

自动化脚本编写规范

目录

1 前言	3
2 自动化脚本规范.....	3
2.1 每个模块脚本的命名规范	3
2.2 变量声明和使用范围	3
2.3 局部全量命名规范	6
2.4 全局变量命名规范	6
2.5 被调用子脚本（功能脚本不单独执行）的命名规范	7
2.6 控件/图像/坐标识别使用的先后顺序和基本场景	7
2.7 测试数据（参数化数据源）准备规范	8
3 版本控制.....	9
3.1 应用版本控制和脚本迁移	9
3.2 脚本版本控制	9

1 前言

本文的目标是为自动化测试工程师编码规范统一、命名规则统一，确保自动化脚本的规范性与可阅读性，方便自动化测试脚本的管理。

2 自动化脚本规范

2.1 每个模块脚本的命名规范

每个模块脚本的命名规范可以与相对应每条测试用例（案例）建立联系。

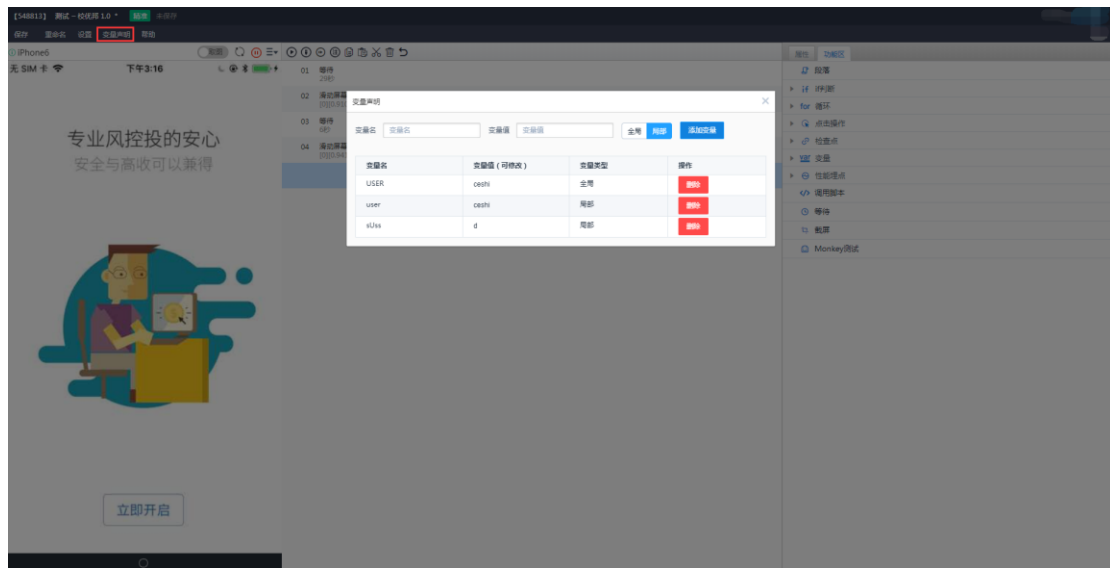
- 可以取每条测试用例的用例编号为脚本命名的开头，即脚本名的第一级
- 将待测系统的名称（即应用的名称）作为脚本名的第二级
- 将被测的功能模块作为脚本名的第三级
- 将被测的功能子模块作为脚本名的第四级
- 最后跟上测试描述（即该用例的检测点）作为脚本名的第五级
- 每级名称以下划线“_”间隔，示例如下：

