

# Selenium WebDriver阶段九 - (Maven实战)

## WebDriver (阶段九)

课程目的：学习使用maven执行测试脚本，及一些高级应用；

培训结果：掌握maven基本用法，及关键配置

课程相关脚本：

作者：Terry

QQ:314768474

个人微博：<http://weibo.com/alwaysterry>

版权所有禁止传播。

工具准备：

以前我们都是用m2e的插件，在eclipse ide中使用，现在我们为了脱离eclipse，用maven工具运行脚本。我们为什么这么做呢？我要开发在ide上进行，所有的脚本运行可以用maven工具执行，这样可以将执行与开发相分离。

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

<http://www.cnblogs.com/bigtall/archive/2011/03/23/1993253.html>

### 1-1)Maven surefire-plugin : Maven调用testng xml实现测试脚本运行。

①如果想通过maven去执行testng的cases，首先我们必须在pom.xml中添加testng的依赖。

②然后使用surefire的maven插件，指定你的testsuite.xml文件

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-surefire-plugin</artifactId>
      <version>2.4</version>
      <configuration>
        <suiteXmlFiles>
          <suiteXmlFile>src\main\java\fireflyautomation\p2\testsuite.xml</suiteXmlFile>
        </suiteXmlFiles>
      </configuration>
    </plugin>
  </plugins>
</build>
```

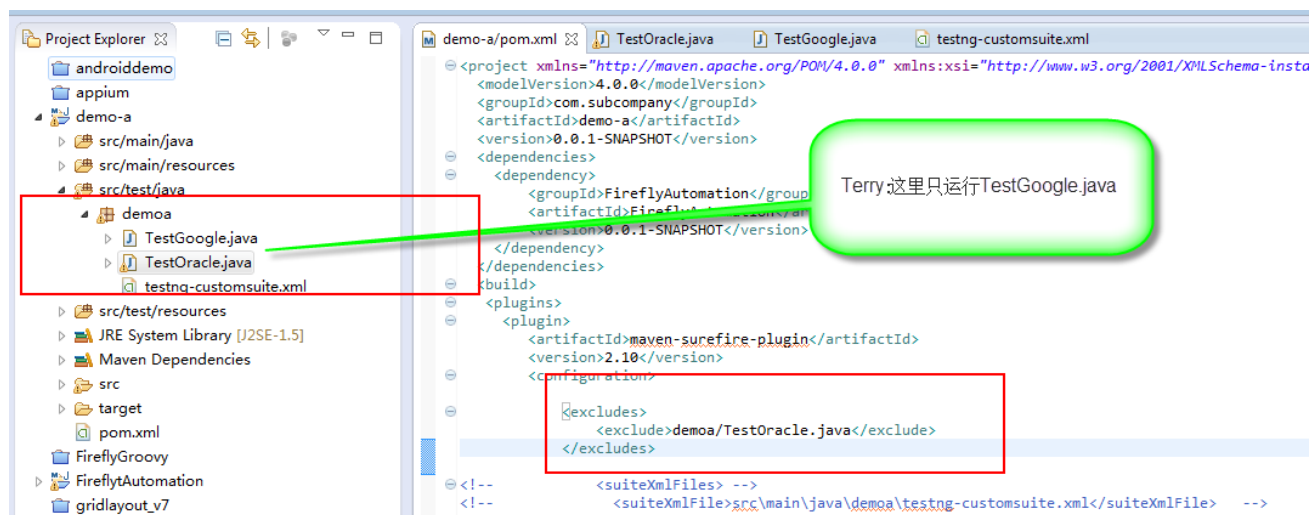
③这样我们就可以通过cmd执行测试命令

