自动化测试全流程指导

目录

**[自动化测试全流程指导](#_Toc522631611)** [1](#_Toc522631611)

**[I自动化测试准备阶段](#_Toc522631612)** [2](#_Toc522631612)

[1 自动化测试范围评估 2](#_Toc522631613)

[2 获取项目接口文档 3](#_Toc522631614)

[3 申请自动化测试环境及测试执行机 3](#_Toc522631615)

[4 Robot Framework及相关测试库学习 3](#_Toc522631616)

[5 自动化工程及相关样例模板的提供 3](#_Toc522631617)

**[II自动化脚本写作阶段](#_Toc522631618)** [4](#_Toc522631618)

[1 如何搭建本地自动化脚本写作环境 4](#_Toc522631619)

[1.1安装python 4](#_Toc522631620)

[1.2安装robot framework 4](#_Toc522631621)

[1.3安装wxPython 4](#_Toc522631622)

[1.4安装RIDE 4](#_Toc522631623)

[1.5安装依赖的第三方测试库 4](#_Toc522631624)

[1.6 migu\_library安装和使用 8](#_Toc522631625)

[1.7如何注释掉不需要的库 9](#_Toc522631626)

[2 自动化脚本写作规范 10](#_Toc522631627)

[规范1：工程目录结构及文件格式规范 10](#_Toc522631628)

[规范2：Documentation使用规范 11](#_Toc522631629)

[规范3：测试套/测试用例/变量命名规范 13](#_Toc522631630)

[规范4：关键字封装及使用规范 13](#_Toc522631631)

[规范5：资源文件使用规范 14](#_Toc522631632)

[规范6：Setup和Teardown的使用规范 14](#_Toc522631633)

[规范7：Sleep和Timeout的使用规范 15](#_Toc522631634)

[规范8：自动化用例校验点规范 15](#_Toc522631635)

[规范9：关于注释的规范 15](#_Toc522631636)

[3 自动化脚本写作指导 16](#_Toc522631637)

[3.1常用关键字使用说明 16](#_Toc522631638)

[3.2接口报文配置文件说明 19](#_Toc522631639)

[3.3资源文件的使用说明 20](#_Toc522631640)

[3.4脚本如何适配多套测试环境 21](#_Toc522631641)

[3.5基于报文模板的脚本写作方法 22](#_Toc522631642)

[3.6接口字段异常校验场景脚本写作方法 23](#_Toc522631643)

[3.7脚本写作小技巧 24](#_Toc522631644)

[4 自动化脚本与用例互转工具使用指导 25](#_Toc522631645)

[4.1工具安装： 25](#_Toc522631646)

[4.2用例转脚本工具使用： 25](#_Toc522631647)

[4.3脚本转用例工具使用： 26](#_Toc522631648)

[5 自动化脚本规范扫描工具使用指导 29](#_Toc522631649)

[5.1 扫描规则 29](#_Toc522631650)

[5.2 工具使用 31](#_Toc522631651)

[6 版本控制使用指导 32](#_Toc522631652)

**[III自动化测试应用阶段](#_Toc522631653)** [32](#_Toc522631653)

[1 自动化测试与Jenkins集成指导 33](#_Toc522631654)

[1.1 权限申请 33](#_Toc522631655)

[1.2 svnupdate任务配置 34](#_Toc522631656)

[1.3 自动化测试任务的配置 37](#_Toc522631657)

[1.4 测试任务执行 40](#_Toc522631658)

[2 ATP平台使用指导 40](#_Toc522631659)

[2.1 ATP知识库 41](#_Toc522631660)

[2.2自动化建设数据 42](#_Toc522631661)

[2.3 Migu library 44](#_Toc522631662)

[2.4 Robot Framework 45](#_Toc522631663)

[3 CTMP自动化测试平台使用指导 46](#_Toc522631664)

[3.1 权限申请 46](#_Toc522631665)

[3.2 脚本调试 47](#_Toc522631666)

[3.3 项目运行任务 49](#_Toc522631667)

**[IV 自动化测试相关考核说明](#_Toc522631668)** [51](#_Toc522631668)

[1 自动化测试应用于版本冒烟及回归测试 51](#_Toc522631669)

[2 提测版本自动化脚本通过率考核 52](#_Toc522631670)

[3 接口自动化覆盖率考核 52](#_Toc522631671)

[4 失败脚本分析反馈 52](#_Toc522631672)

[5 自动化月报数据反馈 52](#_Toc522631673)

**[V 自动化需求及问题接口人](#_Toc522631674)** [54](#_Toc522631674)

[1 成都 54](#_Toc522631675)

[2 南京 54](#_Toc522631676)

[3 Jenkins相关问题 54](#_Toc522631677)

按照自动化测试各阶段的工作内容，本指导包含以下几个部分，旨在快速解决大家在开展自动化测试过程中可能遇到的问题。

# I自动化测试准备阶段

**注：以下几个步骤可并行进行**

## 1 自动化测试范围评估

开展自动化测试前，测试经理、项目经理、自动化技术支撑人员一起沟通项目需要开展自动化测试的范围，包括系统对外接口、系统各模块间的内部接口、界面功能等，根据测试重要性、优先级制定计划。

## 2 获取项目接口文档

测试经理推动研发人员提供项目接口文档，接口文档内容要完整，包括但不限于接口URL地址、报文头、出入参描述、字段长度、取值范围、错误码、错误描述，请求选填必填字段，返回选填必填字段等，足以指导开展接口自动化测试，如果有不明确或不完整的，相关测试人员要及时推动研发完善。

## 3 申请自动化测试环境及测试执行机

由项目经理负责申请并部署一套独立的自动化测试环境(与手工测试环境，研发自测调试环境等区分开，避免互相影响)，机器尽量在北京资源池，如果不在，可能会影响与Jenkins拉通。

测试执行机，如果测试环境在北京测试区，可以直接使用测试区的执行机10.148.134.136，如果不在，需要单独再申请一台测试执行机，要求该机器与测试环境网络互通。

## 4 Robot Framework及相关测试库学习

我们的自动化测试框架使用Robot Framework，写脚本前，项目测试人员需要提前学习该框架及相关测试库，SVN有各种学习材料。



svn://10.148.68.6/自动化测试

svn://10.148.68.6/TSG测试团队沉淀与分享/自动化测试/RF相关

svn://10.148.68.6/南京测试团队项目/02-项目管理/08-自动化/98-自动化文档

## 5 自动化工程及相关样例模板的提供

各项目自动化工程及相关样例模板由自动化技术支撑人员提供。详见最后接口人。

南京各项目的自动化工程脚本在以下SVN路径：svn://10.148.68.6/南京测试团队项目/02-项目管理/08-自动化

成都各项目的自动化工程脚本在以下SVN路径：svn://10.148.68.6/自动化测试/自动化脚本

测试人员需要申请该svn的权限，方可访问。

Git或svn路径项目组也可自行申请，方便测试脚本和项目代码一起管理。Git在spms上提单，填写方式如下：



# II自动化脚本写作阶段

## 1 如何搭建本地自动化脚本写作环境

Robot Framework正常运行需要安装以下工具：

* Python
* Robot Framework
* wxPython（Python的一套GUI图形库）
* RIDE（RF框架编辑用例、执行用例的工具）

### 1.1安装python

下载地址： <https://www.python.org/downloads>，选择python2.7版本（wxPython工具目前最高支持Python2.7），建议使用默认安装路径，避免路径带来的麻烦，安装完成后需要设置环境变量，PATH=PATH;C:\Python27、PATHEXT=PATHEXT;.PY;.PYM.

### 1.2安装robot framework

下载地址：<https://pypi.python.org/pypi/robotframework>，解压后，使用cmd打开控制台，进入解压后的文件夹，执行命令python setup.py install

### 1.3安装wxPython

下载地址： <https://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/>，同python一样，直接安装，不用改路径，由于RIDE工具依赖wxPython，所以要先装wxPython.

### 1.4安装RIDE

下载地址：<https://pypi.python.org/pypi/robotframework-ride>，安装方法和robot framework一样，打开控制台进入文件夹，执行python setup.py install

安装完成后，双击桌面上的[机器人](http://lib.csdn.net/base/robot" \t "_blank" \o "机器人知识库)图标，打开RIDE工具（桌面也可能没有自动创建快捷方式，请参考<http://blog.sina.com.cn/s/blog_7447a0160102wi51.html>添加）。创建桌面快捷方法，在桌面新建一个txt文件，并在文件中写入ride.py，保存退出。修改文件后缀名为.bat。

### 1.5安装依赖的第三方测试库

注意：

1.以下库非必须，项目组可以根据自己使用情况选择安装。

2.在线安装第三方测试库，有时由于安装版本不对，导入migu\_library时报错，大家在线安装时可以加上版本号，例如：pip install appdirs==1.4.3

pip install appdirs==1.4.3

pip install asn1crypto==0.23.0

pip install bcrypt==3.1.4

pip install beautifulsoup4==4.6.0

pip install bit-array==0.1.0

pip install bitarray==0.8.1

pip install certifi==2017.11.5

pip install cffi==1.11.2

pip install chardet==3.0.4

pip install coverage==4.4.1

pip install cryptography==2.1.1

pip install cx-Oracle==5.1.2

pip install decorator==4.1.2

pip install enum34== 1.1.6

pip install future== 0.16.0

pip install happybase==1.1.0

pip install hbase-thrift==0.20.4

pip install idna==2.6

pip install impyla== 0.14.1

pip install ipaddress==1.0.18

pip install jpype1==0.6.2

pip install jsonpatch==1.16

pip install jsonpath-rw==1.4.0

pip install jsonpath-rw-ext==1.1.3

pip install jsonpointer==1.12

pip install kafka==1.3.5

pip install kafka-python==1.3.5

pip install kazoo==2.4.0

pip install ldap==1.0.2

pip install ldap3==2.3

pip install lxml==4.1.0

pip install modlist==0.1

pip install numpy==1.13.3

pip install packaging==16.8

pip install paho-mqtt==1.3.1

pip install pandas==0.22.0

pip install paramiko==2.4.1

pip install pbr==3.1.1

pip install pip==10.0.1

pip install pluggy==0.5.2

pip install ply==3.10

pip install potsdb==1.0.3

pip install pure-sasl==0.5.0

pip install py==1.4.34

pip install pyasn1==0.3.7

pip install pycparser==2.18

pip install pycrypto==2.6.1

pip install PyHive==0.5.2

pip install pyhs2==0.6.0

pip install PyMySQL==0.7.11

pip install PyNaCl==1.1.2

pip install pyparsing==2.2.0

pip install python-dateutil==2.6.1

pip install python-ldap==2.5.2

pip install pytz==2017.3

pip install redis==2.10.5

pip install redis-py-cluster==1.3.4

pip install requests==2.18.4

pip install requests-toolbelt==0.8.0

pip install robotframework==3.0.2

pip install robotframework-databaselibrary==1.0.1

pip install robotframework-httplibrary==0.4.2

pip install robotframework-jsonlibrary==0.2

pip install robotframework-mqttlibrary==0.5.0

pip install robotframework-pabot==0.43

pip install robotframework-redislibrary==0.1

pip install robotframework-requests==0.4.7

pip install robotframework-ride==1.5.2.1

pip install robotframework-selenium2library==3.0.0b1

pip install robotframework-seleniumlibrary==3.0.0rc1

pip install robotframework-sshlibrary==2.1.3

pip install robotframework-zookeeperlibrary==0.1.4

pip install robotremoteserver==1.1

pip install rsa==3.4.2

pip install sasl==0.2.1

pip install selenium==3.7.0

pip install setuptools==28.8.0

pip install six==1.11.0

pip install ssh==1.8.0

pip install thrift== 0.9.3

pip install thrift-sas==l 0.3.0

pip install thriftpy==0.3.9

pip install tox==2.9.1

pip install urllib3==1.22

pip install virtualenv==15.1.0

pip install waitress==1.1.0

pip install WebOb== 1.7.3

pip install WebTest==2.0.28

pip install wheel== 0.30.0

pip install win-inet-pton==1.0.1

pip install xlrd==1.0.0

pip install xlutils==2.0.0

pip install xlwt==1.3.0

pip install xmltodict==0.10.2

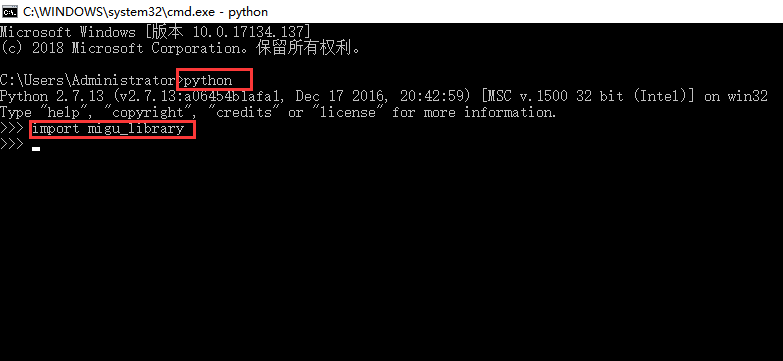
### 1.6 migu\_library安装和使用

**安装：** migu\_library是TSG自动化团队维护的支撑库，用于支撑TSG各项目的自动化需求。支撑库svn地址：

全部：svn://10.148.68.6/自动化测试/migu\_library

南京：svn://10.148.68.6/南京测试团队项目/02-项目管理/08-自动化/99-支撑库。migu\_library的安装:从svn检出支撑库，将migu\_library整个目录放置到本机的Python安装目录\Lib\site-packages即可。

