

Table of Contents

| | |
|-------------------|-------|
| Web自动化测试课程 | 1.1 |
| 第1章-Web自动化入门 | 1.2 |
| Web自动化测试 | 1.2.1 |
| Web自动化测试工具选择 | 1.2.2 |
| Selenium IDE安装与运行 | 1.2.3 |
| WebDriver概述、环境搭建 | 1.2.4 |

Web自动化测试课程

| 序号 | 阶段 | 知识点 |
|----|---------------------------------|---|
| 1 | 第一阶段 Web 自动化入门 | 1. 认识自动化及自动化测试 2. 自动化测试工具选择 3. SeleniumIDE插件的使用 4. Firebug插件工具及使用 5. WebDriver概述、环境搭建 |
| 2 | 第二阶段 WebDriver-API 操作 | 1. 元素定位方式 2. 元素操作 3. 浏览器的操作方法 4. 鼠标操作 5. 键盘操作 6. 元素等待 7. HTML特殊元素处理 8. 窗口截图 |
| 3 | 第三阶段 UnitTest | 1. UnitTest框架 2. UnitTest断言 3. 生成HTML测试报告 |

课程目标

1. 掌握使用Selenium进行Web自动化测试的流程和方法，并且能够完成自动化测试脚本的编写。
2. 掌握如何通过UnitTest管理用例脚本，并生成HTML测试报告。

第1章-Web自动化入门

Web自动化测试

目标

1. 了解什么是自动化
2. 理解什么是自动化测试
3. 知道自动化测试能解决什么问题
4. 理解什么样的Web项目适合自动化测试

1. 什么是自动化

概念：由机器设备代替人为自动完成指定目标的过程

1.1 优点

1. 减少人工劳动力
2. 工作效率提高
3. 产品规格统一标准
4. 规模化(批量生产)

2. 什么是自动化测试

概念：让程序代替人为去验证程序功能的过程

2.1 自动化测试能解决什么问题？

1. 解决-回归测试
2. 解决-压力测试
3. 解决-兼容性测试
4. 提高测试效率,保证产品质量

回归测试：项目在发新版本之后对项目之前的功能进行验证；

压力测试：可以理解多用户同时去操作软件，统计软件服务器处理多用户请求的能力

兼容性测试：不同浏览器（IE、Firefox、Chrome）等等

2.2 自动化测试相关知识

优点

1. 较少的时间内运行更多的测试用例；
2. 自动化脚本可重复运行；
3. 减少人为的错误；
4. 克服手工测试的局限性；

误区

1. 自动化测试可以完全替代手工测试；
2. 自动化测试一定比手工测试厉害；
3. 自动化测试可以发掘更多的BUG；
4. 自动化测试适用于所有功能；

自动化测试分类

1. Web-自动化测试(本阶段学习)
2. 移动-自动化测试
3. 接口-自动化测试
4. 单元测试-自动化测试

3. 什么是Web自动化测试

概念：让程序代替人为自动验证Web项目功能的过程

3.1 什么Web项目适合做自动化测试？

1. 需求变动不频繁
2. 项目周期长
3. 项目需要回归测试

3.2 Web自动化测试在什么阶段开始？

功能测试完毕(手工测试)

3.3 Web自动化测试所属分类

1. 黑盒测试(功能测试)
2. 白盒测试(单元测试)
3. 灰盒测试(接口测试)

Web自动化测试属于黑盒测试(功能测试)

4. 总结

1. 自动化测试的概念?
2. 自动化测试能解决什么问题?
3. 什么样的Web项目适合自动化测试?
4. Web自动化测试所属分类?

Web自动化测试工具选择

目标

1. 了解Web自动化测试常用工具
2. 熟悉Selenium的特点

1. 主流的Web自动化测试工具

1. QTP
QTP是一个商业化的功能测试工具，收费，支持web，桌面自动化测试。
2. Selenium（本阶段学习）
Selenium是一个开源的web自动化测试工具，免费，主要做功能测试。
3. Robot framework
Robot Framework是一个基于Python可扩展地关键字驱动测试的测试自动化框架。

