# **Selenium IDE工具介绍**

## **简要介绍**

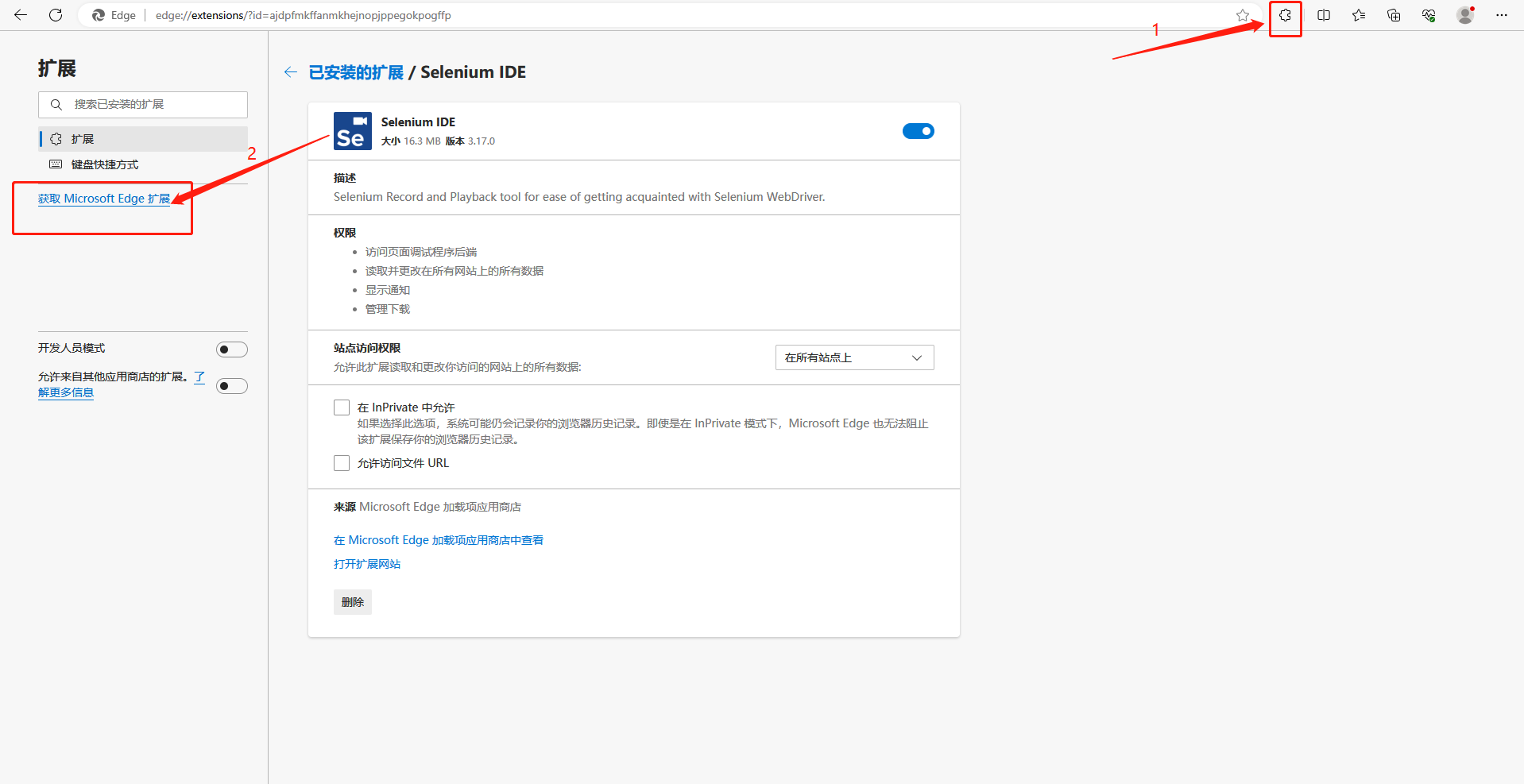
Selenium IDE是一个浏览器插件，提供了脚本的录制，回放以及编辑脚本的功能，还能将脚本导出为对应语言（C#、Java、Ruby或Python）的代码。

