<http://www.tuicool.com/articles/67zeim>

利用selenium Server实现IE firefox 和 chrome兼容性测试 - lightsong

<https://github.com/SeleniumHQ/docker-selenium/>

## StandaloneChromeDebug

<https://github.com/SeleniumHQ/docker-selenium/blob/master/StandaloneChromeDebug/Dockerfile>

FROM selenium/standalone-chrome:2.53.0

MAINTAINER Selenium <selenium-developers@googlegroups.com>

USER root

#=====

# VNC

#=====

RUN apt-get update -qqy \

&& apt-get -qqy install \

x11vnc \

&& rm -rf /var/lib/apt/lists/\* \

&& mkdir -p ~/.vnc \

&& x11vnc -storepasswd secret ~/.vnc/passwd

#=================

# Locale settings

#=================

ENV LANGUAGE en\_US.UTF-8

ENV LANG en\_US.UTF-8

RUN locale-gen en\_US.UTF-8 \

&& dpkg-reconfigure --frontend noninteractive locales \

&& apt-get update -qqy \

&& apt-get -qqy --no-install-recommends install \

language-pack-en \

&& rm -rf /var/lib/apt/lists/\*

#=======

# Fonts

#=======

RUN apt-get update -qqy \

&& apt-get -qqy --no-install-recommends install \

fonts-ipafont-gothic \

xfonts-100dpi \

xfonts-75dpi \

xfonts-cyrillic \

xfonts-scalable \

&& rm -rf /var/lib/apt/lists/\*

#=========

# fluxbox

# A fast, lightweight and responsive window manager

#=========

RUN apt-get update -qqy \

&& apt-get -qqy install \

fluxbox \

&& rm -rf /var/lib/apt/lists/\*

#==============================

# Scripts to run Selenium Node

#==============================

COPY entry\_point.sh /opt/bin/entry\_point.sh

COPY clear\_x\_locks.sh /opt/bin/clear\_x\_locks.sh

RUN chmod +x /opt/bin/entry\_point.sh

RUN chmod +x /opt/bin/clear\_x\_locks.sh

EXPOSE 5900

## standalone-chrome

FROM selenium/node-chrome:2.53.0

MAINTAINER Selenium <selenium-developers@googlegroups.com>

USER root

#====================================

# Scripts to run Selenium Standalone

#====================================

COPY entry\_point.sh /opt/bin/entry\_point.sh

RUN chmod +x /opt/bin/entry\_point.sh

USER seluser

EXPOSE 4444

## node-chrome

FROM selenium/node-base:2.53.0

MAINTAINER Selenium <selenium-developers@googlegroups.com>

USER root

#===============

# Google Chrome

#===============

RUN wget -q -O - https://dl-ssl.google.com/linux/linux\_signing\_key.pub | apt-key add - \

&& echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google-chrome.list \

&& apt-get update -qqy \

&& apt-get -qqy install \

google-chrome-stable \

&& rm /etc/apt/sources.list.d/google-chrome.list \

&& rm -rf /var/lib/apt/lists/\*

#==================

# Chrome webdriver

#==================

ENV CHROME\_DRIVER\_VERSION 2.21

RUN wget --no-verbose -O /tmp/chromedriver\_linux64.zip https://chromedriver.storage.googleapis.com/$CHROME\_DRIVER\_VERSION/chromedriver\_linux64.zip \

&& rm -rf /opt/selenium/chromedriver \

&& unzip /tmp/chromedriver\_linux64.zip -d /opt/selenium \

&& rm /tmp/chromedriver\_linux64.zip \

&& mv /opt/selenium/chromedriver /opt/selenium/chromedriver-$CHROME\_DRIVER\_VERSION \

&& chmod 755 /opt/selenium/chromedriver-$CHROME\_DRIVER\_VERSION \

&& ln -fs /opt/selenium/chromedriver-$CHROME\_DRIVER\_VERSION /usr/bin/chromedriver

