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___王兆盟__________

学　　号___3017218072______

年　　级___ 2017 级_　　　___

班　　级___软件工程一班____

1、Experiment Requirement

Tasks:

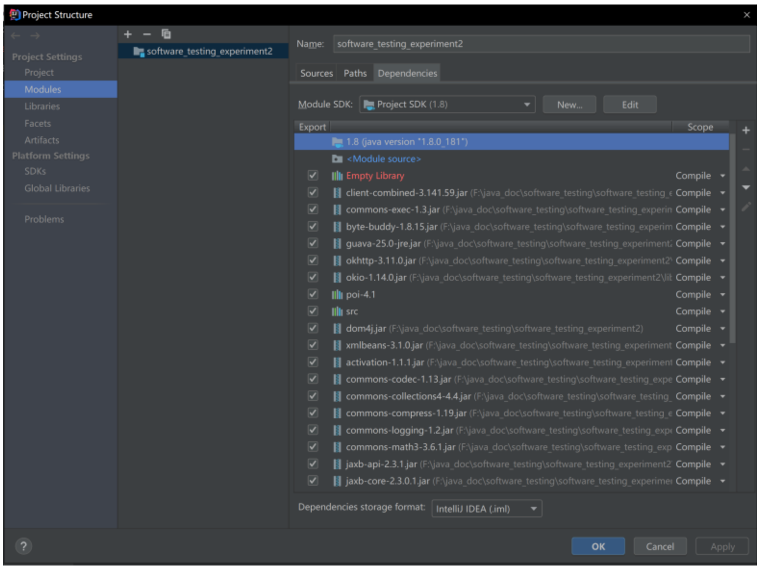
1. Install Selenium with Eclipse.
2. Install Firefox and SeleniumIDE plugin.
3. Try to record and export scripts using SeleniumIDE.
4. Please complete the following task using Selenium Webdriver: “Selenium Lab.xlsx” contains information about the students, and http://103.120.226.190/selenium-demo/git-repo 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、Source Code And Experiment Process

Firstly, export all the packages of Selenium into the project



software_testing_experiment2

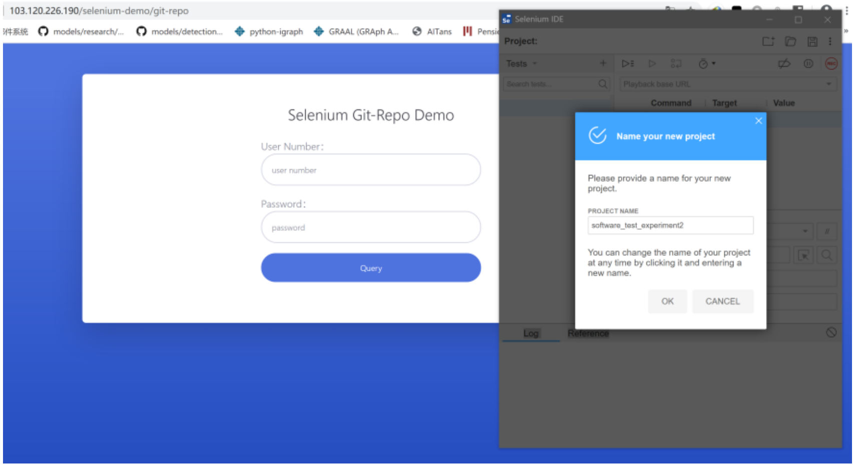
Name: software_testing_experiment2

Module SDK: Project SDK (1.8)