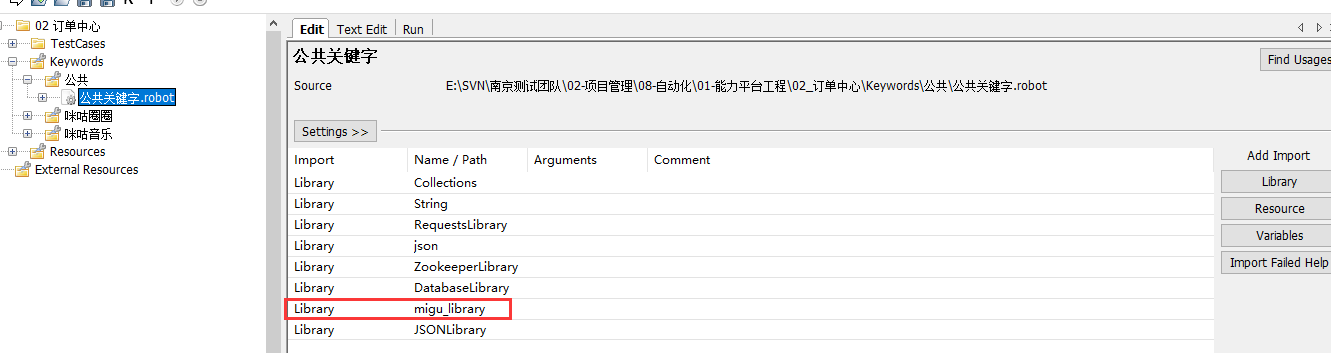
**验证：**cmd命令进入，执行命令python、import migu\_library，查看提示信息。若无任何提示信息则说明migu\_library库可以正常引用，如下图：



**引用：**选中某个测试套或资源文件，点击Library，在Name中填写migu\_library，点击OK即可。

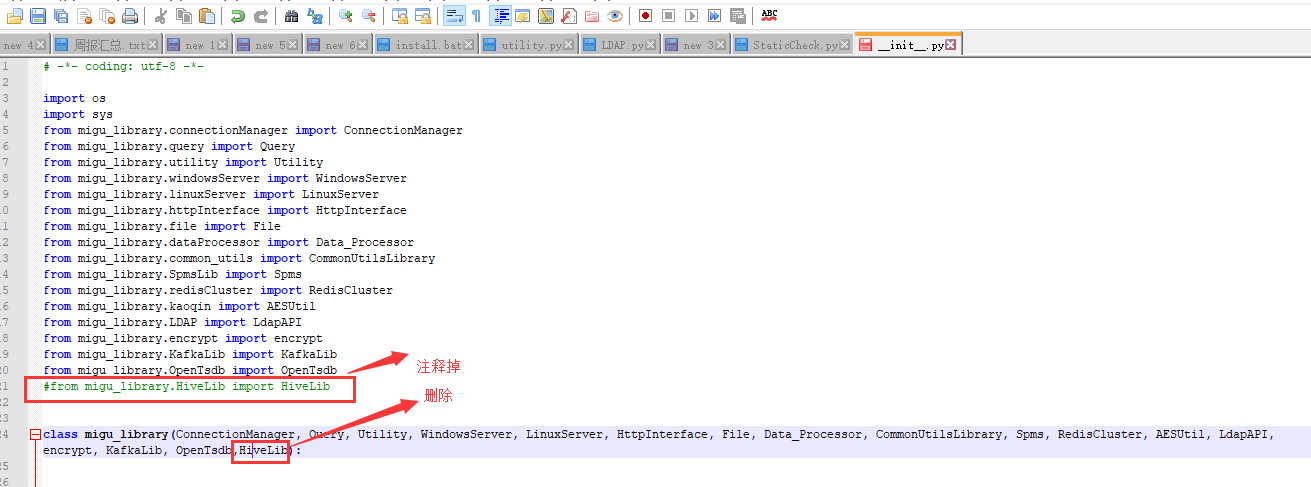


如果此时migu\_library字体为黑色，表示可以正常使用。若为红色，表示引用有异常



### 1.7如何注释掉不需要的库

由于migu\_library是TSG自动化团队维护的支撑库，因此migu\_library中的库不是所有的小组都能用得到，对于这些不需要的库可以采取注释的方式注释掉。在migu\_library中找到\_init\_.py文件。进行图中操作：



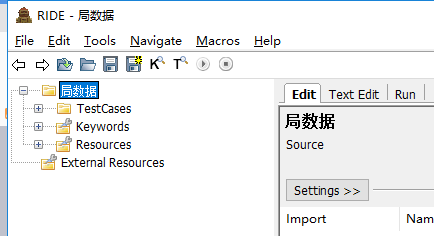
重新执行import migu\_library，查看提示信息，无报错后，重启Ride。

## 2 自动化脚本写作规范

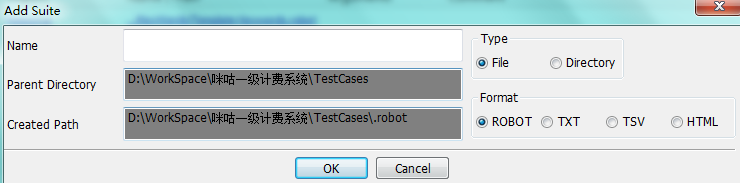
为了统一自动化用例格式，提升写作效率，降低维护成本，制定以下接口自动化用例写作规范，期望自动化验收能够在版本质量守护中发挥更大的价值。

### 规范1：工程目录结构及文件格式规范

1.1每个项目的自动化工程都要包含TestCases、KeyWords、Resources三个目录，分别保存测试用例、业务关键字及资源类文件。

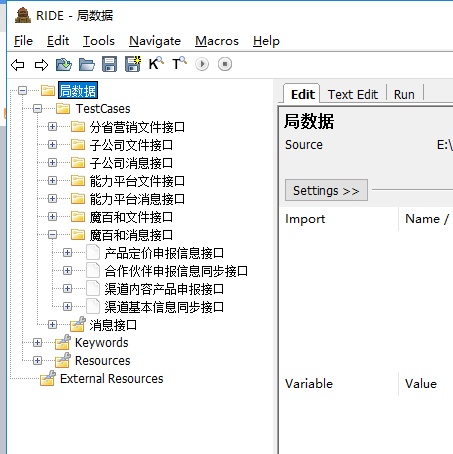


1.2除目录以外，其他文件全部使用robot格式，例如测试套文件、资源文件等。

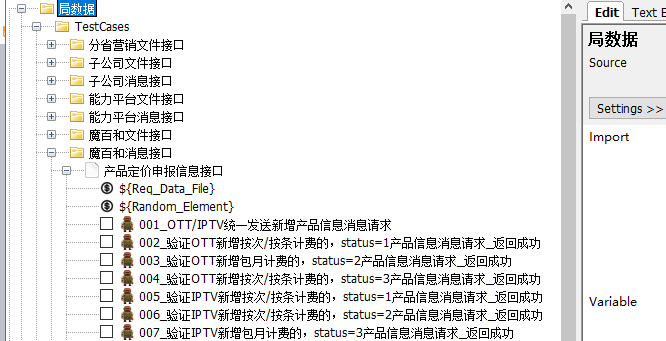




1.3 TestCases目录下按照系统特性分类，特性目录下按照子特性分类。

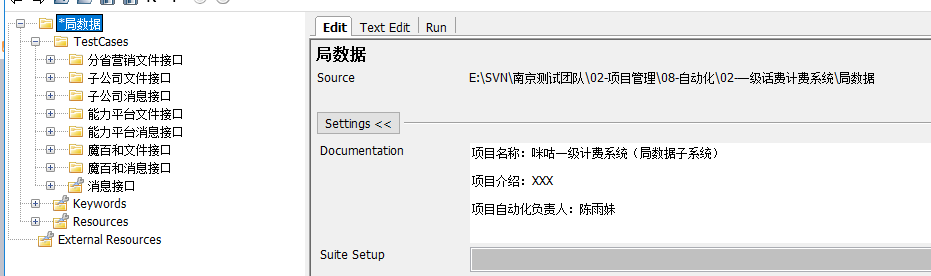


1.4 每个接口对应一个测试套文件，下面包含多个测试用例，测试套及测试用例按规范命名。

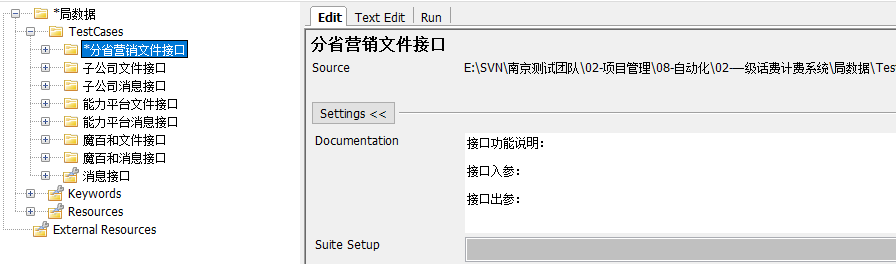


### 规范2：Documentation使用规范

2.1 工程根目录（项目名称）的Settings里必须包含Documentation，需要写清项目名称、项目介绍、项目自动化接口人。



2.2 测试套文件的Settings里必须包含Documentation，需要写清接口功能说明、接口入参、接口出参；



2.3测试用例的Settings里必须包含Documentation，需要写清用例名称、用例编号、用例版本、用例级别、用例类型、用例作者、前置条件、操作步骤、预期结果。

Documentation demo：





2.4自定义关键字的Settings里必须包含Documentation，需要写清关键字的使用场景、出入参描述，必要时给出使用举例。



### 规范3：测试套/测试用例/变量命名规范

3.1测试套的命名：**接口名**，例如余额鉴权接口、撤销扣费接口等。

3.2测试用例的命名：**编号\_\_入参描述\_返回结果描述**，或者**编号\_\_接口请求描述\_返回结果描述**，编号从001开始，编号需为三位，后面是2根下划线，如果存在多个入参，用例名称中只需要描述关键参数的取值即可。

3.3全局变量命名：必须以G\_开头，单词首字母大写，只有一个单词的话也可以全部大写，例如${G\_Server}或${G\_SERVER}都可以，但有两个及以上单词时建议使用${G\_ServerUrl}.

3.4套件范围的变量命名：建议单词首字母大写，单词之间下划线分割，例如Login\_Address.

3.5测试用例和关键字中的变量命名：建议首个单词首字母小写，其他单词首字母大写，例如userName、loginAddress.

3.6全局变量的定义必须放到Resources目录下的Global\_Variables.robot文件。

3.7测试用例步骤里的字符串尽量使用变量代替，所有变量的命名必须要有意义。

### 规范4：关键字封装及使用规范

4.1关键字命名：每个单词首字母大写，单词之间空格分割。

4.2上层关键字的定义必须在KeyWords目录下的资源文件里，禁止在测试套中定义关键字。

4.3 KeyWords目录下建议按照系统特性创建多个资源文件，分别存放对应特性的自定义关键字。

4.4多个用例重复使用到的业务流程，建议封装成上层关键字，供所有用例调用。

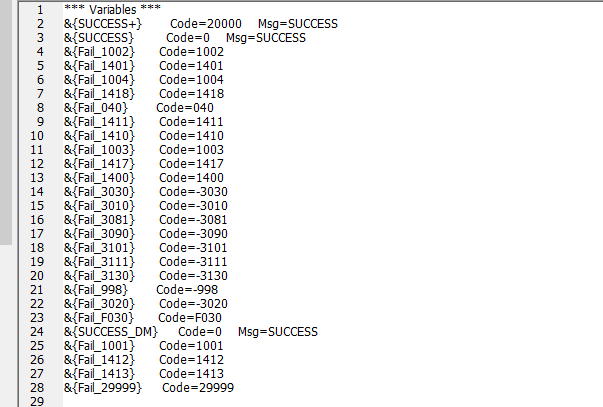
4.5 自动化脚本要尽量使用有意义的业务逻辑关键字。不建议脚本中的每个步骤都使用测试库提供的底层关键字。脚本步骤太多时，要考虑是否能够通过封装逻辑关键字的方式进行优化。

### 规范5：资源文件使用规范

5.1 Resources目录下按系统特性创建多个资源文件，分别存放对应特性的变量，文件名建议为*特性英文缩写\_Variable.robot*，变量名建议以*特性英文缩写\_*开始。

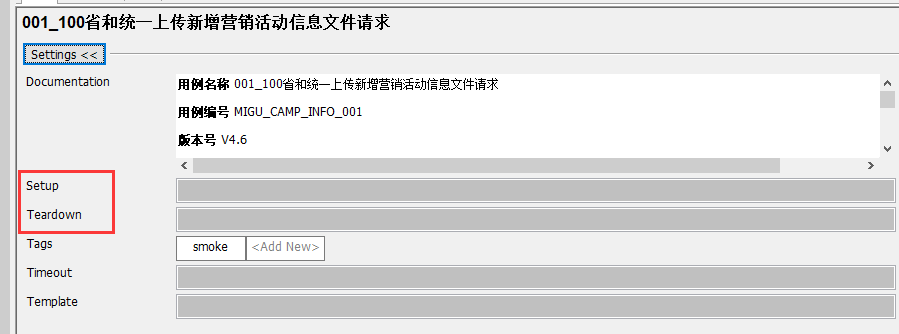
5.2 全局变量存放在*Global\_Variable.robot*文件中，变量名以G\_开始。

5.3 系统接口返回码及提升信息建议保存在一个资源文件里，文件名ReturnCodeMsg.robot，返回码定义成字典格式的变量。



### 规范6：Setup和Teardown的使用规范

6.1 Setup完成测试用例的准备工作，如数据预置，Teardown进行测试用例结束后的清理工作，例如数据库清理、测试环境恢复等，如果存在某些公共的环境准备或清理操作，建议封装成关键字，测试用例使用Setup和Teardown，避免多次重复编写一样的流程。



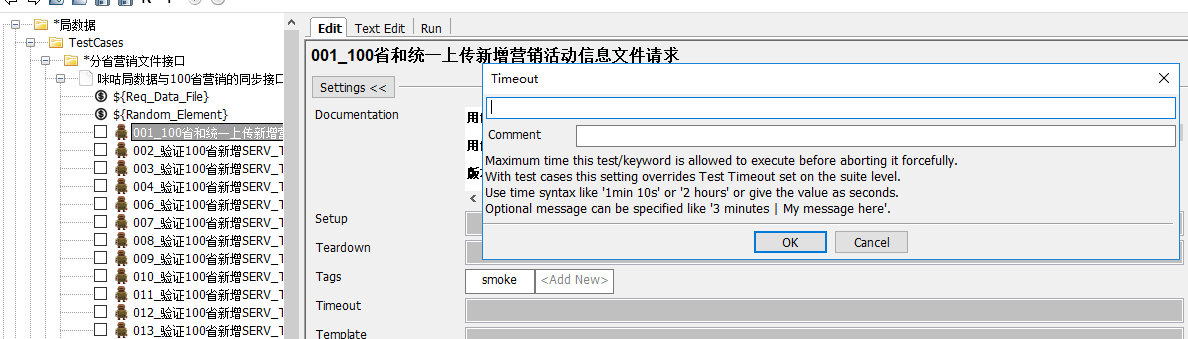
6.2 Suite Setup和Suite Teardown是某个测试套的前置和后置操作，Suite Setup在该测试套的所有测试用例执行前执行，Suite Teardown在该测试套的所有测试用例执行结束后执行。



