#### 基于 python+Testlink+ enkins 实现的接口自动化测试框架 V4.0

**by:授客 QQ：1033553122**

**博客：**[**http://blog.sina.com.cn/ishouke**](http://blog.sina.com.cn/ishouke)

**欢迎加入软件性能测试交流 QQ 群：7156436**

**目录**

[1、 开发环境 1](#_bookmark0)

[2、 主要功能逻辑介绍 2](#_TOC_250011)

[3、 框架功能简介 4](#_TOC_250010)

[4、 数据库的创建 6](#_TOC_250009)

[5、 框架模 详细介绍 6](#_TOC_250008)

[6、 Testlink 相关配置与用例管理 14](#_TOC_250007)

[a) API 相关配置 14](#_TOC_250006)

[b) 项目相关配置 18](#_TOC_250005)

[c) 用例管理 20](#_TOC_250004)

① 步骤动作和预期结果填写规范 21

[② 参数化 25](#_TOC_250003)

③ 用例执行依赖 29

④ 禁用用例 29

[7、 运行结果 30](#_TOC_250002)

[8、 源码下载 31](#_TOC_250001)

[9、 说明 31](#_TOC_250000)

#### 1、 开发环境

win7\Windows Server 2008 R2 x64 PyCharm 4.0.5

setuptools-29.0.1.zip

下载 ：<http://pan.baidu.com/s/1mhMSAkK>

官方下载 ：[https://pypi.python.org/pypi/setuptools#downloads](https://pypi.python.org/pypi/setuptools)

python 3.3.2

mysql-connector-python-2.1.3-py3.3-win32

下载 ：<http://pan.baidu.com/s/1kTRqRht> mysql-connector-python-2.1.4-py3.3-winx64.msi 下载 ：<http://pan.baidu.com/s/1cDtP1O>

官方下载 ：<http://dev.mysql.com/downloads/connector/python/>

testlink-1.9.14

下载 ：<http://pan.baidu.com/s/1pLrcunT> 安装教程：<http://blog.sina.com.cn/s/blog_13cc013b50102w9am.html>

TestLink-API-Python-client-master

下载 1：<http://pan.baidu.com/s/1c16H50O>

下载 2：

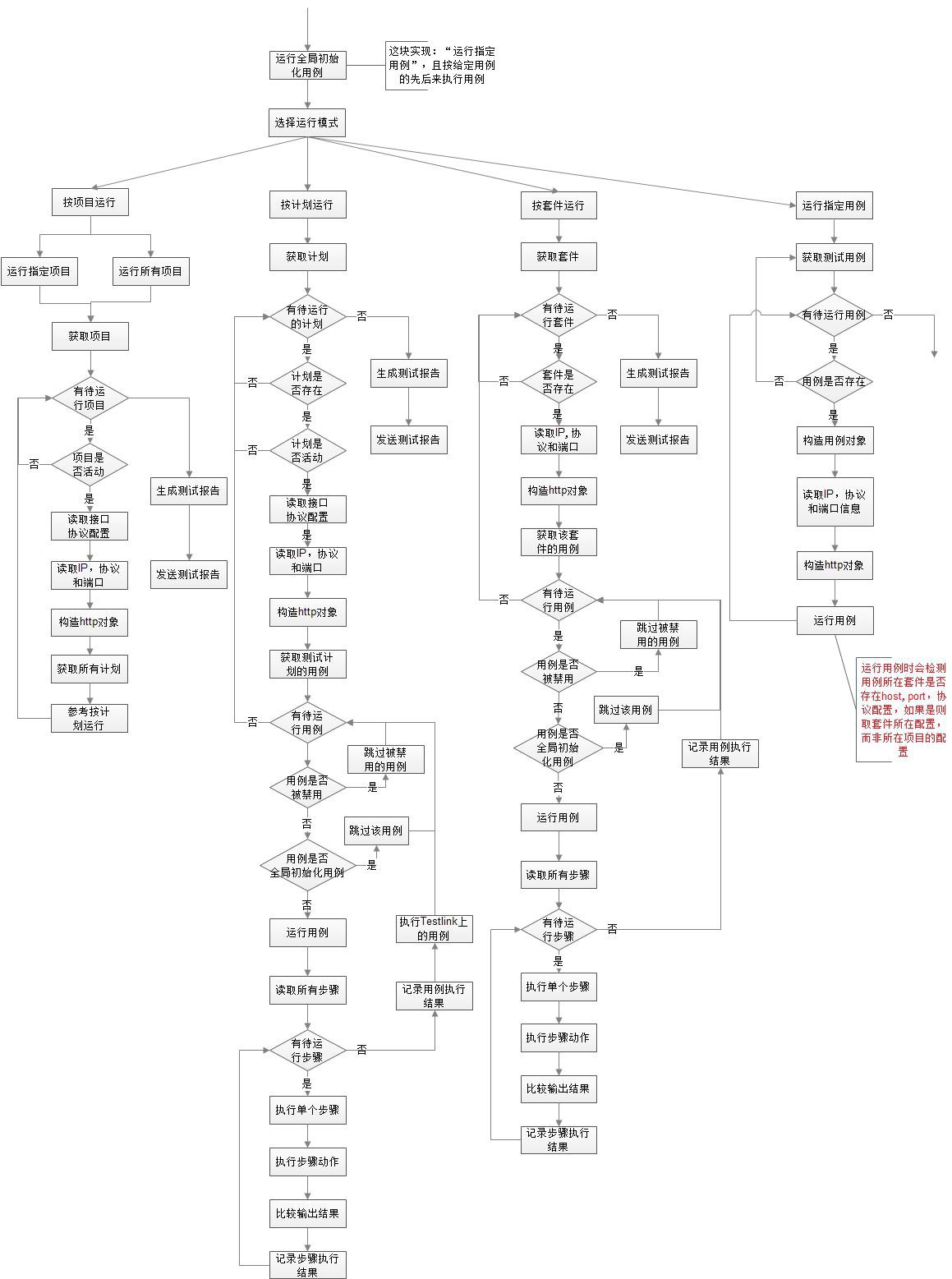
[https://github.com/lczub/TestLink-API-Python-client#testlink-api-python-client-developers](https://github.com/lczub/TestLink-API-Python-client)