Export		Scope
	1.8 (java version "1.8.0_181")	
	<Module source>	
☑	Empty Library	Compile
☑	client-combined-3.141.59.jar (F:\java_doc\software_testing\software_testing_ε	Compile
☑	commons-exec-1.3.jar (F:\java_doc\software_testing\software_testing_experin	Compile
☑	byte-buddy-1.8.15.jar (F:\java_doc\software_testing\software_testing_experim	Compile
☑	guava-25.0-jre.jar (F:\java_doc\software_testing\software_testing_experiment:	Compile
☑	okhttp-3.11.0.jar (F:\java_doc\software_testing\software_testing_experiment2\	Compile
☑	okio-1.14.0.jar (F:\java_doc\software_testing\software_testing_experiment2\lit	Compile
☑	poi-4.1	Compile
☑	src	Compile
☑	dom4j.jar (F:\java_doc\software_testing\software_testing_experiment2)	Compile
☑	xmlbeans-3.1.0.jar (F:\java_doc\software_testing\software_testing_experiment	Compile
☑	activation-1.1.1.jar (F:\java_doc\software_testing\software_testing_experiment	Compile
☑	commons-codec-1.13.jar (F:\java_doc\software_testing\software_testing_expe	Compile
☑	commons-collections4-4.4.jar (F:\java_doc\software_testing\software_testing_	Compile
☑	commons-compress-1.19.jar (F:\java_doc\software_testing\software_testing_ε	Compile
☑	commons-logging-1.2.jar (F:\java_doc\software_testing\software_testing_exp	Compile
☑	commons-math3-3.6.1.jar (F:\java_doc\software_testing\software_testing_exp	Compile
☑	jaxb-api-2.3.1.jar (F:\java_doc\software_testing\software_testing_experiment2	Compile
☑	jaxb-core-2.3.0.1.jar (F:\java_doc\software_testing\software_testing_experimei	Compile

Dependencies storage format: IntelliJ IDEA (.iml)

Then I downloaded the SeleniumIDE for chrome, and export it as an extension. I run SeleniumIDE, created a new project.



Selenium Git-Repo Demo

User Number: user number

Password: password

Query

Name your new project

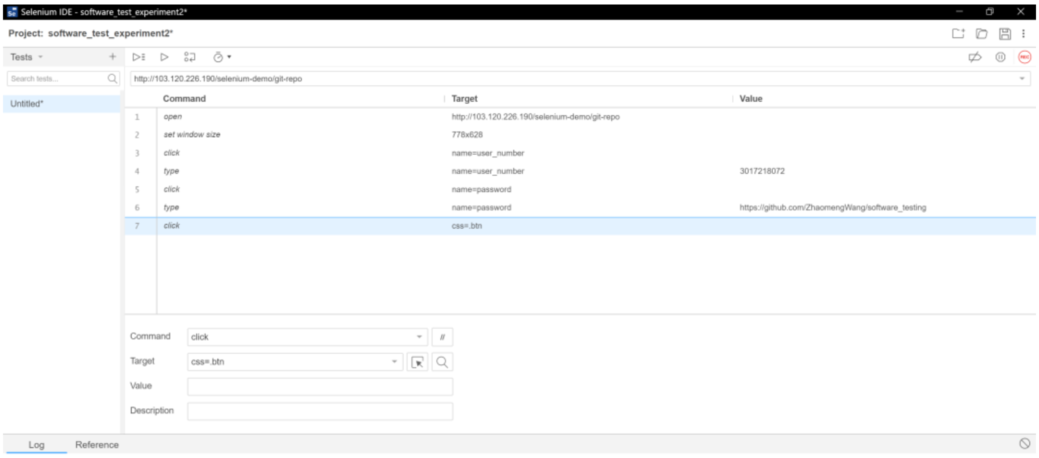
Please provide a name for your new project.

PROJECT NAME: software_test_experiment2

You can change the name of your project at any time by clicking it and entering a new name.

OK CANCEL

And recorded the whole process of clicking and typing on the website.



Project: software_test_experiment2*

http://103.120.226.190/selenium-demo/git-repo

	Command	Target	Value
1	open	http://103.120.226.190/selenium-demo/git-repo	
2	set window size	778x628	
3	click	name=user_number	
4	type	name=user_number	3017218072
5	click	name=password	
6	type	name=password	https://github.com/ZhaomengWang/software_testing
7	click	css=.btn	

