03 WebDriver –基本使用

本章大纲

3.1 打开关闭浏览器

3.2 WebDriver对浏览器操作

3.3 访问链接-打印页面信息

3.4 自定义profile

3.5 元素定位

3.6 定位一组对象

打开浏览器

- 火狐浏览器

System.*setProperty("webdriver.gecko.driver", "D:\\demo\\geckodriver.exe");*

System.*setProperty("webdriver.firefox.bin", "D:\\Program Files\\Mozilla* Firefox\\firefox.exe");

WebDriver wd = new FirefoxDriver();

- 谷歌浏览器

System.setProperty("webdriver.chrome.driver", "D://demo//chromedriver.exe");

WebDriver driver = new ChromeDriver();

- IE浏览器

System.setProperty("webdriver.ie.driver", "D://demo//IEDriverServer.exe");

WebDriver wd = new InternetExplorerDriver ();

关闭浏览器

close方法关闭当前的浏览器窗口；
quit方法不仅关闭窗口，还会彻底的退出webdriver，释放与driver server之间的连接。
所以简单来说quit是更加彻底的close，quit会更好的释放资源。

```
    @AfterMethod
    public void closeBrowser() {
        driver.quit();
//      driver.close();
    }
```

本章大纲

3.1 打开关闭浏览器

3.2 WebDriver对浏览器操作

3.3 访问链接-打印页面信息

3.4 自定义profile

3.5 元素定位

3.6 定位一组对象

访问某网页地址

String baseUrl="https://www.baidu.com/";

- 方法1：

```
@Test
public void visitURL(){
        driver.get(baseUrl);

}
```

- 方法2：

```
@Test
public void visitURL(){
        driver.navigate().to(baseUrl);
}
```

返回/前进/刷新

```
@Test
public void visitRecentURL() {
      driver.get(url1);
      driver.get(url2);
      driver.navigate().back();
      driver.navigate().forward();
      driver.navigate().refresh();
}
```

浏览器尺寸的控制

浏览器最大化

```
driver.manage().window().maximize()
```

设置浏览器大小

```
Dimension  size = new Dimension (300,400);
driver.manage().window().setSize(size);
```

本章大纲

3.1 打开关闭浏览器

3.2 WebDriver对浏览器操作

3.3 访问链接-打印页面信息

3.4 自定义profile

3.5 元素定位

3.6 定位一组对象

获取页面信息

```
@Test
public void getPage() {
    driver.get(baseUrl);
    String title = driver.getTitle();// 获得title
    String pageSource = driver.getPageSource();// 获得源代码
    String pageUrl = driver.getCurrentUrl();
    System.out.println(title + pageSource + pageUrl);
}
```

本章大纲

3.1 打开关闭浏览器

3.2 WebDriver对浏览器操作

3.3 访问链接-打印页面信息

3.4 自定义profile

3.5 元素定位

3.6 定位一组对象

自定义profile

ProfilesIni allprofiles = **new ProfilesIni();**

FirefoxProfile pro = allprofiles.getProfile("tom");

FirefoxOptions options = **new FirefoxOptions();**

options.setProfile(pro);

System.*setProperty("webdriver.gecko.driver", "D:\\demo\\geckodriver.exe");*

System.*setProperty("webdriver.firefox.bin", "D:\\Program Files\\Mozilla Firefox\\firefox.exe");*

WebDriver driver = **new FirefoxDriver(options);**

查看 firefox.exe -p -no-remote

本章大纲

3.1 打开关闭浏览器

3.2 WebDriver对浏览器操作

3.3 访问链接-打印页面信息

3.4 自定义profile

3.5 元素定位

3.6 定位一组对象

测试程序对页面元素的操作步骤

自动化测试实施过程中，测试程序对页面元素的操作步骤如下：

1. 定位页面元素，储存到一个变量中
2. 对变量中存储的页面元素进行操作，例如：单击链接、选择下拉列表或输入文字
3. 设定页面元素的操作值，例如：输入什么文字，选择哪一项