chardet-2.3.0

下载 1：<https://pypi.python.org/pypi/chardet/>

下载 2：<http://pan.baidu.com/s/1nu7XzjN>

#### 2、 主要功能逻辑介绍



#### 3、 框架功能简介

1、框架集成了 Testlink,可使用 Testlink 灵活对测试项目，测试计划，测试用例进行管理

2、可通过配置文件灵活配置运行模式：

 支持按测试项目运行：一次运行单个、多个指定的项目或者全部项目；

 支持按测试计划运行：一次运行单个、多个指定的测试计划；

 支持按测试套件运行：一次运行单个、多个指定的测试套件(注：支持套件嵌套，套件 -- testlink 中的 测试集)

 支持按用例运行：一次运行单个\多个用例，这点对特别方便开发阶段时，对单个接口的实现代码进行调试

3、支持 HTTPS，HTTP，WebService 协议，支持 POST, GET 方法，支持 JSON，非 JSON 数据格式的请求，支持 多种形式的数据校验，包含数据库级别的数据校验

4、支持 界面化操作，无须写代码就可以实现如下操作：

a) 自定义变量存储 web 服务器、数据库服务器返 请求/查询结果

b) 根据自定义模式对 web 服务器返 结果进行自动校验，支持多种模式的校验，包含字符串，不包含字符串， 键值提取，包含成员，不包含成员，匹配/不匹配正则表达式，完全匹配列表/元组/集合/字典

c) 根据界面输入的 sql 语句，执行 sql 查询/更新操作，针对只对返 单条记录的 sql 查询，还支持对查询结 果进行提取， 存

d）支持 url 及参数体的动态参数化，支持全局动态参数，非全局动态参数（如存储某个接口返 结果的自定义 变量）

5、针对脚本中已经支持的常见协议及常用数据格式，且不需对接口执行结果进行数据库级别的逻辑校验，支持 界面直接增加用例而不需要改动脚本代码，即不会编码的人也可以使用本框架

6、支持不同编码(utf8,ascii,gb2312)的返 结果，且可自由扩展

7、可自动生成 HTML 可视化接口测试报告

8、可根据配置 测试完成后，自动发送测试报告邮件，邮件发送支持 SSL 加密发送和非 SSL 加密发送，同时支 持往多个邮箱发送邮件

9、支持文件、控制台的日志打印，可分别控制开关

10、支持模 化开发

11、支持测试环境的“一键”切换：python main.py arg，eg python main.py 1

其中,arg: 1-测试环境 2-预发布环境 3-集成环境，可根据实际需要 代码、配置文件中做适当调整,支持自由 扩展和更改

12、可集成 Jenkins 自动运行脚本 参考文章：

“[为 Jenkins 添加 Windows Slave 远程执行 python 项目脚本](http://blog.sina.com.cn/s/blog_13cc013b50102w1yr.html)” “[利用 Build With Parameters Plugin 实现 Jenkins 参数化构建](http://blog.sina.com.cn/s/blog_13cc013b50102wrjj.html)” “[利用 HTML Publisher plugin 实现 HTML 文档报告展示](http://blog.sina.com.cn/s/blog_13cc013b50102wryh.html)”

如果需要支持参数化构建，需要替换 main.py,globalpkg/global\_function.py,config/runmodeconfig.py

文件

Windows batch，main.py 参数按如下格式写，注意顺序必须遵守如下：

cd /d C:\Projects\interface\_project\_for\_test python

main.py %run\_env% %runmode% %project\_mode% %projects% %project% %plans% %testsuites% %case\_id

\_list% %global\_case\_str%

完整批处理参考：jenkins 集成 html 报告显示后的 windows 构建批处理命令.txt



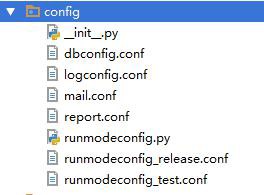
注意：集成里面的参数默认值只能为英文的，另外，字符必须加双引号

#### 4、 数据库的创建

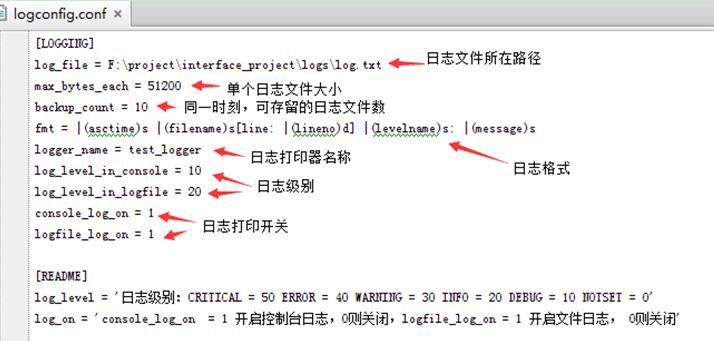
CREATE DATABASE IF NOT EXISTS interface\_autotest DEFAULT CHARACTER SET utf8;

