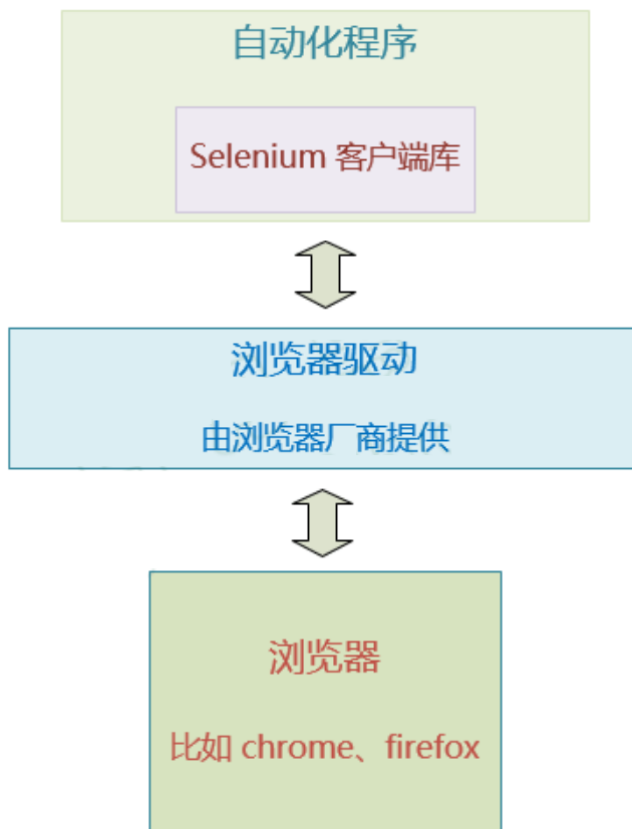


1. Selenium简介

Selenium 是一个 Web 应用的自动化测试工具。

通过它，我们可以写出自动化程序，模拟用户在浏览器里操作web界面。比如点击界面按钮，在文本框中输入文字等操作。



selenium 自动化流程如下：

1. 自动化程序调用Selenium客户端库函数（比如点击按钮元素）
2. 客户端库会发送Selenium命令给浏览器的驱动程序
3. 浏览器驱动程序接收到命令后，驱动浏览器去执行命令
4. 浏览器执行命令
5. 浏览器驱动程序获取命令执行的结果，返回给自动化程序
6. 自动化程序对返回结果进行处理

