

## Appium-阶段三 - Webapp自动化及持续集成

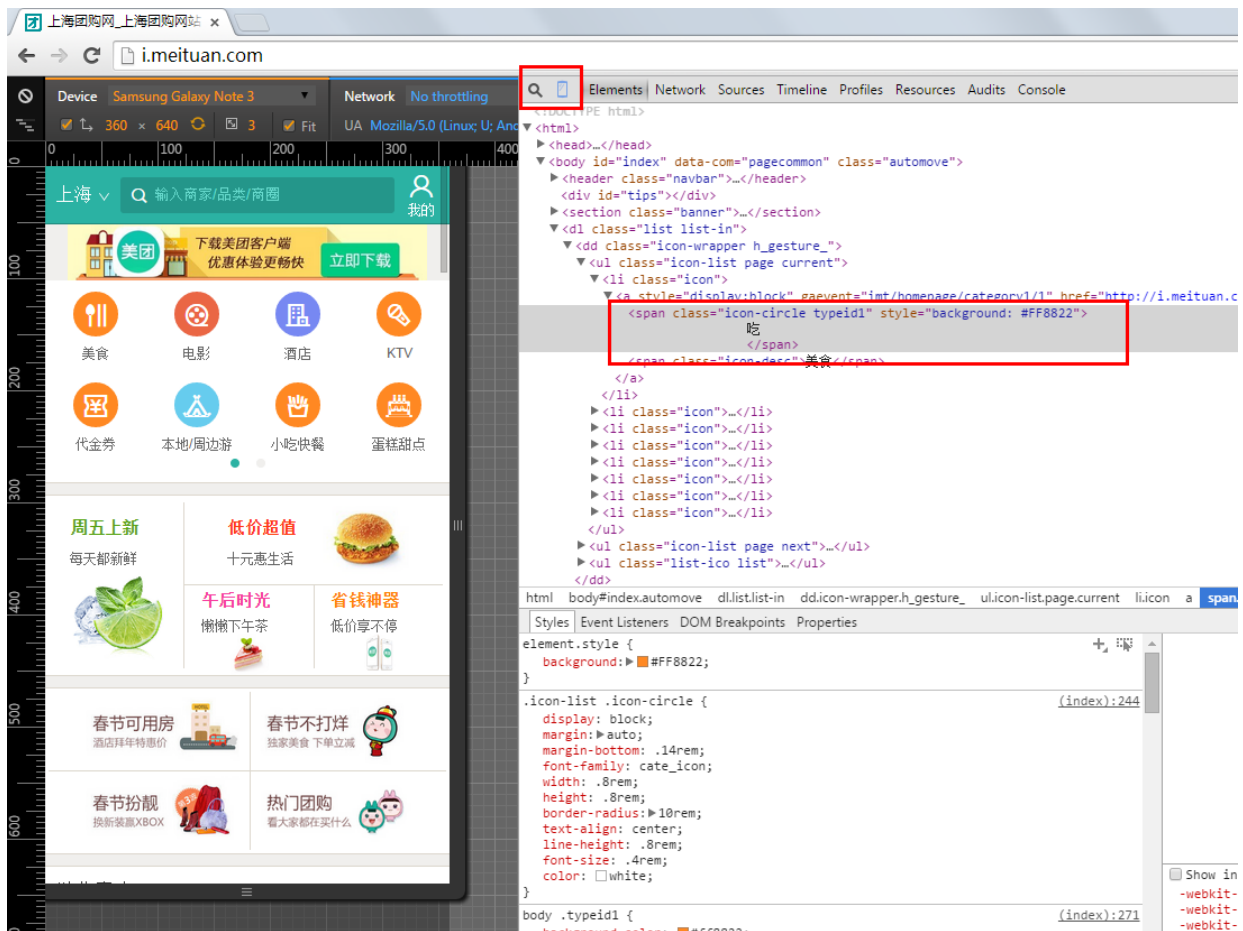
### Webapp自动化:

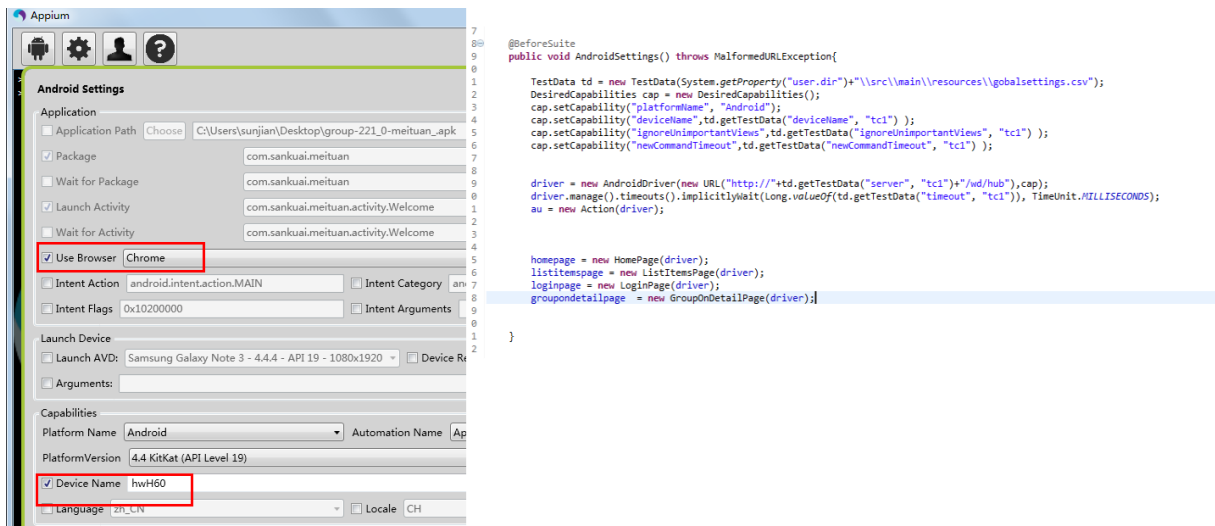
除了native app外,很多时候我们也需要在web app上做我们的自动化测试开发,但是对于webapp,如果要使用appium的话,怎样能抓到页面元素呢?

环境构筑:

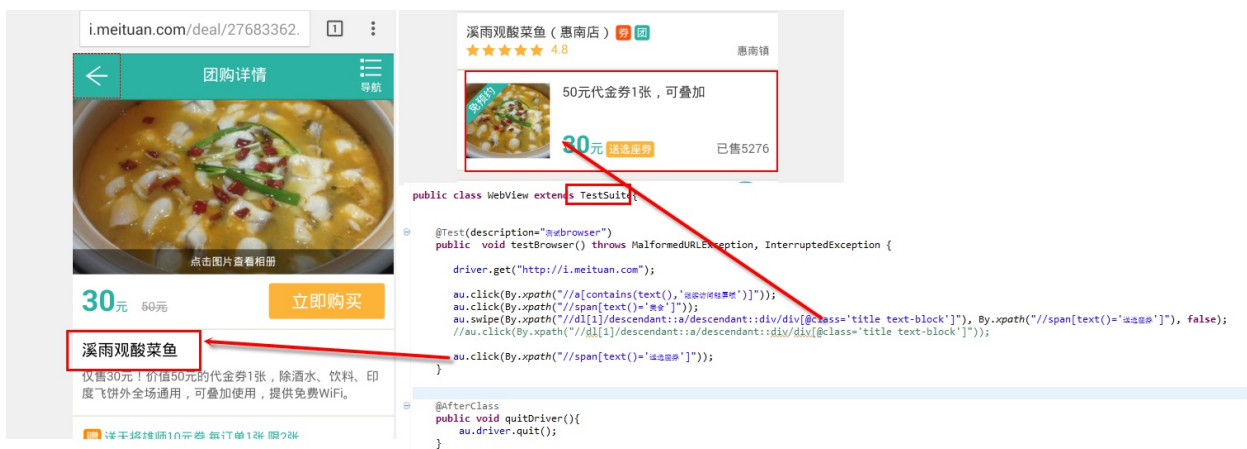
- 1.pc上安装有最新的chrome browser
- 2,下载最新的chromedriver,目前是2.14版本,可以到Terry的工作目录下去下载。
- 3,下载最新的android chrome apk(这个可以到手机厂商对应的应用商店搜索chrome),并安装到手机中