## **工具安装**

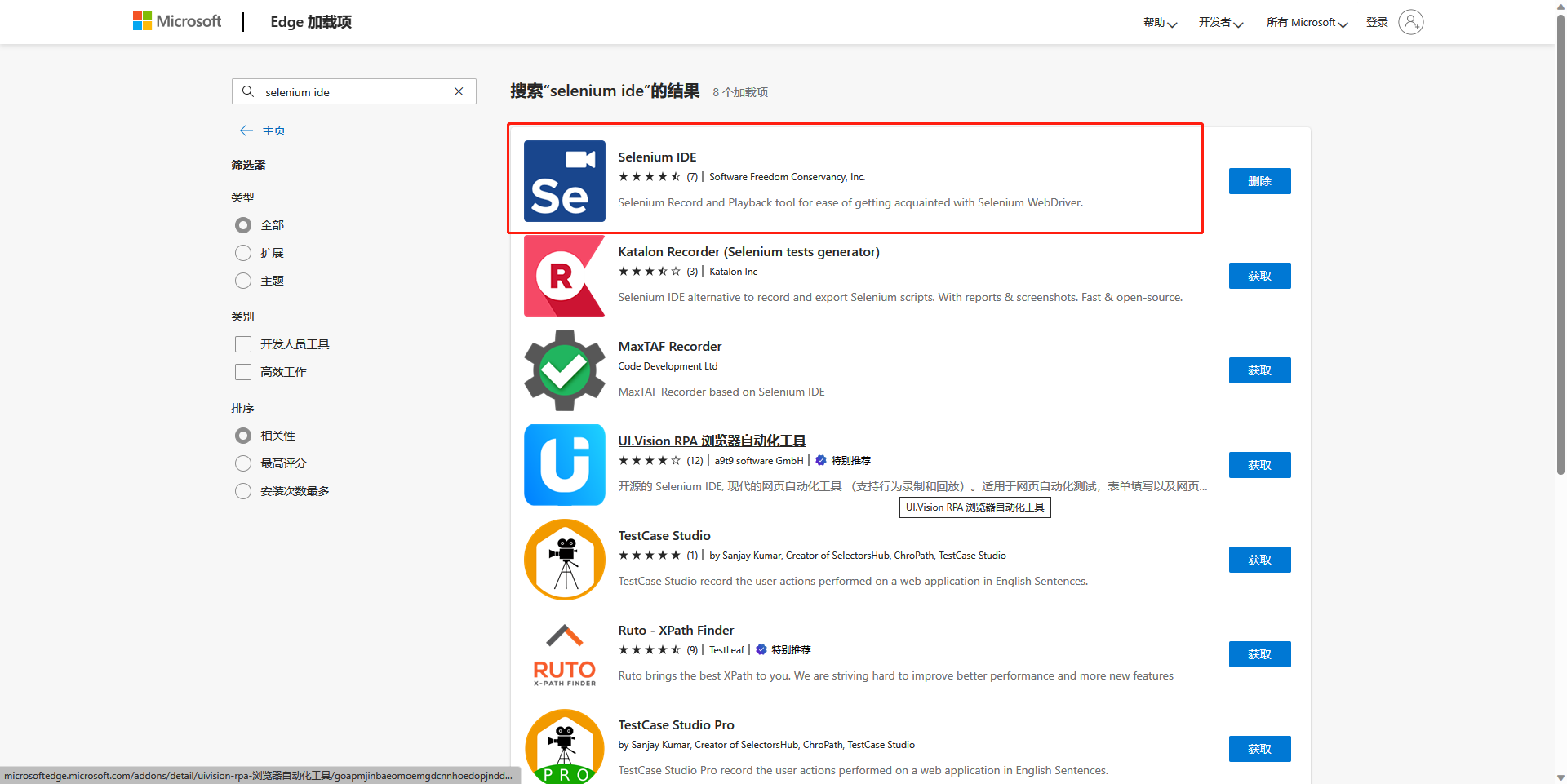
Selenium IDE支持Chrome，Firefox或者edge这三种浏览器，因为它们都有插件商店（但是其中谷歌需要代理服务才能使用谷歌的插件商店），在插件市场搜索selenium IDE即可找到对应的插件，安装完插件后点击就可以使用selenium IDE了。