```
[INFO] -----
D:\project\Eclipse_Workspace\p1>%M2_HOME%\bin/mvn test
```

④我们可以去target的surefire-reports去得到一个testng的report

The screenshot displays the 'Test results' window in Eclipse. The top bar is blue with the text 'Test results' and '1 suite'. Below this, a table lists the test suite: 'fireflyautomation.p2.P2Project' with a green checkmark icon. The table has a column for the test name, showing 'test2'. On the left, a sidebar titled 'All suites' shows a tree view for 'mvn project'. Under 'Info', it lists the test suite file path, the number of tests (1), groups (0), and other details. Under 'Results', it shows '1 method, 1 passed' and 'Passed methods (hide)', with 'test2' listed as a passed method with a green checkmark icon.

当然这里也可以直接使用java文件，但是所有的测试用例文件必须都放在test目录下：



### 1-2)Maven surefire-plugin : Maven多线程运行测试用例。

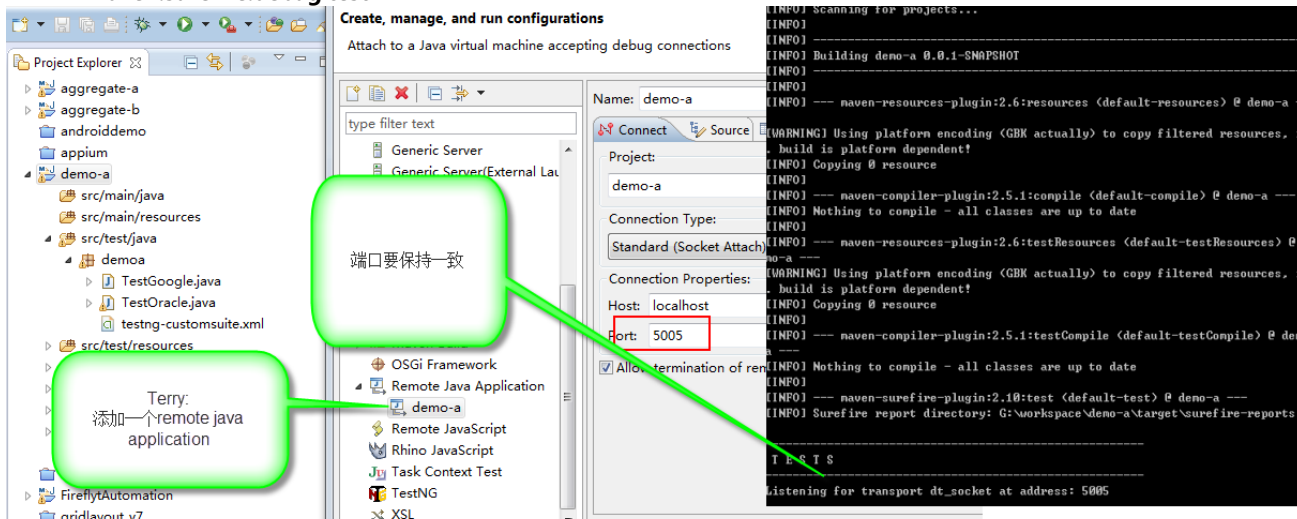


### 1-3)Maven surefire-plugin : Maven远程调试。

有时候你在使用mvn test过程中会碰到一系列错误，这个时候你可能面临的问题是最后能够结合代码调试。我们现在就可以这样做，将项目导入到eclipse,使用cmd去运行pom.xml通过远程的端口监听的方式进行调试：



## mvn -Dmaven.surefire.debug test



## 2) 远程仓库 (<http://search.maven.org>)

一般来说, 在一个maven目录下, 没有诸如lib的这样一个存放依赖文件的目录。当maven编译或测试时, 如果需要使用依赖文件的话, 他总是基于坐标使用本地仓库的依赖文件。默认情况下每个用户都有这样一个%userprofile%\.m2\repository本地仓库

默认情况下pom.xml文件中没有tagname:repositories。那是因为默认maven继承一个超级pom server, 这样你可以去下载你需要的依赖。

很多时候中央仓库不能满足你的项目需要这样我们需要修改一下远程仓库。

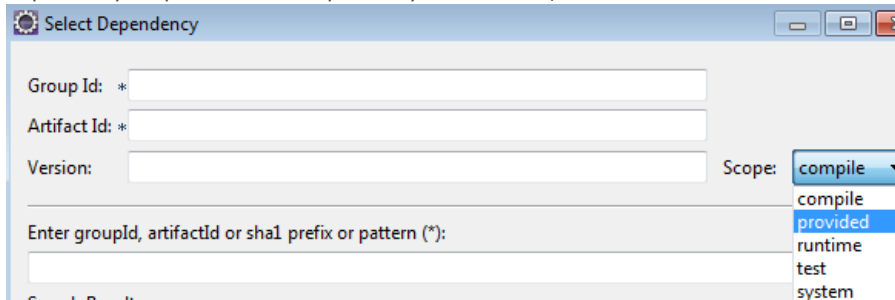
```