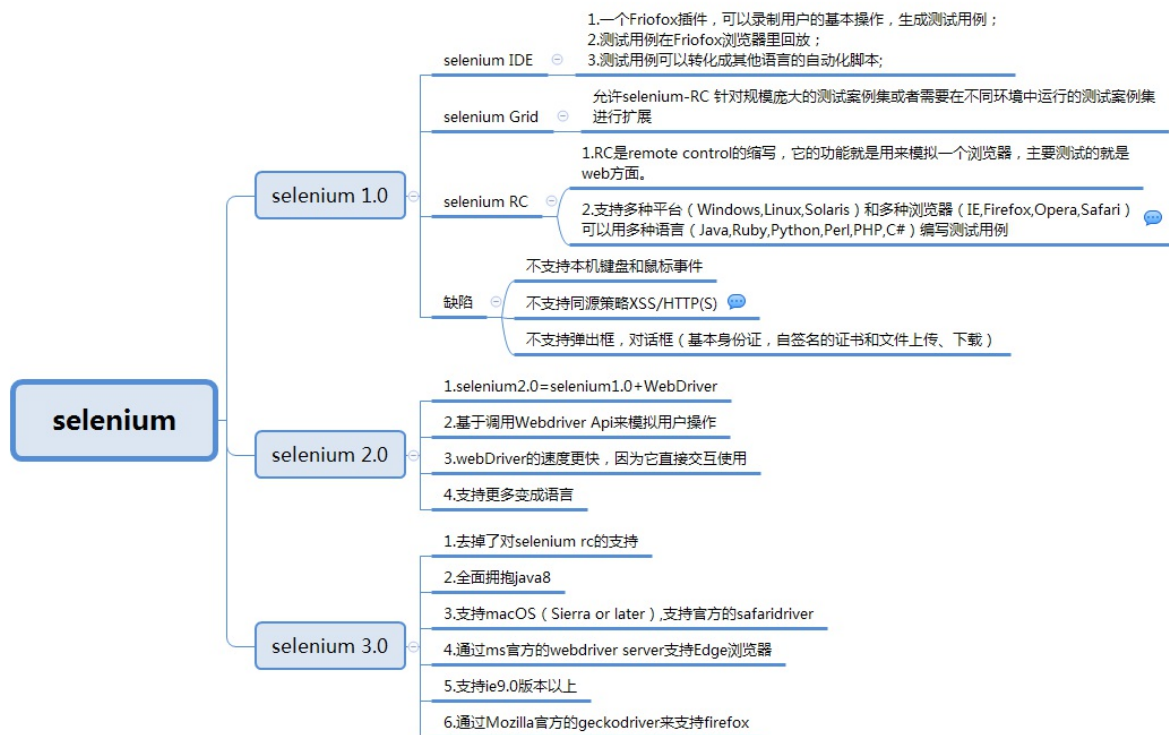
2. 什么是Selenium？

Selenium是一个用于Web应用程序的自动化测试工具；中文的意思（硒）

2.1 Selenium特点

1. 开源软件：源代码开放可以根据需要来增加工具的某些功能
2. 跨平台：linux 、 windows 、 mac
3. 核心功能：可以在多个浏览器上进行自动化测试
4. 支持多种语言：Python、Java、C#、JavaScript、Ruby、PHP等
5. 成熟稳定：目前已经被google，百度， 腾讯等公司广泛使用
6. 功能强大：能够实现类似商业工具的大部分功能，因为开源性，可实现定制化功能

2.2 Selenium发展史【了解】



重点：

1. SeleniumIDE
2. Selenium2.0(WebDriver)

Selenium IDE安装与运行

目标

1. 能够使用seleniumIDE进行简单脚本的录制
2. 能够使用seleniumIDE导出python语言的脚本
3. 能够使用firebug定位元素和查看元素信息

1. Selenium IDE 是什么？

Selenium IDE: 是一个Firefox插件，用于记录和播放用户与浏览器的交互。（录制Web操作脚本）

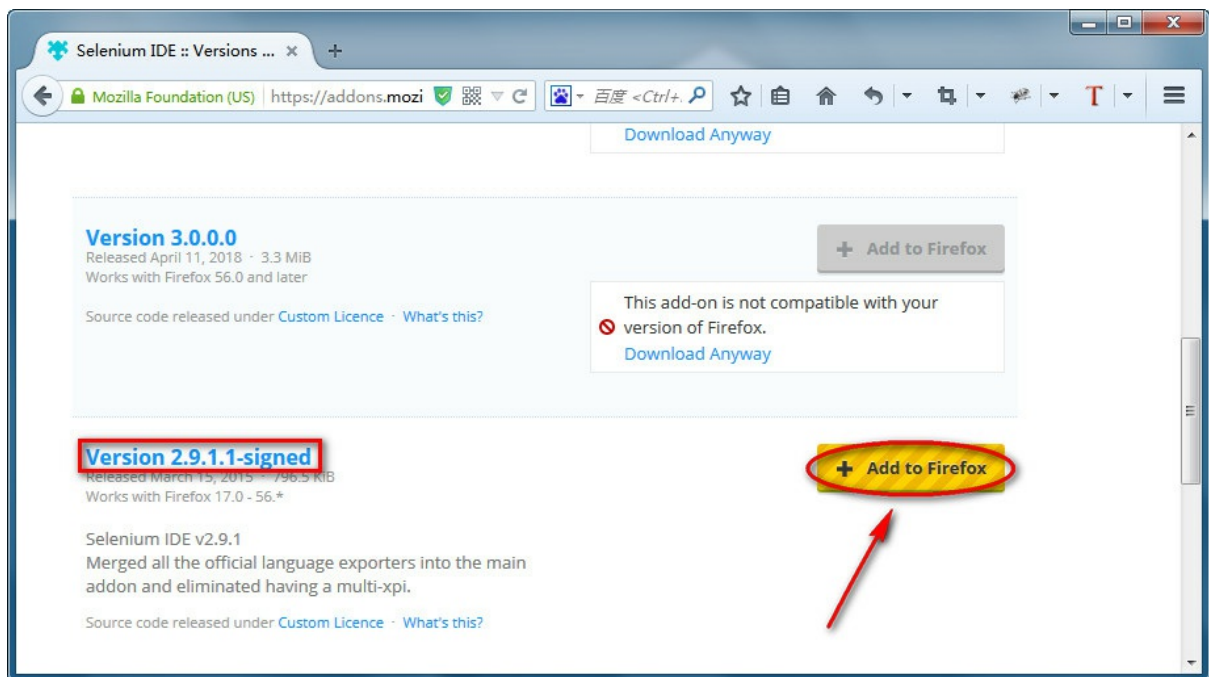
1.1 为什么要学习Selenium IDE？

1. 使用Selenium IDE录制脚本，体验自动化脚本魅力
2. 使用Selenium IDE录制的脚本转换为代码语言(参考借鉴)