A1000100_LIB 个人银行_转账汇款_境内转账_验证余额转账前后变化

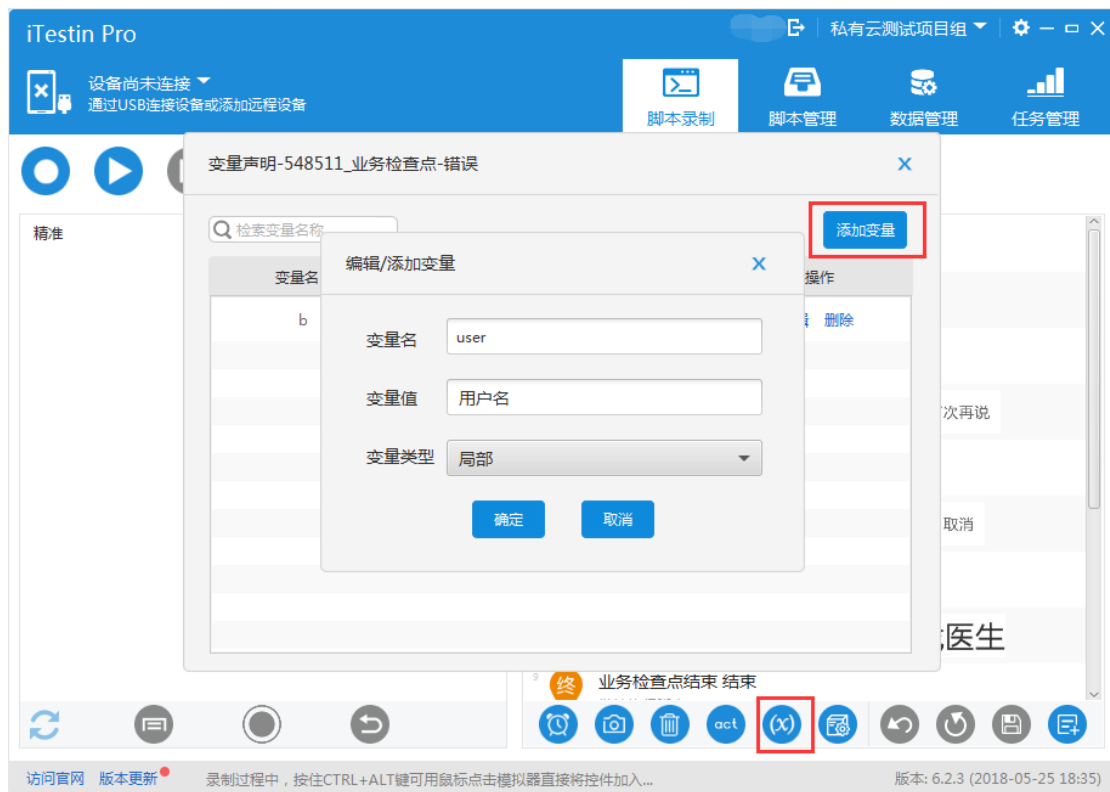
2.2 变量声明和使用范围

1. 变量声明：

通过录制工具的变量声明功能声明想要的变量名，全局变量名需求字母全部大写，局部变量名称需要首字母小写，声明后即可使用该变量。（如图一所示）

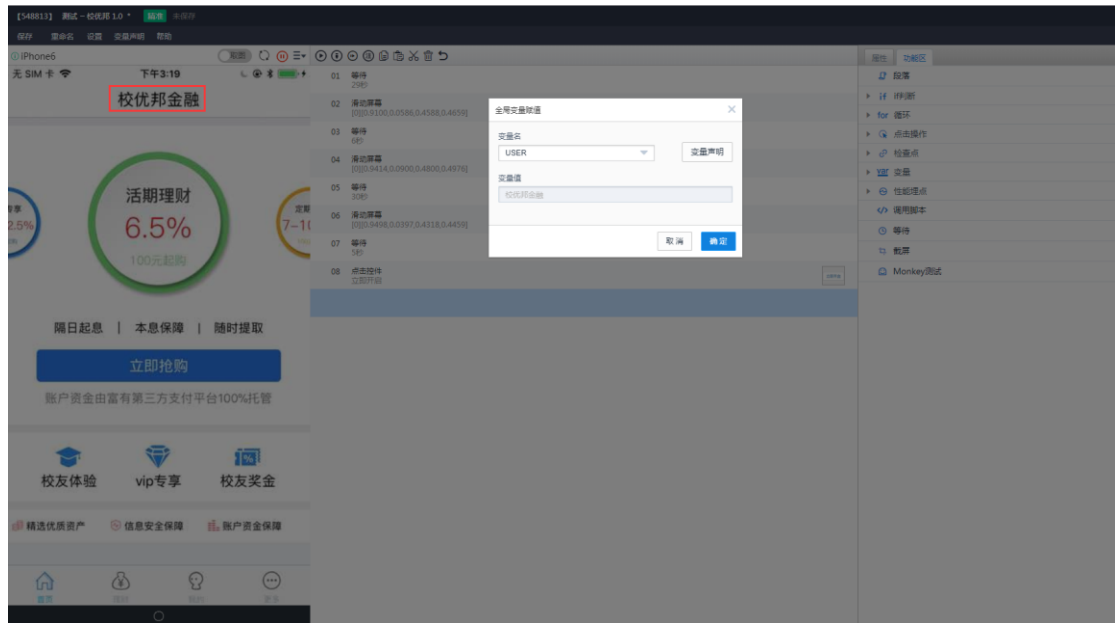


图一_Online



图一_客户端

如需在应用界面上获取文本信息赋值给变量，可使用变量赋值功能。（如图二所示）



图二_Online



图二_客户端

局部变量使用范围：只在当前脚本中生效，即只能在当前脚本中使用。

全局变量使用范围：可以在该应用的所有脚本中使用。

2.3 局部变量命名规范

- 局部变量的命名一定要有意义，能够让任意使用者拿到就能看懂这个变量的作用是什么，一般情况下就以该变量表示事物的英文单词命名即可。
- 同种类型的数据命名时（如都表示账户余额的数据），建议以尾数编号区分，示例如下：

起投金额，命名为 money1

实际金额，命名为 money2

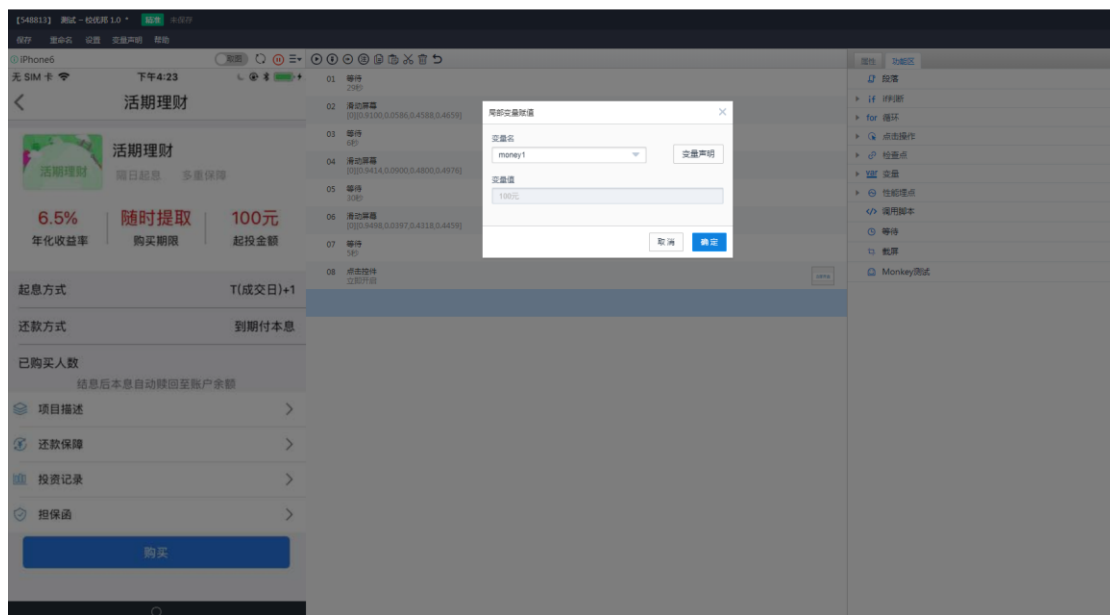


图 2.3

