

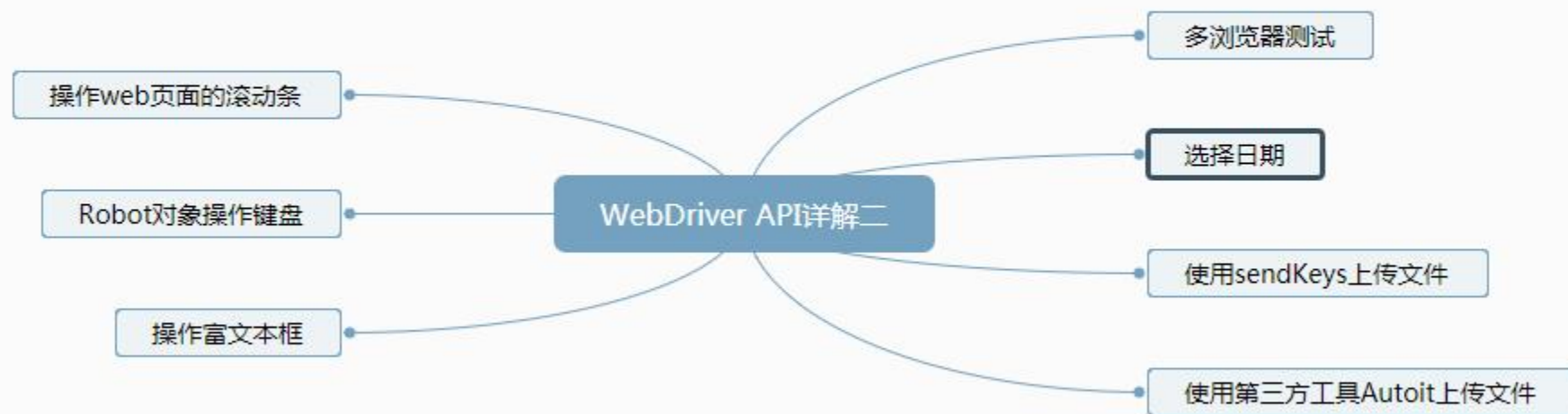
WebDriver API 详解二

讲师：芳姐

金斧子

| 5 金斧子五周年
ANNIVERSARY

科技提升投资品质



一、多浏览进行兼容性测试

■ 1.1.Java代码实现

Java代码:

```
@Parameters("browser")
@BeforeClass
public void beforTest(String Browser) {
    if(Browser.equalsIgnoreCase( "firefox" )) {
        System.setProperty( "webdriver.firefox.bin","D:\\Program Files\\Mozilla
Firefox\\firefox.exe");
        driver = new FirefoxDriver( );
        System.out.println("火狐浏览器实例已经声明");
    }
    else if (Browser.equalsIgnoreCase( "ie" )) {
        System.setProperty( "webdriver.ie.driver","D:\\workspace\\ERP-
KeywordsFramework\\driver\\IEDriverServer.exe" );
        driver = new InternetExplorerDriver( );
        System.out.println("ie浏览器实例已经声明");
    }
    else {
        System.setProperty( "webdriver.chrome.driver","D:\\workspace\\ERP-
KeywordsFramework\\driver\\chromedriver.exe");
        driver = new ChromeDriver( );
```

testng.xml配置文件

#通过testng.xml配置文件进行控制

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Suite2"
parallel="tests" thread-count="2">
    <test name="FirefoxTest">
        <parameter name="browser" value="firefox" />
        <classes>
            <class name="cn.TestScripts.BrowserTest"/>
        </classes>
    </test>
    <test name="ChromeTest">
        <parameter name="browser" value="chrome" />
        <classes>
            <class name="cn.TestScripts.BrowserTest"/>
        </classes>
    </test>
</suite>
```

1.2、Python 代码实现

#谷歌浏览器:

```
def setUp(self):  
    ''' 测试固件setUp的代码，主要用于测试的前期准备工作'''  
    self.driver = webdriver.Chrome()  
    print('谷歌浏览器启动')  
    self.driver.maximize_window()  
    self.driver.implicitly_wait(8)  
  
    self.driver.get("https://qa1-erp.jfz.com")  
  
def tearDown(self):  
    ''' 测试结束之后的清理，一般是关闭浏览器'''  
  
    self.driver.quit()  
  
def test_firefox(self):  
    ''' 这里一定要以test开头'''  
  
    self.driver.find_element_by_xpath("//*[@name='username']").send_keys("defang?")
```

1.2、Python 代码实现

#火狐浏览器:

