Testing frame with GitHub + Selenium + Eclipse + Maven + TestNG + Log4j

# Environment setup:

1. JDK (skip)
2. Eclipse(skip)
3. Maven

Download from <http://maven.apache.org/download.cgi>

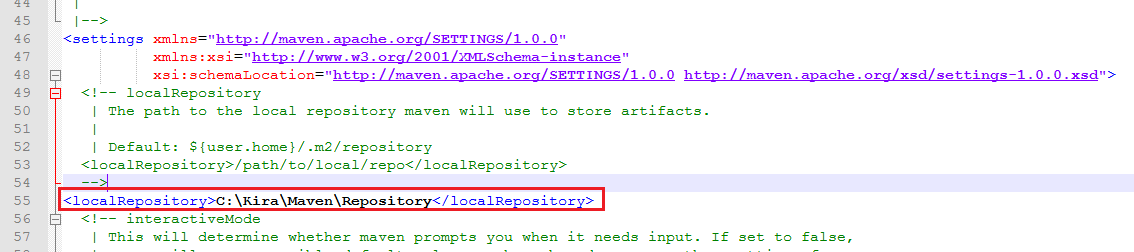
<http://jingyan.baidu.com/album/295430f136e8e00c7e0050b9.html?picindex=1>

Note:

If needed, you could change the default local repository if your needed:

Add the following line into C:\Kira\Maven\apache-maven-3.3.9\conf\settings.xml

|  |
| --- |
| <localRepository>C:\Kira\Maven\Repository</localRepository> |



1. TestNG

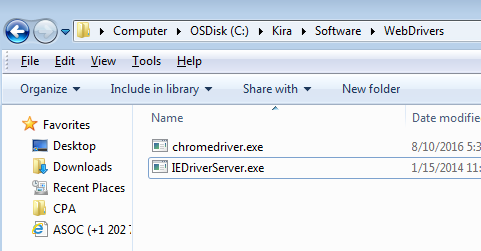
[TestNG 入门教程](http://www.cnblogs.com/TankXiao/p/3888070.html)

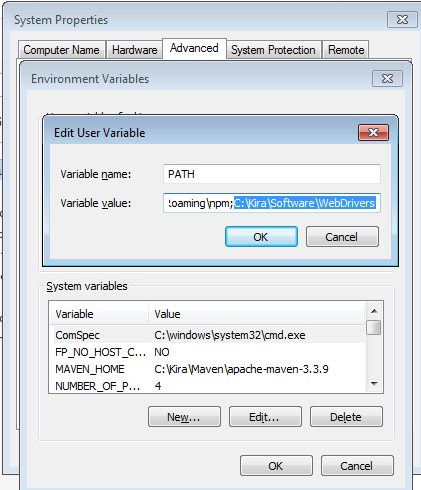
1. GitHub

[**github for windows上传本地项目到github**](http://www.cnblogs.com/wdlhao/p/5810955.html)

Demo project: https://github.com/ACCKira/BaiduTest.git

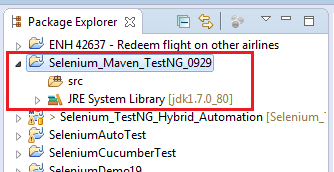
1. Download all webdrivers for different browsers you needed in one folder, and add the path into Environment Variables… (PATH).



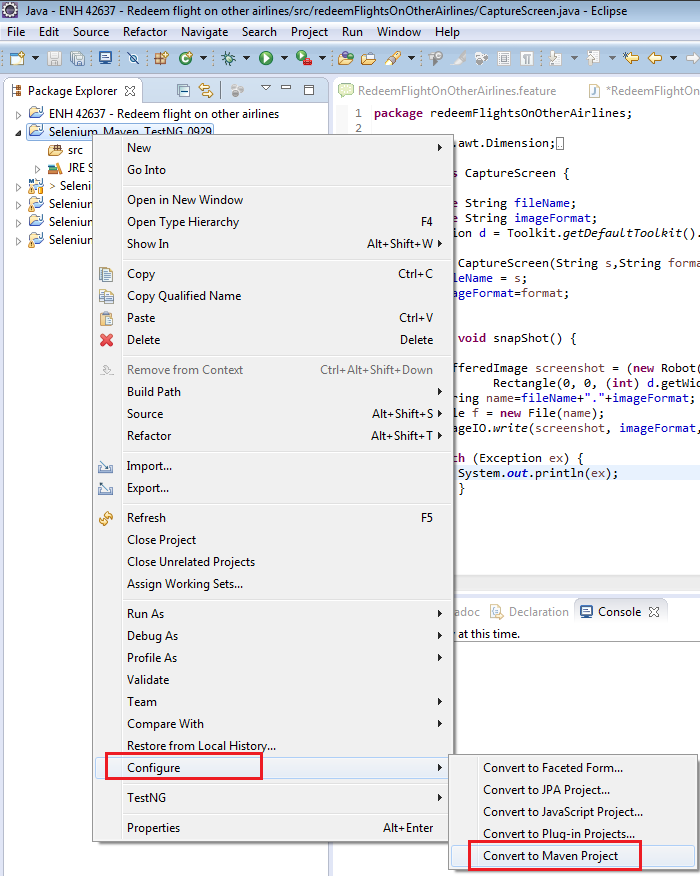


# Demo creation

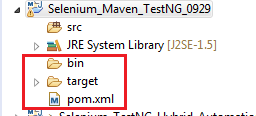
## New a java project Selenium\_Maven\_TestNG\_0929 (<http://blog.csdn.net/a542551042/article/details/45873693?locationNum=6> )



## Convert it into Maven Project



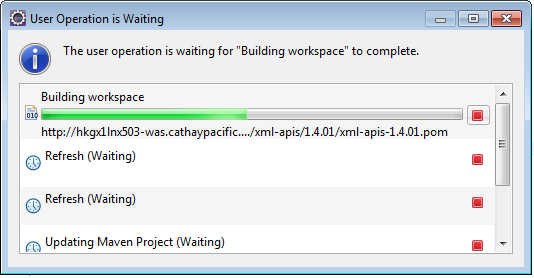
After clicking Finish button, will have some new files add.



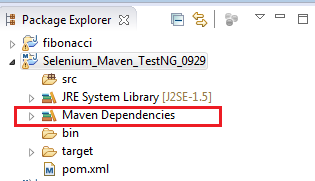
## Edit pom.xml to manage our .jar files.

|  |
| --- |
| <dependencies>  <dependency>  <groupId>org.seleniumhq.selenium</groupId>  <artifactId>selenium-java</artifactId>  <version>2.53.0</version>  </dependency>  <dependency>  <groupId>org.testng</groupId>  <artifactId>testng</artifactId>  <version>6.9.10</version>  </dependency>    <dependency>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  <version>1.2.17</version>  </dependency>    </dependencies> |