打开美团web app: i.meituan.com,chrome中使用f12后找到手机模拟图标后可以选择device然后我们就可以定位元素了。





大家看一下如下代码我们同样也是继承TestSuite，可以使用同样的Action中的swipe操作和click操作  
注意这里tap就不能用了，因为这个实际也是web，当然可能以后会支持。



大家可以试试做一下，下面的登录。



Wait方式:

对于需要精确等待某些元素的出现或者可点击可以使用如下方式，注意这里是WebDriverWait来自于selenium

```

.19
.20 public void waitForElementPresent(By by){
.21     try{
.22         (new WebDriverWait(driver, timeout)).until(ExpectedConditions.presenceOfElementLocated(by));
.23     }catch(Exception e){
.24     }
.25 }
.26
.27 }
.28
.29
.30
.31
.32 public void waitForElementIsEnable(By by){
.33     (new WebDriverWait(driver, timeout)).until(ExpectedConditions.elementToBeClickable(by));
.34 }
.35

```

对于全局的等待的方式可以使用如下方式，对于webapp和native都适用。

**driver.manage().timeouts().implicitlyWait(Long.valueOf(td.getTestData("timeout", "tc1")), TimeUnit.MILLISECONDS);**

Jenkins:jenkins安装比较简单可以从<http://jenkins-ci.org/>下载，jenkins集成了很多插件，凡是流行的开源软件都有集成  
安装好后我们可以建一个项目

有时候我们的8080端口号可能被占用，如果修改的话可以找到你的安装目录，然后修改jenkins.xml->httpPort=8080修改一下为8089

The screenshot shows the Jenkins web interface. On the left, the 'New Item' button is highlighted with a red arrow. A yellow callout box points to it with the text '这里新建一个maven项目的job'. The main area displays a table of existing jobs:

Name	Last Success	Last Failure	Last Duration
ApplumTraining	1 min 16 sec - #6	N/A	27 sec
DemaA	6 mo 21 days - #26	N/A	16 sec
SeleniumTraining	4 mo 4 days - #8	N/A	21 sec

Below the table, the 'Build Environment' section shows 'SSH Agent' unchecked. The 'Pre Steps' section has an 'Add pre-build step' button. The 'Build' section is highlighted with a red box and contains the following configuration:

- Root POM: pom.xml
- Goals and options: test

The 'Post Steps' section shows three radio buttons for when to run post-build steps: 'Run only if build succeeds', 'Run only if build succeeds or is unstable', and 'Run regardless of build result' (which is selected). Below this, the 'Execute shell' section is highlighted with a red box and contains the command: 'call g:\mvntest.bat'. A link 'See the list of available environment variables' is provided at the bottom.

```

cd %WORKSPACE%
G:\applumInstall\apache-maven-3.1.1-bin\apache-maven-3.1.1\bin\mvn test

```

[Back to Project](#)[Status](#)[Changes](#)[Console Output](#)[Edit Build Information](#)[Delete Build](#)[Git Build Data](#)[No Tags](#)[Promotion Status](#)[Test Result](#)[Redeploy Artifacts](#)[TestNG Results](#)[See Fingerprints](#)

## Package lesson3

0 failures

### All Classes

[hide/expand the table](#)[TestWebView](#)

### Order of Execution by Test

[hide/expand the table](#)[TestWebView.testBrowser](#)

#### JDK

JDK installations

JDK

Name

jdk

JAVA\_HOME

C:\Program Files\Java\jdk1.7.0\_25

☐ Install automatically

Add JDK

List of JDK installations on this system

#### Git

Git installations

Git

Name

git

Path to Git executable

C:\Program Files (x86)\Git\bin\git.exe

☐ Install automatically

Add Git

description

#### Maven

Maven installations

Maven

Name

maven

MAVEN\_HOME

G:\appiumInstall\apache-maven-3.1.1-bin\apache-maven-3.1.1

☐ Install automatically

List of Maven installations on this system

#### Shell

Shell executable

C:\Windows\System32\cmd.exe

#### Extended E-mail Notification