**以Edge浏览器为例安装Selenium IDE插件**

* 1. 浏览器右上角找到拓展模块

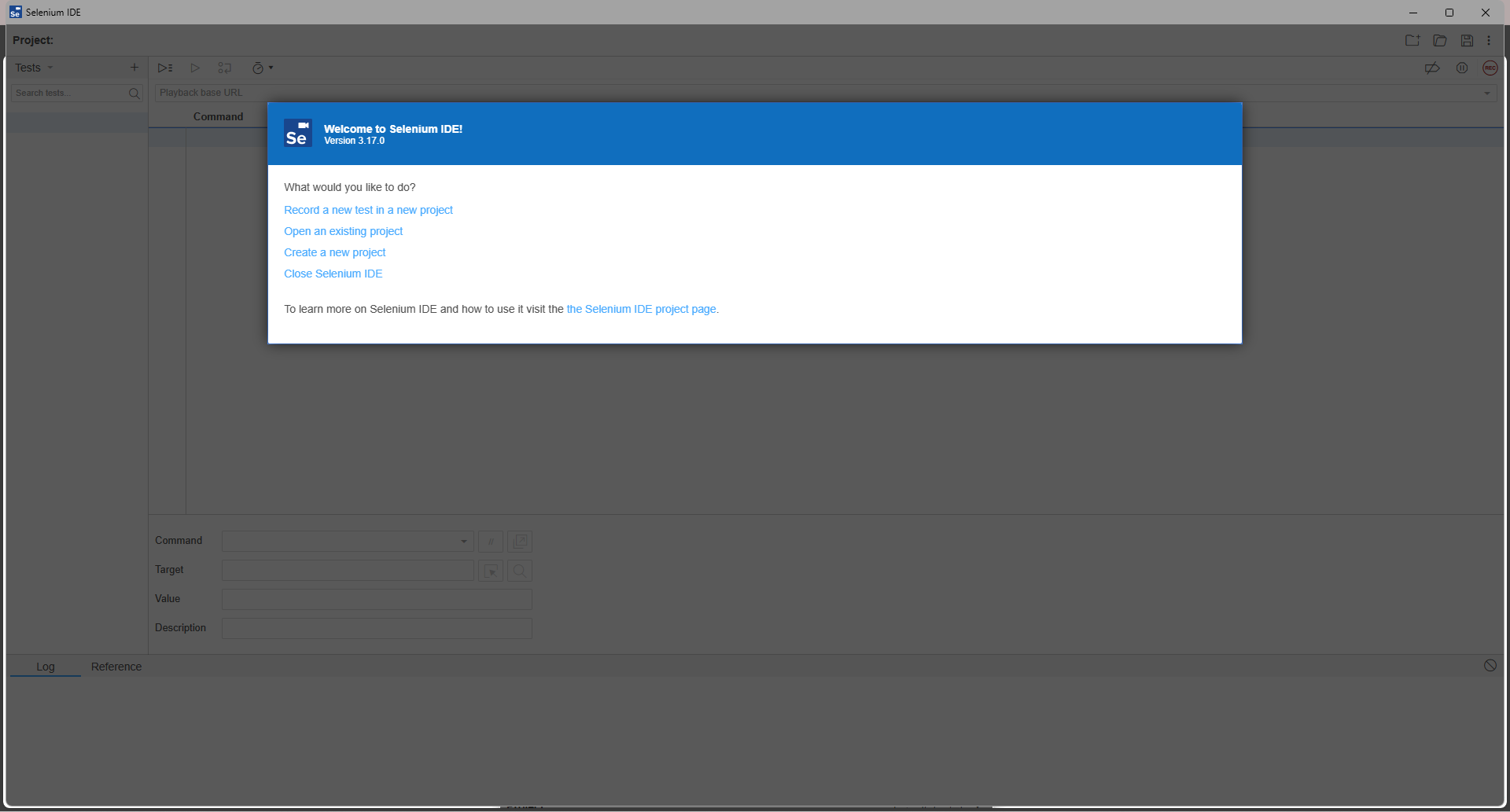


* 1. 搜索Selenium IDE，并下载安装即可

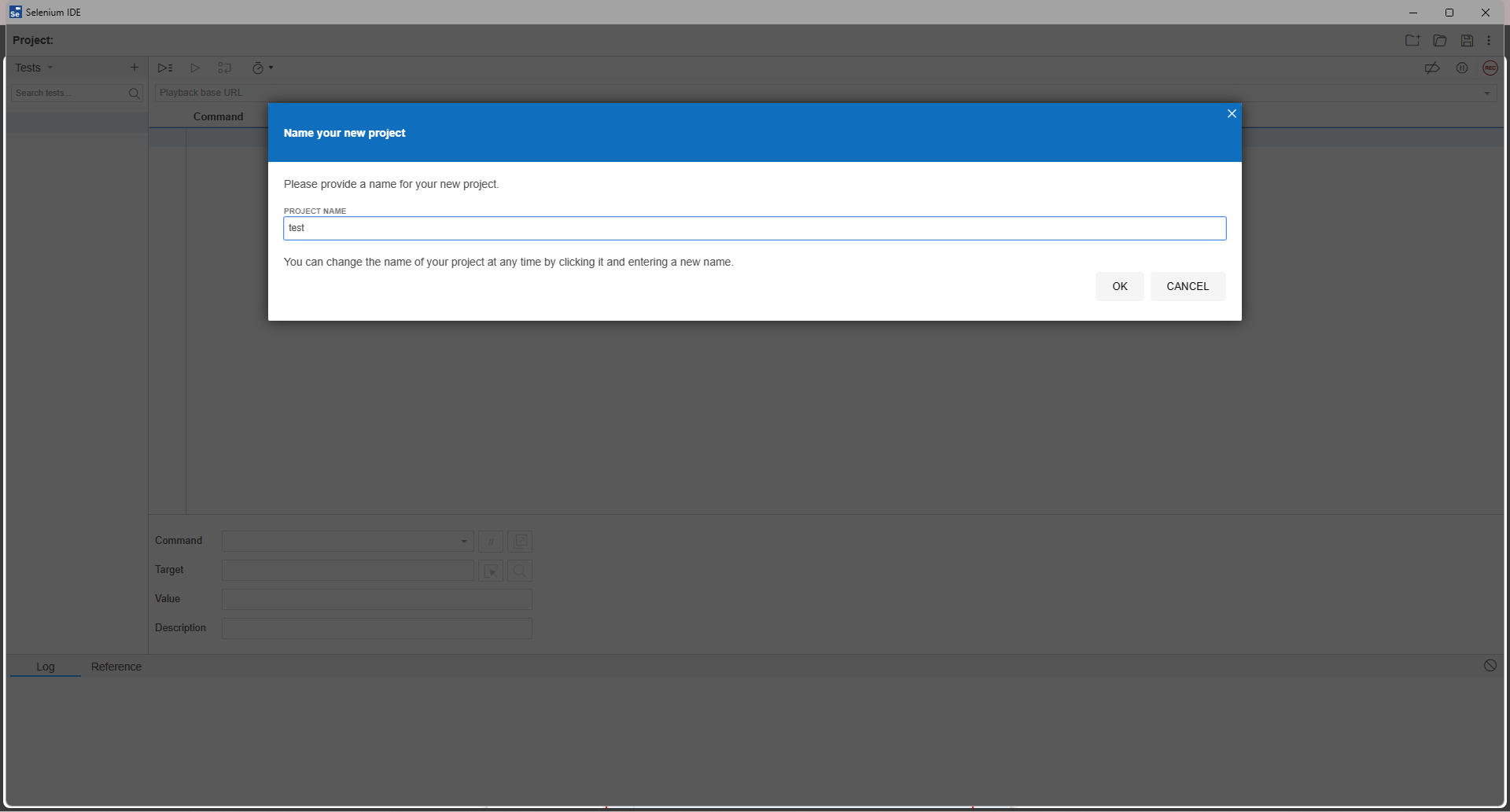