#### 5、 框架模 详细介绍

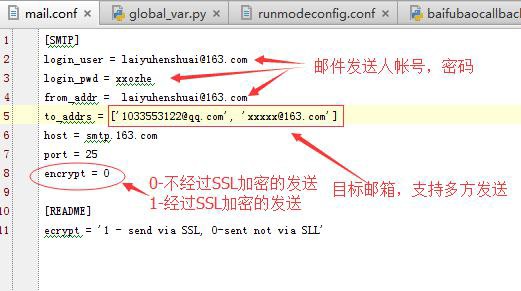
1. **config**



**dbconfig.conf：**包含测试数据库，应用数据库的配置信息 **logconfig.conf：**包含日志配置信息，具体如下：



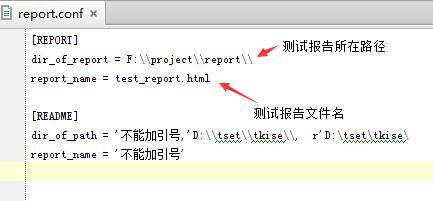
**mail.conf：**包含邮件发送配置信息，如下，



注：不同类咋的邮箱(发件人邮箱)，需要修改配置文件为对应的 host 和端口

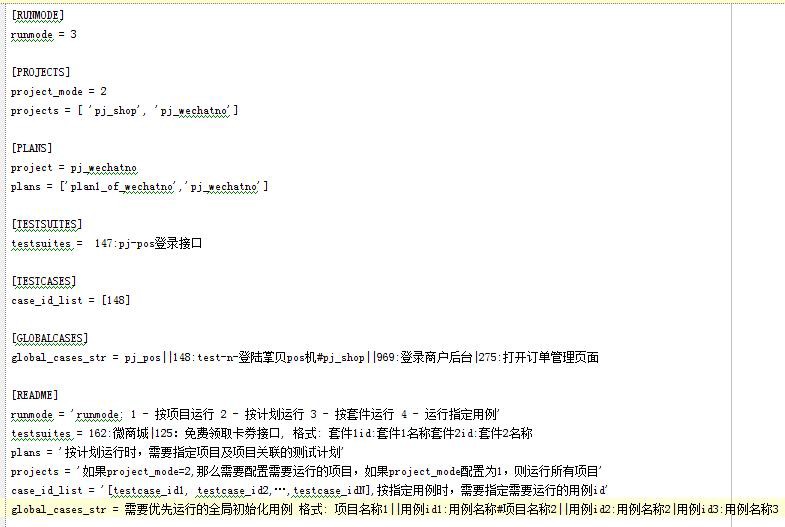
smtp.163.com:25 smtp.qq.com:465

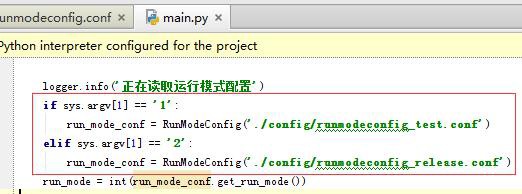
**report.conf：**包含测试报告文件配置信息，如下



**runmodeconfig\_xxxx.conf：**包含运行模式配置信息, runmodeconfig\_test.conf 和

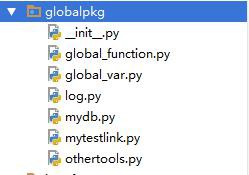
runmodeconfig\_relase.conf 分别为测试环境和预发布环境的运行时配置信息，可根据实际情况进行调整





**runmodeconfig.py：**运行配置类

#### globalpkg

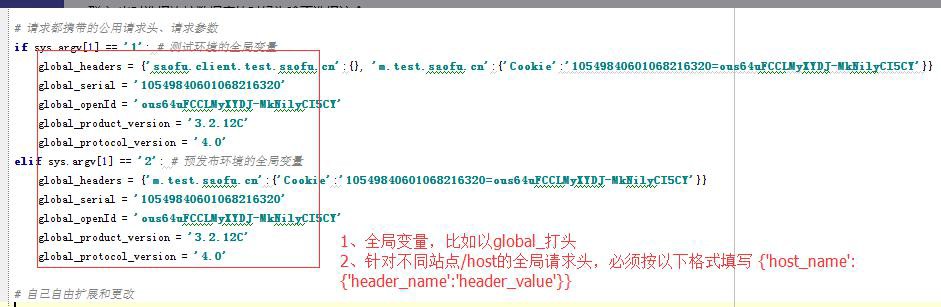


**log.py：**实现日志打印类 **mydb.py：**实现数据库类，封装数据库相关操作 **mytestlink.py：**主要用于获取 testlink 连接实例 **othertools.py：**实现其它通用功能，比如数据转换，批量创建目录等



**global\_var.py：**主要提供全局变量，全局实例等





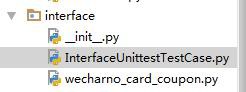
**global\_function.py：**主要提供全局函数，比如提供运行单个用例的函数 run\_testcase\_by\_id 等

#### logs 及 testreport

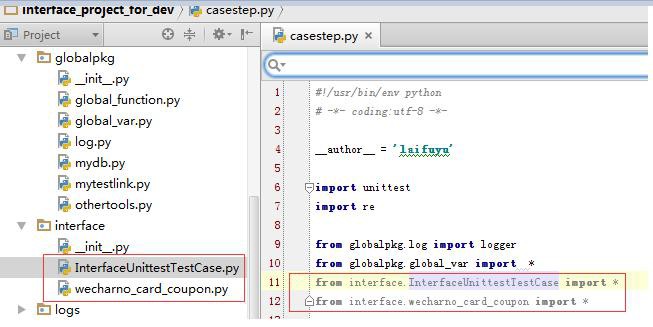
可分别用于存放日志文件，测试报告

#### interface

封装接口测试方法类

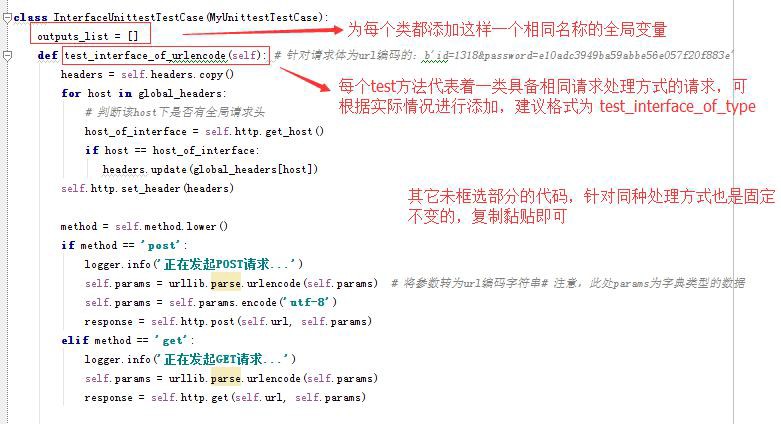


说明：可根据需要，每个接口对应一个模 ，对应一个类；也可以多个接口对应一个模 ，对应一个类 需要注意的是，这里添加的模 及类，需要 casestep.py 中导入



