Software Testing Technique
Chapter 8

Test Automation and Selenium

Zan Wang（王赞）
Office: 55-A413
Email: wangzan@tju.edu.cn

2022

Outline

- Introduction to Test Automation
- Introduction to Selenium
- Selenium IDE
- Selenium Webdriver (Selenium 2)

INTRODUCTION TO TEST AUTOMATION

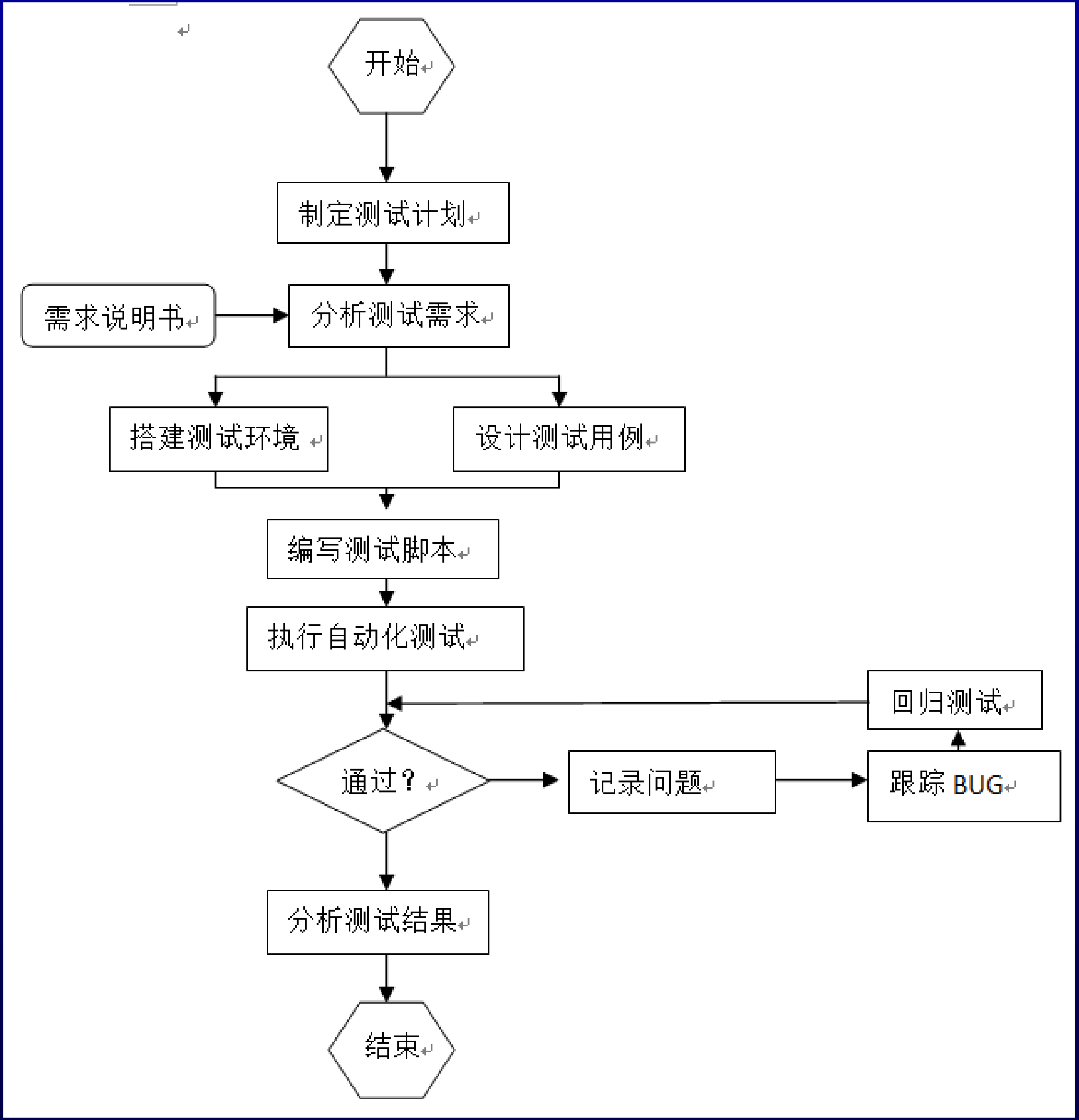
Test Automation

- Software testing is expensive and labor intensive.
- Software testing requires up to 50% of software development costs, and even more for safety-critical applications.
- Two kinds of task in software engineering
  - – Revenue task
  - – Exercise task
    - Candidates for automation
- Test automation means using a software tool to run repeatable tests against the application to be tested.