#========================

# Selenium Configuration

#========================

COPY config.json /opt/selenium/config.json

#=================================

# Chrome Launch Script Modication

#=================================

COPY chrome\_launcher.sh /opt/google/chrome/google-chrome

RUN chmod +x /opt/google/chrome/google-chrome

USER seluser

# Following line fixes

# https://github.com/SeleniumHQ/docker-selenium/issues/87

ENV DBUS\_SESSION\_BUS\_ADDRESS=/dev/null

## node-base

FROM selenium/base:2.53.0

MAINTAINER Selenium <selenium-developers@googlegroups.com>

ENV DEBIAN\_FRONTEND noninteractive

ENV DEBCONF\_NONINTERACTIVE\_SEEN true

#===================

# Timezone settings

# Possible alternative: https://github.com/docker/docker/issues/3359#issuecomment-32150214

#===================

ENV TZ "US/Pacific"

RUN echo "${TZ}" > /etc/timezone \

&& dpkg-reconfigure --frontend noninteractive tzdata

#==============

# VNC and Xvfb

#==============

RUN apt-get update -qqy \

&& apt-get -qqy install \

xvfb \

&& rm -rf /var/lib/apt/lists/\*

#==============================

# Scripts to run Selenium Node

#==============================

COPY \

entry\_point.sh \

functions.sh \

/opt/bin/

RUN chmod +x /opt/bin/entry\_point.sh

#============================

# Some configuration options

#============================

ENV SCREEN\_WIDTH 1360

ENV SCREEN\_HEIGHT 1020

ENV SCREEN\_DEPTH 24

ENV DISPLAY :99.0

USER seluser

CMD ["/opt/bin/entry\_point.sh"]

## selenium/base

FROM ubuntu:16.04

MAINTAINER Selenium <selenium-developers@googlegroups.com>

#================================================

# Customize sources for apt-get

#================================================

RUN echo "deb http://archive.ubuntu.com/ubuntu xenial main universe\n" > /etc/apt/sources.list \

&& echo "deb http://archive.ubuntu.com/ubuntu xenial-updates main universe\n" >> /etc/apt/sources.list \

&& echo "deb http://security.ubuntu.com/ubuntu xenial-security main universe\n" >> /etc/apt/sources.list

#========================

# Miscellaneous packages

# Includes minimal runtime used for executing non GUI Java programs

#========================

RUN apt-get update -qqy \

&& apt-get -qqy --no-install-recommends install \

bzip2 \

ca-certificates \

openjdk-8-jre-headless \

sudo \

unzip \

wget \

&& rm -rf /var/lib/apt/lists/\* \

&& sed -i 's/securerandom\.source=file:\/dev\/random/securerandom\.source=file:\/dev\/urandom/' ./usr/lib/jvm/java-8-openjdk-amd64/jre/lib/security/java.security

#==========

# Selenium

#==========

RUN mkdir -p /opt/selenium \

&& wget --no-verbose https://selenium-release.storage.googleapis.com/2.53/selenium-server-standalone-2.53.0.jar -O /opt/selenium/selenium-server-standalone.jar

#========================================

# Add normal user with passwordless sudo

#========================================

RUN sudo useradd seluser --shell /bin/bash --create-home \

&& sudo usermod -a -G sudo seluser \

&& echo 'ALL ALL = (ALL) NOPASSWD: ALL' >> /etc/sudoers \

&& echo 'seluser:secret' | chpasswd

## Driver

<http://chromedriver.storage.googleapis.com/index.html>

<http://selenium-release.storage.googleapis.com/index.html>

## WebElement操作

<http://blog.sina.com.cn/s/blog_6966650401012a7q.html>

@FindBy(css = **"#mcon i"**)  
 **private** WebElement **headerText**;  
 @FindBy(id = **"kw"**)  
 **private** WebElement **editThisPage**;  
*//@FindBys 相当于是取交集，是先根据第一个注解获取到对应元素，然后根据第二个注解再帅选出对应的页面元素,。* @FindBys({  
 @FindBy(className = **"A"**),  
 @FindBy(className = **"B"**)  
 })  
 *//@FindAll相当于是取并集，如找到id=A和id=B的所有元素* **public** WebElement **AB**;  
 @FindAll({  
 @FindBy(id = **"A"**),  
 @FindBy(id = **"B"**)  
 })  
 **public** List<WebElement> **aAndB**;