### 规范7：Sleep和Timeout的使用规范

7.1 尽量避免使用Sleep关键字（调试过程除外），可以使用Wait Unitl xxx等替代。

7.2 测试套文件建议设置超时时间Timeout，防止因为系统异常，导致测试用例长时间挂死。该超时时间用于控制测试套下每个测试用例的最大执行时长，测试用例单独设置的Timeout会覆盖测试套的Timeout。



### 规范8：自动化用例校验点规范

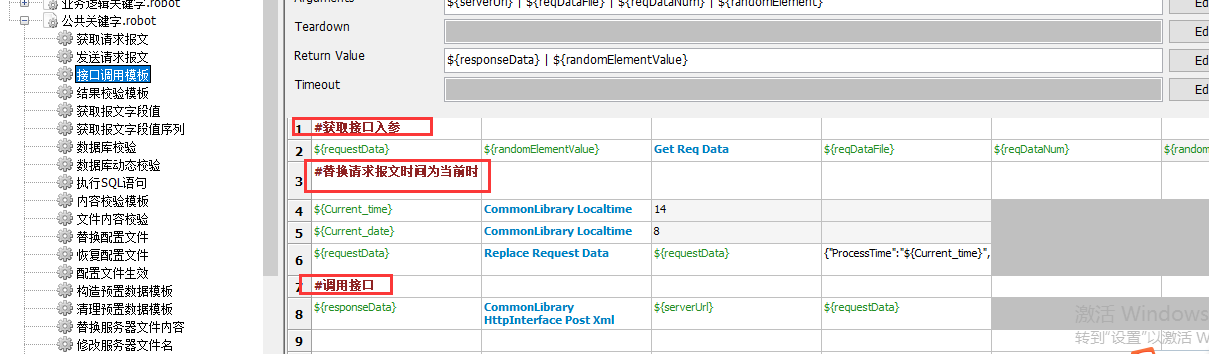
8.1所有的自动化用例必须包含对返回结果校验的步骤，其中包含响应报文的校验、数据库校验及其他业务相关的校验。

8.2接口响应报文的校验点包括状态、提示信息、错误码、执行结果等。

8.3查询类接口，需要校验接口返回内容与数据库实际值是否一致，新增修改删除类接口，需要校验接口执行成功或失败后，数据库相关表的内容与预期结果是否一致，执行类接口需要校验数据库相关表的信息记录是否正确。

### 规范9：关于注释的规范

9.1 所有上层关键字的封装逻辑必须包含注释，说明每个步骤的目的。



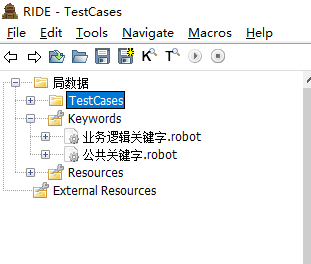
9.2 如果自动化脚本包含多个步骤，在关键字名称意义不够明确的情况下，步骤之间建议添加注释，利于后续维护。

9.3 关键字必须包含使用举例，关键步骤必须包含注释。

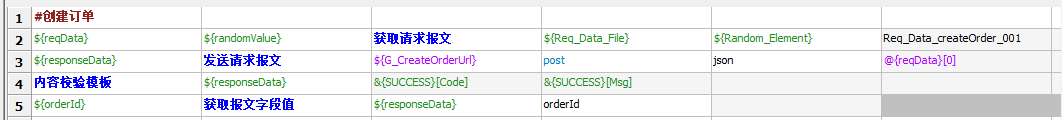
## 3 自动化脚本写作指导

### 3.1常用关键字使用说明

1. 所有的公共关键字定义全部放在Keywords目录下的**公共关键字.robot**文件里。
2. 所有和业务逻辑相关的关键字全部放在Keywords目录下的业务逻辑关键字.robot文件里。



1. 获取请求报文：包含三个个入参：${Req\_Data\_File} | ${Random\_Element} | @{Req\_Data\_Num}，两个出参：${req\_data\_list} | ${randomElementValue}。



入参：

* ${reqDataFile} 接口报文配置文件，.conf后缀，建议直接引用测试套变量${Req\_Data\_File}。
* ${randomElement} 报文里需要被替换成随机数的字段，例如报文里的ORDER\_ID需要替换成22位的随机数，ProcessTime需要替换成14位的随机数，变量值为{"ORDER\_ID":"0000000000000000000000-9999999999999999999999","ProcessTime":2017000000000-20179999999999"}，报文配置文件里的字段值需要写成变量的形式：${ORDER\_ID}，${ProcessTime}。
* @{Req\_Data\_Num} 报文配置文件里的报文编号，例如Req\_Data\_01，可以同时指定多个。

出参：

* ${req\_data\_list} 返回的请求报文list
* ${randomvalue} 请求报文里替换的随机数的值，例如：{"ORDER\_ID":"92345678912345643213456","ProcessTime":"20171234543445"}

1. 更新报文数据：包含两个入参：${requestdata} | ${fields}，出参一个：${requestdata}



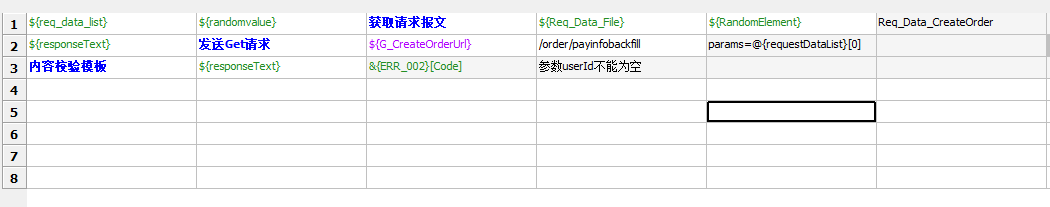
入参：

* ${requestdata}：json格式的报文字符串
* ${fields}：jsonpath语法格式的字符串，例如$.appId:100,$.userId:10001，表示将报文中的appId替换为100，userId替换为10001

出参：

* ${requestdata}：更新后的报文字符串

1. 发送get请求：包含四个入参${URL} | ${URI} | ${params}=${None} | ${headers}=${None}，一个出参：${responseText}



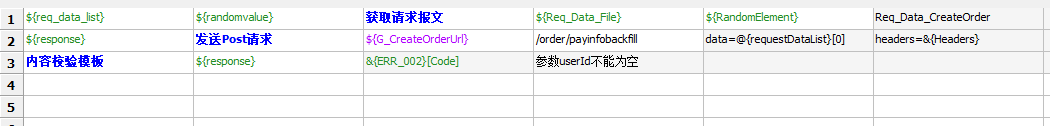
入参

* ${URL}：服务器URL地址
* ${URI}：服务器资源标识符
* ${params}：URL参数
* ${headers}：消息头

出参：

* ${responseText}:响应报文

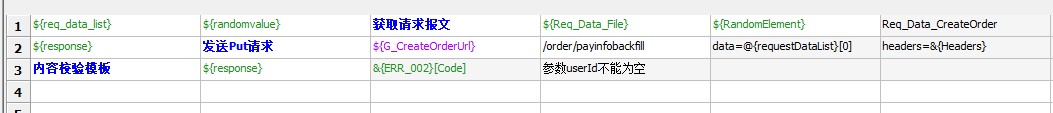
1. 发送Post请求，包含五个入参：${URL} | ${URI} | ${data}=${None} | ${headers}=${None} | ${params}=${None}，一个出参：${responseText}



入参

* ${URL}：服务器URL地址
* ${URI}：服务器资源标识符
* ${data}：请求数据
* ${params}：URL参数
* ${headers}：消息头

1. 发送Put请求，包含五个入参：${URL} | ${URI} | ${data}=${None} | ${headers}=${None} | ${params}=${None}，一个出参：${responseText}



入参

* ${URL}：服务器URL地址
* ${URI}：服务器资源标识符
* ${data}：请求数据
* ${params}：URL参数
* ${headers}：消息头

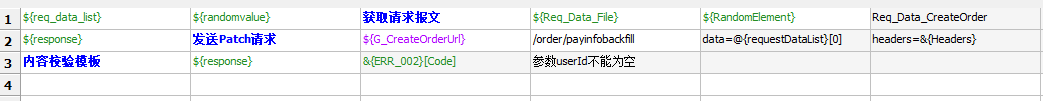
1. 发送Delete请求：包含五个入参：${URL} | ${URI} | ${data}=${None} | ${headers}=${None} | ${params}=${None}，一个出参：${responseText}

入参

* ${URL}：服务器URL地址
* ${URI}：服务器资源标识符
* ${data}：请求数据
* ${params}：URL参数
* ${headers}：消息头



1. 发送Patch请求：包含五个入参：${URL} | ${URI} | ${data}=${None} | ${headers}=${None} | ${params}=${None}，一个出参：${responseText}



入参

* ${URL}：服务器URL地址
* ${URI}：服务器资源标识符
* ${data}：请求数据
* ${params}：URL参数
* ${headers}：消息头

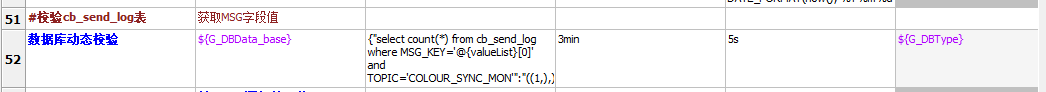
1. 内容校验模板：包含两个入参：${result} | @{args}



入参：

* ${result} ：待校验的对象
* @{args}：需要校验的字符串

1. 数据库动态校验：包含五个入参:${DBConnStr} | ${sqlStatement} | ${retry} | ${interval} | ${DBType}=cx\_Oracle



入参：

* ${DBConnStr}：[数据库连接串，例如username/passwd@127.0.0.1:1521/oracle](mailto:数据库连接串，例如username/passwd@127.0.0.1:1521/oracle)
* ${sqlStatement}：数据库查询SQL语句及预期结果，字典格式，例如

{"select t1.contentid from TG\_FULL\_CHARGE\_TMP t1 where order\_id = '8739619410065812898203' union all select t2.contentid from TG\_FULL\_CHARGE\_his\_07 t2 where order\_id = '8739619410065812898203'":"[('30018283746272737646',)]"}

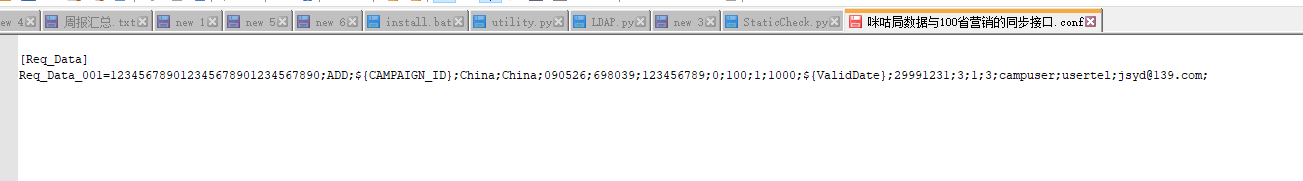
* ${retry}：可以设定为超时时长或重试次数，例如5 min或10 times
* ${interval}：重试间隔，例如30s
* ${DBType}：数据库类型，默认值是cx\_Oracle，也支持其他类型的数据库，例如mysql

### 3.2接口报文配置文件说明

1. 一个接口对应一个.robot后缀的测试套文件，同时对应一个.conf后缀的配置文件，配置文件里包含该接口下所有自动化用例执行需要的报文，配置文件放到和测试套相同的目录下，例如：



1. 由于一个测试套往往是验证一个接口的功能，所以报文结构是固定的，不同的脚本往往只是调整报文中的某个或某几个参数的值，因此，我们在.conf 文件中仅保存一条完整的报文，所有脚本发送的报文都在该报文基础上修改。脚本获取同一条报文之后，通过更新报文数据关键字将报文中的某个或某几个字段替换成特定值，具体方法见：3.5基于报文模板的写作方法介绍



1. 每个测试套下定义一个变量${Req\_Data\_File}，保存该接口的报文配置文件路径，例如${CURDIR}${/}产品定价申报信息接口.conf，其中${CURDIR}为RF内置变量，表示当前测试套所在目录。

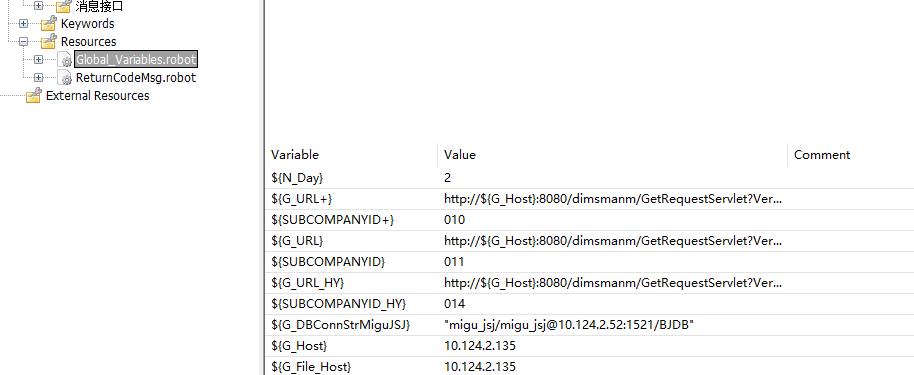


1. 测试报文里的某些字段由于业务限制，每次执行都不能重复，针对这类字段，配置文件里可以将字段值替换为变量，例如${TransactionID} ，再定义测试套变量${Random\_Element}，变量值为字典格式，例如：

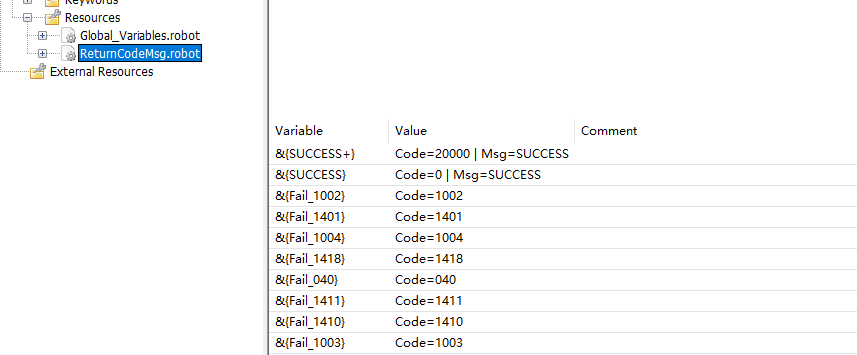
{"TransactionID":"00000000-99999999","ReqDateTime":"0000-9999"}，执行获取请求报文关键字的时候，引入${Random\_Element}这个入参，关键字从配置文件获取到报文后，会使用指定长度的随机数替换报文中的${TransactionID}.

