UI自动化测试框架部署及使用

|  |  |  |  |
| --- | --- | --- | --- |
| **文件版本：** | **V1.0** | **文件编号：** | UI自动化测试框架部署及使用 |
| **首版发布日期：** | **2016.11.7** | **编 制：** | **吴丁飞** |
| **审 核：** |  | **批 准：** |  |

**铭师堂教育 版权所有**

**内部资料 注意保密**

修订记录：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本** | **修改内容** | **修订日期** | **编写人** | **审核人** | **批准人** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目 录

[1. 编写目的 1](#_Toc466299860)

[2. 环境准备 1](#_Toc466299861)

[**2.1.** **安装node.js** 1](#_Toc466299862)

[**2.2.** **安装、配置Android sdk环境** 2](#_Toc466299863)

[**2.3.** **安装手机驱动并测试连接真机** 2](#_Toc466299864)

[**2.4.** **安装Appium** 3](#_Toc466299865)

[**2.5.** **Ant安装及配置** 3](#_Toc466299866)

[3. 框架目录结构说明 4](#_Toc466299867)

[**3.1.** **源码src** 4](#_Toc466299868)

[**3.2.** **config** 5](#_Toc466299869)

[**3.3.** **doc** 5](#_Toc466299870)

[**3.4.** **lib** 5](#_Toc466299871)

[**3.5.** **res** 5](#_Toc466299872)

[**3.6.** **build.xml** 6](#_Toc466299873)

[4. 框架使用 7](#_Toc466299874)

[**4.1.** **测试用例填写说明** 7](#_Toc466299875)

[**4.2.** **用例填写实例** 7](#_Toc466299876)

[**4.3.** **页面元素定位实例** 7](#_Toc466299877)

[**4.4.** **生成测试类** 8](#_Toc466299878)

[**4.5.** **脚本运行** 8](#_Toc466299879)

# **编写目的**

为了方便大家能够顺利的搭建好UI自动化测试环境，并较好的使用UI自动化测试框架进行自动化测试脚本开发，提高自动化工作效率，特编写此《UI自动化测试框架部署及使用》。通过对环境搭建、框架使用的说明，并配合使用《UI自动化测试框架--元素定位》,可以较好的使用框架进行脚本开发。以规避相关容易犯错的地方，及出现问题时可以找到相关帮助，使自动化测试工作效率得以提升。

本文的阅读针对自动化测试人员，在工作过程中如有其他相关问题未在本文中详尽说明，也请大家提出宝贵的意见，促进文档的完善，以便大家更好的利用该文档及框架。

# **环境准备**

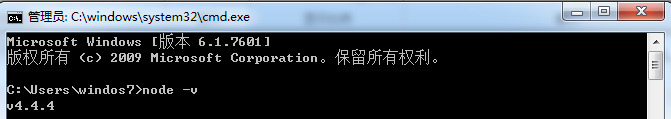
## **安装node.js**

　1、到中文官网下载node.js：http://nodejs.cn/download/



　　2、获取到安装文件后，直接双击安装文件，根据程序的提示，完成nodejs的安装。

　　3、安装完成后，运行cmd，输入node -v，如果安装成功，会输出如下版本信息：



## **安装、配置Android sdk环境**

1、JDK安装及环境配置：(安装步骤略)

java环境变量：

JAVA\_HOME：

D:\Program Files\Java\jdk1.8.0\_31 （JDK安装目录）

Path：

%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin;

CLASSPATH：

.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;

2、到网站<http://adt.android-studio.org/>下载相应平台、版本的ADT，下载完成后解压即可(目前公司环境可以使用我提供的相应ADT,已集成testng、svn相关插件)。

3、配置SDK环境

　　 3.1、确保安装了 Level 17 或以上的版本 api

　　 3.2、设置 ANDROID\_HOME 系统变量为你的 Android SDK 路径

3.3、把tools和platform-tools两个目录加入到系统的 Path 路径里

SDK环境变量：

ANDROID\_HOME：

D:\Program Files\adt\sdk （SDK安装目录）

Path:

%ANDROID\_HOME%\tools;%ANDROID\_HOME%\platform-tools;

## **安装手机驱动并测试连接真机**

完成上述步骤以后，为了能够让手机连接到PC端进行真机测试。还需要安装测试手机对应的驱动程序。将手机与PC通过usb线相连，通常开启usb调试即可自动安装驱动程序。在cmd中输入以下命令，如果能够看到设备，则表示安装成功。



## **安装Appium**

　　1、方法一：（需要联网，推荐使用方法二）

　　使用npm安装appium。 在cmd使用命令npm install –g appium 下在appium（整个过程较慢，请耐心等待）安装成功后，在cmd输入appium出现以下信息表明安装成功。　　

　　2、方法二：

　　　1）下载安装文件：http://appium.io/

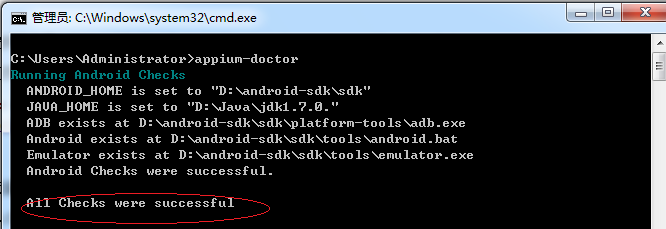
　　　2）直接双击appium-installer.exe文件安装就好

3、安装后配置appium环境变量

Path     D:\tools\appium\Appium\node\_modules\.bin;

4、检查appium所需的环境是否OK：

进入cmd命令行，输入appium-doctor ，出现以下提示，All Checks were successful ,说明环境成功。



Tips**：**Appium是用.net开发的，所以安装之前需要安装.net framework，否则安装过程中会报错。

## **Ant安装及配置**

1、在官网：<http://ant.apache.org/bindownload.cgi>下载相应版本的ant，下载后解压到指定目录。

2、配置环境变量

ANT\_HOME：

D:\Program Files\apache-ant-1.9.4

Path：

%ANT\_HOME%\bin;

# **框架目录结构说明**

## **源码src**

1、com.mistong.base

该包下的BaseParpare类是自动化测试基类，只要是现实测试准备、销毁等相关功能，供测试类继承。主要方法有：

beforeTest() : 该方法使用@BeforeTest注解，以便用于在测试之前进行日志初始化，以动态生成各个模块中的每条用例的日志，运行完成用例之后可以到result/log目录下查看，以便结果分析。

startTest((ITestContext context)): 主要为读取config/driver.properties中的配置信息，根据配置信息分别实现WebDriver、AndroidDriver的实例。该方法使用@ BeforeClass注解，以实现在运行每个测试类的时候，都能够去创建相应的WebDriver、AndroidDriver实例。

endTest():该方法使用@AfterClass注解，以便在每个测试类运行完后都会自动销毁WebDriver、AndroidDriver实例。

AfterTest():该方法使用@AfterTest注解，主要是为了解决android测试后appium进程的关闭。

2、com.mistong.testcases包

该包下的模块包及测试类由工具类CreateTestCase.java来自动生成。

3、com.mistong.util包

该包为工具类包。主要是日志、数据库操作、属性获取、selenium API、关键字驱动、测试类生成等相关功能。

DBHelper.java:该类为数据库操作类，实现数据库的增、删、改、查操作。

LogConfiguration.java:该类为日志初始化类。

PropertiesDataProvider.java：该类实现从.properties文件中读取相关测试数据。

SeleniumUtil.java:封装selenium API，并同时支持web、android。

KeyWordDriver.java:关键字驱动类。

CreateTestCase.java:用于测试类自动生成，根据模块名、功能名(excel名称)来自动生成测试用例脚本。

4、testng.xml

该xml中主要是配置需要测试的相关类，以便运行xml文件的时候去执行相应的测试类。

该类中还可以配置相应的监听，如自动生成自定义的报告等。目前还未实现该功能，后续会实现。

运行testng.xml文件测试结束后，会自动生成result、test-output目录。result记录的是运行每个测试类的日志，test-output是testng生成的测试报告。