WebDriver拾级而上·之四 操作页面元素

 (2012-05-16 22:48:00)

[IMG_256](http://blog.sina.com.cn/s/javascript:;)转载▼

|  |  |
| --- | --- |
| 标签： [selenium](http://search.sina.com.cn/?c=blog&q=selenium&by=tag" \t "http://blog.sina.com.cn/s/_blank)  [webdriver](http://search.sina.com.cn/?c=blog&q=webdriver&by=tag" \t "http://blog.sina.com.cn/s/_blank)  [it](http://search.sina.com.cn/?c=blog&q=it&by=tag" \t "http://blog.sina.com.cn/s/_blank) | 分类： [Selenium](http://blog.sina.com.cn/s/articlelist_1768318212_6_1.html" \t "http://blog.sina.com.cn/s/_blank) |

一、输入框（text field or textarea）  
//找到输入框元素：  
WebElement element = driver.findElement(By.id("passwd-id"));  
//将输入框清空：  
element.clear();  
//在输入框中输入内容：  
element.sendKeys(“test”);  
//获取输入框的文本内容：  
element.getText();  
  
二、下拉选择框(Select)  
//找到下拉选择框的元素：  
Select select = new Select(driver.findElement(By.id("select")));  
   
//选择对应的选择项：  
select.selectByVisibleText(“mediaAgencyA”);  
或  
select.selectByValue(“MA\_ID\_001”);  
   
//不选择对应的选择项：  
select.deselectAll();  
select.deselectByValue(“MA\_ID\_001”);  
select.deselectByVisibleText(“mediaAgencyA”);  
或者获取选择项的值：  
select.getAllSelectedOptions();  
select.getFirstSelectedOption();  
  
对下拉框进行操作时首先要定位到这个下拉框，new 一个Selcet对象，然后对它进行操作  
  
例如：  
以http://passport.51.com/reg2.5p这个页面为例。这个页面中有4个下拉框，下面演示4种选中下拉框选项的方法。  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.support.ui.Select;  
  
public class SelectsStudy {  
    public static void main(String[] args) {  
        System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\\firefox.exe");   
        WebDriver dr = new FirefoxDriver();  
        dr.get("http://passport.51.com/reg2.5p");  
         
        //通过下拉列表中选项的索引选中第二项，即2011年  
        Select selectAge = new Select(dr.findElement(By.id("User\_Age")));  
        selectAge.selectByIndex(2);//Select.selectByIndex  
         
        //通过下拉列表中的选项的value属性选中"上海"这一项  
        Select selectShen = new Select(dr.findElement(By.id("User\_Shen")));  
        selectShen.selectByValue("上海");//Select.selectByValue  
         
        //通过下拉列表中选项的可见文本选 中"浦东"这一项  
        Select selectTown = new Select(dr.findElement(By.id("User\_Town")));  
        selectTown.selectByVisibleText("浦东");Select.selectByVisibleText  
         
        //这里只是想遍历一下下拉列表所有选项，用click进行选中选项  
        Select selectCity = new Select(dr.findElement(By.id("User\_City")));  
        for(WebElement e : selectCity.getOptions())//Select.getOptions()  
            e.click();  
    }  
}  
  
三、单选项(Radio Button)  
//找到单选框元素：  
WebElement bookMode =driver.findElement(By.id("BookMode"));  
//选择某个单选项：  
bookMode.click();  
//清空某个单选项：  
bookMode.clear();  
//判断某个单选项是否已经被选择：  
bookMode.isSelected();  
  
四、多选项(checkbox)  
//多选项的操作和单选的差不多：  
WebElement checkbox =driver.findElement(By.id("myCheckbox."));  
checkbox.click();  
checkbox.clear();  
checkbox.isSelected();  
checkbox.isEnabled();  
  
