

软件测试上机报告



第二次上机作业

学 院__智能与计算学部__
专 业__软件工程__
姓 名__陈姝宇__
学 号__3017218119__
年 级__2017 级__
班 级__3 班__

1. Experimental Requirements

Tasks:

1. Install Selenium with Eclipse.
2. Install Firefox and SeleniumIDE plugin.
3. Try to record and export scripts using SeleniumIDE.
4. Please complete the following task using Selenium Webdriver:
“Selenium Lab.xlsx” contains information about the students, and <http://103.120.226.190/selenium-demo/git-repo> can view someone's information after logging in (student id as username, git address as password). Please check each record in the excel to make sure that each student's information is consistent with the information on the website.

Requirements for the experiment:

1. Finish the tasks above individually.
2. Check in your java code to github or gitee
3. Please send your experiment report to 智慧树, the following information should be included in your report:
 - a) The brief description that you install Selenium, Firefox and SeleniumIDE.
 - b) Steps for recording and exporting scripts.
 - c) Steps for testing the website using Selenium.

2. Environmental configuration

The brief description that I install Selenium in IntelliJ IDEA:

- 1) Create maven project : ex2, add selenium dependency in pom.xml

```
<!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java -->
<dependencies>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>3.141.59</version>
  </dependency>
</dependencies>
```

- 2) Auto-import :

Resolving dependencies of ex2...

- 3) After import successfully, the jar package imported through Maven will be displayed in [Maven projects] on the right side of the idea:

The screenshot shows the 'Maven Projects' tool window in IntelliJ IDEA. The 'Dependencies' section is expanded, displaying a list of resolved dependencies for the 'org.seleniumhq.selenium:selenium-java:3.141.59' artifact. The dependencies include selenium-api, selenium-chrome-driver, selenium-edge-driver, selenium-firefox-driver, selenium-ie-driver, selenium-opera-driver, selenium-remote-driver, selenium-safari-driver, selenium-support, net.bytebuddy:byte-buddy, org.apache.commons:commons-exec, com.google.guava:guava, and com.squareup.okhttp3:okhttp3. The 'Ant Build' button is visible on the right side of the tool window.

- org.seleniumhq.selenium:selenium-java:3.141.59
 - org.seleniumhq.selenium:selenium-api:3.141.59
 - org.seleniumhq.selenium:selenium-chrome-driver:3.141.59
 - org.seleniumhq.selenium:selenium-edge-driver:3.141.59
 - org.seleniumhq.selenium:selenium-firefox-driver:3.141.59
 - org.seleniumhq.selenium:selenium-ie-driver:3.141.59
 - org.seleniumhq.selenium:selenium-opera-driver:3.141.59
 - org.seleniumhq.selenium:selenium-remote-driver:3.141.59
 - org.seleniumhq.selenium:selenium-safari-driver:3.141.59
 - org.seleniumhq.selenium:selenium-support:3.141.59
 - net.bytebuddy:byte-buddy:1.8.15
 - org.apache.commons:commons-exec:1.3
 - com.google.guava:guava:25.0-jre
 - com.squareup.okhttp3:okhttp3:3.11.0
 - com.squareup.okio:okio:1.14.0

```
13      <dependency>
14          <groupId>org.seleniumhq.selenium</groupId>
15          <artifactId>selenium-java</artifactId>
16          <version>3.141.59</version>
17      </dependency>
```

The brief description that I install Firefox and SeleniumIDE:

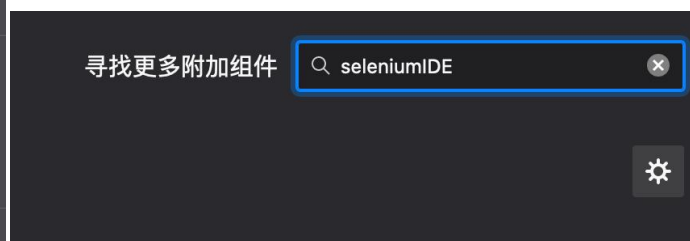
1) Visit the official website: <http://www.firefox.com.cn/download/>, and click 'download'



2) Click 'Firefox-latest.dmg' and move Firefox into applications, then open Firefox



3) Find component: seleniumIDE





Selenium IDE

作者: [Selenium](#)

Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests.