如图 2.3 所示为创建局部变量的过程，在映射屏中通过变量赋值的功能将起投金额的值 100 元赋给变量，只需给此变量起一个有意义的变量名即可（没有给变量加注释的功能，此变量对应的即是该步骤时创建的变量通过脚本步骤或变量名的意义即可快速明白该变量用途）。

2.4 全局变量命名规范

1. 全局变量的命名一定要有意义，能够让任意使用者拿到就能看懂这个变量的作用是什么，一般情况下就以该变量表示事物的英文单词命名即可。全局变量和局部变量的命名规则不同，脚本步骤里相关变量的操作都会显示变量名称，根据变量名称可轻松区分步骤中使用的是局部变量还是全局变量。

2. 如果变量有多个单词可以表示的情况，可以选用不同的单词表示不同情景下的变量。

3. 同种类型的数据命名时（如都表示登录账号的数据），建议以尾数编号区分，示例如下：

登录账号可能存在多个，为了实现账号的参数化，需要把每个账号都设置为全局变量。

账号 1，命名为 USER1

账号 2，命名为 USER2

2.5 被调用子脚本（功能脚本不单独执行）的命名规范

在脚本开发过程中为了满足一些脚本能够被多个脚本复用，即提高脚本的复用性，通常把这类脚本单独开发被称为子脚本，这类脚本经常被其他脚本调用使用，在调用他的脚本中为被调用子脚本，被调用子脚本一般分为两种情形：

