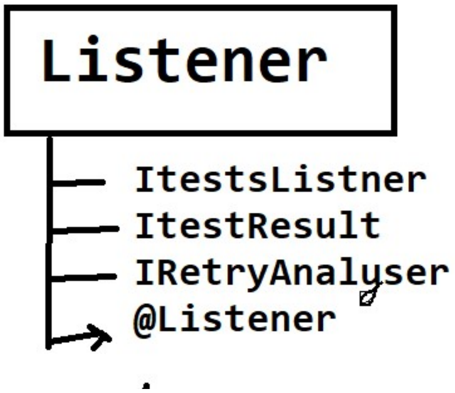
```java
import org.openqa.selenium.support.events.EventFiringWebDriver;
import org.testng.annotations.Test;

Run All
public class SampleTest {

    @Test
    Run | Debug
    public void createOrgTest(Method mtd) throws IOException {

        String currentMtdName = mtd.getName();
        Date date = new Date();
        String currentDate = date.toString().replace(" ", "_").replace(":", "_");

        System.out.println(currentDate);

        WebDriver driver = new FirefoxDriver();
        driver.get("http://localhost:8888");

        //step 1 : create an Object EventFiringWebDriver class & pass driver object
        EventFiringWebDriver eDriver = new EventFiringWebDriver(driver);
         // step 2 : take scrrenShot & return File type of Scerenshot
         File scrImg = eDriver.getScreenshotAs(OutputType.FILE);
         // step 3 : create one empty file is in specified location
         File dstImg = new File(".//sreenshot//"+currentMtdName + "_"+currentDate+".png");
         // step 4 : copy the screenshot using common-io 3rd party tool
         FileUtils.copyFile(scrImg, dstImg);

    }
```

When ever test scripts got failed , how to take a screenshot only failed test Script?

Inorder to take screenshot for failed test , we have to go for listener available in testNG

Listener

⇨ Listen is feature available in TESTNG , which is used to capture runtime events during execution & perform appropriate action based on eventtype



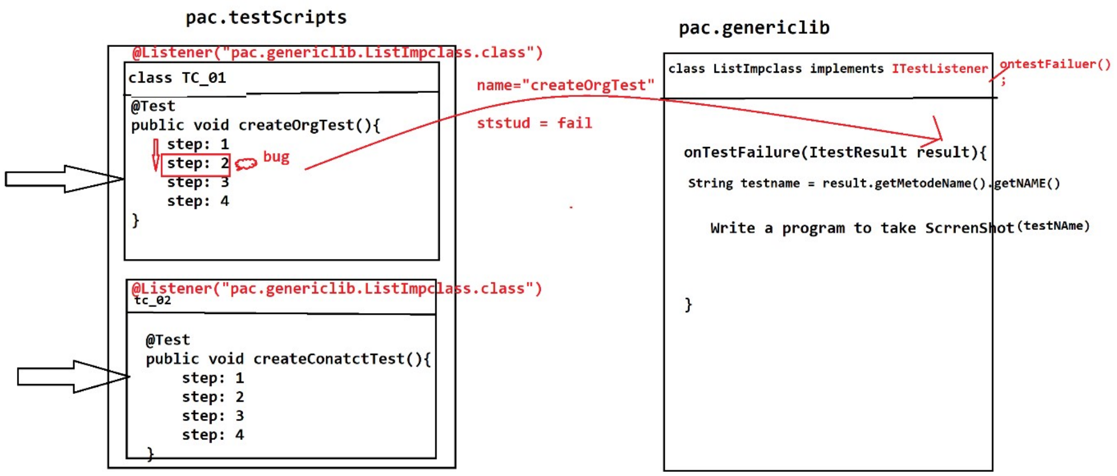
ITestListener:

- ➪ This is a special feature in testNG which enables the user to take run time events whenever the test scripts fail/pass. Implementation class for ITestListener is mandatory to use Listener feature
- ➪ @Listener is testing annotation , which is used to monitor the test execution in the runtime & generate a runtime event to Listener Implementation class , if test is pass / fail
- ➪ @Listener annotation will be declared in every testScripts class before class definition block
- ➪ ListImplemenation class helps us to receive the failure events from the @Listener & perform appropriate actions