Advantages of test automation

- Frequent regression testing
- Rapid feedback to developers
- Virtually unlimited iterations of test case execution
- Support for Agile and extreme development methodologies
- Disciplined documentation of test cases
- Customized defect reporting
- Finding defects missed by manual testing

自动化测试的基本流程


开始
制定测试计划
需求说明书
分析测试需求
搭建测试环境
设计测试用例
编写测试脚本
执行自动化测试
回归测试
通过？
记录问题
跟踪 BUG
分析测试结果
结束

UI Automating Test

- UI automation testing, is similar to manual testing, but instead of having a user click through your application, and visually verify the data.
- Some UI testing tools：Selenium Webdriver，QTP
- Key steps：
  - Record
  - Replay
  - Compare and analyze

自动化测试的适用项目

- 软件需求变动不频繁
- 项目周期较长
- 测试脚本可重复使用
  - 兼容性测试

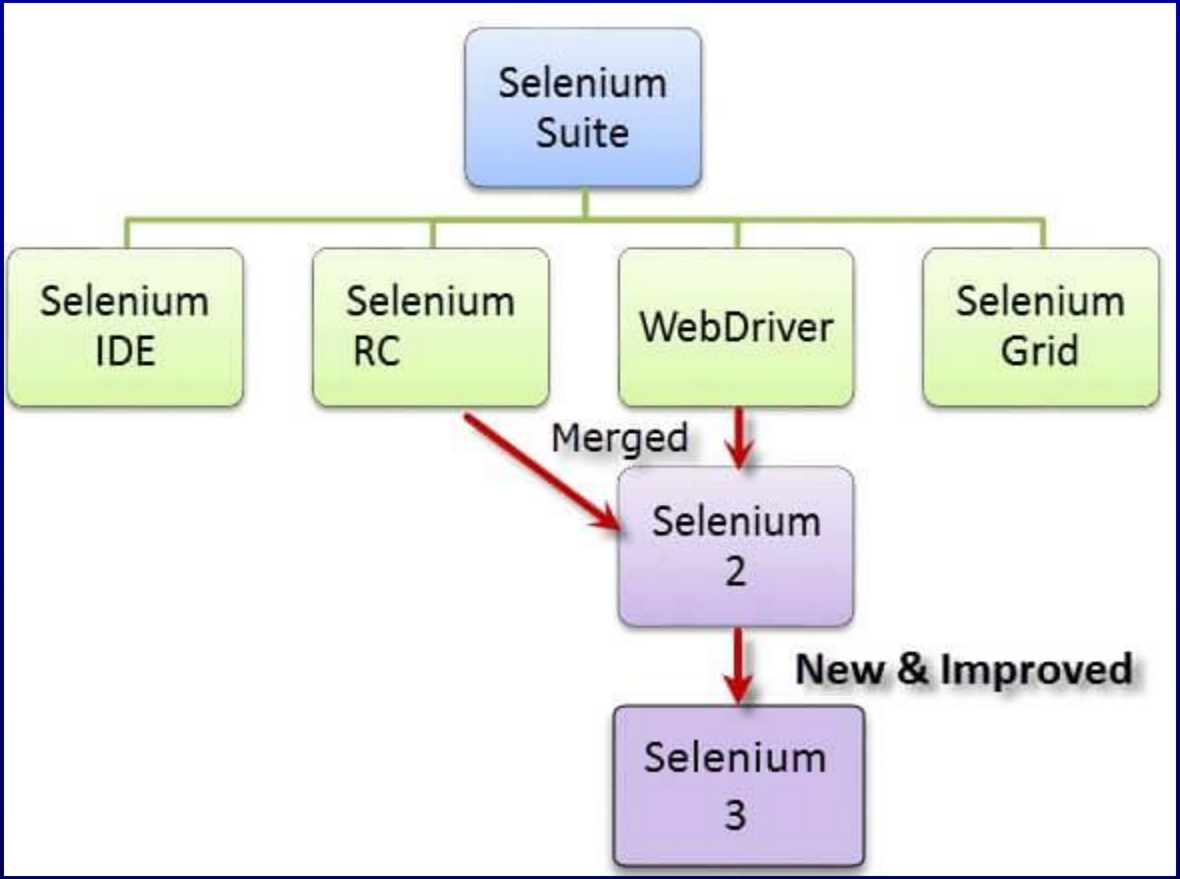
Some tools for UI automation testing

- HP Quick Test Professional (QTP)
  - Mercury，HP
  - Commercial
- Selenium
  - Thoughtworks
  - Open Source
- Appium
  - Mobile Testing
  - Open Source

INTRODUCTION TO SELENIUM

Selenium

- SELENIUM is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms.
- You can use multiple programming languages like Java, C#, Python etc. to create Selenium Test Scripts.
- Selenium is a set of different software tools each with a different approach to supporting test automation.
- The entire suite of tools results in a rich set of testing functions specifically geared to the needs of testing of web applications of all types. These operations are highly flexible, allowing many options for locating UI elements and comparing expected test results against actual application behavior.