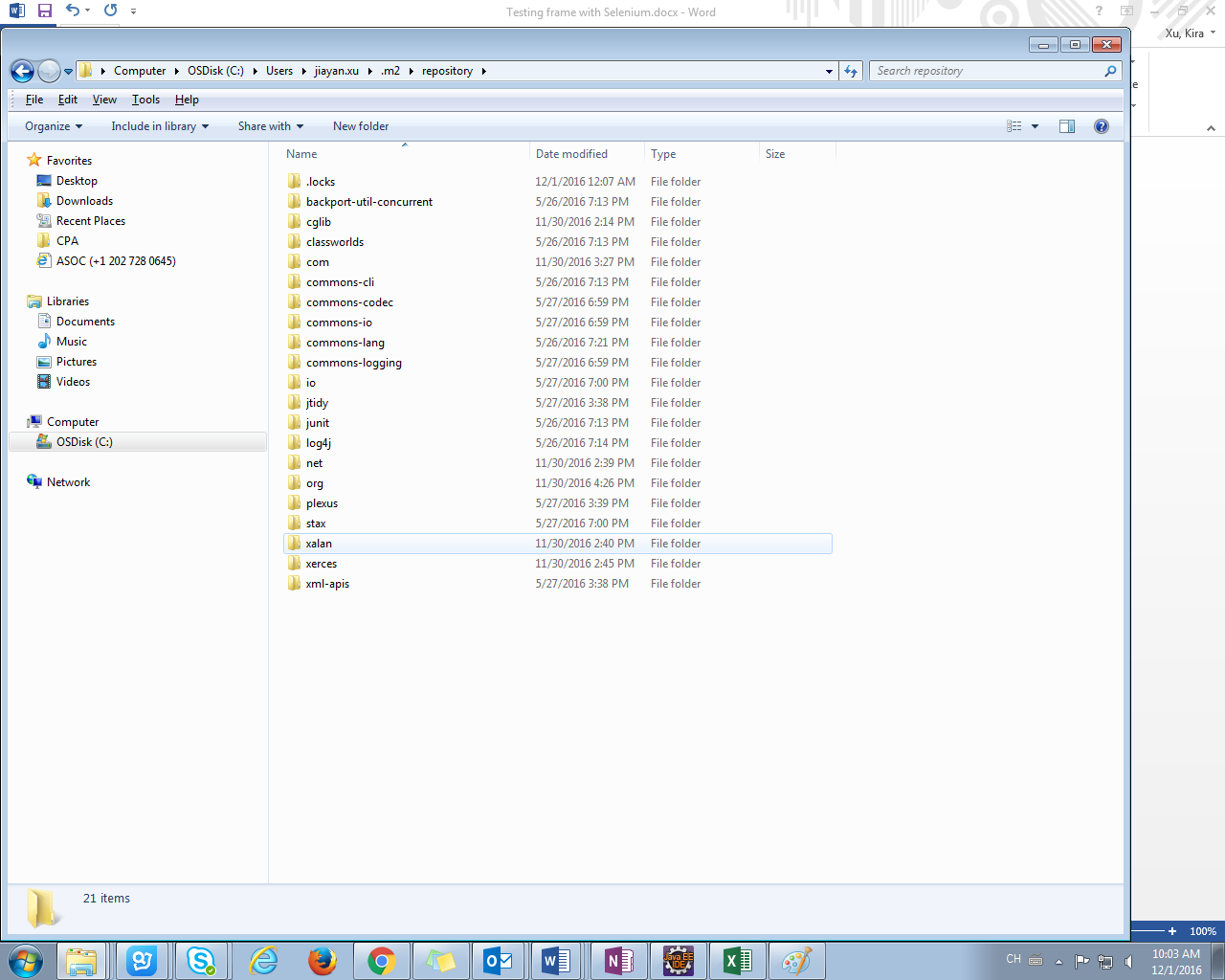
Press Ctrl+s.



It will last for a few minutes.

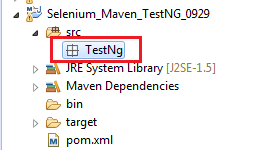


All .jar files download into C:\Users\jiayan.xu\.m2\repository



## Crest TestNG class

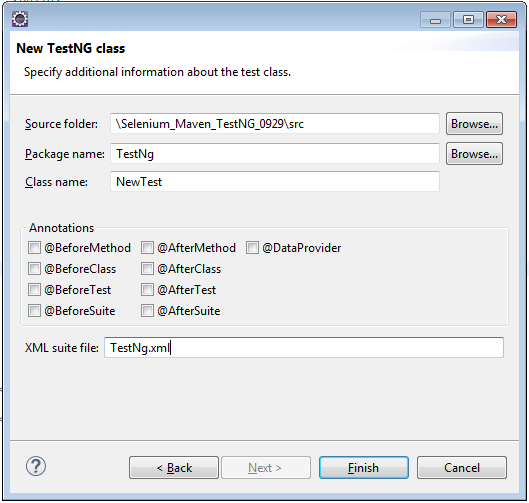
1. Create a package TestNg under src



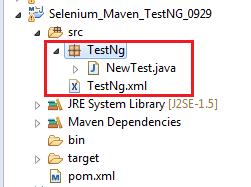
1. Create TestNG class under TestNg package.

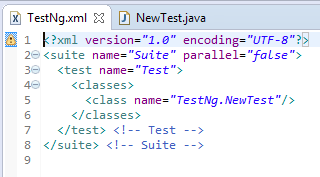
Right click the TestNg package -> TestNG -> Create TestNG class

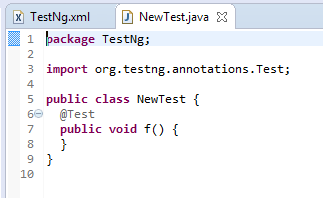
(Right click the TestNg package -> New -> Other… (Ctrl+N) -> TestNG -> TestNG class)