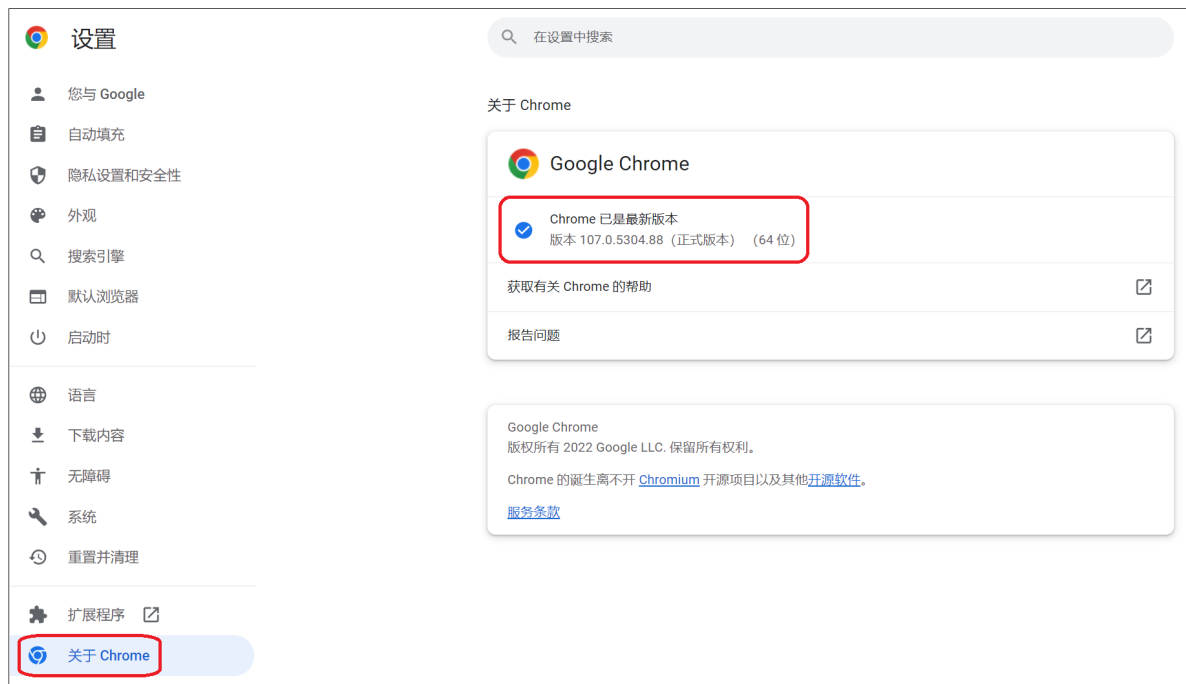
2. 基于Java语言的Selenium 环境搭建

① 创建Maven项目，在pom.xml中导入依赖：

```
<dependencies>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-server-standalone</artifactId>
    <version>2.53.0</version>
  </dependency>
</dependencies>
```

② 准备浏览器对应的驱动器：

IE使用IEDriverServer，Firefox使用geckodriver，Chrome使用chromedriver，此处以Chrome为例。



Chrome可在设置中查看浏览器版本，然后下载对应版本的浏览器驱动（如果找不到就选择最接近的版本）

浏览器驱动下载地址：

- 版本号 < 115，在 <https://chromedriver.storage.googleapis.com/index.html> 中找到对应版本的文件夹下载浏览器驱动。Windows平台下载其中的 `chromedriver_win32.zip`，解压后得到 `chromedriver.exe`。
- 版本号 ≥ 115，在 <https://googlechromelabs.github.io/chrome-for-testing/known-good-versions-with-downloads.json> 中找到对应版本的chromedriver下载链接。Windows平台下载其中的 `chromedriver-win64.zip`，解压后得到 `chromedriver.exe`。

3. 指明浏览器驱动，打开对应浏览器，完成指定操作后关闭

```
// 创建WebDriver对象，指明使用chrome浏览器驱动
System.setProperty("webdriver.chrome.driver", "C:\\...\\chromedriver.exe");
WebDriver driver = new ChromeDriver();

// 调用WebDriver对象的get方法 可以让浏览器打开指定网址
driver.get("https://www.baidu.com");

//-----
// 要执行的操作（待补充）
//-----

// 设置等待时间，等待4s
Thread.sleep(4000);

// 关闭页面并结束相关进程
driver.quit();
```

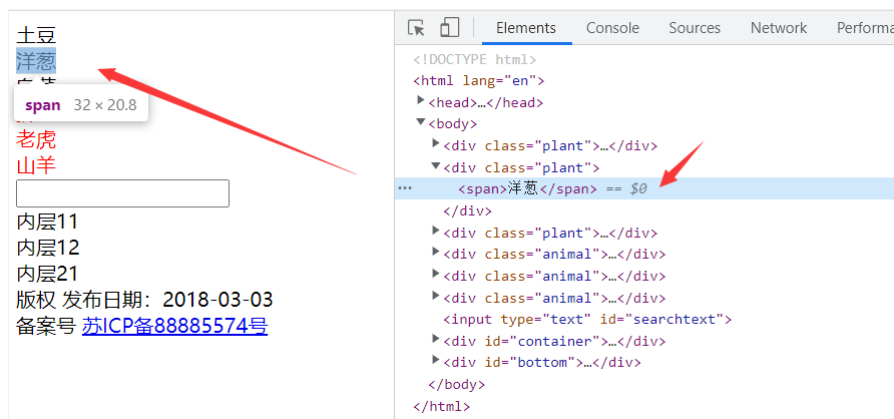
```

SelTest.java x
1 import org.openqa.selenium.WebDriver;
2 import org.openqa.selenium.chrome.ChromeDriver;
3
4 public class SelTest {
5     public static void main(String[] args) throws InterruptedException {
6         // 创建WebDriver对象, 指明使用chrome浏览器驱动
7         System.setProperty("webdriver.chrome.driver", "C:\\Users\\WH\\Desktop\\Temp\\chromedriver.exe");
8         WebDriver driver = new ChromeDriver();
9
10        // 调用WebDriver对象的get方法 可以让浏览器打开指定网址
11        driver.get("http://www.baidu.com");
12
13        //-----
14        // 要执行的操作 (待补充)
15        //-----
16
17        // 设置等待时间, 等待4s
18        Thread.sleep(4000);
19
20        // 关闭页面并结束相关进程
21        driver.quit();
22    }
23 }