Selenium Suite
Selenium IDE
Selenium RC
WebDriver
Selenium Grid
Merged
Selenium 2
New & Improved
Selenium 3

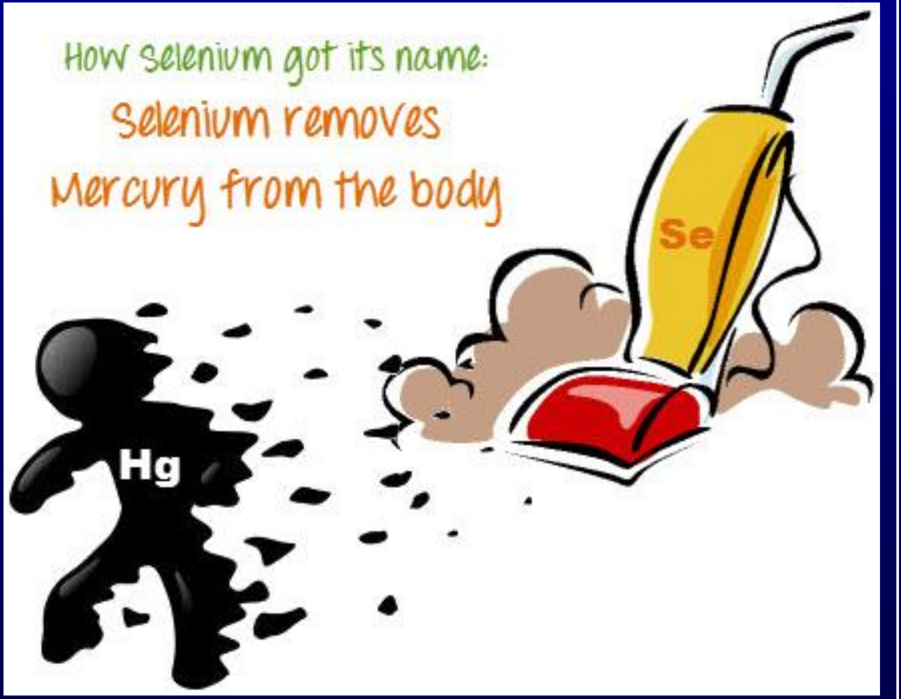
Brief History of the Selenium Project

- 2004, Jason Huggins at Thoughtworks. Selenium RC
  - Drive interactions with the page
  - Automatically rerun tests against multiple browsers
  - Selenium Remote Control(RC)
  - Ground-breaking for controlling a browser from a language of your choice.
- 2006, Shinya Kasatani, Selenium IDE
  - a Firefox extension that can automate the browser through a record-and-playback feature. He came up with this idea to further increase the speed in creating test cases. He donated Selenium IDE to the Selenium Project in 2006.
- 2007, Simon Stewart at Google, WebDriver
  - Google had long been a heavy user of Selenium
  - Pain-points
- 2009, merging of Selenium and WebDriver.
  - Most of the Selenium Project's efforts are focused on Selenium 2
- 2016, Selenium 3.0
- Now, Selenium 4 is released

- “Why are the projects merging? Partly because WebDriver addresses some shortcomings in selenium (by being able to bypass the JS sandbox, for example. And we’ve got a gorgeous API), partly because selenium addresses some shortcomings in WebDriver (such as supporting a broader range of browsers) and partly because the main selenium contributors and I felt that it was the best way to offer users the best possible framework.”