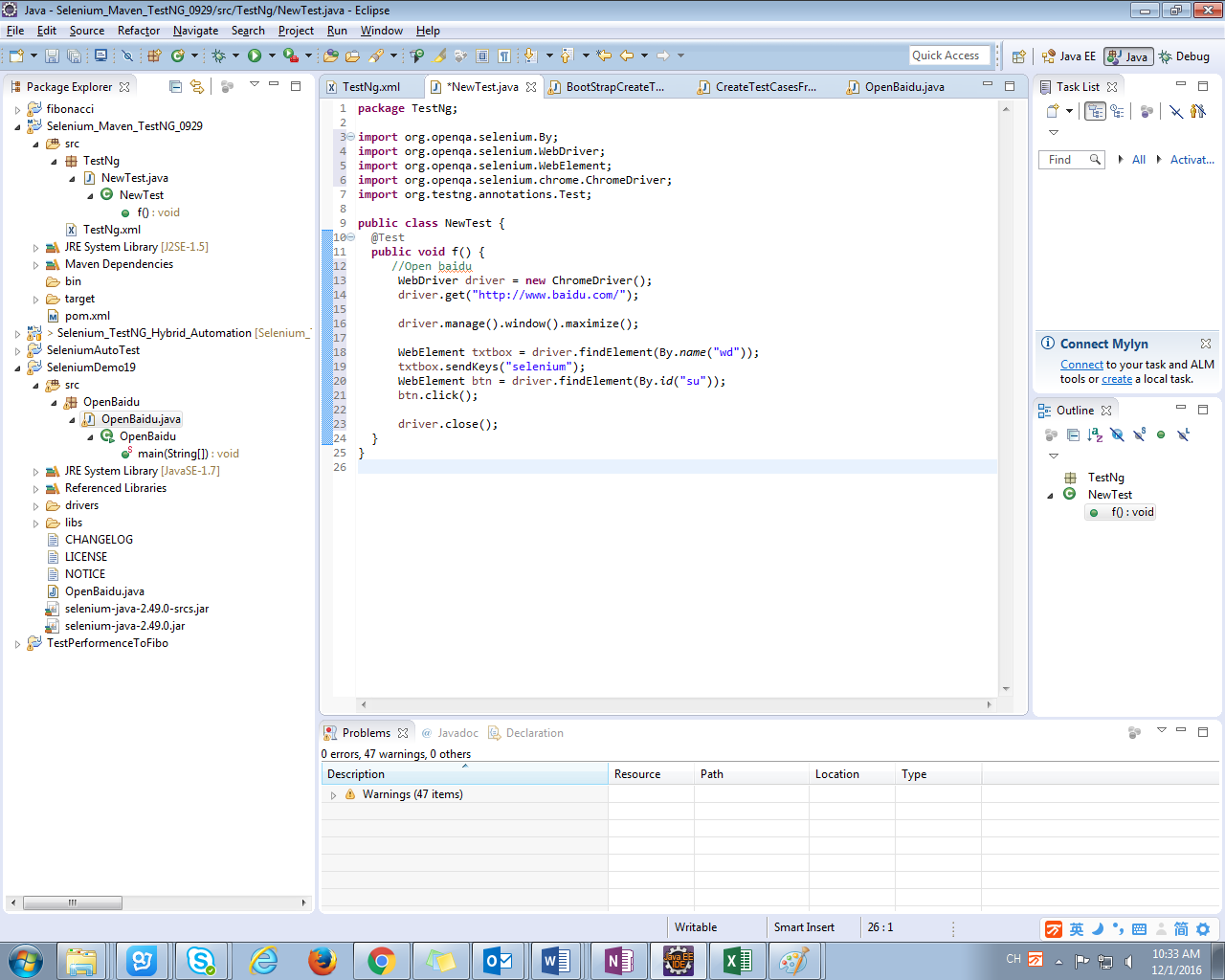
Finish







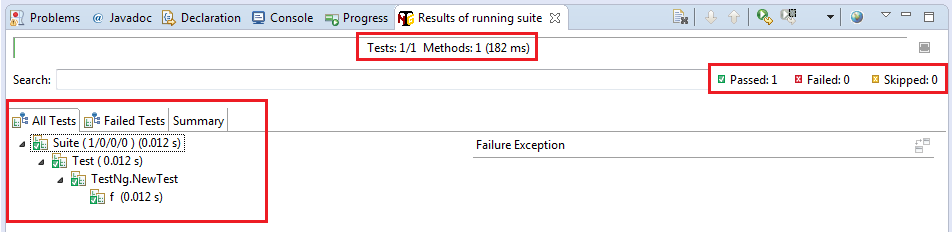
Coding for TestNg



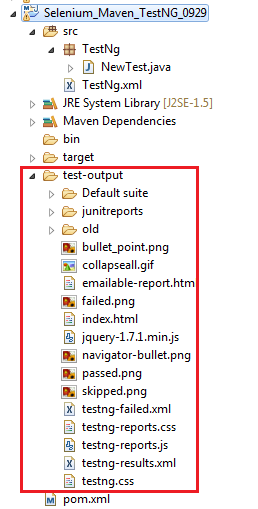
Run the test case by right clicking TestNg.xml -> Run As -> TestNG Suite.

Or right click the test case -> Run As -> TestNG Test.

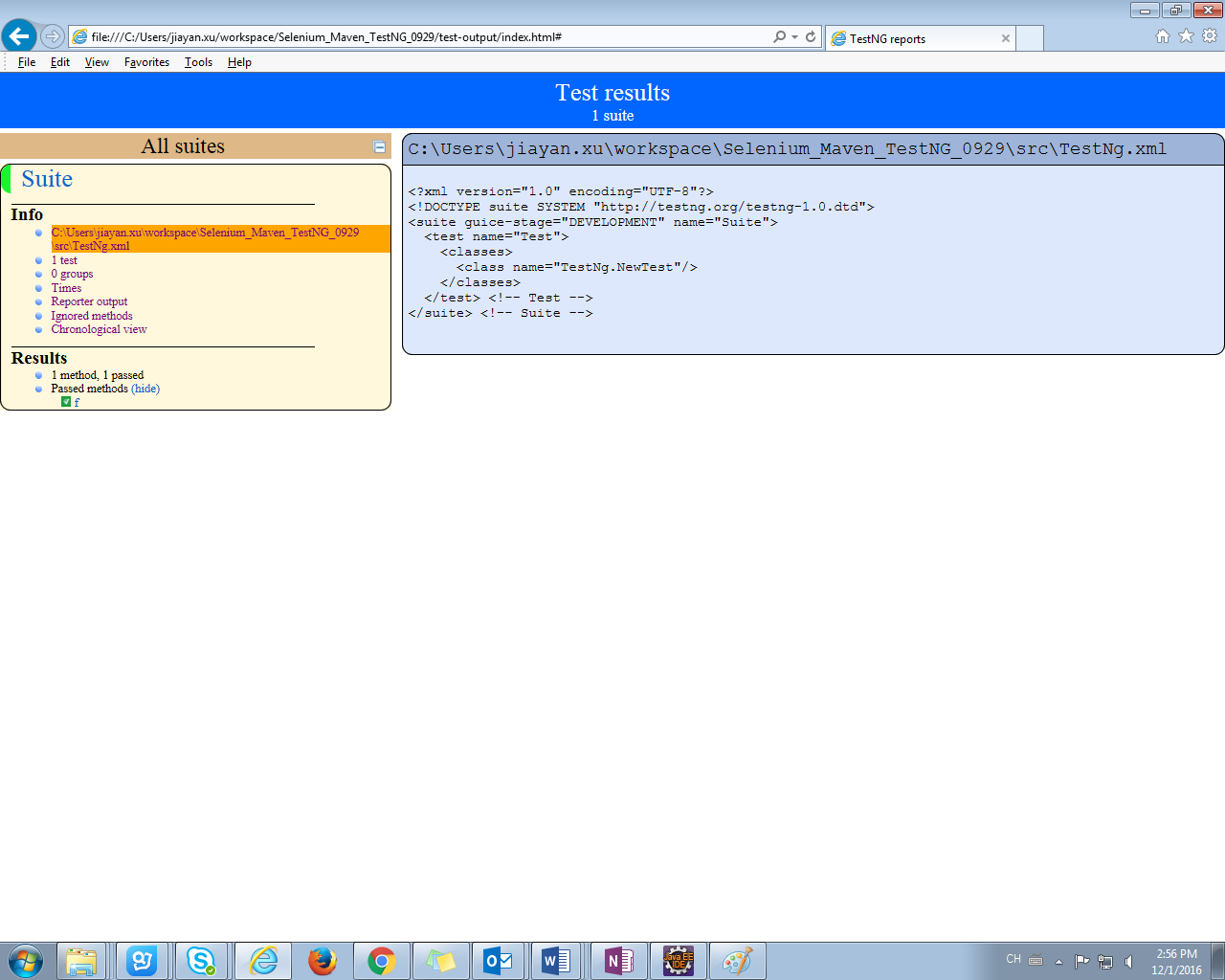
## Test Result



Refresh the project, then you can see “test-output” folder added.