## Selenium IDE的基础使用

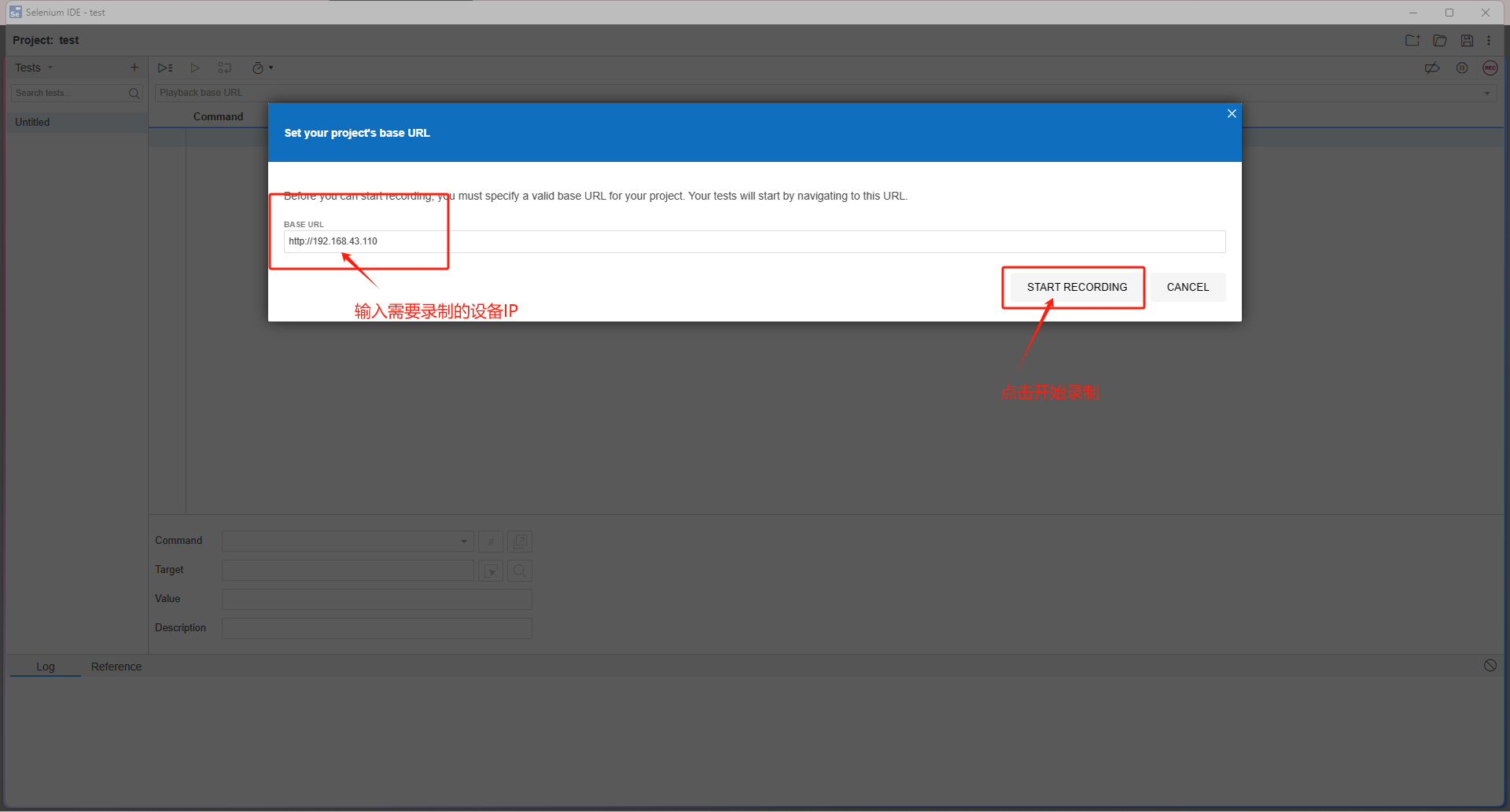
点击selenium IDE按钮，进入如下页面：



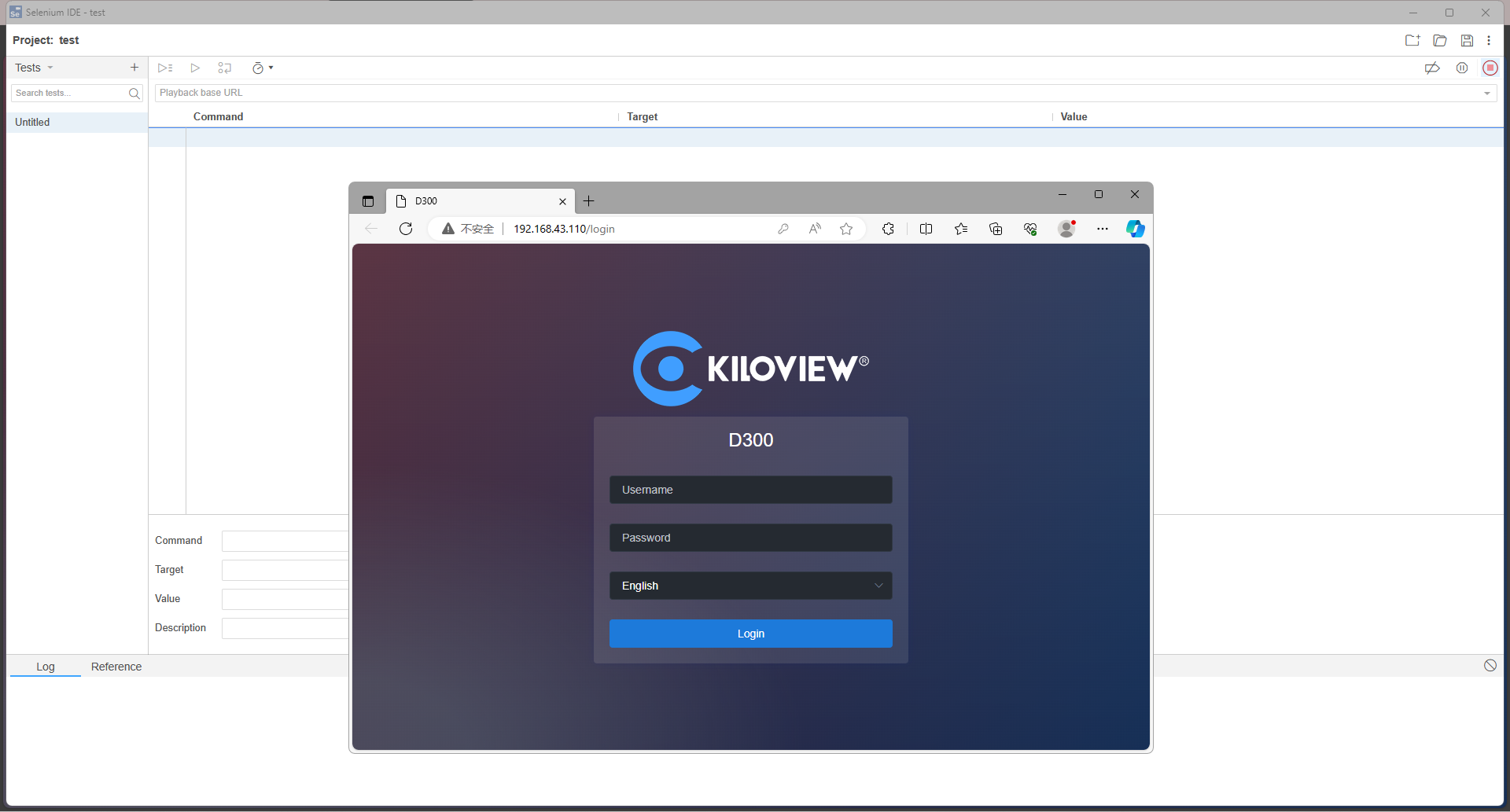
点击 create a new project​，输入项目名称后，可以创建一个测试项目。如下所示：