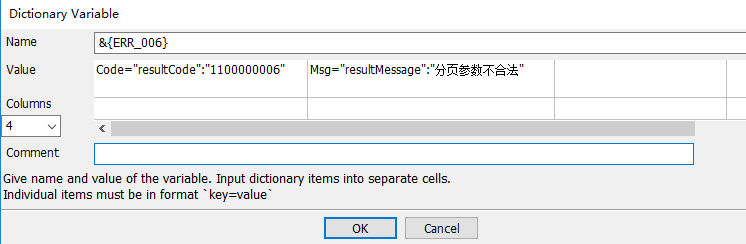
### 3.3资源文件的使用说明

1. 全局变量的定义文件：Resources目录下的Global\_Variables.robot文件。
2. IP地址、端口、用户名、数据库信息等必须定义为全局变量，测试套引入该资源文件即可使用定义的全局变量。



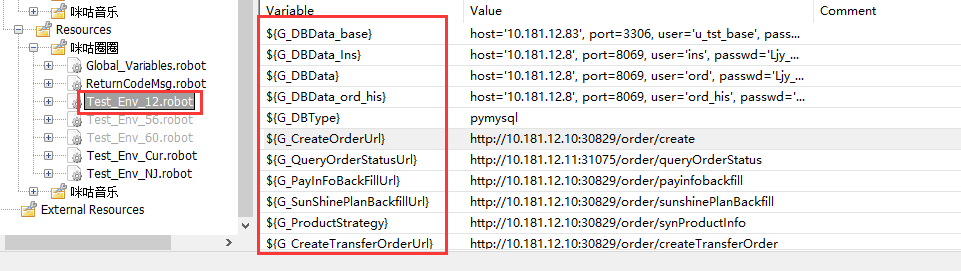
1. 接口返回码及提示信息定义文件：Resources目录下的ReturnCodeMsg.robot文件，该文件以字典变量的格式存放所有可能的接口返回码及提示信息。

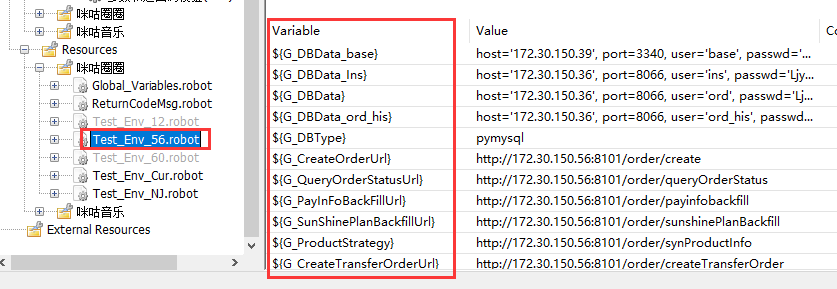




### 3.4脚本如何适配多套测试环境

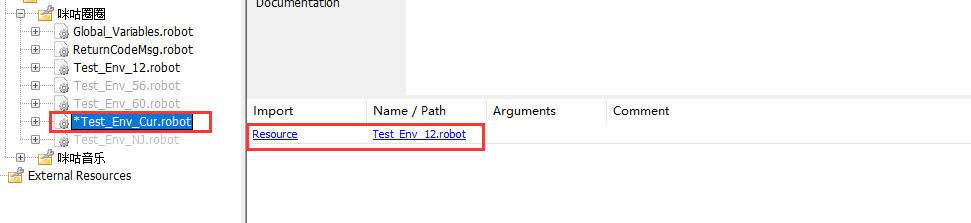
以订单中心为例，订单中心一共有4套环境，则创建4个资源文件，Test\_Env\_001-004.robot,文件中存放和测试环境相关的所有变量，例如IP地址、端口等，变量名称在4个文件中要求完全一致，然后创建资源文件Test\_Env\_Cur.robot，该文件引用当前生效的测试环境资源文件，例如Test\_Env\_001.robot，所有的测试套全部引用Test\_Env\_Cur.robot这个文件，如果需要切换测试环境为Test\_Env\_002.robot，只需要修改Test\_Env\_Cur.robot文件中引用的文件名即可，所有的测试套不需要做任何修改。





测试套和测试环境的关联：





### 3.5基于报文模板的脚本写作方法

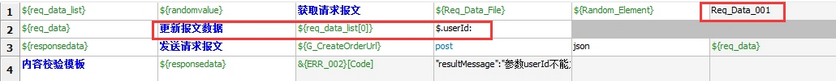
（1）背景：目前每个自动化脚本在.conf 文件中都会对应一条报文，如果一个测试套有100个脚本，那.conf 文件可能会记录100条报文，

如果发生需求变更，例如新增或修改了一个字段，带来的维护工作量很大，因此，我们提出了基于报文模板的写作方法。

（2）写作方法说明：一个测试套往往是验证一个接口的功能，所以报文结构是固定的，不同的脚本往往只是调整报文中的某个或某几个参数的值，

因此，我们在.conf 文件中仅保存一条完整的报文，所有接口都在该报文基础上修改。

改写后的自动化脚本如下：



所有的脚本都获取同一条报文，获取之后通过更新报文数据关键字将报文中的某个或某几个字段替换成特定值，例如该脚本要验证userId为空的场景，按照JsonPath语法输入参数即可。

如果要改变多个字段的值，使用逗号分隔，例如$.userId:,$.orderItemRequests[0].isRoot:0（支持替换list里的某个参数）

如果要删除报文中的某个字段，参数格式为$.userId:None

1. 优化效果评估

随机看了能力平台订单中心的一些测试套，每个测试套包含的脚本数量都在100以上，.conf文件中的报文巨多，优化后，大部分的.conf文件可以仅保留一条报文，应对报文结构的变更，改动量极小。

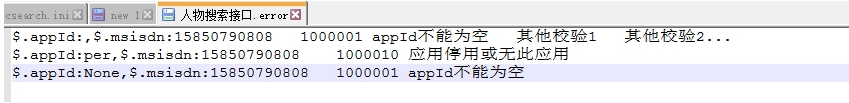
90%以上的脚本都是只改变报文中某一个字段的值，更新报文数据关键字的修改量也很小，综合评估，该写作方法不仅可以降低脚本维护的工作量，对脚本写作效率也有一定提升。

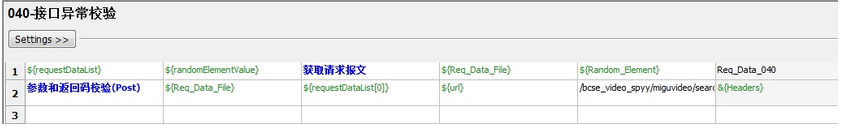
### 3.6接口字段异常校验场景脚本写作方法

接口测试存在很多字段异常校验的场景，例如字段为空、字段超长、字段取值错误等，现在的测试用例和自动化脚本都是一个异常场景对应一条测试用例和一个自动化脚本，而这些用例或脚本的步骤基本一样，鉴于此，我们提出一种新的脚本写作方法，简化异常校验场景用例和脚本的写作，具体说明如下。

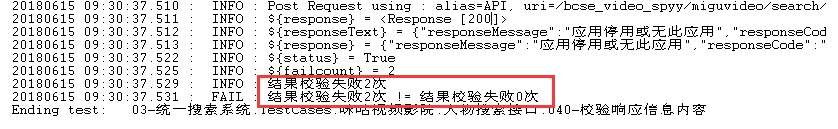
1、针对某个接口的字段异常校验，使用一个测试用例完成描述，涉及多个字段多种场景的异常校验，全部写在一个测试用例里。

2、针对该测试用例，在测试套目录下新增 接口名.error 文件，文件格式如下，有多少个异常场景，文件包含多少行，第一列表示报文中的异常场景，例如appId为空，appId为错误值、删除报文中的appId，如涉及多个字段的异常，使用英文逗号隔开，$表示json报文的根元素；第二列开始为该场景下响应结果要校验的内容，例如错误码，错误描述等，使用tab键隔开。



3、封装了参数和返回码校验(Post) 和参数和返回码校验(Get) 两个关键字，分别用于post和get两种类型的请求，最终的脚本写作只要以下两步：

4、脚本执行过程中，会循环将error文件中的异常信息替换到报文中，发送get或post请求，然后进行响应结果的校验，中间某次校验失败，不会终止脚本执行，脚本最后会打印失败总次数。只要有校验失败发生，整个脚本的结果就是Fail。



5、失败场景快速定位方法，每次校验失败，都会打印ERROR级别的日志：结果校验失败，打开log文件，Test Execution Errors 部分会集中显示这类ERROR，点击链接即可快速定位到失败的场景。

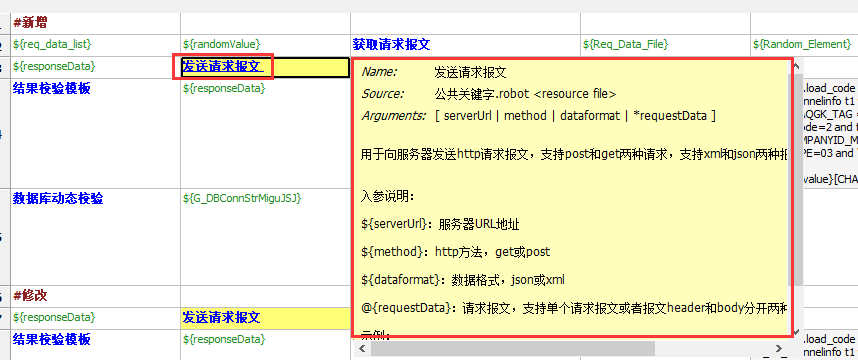




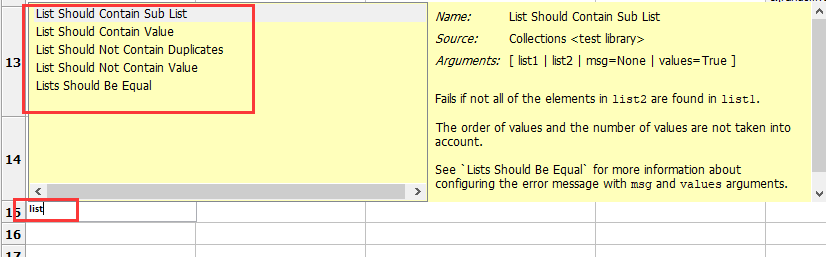


### 3.7脚本写作小技巧

1. 查看关键字帮助信息，鼠标选中关键字，长按“Ctrl”键，用鼠标点击弹出页面，既可以查看帮助信息



1. 关键字联想方式，在我们不确定需要用到的关键字的准确名称时，可以输入部分内容，同时按住Ctrl+Alt+空格键，会弹出页面，并在页面中展示和输入内容相关的关键字。



## 4 自动化脚本与用例互转工具使用指导

### 4.1工具安装：

1、运行cmd，执行pip uninstall robotframework-ride，卸载本地的RIDE；

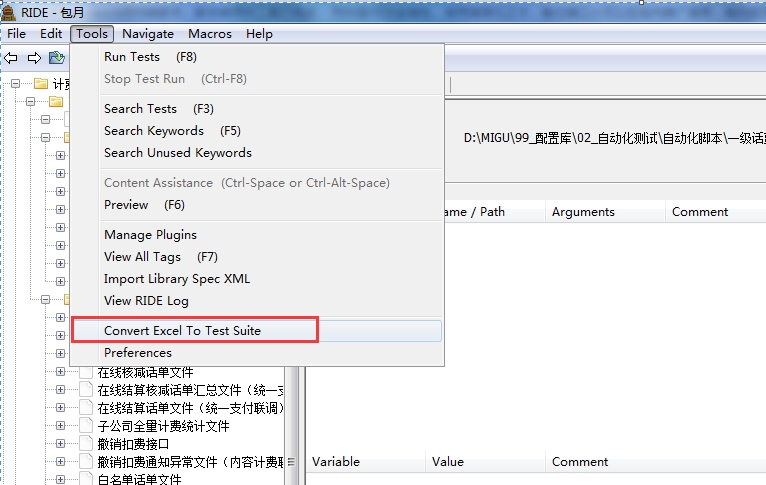
2、执行pip install xlrd、pip install xlwt、pip install xlutils安装依赖库；

2、解压附件，cmd进入解压后的目录，执行python ./setup.py install，安装RIDE；

 安装包：

### 4.2用例转脚本工具使用：

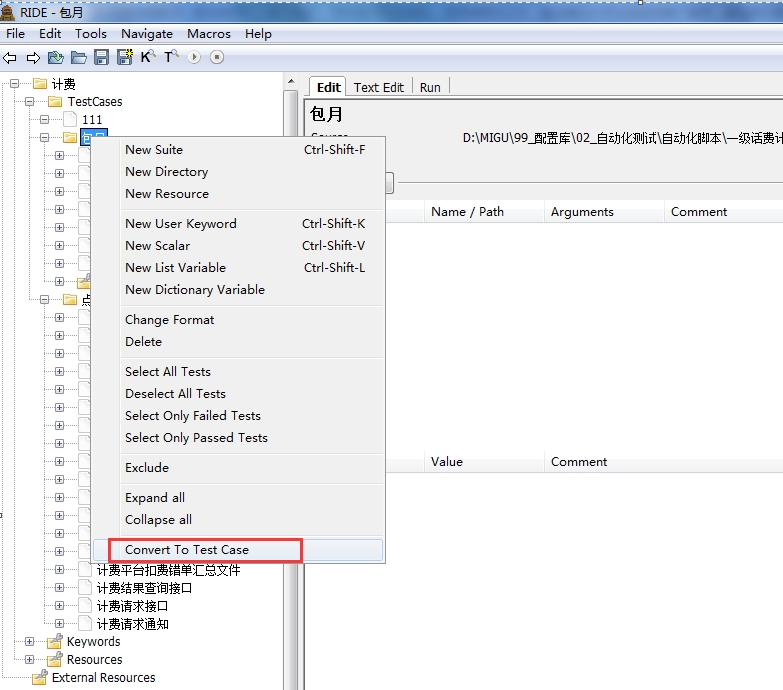
1、安装完成后，启动RIDE，点击Tools--Convert Excel To Test Suite，选择可以导入CTMP平台的Excel用例文件，即可自动转换成测试套，测试套会按规范写入脚本的documentation信息。