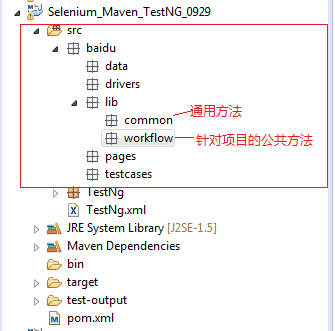


HTML report



# Design automation test frame

|  |  |
| --- | --- |
| Reference | <http://blog.csdn.net/snakeshiy/article/details/8565178>  <http://mt.sohu.com/20160812/n463987511.shtml>  <http://wenku.baidu.com/view/4f1042f77c1cfad6195fa7ff.html>  <http://www.ltesting.net/ceshi/ceshijishu/zdcs/zdcskj/2011/0311/202192.html> |



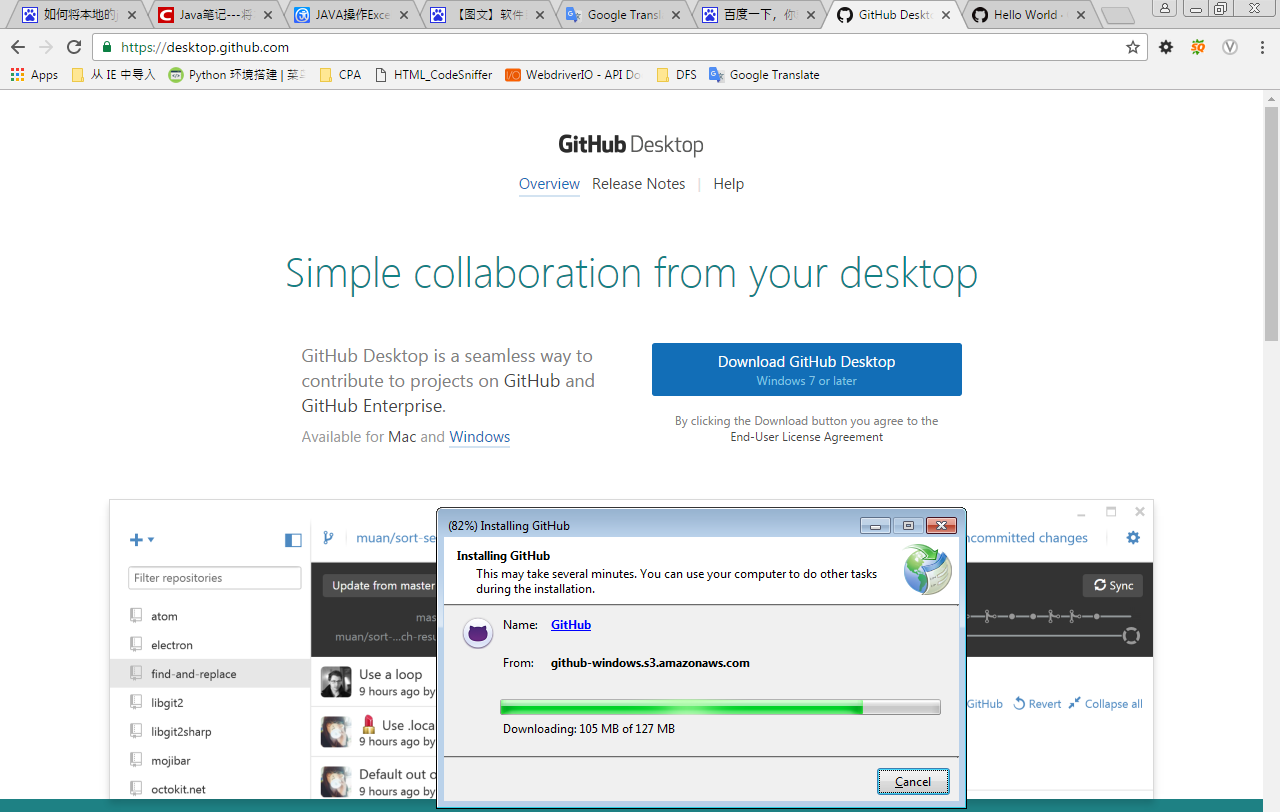
# Version Management

Configure GitHub as my version management tool.

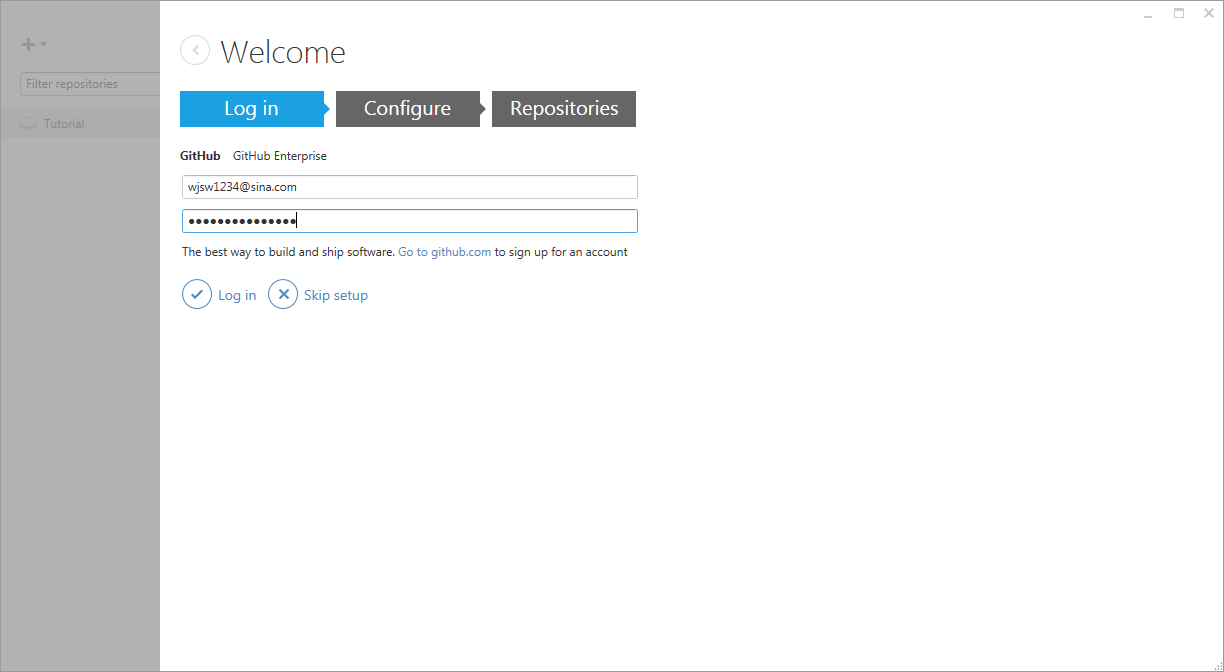
Apply a GitHub account in <https://github.com/>

Create a Create a Repository, refer to <https://guides.github.com/activities/hello-world/#repository>

Download and install GitHub Desktop



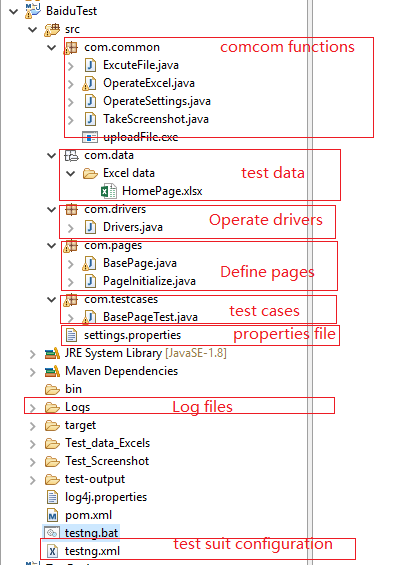
Login



The detail steps for how to add located project into your GitHub see [**github for windows上传本地项目到github**](http://www.cnblogs.com/wdlhao/p/5810955.html)