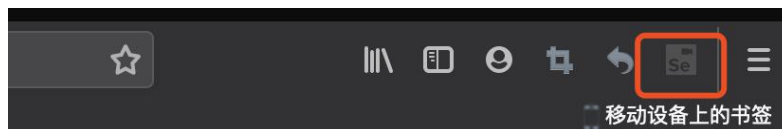
+ 添加到 Firefox

⚠ 这不是一个获推荐的扩展。安装前请确认您信任该扩展。 [详细了解](#)

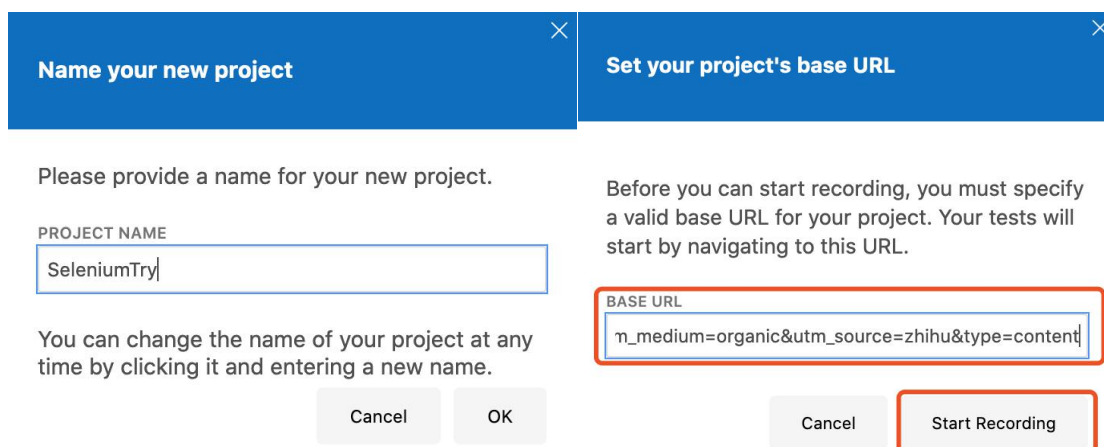
3. Recording and exporting scripts

Steps for recording and exporting scripts.

1) The icon of seleniumIDE at the right and top of firefox windows.



2) Click the seleniumIDE icon and choose 'Record a new test in a new project':



3) Create Project and add command

The screenshot shows the Selenium IDE interface. At the top, the title bar reads '扩展: (Selenium IDE) - Selenium IDE - SeleniumTry*'. Below it, the project name 'Project: SeleniumTry*' is displayed. The main area is divided into two panes. The left pane, labeled 'Executing', shows a list of commands for the 'SeleniumTry*' project. The right pane shows the details of the selected command, which is 'open'. The URL bar shows 'https://www.zhihu.com'. Below the URL bar, there is a table with columns 'Command', 'Target', and 'Value'. The first row shows 'open' with a target of '/search?q=%E6%AD%8C%E6%89%8B...' and a value of 'linkText=歌手·当打之年'. The second row shows 'click' with a target of 'linkText=歌手·当打之年' and a value of 'linkText=歌手·当打之年'. At the bottom, there is a progress bar and a status bar showing 'Runs: 1 Failures: 0'.