1.2 安装方式

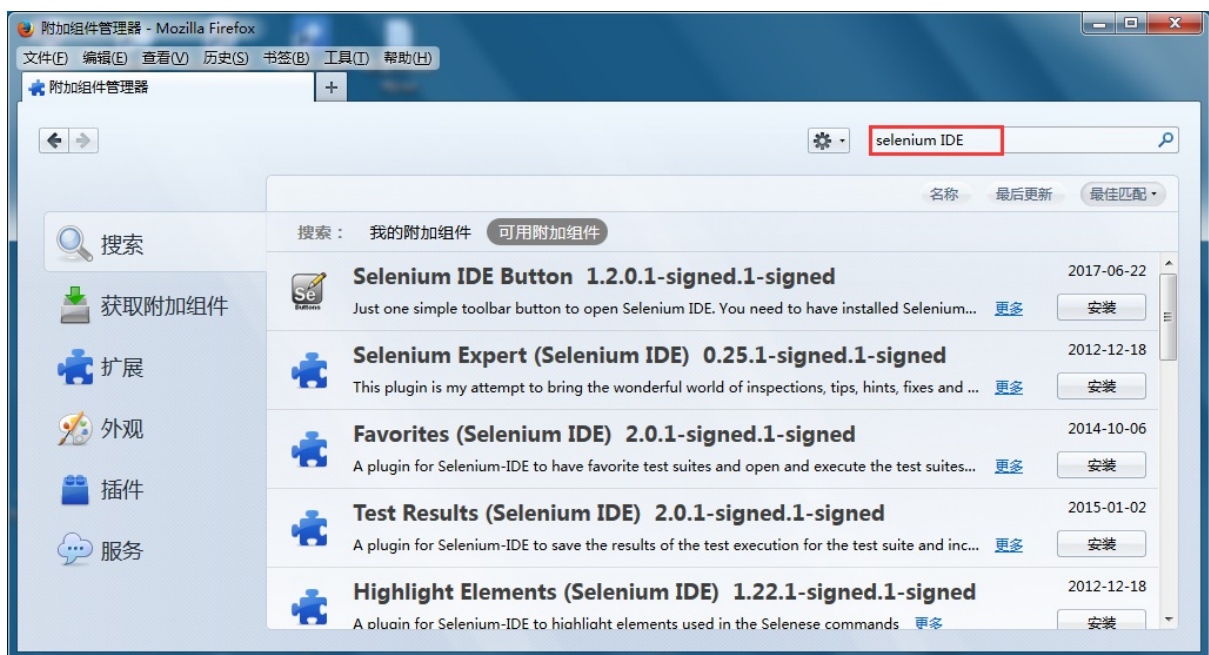
1. 官网安装
Version: 2.9.1.1
通过官网安装插件: <https://addons.mozilla.org/en-GB/firefox/addon/selenium-ide/versions/>
2. 附加组件管理器
 - 1). 火狐浏览器 V24-V35
 - 2). 附加组件管理器-->搜索selenium IDE提示:
 1. IDE前面有个空格
 2. 附加组件管理器启动方式-
 - 1) 工具菜单->附加组件
 - 2) Ctrl+Shift+A
3. 离线安装
下载: <https://github.com/SeleniumHQ/selenium-ide/releases>
安装: 下载好selenium_ide-2.9.1-fx.xpi直接拖入浏览器安装