Some interesting things about the name

- Selenium vs. Mercury
  - Mercury: a great company in software testing
  - Flagship products:
    - LR: Loadrunner
    - QTP: Quicktest Professional
  - Acquired by HP in 2006.
- Eclipse vs. Sun



Selenium’s Tool Suite

- https://www.selenium.dev/
- Selenium 2,3 (aka. Selenium Webdriver)
- Selenium IDE
  - A prototyping toll for building test scripts.
  - A Firefox plugin and provides an easy-to-use interface for developing automated test.
- Selenium Grid
  - Run your tests in parallel.
  - Different tests can be run at the same time on different remote machines.

Supported Browsers and Platforms

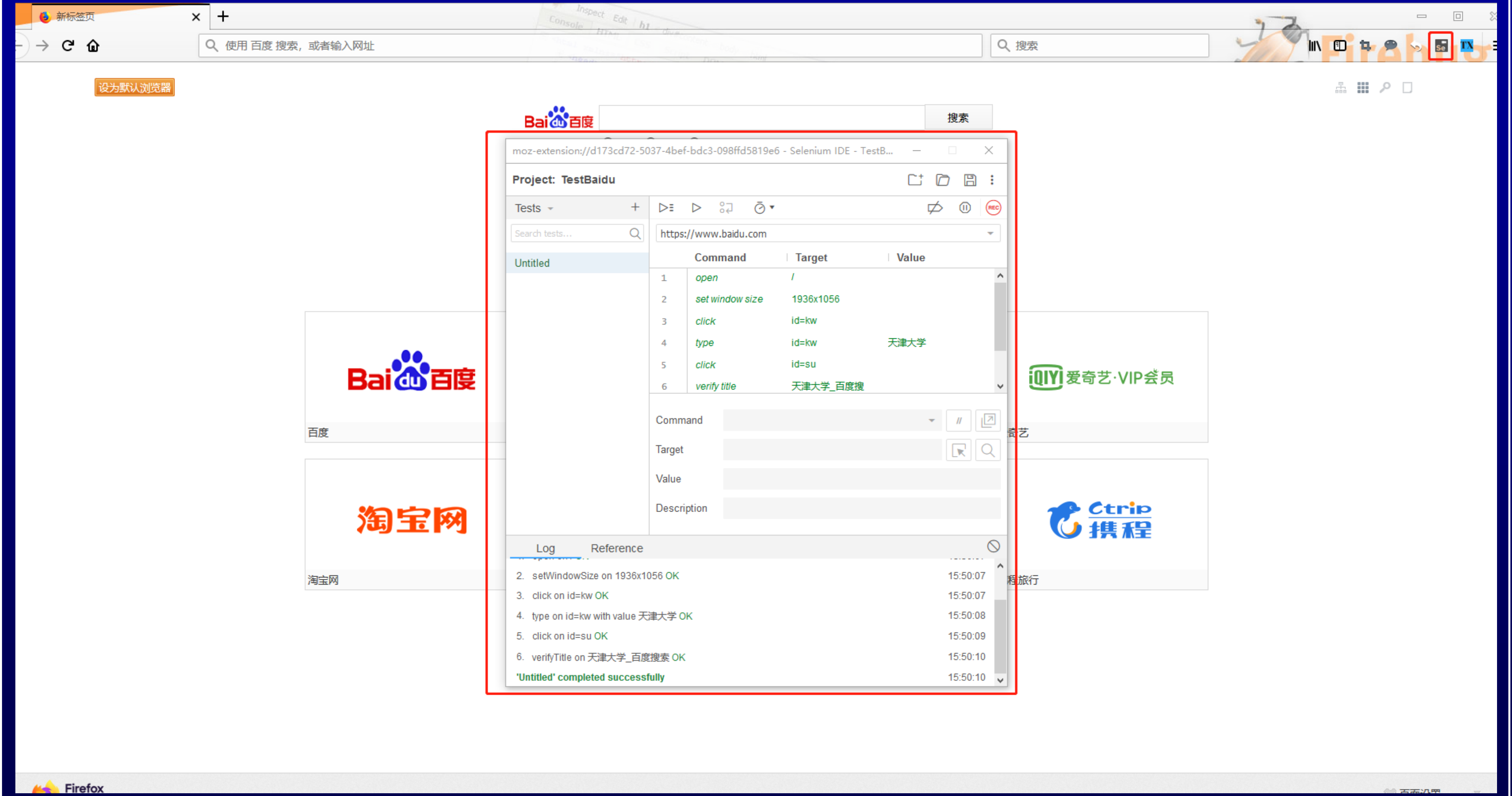
- Browsers:
  - Internet Explorer: 32 and 64-bit where applicable
  - Google Chrome
  - Firefox
  - Safari
  - Opera
- Mobile Platforms
  - Android(with Selendroid or appium)
  - iOS (with ios-driver or appium)