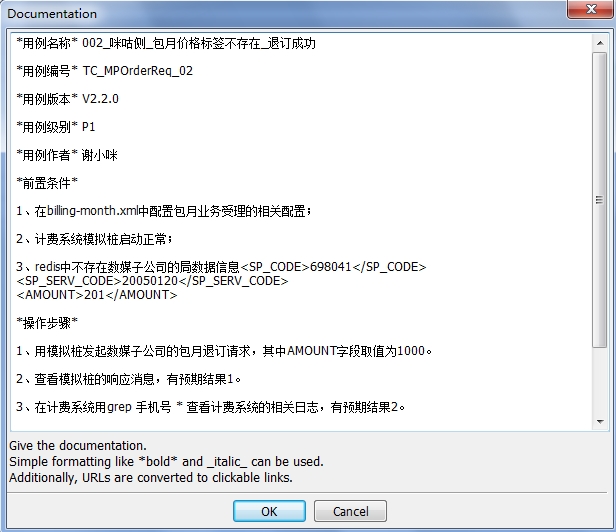
使用场景：Excel测试用例完成之后，使用该工具进行转换，生成对应的测试套文件及目录，测试人员使用RIDE打开生成的文件后，直接添加脚本步骤即可。

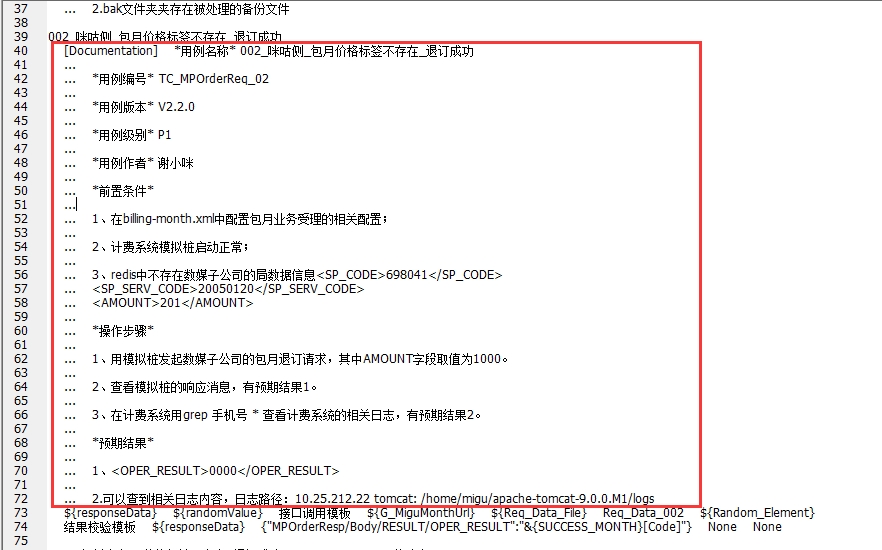
### 4.3脚本转用例工具使用：

（1）启动RIDE，打开某个项目的自动化工程，选中某个自动化脚本目录或者单个测试套—右键—点击Convert To Test Case，即可将选中的目录或测试套转换为excel用例文件，用例文件的内容从自动化脚本的documentation中提取，目前会提取：用例名称、用例编号、用例版本、用例级别、用例类型、用例作者、前置条件、操作步骤、预期结果 这9个字段的信息。



(2)、脚本documentation写作格式：





使用场景：根据需求直接写了自动化脚本，希望通过自动化脚本转换生成excel测试用例。

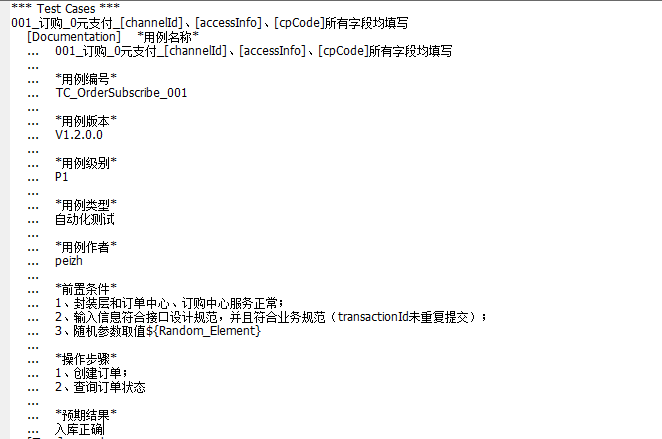
## 5 自动化脚本规范扫描工具使用指导

为了保证自动化脚本的规范性，我们开发了规范扫描工具，自动识别部分规范问题。并接入jenkins自动检查。

### 5.1 扫描规则

规则1：自动化脚本的Documentation要满足规范（强制）

Documentation的格式要求，必须包含： 用例名称、用例编号、用例版本（或版本号）、用例级别、用例类型、用例作者、前置条件、操作步骤（测试步骤或测试歩骤）、预期结果，8个标题。并且标题下的内容不能为空。如下图：



规则2：自动化脚本非特殊情况不能使用Sleep（强制）

除了一些特殊业务需求外，脚本中不能使用sleep

规则3：自动化脚本步骤不能为空（强制）

禁止提交步骤为空的自动化脚本。

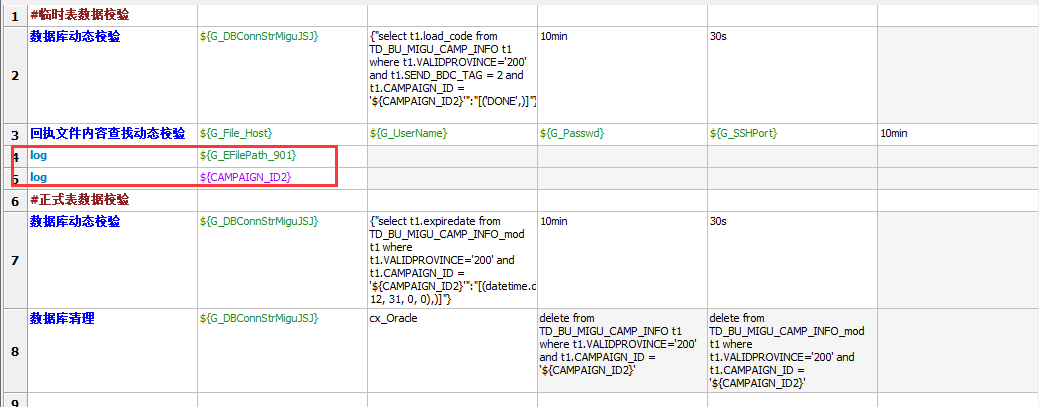
规则4：自动化脚本不能缺少结果校验步骤（强制）

脚本中不能缺少校验步骤，有效的校验关键字包含：校验|验证|比对|比较|检查|检验|Should Be|Should Not Be



规则5：自动化脚本步骤不能包含调试用的Log（建议）

提交svn或git的脚本，建议不要包含Log步骤。



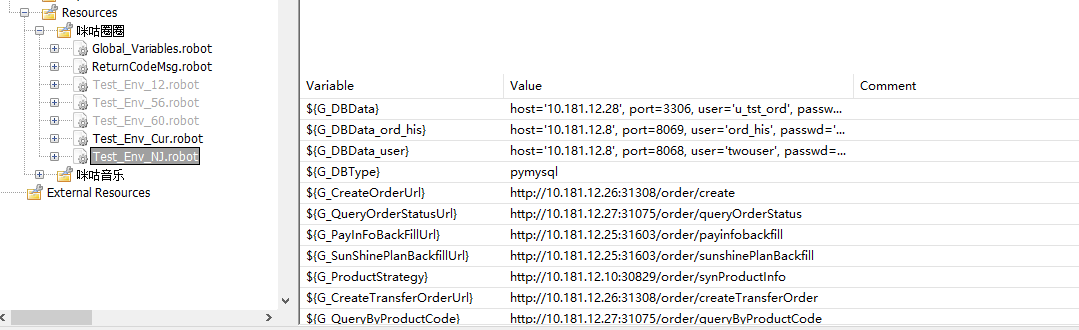
规则6：逻辑关键字要包含帮助信息、出入参描述、使用举例（强制）

新增的关键字必须要包含帮助信息、出入参描述、使用举例等，便于大家的理解和使用。



规则7：URL、IP、Port、UserName等不能写死在脚本中（强制）

脚本中用到的URL、IP、Port、UserName等不能写死在脚本中，否则当环境改变时，就需要大量修改脚本，带来额外的工作。要以全局变量的形式，保存在资源文件中，脚本中引用全局变量，如下图：



规则8：自动化脚本步骤数超过10，建议优化（建议）

此规则为建议性规则，主要目的是希望大家尽量使用公共关键字或适当封装业务逻辑关键字，简化脚本步骤，方便理解以及脚本的维护。

### 5.2 工具使用

安装依赖库：

pip install pyh

pip install xlutils

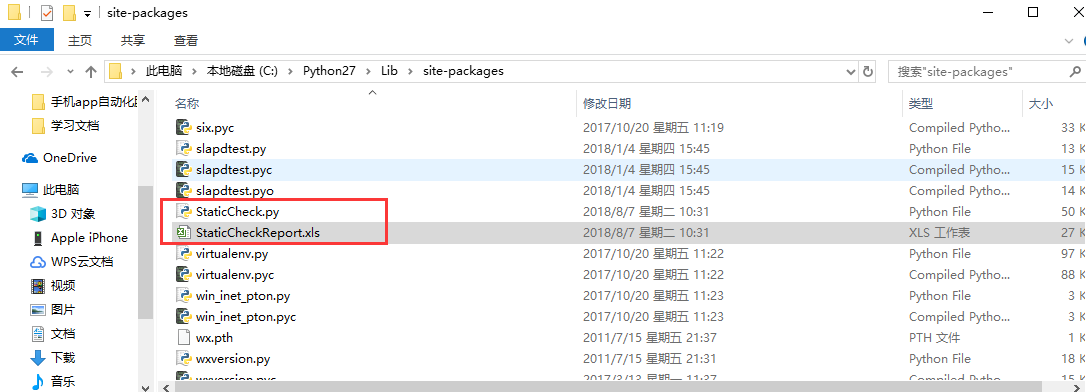
pip install xlwt

注意：在线pyh包可能会安装不成功，可以把安装包下载到本地，解压附件，cmd进入解压后的目录，执行python ./setup.py install，安装pyh包。

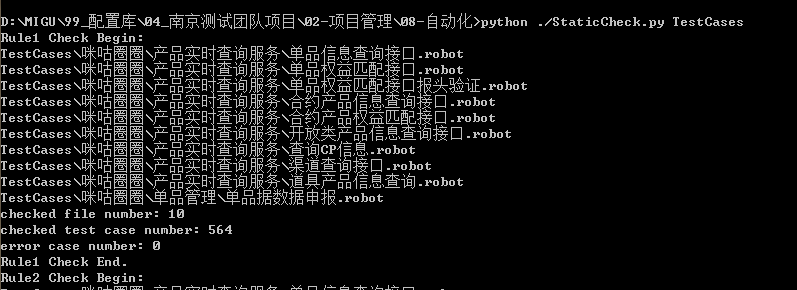


把StaticCheck.py和StaticCheckReport.xls复制到本地任意路径下



cmd进入该目录，执行：python ./StaticCheck.py 测试套所在路径（路径不能包含中文），扫描结束后会生成html和xls格式的报告。





## 6 版本控制使用指导

本地写的代码需要上传到远程仓库，方便版本管理及大家协同工作。可使用svn或者git。

Git和svn使用网上资料较多，下面是我们自己总结的git使用教程：



最后，特别注意：

* 写脚本的时候不能依赖环境已有数据，脚本开始的时候在setup里面构造数据，完毕再teardown里面清理数据；保证环境迁移脚本也能通过；
* 模板里面最后demo只做了返回码校验，没做业务校验，这块需要根据业务自己需要校验的字段收到返回后查数据库校验。这块写的时候不清楚可以联系我们；

# III自动化测试应用阶段

目前的自动化测试主要用于提测版本的冒烟测试和回归测试，都需要在Jenkins上触发测试任务的执行。相关任务的创建和维护请联系Jenkins接口人毛润生。

## 1 自动化测试与Jenkins集成指导

### 1.1 权限申请

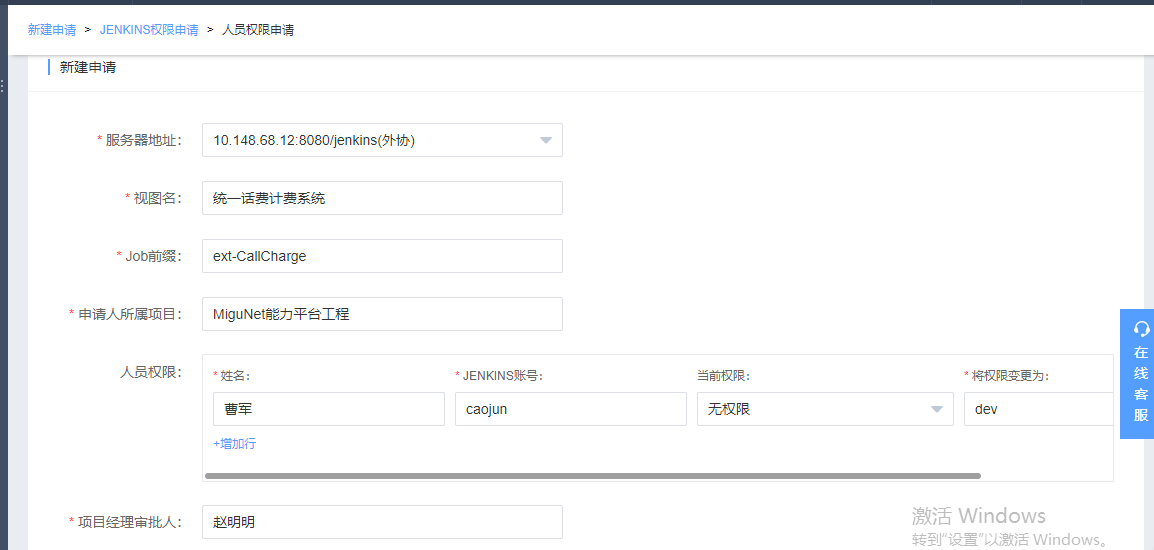
<http://10.148.68.12:8080/jenkins/>是咪咕所有项目共用的Jenkins环境，使用前需要申请Jenkins权限和VPN权限，权限申请统一在SPMS进行，权限申请通过后，使用手机号yw账号登陆VPN，浏览器打开<http://10.148.68.12:8080/jenkins/>，即可访问Jenkins环境。