# Perfect the design with this test frame

Demo project: https://github.com/ACCKira/BaiduTest.git



## Configure **setting.properties** file for some basic configuration.

|  |
| --- |
| 1. #set your path of different webdrivers in your PC. 2. IEDriverPath=C:/Kira/Software/WebDriver/IEDriverServer.exe 3. #FireFoxDriverPath=C://Program Files (x86)/Mozilla Firefox/firefox.exe 4. FireFoxDriverPath=C:/Kira/Software/WebDriver/geckodriver.exe 5. ChromeDriverPath=C:/Kira/Software/WebDriver/chromedriver.exe 6. #browsers: ie, chrome, firefox #Specify the browser to run 7. Browser=chrome 8. #test data path 9. DataPath=C://Kira/WorkSpace/BaiduTest/Test\_data\_Excels/ 10. #Set the path of screenshots for cases running steps. 11. ScreenshotPath=C://Kira/WorkSpace/BaiduTest/Test\_Screenshot |

Coding for reading the setting.properties:

(The complete code, please refer to github source code.)

|  |
| --- |
| **static** {  Properties prop = **new** Properties();  InputStream is = Object.**class**.getResourceAsStream("/settings.properties");  **try** {  prop.load(is);  *browser* = prop.getProperty("Browser").trim();  *ieDriverPath* = prop.getProperty("IEDriverPath").trim();  *fireFoxDriverPath* = prop.getProperty("FireFoxDriverPath").trim();  *chromeDriverPath* = prop.getProperty("ChromeDriverPath").trim();  *dataPath* = prop.getProperty("DataPath").trim();  *screenshotPath* = prop.getProperty("ScreenshotPath").trim();  } **catch** (IOException e) {  e.printStackTrace();  }  } |

## Configure **log4j for** logs management.

### Add reference in **pom.xml** to load the log4j jars with maven.

|  |
| --- |
| <dependency>  <groupId>commons-logging</groupId>  <artifactId>commons-logging</artifactId>  <version>1.2</version>  </dependency>  <dependency>  <groupId>log4j</groupId>  <artifactId>log4j</artifactId>  <version>1.2.17</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-log4j12</artifactId>  <version>1.7.21</version>  </dependency>  <dependency>  <groupId>org.slf4j</groupId>  <artifactId>slf4j-api</artifactId>  <version>1.6.4</version>  </dependency> |

### Add log4j.properties file to configure relevant properties.

|  |
| --- |
| log4j.rootCategory=INFO, stdout , R    # Console  log4j.appender.stdout=org.apache.log4j.ConsoleAppender  log4j.appender.stdout.layout=org.apache.log4j.PatternLayout  log4j.appender.stdout.layout.ConversionPattern=[BaiduTest] %p [%t] %C.%M(%L) | %m%n  log4j.appender.stdout.Encoding=UTF-8    # File  log4j.appender.R=org.apache.log4j.DailyRollingFileAppender  log4j.appender.R.File=C:\\Kira\\WorkSpace\\BaiduTest\\Logs\\log.log  log4j.appender.R.layout=org.apache.log4j.PatternLayout  log4j.appender.R.layout.ConversionPattern=%d-[BaiduTest] %p %t %c - %m%n  log4j.appender.R.Encoding=UTF-8    log4j.logger.org.apache.commons=ERROR  log4j.logger.org.apache.struts=WARN  log4j.logger.org.displaytag=ERROR  log4j.logger.org.springframework=DEBUG  log4j.logger.org.apache.velocity=FATAL    log4j.logger.org.hibernate.ps.PreparedStatementCache=WARN  log4j.logger.org.hibernate=DEBUG  log4j.logger.org.logicalcobwebs=WARN |

The detailed explanation refer to <http://www.cnblogs.com/pigtail/archive/2013/02/16/2913195.html>

### Note:

If the log4j log can't be output with the red warning as follows

log4j:WARN No appenders could be found for logger (dao.hsqlmanager).  
log4j:WARN Please initialize the log4j system properly.  
log4j:WARN See <http://logging.apache.org/log4j/1.2/faq.html#noconfig> for more info.

You can resolve the issue with following steps:

It looks like you need to add the location of your log4j.properties file to the Classpath in Eclipse. Make sure your project is open in Eclipse, then click on the "Run" menu at the top of Eclipse and click on the following:

1. Run
2. Run Configurations
3. Classpath (tab)
4. User Entries
5. Advanced (button on the right)
6. Add Folders
7. then navigate to the folder that contains your log4j.properties file
8. Apply
9. Run