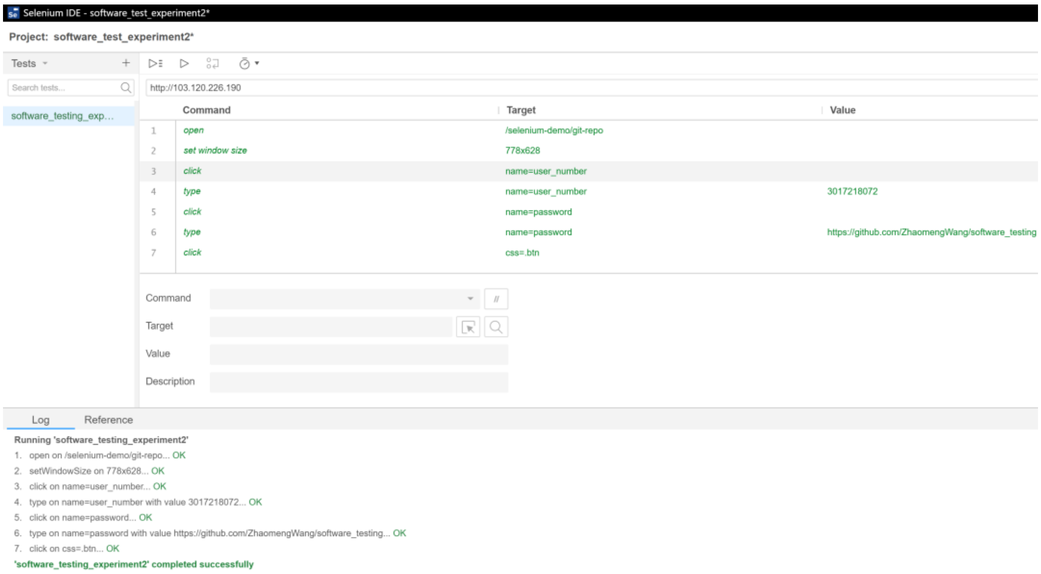
Command: click

Target: css=.btn

Value:

Description:

Then I run the recorded test case. In the end, I clicked the related button, chose the Java Unit as the language to export the script.



Project: software_test_experiment2*

http://103.120.226.190

	Command	Target	Value
1	open	/selenium-demo/git-repo	
2	set window size	778x628	
3	click	name=user_number	
4	type	name=user_number	3017218072
5	click	name=password	
6	type	name=password	https://github.com/ZhaomengWang/software_testing
7	click	css=.btn	

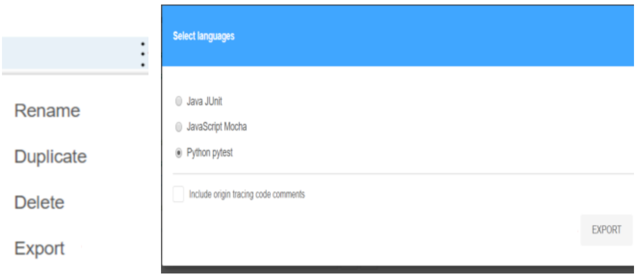
Command:

Target:

Value:

Description:

```
Running 'software_testing_experiment2'
1. open on /selenium-demo/git-repo... OK
2. setWindowSize on 778x628... OK
3. click on name=user_number... OK
4. type on name=user_number with value 3017218072... OK
5. click on name=password... OK
6. type on name=password with value https://github.com/ZhaomengWang/software_testing... OK
7. click on css=.btn... OK
'software_testing_experiment2' completed successfully
```



Select languages

○ Java JUnit
○ JavaScript Mocha
● Python pytest

□ Include origin tracing code comments

EXPORT

Next, in order to accomplish the last task, I wrote two Java class which are POIExcel.java and SelenuiumWebTest.java. The first was designed to read the Excel xlsx file by row and generated an ArrayList of String which contain each row of the file. The source code and annotations are shown below.