```
def setUp(self):  
    '''测试固件setUp的代码，主要用于测试的前期准备工作'''  
    self.driver = webdriver.Firefox()  
    print('火狐浏览器启动')  
    self.driver.maximize_window()  
    self.driver.implicitly_wait(8)  
  
    self.driver.get("https://qa1-erp.jfz.com")
```

#用例路径

```
case_path = os.path.join(os.getcwd(), "testcase")  
def all_case():  
    discover =  
    unittest.defaultTestLoader.discover(case_path, pattern="test*.py", top_level_dir=None)  
  
    print("discover")  
  
    return discover  
  
if __name__ == '__main__':
```

二、选择日期

■ 2.1 Java代码实现

```
// 创建一个dataformat对象
SimpleDateFormat dataFormat = new SimpleDateFormat(pattern: "yyyy-MM-dd");

// 利用Date()获取当前时间
Date date = new Date();
// 格式化时间,并用String对象存储
String date1 = dataFormat.format(date);
Calendar ca = Calendar.getInstance();
ca.add(Calendar.DATE, amount: 10); // 当前日期加10天
date = ca.getTime();
String date2 = dataFormat.format(date);
System.out.println(date1);
System.out.println(date2);

Thread.sleep(millis: 2000);
driver.findElement(By.xpath("//*[@id='validityFrom']")).sendKeys(date1);
driver.findElement(By.xpath("//*[@id='validityTo']")).sendKeys(date2);

driver.findElement(By.xpath("//*[@name='InsRatioDate']")).sendKeys(date2);
```

2.2 Python代码实现

```
agreementType.find_element_by_xpath("//option[@value='1']").click()  
#输入协议有效时间  
driver.find_element_by_xpath("//*[@id='validityFrom']").send_keys("2018-01-12")  
  
driver.find_element_by_xpath("//*[@id='validityTo']").send_keys("2018-01-20")
```


三.上传文件

■ 3.1 使用 sendKeys 方法 上传 一个 文件 附件

■ 3.1.1 Java代码实现

- //查找页面上name名称为"file"的文件上传框
- WebElement ele =
driver.findElement(By.xpath("//input[@name='file']"));

- //在文件上传框的路径框里面输入要上传的文件路径"D:\\图片\\协议主体.jpg"
- ele.sendKeys("D:\\图片\\协议主体.jpg");

■ 3.1.2 Python代码实现:

- driver.find_element_by_xpath("//input[@name='file']").send_keys("D:\\图片\\协议主体.jpg")
- assert "下载" in driver.page_source

```
height: #0px; overflow: hidden; bottom: auto; right: auto;"}<input class="webuploader-element-invisible" name="file" multiple="multiple" type="file"><label style="opacity: 0; width: 100%; height: 100%; display: block; cursor: pointer; background: rgb(255, 255, 255) none repeat scroll 0% 0%;"></label></div></div>
```

3.2使用第三方工具Autoit工具上传

3.2.1 操作步骤见链接：

<http://note.youdao.com/noteshare?id=5465df2c248668c5af3f743e965b8e7d&sub=36D1027AFD664ACC9E0CB0776DFF8B6B>

java代码：

3.2.2 //先定位到选择文件的元素

```
WebElement ele = driver.findElement(By.tagName("label"));  
ele.click();  
//模拟操作打开选择文件框  
Runtime.getRuntime().exec("D:\\workspace\\Erp-Automated-Test\\upload.exe" + " "  
+ "chrome" + " " + "D:\\图片\\合同签字页.jpg");
```

Python代码：

```
time.sleep(5)  
driver.find_element_by_tag_name("label").click()  
os.system("D:\\workspace\\Erp-Automated-Test\\upload.exe" + " "  
+ "chrome" + " " + "D:\\图片\\合同签字页.jpg")
```

四.操作web页面的滚动条

#滚动条

Java代码:

#滚动页面的滚动条到页面最下面

```
((JavascriptExecutor)  
driver).executeScript( "window.scrollTo(0,document.body.scrollHeight)" );
```

#滚动页面的滚动条到页面的某个元素

```
WebElement element = driver.findElement( By.xpath( "//*[text()='下载']" ) );  
  
//使用JavaScript的scrollIntoView()函数将滚动条滚动到页面的指定元素位置  
((JavascriptExecutor)  
driver).executeScript( "arguments[0].scrollIntoView();",element );
```

```
//使用JavaScript的scrollto函数,使用0和800的横纵坐标参数,将页面的滚动条纵向下滑800个像素  
((JavascriptExecutor) driver).executeScript( "window.scrollTo(0,800)" );
```

四.操作web页面的滚动条

#滚动条

Python代码:

#滚动页面的滚动条到页面最下面

#进行滚动，滚动到最下方进行操作

```
driver.execute_script("window.scrollTo(100, document.body.scrollHeight);")
```

#使用JavaScript的scrollIntoView函数将被遮挡的元素滚动到可见屏幕

#scrollIntoView(true)表示将元素滚动屏幕中间