Jenkins权限申请：

dev:查看和执行job权限

Master：编辑，新增，修改job权限。



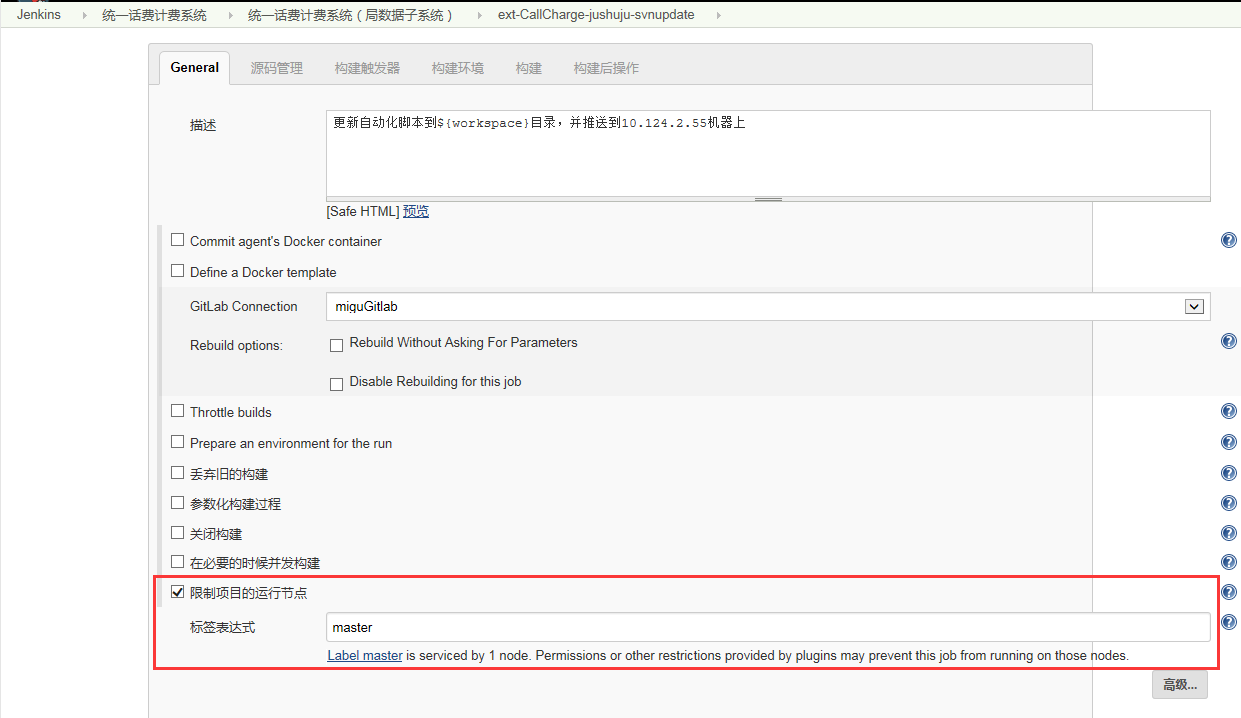
VPN权限申请：



### 1.2 svnupdate任务配置

以局数据项目为例，svnupdate任务配置如下：

1、限制任务在Jenkins Master执行。



2、源码管理选择Subversion

Repository URL：局数据脚本所在的svn路径

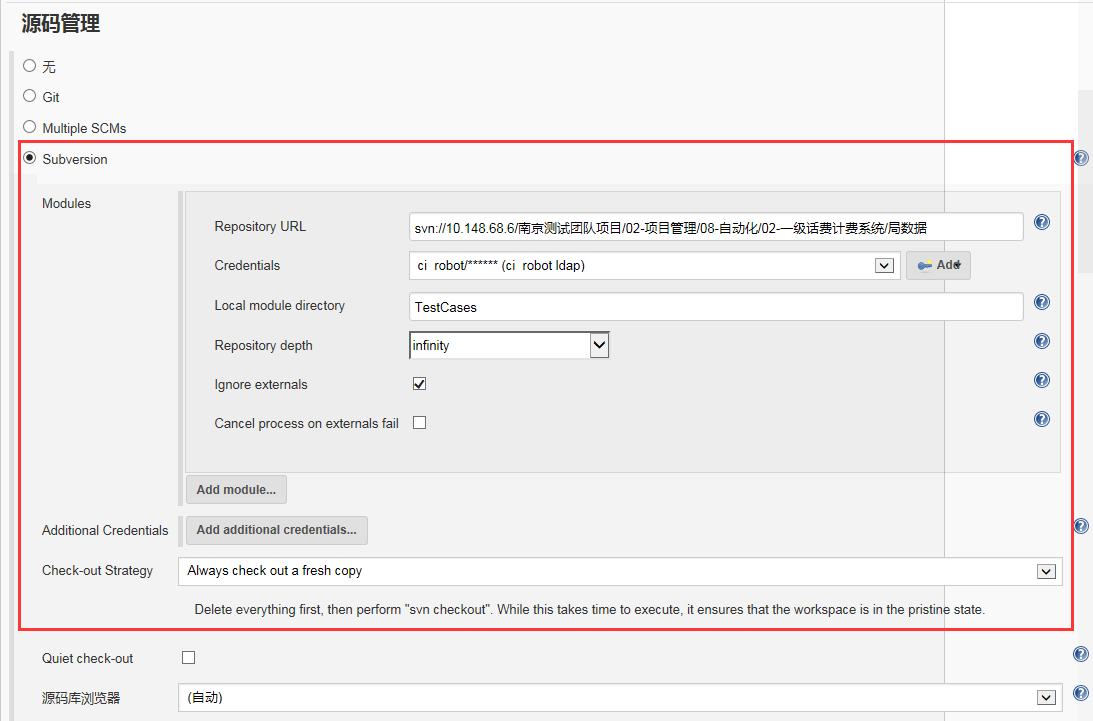
Credentials：选择ci robot/\*\*\*\*\*\*(ci robot ldap)

Local module directory:脚本checkout到${workspace}下的那个目录

Repository depth：infinity

Ignore externals：勾选

Check-out Strategy：Always check out a fresh copy



3、构建触发器选择定时构建，下图配置为每周三、五凌晨1点半自动构建。

日程表参数解释：

第一个参数代表的是分钟 minute，取值 0~59；

第二个参数代表的是小时 hour，取值 0~23；

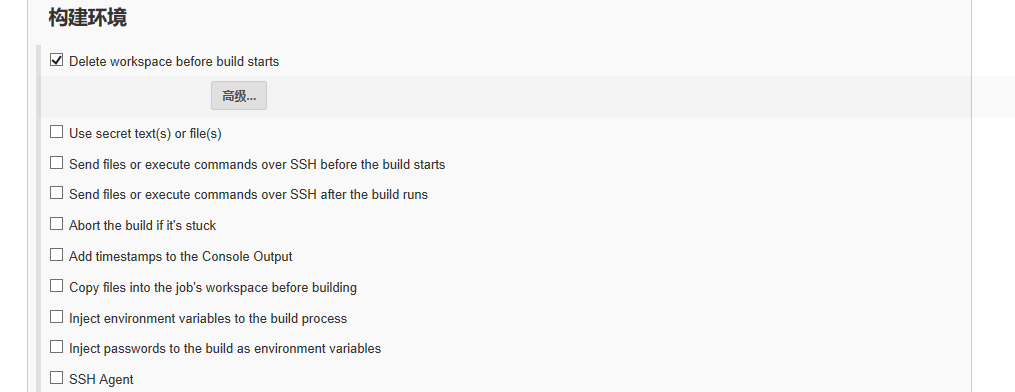
第三个参数代表的是天 day，取值 1~31；

第四个参数代表的是月 month，取值 1~12；

最后一个参数代表的是星期 week，取值 0~7，0 和 7 都是表示星期天。



1. 构建环境勾选：Delete workspace before build starts



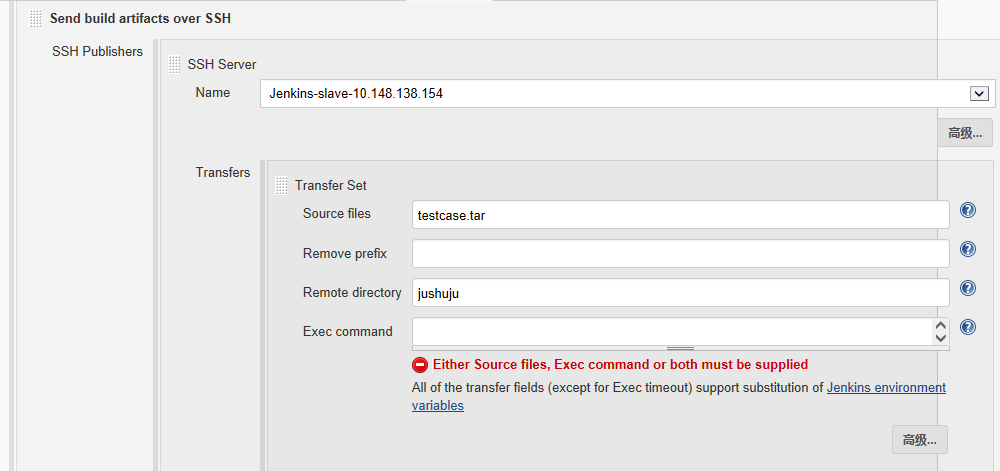
5、 构建步骤选择Execute shell，命令栏填写下图内容。

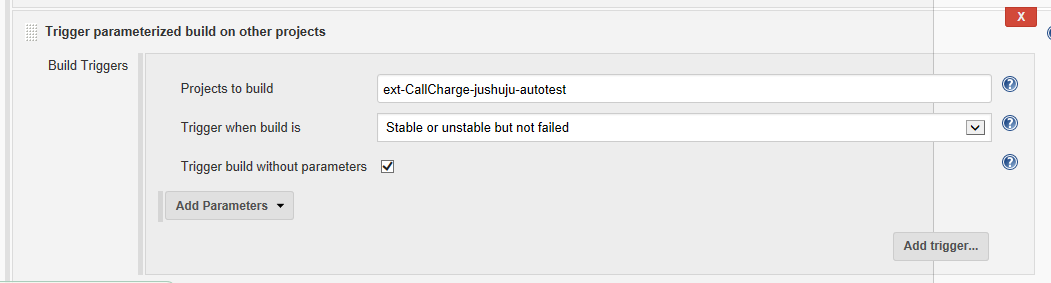
打包${workspace}目录下的脚本为testcase.tar



6、 构建后操作

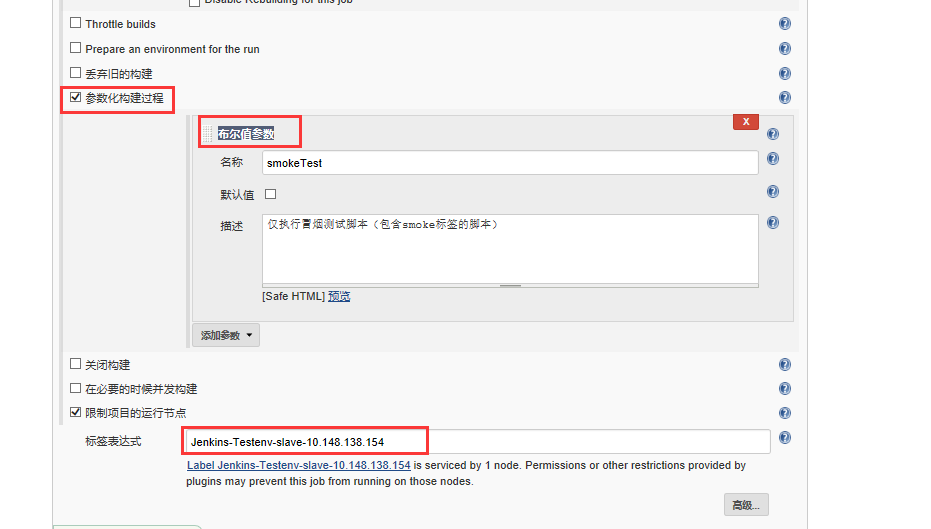
增加构建步骤Send build artifacts over SSH，和Trigger parameterized build on other projects结果输出目录设置，如图：





### 1.3 自动化测试任务的配置

1、限制任务在Jenkins测试区 slave机器上执行，勾选参数化构建，添加布尔参数，名称为smokeTest.