## **config**

config目录下主要是相关的属性配置文件。

**dbconfig.properties:**

该配置文件中主要配置数据库的URL、用户名、密码。URL、用户名、密码配置需遵循一定的规则，以便进行数据库操作的时候正确获取URL、用户名、密码来执行数据库操作。

1、URL配置格式为:数据库名称\_url = jdbc:数据库类型://ip地址:端口/数据库名称,如:ikukocesi\_url = jdbc:mysql://10.0.0.233:3306/ikukocesi

2、用户名配置格式为:数据库名称\_user = 用户名,如:ikukocesi\_user = kaikelaTest

3、密码配置格式为:数据库名称\_ password = 密码,如: ikukocesi\_password = HaoHaoCe(234ABC

**driver.properties**

该配置文件中主要配置测试驱动相关信息。以便在BaseParpare基类中的startTest((ITestContext context))方法能够正确的调用配置信息生成相应的实例。

配置文件中的system参数为要测试的系统类型，可选项为web、android、ios。下面可分别对web、android、ios中相应的参数进行设置。每个参数都有详细说明。

## **doc**

该目录下存放的是框架的相关使用文档，有《UI自动化测试框架部署及使用》、《UI自动化测试框架--元素定位》。

## **lib**

该目录存放框架中需要依赖的jar包。

## **res**

该目录存放的是相关的资源文件。主要是selenium需要调用的IE浏览器驱动、chrome浏览器驱动及文件上传操作中需要使用到的exe可执行脚本文件、page下的元素定位的excel、testcase下的用例的相关测试步骤。

## **build.xml**

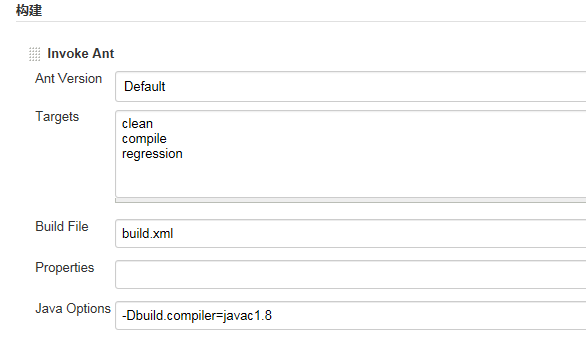
build.xml为使用ant方式运行脚本的配置文件。主要配置引用包设置、每次构建前需要做的操作、编译文件、运行指定的xml。