SELENIUM IDE

Selenium IDE

- Integrated development environment for Selenium tests
- Enables you to record a browser session
- Implemented as a Mozilla FireFox extension
- Allows you to record, edit, and debug tests.
- Selenium IDE should only be used as a prototyping tool

Selenium IDE

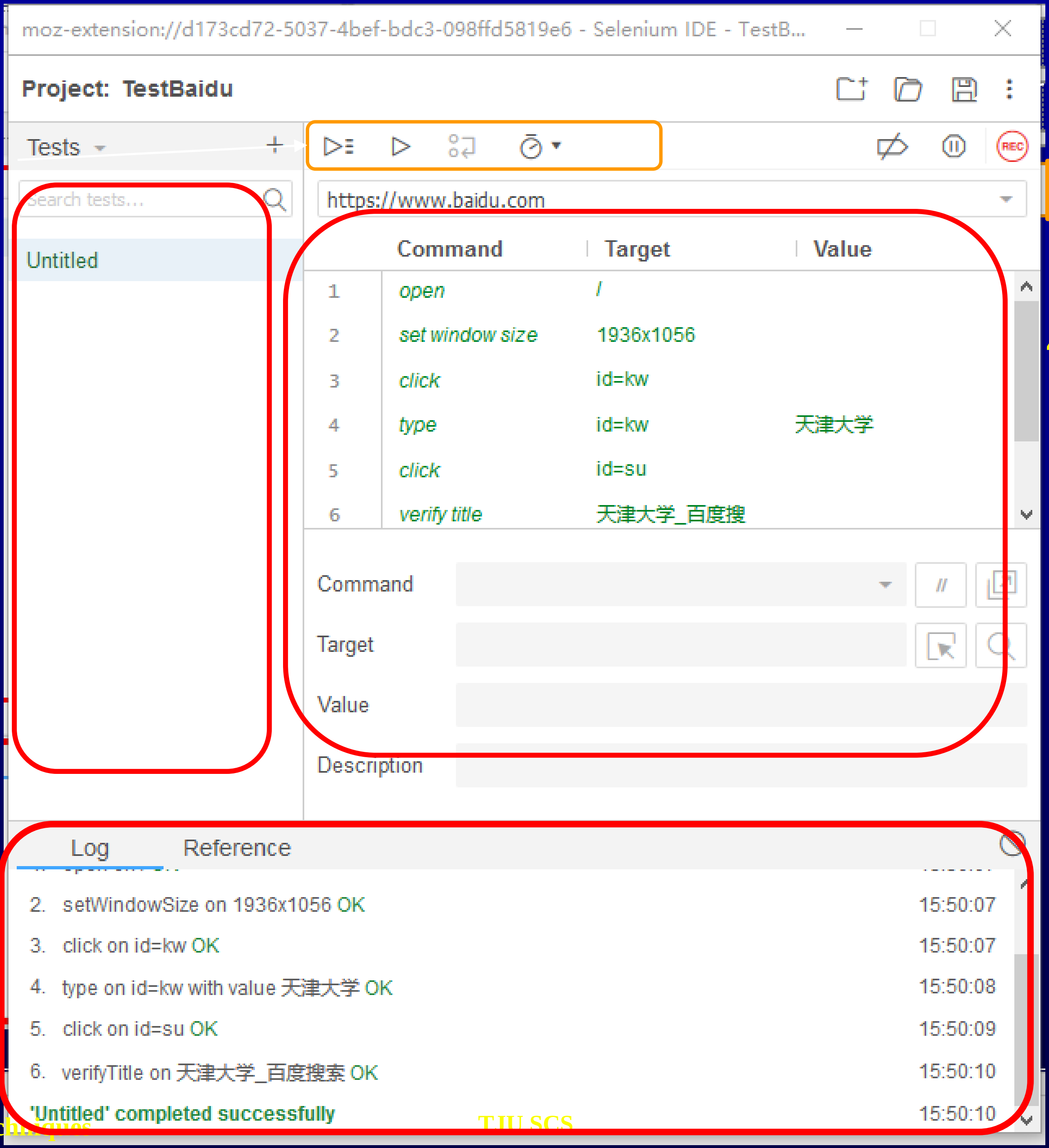


https://www.baidu.com

	Command	Target	Value
1	open	/	
2	set window size	1936x1056	
3	click	id=kw	
4	type	id=kw	天津大学
5	click	id=su	
6	verify title	天津大学_百度搜	

Command
Target
Value
Description

```
2.  setWindowSize on 1936x1056 OK                15:50:07
3.  click on id=kw OK                            15:50:07
4.  type on id=kw with value 天津大学 OK          15:50:08
5.  click on id=su OK                            15:50:09
6.  verifyTitle on 天津大学_百度搜索 OK            15:50:10
'Untitled' completed successfully                15:50:10
```



moz-extension://d173cd72-5037-4bef-bdc3-098ffd5819e6 - Selenium IDE - TestB...

Project: TestBaidu

Tests

Search tests...

Untitled

https://www.baidu.com

	Command	Target	Value
1	open	/	
2	set window size	1936x1056	
3	click	id=kw	
4	type	id=kw	天津大学
5	click	id=su	
6	verify title	天津大学_百度搜	

Command

Target

Value

Description

Log | Reference

```
2.  setWindowSize on 1936x1056 OK                      15:50:07
3.  click on id=kw OK                                  15:50:07
4.  type on id=kw with value 天津大学 OK                15:50:08