```
driver.execute_script("document.getElementById('editOpenDay').scrollIntoView(true);")
```

#使用JavaScript的scrollby方法，使用0和800横纵坐标参数

```
driver.execute_script("window.scrollBy(0, 800);")
```

五.RoBot对象操作 键盘

Java代码:

```
WebDriverWait wait = new WebDriverWait( driver,10 );
//使用显示等待，判断页面是否显示搜索框
wait.until( ExpectedConditions.presenceOfElementLocated( By.name( "username" ) ) );
//调用封装好的函数setAndctrlVClipboardData，将“D20171228001”产品编码使用ctrl+v组合键方式粘贴
到搜索框中
setAndctrlVClipboardData("defang3");

/*
封装的粘贴函数，可以将函数的string参数值放入到剪贴板中，然后再使用Robot对象的KeyPress和
keyRelease函数来模拟ctrl+v组合键完成粘贴操作
*/
public void setAndctrlVClipboardData(String string){
//声明StringSelection对象，并使用函数的string参数完成实例化
StringSelection stringSelection = new StringSelection( string );
//使用Toolkit对象的setContents方法将字符串放到剪切板中
Toolkit.getDefaultToolkit().getSystemClipboard().setContents( stringSelection,null );
//声明Robot对象
Robot robot = null;
try {
//生成Robot的对象实例
robot = new Robot();
} catch (AWTException el) {
el.printStackTrace();
}
//调用keyPress方法来实现按下Ctrl键
robot.keyPress( KeyEvent.VK_CONTROL);
//调用keyPress方法来实现按下V键
robot.keyPress(KeyEvent.VK_V);
//调用keyRelease方法来实现释放V键
robot.keyRelease(KeyEvent.VK_V);
```

封装tab建和回车键

```
public void pressTabKey() {  
    Robot robot = null;  
    try {  
        robot = new Robot();  
  
    } catch (AWTException e) {  
        e.printStackTrace();  
    }  
    //调用keyPress方法来实现按下tab键  
    robot.keyPress(KeyEvent.VK_TAB);  
    //调用keyRelease方法来实现释放Tab键  
    robot.keyRelease(KeyEvent.VK_TAB);  
}  
  
public void pressEnterKey() {  
    Robot robot = null;  
    try {  
  
        robot = new Robot();  
    } catch (AWTException e) {  
        e.printStackTrace();  
    }  
    //调用keyPress方法来实现按下Enter键  
    robot.keyPress(KeyEvent.VK_ENTER);  
}
```

Python代码：

```
def getText():
    # 打开剪贴板
    w.OpenClipboard()
    # 获取剪贴板中的数据
    d = w.GetClipboardData(win32con.CF_TEXT)
    # 关闭剪贴板
    w.CloseClipboard()
    # 返回剪贴板数据给调用者
    return d

# 设置剪贴板内容

def setText(aString):
    # 打开剪贴板
    w.OpenClipboard()
    # 清空剪贴板
    w.EmptyClipboard()
    # 将数据astring写入剪贴板
    w.SetClipboardData(win32con.CF_UNICODETEXT, aString)
    # 关闭剪贴板
    w.CloseClipboard()
```

```
#定义即将要设置到剪贴板中的内容
content= 'defang3'
#将content变量中的内容设置到剪贴板中
setText(content)
#从剪贴板中获取刚设置到剪贴板中的内容
getContent = getText()
print(getContent.decode("gbk").encode("utf-8"))
#将焦点切换到用户名输入框中
driver.find_element_by_xpath("//*[@name='username']").click()

#模拟ctrl+v组合键，将从剪贴板中获取到的内容粘贴到用户名输入框中
ActionChains(driver).key_down(Keys.CONTROL).send_keys('v').key_up(Keys.CONTR
OL).perform()
```


五.操作富文本框

#Java代码

```
Thread.sleep( 5000 );
String body_string = "测试打款心得信息";
driver.switchTo().frame("ueditor_0");
driver.findElement( By.tagName( "body" ) ).clear();
driver.findElement( By.tagName( "body" ) ).sendKeys( body_string );
driver.switchTo().parentFrame();

Thread.sleep( 5000 );
driver.findElement( By.className( "l-dialog-btn-inner" ) ).click();
```

Python代码:

#点击编辑按钮

```
driver.find_element_by_xpath("//*[text()=' 编辑']").click()
time.sleep(5)
driver.switch_to.frame("ueditor_0")
driver.find_element_by_tag_name("body").clear()
driver.find_element_by_tag_name("body").send_keys("测试打款心得-Python")
driver.switch_to.parent_frame()
time.sleep(5)
```

作业

- 1、Java代码实现多浏览器操作
- 2、Java和Python代码实现输入日期
- 3、使用sendkeys上传文件和第三方工具上传文件
- 4、操作页面的滚动条
- 5、操作富文本框内容

Thanks!

科技提升投资品质