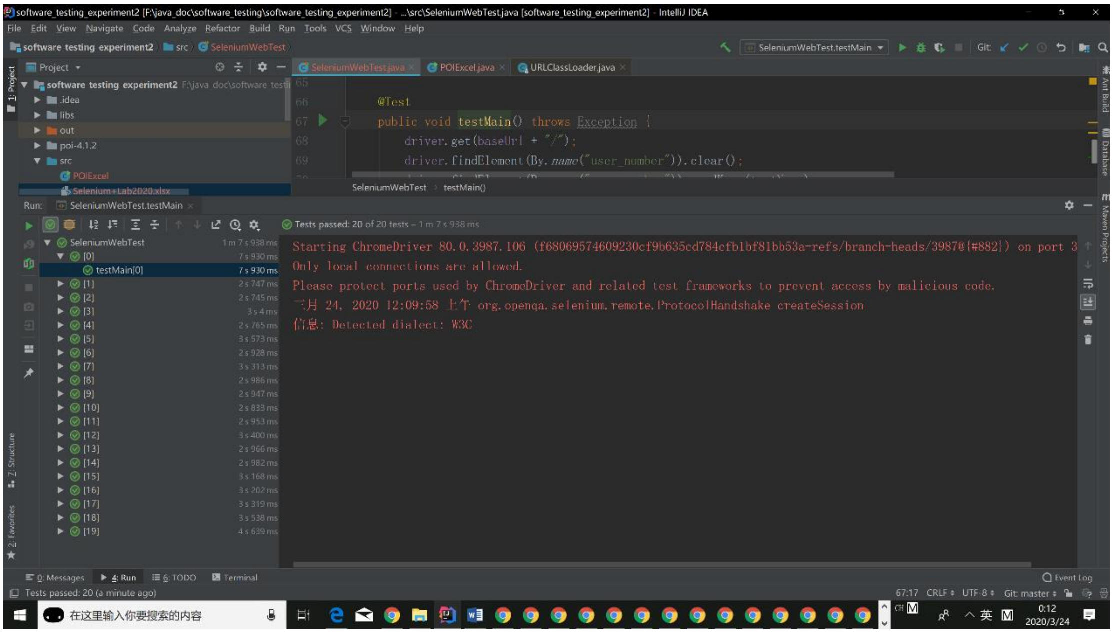
```
public class POIExcel {
    public static ArrayList<ArrayList<String>> readExcelRow() throws IOException {
        Workbook workbook = null;
        File excelFile = new
File(”F:\\java_doc\\software_testing\\software_testing_experiment2\\src\\Selenium+L
ab2020.xlsx”);
        InputStream is = new FileInputStream(excelFile);
        workbook = new XSSFWorkbook(is);
        ArrayList<ArrayList<String>> als = new ArrayList<ArrayList<String>>();
        for (int numSheet = 0; numSheet < workbook.getNumberOfSheets(); numSheet++)
{
            Sheet sheet = workbook.getSheetAt(numSheet);
            //当前 sheet 页面为空,继续遍历
            if (sheet == null) {
                continue;
            }
            // 对于每个 sheet，读取其中的每一行
            for (int rowNum = 0; rowNum <= sheet.getLastRowNum(); rowNum++) {
                Row row = sheet.getRow(rowNum);
                if (row == null) {
                    continue;
                }
                // 遍历每一行的每一列
                ArrayList<String> al = new ArrayList<String>();
                for (int columnNum = 1; columnNum < row.getLastCellNum();
columnNum++) {
                    Cell cell = row.getCell(columnNum);
                    al.add(cell.getRichStringCellValue().getString());
                }
                als.add(al);
            }
        }
        is.close();
        return als;
    }
```