五、按钮(button)  
//找到按钮元素：  
WebElement saveButton = driver.findElement(By.id("save"));  
//点击按钮：  
saveButton.click();  
//判断按钮是否enable:  
saveButton.isEnabled ();  
  
六、左右选择框  
也就是左边是可供选择项，选择后移动到右边的框中，反之亦然。  
例如：  
Select lang = new Select(driver.findElement(By.id("languages")));  
lang.selectByVisibleText(“English”);  
WebElement addLanguage =driver.findElement(By.id("addButton"));  
addLanguage.click();  
  
七、弹出对话框(Popup dialogs)  
Alert alert = driver.switchTo().alert();  
alert.accept();  
alert.dismiss();  
alert.getText();  
  
八、表单(Form)  
Form中的元素的操作和其它的元素操作一样，对元素操作完成后对表单的提交可以：  
WebElement approve = driver.findElement(By.id("approve"));  
approve.click();  
或  
approve.submit();//只适合于表单的提交  
  
九、上传文件 (Upload File)  
上传文件的元素操作：  
WebElement adFileUpload = driver.findElement(By.id("WAP-upload"));  
String filePath = "C:\test\\uploadfile\\media\_ads\\test.jpg";  
adFileUpload.sendKeys(filePath);  
  
  
十、拖拉(Drag andDrop)  
WebElement element =driver.findElement(By.name("source"));  
WebElement target = driver.findElement(By.name("target"));  
(new Actions(driver)).dragAndDrop(element, target).perform();  
  
例如：下面这个页面是一个演示拖放元素的页面，你可以把左右页面中的条目拖放到右边的div框中。  
http://koyoz.com/demo/html/drag-drop/drag-drop.html  
  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.interactions.Actions;  
  
public class DragAndDrop {  
    public static void main(String[] args) {  
        System.setProperty("webdriver.firefox.bin","D:\\Program Files\\Mozilla Firefox\\firefox.exe");   
        WebDriver dr = new FirefoxDriver();  
        dr.get("http://koyoz.com/demo/html/drag-drop/drag-drop.html");  
         
        //首先new出要拖入的页面元素对象和目标对象，然后进行拖入。  
        WebElement  element = dr.findElement(By.id("item1"));  
        WebElement  target = dr.findElement(By.id("drop"));  
        (new Actions(dr)).dragAndDrop(element, target).perform();  
         
        //利用循环把其它item也拖入  
        String id="item" ;  
        for(int i=2;i<=6;i++){  
            String item = id+i;  
            (new Actions(dr)).dragAndDrop(dr.findElement(By.id(item)), target).perform();  
        }  
    }  
}  
代码很简单，需要注意的是(new Actions(dr)).dragAndDrop(element, target).perform();这句话中，dragAndDrop(element, target)这个方法是定义了“点击element元素对象，然后保持住，直到拖到目标元素对象里面才松开”这一系列动作的Actions,如果你不调用perform()方法，这个Actions是不会执行的。  
  
  
十一、导航 (Navigationand History)  
//打开一个新的页面：  
 driver.navigate().to("http://www.example.com");  
//通过历史导航返回原页面：  
driver.navigate().forward();  
driver.navigate().back();

十二、获取页面CSS属性

1.获取文字颜色

dr.findElement(By.id("tooltip")).getCssValue("color")