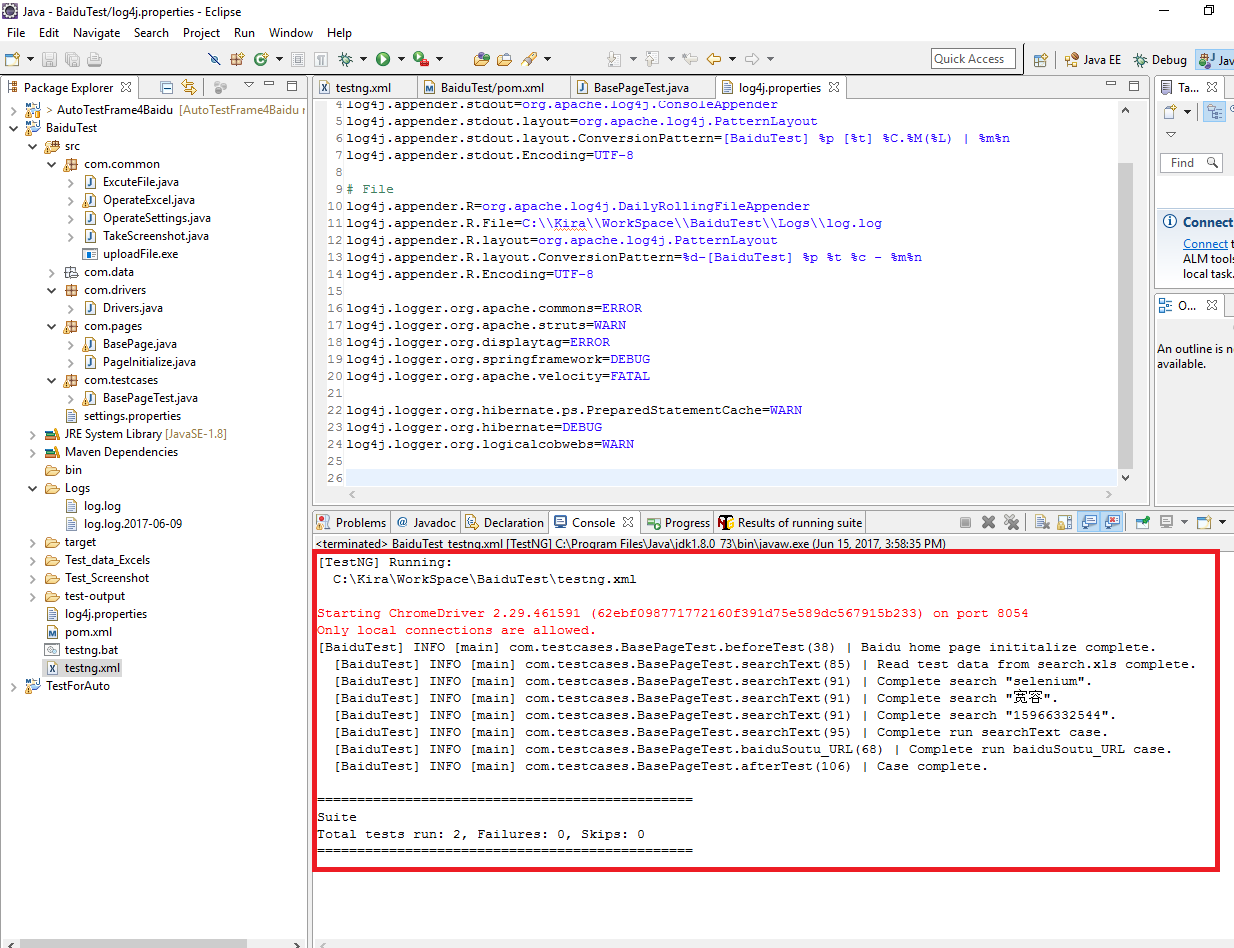
Machine generated alternative text:
Java - BaiduTest/src/Icg4j.prcperties - Eclipse 
File Edit Source Refactor 
Package Explorer 
> AutoTestFrame48aidu 
8aiduTest 
coma:ommon 
com.data 
com.drivers 
com.pages 
com.testcases 
Navigate Search Proje 
in dow Help 
[AutoTe 
u] 8asePageTest.java 
8asePagejava log4j.properties 
settin # 
Run Configurations 
Create. manage. and run configurations 
Name: 8asePageTest 
type filter text 
J2EE Preview 
Java Applet 
Java Application 
Ju JUnit 
JUnit Plug-in Test 
m2 Maven Build 
OSGi Framework 
Ju Task Context Test 
TestNG 
AutoTestFrame4E 
8aiduTest testng 
Base PageTest 
HomePageTest 
MainFunction 
TestForAuto_src.t 
x: XSL 
Filter matched 30 of 32 items 
Test Arguments Classpath 
Classpath: 
v 0.0 Bootstrap Entries 
JRE System Library [JavaSE- 1 
v 00 User Entries 
8aiduTest 
Maven De endencies 
src - 
JRE 
Source 
Environment 
8asePa eTest. 
Icg4J.prcperties 
— settings.properties 
JRE System Library [JavaSE-I 
Maven Dependencies 
Logs 
target 
Test data Excels 
Test Screenshot 
test-output 
@ pom.xml 
testng.xml 
Test ForAuto 
Common 
Up 
Add Projects.. 
Add JARs... 
Add External JARs... 
Adva nced... 
Restore Default Entries v 
Run 

The error message should no longer appear.

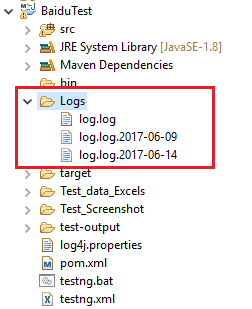
  From <[*https://stackoverflow.com/questions/12532339/no-appenders-could-be-found-for-loggerlog4j*](https://stackoverflow.com/questions/12532339/no-appenders-could-be-found-for-loggerlog4j)>

### The log output

Console:

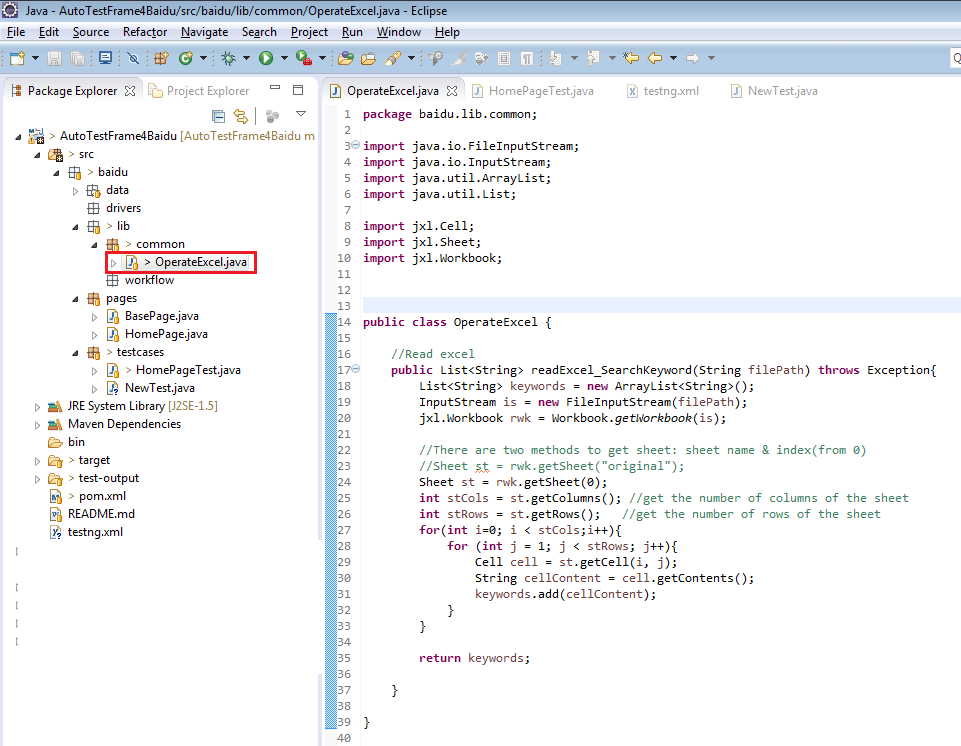


Log files:



Read test data.

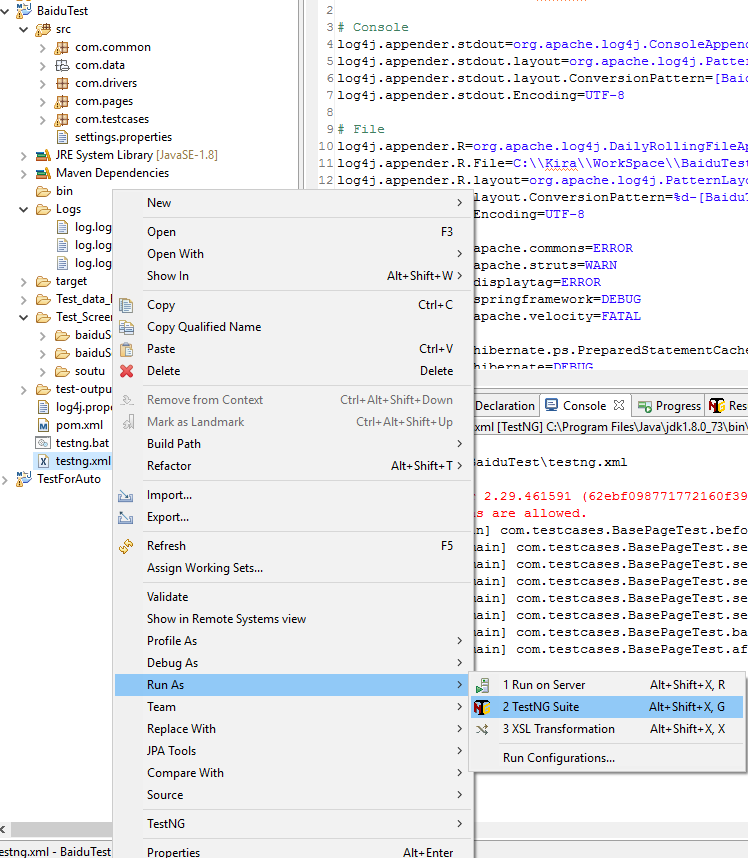
Save test data into an excel file. And define the data reading function in lib.common folder.



