# 说明文档

## 一：简介

本组件是根据selenium源码进行封装的，selenium是web自从测试的轻量级框架，简单的说，selenium是一个web引用功能测试工具，根据网页中每个控件的ID或者xpath进行定位，能够帮助测试人员解决繁琐的重复性操作

## 二：组件特点

开源、免费

支持多浏览器。如Firefox、Chrome、IE、Opera、Safari

语言支持：Java

平台支持：windows

## 三：组件的适用范围和方向

范围：对于某些页面要经常重复、定时的操作

方向：web自动化测试

该组件目前在《中国移动支撑网自动化测试服务平台》上应用

## 四：组件功能

### 1：com.changeman.eliteengine.component.RCommand 下的主要组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| wait | 无 | 无 | 等待时长，以秒为单位 |
| startBrowser | String browser | 浏览器对象，IE、火狐还是谷歌 | 打开浏览器 |
| String url | 目标地址 |
| closeAllBrowsers | 无 | 无 | 关闭浏览器 |
| pictureSetVerifyCode | String compAttr1 | 验证码输入框主要属性 | 输入从指定文件获取的图片验证码值 |
|  | String compAttr2 | 验证码输入框次要属性 |
| String verifyCodeExePath | 验证码解析工具一个(exe文件－verify.exe) |
| String verifyIMG\_path | 图片的路径 |
| String verifyCdsPath | 对应省份的字库文件 |
| smsSend | String phonesend | 发送手机号 | 短信发送 |
| String phone | 接收手机号 |
| String msmessage | 短信 |
| closePage | String attrOfClose | 关闭 | 页面关闭 |
| String index1 | 页面 |
| String attrOfSwitch | 跳转 |
| String index2 | 页面 |
| getSmsSubString | String phone | 接收号码 | 截取短信信息 |
| String prefixStr | 前缀 |
| String length | 截取长度 |
| String[] varAssign | 赋值变量 |
| manualVerificationCode | String imageName | 图片名称xpath | 手动验证码 |
| String[] varName | 存值参数 |
| getSmsCode | String phone | 手机号码 | 获取短信验证码 |
| String[] tempAssign | 赋值参数 |
| switchToPage | attribute | 属性 | 页面跳转 |
| index | 页面 |
| addSnapShot | 无 | 无 | 在当前页面截图，并保存截图到指定路径 |
| timeConsume | String startStep | 开始步骤 | 验证从某一步骤开始到某一步骤结束是否大于(小于)某个指定的值 |
| String endStep | 结束步骤 |
| String verifyType | 验证类型 值： 1为大于等于 2为小于 |
| String timeConsume | 期望耗时 毫秒 正整数 |
| List<EliteTestcaseStepRuntime> stepRuntimeListSouce | 当前步骤结果集 |
| int seqNo | 执行初始序号 |
| String testCaseName | 当前运行的用例名称 |
| int executeNum | 0 表示主用例,其他值表示被引用用例 |

### 2：com.changeman.eliteengine.component.Radio下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| click | 无 | 无 | 单击 |

### 3：com.changeman.eliteengine.component.RButton下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Click | 无 | 无 | 单击 |
| Exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |

### 4：com.changeman.eliteengine.component.RCheckBox下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Check | 无 | 无 | 选中多选框 |
| Uncheck | 无 | 无 | 去掉对当前多选框的选择 |
| Exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |
| isChecked | String prop | 对象属性 | 判断CheckBox是否为选中状态 |
| String value | 对象属性值 |
| isNotChecked | String prop | 对象属性 | 判断CheckBox是否不为选中状态 |
| String value | 对象属性值 |