1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi
2 <modelVersion>4.0.0</modelVersion>
3 <groupId>FireflyAutomation</groupId>
4 <artifactId>FireflyAutomation</artifactId>
5 <version>0.0.1-SNAPSHOT</version>
6 <build>
7 <sourceDirectory>src</sourceDirectory>
8 <plugins>
9 <plugin>
10 <artifactId>maven-compiler-plugin</artifactId>
11 <version>3.0</version>
12 <configuration>
13 <source/>
14 <target/>
15 </configuration>
16 </plugin>
17 </plugins>
18 </build>
19 <repositories>
20 <repository>
21 <id>ibiblio</id>
22 <name>ibiblio repository</name>
23 <url>http://www.ibiblio.org/maven/mule/dependencies/maven2/</url>
24 <releases><enabled>true</enabled></releases>
25 <snapshots><enabled>false</enabled></snapshots>
26 <layout>default</layout>
27 </repository>
28 </repositories>
29
30 <dependencies>

```

### 3) Dependency scope

Dependency scope 是用来限制Dependency的作用范围的, 影响maven项目在各个生命周期时导入的package的状态。



#### compile

默认的scope, 表示 dependency 都可以在生命周期中使用。而且, 这些dependencies 会传递到依赖的项目中。

#### provided

跟compile相似, 但是表明了dependency 由JDK或者容器提供, 例如Servlet AP和一些Java EE APIs。这个scope 只能作用在编译和测试时, 同时没有传递性。

#### runtime

表示dependency不作用在编译时, 但会作用在运行和测试时

#### test

表示dependency作用在测试时, 不作用在运行时。

#### system

跟provided 相似, 但是在系统中要以外部JAR包的形式提供, maven不会在repository查找它。

### 4)maven项目与maven项目的依赖

假设公司里有两个自动化maven项目一个是p1,一个是p2

其中p1有个类