Advantages of Listener :

- ➪ We can use Listener to take a screenshot for the failed test case, when we execute in bulk
- ➪ We can also use Listener for connect to DB , Launch browser & login precondition program
- ➪ We can also use Listener for Extend Report Configuration

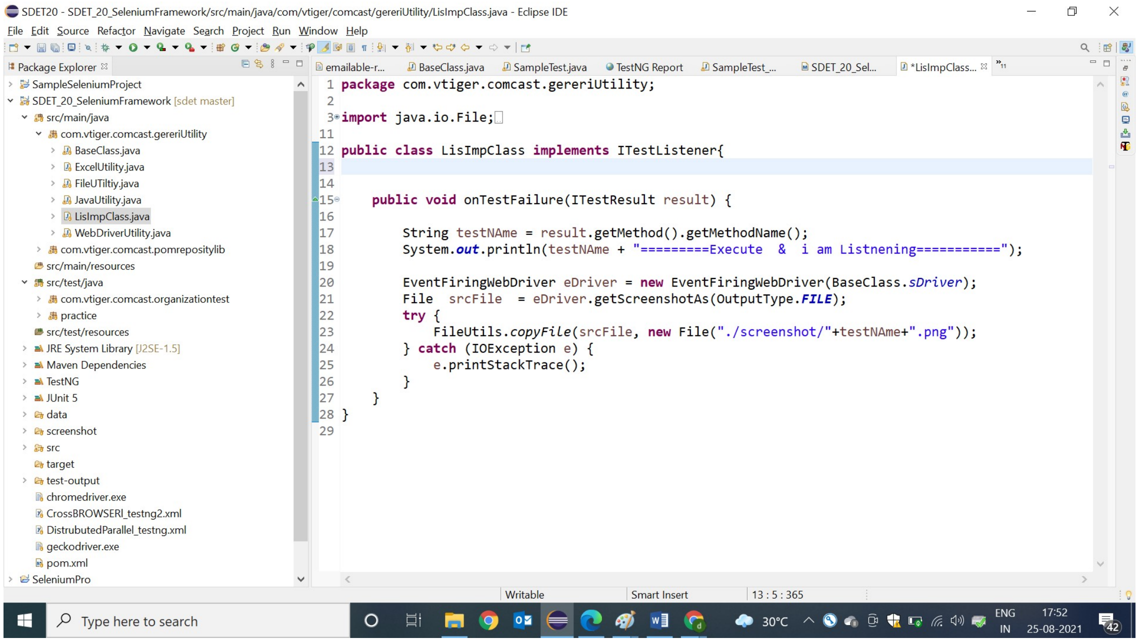
Listener Implementation class :



```
@Listener("pac.genericlib.ListImpclass.class")
class TC_01
@Test
public void createOrgTest(){
    step: 1
    step: 2
    step: 3
    step: 4
}
```