### 5：com.changeman.eliteengine.component.RComboBox下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 功能说明 |
| Select | String sItem | 列表 | 选择列表中显示内容为sItem的一项 |
| getContents | 无 | 无 | 获取下拉列表中所有选项的内容 |
| doesItemExist | String sItem | 列表中某选项的显示内容 | 判断下拉列表中是否存在显示内容为sItem的项 |
| getItemIndex | String sItem | 显示内容 | 根据显示内容查找下拉列表中的选项 |
| getItemText | int index | 待查找的选项的序号 | 获取下拉列表中序号为index的选项的内容 |
| exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |
| doesNotEqual | String expectedValue | 期望值 | 判断CheckBox选中项的值是否与预期值不同 |

### 6：com.changeman.eliteengine.component.RDialogStatic下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | String propName | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |

### 7：com.changeman.eliteengine.component.RDiv下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Hover | 无 | 无 | 移动在上面 |
| Click | 无 | 无 | 单击 |
| doubleClick | 无 | 无 | 双击 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
|  | String expectedValue | String expectedValue |  |
|  | String typeStr | 验证方式 |  |
| doesNotExist | 无 | 无 |  |
| exists | 无 | 无 | 判断对象是否存在 |

### 8：com.changeman.eliteengine.component.RImage下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Click | 无 | 无 | 单击 |
| rightClick | 无 | 无 | 右击 |
| exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |

### 9：com.changeman.eliteengine.component.RLi下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Hover | 无 | 无 | 移动在上面 |
| Click | 无 | 无 | 单击 |
| doubleClick | 无 | 无 | 双击 |
| exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |

### 10：com.changeman.eliteengine.component.RLink下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Hover | 无 | 无 | 移动在上面 |
| Click | 无 | 无 | 单击 |
| Exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| doubleClick | 无 | 无 | 验证控件属性的值是否与期望值相同 |
| rightClick | 无 | 无 | 右击 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |

### 11：com.changeman.eliteengine.component.RRadio下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Click | 无 | 无 | 右击 |
| Exists | 无 | 无 | 判断是否存在 |
| doesNotExist | 无 | 无 | 判断是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |

### 12：com.changeman.eliteengine.component.RSpan下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Hover | 无 | 无 | 移动在上面 |
| Click | 无 | 无 | 单击 |
| doubleClick | 无 | 无 | 双击 |
| exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |

### 13：com.changeman.eliteengine.component.RTable下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| Click | 无 | 无 | 单击 |
| doubleClick | 无 | 无 | 双击 |
| buttonClick | String rowStr | 行 | 单元格中Button对象的点击方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | 控件辅助属性键值对 |
| hrefClick | String rowStr | 行 | 单元格中Link对象的点击方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | String secondAttrs |
| radioClick | String rowStr | 行 | 单元格中Radio对象的点击方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | 控件辅助属性键值对 |
| textSetText | String rowStr | 行 | 单元格中TextFiled对象的输入方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | 控件辅助属性键值对 |
| String value | String value |
| comboxSelect | String rowStr | 行 | 单元格中下拉框的选择方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | 控件辅助属性键值对 |
| String option | String option |
| checkBoxCheck | String rowStr | 行 | 单元格中CheckBox对象的check方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | 控件辅助属性键值对 |
| checkBoxUncheck | String rowStr | 行 | 单元格中CheckBox对象的uncheck方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | 控件辅助属性键值对 |
| check | String rowStr | 行 | 单元格中CheckBox对象的点击方法 |
| String columnStr | 列 |
| String mainAttr | 控件主要属性键值对 |
| String secondAttrs | 控件辅助属性键值对 |
| exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |
| getText | String row | 行数 | 获取第row行，第column列的内容 |
| String column | 列数 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |
| checkRowAndColumn | String rowStr | 行数 | 检测表格中的行和列是否有效 |
| String columnStr | 列数 |

### 14：com.changeman.eliteengine.component.RTextArea下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| setText | String s | 数值 | 为文本框赋值 |
| getText | 无 | 无 | 获取文本框中的值 |
| exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |
| assign | String propName | 属性名称 | 将指定属性的值赋值到varName中 |
| String[] varName | 存值参数 |
| String prefixStr | 前缀 |
| String length | 长度 |
| String suffixStr | 后缀 |