官网安装

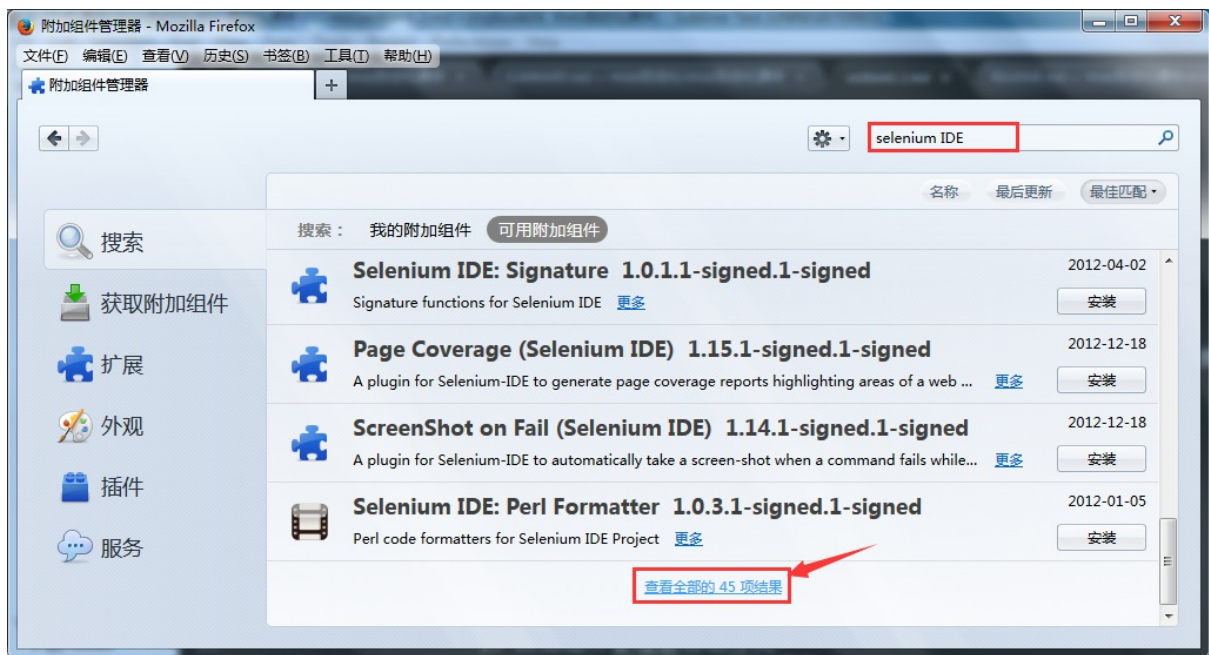


附加组件管理器 安装

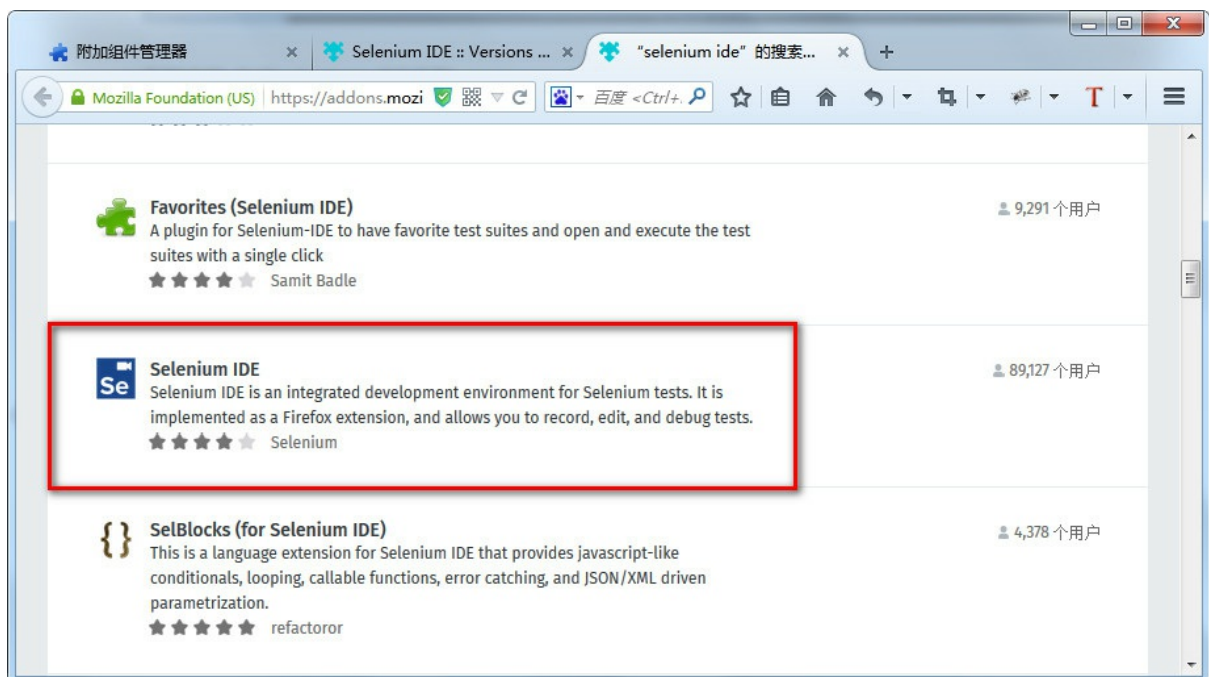
搜索selenium IDE示意图



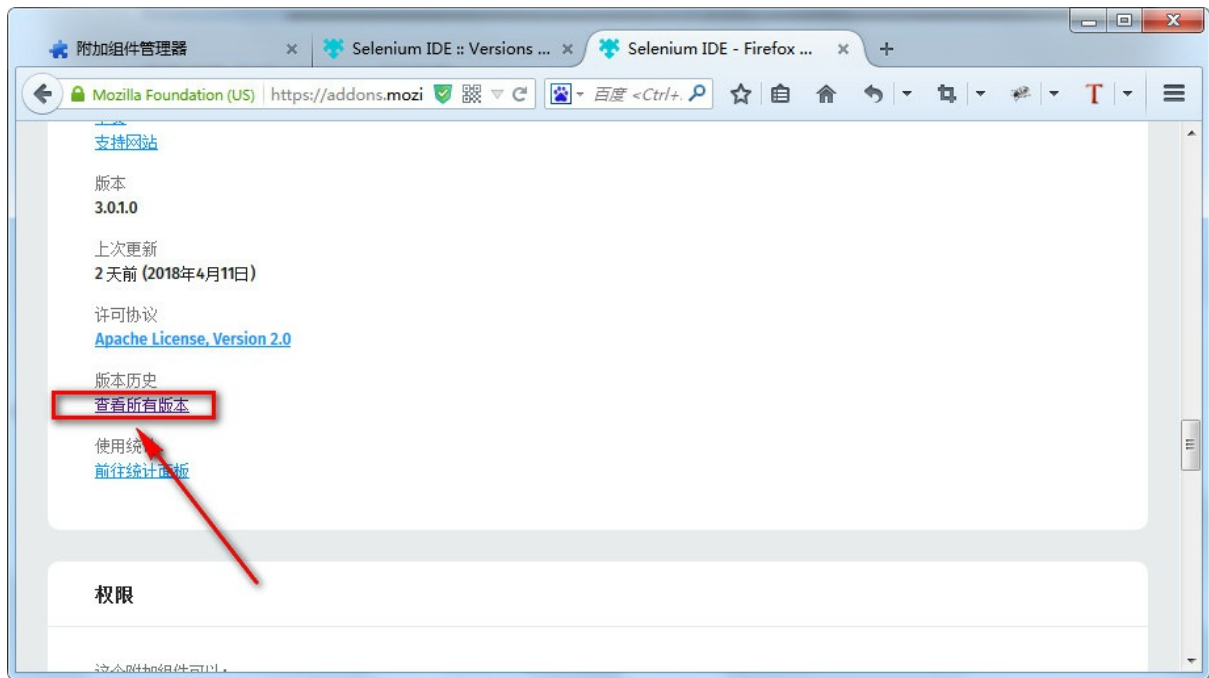
点击查看全部XX项结果



选择selenium IDE安装



滚动条下拉选择 查看所有版本

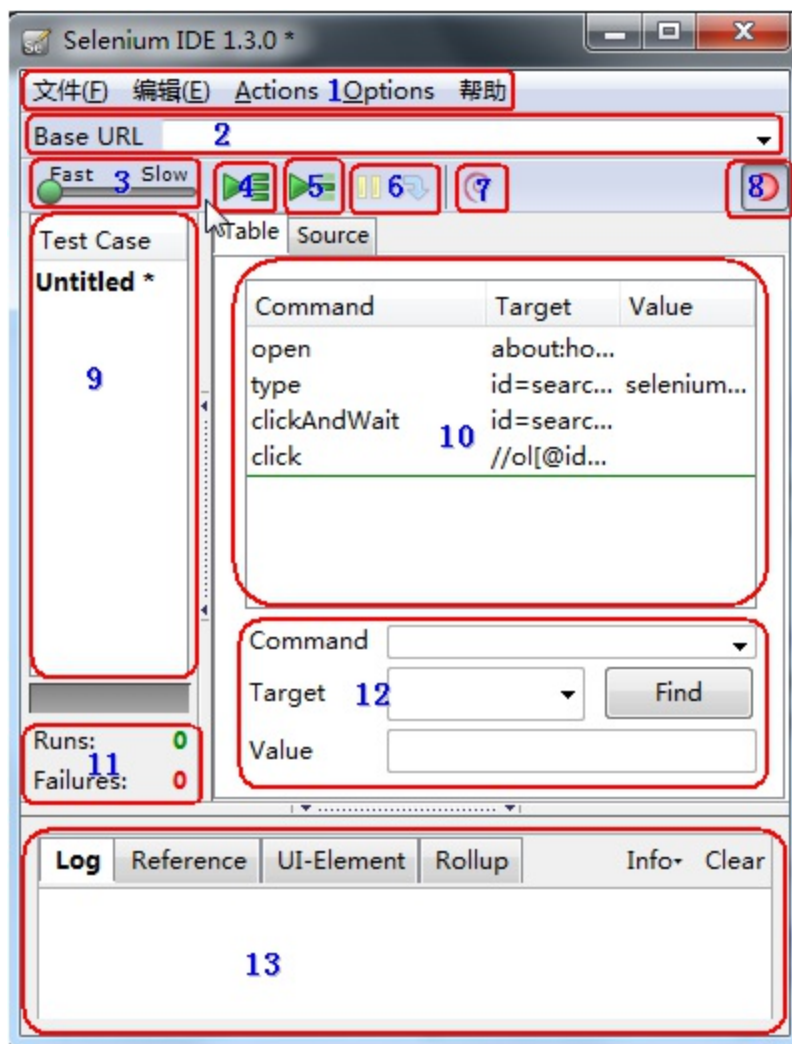


火狐浏览器 V35.0 选择 selenium IDE 2.9.1.1

1.3 Selenium IDE运行

启动方式

1. 点击菜单‘工具’—>Selenium IDE