#### 接口单元测试类代码说明





说明：新版本代码中，

outputs\_list = []已经改成 outputs\_dict = {}

处理方式目前支持 3 中 test\_interface\_of\_urlencode（针对请求体为 url 编码），test\_interface\_of\_json(针 对请求体为 json 格式)，test\_interface\_of\_xml(针对请求体为 xml 格式)





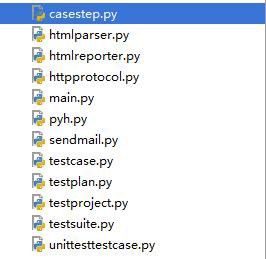
**问题：假如要进行数据库后台的逻辑校验咋办？**

针对核心用例我们可以这样，复制整个 test\_xxx 为 1-到 n 份，修改方法名，然后 代码最下方新增数据 库后台校验的代码，同时如果有必要，对其它代码进行适当的修改（新增、删除、修改等），然后用例 步骤中显示指定要调用的方法即可。

**或者把逻辑写成存储过程，扩展代码，增加存储过程调用**

**e) 其它模**

如下，顾名思义

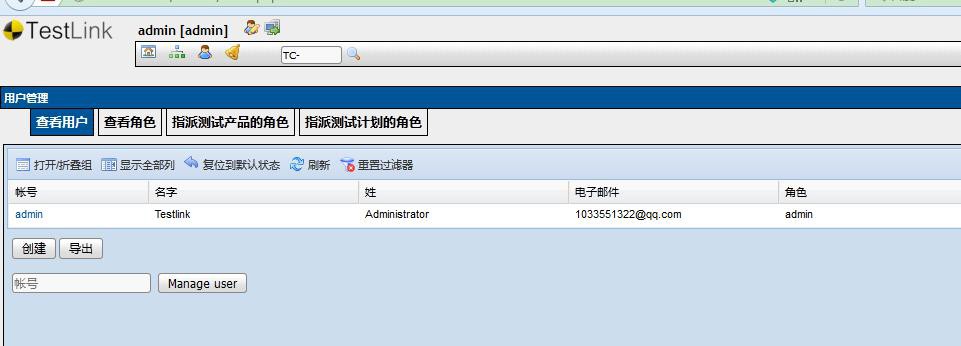


#### 6、 Testlink 相关配置与用例管理

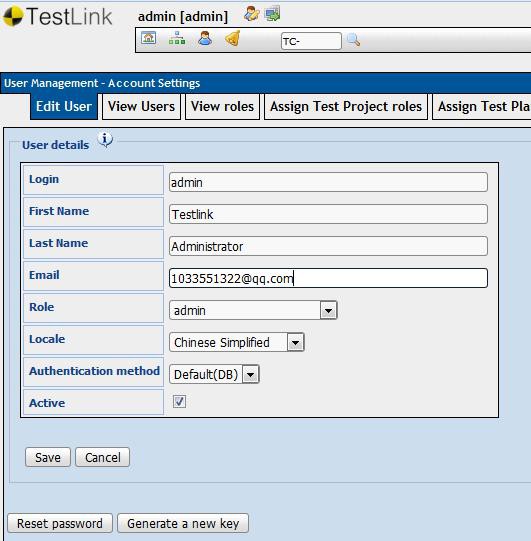
为了批量设置接口 ip，端口(主要是这两个)，协议信息(仅用于展示)，需要对项目，计划，套件等必要的配置， 以及客户端环境变量配置