### 15: com.changeman.eliteengine.component.RTextField下的组件

|  |  |  |  |
| --- | --- | --- | --- |
| 组件 | 参数 | 参数说明 | 组件功能 |
| setText | String s | 数值 | 为文本框赋值 |
| getText | 无 | 无 | 获取文本中的数值 |
| exists | 无 | 无 | 判断对象是否存在 |
| doesNotExist | 无 | 无 | 判断对象是否不存在 |
| verify | String propName | 属性名 | 验证控件属性的值是否与期望值相同 |
| String expectedValue | 期望值 |
| String typeStr | 验证方式 |
| click | 无 | 无 | 单击 |
| assign | String propName | 属性名称 | 将指定属性的值赋值到varName中 |
| String[] varName | 存值参数 |
| String prefixStr | 前缀 |
| String length | 截取长度 |
| String suffixStr | 后缀 |
| clear | 无 | 无 | 清除文本框中的数值 |

## 五：运行环境

系统：Windows7、2008server、10，

内存：4G以上

磁盘：80G以上

网卡：100M/1000M自适应

JDK: jdk1.7.0\_80

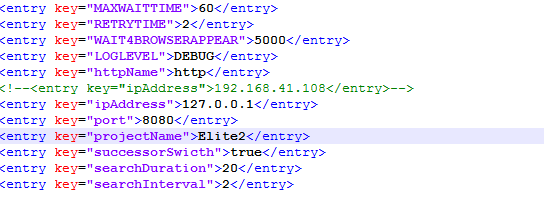
浏览器：IE 9,firefox 47,chrome 39,opera 39，safari

缓存服务器：memcached

语言：java

## 六：组件安装

改组件无需安装，只要把压缩包（SeleniumEngine.zip）解压到您某个盘符下（如E盘），首先配置文件E:\SeleniumEngine\engine\config.xml需要配置的文件如下：





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 引擎端config.xml | | | | | |
| 属性 | 默认值 | 备注 | | 是否需要修改 | |
| MAXWAITTIME | 60 | 最大等待时间 | | 否 | |
| RETRYTIME | 2 | 重试时间 | | 否 | |
| WAIT4BROWSERAPPEAR | 5000 |  | |  | |
| LOGLEVEL | INFO |  | |  | |
| <!--REST service--> | | | | | |
| httpName | http | 协议名称 | | | 否 |
| ipAddress | 192.168.66.101当前应用服务IP地址 | 当前应用服务IP地址(既回调地址) | | | 是 |
| port | 8085 | 端口号 | | | 是 |
| projectName | 为空 | 项目名 | | | 是 |
| <!--search args--> | | | | | |
| successorSwicth | true | 是否全局查找 | | |  |
| searchDuration | 20 | 查找次数 | | |  |
| searchInterval | 2 | 查找间隔时间 | | |  |
| <!--SVN Server--> | | | | | |
| svnProtocol | svn | svn名称 | | | 否 |
| svnIpAddress | 192.168.66.105 | svn服器的ip地址 | | | 是 |
| svnPort | 80 | 端口号 | | | 是 |
| svnUserName | svnAdmin | svn管理员 | | | 是 |
| svnPassword | ichangeman123 | svn管理员密码 | | | 是 |
| <!--Engine--> | | | | | |
| EngineWebName | EngineWebApp | Rest服务名 | | | 否 |
| EngineJarName | EliteEngine | 引擎jar包名 | | | 否 |
| EngineLocalIp | 127.0.0.1 | 引擎本机IP | | | 否 |
| EnginePort | 8085 | 引擎端口 | | | 是 |
| EngineClosePort | 8086 | 引擎升级端口 | | | 否 |
| applicationVersionFilePath | /ATElite/engineVersion | 平台版本文件路径 | | | 否 |
| applicationVersionFileName | version.txt | 平台版本文件名 | | | 否 |
| <!--TestTools--> | | | | | |
| TestToolTypeForAnt | rft | |  | | 是 |
| TestTools | Selenium | | 引擎工具 | | 是 |
| taskRecordFile | C:/ATElite/resource/task.xml | | 引擎任务记录文件 | | 否 |
| scriptPath | C:/ATElite/resource | | 脚本副本文件位置 | | 否 |
| imageScanSleepTime | 5 | |  | | 否 |
| scriptSubPathForSms | /Script/sms | |  | | 否 |
| scriptSubPath | /Script/selenium | |  | | 否 |
| <!--Driver--> | | | | | |
| default\_type | chrome | 默认浏览器 | | | 是 |
| IE\_Driver | E:/SeleniumEngine/engine/IEDriverServer.exe | IE浏览器 | | | 是 |
| Chrome\_Driver | E:/SeleniumEngine/engine/chromedriver.exe | 谷歌浏览器 | | | 是 |
| Opera\_Driver | E:/SeleniumEngine/engine/operadriver.exe | 欧朋浏览器 | | | 是 |