页面元素的定位方式

定位方法	实例
ID	driver.findElement(By.id(“id值”))
name	driver.findElement(By.name(“name值”))
链接的全部文字	driver.findElement(By.linkText(“linkText”))
部分链接文字	driver.findElement(By.partialLinkText(“部分linkText值”))
xpath方式	driver.findElement(By.xpath(“xpath 值”))
CSS	driver.findElement(By.cssSelector(“css定位表达式”))
Class名称	driver.findElement(By.className(“class属性值”))
标签名称	driver.findElement(By.tagName(“标签名称”))
jQuery	js.executeScript("return jQuery.find(“css定位表达式')")

页面元素的定位方式

selenium查找元素是通过“By”这个类指定定位方式的

Method Summary

Methods

Modifier and Type	Method and Description
static By	className(java.lang.String className) Finds elements based on the value of the "class" attribute.
static By	cssSelector(java.lang.String selector) Finds elements via the driver's underlying W3 Selector engine.
boolean	equals(java.lang.Object o)
WebElement	findElement(SearchContext context) Find a single element.
abstract java.util.List<WebElement>	findElements(SearchContext context) Find many elements.
int	hashCode()
static By	id(java.lang.String id)
static By	linkText(java.lang.String linkText)
static By	name(java.lang.String name)
static By	partialLinkText(java.lang.String linkText)
static By	tagName(java.lang.String name)
java.lang.String	toString()
static By	xpath(java.lang.String xpathExpression)

id

- 通过 By.id(“ID值")在页面上查找元素

<input id="kw" class="s_ipt" name="wd" maxlength="100" autocomplete="off" type="text">

WebElement element = driver.findElement(By.id(“kw"));

WebElement submitButton= driver.findElement(By.id(“submit"));

submitButton.click();

练习：实现百度搜索的测试，输入“淘宝”，进行搜索

name

- 通过 By.name(“name值")在页面上查找元素

```
<input id="kw" class="s_ipt" name="wd" maxlength=
100" autocomplete="off" type="text">
WebElement element =
driver.findElement(By.name( “wd"));
```

className

- 通过 By.className(“页面元素的class属性值”)在页面上查找元素

<input id="kw" class="s_ipt" name="wd" maxl
ength="100" autocomplete="off" type="text">

WebElement element =
driver.findElement(By.*className*(“s_ipt "));

tagName

- 通过 By.tagName(“页面中的html标签名称”)在页面上查找元素

<input id="kw" class="s_ipt" name="wd" maxlength="100" autocomplete="off" type="text">

WebElement element = driver.findElements(By.*tagName*(“input "));

- 因为同一个页面相同的tagName会有多个，建议使用findElements()来查找

WebElement searchInput= driver.findElements(By.*tagName*(“a"));

List<WebElement> scriptList= driver.findElements(By.*tagName*(“a"));

linkTest

- 通过 By.linkText(“链接的全部文字内容")

```
<a class="mnav" href="http://news.baidu.com" target="_blank">新闻</a>
WebElement newsLink= driver.findElement(By.linkText(“新闻"));
newsLink.click();
```

partialLinkText

- parial link定位是对link定位的一种补充，有些文本链接会比较长，这个时候可以取文本链接的一部分定位，只要这一部分信息可以唯一地标识这个链接。
- 通过 By. partialLinkText(“链接的部分文字内容")在页面上查找元素

```
<a class="mnav" href="http://www.hao123.com" target="_blank">hao123</a>
WebElement pLink= driver.findElement(By.partialLinkText(“123"));
```

findElement()只会返回页面上第一个满足partialLinkText为元素。

xpath

XPath 是一门在 XML 文档中查找信息的语言。XPath 可用来在 XML 文档中对元素和属性进行遍历。
http://www.w3school.com.cn/xpath/

- 绝对路径定位（不建议使用）

WebElement button= driver.findElement(By.xpath(“/html/body/div[1]/input”))

例如div[1]表示当前层级下的第一个div标签。
driver.findElement(By.xpath(“/html/body/div[1]/html/body/div[1]/input[@value=‘查询’]”))