2. Ctrl+Alt+S
3. 点击浏览器工具栏中的Selenium IDE图标



1. 文件：创建、打开和保存测试案例和测试案例集。编辑：复制、粘贴、删除、撤销和选择测试案例中的所有命令。

Options：用于设置selenium IDE。

2. 用来填写被测网站的地址。

3. 速度控制：控制案例的运行速度。

4. 运行所有：运行一个测试案例集中的所有案例。

5. 运行：运行当前选定的测试案例。

6. 暂停/恢复：暂停和恢复测试案例执行。

7. 单步：可以运行一个案例中的一行命令。

8. 录制：点击之后，开始记录你对浏览器的操作。

9. 案例集列表。

10. 测试脚本；table标签：用表格形式展现命令及参数。source标签：用原始方式展现，默认是HTML语言格式，

也可以用其他语言展示。

11. 查看脚本运行通过/失败的个数。

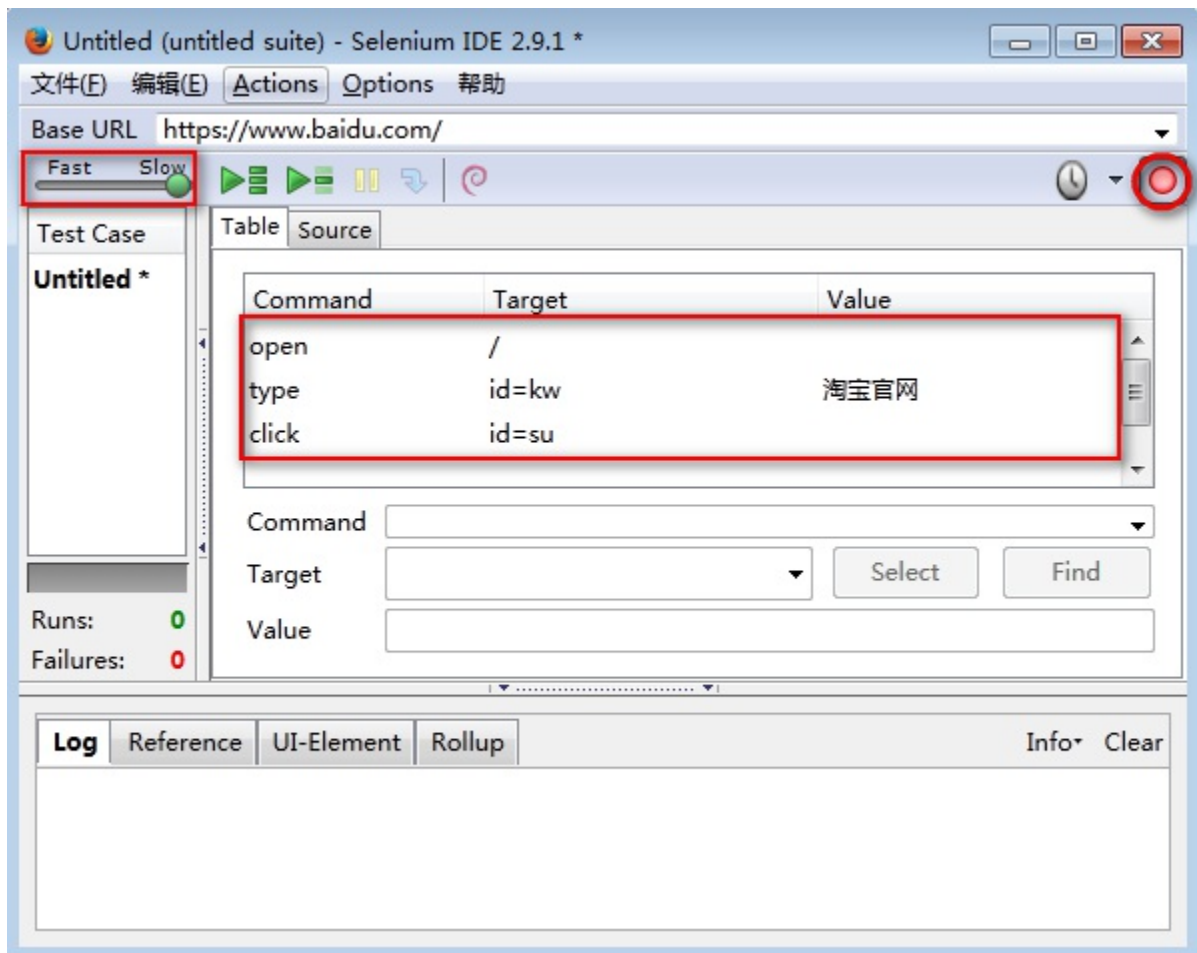
12. 当选中前命令对应参数。

13. 日志/参考/UI元素/Rollup

练习

需求：使用Selenium IDE插件录制->打开百度搜索引擎，搜索框输入[淘宝官网]，点击[百度一下]按钮

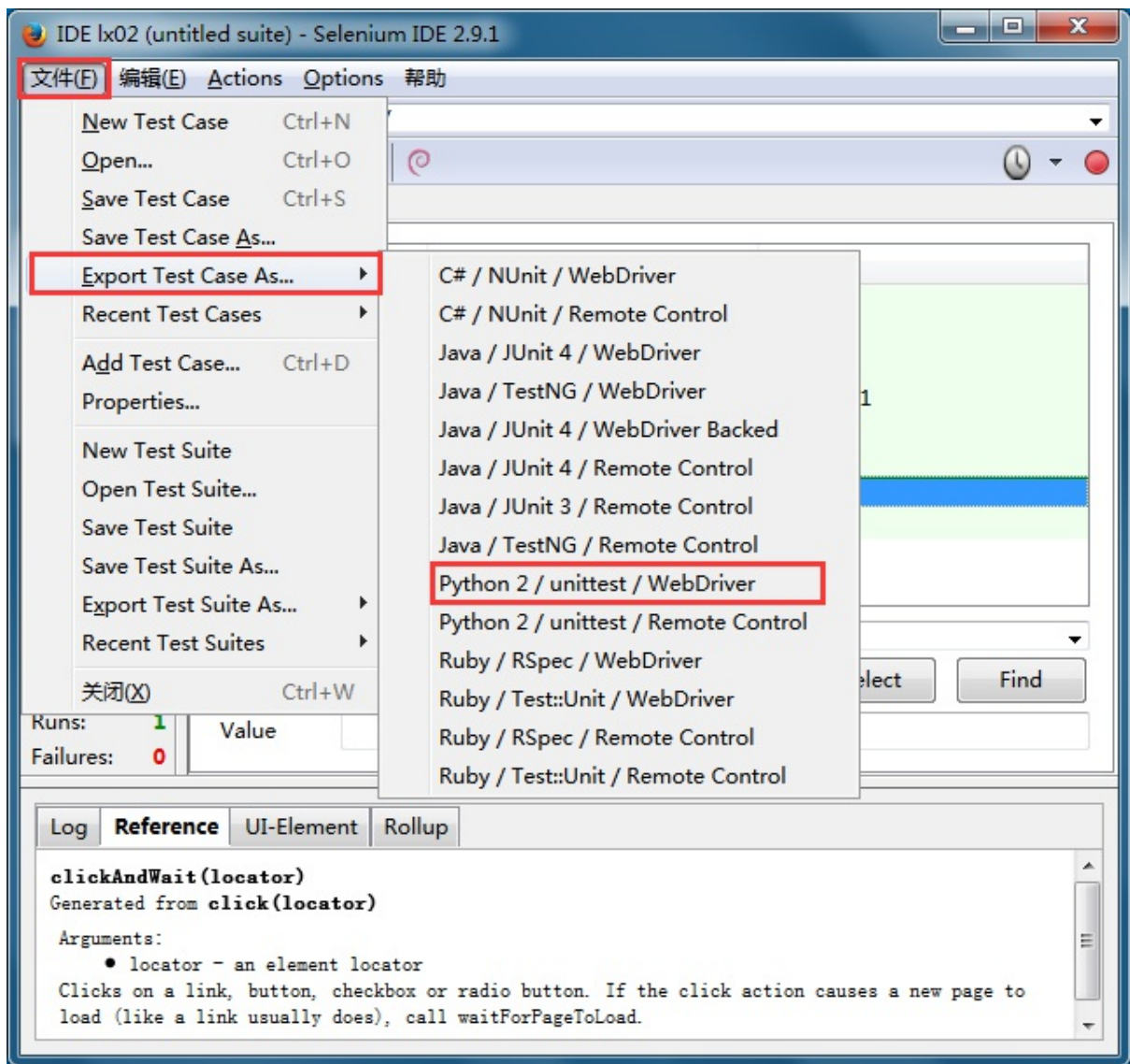
练习 脚本



注意

1. 录制：录制时红色录制按钮一定要打开->按下状态
2. 回放：由于网络延迟原因-建议选择最低
3. 浏览器：回放时浏览器要保持打开状态（否则点击回放，脚本无响应）