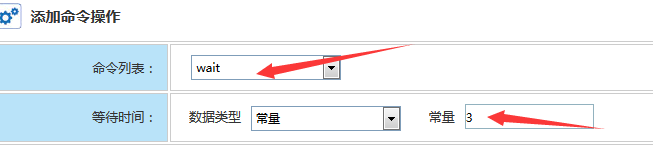
使用时需要下载指定的浏览器（因为浏览器驱动对应浏览器版本），版本在运行环境中有介绍，否则改组件将无法打开浏览器执行，运行时缓存系统也必须启动，建议设置为开机启动

## 七：用法

首先选择打开浏览器的命令，输入百度地址，如图所示



在打开浏览器后，可能需要等待时间，我们可以选这wait来实现，单位秒

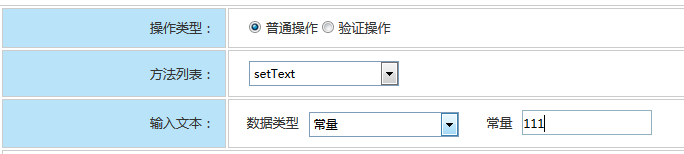


2：选择按钮的点击操作,我这里分多步实现的

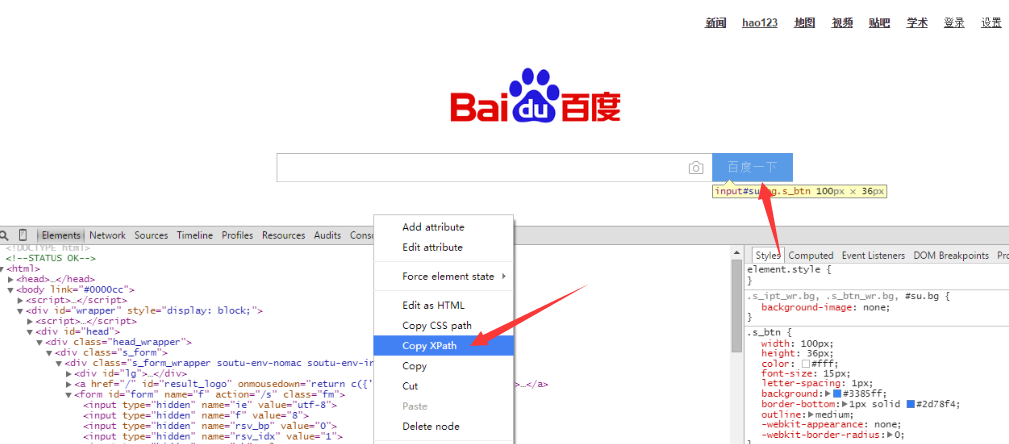
百度输入框的建立，也是先获取xpath或者ID



然后选择setText，表示向文本框赋值，赋值内容111（填你想填的数值就可以了）



第三，建立百度一下按钮的事件，填写xpath或者id，得到xpath的方法如下所示，你也可以自己网上查找怎么获取