5.  click on id=su OK                                  15:50:09
6.  verifyTitle on 天津大学_百度搜索 OK                 15:50:10
'Untitled' completed successfully                      15:50:10
```

Selenium IDE

- **Base URL:**
  - The root of web application you want to test
- **Tool Bar:**
  - Buttons with Execution Commands
    - Run all tests
    - Run current test
    - Step over current command
    - Test execution speed: fast to slow
    - Disable breakpoints
    - Pause/ Resume
    - Record
- **Test Case**
    - Export

- **Test Steps**
  - **Command:**
  - **Target: the expression of locating the element of webpage**
  - **Value:**
  - **Insert new command**
- **Information Bar**
  - **Log:**
  - **Reference**

Create a test case

- Steps of creating a test case by Selenium IDE:
  - Record
  - Base URL
  - A sequence of actions

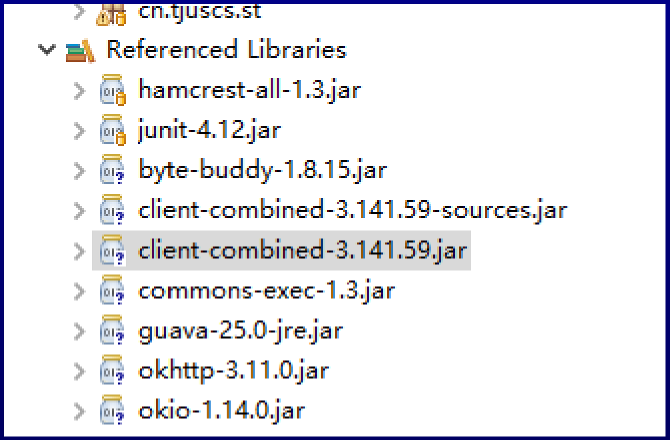
SELENIUM WEBDRIVER

Selenium Webdriver

- Designed to provide a simpler, more concise programming interface in addition to addressing some limitations in the Selenium API
- Developed to better support dynamic web pages where elements of a page may change without the page itself being reloaded
- Makes direct calls to the browser using each browser's native support for automation.
- Not tied to any particular test framework, so it can be used equally well in unit testing or from a plain old "main" method.