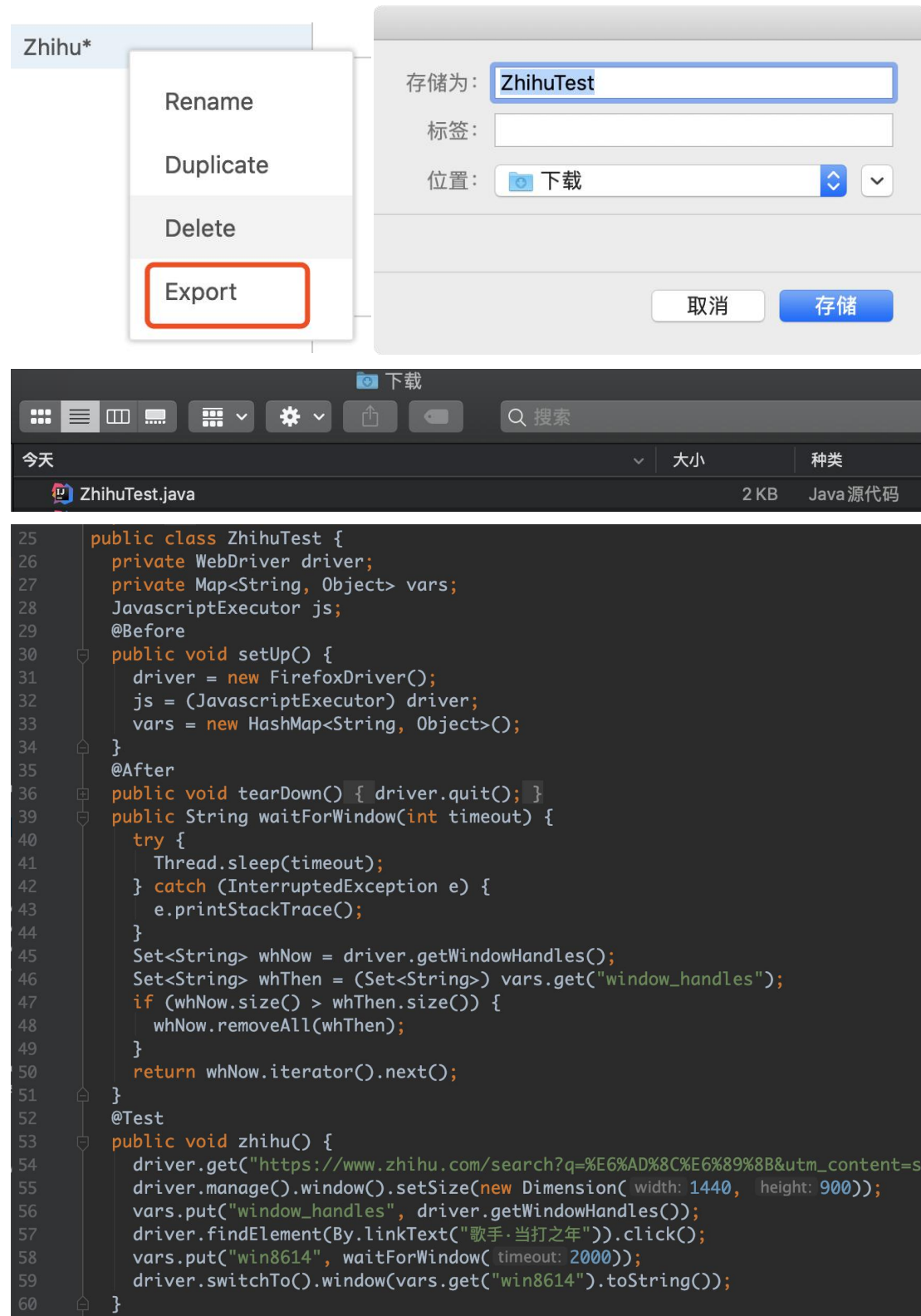
Command	Target	Value
1 open	/search?q=%E6%AD%8C%E6%89%8B...	linkText=歌手·当打之年
2 click	linkText=歌手·当打之年	linkText=歌手·当打之年

4) Click run tests and see log. At first, the test if failed because of popupwindow stoped by firefox. After I set the pop-up operation to allow, and then rerun the test case, it is successful.