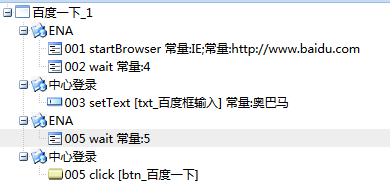


然后添加按钮的click事件，这样就相当于点击了百度一下那个按钮



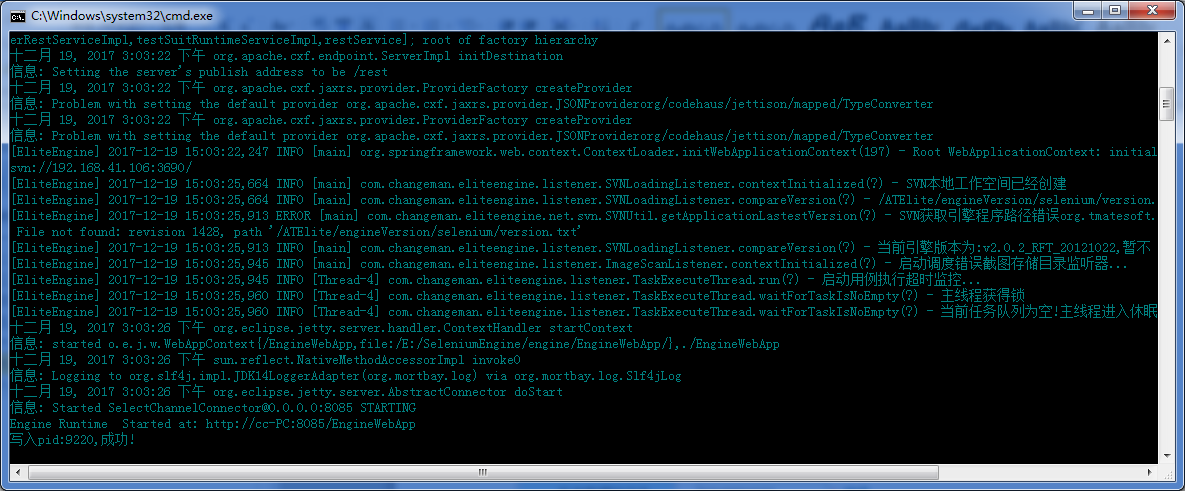
把所有的步骤按照类似的方式建立完成以后，会得到如下这样的步骤列表

它的意思很明显，首先打开浏览器—>等待4秒->然后在百度输入框中输入奥巴马-》然后等待5秒-》点击百度一下，这样一套流程



流程建立完成以后，下面就可以执行了，引擎就是该组件，必须先启动

如下所示：



如下所示就是开始运行用例的界面，你的也许不一样，但是引擎一定要有



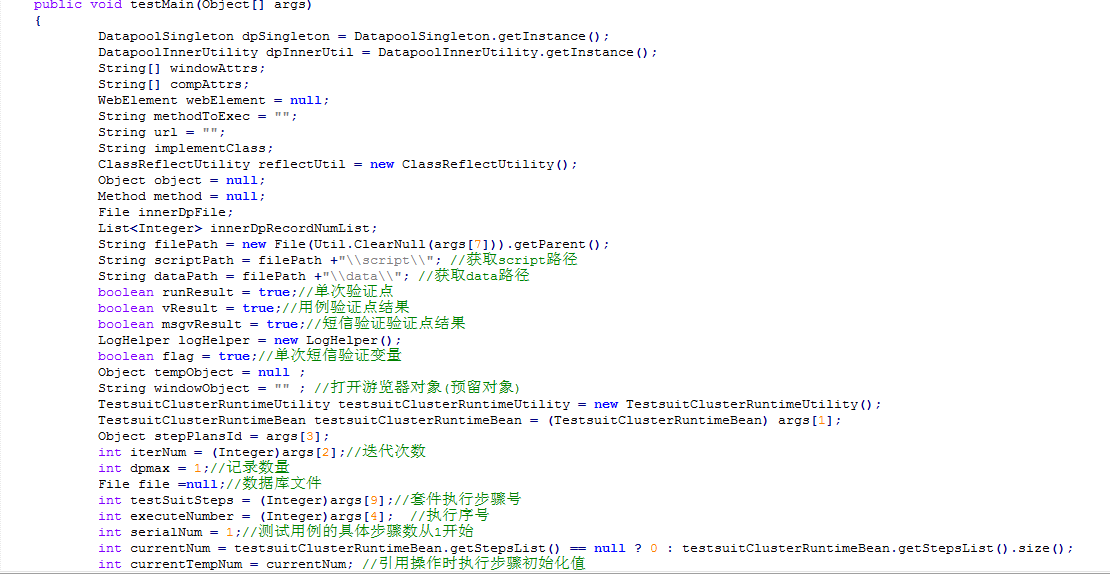
这样点击运行就可以看见浏览器会自动打开，并且会按照你写的步骤一步步执行，如果打开的浏览器是在你本地机器上的话，这期间请不要随意移动鼠标，随意移动会导致鼠标指针的轨迹发生错乱，从而导致不能正常执行完毕