#### a) API 相关配置

如下，登陆 Testlink，进入用户管理-查看用户，如下

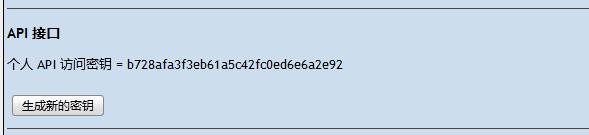


点击目标用户(例中为 admin)，打开如下界面

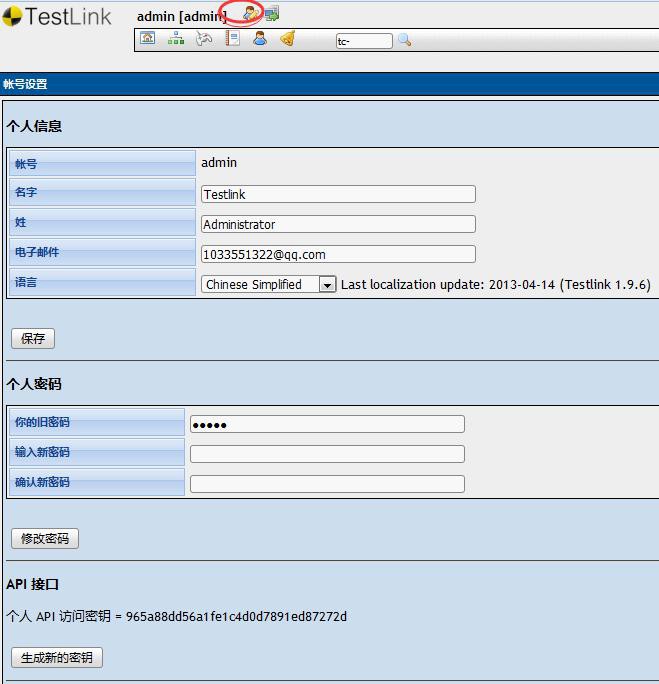




点击生成新的密钥，如下

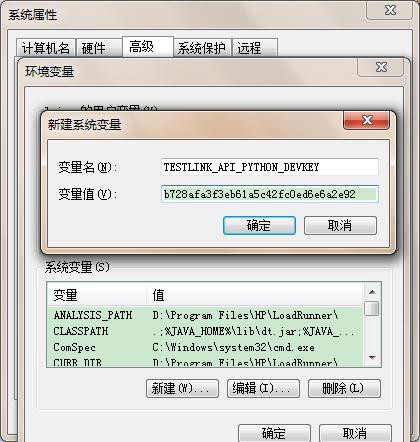


或者是点击“编辑用户按钮”按钮进入界面



运行 python 脚本端进行环境变量的配置，如下：

1、新建系统环境变量“TESTLINK\_API\_PYTHON\_DEVKEY”，变量值为上述秘钥



2、新建“TESTLINK\_API\_PYTHON\_SERVER\_URL”系统环境变量，变量值为“ http://{host}/testlink/lib/api/xmlrpc/v1/xmlrpc.php”,其中 host 为 testlink 的访问



测试是否生效：

C:\Users\laiyu>python

Python 3.3.2 (v3.3.2:d047928ae3f6, May 16 2013, 00:03:43) [MSC v.1600 32

tel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import testlink

>>> tls = testlink.TestLinkHelper().connect(testlink.TestlinkAPIClient)

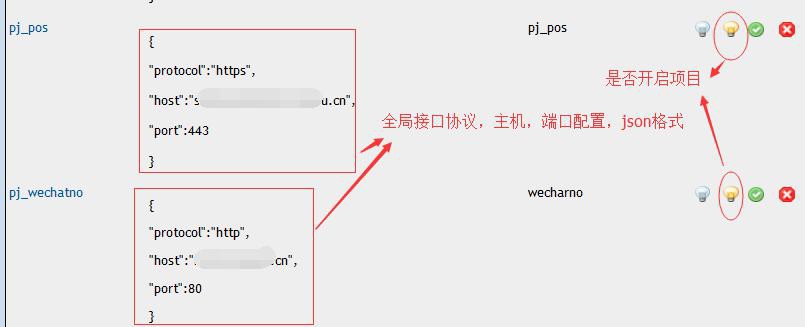
>>> tls.testLinkVersion() '1.9.14'

注意：如果还不行，提示 404 错误，则还需要配置 testlinkhelper.py 中的 DEFAULT\_SERVER\_URL，将其设置 为 http://{host}/testlink/lib/api/xmlrpc/v1/xmlrpc.php

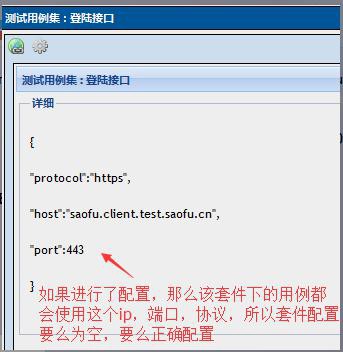
python\_installation\_home\Lib\site-packages\TestLink-API-Python-client-master\build\lib\t estlink\testlinkhelper.py

## b) 项目相关配置

#### 测试项目配置



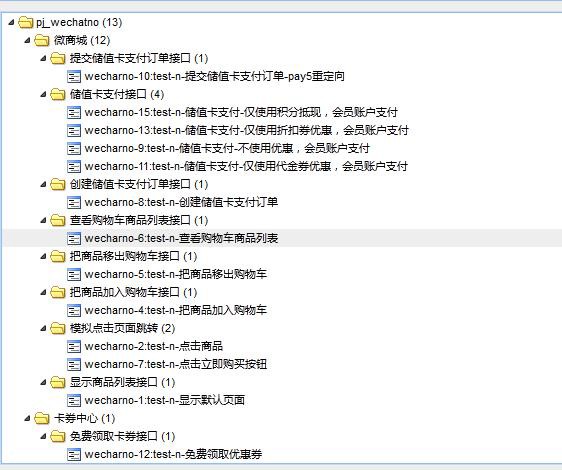
**注：计划，也存 是否开启的配置，需要开启才会运行。 套件配置**





#### c) 用例管理

如下，用例层级，支持嵌套管理



用例编写如下，左侧步骤动作，必须按如下格式编写 **步骤动作和预期结果填写规范**

**1**

#### 针对 http,https 请求，且步骤不是执行单个用例、不是执行 sql，按如下填写： 步骤动作

**json 格式**

{

"类名":"InterfaceUnittestTestCase", "函数":"test\_interface\_of\_urlencode",

"请求头"：{"Content-Type":"application/x-www-form-urlencoded","charset":"utf-8"}, "方法":"post",

"url":"/cmorder/1/4", "参数":{"serial":"[global\_serial]","openId":"

[global\_openId]","fm":1,"mallGoodsId":"26838", "amount":1,"cartIds":"","fetchTime":"", "remark":""}

}

注：

#### 1、如果不填写“类名键值对”，那么类名默认为 InterfaceUnittestTestCase，如果不填写“函数键值对”， 那么函数默认为 test\_interface\_of\_urlencode

**2、键对应的参数值为空，则填写为 ""，“键”，用双引号包含，键对应的“值”，有必要的 也是双引号包含， 预期结果的填写也是一样，不再赘述**

**3、如果“参数”值为空，则填写为 "参数":""**

**4、用例步骤对应的 test\_xxx 方法实现，要尽量做到接口测试数据和业务逻辑分离，增强复用性和用例的可维 护性。**

**预期结果**

json 格式

{"检查":"body",

"匹配规则":"包含字符串",

"条件":[{"模式":"\"success\":true", "消息":"创建储值卡支付订单失败，success 不为 True"}], "输出":{"re":{"success": "\"success\":(.+?),","attach":"attach\":\"(.+?)\""}}