## Autorun test case demonstration

Start the automation test.

Right click the testng.xml 🡪 Run As 🡪 TestNG Suite

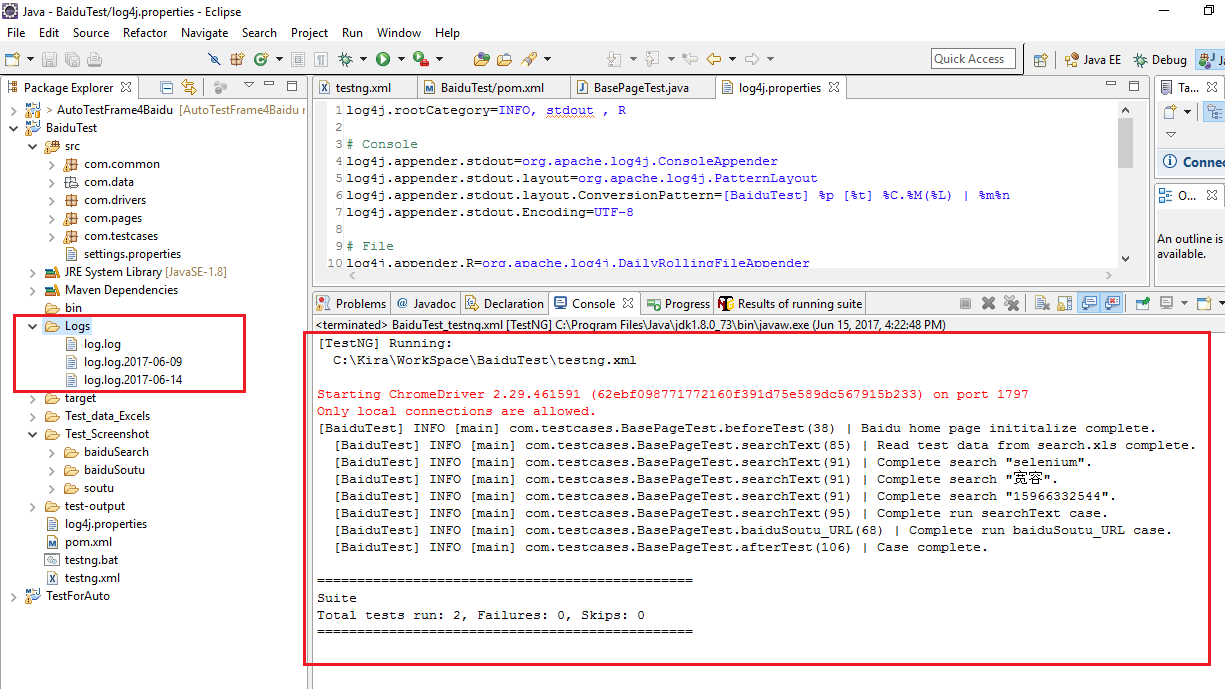


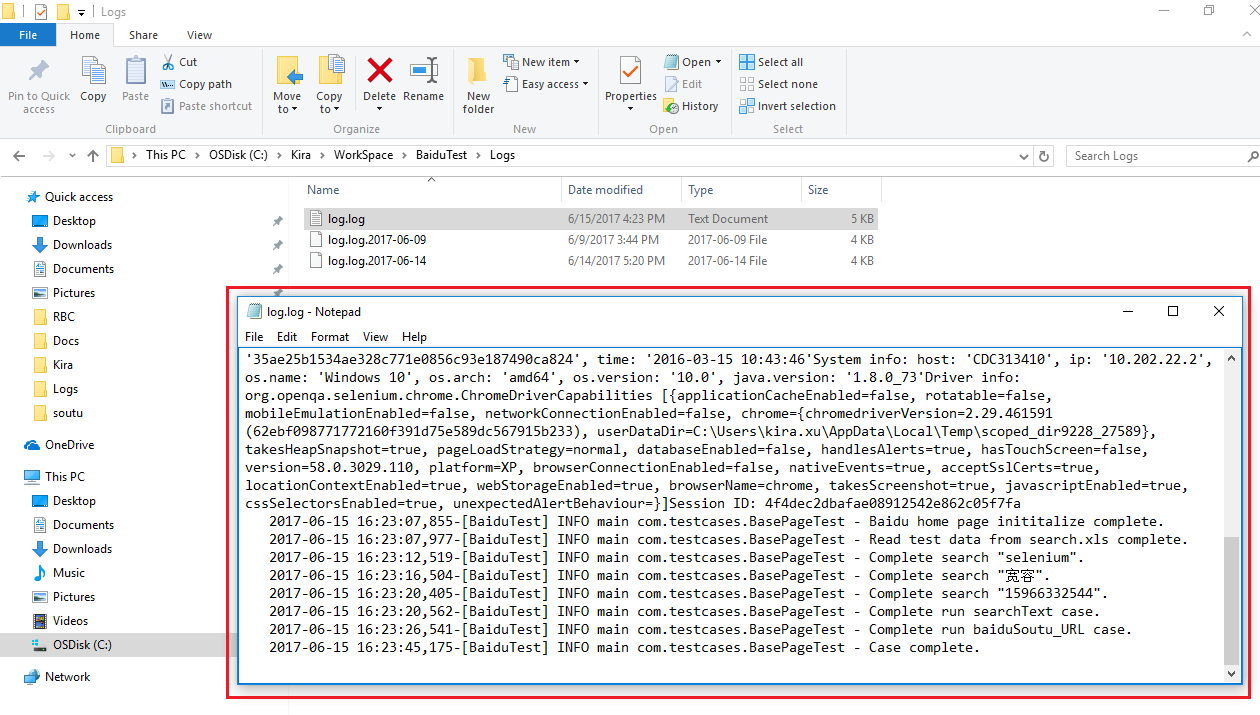
The case will run in browser you set in setting.properties file.



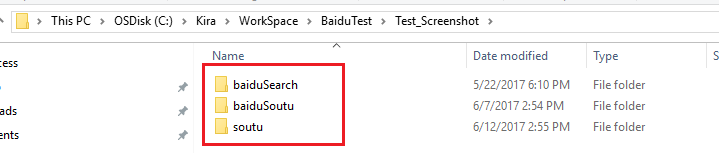
Until all test suites complete run.