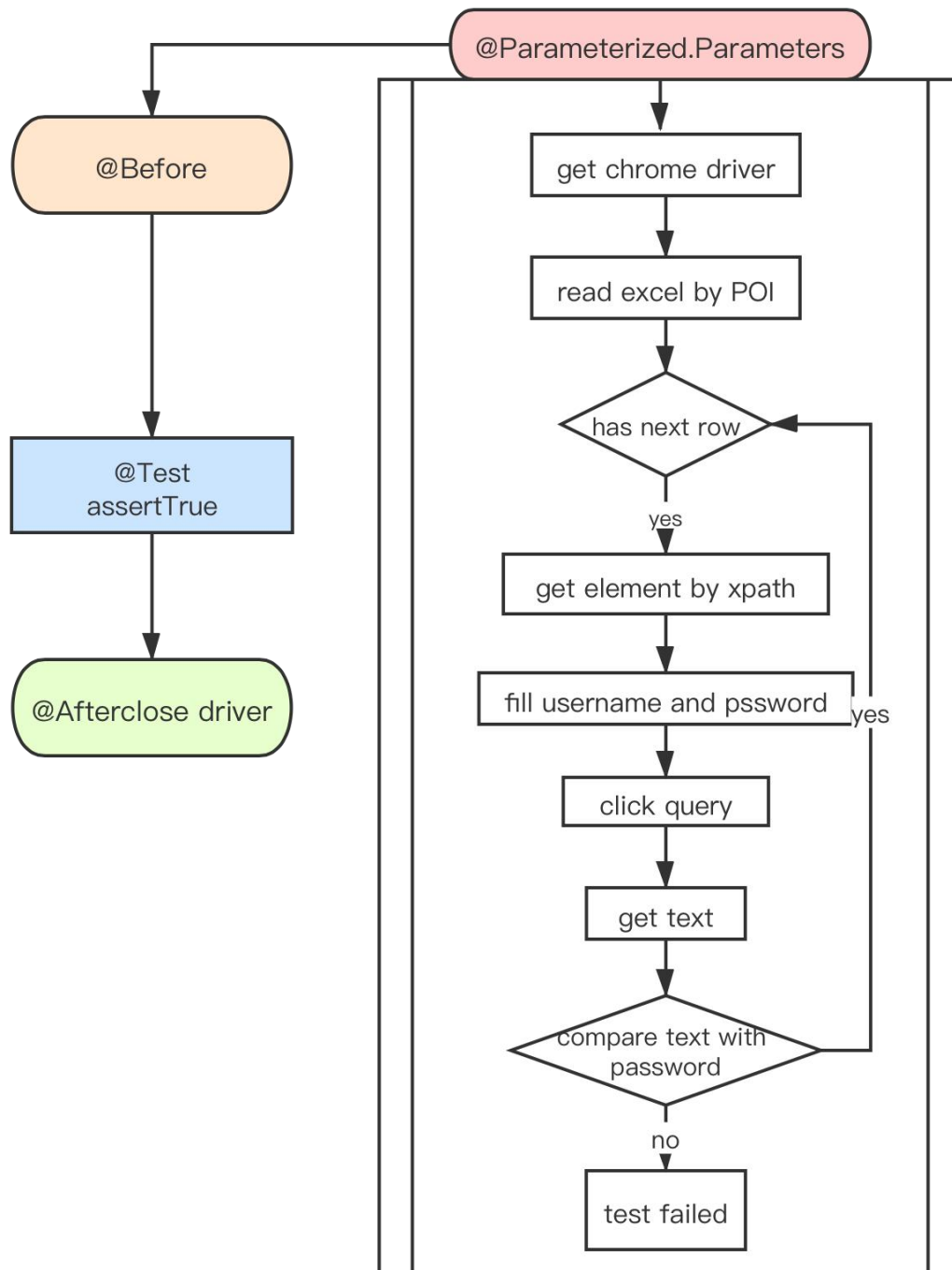
The screenshot shows the Selenium IDE interface with the test log displayed. The top bar shows 'Executing' and a 'Run all tests' button. The log shows the following steps:

- Running 'SeleniumTry' 20:41:09
- 1. open on /search?q=%E6%AD%8C%E6%89%8B&utm_content=search_hot&utm_medium=organic&utm_source=zhihu&type=content OK 20:41:10
- 2. click on linkText=歌手·当打之年 Failed: 20:41:11
Exceeded waiting time for new window to appear 2000ms
- 'SeleniumTry' ended with 1 error(s) 20:41:16
- Running 'SeleniumTry' 20:41:44
- 1. open on /search?q=%E6%AD%8C%E6%89%8B&utm_content=search_hot&utm_medium=organic&utm_source=zhihu&type=content OK 20:41:46
- 2. click on linkText=歌手·当打之年 OK 20:41:47
- 'SeleniumTry' completed successfully 20:41:50

5) Right click Zhihu* and Choose Export, resulting in exporting script. And in Download document, there is a ZhihuTest.java. Open ZhihuTest.java.