```

public class P1Project
{
    public void dispP1(){
        System.out.println("p1 project");
    }
}

```

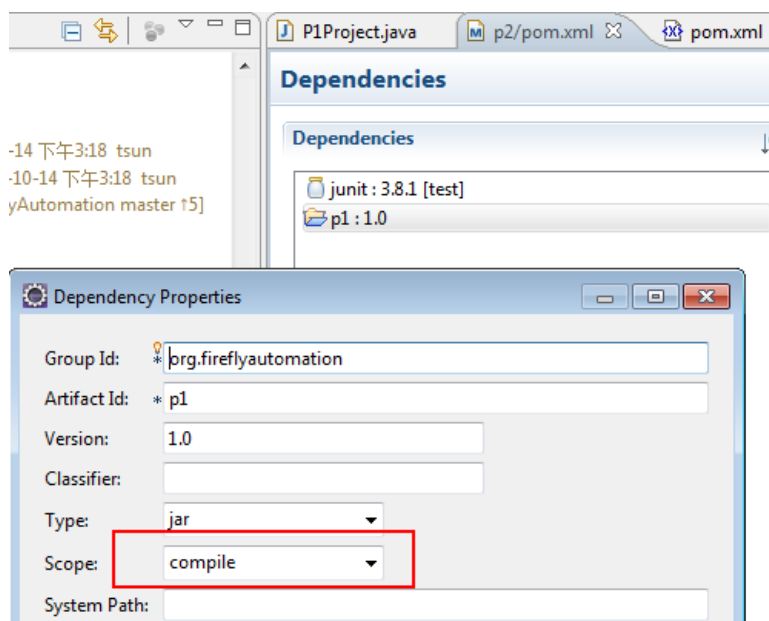
p2项目如果想使用这个p1项目中的类P1Project的方法, 怎么办呢?

我可以这样

将p1项目 maven install后自动化会在你本地仓库中安装该项目，这样你就可以该项目的坐标了。



引入p1项目，注意这里的scope，不可选择system，因为我们把他安装到本地的repository：

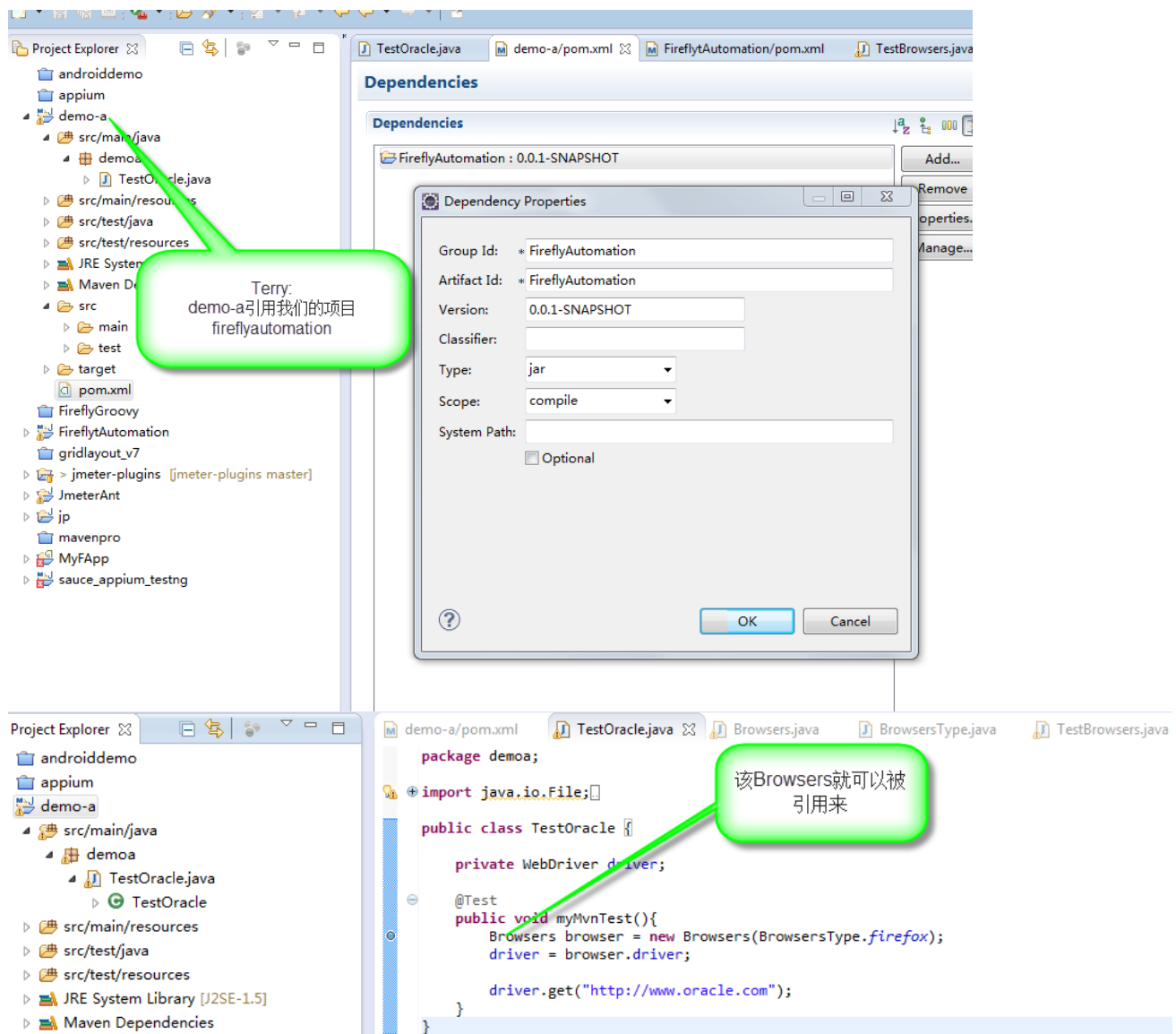


这样我们就可以在项目p2中使用p1的东西了。

