# Selenium webDriver

## JDK1.8以上才支持selenium/appium

### 截取验证码：



package com.app.test;

import java.awt.image.BufferedImage;

import java.io.\*;

import javax.imageio.ImageIO;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.Point;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class capature {

public static void main(String[] args) throws IOException {

System.setProperty("webdriver.chrome.driver","D:\\selenium\\chromedriver.exe");

WebDriver driver = new ChromeDriver();

driver.get("http://211.139.11.136:18123/");

WebElement ele = driver.findElement(By.id("captchaImg"));

// Get entire page screenshot让整个页面截图

java.io.File screenshot = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

BufferedImage fullImg = ImageIO.read(screenshot);

// Get the location of element on the page获取页面上元素的位置

Point point = ele.getLocation();

// Get width and height of the element得到元素的高度、宽度

int eleWidth = ele.getSize().getWidth();

int eleHeight = ele.getSize().getHeight();

// Crop the entire page screenshot to get only element screenshot裁剪整个页面截图，只获取元素截图

BufferedImage eleScreenshot= fullImg.getSubimage(point.getX(), point.getY(),

eleWidth, eleHeight);

ImageIO.write(eleScreenshot, "png", screenshot);

// Copy the element screenshot to disk将元素截图复制到磁盘

File screenshotLocation = new File("D:/test.png");

FileUtils.copyFile(screenshot, screenshotLocation);

}

}

### 1.1  下载selenium2.0的包

官方download包地址：<http://code.google.com/p/selenium/downloads/list>

官方User Guide：　　 <http://seleniumhq.org/docs/>

官方API:　　　　　　  <http://selenium.googlecode.com/git/docs/api/java/index.html>

### 1.2.**1**用webdriver打开一个浏览器

打开firefox浏览器：

        　　　　WebDriver driver = new FirefoxDriver();

打开IE浏览器

       　　 　　WebDriver driver = new InternetExplorerDriver ();

打开HtmlUnit浏览器

        　　　　WebDriverdriver = new HtmlUnitDriver();

打开chrome浏览器

　　　　　　　WebDriverdriver = new ChromeDriver();

### 1.2.**2**最大化浏览器

　　WebDriver driver = new FirefoxDriver();  
　　driver.manage().window().maximize();

### 1.2.3 关闭浏览器

WebDriver driver = new FirefoxDriver();

　　driver.close();

　　driver.quit();

### 1.**3**打开测试页面

driver.get("http://www.google.com");

driver.navigate().to("http://www.baidu.com/");

　　　　　　P.S.navigate方法会产生1个Navigator对象，其封装了与导航相关的一些方法，比如前进后退等

### 1.4  页面元素定位

Webdriver提供下面两种方法来定位页面元素，参数是By对像，最常用是By.id和By.name查找。

findElement　　 定位某个元素，如果没有找到元素会抛出异常:NoSuchElementException

findElements     定位一组元素

 例如需要定位如下元素：

　　<input class="input\_class" type="text" name="passwd" id="passwd-id" />

By.id：

　　　　　　WebElement element = driver.findElement(By.id("passwd-id"));

By.name：

　　　　　　WebElement element = driver.findElement(By.name("passwd"));

By.xpath：

　　　　　　WebElement element =driver.findElement(By.xpath("//input[@id='passwd-id']"));

By.className

　　　　　　WebElement element = driver.findElement(By.className("input\_class"));

By.cssSelector

　　　　　　WebElement element = driver.findElement(By.cssSelector(".input\_class"));

By.linkText:

　　　　　　//通俗点就是精确查询

　　　　　　WebDriver driver = new FirefoxDriver();  
　　　　　　driver.get("http://www.baidu.com/");   
　　　　　　WebElement element = driver.findElement(By.linkText("百科"));

By.partialLinkText：

　　　　　　//这个方法就是模糊查询  
　　　　　　WebDriver driver = new FirefoxDriver();  
　　　　　　driver.get("http://www.baidu.com/");   
　　　　　　WebElement element = driver.findElement(By.partialLinkText("hao"));

By.tagName:

　　　　　　WebDriver driver = new FirefoxDriver();  
　　　　　　driver.get("http://www.baidu.com/");   
　　　　　　String test= driver.findElement(By.tagName("form")).getAttribute("name");  
　　　　　　System.out.println(test);

### 1.**5**如何对页面元素进行操作

### 1.5.1 输入框（text field or textarea）

WebElement element = driver.findElement(By.id("passwd-id"));

element.sendKeys(“test”);//在输入框中输入内容：

element.clear();　　　　   //将输入框清空

element.getText();　　   //获取输入框的文本内容：

### 1.5.2下拉选择框(Select)

Select select = new Select(driver.findElement(By.id("select")));

select.selectByVisibleText(“A”);

select.selectByValue(“1”);

select.deselectAll();

select.deselectByValue(“1”);

select.deselectByVisibleText(“A”);

select.getAllSelectedOptions();

select.getFirstSelectedOption();

### 1.5.3单选项(Radio Button)

WebElement radio=driver.findElement(By.id("BookMode"));

radio.click();　　　　   //选择某个单选项

radio.clear();　　　　  //清空某个单选项

radio.isSelected();　　//判断某个单选项是否已经被选择

### 1.5.4多选项(checkbox)

WebElement checkbox = driver.findElement(By.id("myCheckbox."));

checkbox.click();

checkbox.clear();

checkbox.isSelected();

checkbox.isEnabled();

### 1.5.5按钮(button)

WebElement btn= driver.findElement(By.id("save"));

btn.click();　　　　　 //点击按钮

btn.isEnabled ();　　//判断按钮是否enable

### 1.5.6弹出对话框(Popup dialogs)

Alert alert = driver.switchTo().alert();

alert.accept();　　//确定

alert.dismiss();　 //取消

alert.getText();　//获取文本

### 1.5.7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

### 1.5.8上传文件

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);

### 1.6  Windows 和 Frames之间的切换

driver.switchTo().defaultContent();　　　　　//返回到最顶层的frame/iframe

driver.switchTo().frame("leftFrame");　　　　//切换到某个frame：

driver.switchTo().window("windowName");　//切换到某个window

### 1.7  调用Java Script

Web driver对Java Script的调用是通过JavascriptExecutor来实现的，例如：

JavascriptExecutor js = (JavascriptExecutor) driver;

        js.executeScript("JS脚本");

### 1.**8**超时设置

WebDriver driver = new FirefoxDriver();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);　　　　  //识别元素时的超时时间

driver.manage().timeouts().pageLoadTimeout(10, TimeUnit.SECONDS);　　//页面加载时的超时时间

driver.manage().timeouts().setScriptTimeout(10, TimeUnit.SECONDS);　　//异步脚本的超时时间