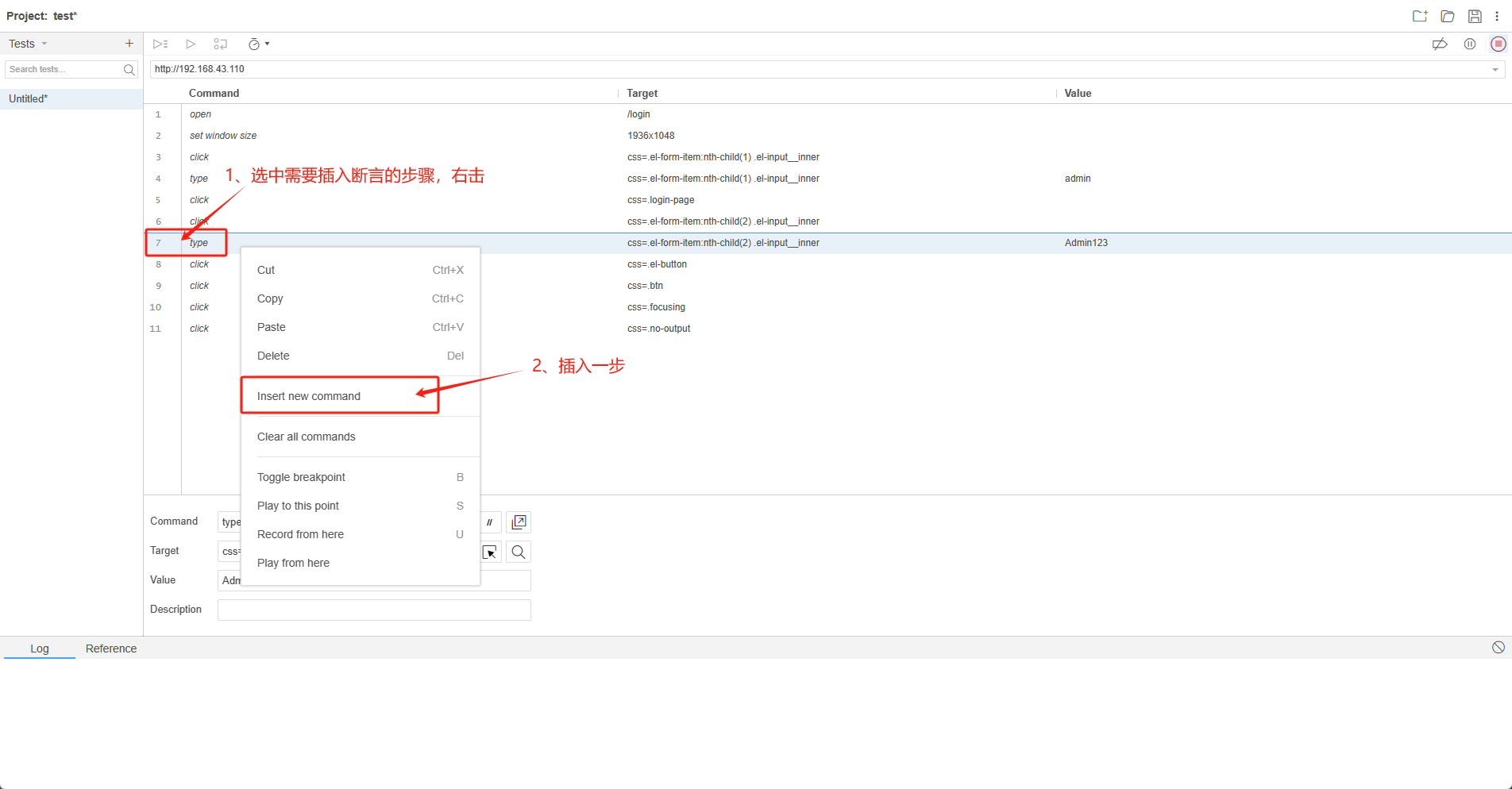
点击右上角REC按钮，输入IP，再点击进行录制。如下所示：

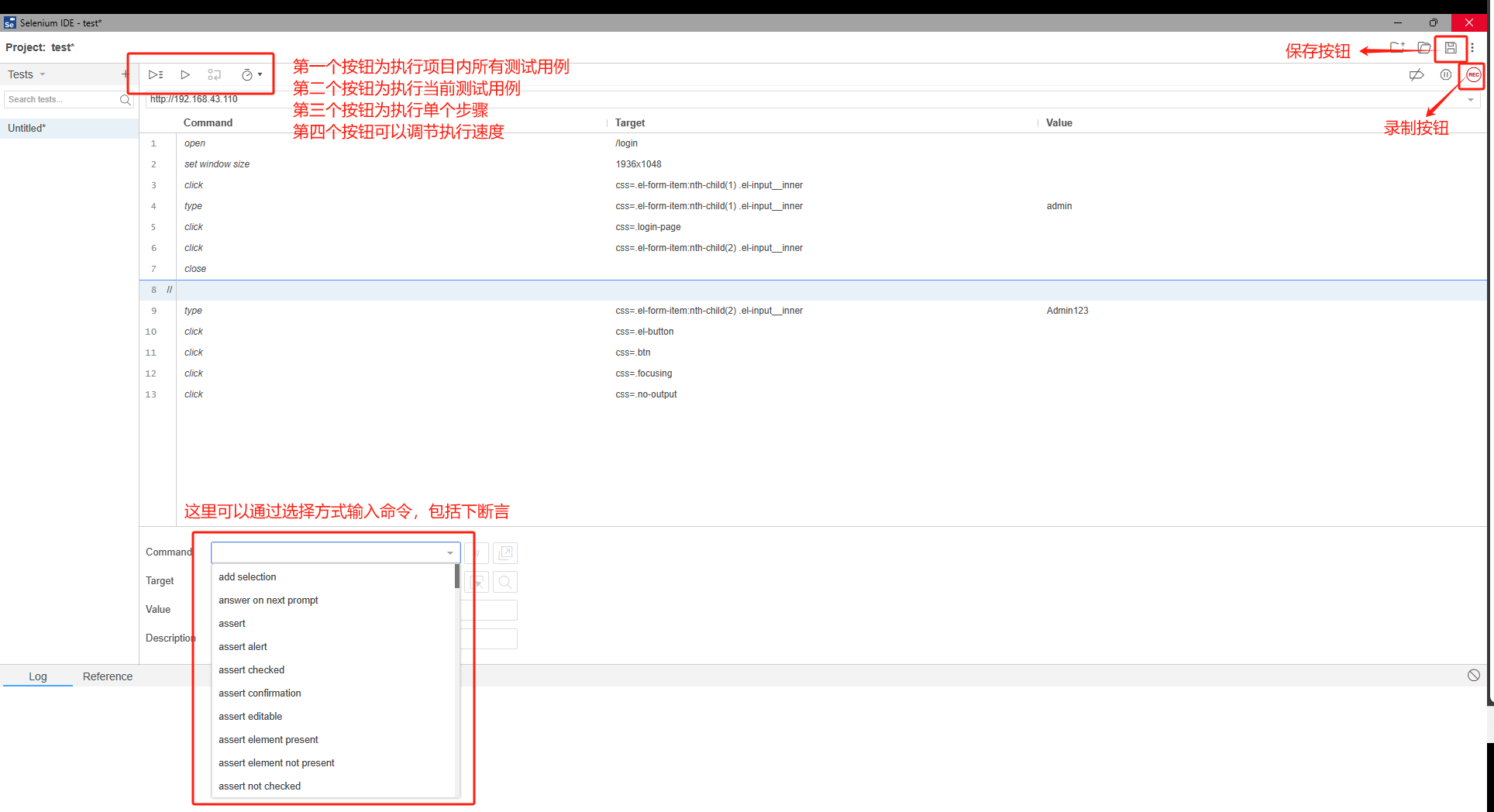


在进行录制的时候，seleniumIDE会打开一个新窗口，点击停止录制该窗口也不会自动关闭，需要手动关闭；若结束录制后想下断言时，可以使用该界面定位元素位置后再手动关闭窗口。如下所示：：