```
1 /
2 /*
3
4 public class P2Project
5 {
6     public static void main(String args[]){
7         P1Project p1 = new P1Project();
8         p1.dispP1();
9     }
10 }
11
12
```

-----也可以看一下fireflyautomation的东西-----

如果对于一个maven项目的话，我可以这样将maven install安装到本地仓库，然后在自己pom.xml中找该项目：



## 5)自定义引入项目的jar包，使用maven管理。

就是你自己开发的jar包，并没有公网出来，或者本地私服出来。而是自己加入了项目，那么我们可以引入到maven的repository的。之前我们说到那个scope有个选项是system，我们可以使用这个。

①我们首先将需要引入的包放到lib目录中。

②在pom.xml文件中我可以添加哪个依赖

```
<dependency>
  <groupId>g1</groupId>
  <artifactId>a1</artifactId>
  <version>1.0</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/libs/xxx.jar</systemPath>
</dependency>
```

以上我们就将本地的jar引入到maven repository。

## 6)Maven的继承与聚合：

Maven继承：

也就是说比如有个mvn项目demo-parent其中pom.xml中有testng和selenium这两个依赖。那么我们可以在一个新的mvn项目demo-a中继承这个demo-parent的pom，而不需要重复定义或者避免定义不一致情况。

Project Explorer

- demo-a
  - src/main/java
  - src/main/resources
  - src/test/java
    - (default package)
    - ATC.java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - test-output
  - pom.xml
- demo-parent
  - src/main/java
    - firefox
      - FirefoxOps.java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Dependencies

Filter:

Dependencies

- selenium-java : 2.40.0
- testng : 6.8.7

Add... Remove Properties... Manage...

Dependency Management

TestNG C:\Program Files (x86)\Java\jre7\bin\javaw.exe (Apr 23, 2014 10:54:42 PM)

Default test  
Tests run: 1, Failures: 0, Skips: 0

Default suite  
Total tests run: 1, Failures: 0, Skips: 0

Terry:demo-parent中的依赖testng和selenium

Project Explorer

- demo-a
  - src/main/java
  - src/main/resources
  - src/test/java
    - (default package)
    - ATC.java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - test-output
  - pom.xml
- demo-parent
  - src/main/java
    - firefox
      - FirefoxOps.java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Overview Version is duplicate of parent version

Artifact

Group Id: com.subaccompany

Artifact Id: \* demo-a