}

注：

**1、预期结果要么按规范填写，要么为空，啥都不写**

**2、如果不想定义变量存储服务器返 结果，则不填写“输出键值对”**

**3、注：匹配规则和条件，要么都写，要么都不写，如下**

{"检查":"body",

"输出":{"re":{"success": "\"success\":(.+?),","attach":"attach\":\"(.+?)\""}}

}

**样例**



**注：4.0 中“输出”增加了提取方式，见上文**

**针对 http,https 请求，且步骤执行单个用例、不是执行 sql，按如下填写，json 格式 步骤动作**

{

"步骤类咋":"执行用例", "用例 id":106,

"用例名称":"test-n-创建储值卡支付订单"

}

**预期结果** 为空，不填写

#### 样例



针对步骤行 sql，按如下填写： **更新(UPDATE)**

#### 步骤动作

**json 格式**

{

"步骤类咋":"执行 sql",

"数据库":"db\_name",

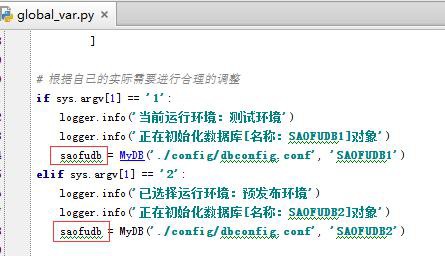
"更新":"UPDATE mall\_shopping\_cart SET amount = 0 WHERE customer\_id = %s and closed=1", "参数":"([CaseStep.customer\_id],)"

}

#### 注意：

**1、如果“参数”值为空，则填写为 "参数":""**

**2、“数据库”，要从那个数据库查数据，其值来源于 global\_var 里面的数据库对象名称。如下**



注意：这里如果添加了数据库对象，需要 main.py 近末尾处中添加关闭数据库连接的代码 db\_name.close()

**预期结果** 为空，不填写

#### 查询(SELECT)

**步骤动作**

**json 格式**

{

"步骤类咋":"执行 sql",

"数据库":"db\_name",

"单条查询":"SELECT id FROM customer WHERE channel\_serial=%s", "参数":"(\"[global\_openId]\",)"

}

#### 注:参数值为元组的字符串类咋表示 (value1,value2,……)，如果参数为空则 "参数"："",如果参数值为字符 串类咋的，则按这样填写： (\"参数值\",)

**预期结果**

**json 格式**

{"检查":"body",

"输出":{"customer\_id":1}

}

或者如下

{"检查":"body",

"输出":{"customer\_id":1}, "匹配规则":"相等",

"条件"：[{"模式"："[CaseStep.device\_no]=123456", "消息"：“登录失败，device\_no 不为 123456”}]

}

**注：预期结果要么为空，要么严格按照上述填写**

**样例**



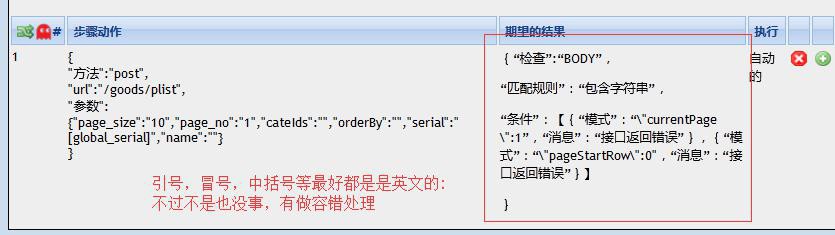
其它一些样例 发送请求参数为 xml 格式



发送请求参数为 json 格式



容错处理



#### 参数化

**2**

**1) 针对 https,http 请求，且不是执行 sql，也不是执行整个用例的步骤动作、预期结果** 如果参数位于“参数”、“请求体”，且“参数”、“请求体”的值为字典，按如下格式填写

#### "[var\_name]"

{

……, "参数":{"key1":"[global\_var1]","key2":"[global\_var2]"}

}

{

……,

"请求头":{"DeviceId":"[CaseStep.device\_no]", "SerialNo":"[CaseStep.serial\_no]",

"Content-Type":"application/json;charset=utf-8","ProductVersion":"[global\_product\_version] ","ProtocolVersion":"[global\_product\_version]","OperatorId":"[OperatorId]", "Token":"[Token]"},

……

}

如果参数位于“url”或者预期结果“条件”中的字典 key 对应的值，按如下格式填写

#### [var\_name]

{

# ……, "urL":"/kq/getincoupon/40758966216286146560/[global\_openId]",

……

}

{

# ……,

"条件"：[{"模式"：｛"key": "[CaseStep.branch\_id]"},"消息"：“key 错误”}]

……

}

注意：所有变量名 不能为纯数字，比如[1], 为需要支持 xpath 提取，这类纯数字的都视为非变量

**2) 针对执行 sql 的步骤动作、预期结果** 参数位于步骤动作、步骤预期结果，按如下格式填写 **[var\_name]或者 \"[var\_name]\"**

{

……，

"参数":"(\"[global\_openId]\",)"

……

}

{

"检查":"body",

"匹配规则":"相等",

"条件"：[{"模式"："[CaseStep.device\_no]=123456", "消息"：“登录失败，device\_no 不为 123456”}]