```

4. 识别页面元素

用chrome浏览器访问网页, 按F12后, 点击下图箭头处的Elements标签, 即可查看页面对应的HTML 元素



① driver.findElement(By.属性)

查询符合属性要求的第一个元素。

```

driver.get("https://cdn2.byhy.net/files/selenium/sample1.html");

// 根据id属性选择元素
WebElement element1 = driver.findElement(By.id("searchtext"));
element1.sendKeys("111");

// 根据class属性选择元素
WebElement element2 = driver.findElement(By.className("animal"));
System.out.println("元素2: " + element2.getText());

// 根据tag名选择元素
WebElement element3 = driver.findElement(By.tagName("div"));
System.out.println("元素3: " + element3.getText());

// 根据link text (<a href>和</a>之间的内容) 选择元素
WebElement element4 = driver.findElement(By.linkText("苏ICP备88885574号"));

```

```

System.out.println("元素4: " + element4.getAttribute("href"));

// 根据partialLink text (<a href>和</a>之间的部分文本值) 选择元素
WebElement element5 = driver.findElement(By.partialLinkText("88885574"));
System.out.println("元素5: " + element5.getAttribute("href"));

// 通过WebElement对象选择元素
WebElement element6 = driver.findElement(By.id("container"));
WebElement element7 = element6.findElement(By.tagName("div"));
System.out.println("元素7: " + element7.getText());

```

② driver.findElements(By.属性)

查询符合属性要求的所有元素，再使用for循环逐个访问集合中的元素。

```

// driver.findElements(By.属性)
List<WebElement> elements = driver.findElements(By.className("animal"));

for(WebElement element:elements){
    System.out.println("遍历获取元素: " + element.getText());
}

```

查询符合属性要求的所有元素，再使用get(下标)获取指定元素。

```

// driver.findElements(By.属性).get(下标)
WebElement element8 = driver.findElements(By.className("animal")).get(1);
System.out.println("元素8: " + element8.getText());

```

```

WebElement element9 = driver.findElements(By.tagName("div")).get(1);
System.out.println("元素9: " + element9.getText());

```

5. 操控页面元素

`clear()`：清空

`sendKeys("数据")`：输入字符串

`getAttribute("属性名")`：获取元素对应的属性值，如value, id, class, type等

`getText()`：获取文本值，可能获得不到数据

`click()`：点击元素

输入框测试

```

driver.get("https://cdn2.byhy.net/files/selenium/test3.html");

WebElement element = driver.findElement(By.id("input1"));
// 获取输入框的value值，得到输入框中的文本
System.out.println(element.getAttribute("value"));
// 获取输入框的id值

```

```
System.out.println(element.getAttribute("id"));

Thread.sleep(2000);
// 清除输入框已有的字符串
element.clear();

Thread.sleep(2000);
// 输入新字符串
element.sendKeys("111");
```

获取元素文本内容测试

```
driver.get("https://cdn2.byhy.net/files/selenium/sample1.html");

WebElement element = driver.findElements(By.className("plant")).get(1);
// 获取元素对应的HTML文本内容
System.out.println(element.getAttribute("outerHTML"));
System.out.println("-----");
// 获取元素内部的HTML文本内容
System.out.println(element.getAttribute("innerHTML"));
System.out.println("-----");
// 获取元素内部的文本内容
System.out.println(element.getText());
```

按钮点击测试

```
driver.get("https://cdn2.byhy.net/files/selenium/sample2.html");

Thread.sleep(2000);
driver.findElement(By.id("outerbutton")).click();
```