- 相对路径定位（结合属性值）

WebElement button= driver.findElement(By.xpath(“//input[@value=‘查询’]”))
从本页面中所有节点查找属性是value的并且value的属性值是查询的。或者省略标签
driver.findElement(By.xpath(“//*[@value=‘查询’]”))
findElement(By.xpath("//input[@id='kw' and @class='su']/span/input"))

- 使用索引号进行定位

WebElement button= driver.findElement(By.xpath(“//input[2]”));

- 模糊的属性值

WebElement button= driver.findElement(By.xpath(“//img[starts-with(@alt,’div1’)]”))
By.xpath("//input[ends-with(@id,'很漂亮')]”)
WebElement button= driver.findElement(By.xpath(“//img[contains(@alt,’div1’)]”))

xpath

- 使用页面元素的文本来定位元素

WebElement button= driver.findElement(By.xpath(“// a[text()＝’百度搜索’)]”))

WebElement button= driver.findElement(By.xpath(“//a[contains(text(),’百度’)]”))

xpath 轴(Axis)（了解）

xpath轴关键字	轴的含义说明	实例
parent	选择当前节点的上层父节点	//img[@alt=‘div2-img2’]/parent::div
child	选择当前节点的下层父节点	//div[@id=‘div1’]/child::img
ancestor	选择当前节点的所有上层节点	//img[@alt=‘div2-img2’]/ancestor::div
descendant	选择当前节点的所有下层节点	//div[@id=‘div1’]/descendant::img
following	选择当前节点之后显示的所有节点	//div[@id=‘div1’]/following::img
following-sibling	选择当前节点的所有后面的平级节点	//a[@href=‘’]/following-sibling:input
preceding	选择当前节点前面显示的所有节点	//img[@alt=‘div2-img2’]/preceding::div
preceding-sibling	选择当前节点前面显示的所有同级节点	//img[@alt=‘div2-img2’]/preceding-sibling::div

- Cascading Style Sheets (CSS)是一种样式风格语言用来描述元素的外观和格式。主流的浏览器实现 CSS 解析引擎使用 CSS 语法来格式化和样式化页面。CSS 的引进是为了让 页 面 信 息 和 样 式 信 息 可 以 分 开 。
- css定位速度要比xpath快
- 通过By.cssSelector("css定位表达式")在页面上查找元素

- 使用绝对路径来定位元素（不推荐）

WebElement userName = driver.findElement(By.cssSelector("html body div div form input"));

- 也可以以父子关系的方式”>”来描述这个选择器
  WebElement userName = driver.findElement(By.cssSelector("html >body > div > div > form > input"));

By.cssSelector("html>body>div[1]>input[type=‘button’]"));

css selector

- 相对路径

可以用这样的方法来定位用户输入字段，假设它在 DOM 中是第一个<input>元素：

WebElement userName = driver.findElement(By.cssSelector("input"));

- 相对路径（结合属性定位）

By.cssSelector("input[type= 'button']");

- class属性

By.cssSelector("input.spread");

- id属性

By.cssSelector("input#divinput");

- 复合属性

*By.cssSelector("img[alt= 'aaa'][href= '****']");*

- 模糊匹配

开头包含　　a[href^= 'http://www.baidu']

结尾包含 a[href$= 'baidu.com']

包含 a[href*= 'baidu']

- 子页面元素的查找

By.cssSelector("div#div1>input#divinput1");

XPath与CSS的类似功能的对比

定位方式	XPath	CSS
标签	//div	div
By id	//div[@id='eleid']	div#eleid
By class	//div[@class='eleclass']	div.eleid
By 属性	//div[@title='Move mouse here']	div[title=Move mouse here] div[title^=Move] div[title$=here] div[title*=mouse]
定位子元素	//div[@id='eleid']/* //div/h1	div#eleid>* div >h1

遍历表格

```
WebElement element = driver.findElement(By.id("table1"));
List<WebElement> rows = element.findElements(By.tagName("tr"));

for(WebElement row : rows){
    List<WebElement> cols = row.findElements(By.tagName("td"));
    for(WebElement col : cols){
        System.out.print(col.getText()+"\t");
}
        System.out.println();
```

- 问题：怎么显示第二行第一列的单元格呢?

- WebDriver Element Locator