2、源码管理，选择无。



3、构建触发器，选择定时构建：

构建触发器选择定时构建，下图配置为每周三、五凌晨2点自动构建。

日程表参数解释：

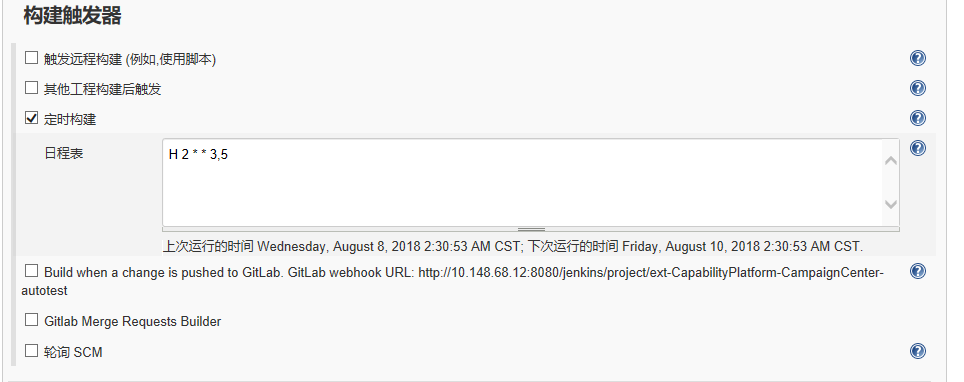
第一个参数代表的是分钟 minute，取值 0~59；

第二个参数代表的是小时 hour，取值 0~23；

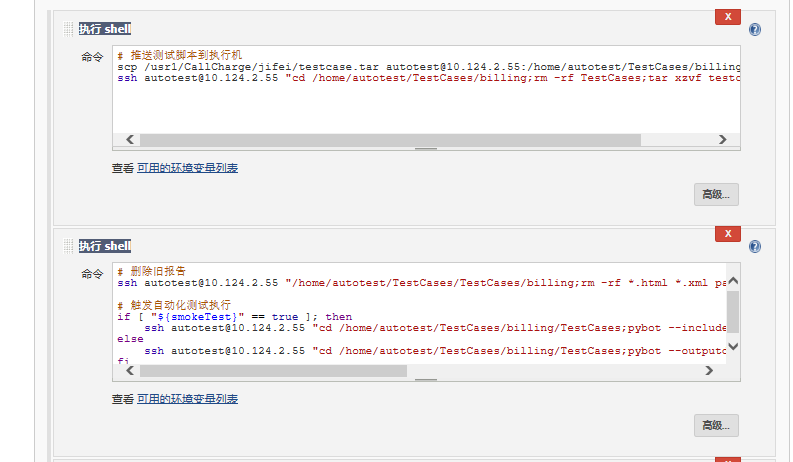
第三个参数代表的是天 day，取值 1~31；

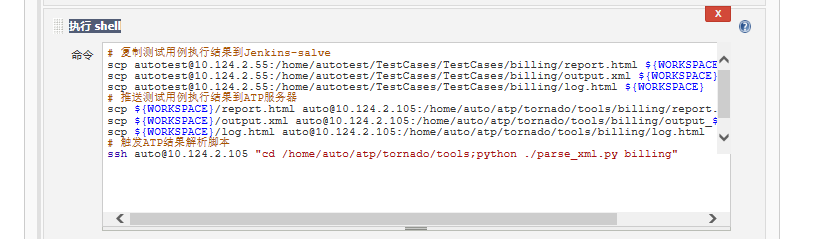
第四个参数代表的是月 month，取值 1~12；

最后一个参数代表的是星期 week，取值 0~7，0 和 7 都是表示星期天。



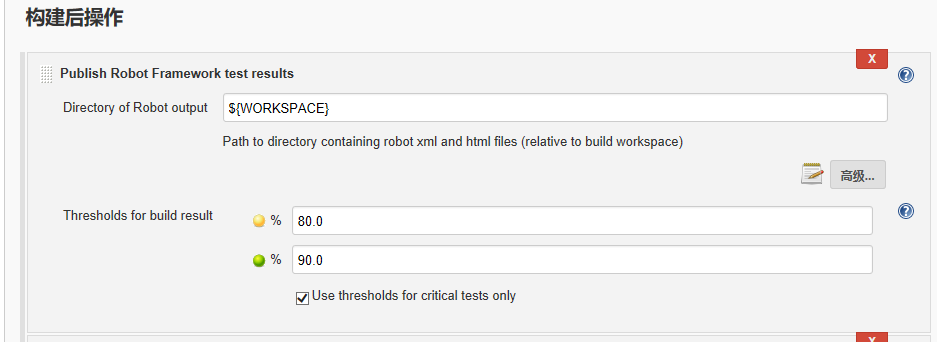
4、构建步骤选择Execute shell，命令栏填写下图内容

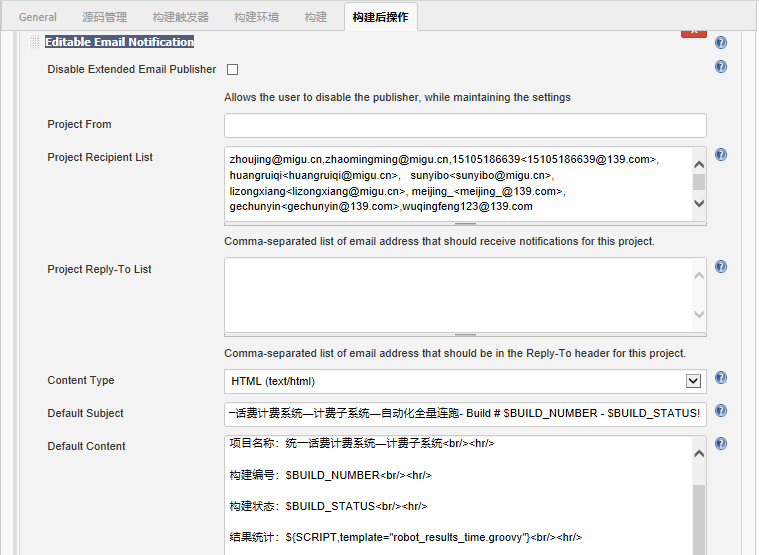


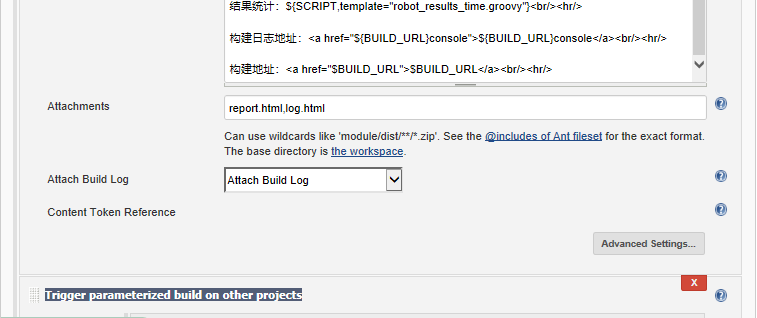


5、构建后操作

构建步骤选择Publish Robot Framework test results，配置如下：







### 1.4 测试任务执行

单击某个测试任务右侧的小箭头，点击build with parameters



不勾选smokeTest为执行全量自动化脚本，勾选smokeTest为仅执行冒烟测试脚本（包含smoke标签的脚本）



## 2 ATP平台使用指导

登录地址：http://10.148.133.134:8080/login

需要提前申请10.148.133.134的VPN权限，权限申请审批通过后，使用手机号yw的账号登陆VPN。

该平台包含以下4块功能：

1. ATP知识库
2. 自动化建设数据
3. Migu Library
4. Robot Framework



### 2.1 ATP知识库

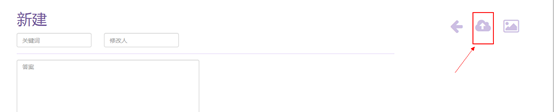
1. 点此进行搜索关键字：



1. 点此新增数据：



1. 填写完毕后点此上传：修改人写自己名字

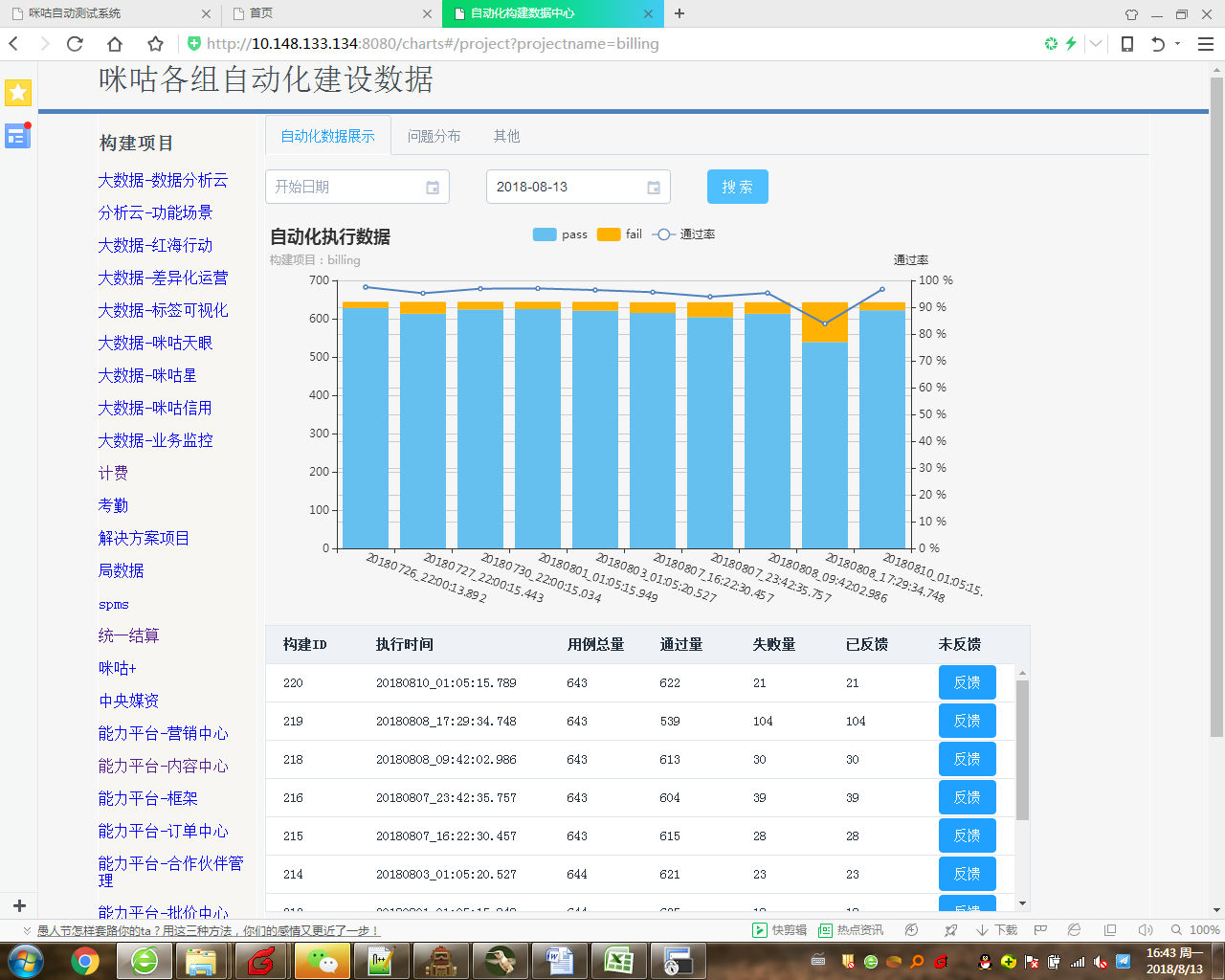


1. 编写支持markdown语法：可嵌入代码，表格，列表，图片与链接，引用等，点击此或者直接搜索markdown获取说明：



### 2.2自动化建设数据

该页面可以查看各组自动化脚本通过率情况，进行失败脚本分析反馈等。



Jenkins上每次自动化测试的构建结果都会推送到ATP，平台会将本次构建的通过率情况进行展示，各组要针对失败脚本进行分析反馈。单击上图某次构建的反馈按钮，在以下页面选择失败原因分类、模块、失败数量、描述等，点击提交，即可生成分析数据。