4. testing the website using Selenium



5. Source Code

```
@RunWith(Parameterized.class)
public class Ex2 {

    private static WebDriver driver;
    private String expected;
    private String actual;

    public Ex2(String expected, String actual) {
        this.expected = expected;
        this.actual = actual;
    }

    @Before
    public void setUp() {
    }

    @Parameterized.Parameters
    public static Collection<Object[]> getData() {

        // create chromeDriver
        String driverPath = System.getProperty("user.dir") +
"/src/main/resources/driver/chromedriver";

        System.setProperty("webdriver.chrome.driver", driverPath);
        driver = new ChromeDriver();

        // baseUrl
        driver.get("http://103.120.226.190/selenium-demo/git-repo");

        ArrayList<Object[]> datas = new ArrayList<Object[]>();

        String filePath = System.getProperty("user.dir") +
```

```

"/src/main/resources/info/Selenium+Lab2020.xlsx";

File xlsFile = new File(filePath);
Workbook workbook;

try {
    workbook = WorkbookFactory.create(xlsFile);
    // first sheet
    Sheet sheet = workbook.getSheetAt(0);
    int row = 0;
    while (true) { // view every row
        Row r = sheet.getRow(row); // get this row
        String c1 = r.getCell(1).getStringCellValue(); // get username
        if (c1.equals("")) break; // if this row is null, then break out
        String c2 = r.getCell(2).getStringCellValue(); // get password

        // username
        driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/div/form/div[1]/input")).sendKeys(c1);

        // password
        driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/div/form/div[2]/input")).sendKeys(c2);

        // click query
        driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/div/form/div[3]/input")).click();

        // query text
        String text =
        driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/div/form/div[5]/code")).getText();

        datas.add(new Object[]{text, c2});
        row++; // next row
    }
}

```

```

    } catch (IOException e) {
        e.printStackTrace();
    }

    return datas;
}

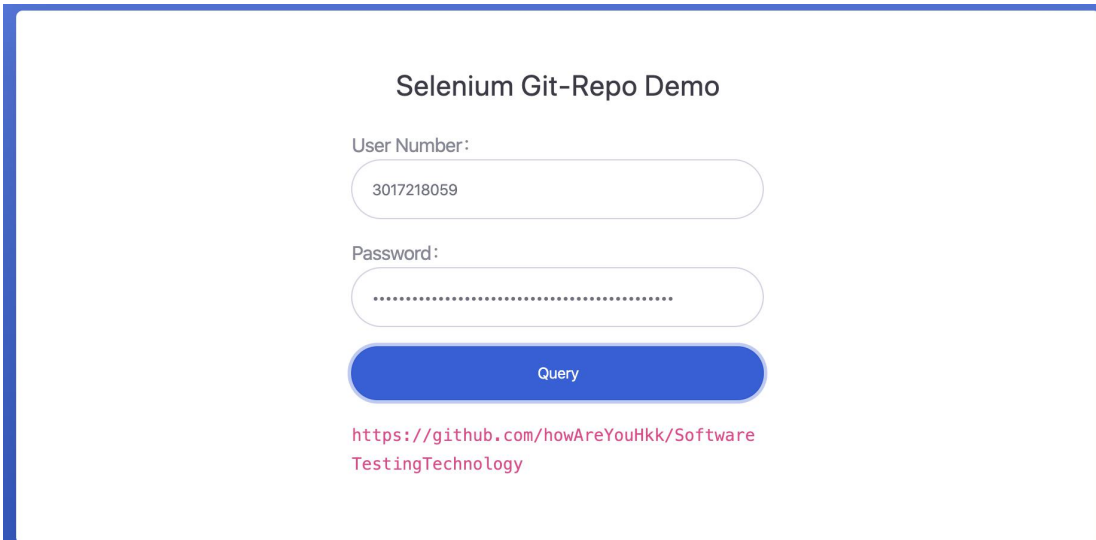
@After
public void tearDown() {
    driver.quit(); // close driver
}

@Test
public void seleniumTest() {
    Assert.assertEquals(this.expected, this.actual);
}
}

```

6. Operation Result

During test time, chrome look like below fiure:



The screenshot shows a web application titled "Selenium Git-Repo Demo". It features a login form with two input fields: "User Number:" containing the text "3017218059" and "Password:" containing a series of dots. Below these fields is a blue button labeled "Query". At the bottom of the form, there is a red text link: <https://github.com/howAreYouHkk/SoftwareTestingTechnology>.

After test finished, 'Tests passed' is exhibited on console. So each student's information is consistent with the information on the website.