将录制脚本转换成**Python**语言



文件菜单->Export Test Cast As...->python2/unittest/WebDriver

提示

1. id=kw: 为百度搜索文本框id属性和值
2. id=su: 为百度一下按钮id属性和值

思考？

如何快速的查看一个元素的相关信息？

2. Firebug定位调试插件【重要】

Firebug是火狐浏览器一款插件，它集HTML查看和编辑、Javascript控制台、网络状况监视器于一体，是开发JavaScript、CSS、HTML和Ajax的得力助手。

作用：快速定位元素，查看元素信息

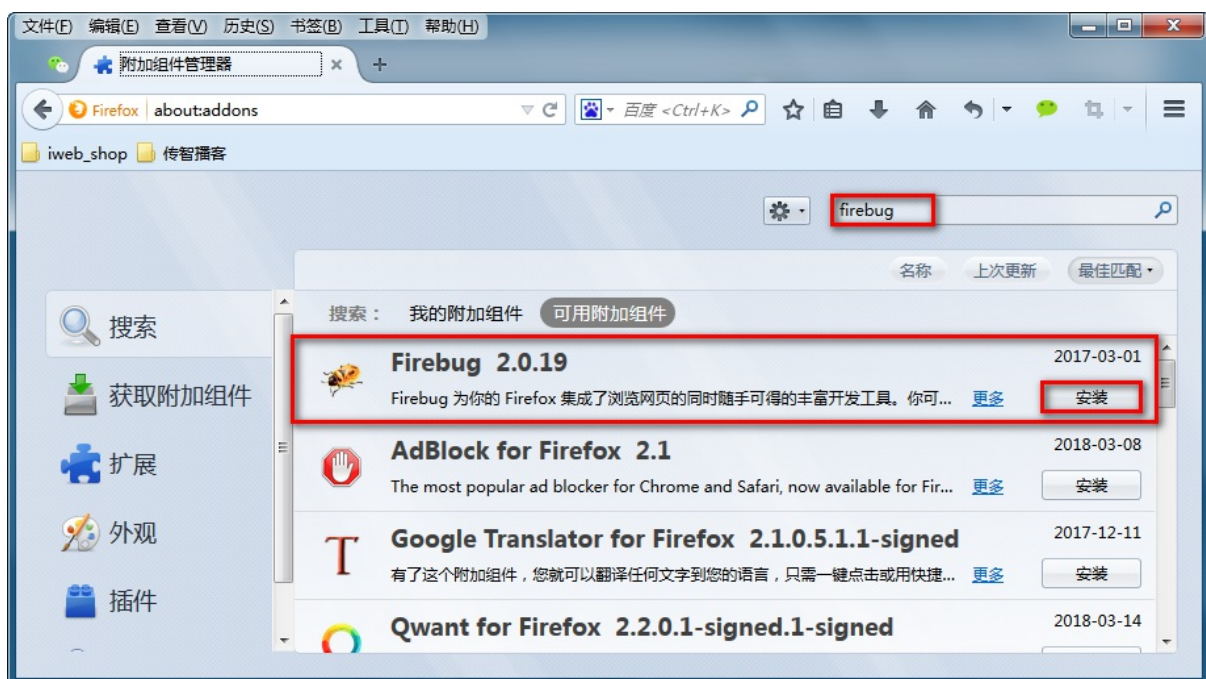
2.1 Firebug 插件安装

在线安装：

- 1). 附加组件管理器-->搜索FireBug

离线安装：

- 1). 将下载好的 firebug-2.0.19.xpi 直接拖入浏览器安装



3. Selenium IDE脚本编辑与操作 【了解】

提示：

通过Selenium IDE录制生成的用例步骤，有时候我们需要对脚本步骤进行修改（比如修改一些输入的数据），

或者要手动添加一些步骤，所以我们有必要学习对Selenium IDE脚本的编辑和操作。

3.1 编辑一行命令

在Table标签下选中某一行命令，命令由command、Target、value三部分组成。可以对这三部分内容那进行编辑。

3.2 插入命令

在某一条命令上右击，选择“insert new command”命令，就可以插入一个空白，然后对空白行进程编辑

3.3 插入注释

鼠标右击选择“insert new comment”命令插入注解空白行，本行内容不被执行，可以帮助我们更好的理解脚本，
插入的内容以紫色字体显示。

3.4 移动命令

有时我们需要移动某行命令的顺序，我们只需要左击鼠标拖动到相应的位置即可。

3.5 删除命令

选择单个或多个命令，然后点击鼠标右键选择“Delete”

3.6 命令执行

选定要执行的命令右键选择‘Execute this command’

注意：有一些命令必须依赖于前面命令的运行结果才能成功执行，否则会导致执行失败。

4. Selenium IDE常用命令【了解】

在这里我们只对几个常用的命令做个介绍

4.1 open(url)命令

作用：打开指定的URL，URL可以为相对或是绝对URL；

Target：要打开的URL； value值为空

- 1). 当Target为空，将打开Base URL中填写的页面；
- 2). 当Target不为空且值为相对路径，将打开Base URL + Target页面。
例如，假设Base URL为http://www.baidu.com，而Target为/error.html，
则执行open命令时，将打开http://www.baidu.com/error.html
- 3). 当Target以http://开头时，将忽略Base URL，直接打开Target的网址；

4.2 pause(waitTime)

作用：暂停脚本运行

waitTime: 等待时间，单位为ms; //Target=1000

4.3 goBack()

作用：模拟单击浏览器的后退按钮；

提示：由于没有参数，所以Target和Value可不填；

4.4 refresh()

作用：刷新当前页；

提示：由于没有参数，所以Target和Value可不填；

4.5 click(locator)