The second class was designed to do the test by using Selenium. I use Junit as a tool to do the unit test. I annotate test class with @RunWith(Parameterized.class) and Create a public static method annotated with @Parameters that cooperate with the POIExcel which read the Excel file aims to return a collection of objects as test data set. Each test case is a tuple which contains a student number, a password also known as the github URL and the github URL as an expected value. In the test function, I used Selenuim to find the HTML elements and mimic the clicking, through name, cssSelector and xpath. The source code and annotations are shown below.

```
@RunWith(Parameterized.class)
public class SeleniumWebTest
{
    private WebDriver driver;
    private String baseUrl;

    private String testName;
    private String testPwd;
    private String gitHubUrl;

    public SeleniumWebTest(String testName, String testPwd, String githubUrl) {
        this.testName = testName;
        this.testPwd = testPwd;
        this.gitHubUrl = githubUrl;
    }

    @Before
    public void setUp() throws Exception {

System.setProperty("webdriver.chrome.driver","F:\\java_doc\\software_testing\\software_testi
ng_experiment2\\chromedriver.exe");
        driver =new ChromeDriver();
        baseUrl = "http://103.120.226.190/selenium-demo/git-repo";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Parameterized.Parameters
    public static Collection<Object[]> getData() {
        ArrayList<ArrayList<String>> excelData = new ArrayList<ArrayList<String>>();
        try {
            excelData = POIExcel.readExcelRow();
        } catch (IOException e) {
            e.printStackTrace();
        }

        Object[][] obj = new Object[20][];

        for (int i = 0; i < excelData.size() ; i++)
        {
            String stuNum = excelData.get(i).get(0);
            String pwd = excelData.get(i).get(1);
            String githubUrl = excelData.get(i).get(1);
            if (stuNum.equals(""))
                break;
            else
                obj[i] = new Object[]{stuNum, pwd, githubUrl};

        }
        return Arrays.asList(obj);
    }
    @Test
    public void testMain() throws Exception {
        driver.get(baseUrl + "/");
        driver.findElement(By.name("user_number")).clear();
        driver.findElement(By.name("user_number")).sendKeys(testName);
        driver.findElement(By.name("password")).clear();
        driver.findElement(By.name("password")).sendKeys(testPwd);
        driver.findElement(By.cssSelector("[type='submit']")).click();
        assertEquals(this.gitHubUrl,
driver.findElement(By.xpath("/html/body/div/div/div/div/div/div/div[2]/div/form/div[5]/code"
)).getText());
    }



}
```

3、 Experiment Result

I run the test to verify if the website show up the expect github URL when we input the right username and password. I run 20 test cases in total and the results proved that in each case the test value equals the expect value which means each student’s information is consistent with the information on the website.



software_testing_experiment2 [F:\java_doc\software_testing\software_testing_experiment2] - ...\src\SeleniumWebTest.java [software_testing_experiment2] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
software testing experiment2 src SeleniumWebTest
SeleniumWebTest.testMain
SeleniumWebTest.java POIExcel.java URLClassLoader.java
Project
software testing experiment2 F:\java doc\software testi
.idea
libs
out
poi-4.1.2
src
POIExcel
Selenium+Lab2020.xlsx
SeleniumWebTest › testMain()
Run: SeleniumWebTest.testMain
0: Messages
4: Run
6: TODO
Terminal
Event Log
67:17 CRLF UTF-8 Git: master
在这里输入你要搜索的内容
0:12
2020/3/24

```
    @Test
    public void testMain() throws Exception {
        driver.get(baseUrl + "/");
        driver.findElement(By.name("user_number")).clear();
```

```
Tests passed: 20 of 20 tests – 1 m 7 s 938 ms

SeleniumWebTest      1 m 7 s 938 ms
  [0]                      7 s 930 ms
    testMain[0]            7 s 930 ms
  [1]                      2 s 747 ms
  [2]                      2 s 745 ms
  [3]                        3 s 4 ms
  [4]                      2 s 765 ms
  [5]                      3 s 573 ms
  [6]                      2 s 928 ms
  [7]                      3 s 313 ms
  [8]                      2 s 986 ms
  [9]                      2 s 947 ms
  [10]                     2 s 833 ms
  [11]                     2 s 953 ms
  [12]                     3 s 400 ms
  [13]                     2 s 966 ms
  [14]                     2 s 982 ms
  [15]                     3 s 168 ms
  [16]                     3 s 202 ms
  [17]                     3 s 319 ms
  [18]                     3 s 538 ms
  [19]                     4 s 639 ms
```

```
Starting ChromeDriver 80.0.3987.106 (f68069574609230cf9b635cd784cfb1bf81bb53a-refs/branch-heads/3987@{#882}) on port 3
Only local connections are allowed.
Please protect ports used by ChromeDriver and related test frameworks to prevent access by malicious code.
三月 24, 2020 12:09:58 上午 org.openqa.selenium.remote.ProtocolHandshake createSession
信息: Detected dialect: W3C
```

Tests passed: 20 (a minute ago)