## 八：代码结构

第一部分：拼接java代码，组装成一段引擎可以识别的脚本文件，类似下面这样，文件



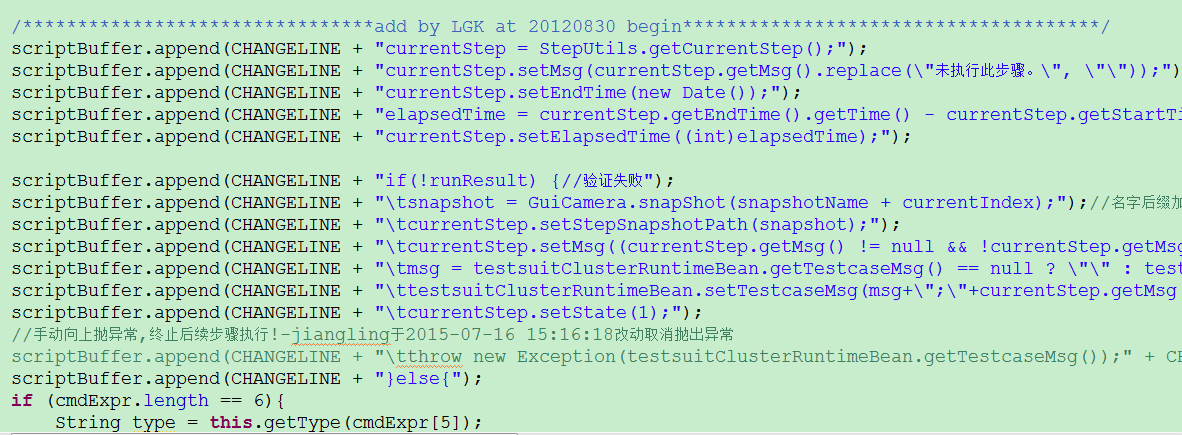
下面是代码片段：



首先代码中要拼凑一些列你需要导入到的包，这些包是引擎中必须具备的



接下来拼凑一系列的执行程序代码，比如，这只是其中一部分，全部代码就不贴出来了



最后拼凑的文件，在代码执行的时候就要保存到你的某个位置，我这边保存的地址是：E:\ATElite\resource\tenant3\project20\script,