Version: 0.0.1-SNAPSHOT

Packaging: jar

Parent

Group Id: \* com.parentcompany

Artifact Id: \* demo-parent

Version: \* 0.0.1-SNAPSHOT

Relative Path: ../demo-parent/pom.xml

Properties

Modules

Dependencies

Filter:

Dependencies

Add... Remove Properties... Manage...

TestNG C:\Program Files (x86)\Java\jre7\bin\javaw.exe (Apr 23, 2014 10:54:42 PM)

Terry:我们就可以定义如下的parent了注意相对pom.xml路劲哦

Project Explorer

- demo-a
  - src/main/java
  - src/main/resources
  - src/test/java
    - (default package)
    - ATC.java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - test-output
  - pom.xml
- demo-parent
  - src/main/java
    - firefox
      - FirefoxOps.java
  - src/main/resources
  - src/test/java
  - src/test/resources
  - JRE System Library [J2SE-1.5]
  - Maven Dependencies
  - src
  - target
  - pom.xml

Dependency Hierarchy [test]

Filter:

Dependency Hierarchy

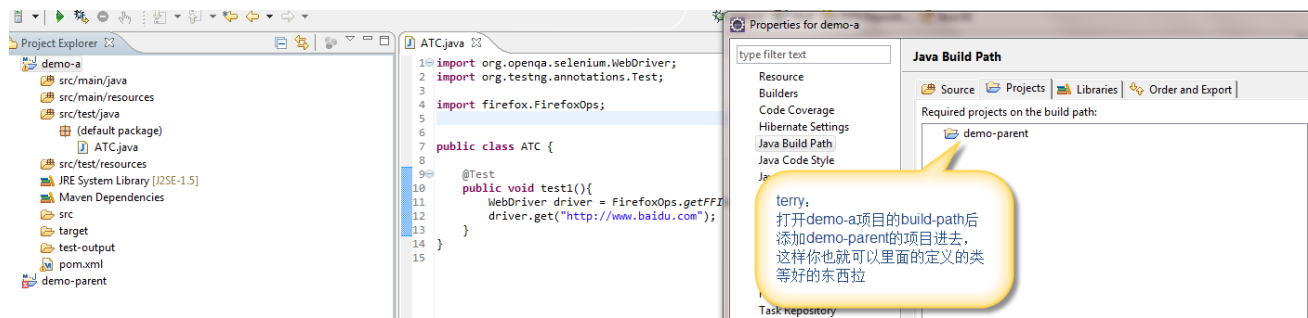
- selenium-java : 2.40.0 [compile]
- testng : 6.8.7 [compile]

Resolved Dependencies

- bsh : 2.0b4 [compile]
- cglib-nodeps : 2.1.3 [compile]
- commons-codec : 1.8 [compile]
- commons-collections : 3.2.1 [compile]
- commons-exec : 1.1 [compile]
- commons-io : 2.2 [compile]
- commons-lang3 : 3.1 [compile]
- commons-logging : 1.1.3 [compile]
- cssparser : 0.9.11 [compile]
- quava : 15.0 [compile]

但是你在hierarchy中也看到继承来的两个jar包





假设你现在公司里自己创建一个maven项目的webdriver的自动化项目很多jar都是在其他项目中定义随着他们的版本的变更而变更。那么我们就可以使用继承了。

## 聚合

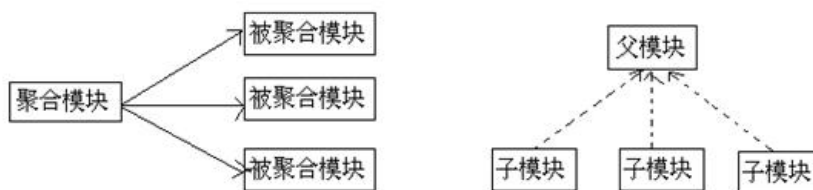
### 三、聚合与继承的关系

#### 区别：

- 1.对于聚合模块来说，它知道有哪些被聚合的模块，但那些被聚合的模块不知道这个聚合模块的存在。
- 2.对于继承关系的父 POM来说，它不知道有哪些子模块继承与它，但那些子模块都必须知道自己的父 POM是什么。

#### 共同点：

- 1.聚合 POM与继承关系中的父POM的 packaging都是pom
- 2.聚合模块与继承关系中的父模块除了 POM之外都没有实际的内容。



聚合关系与继承关系的比较

**注：在现有的实际项目中一个 POM既是聚合POM，又是父 POM，这么做主要是为了方便**

#### Add local Jars

<http://code.google.com/p/addjars-maven-plugin/wiki/UsagePage>

#### 自动化学习网站：

<http://www.supermind.org/blog/968/howto-collect-webdriver-http-request-and-response-headers>

<http://www.guru99.com>