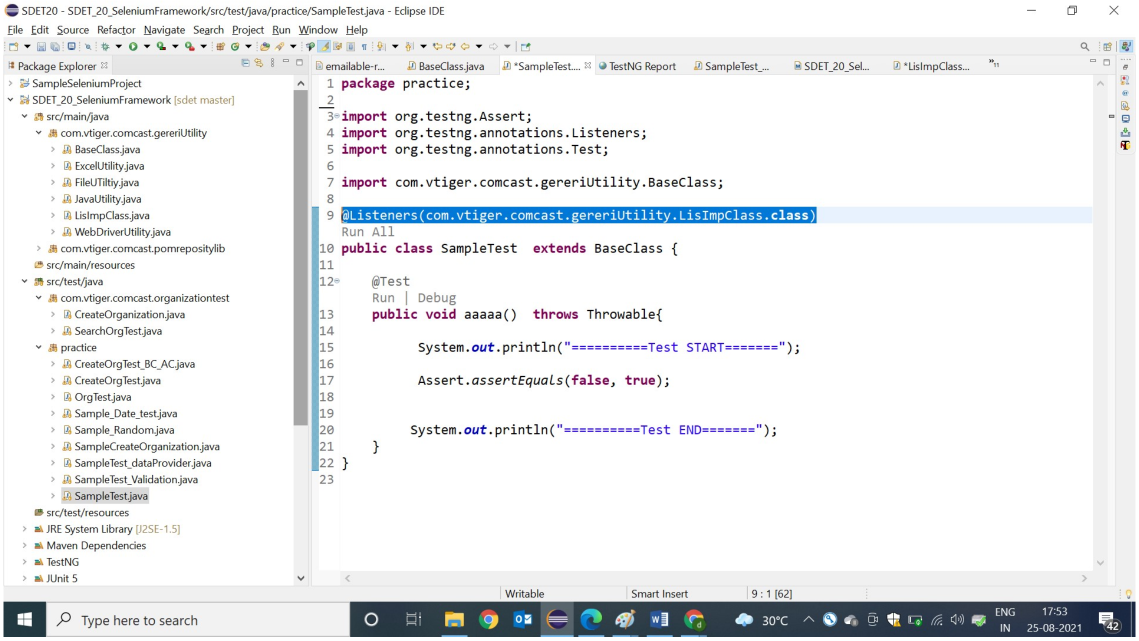
```
@Listener("pac.genericlib.ListImpclass.class")
tc_02
 @Test
 public void createConatctTest(){
     step: 1
     step: 2
     step: 3
     step: 4
 }
```

```
class ListImpclass implements ITestListener

 onTestFailure(ItestResult result){
  String testname = result.getMetodeName().getNAME()

    Write a program to take ScrrenShot(testNAme)


 }
```



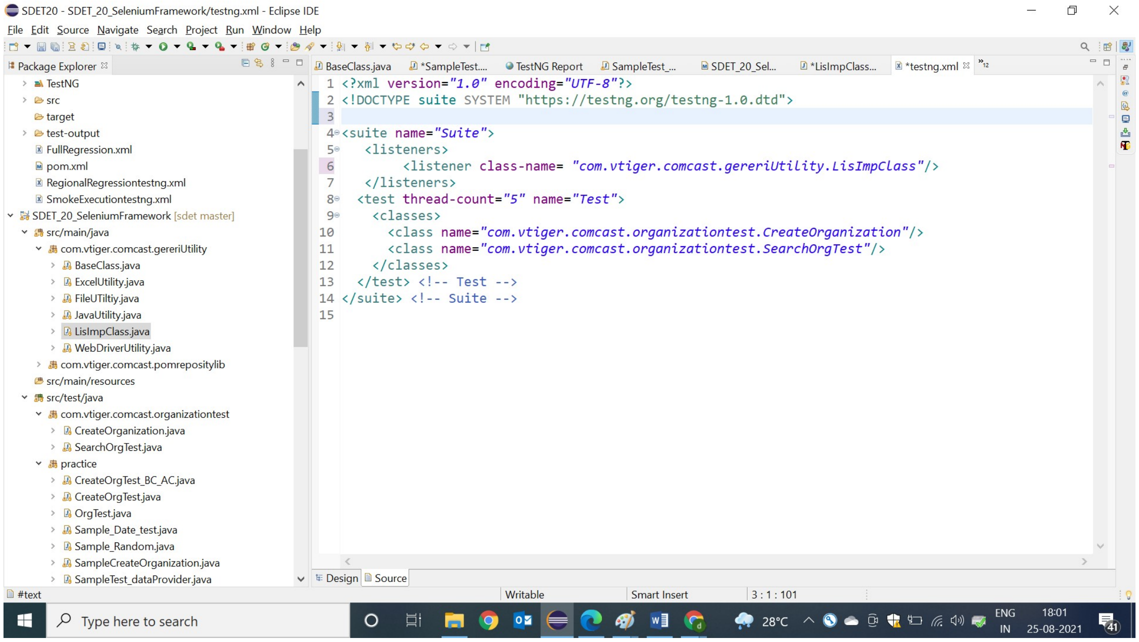
```java
package com.vtiger.comcast.gereriUtility;

import java.io.File;

public class LisImpClass implements ITestListener{


    public void onTestFailure(ITestResult result) {

        String testNAme = result.getMethod().getMethodName();
        System.out.println(testNAme + "=========Execute  &  i am Listnening===========");

        EventFiringWebDriver eDriver = new EventFiringWebDriver(BaseClass.sDriver);
        File  srcFile  = eDriver.getScreenshotAs(OutputType.FILE);
        try {
            FileUtils.copyFile(srcFile, new File("./screenshot/"+testNAme+".png"));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Sample test for Listener :


SDET20 - SDET_20_SeleniumFramework/src/test/java/practice/SampleTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
*SampleTest....
Writable
Smart Insert
9 : 1 [62]