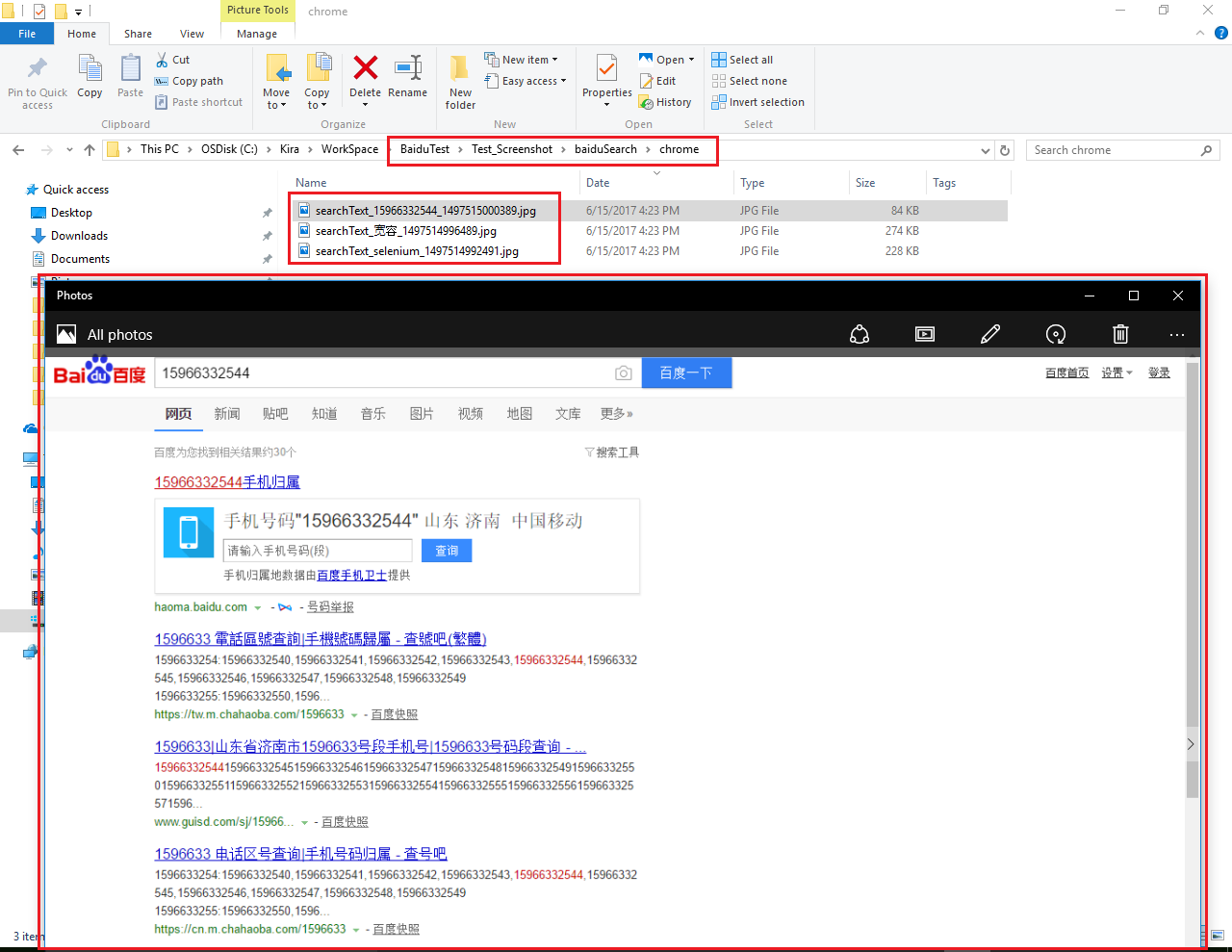
Output the log in console and log file.



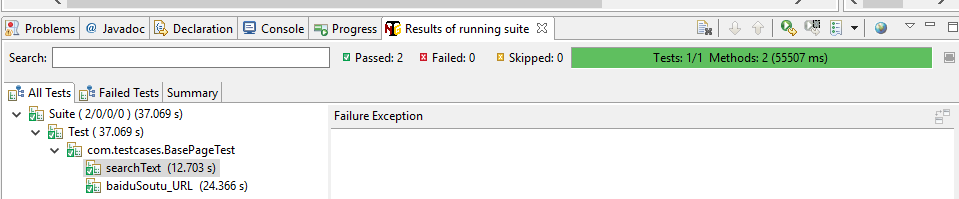


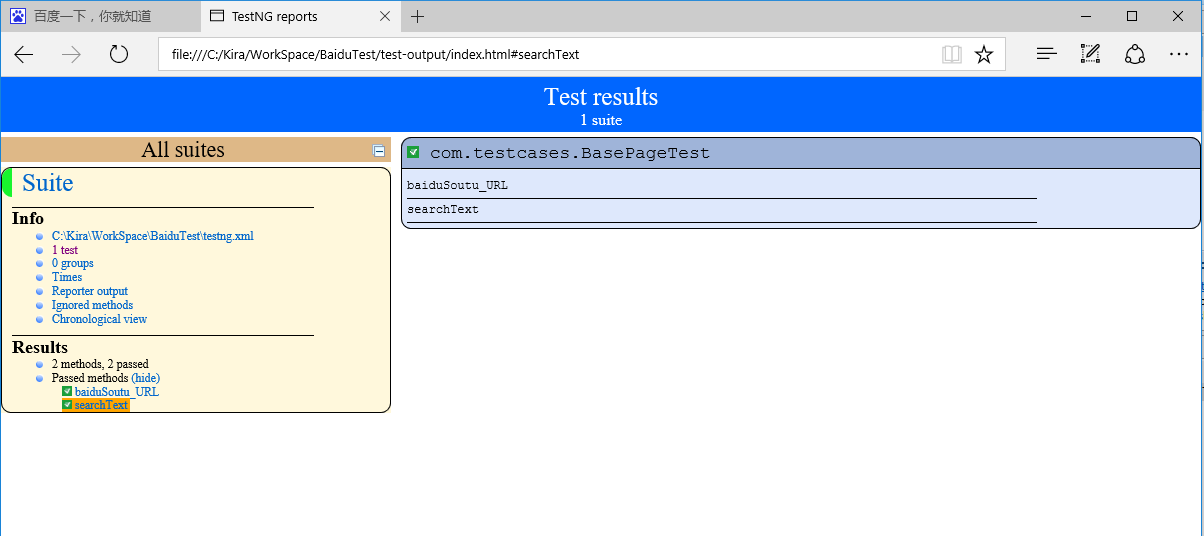
Output the screenshots for your necessary steps.





Output the TestNG Report.





## Bind testng into Maven

If you want to run testng suite with Run As -> Maven test, you could bind testng.xml into maven with the following steps.

**Add the following highlighted nodes in <build> </build> of pom.xml file**

|  |
| --- |
| <build>  <sourceDirectory>src</sourceDirectory>  <plugins>  <plugin>  <artifactId>maven-compiler-plugin</artifactId>  <version>3.5.1</version>  <configuration>  <source>1.8</source>  <target>1.8</target>  </configuration>  </plugin>  <plugin>  <groupId>org.apache.maven.plugins</groupId>  <artifactId>maven-surefire-plugin</artifactId>  <version>2.12</version>  <inherited>true</inherited>  <configuration>  <suiteXmlFiles>  <suiteXmlFile>testng.xml</suiteXmlFile>  </suiteXmlFiles>  </configuration>  </plugin>  </plugins>  </build> |

Press Ctrl + S to download the new added reference.

Right click the project -> Run As -> Maven test

The testng suites set in testng.xml will be run.

(To be continued……)