}

#### 注：如果待替换"动态变量" [var\_name]为字符串类咋的变量，需要 其左右两侧加 \" 符号，形如

**\"[global\_openId]\"**

**3) 变量划分**

全局变量[global\_var]，来自 globalpkg.global\_var.py 中定义的，var\_name 和定义的全局变量名称 持一

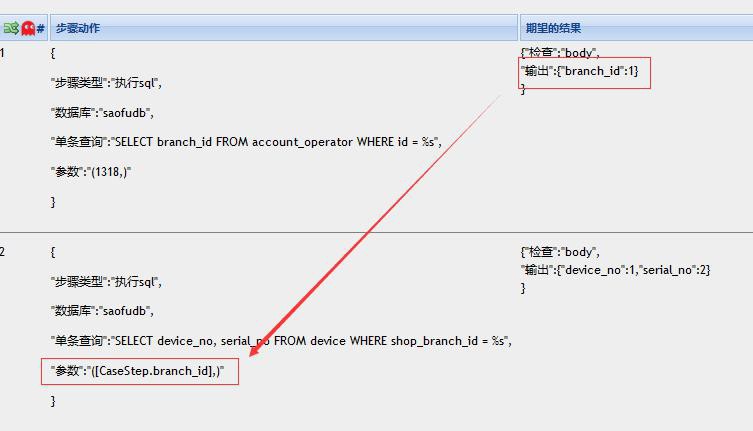
致即可

非全局变量[ClassName.var\_name]，可能来自 sql 查询结果，也可能来自 http(s)请求返 结果中提取的

#### A) “步骤动作”中的非全局变量来源于某个 sql 查询结果，则按如下格式填写

**[CaseStep.var\_name]**

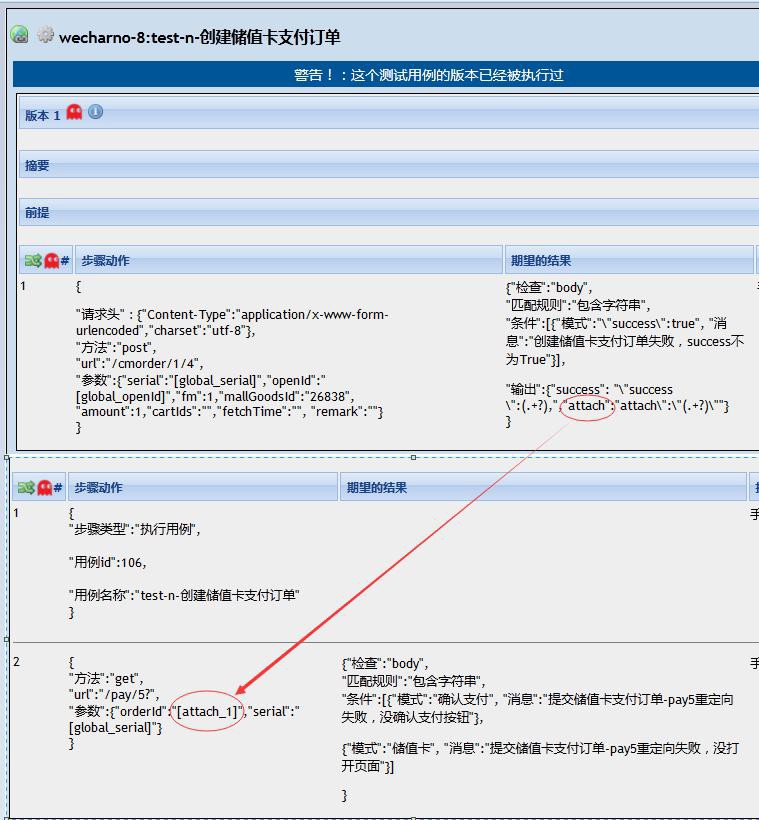
**样例**



B) “步骤动作”中的非全局变量来源于某个 http(s)请求返 结果，则按如下格式填写

#### [ClassName.var\_name]、[ClassName.var\_name\_N]

**样例**



**注：**

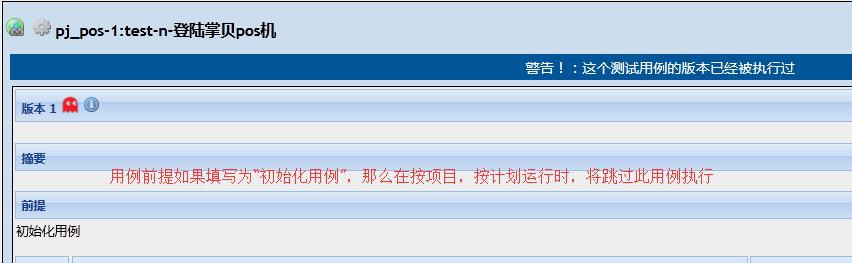
**1、当调用的函数为默认类名，即 InterfaceUnittestTestCase 类中定义的方法时，这里“ClassName.”可以 不写，如果是调用其它新增类中的方法时，需要显示的填写**

**2、当且仅当变量是通过正则表达式提取的时候，采用[ClassName.var\_name\_N]，的形式 N 从 1，2，3，…… 以此类推**

**4) 结果断言和服务器返 内容提取** 参考“结果断言和提取服务器返 结果.txt”文档

**用例执行依赖** 全局初始化用例

**3**



注：全局初始用例 runmodeconfig\_运行环境.conf 中进行配置，优先于其它用例被执行



注意：没有则全局初始化用例 持为空 global\_cases\_str =

当把某个用例作(假设为“用例 1” )为其它用例的某个步骤，不能满足需求(比如提供的接口输入参数值不一样)