Get Webdriver

- Download lib of selenium3.141
  - http://selenium.dev
  - Client-combined-3.141.59.jar
  - Add to the build path of eclipse



```
> cn.tjuscs.st
v Referenced Libraries
  > hamcrest-all-1.3.jar
  > junit-4.12.jar
  > byte-buddy-1.8.15.jar
  > client-combined-3.141.59-sources.jar
  > client-combined-3.141.59.jar
  > commons-exec-1.3.jar
  > guava-25.0-jre.jar
  > okhttp-3.11.0.jar
  > okio-1.14.0.jar
```

Some simple actions by Webdriver

- Install a driver of a kind of web browser
  - Firefox,
  - Chrome: chromedriver.exe
  - Edge: msedgedriver.exe
- Open a URL

```
WebDriver driver = new FirefoxDriver();
driver.get(“http://www.baidu.com”);
```

Using Webdriver

- **Find web elements by webdriver**
  - **By ID**
    - <input type="text" name="passwd“ id="passwd-id" />
    - **WebElement element = driver.findElement(By.id(“passwd-id”));**
  - **By Name**
    - **WebElement element = driver.findElement(By.name(“passwd”));**
  - **By Xpath**
    - WebElement element =driver.findElement(By.xpath("//input[@id='passwd-id']"));
    - Xpath教程
      - **https://www.w3school.com.cn/xpath/index.asp**

- **By Class Name**
  - <div class="cheese">
    - <span>Cheddar</span>
  - </div>
  - <div class="cheese">
    - <span>Gouda</span>
  - </div>
  - List<WebElement>cheeses =   driver.findElements(By.className("cheese"));
- **By Link Text**
  - <ahref="http://www.google.com/search?q=cheese">cheese</a>>
  - WebElement chess = driver.findElement(By.linkText(“cheese”));

Actions on web elements-Input

- Text field or textarea
  - Findelement:
    - WebElement element = driver.findElement(By.id(“passwd-id”));
  - Input :
    - element.sendKeys(“test”);
  - Clear the text field
    - element.clear();
  - Get the text of text field
    - element.getText()

Actions on web elements-Select

- Select(下拉选择框)
  - Find the select element
    - Select select = new Select(driver.findElement(By.id(“select”)));
  - Select the value
    - select.selectByVisibleText(“mediaAgencyA”);
    - select.selectByValue(“001”);
  - Get the selected value
    - select.getAllSelectedOptions()
    - Select.getFirstSelectedOption();

Actions on web elements

- Radio Button
- Checkbox
- Button
- Form
- Popup dialogs
- Upload File
- Windows和Frames之间的切换
  - switchTo()

高级使用

- 读取Cookies
  - 增加Cookie：
    - Cookie cookie = new Cookie(“key”,”value”);
    - driver.manage().addCookie(cookie);
  - 获取Cookie值：
    - Set<Cookie> allCookies = driver.manage().getCookies();
    - For (Cookie loadedCookie: allCookies){

      System.out.println(loadedCookie.getName(), loadedCookie.getValue());

      }
  - 根据某个cookie的name获取cookie的值：
    - driver.manage().getCookieNamed(“mmsid”);

两种方法去操作页面元素

- 单击
  - 直接通过Webdriver函数：
    - WebElement baiduSearch = driver.findElement(By.id(“su”));
    - baiduSearch.click();
  - 键盘模拟单击
    - Actions actions = new Actions(driver);
    - actions.click(driver.findElement(By.id(“su”))).build().perform();

两种方法去操作页面元素

- 输入值
  - **Sendkeys()**
    - WebElement baiduTextBox = driver.findElement(By.id("kw"));
    - baiduTextBox.sendKeys("selenium书籍");
  - 键盘模拟输入
    - Actions actiona = new Actions(driver);
    - actiona.sendKeys(baiduTextBox, "qtp");
    - actiona.build().perform();

参考文档

- https://www.selenium.dev/documentation/zh-cn/getting_started/