build.xml可以被jenkins调用，在jenkins中需要指定运行的target中的name。

build.xml配置文件示例：

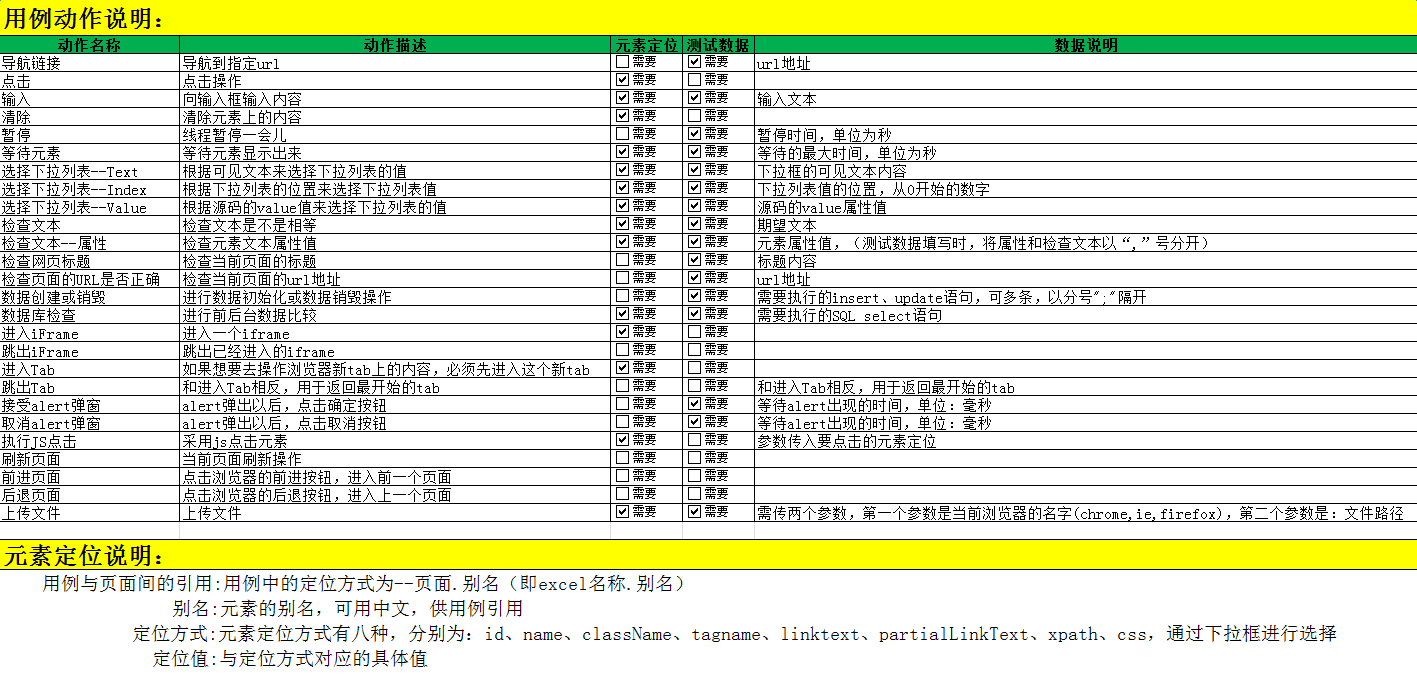


则在jenkins上需要指定clean、compile、regression三个target中的name



# **框架使用**

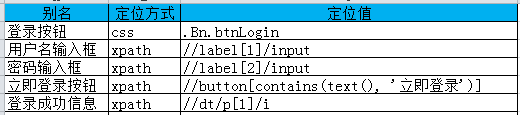
## **测试用例填写说明**



## **用例填写实例**



## **页面元素定位实例**



## **生成测试类**

1、在CreateTestCase.java类中的main方法中指定测试模块及测试用例res/testcase下面的excel名称。如下示例指定为login模块下的Longin文件。

**public static void main**(String[] args) {

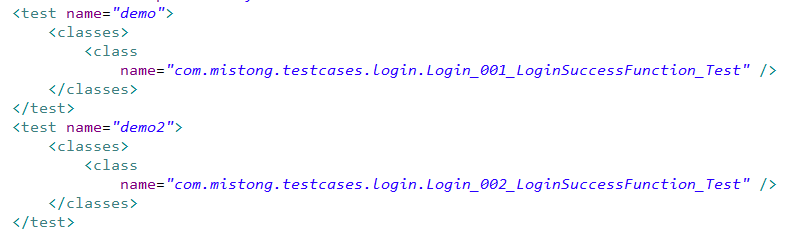
TestCaseFactory("login", "Login");

}

2、运行该类能够自动生成指定模块用例的测试代码。刷新src目录，在com.mistong.testcases包下可以看到生成的相应的模块包及下面的测试用例类。

## **脚本运行**

1、在testng.xml文件中配置需要运行的测试类，可添加多个。如下：



　 2、右键testng.xml，选择Run As>>TestNG Suite,即可运行testng下面配置的测试用例类。运行过程中可以看到控制台输出的测试日志信息，运行后会生成test-output、result目录。