录制中途也可插入一些断言，也可录制结束后插入。如下所示：





选择命令操作如下：



CSS：

使用语法简单，易于理解和编写。

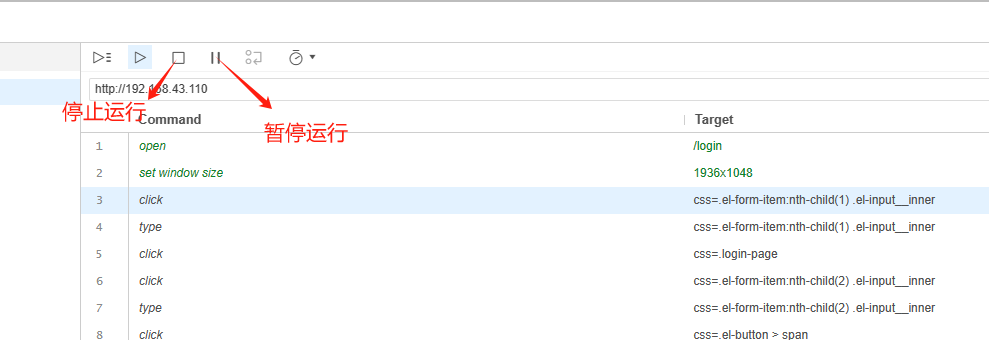
可以根据元素的标签名、类名、ID、属性等进行选择。

XPath：

是一种更加灵活和强大的定位方法，可以通过元素的层级关系、属性、文本内容等进行选择。

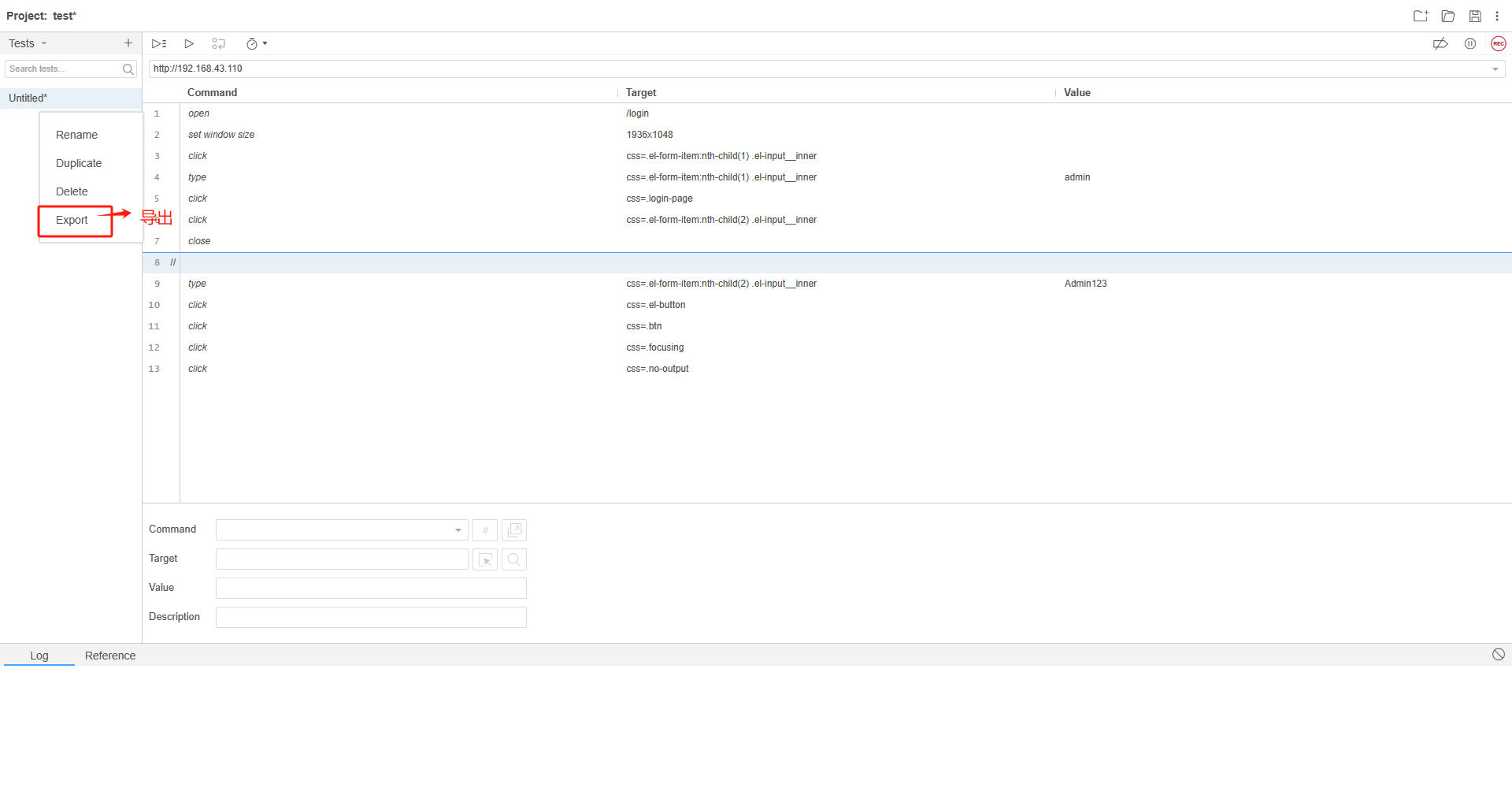
可以使用绝对路径或相对路径来定位元素。

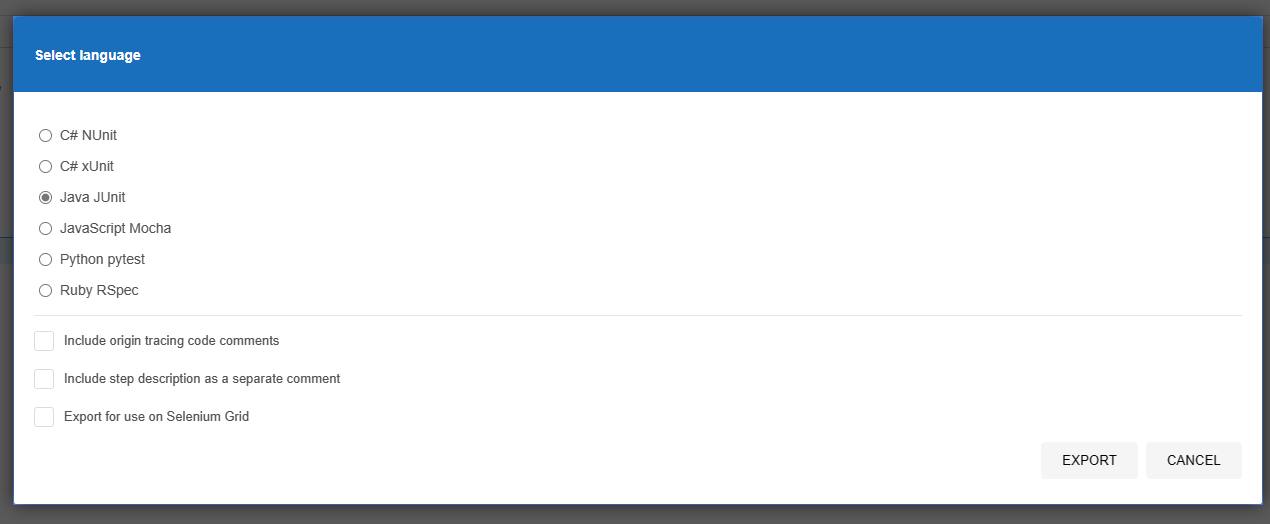
停止运行，如下所示：



点击步骤右侧可设置运行时的断点，运行到该步骤时停止，需手动点击下一步。

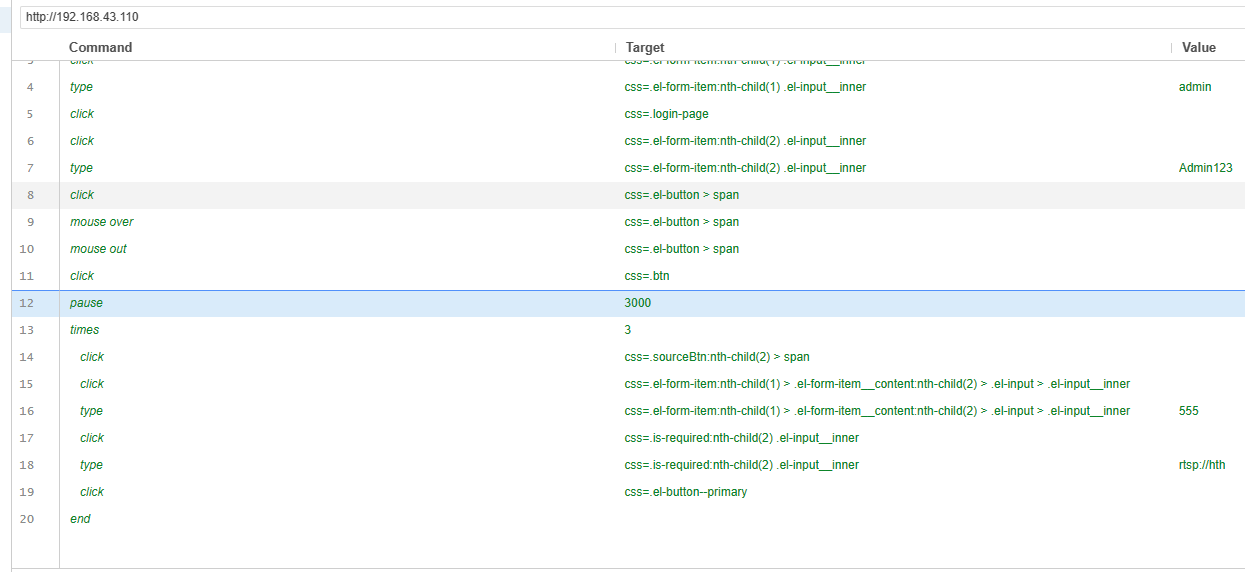
选中用例，右击后点击Export按钮，导出用例。如下所示：

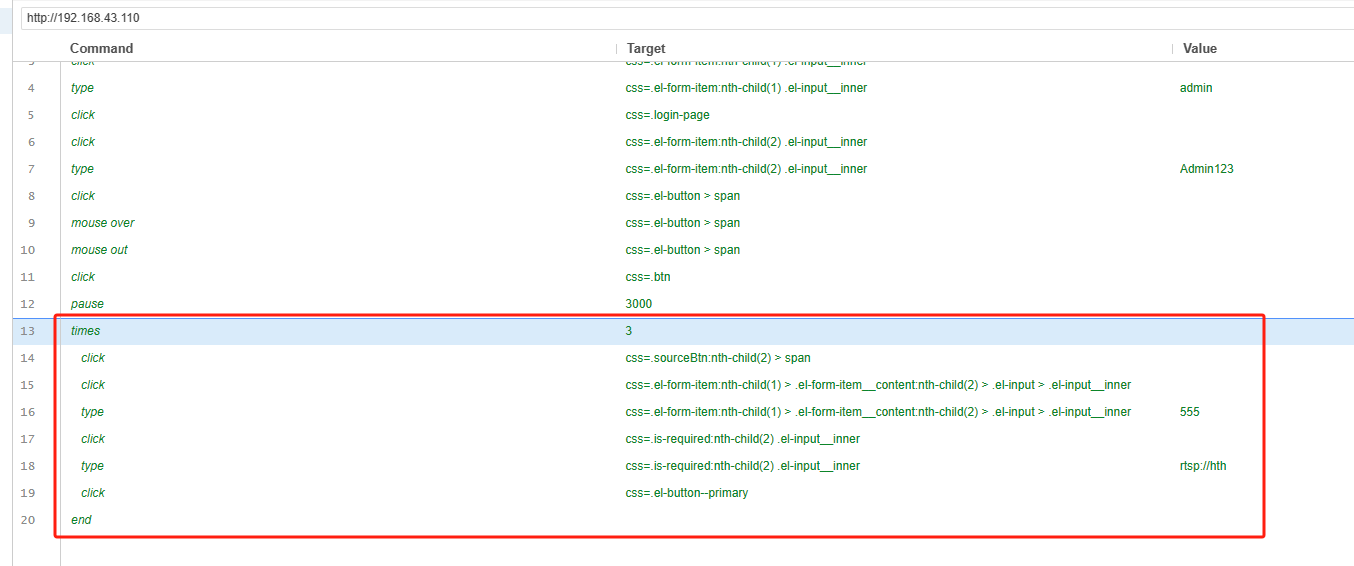
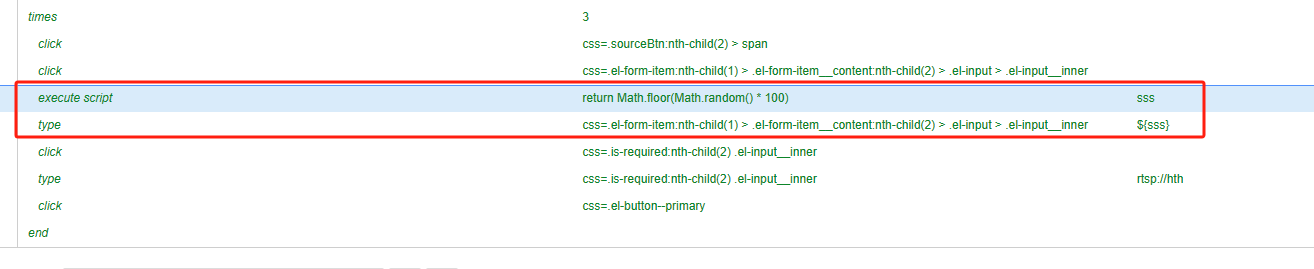