时，可以复用“用例 1”的步骤信息并对参数进行适当的调整

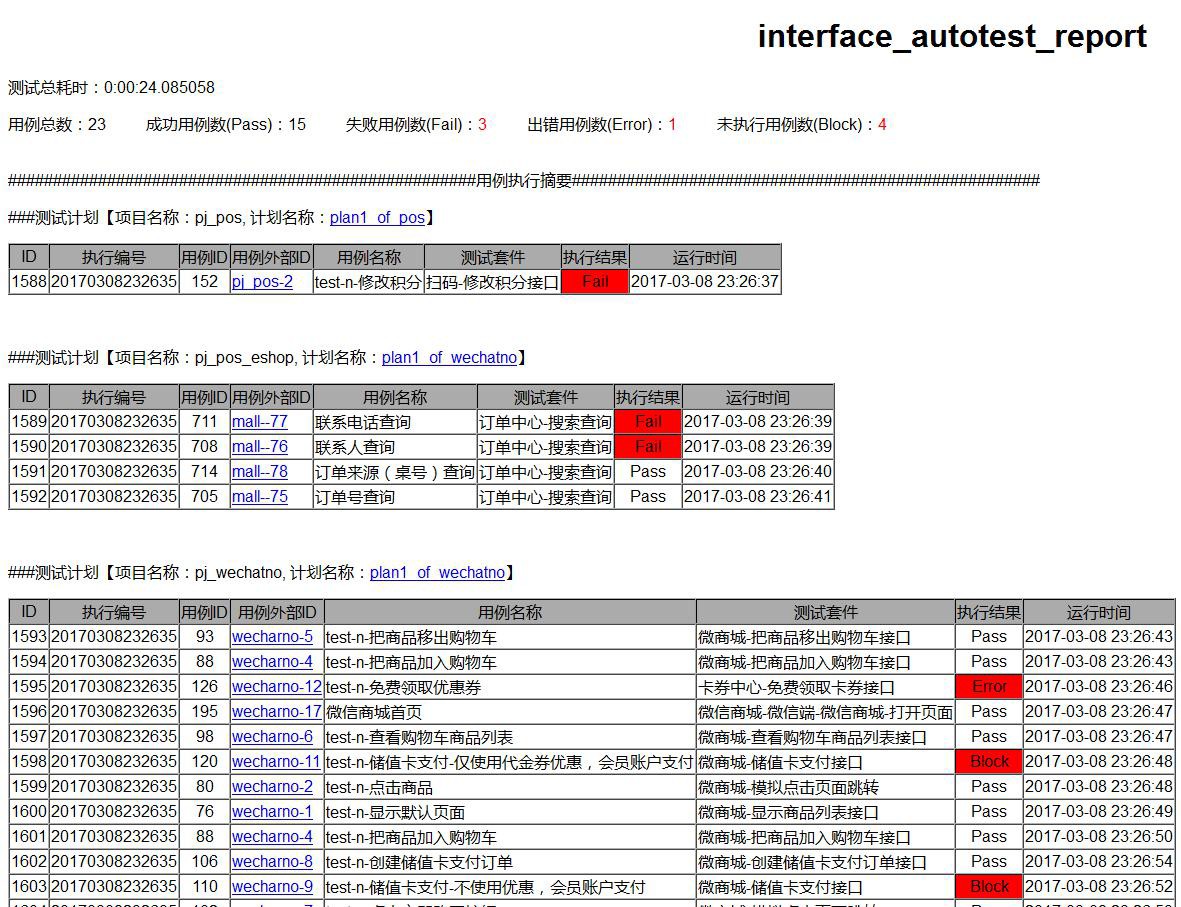
禁用用例

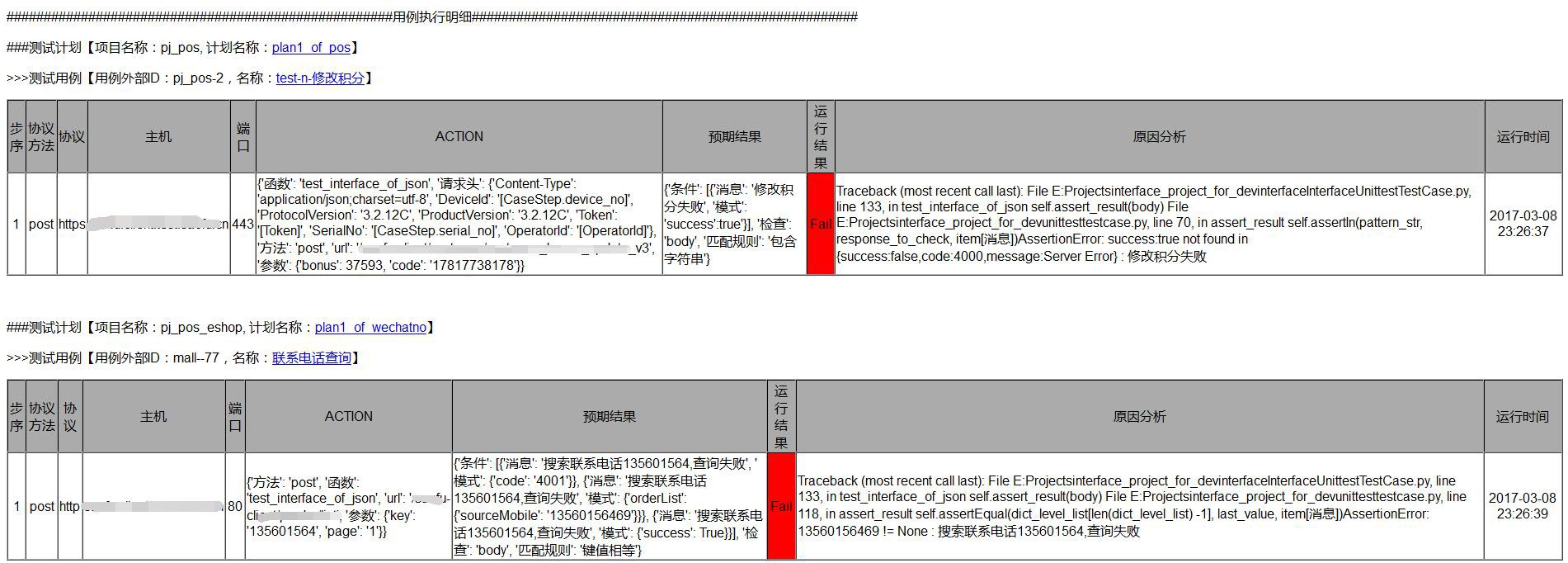
4



#### 注意：被禁用的用例不参与执行，计划，项目也是如此

#### 7、 运行结果





#### 8、 源码下载

下载 ：

下载后解压，用 pycharm 导入项目即可，必要时可能需要修改.idea/workspace.xml 中 python

程序所 路径

#### 9、 说明

时间有限，精力有限，暂且就到这吧，有需要的可以自己扩展、修改框架。