拼凑以后可以看到：

tempStep = new EliteTestcaseStepRuntime();

params = new String[]{"[EliteCommand], [命令操作,属性:无]","1","启动浏览器","IE","百度一下\_1",String.valueOf(executeNumber > 0 ? executeNumber : serialNum)};

this.setTempStep(tempStep,testsuitClusterRuntimeBean,params,serialNum++,iterNum,stepPlansId); tempStep = new EliteTestcaseStepRuntime();

params = new String[]{"[EliteCommand], [命令操作,属性:无]","2","指定等待时间","4","百度一下\_1",String.valueOf(executeNumber > 0 ? executeNumber : serialNum)};

this.setTempStep(tempStep,testsuitClusterRuntimeBean,params,serialNum++,iterNum,stepPlansId)

tempStep = new EliteTestcaseStepRuntime();

params = new String[]{"[Text],[属性:name=wd ]","3","文本框赋值","奥巴马","百度一下\_1",String.valueOf(executeNumber > 0 ? executeNumber : serialNum)};

this.setTempStep(tempStep,testsuitClusterRuntimeBean,params,serialNum++,iterNum,stepPlansId;

tempStep = new EliteTestcaseStepRuntime();

params = new String[]{"[EliteCommand], [命令操作,属性:无]","4","指定等待时间","5","百度一下\_1",String.valueOf(executeNumber > 0 ? executeNumber : serialNum)};

this.setTempStep(tempStep,testsuitClusterRuntimeBean,params,serialNum++,iterNum,stepPlansId;

tempStep = new EliteTestcaseStepRuntime();

params = new String[]{"[Button],[属性:id=su ]","5","点击","无","百度一下\_1",String.valueOf(executeNumber > 0 ? executeNumber : serialNum)};

this.setTempStep(tempStep,testsuitClusterRuntimeBean,params,serialNum++,iterNum,stepPlansId);

这里是你运行测试用例的步骤，但在这个时候用例是还没有运行的，EliteTestcaseStepRuntime（）：是引擎当中的一个类，主要是测试用例运行步骤的记录，

Params:存储的是参数，以数组形式保存的，这里不是死的，是在拼凑代码的时候从参数中获取的，接下来看看下面这段代码，它是运行测试用例步骤的代码实现

currentIndex = currentNum++;

currentStep = testsuitClusterRuntimeBean.getStepsList().get(currentIndex);

currentStep.setStartTime(new Date());

currentStep.setData("http://www.baidu.com");

StepUtils.setCurrentStep(currentStep);

implementClass = "com.changeman.eliteengine.component.selenium.common.SCommand";

methodToExec = "startBrowser";

object = SCommand.getInstance();

method = reflectUtil.getMethodByClassName(implementClass,

methodToExec, new Class[]{String.class, String.class});

method.invoke(object, new Object[]{"IE","http://www.baidu.com" });

上面的一些列都不用管，都是传递参数，com.changeman.eliteengine.component.selenium.common.SCommand这个不能写错，写错了就找不到对应的方法了，可以按照组件功能来查找对应的方法。

methodToExec：方法名称，你需要执行什么样的方法，这里就填写什么方法，执行到method.Invoke（）时就是开始运行测试用例步骤了，我们可以看到，这里有{"IE","http://www.baidu.com"}，这就是代表用IE浏览器打开百度

第二部分，接口调用

在我们这套组件当中，每当执行用例步骤的时候，步骤执行的每一步过程都会返回给项目平台上，让测试人员很清楚的看到执行过程，那么这样一个过程是怎么实现的呢，

先看看下面这段代码，这是一个接口实现

**public** **boolean** offerTaskFromRemote(String entity) {

// **TODO** Auto-generated method stub

*log*.info("任务开始入列...");

**return** TaskExecuteThread.*offerTask*(entity);//任务入列

}

这是一个任务入列的代码，在你的项目一方，通过以下代码

AdaptorHttpClient connection = **new** AdaptorHttpClient("http", ip, **new** Integer(port));

//发送HTTP请求

AdaptorRestMethod.*executePOSTMethod*(connection.getServerURL()

+ engineName

+ "/services/rest/TestSuitRuntime/offerTaskByXML/args",

调用引擎端接口，engineName是引擎名称（你自己定义的），这是初始把执行脚本发送给引擎端执行，其他接口类似的调用

## 九：更多帮助

<http://www.seleniumhq.org/> selenium官网地址

## 十：联系方式

QQ：279971523 陈生