6. 使用XPath选择器识别元素

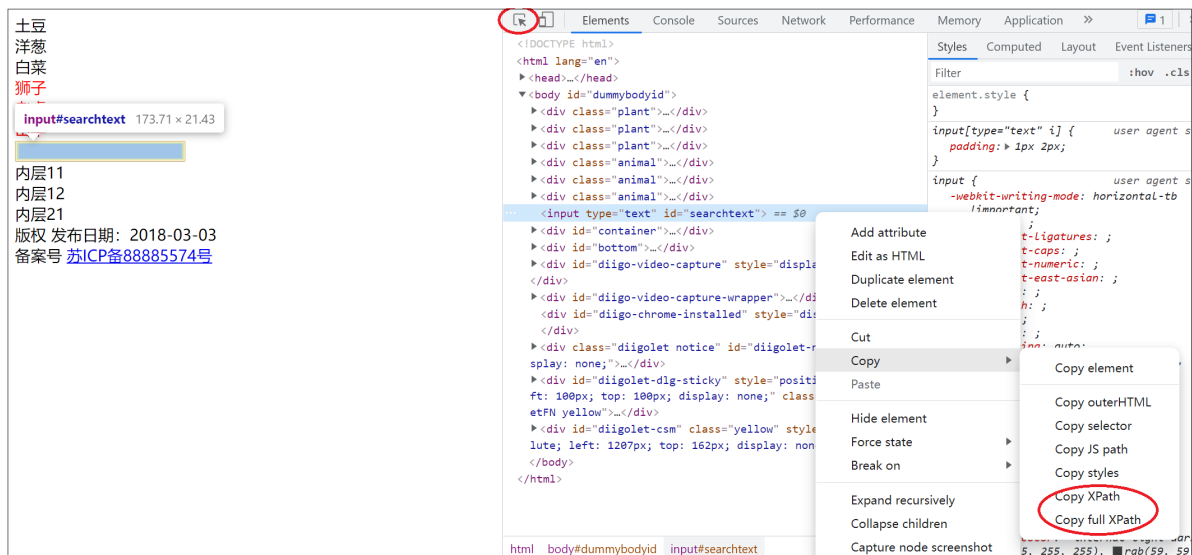
```
driver.get("https://cdn2.byhy.net/files/selenium/sample1.html");

// 以选取网页中的输入框为例

// 1. 使用绝对路径选择元素
WebElement element = driver.findElement(By.xpath("/html/body/input"));

// 2. 使用相对路径选择元素
WebElement element = driver.findElement(By.xpath("//body/input"));

// 3. 根据属性选择, [@属性名='属性值']
// 选择id为"searchtext"的input元素
WebElement element = driver.findElement(By.xpath("//input[@id='searchtext']"));
// 选择id为"searchtext"的元素
WebElement element = driver.findElement(By.xpath("//*[@id='searchtext']"));
```



更便捷的方式：按下F12键调出开发者工具，点击右上角箭头，在网页上选取想要查看的元素。然后在对应的html行右击可获取其XPath的值。

7. 切换frame框架

```
driver.get("https://cdn2.byhy.net/files/selenium/sample2.html");

// 进入id为“frame1”的frame
driver.switchTo().frame("frame1");
WebElement element = driver.findElement(By.className("animal"));
System.out.println(element.getText());

// 切回主文档
driver.switchTo().defaultContent();
driver.findElement(By.id("outerbutton")).click();
```

8. 订票网站测试

1. 打开 <http://8.134.198.4:1080/webtours/> 可访问订票网站，点击**sign up now**注册账号
2. 编写测试代码

```
public class SelTest {
    public static void main(String[] args) throws InterruptedException {
        // 创建WebDriver对象, 指明使用chrome浏览器驱动
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\WH\\Desktop\\Temp\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        // 打开网站
        driver.get("http://1.13.1.202:1080/webtours");

        // 登录
        driver.switchTo().frame(s: "body");
        driver.switchTo().frame(s: "navbar");
        driver.findElement(By.name("username")).sendKeys(...charSequences: "wh");
        driver.findElement(By.name("password")).sendKeys(...charSequences: "1");
        driver.findElement(By.name("login")).click();
        Thread.sleep( millis: 3000);

        // 搜票
        driver.switchTo().defaultContent();
        driver.switchTo().frame(s: "body");
        driver.switchTo().frame(s: "navbar");
        driver.findElement(By.xpath("/html/body/center/center/a[1]/img")).click();
        Thread.sleep( millis: 5000);

        // 选票
        driver.switchTo().defaultContent();
        driver.switchTo().frame(s: "body");
        driver.switchTo().frame(s: "info");
        new Select(driver.findElement(By.name("depart"))).selectByVisibleText("London");
        new Select(driver.findElement(By.name("arrive"))).selectByVisibleText("Paris");
        driver.findElement(By.name("findFlights")).click();
        Thread.sleep( millis: 3000);

        // 选班次
        driver.findElements(By.name("outboundFlight")).get(1).click();
        driver.findElement(By.name("reserveFlights")).click();

        // 确认个人信息
        driver.findElement(By.name("buyFlights")).click();
        Thread.sleep( millis: 3000);

        // 注销
        driver.switchTo().defaultContent();
        driver.switchTo().frame(s: "body");
        driver.switchTo().frame(s: "navbar");
        driver.findElement(By.xpath("/html/body/center/center/a[4]/img")).click();
        Thread.sleep( millis: 3000);

        // 关闭页面并结束相关进程
        driver.quit();
    }
}
```