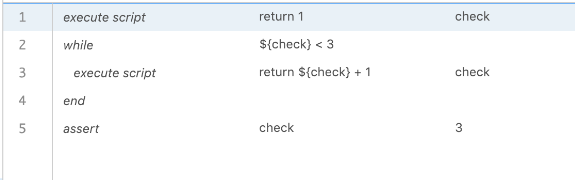
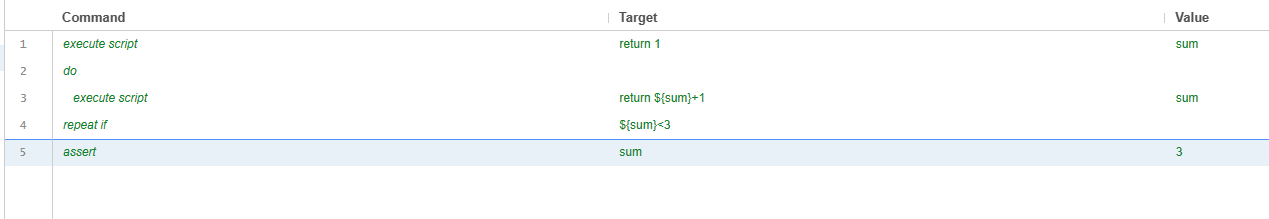




## **常用断言**

* pause 等待指定的时间（以毫秒为单位）



* assert editable 确认目标元素是可编辑的。如果断言失败，则测试将停止。
* assert not editable 确认目标元素不可编辑。如果断言失败，则测试将停止。
* assert element present assert 确认目标元素存在于页面上的某处。如果断言失败，则测试将停止。
* assert element not present 确认目标元素不在页面上任何地方。如果断言失败，则测试将停止。
* times 创建一个循环执行n次执行命令。循环次数默认值设置为1000。
* 生成随100以内的随机数
* verify editable 软断言指定的输入元素是否可编辑。即使验证失败，测试也将继续。
* verify not editable 软断言指定的输入元素是否不可编辑。即使验证失败，测试也将继续。
* verify element present 软断言指定的元素在页面上的某处。即使验证失败，测试也将继续。
* verify element not present 软断言指定的元素不在页面上。即使验证失败，测试也将继续。
* wait for element editable 等待元素可编辑。
* wait for element not editable 等待元素不可编辑
* wait for element present 等待元素出现
* wait for element not present 等待元素不存在
* wait for element visible 等待元素可见
* wait for element not visible 等待元素不可见
* execute script 在当前选定的框架或窗口的上下文中执行一段JavaScript。脚本片段将作为匿名函数的主体执行。要存储返回值，使用“ return”关键字，并在值输入字段中提供一个变量名称。
* for each 创建一个循环，为给定集合中的每个项目执行命令。