第一种为被调用子脚本就是我们开发完成的脚本中的一条或几条，有时为了使多个脚本构成一个场景，需要脚本间的相互调用，即被调用的脚本就成了子脚本，该种子脚本的命名规范就和 2.1 章节说到的模块脚本的命名规范一致了。

第二种被调用子模块是为了满足某种逻辑多写出来的脚本，该脚本不属于我们需要完成测试而开发的脚本。例如：在购买商品时需要添加购买地址，存在其他脚本可能也需要该购买地址添加的步骤的现象，这时候直接开发一个单独添加购买地址模块的脚本，需要使用时直接调用该脚本即可。

这类脚本没有相应用例对应，即在命名时不需要写用例编号，一级名称写应用名称，二级名称写该脚本的描述（该脚本的用途）以下划线“_”间隔即可。

示例：淘宝网_添加购买地址模块

2.6 控件/图像/坐标识别使用的先后顺序和基本场景

1. 先后顺序：

脚本录制过程中建议能够控件识别的优先使用控件操作，控件识别不到的先

考虑图像识别的方式进行操作，最后在考虑坐标识别进行操作。

2. 使用场景：

1) 控件识别：在脚本录制过程中，在映射屏上鼠标放在控件的位置会显示小红框，识别控件后可对该控件进行点击、断言、if 逻辑等操作，例如通常情况下我们对应用功能的一些点击操作都是控件点击。

2) 图像识别：银行类 app 的密码键盘或交易键盘通常为安全键盘，安全性极高，控件无法识别，这时使用图像识别的方式进行密码输入。

- 使用**取图**功能，在界面上框出想要进行操作的区域，右键鼠标就可以对该区域（即框出的图片）进行相应的操作。
- 在生成的脚本步骤中，我们可以对图片操作进行设置，设置扩大图片的**查找范围**，确保键盘乱序后也可以找到图片。
- 设置图片的**容错率**，调低图片容错率可以确保通过像素识别精确地查找该图片。

3) 坐标识别：在启动一些 app 后经常会先进入到广告页，需要滑动屏幕几下后方可进入到 app 首页，此时可以在映射屏上模仿手动去滑动屏幕，即可实现坐标滑屏的操作（再例如新闻页面往下滑动页面去查看相关内容，也使用到的是坐标滑屏的方法）。

2.7 测试数据（参数化数据源）准备规范

- 测试时需要测试的数据准备充分，建议在测试项目开启前准备完毕。例如可能涉及到 10 台手机一起测试 app 的功能，这时需要提前提供 10 个账号，进行参数化，且账号中已经开通需要的相关业务数据等。
- 测试过程中突发测试数据的需求时，建议负责准备数据的同事应能及时响应开发脚本的同事，进行测试数据的准备与提供，确保项目顺利进行。
- 测试过程中需要对变量值进行改变时，则需在参数模板中进行改变。

3 版本控制

3.1 应用版本控制和脚本迁移

应用更新版本后的脚本迁移规范，需要通过以下几个步骤实现：

- 迁移脚本：在旧版本应用中勾选要迁移到新版本应用的脚本
- 批量复制：批量将选择的脚本进行复制
- 选择应用版本：选择脚本需要迁移到的新版本
- 更新 id：复制（迁移）完成所有脚本后，更新新应用下的脚本 id

注意：

- 1) 复制前可能在旧版本应用中没有想要迁移的脚本（该脚本在云端，本地没有）这时则需要我们先从云端下载该脚本，在勾选复制等。
- 2) 批量复制及更新 id 的时间可能会较长（由于脚本过多的原因），则需耐心等待，切不可中途强行退出客户端，打断进程等操作。
- 3) 完成脚本迁移后，可批量将脚本上传云端。

3.2 脚本版本控制

- 开发好的脚本如果存在着一些错误或者异常（比如图像容错率没有调整导致分辨率不同的手机没有点击到）导致脚本不能通过，需要修改脚本时，可以直接在该脚本中进行修改更新脚本，保证脚本执行无误。
- 开发好的脚本需要更新（修改）时，此修改不涉及脚本问题，而是想在该脚本的基础上新增功能，或者检查点，亦或尝试能否实现一些其他的需求时，建议在更新时先复制该脚本，在复制的脚本基础上进行更新。
- 录制其他功能的脚本，或完成新需求的实现时，建议新建脚本实现。