```java
package practice;

import org.testng.Assert;
import org.testng.annotations.Listeners;
import org.testng.annotations.Test;

import com.vtiger.comcast.gereriUtility.BaseClass;

@Listeners(com.vtiger.comcast.gereriUtility.LisImpClass.class)
Run All
public class SampleTest  extends BaseClass {

    @Test
    Run | Debug
    public void aaaaa()  throws Throwable{

          System.out.println("==========Test START=======");

          Assert.assertEquals(false, true);


         System.out.println("==========Test END=======");
    }
}
```

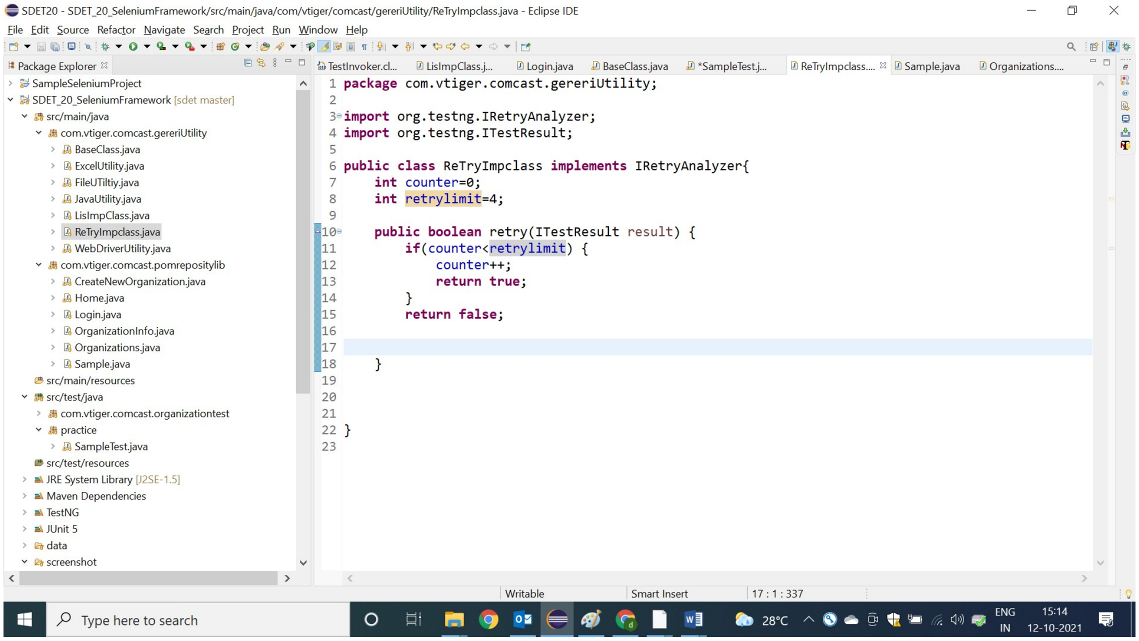
How to use Listener in TestNG.xml


SDET20 - SDET_20_SeleniumFramework/testng.xml - Eclipse IDE
File Edit Source Navigate Search Project Run Window Help
Package Explorer
*testng.xml
Design
Source
Writable
Smart Insert
3 : 1 : 101

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Suite">
   <listeners>
        <listener class-name= "com.vtiger.comcast.gereriUtility.LisImpClass"/>
   </listeners>
  <test thread-count="5" name="Test">
    <classes>
      <class name="com.vtiger.comcast.organizationtest.CreateOrganization"/>
      <class name="com.vtiger.comcast.organizationtest.SearchOrgTest"/>
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```

RetryAnalyzer:

- ⇨ This feature of TestNG helps the user to rerun the test script when ever test is getting failed
- ⇨ In order to use this feature we have to implements RetryAnalyzer interface & override retry method
- ⇨ Inside the retry method, we should specify the upper limit so that test script will get executed specified limit when test is getting failed.

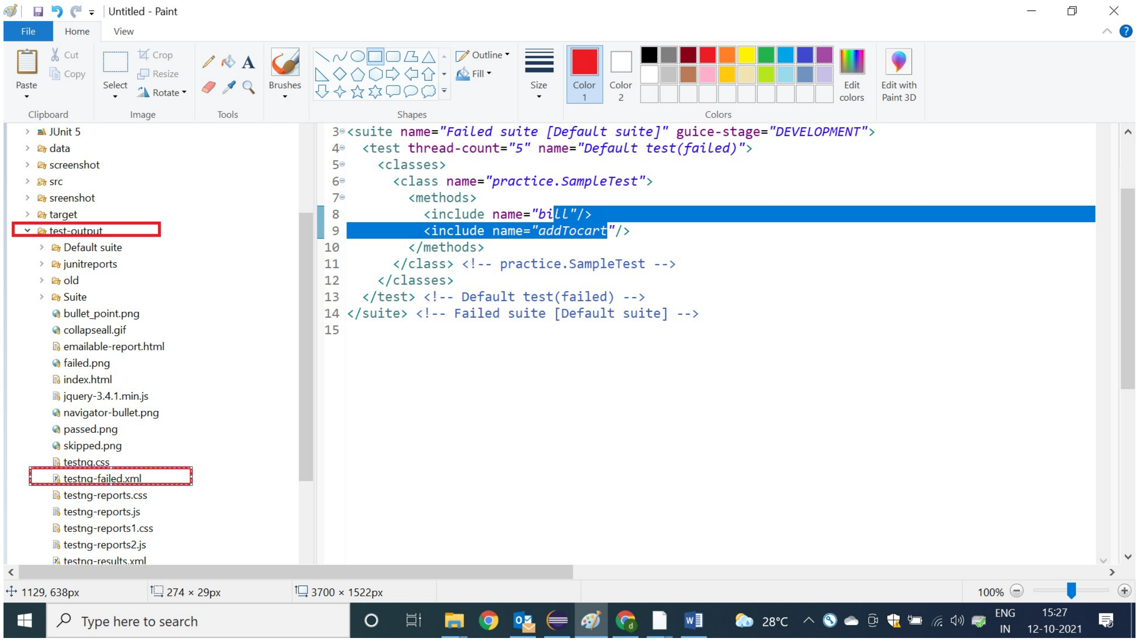