2.获取文字字号

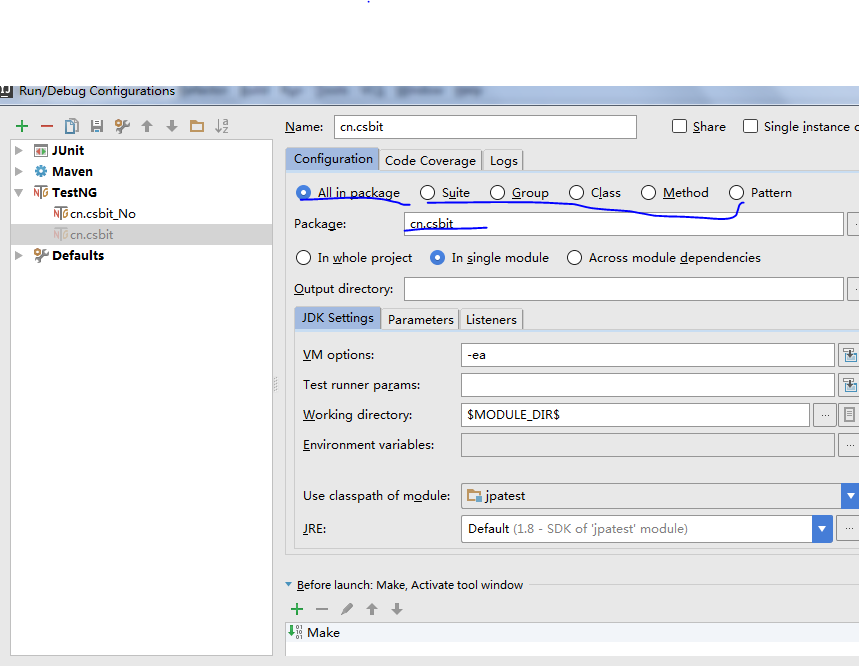
dr.findElement(By.tagName("h3")).getCssValue("font")

## 工程运行

运行方式有多种

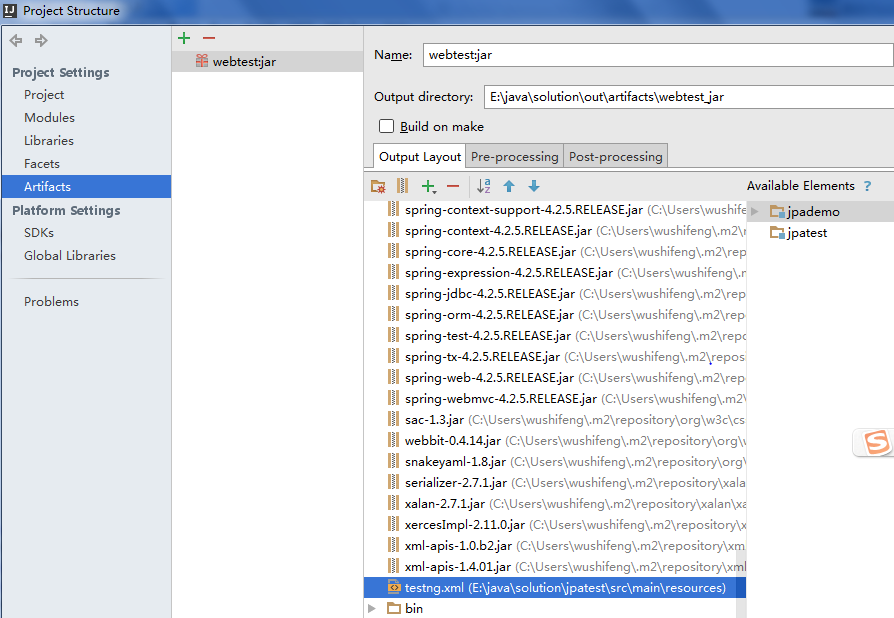
### IDEA IDE下运行

使用的是集成的testng测试



可以分组、单个类、基于testng文件运行

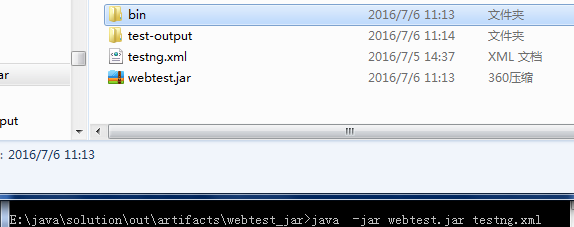
### jar打包



使用idea的打包，注意：

jpatest\src\main\resources\META-INF\MANIFEST.MF org\eclipse\jdt\internal\jarinjarloader 是从Eclipse中拿过来的【默认的jar打包总是提示找不到类】

运行时需要将 testng.xml和bin目录拷贝在一起，才能运行



### 独立的class和依赖包

和jar打包类似，只不过把工程的class文件放在根下，jar包放在lib下，

运行testng.bat即可