作用：单击一个链接、按钮、复选框或单选按钮；

4.6 type(locator,value)

作用：向指定输入域中输入指定值；

Target: 元素的定位表达式；

Value: 要输入的值；

4.7 close()

作用：关闭浏览器窗口

提示：由于没有参数，所以Target和Value可不填；

4.8 课堂练习

在Selenium IDE中手动完成以下操作的命令编写：

1. 访问百度网站: <http://www.baidu.com>
2. 搜索python
3. 点击百度一下
4. 暂停3秒

5. 访问hao123网站: `http://www.hao123.com`
6. 暂停3秒
7. 回退到百度页面
8. 暂停3秒
9. 刷新页面
10. 关闭页面

5. 总结

1. Selenium IDE安装、启动方式
2. 如何使用seleniumIDE导出python语言的脚本
3. Selenium IDE常用命令
4. FireBug作用和使用方式

WebDriver概述、环境搭建

目标

1. 了解WebDriver概述
2. 掌握WebDriver环境搭建
3. 熟练掌握web自动化测试脚本编写的基本步骤

1. 什么是WebDriver?

1. Webdriver (Selenium2) 是一种用于Web应用程序的自动化测试工具;
2. 它提供了一套友好的API;
3. Webdriver完全就是一套类库, 不依赖于任何测试框架, 除了必要的浏览器驱动;

说明:

API: 应用程序编程接口 (WebDriver类库内封装非常多的方法, 通过使用这些方法可以完成Web自动化测试)

思考

WebDriverAPI都支持哪些浏览器?

1.1 WebDriverAPI 支持的浏览器

1. Firefox (FirefoxDriver) 【推荐-本阶段学习使用】
2. Chrome (ChromeDriver)
3. IE (InternetExplorerDriver)
4. Safari (SafariDriver)
5. Opera (OperaDriver)
6. HtmlUnit (HtmlUnit Driver)

提示:

Firefox、Chrome: 对元素定位和操作有良好的支持, 同时对JavaScript支持也非常好。

IE: 只能在windows平台运行, 所有浏览器中运行速度最慢

HtmlUnit: 是一个没有界面的浏览器, 运行速度最快;

推荐原因:

1. Selenium IDE
2. FireBug

3. 对WebDriver API支持良好

1.2 WebDriverAPI 支持的开发语言

自动化测试：让程序代替人为去验证程序功能的过程

1. Python
2. Java
3. PHP
4. JavaScript
5. Perl
6. Ruby
7. C#

2. 环境搭建

基于Python环境搭建

1. Python 开发环境
2. 安装selenium包 — 提供了 WebDriver 的自动化开发工具
3. 安装浏览器驱动 — 保证能够用程序驱动浏览器，实现自动化测试

2.1 安装selenium包

前提：Python3 安装完毕且能正常运行

selenium 安装、卸载、查看命令

pip: 是一个通用的 Python 包管理工具。提供了对 Python 包的查找、下载、安装、卸载的功能。

安装: `pip install selenium==2.48.0`

1). `install`: 安装命令

2). `selenium==2.48.0`: 指定安装selenium2.48.0版本（如果不指定版本默认为最新版本）

卸载: `pip uninstall selenium`

查看: `pip show selenium`

2.2 安装浏览器驱动

火狐浏览器【推荐】

1. FireFox 48 以上版本
Selenium 3.X + FireFox驱动(geckodriver)
驱动下载地址: <https://github.com/mozilla/geckodriver/releases>
2. FireFox 48 以下版本
Selenium 2.X 内置驱动

谷歌浏览器

selenium2.x/3.x + Chrome驱动
驱动下载地址: <https://sites.google.com/a/chromium.org/chromedriver/downloads>

| chromedriver版本 | 支持的Chrome版本 |
|----------------|-------------|
| 2.41 | v67-69 |
| 2.40 | v66-68 |
| 2.39 | v66-68 |
| 2.38 | v65-67 |
| 2.37 | v64-66 |
| 2.36 | v63-65 |
| 2.35 | v62-64 |
| ... | ... |

IE浏览器(了解)

1. IE 9 以上版本
Selenium 3.X + IE驱动
 2. IE 9 以下版本
Selenium 2.X + IE驱动
- 驱动下载地址: <http://selenium-release.storage.googleapis.com/index.html>

总结

各个驱动下载地址: <http://www.seleniumhq.org/download/>

1. 浏览器的版本和驱动版本要一致!
(如果是32bit浏览器而Driver是64bit则会导致脚本运行失败!)
2. 浏览器驱动下载好后需要添加Path环境变量中, 或者直接放到Python安装目录, 因为Python已添加到Path中

3. 入门示例

3.1 需求

通过程序启动浏览器，并打开百度首页，暂停3秒，关闭浏览器

3.2 实现步骤

1. 导包
`from selenium import webdriver`
2. 创建浏览器驱动对象
Firefox浏览器: `driver = webdriver.Firefox()`
Chrome浏览器: `driver = webdriver.Chrome()`
IE浏览器: `driver = webdriver.Ie()`
3. 打开Web页面
`driver.get("http://www.baidu.com/")`
4. 暂停
`time.sleep(3)`
5. 关闭驱动对象
`driver.quit()`

3.3 示例代码

```
# 导包
from selenium import webdriver
import time

# 创建浏览器驱动对象
driver = webdriver.Firefox()
# driver = webdriver.Chrome()
# driver = webdriver.Ie()

# 加载web页面
driver.get("http://www.baidu.com/")

# 暂停3秒
time.sleep(3)

# 关闭驱动
driver.quit()
```

4. 总结

1. WebDriver是什么？
2. WebDriver环境搭建中涉及到的软件？
3. selenium 安装、卸载、查看命令？
4. web自动化测试脚本编写的基本步骤？