```java
package com.vtiger.comcast.gereriUtility;

import org.testng.IRetryAnalyzer;
import org.testng.ITestResult;

public class ReTryImpclass implements IRetryAnalyzer{
    int counter=0;
    int retrylimit=4;

    public boolean retry(ITestResult result) {
        if(counter<retrylimit) {
            counter++;
            return true;
        }
        return false;


    }



}
```

When ever we execute 1000 test scripts in batch, out of 1000 / 100 test scripts got failed , next time when a get new build , I wanted execute only failed test scripts , then is your approach ?

After the batach execution, refresh the project & go to test-output folder which is created inside the projectFolder , then execute “test-failed.xml” file

- ⇨ Testng-failed.xml file is created automatically by testing-tool itself for every failed execution reports .



- JUnit 5
- data
- screenshot
- src
- sreenshot
- target
- test-output
  - Default suite
  - junitreports
  - old
  - Suite
  - bullet_point.png
  - collapseall.gif
  - emailable-report.html
  - failed.png
  - index.html
  - jquery-3.4.1.min.js
  - navigator-bullet.png
  - passed.png
  - skipped.png
  - testng.css
  - testng-failed.xml
  - testng-reports.css
  - testng-reports.js
  - testng-reports1.css
  - testng-reports2.js
  - testng-results.xml

```xml
<suite name="Failed suite [Default suite]" guice-stage="DEVELOPMENT">
  <test thread-count="5" name="Default test(failed)">
    <classes>
      <class name="practice.SampleTest">
        <methods>
          <include name="bill"/>
          <include name="addTocart"/>
        </methods>
      </class> <!-- practice.SampleTest -->
    </classes>
  </test> <!-- Default test(failed) -->
</suite> <!-- Failed suite [Default suite] -->
```