失败脚本分析反馈率要求达到100%，问题分布里，可以下载查看反馈结果。



注：此处从jenkins执行机获取的结果数据，初次打通需要添加脚本命令，详细联系：补冲

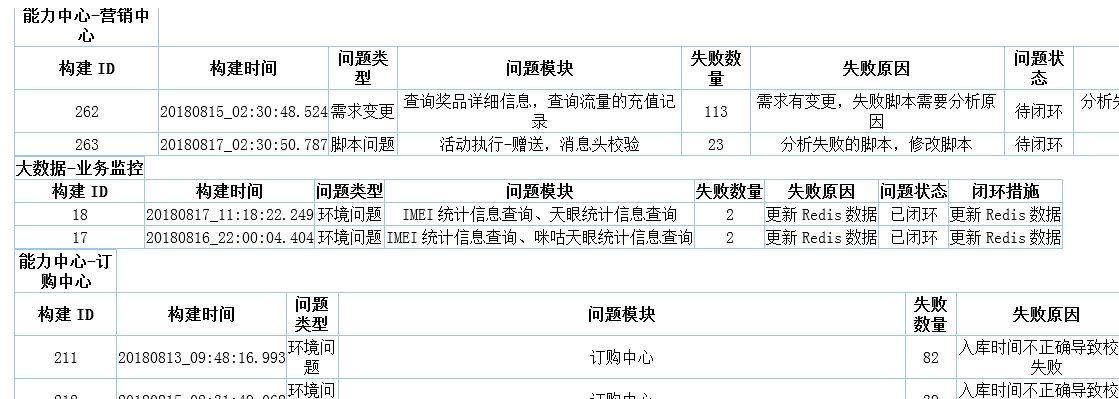
每日和每周数据会自动发送发送统计通知邮件，如下图：

日报数据：



周报数据：





注：此邮件自动发送，若不在此邮件收件人，请联系：补冲

### 2.3 Migu library

该页面可以查看目前migu\_library所有关键字的帮助信息，点击某个关键字即可查看详情，支持搜索。

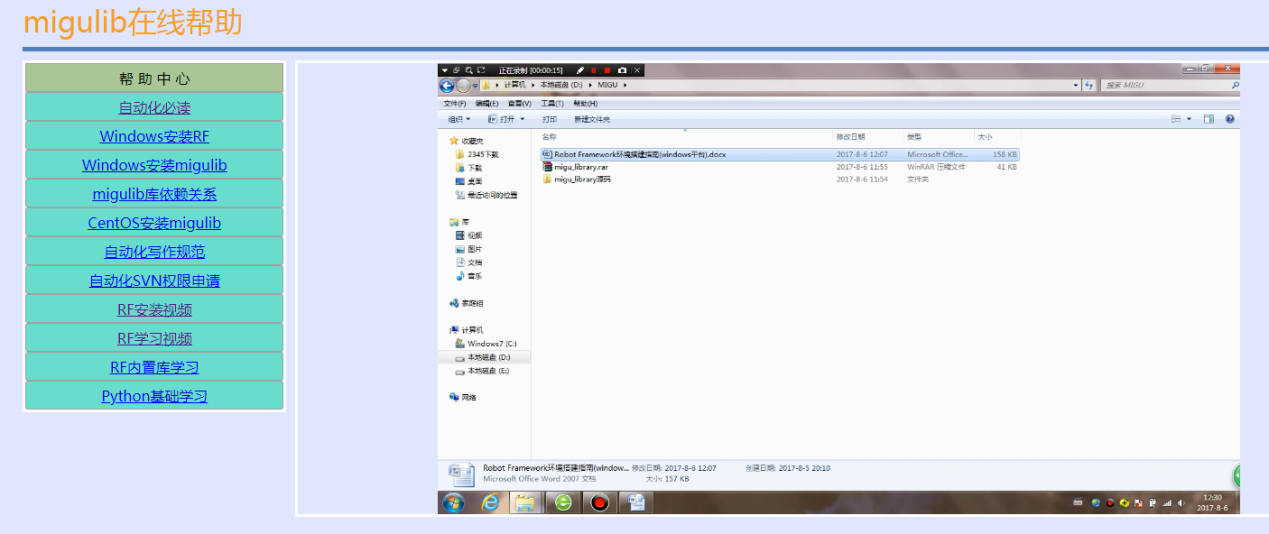


### 2.4 Robot Framework

该部分包含很多在线帮助和学习文档，会定期更新，方便新人学习自动化相关内容。



可在线播放学习和安装视频：



## 3 CTMP自动化测试平台使用指导

全部详情指导见此文件：

### 3.1 权限申请

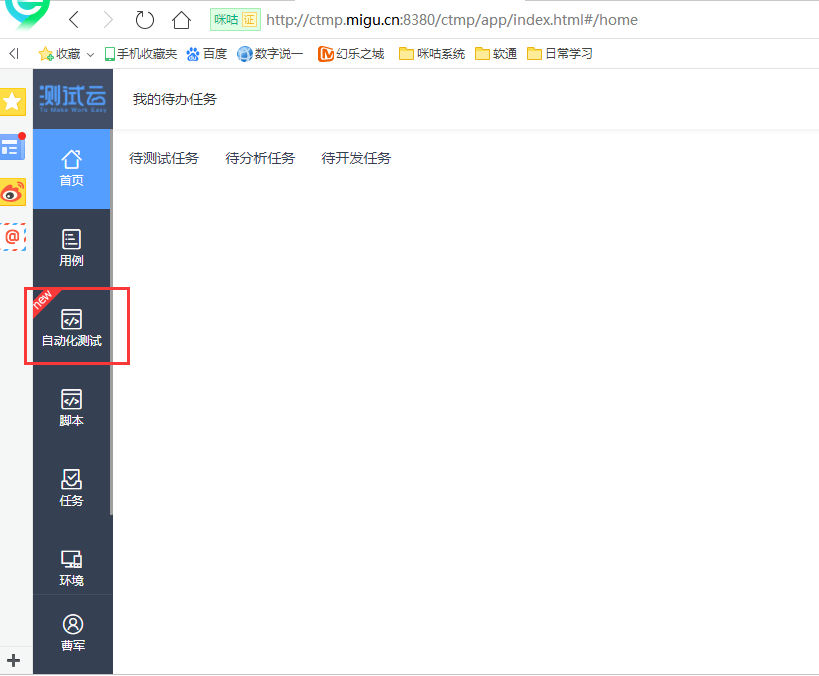
登陆SPMS，在研发工单Plus下申请10.148.64.142, 10.148.64.145的VPN权限，审批通过后，使用手机号+yw账号登陆VPN。



注：此系统项目，版本和人员信息从spms同步，若见不到项目，请联系项目经理或者测试经理添加权限

### 3.2 脚本调试

1、登陆CTMP，打开“自动化测试”子系统；



1. 打开 “脚本管理” 页面，点击“设置”



1. 输入设置名称，脚本管理路径根据项目情况选择svn或git，输入自动化脚本所在的svn或git路径，例如svn://10.148.68.6/南京测试团队项目/02-项目管理/08-自动化/02-一级话费计费系统/局数据；



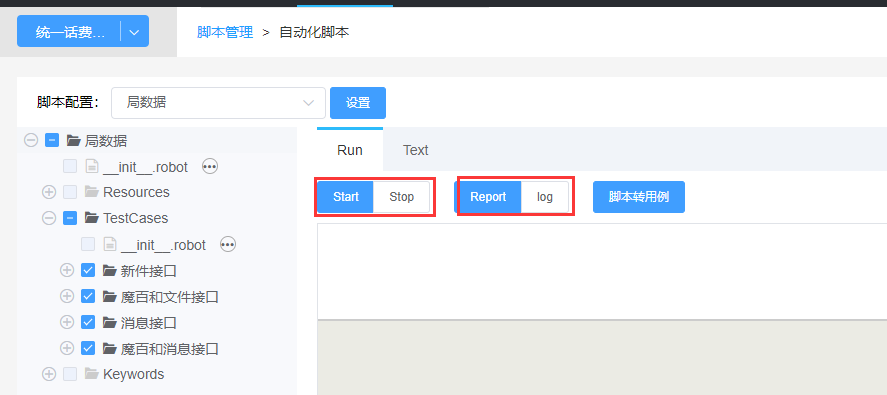
1. 点击确认，稍等一段时间，左侧会树形结构展示该项目的所有自动化脚本。



1. 脚本的执行：选择需要执行的脚本，以及执行机，

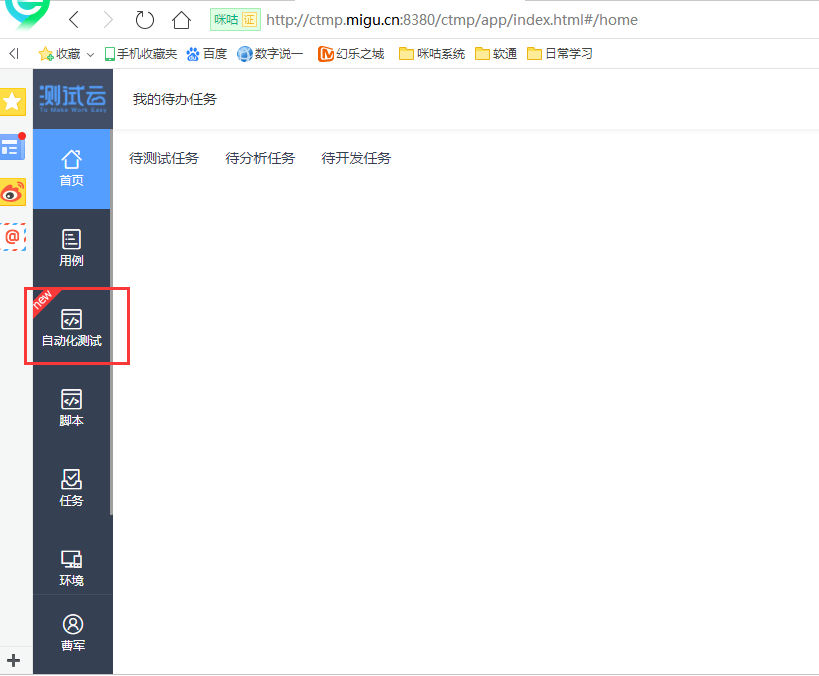


1. 点击“Start”执行脚本，“stop”停止执行，以及查看执行结果“log”。



### 3.3 项目运行任务

1、登陆ctmp系统，打开“自动化测试”子系统。



2、弹出页面选择“任务管理”--点击“新建任务”，选择“项目运行任务”如下图：



3、在项目运行任务详情页面，填写信息，点击提交，如下图：

大家可以根据自己的需要来选择是一次性任务还是定时任务。定时任务时间设置方式可以查看说明。（例如：0 16 ? \* 1-5，周一到周五每天下午的16点执行定时任务）

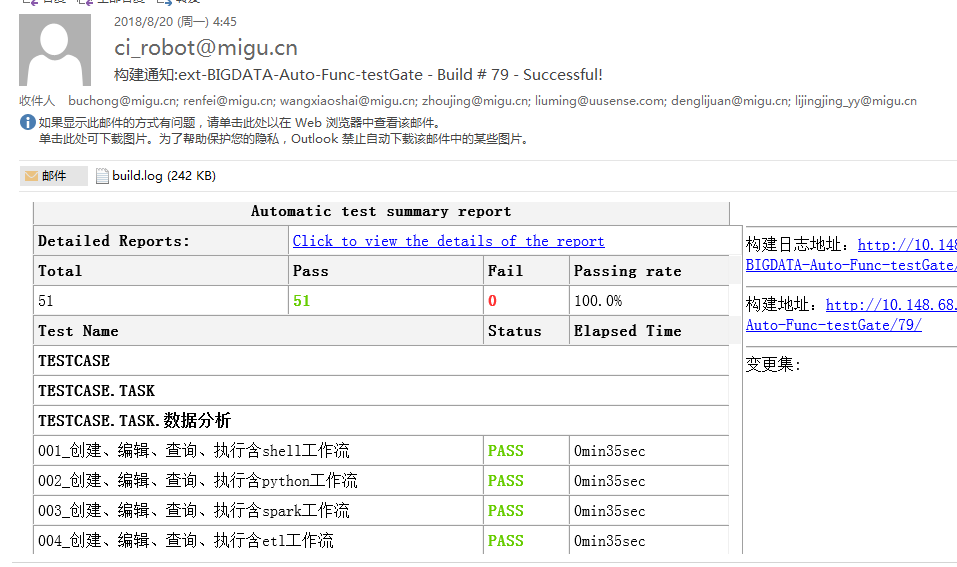


# IV 自动化测试相关考核说明

## 1 自动化测试应用于版本冒烟及回归测试

按照考核要求，自动化测试要应用于项目提测版本的冒烟测试及回归测试，由Jenkins执行结果及相关反馈邮件证明。类似如下，在Jenkins执行结果基础上回复。





## 2 提测版本自动化脚本通过率考核

按照考核要求，18年Q3季度末，各项目提测版本最后一次回归测试自动化脚本**Jenkins通过率要达到95%以上**，需要提供当天或接近上线时间点的执行结果邮件截图和项目版本spms上线申请单截图。

## 3 接口自动化覆盖率考核

按照考核要求，**各项目接口自动化覆盖率要达到100%**，确实不能自动化或不需要自动化的接口，要给出合理的解释。关于接口覆盖率，各项目需要提供全量的接口文档给自动化技术支撑人员，针对当前的自动化脚本进行评估，是否与反馈的覆盖率一致。

**覆盖率计算方式**：全量接口自动化率 = 自动化脚本总数(TestCases) / 总的接口用例数\*100%

1.       总的接口数包含所有版本接口，并不是指某一版本；我们统计的是总覆盖率（即分母是所有版本接口总和（除去废弃接口））

2.       总的接口数不仅仅是项目对外接口，模块对内接口和管理接口等也要算测试点，纳入分母统计；

3. 一个接口正常下至少有10个左右用例，包括正常和异常用例。一个接口若只有1~2个用例需要提前评估是否覆盖全面；

4. 正常情况下一个接口写一个测试套(TestSuit)，一个测试套包含若干测试脚本(TestCases)

注：考核需要提交接口文档作为参考

## 4 失败脚本分析反馈

各项目Jenkins自动化脚本执行结果会上传到ATP平台【初次需要手工添加脚本接入，请联系：补冲】，所有失败的脚本都要在ATP上反馈失败原因，每月会统计失败反馈率，每天的构建及分析情况，ATP都会邮件知会各组测试人员，反馈操作参考ATP平台使用指导，每月月报会通报各组反馈率。

## 5 自动化月报数据反馈

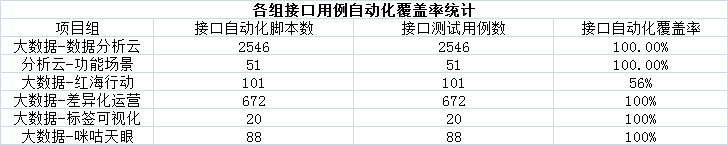
每月月末会收集各组自动化相关数据，由自动化接口人填写，各组组长进行核对，确保数据的准确，真实。以8月份为例，月报数据会在8月30号或31号开始收集，需要反馈以下数据。

1. 各组接口用例自动化覆盖率统计

接口自动化脚本数：填写截止到8月底本组的接口自动化脚本总数，注意是总数，不是某个版本的脚本数量。

接口测试用例数：填写截止到8月底本组接口测试用例总数，针对该项目的全量的接口测试用例数。

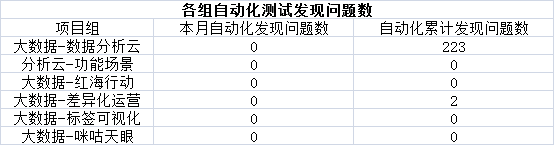
接口自动化覆盖率=接口自动化脚本数/接口测试用例数。



1. 各组自动化测试发现问题数

本月自动化发现问题数：填写8月份，本组自动化测试发现的问题数量。

自动化累计发现问题数：将本月自动化发现的问题数进行累加，例如7月月报里自动化累计发现问题数是100，8月自动化发现10个问题，那么，自动化累计发现问题数为110.



1. 自动化验收提测版本次数

本月自动化验收提测版本次数：填写8月份，本组总共验收的提测版本数（按照要求所有的提测版本都要进行自动化冒烟和回归测试）。

自动化验收提测版本累计次数：将本月自动化验收提测版本次数进行累加，例如7月月报里自动化验收提测版本累积次数是50，8月自动化验收提测数是10，那么，自动化验收提测版本累积次数是60次



1. 各组自动化脚本本月增长幅度统计

本月新增脚本数：填写8月新增的自动化脚本数，直接使用8月底的自动化脚本数减去7月底的自动化脚本数即可。

上月累积脚本数：截止到7月底的自动化脚本数。

增长幅度=本月新增脚本数/上月累积脚本数



1. 各组自动化脚本投入产出情况分析

本月新增脚本数：填写8月新增的自动化脚本数

本月自动化投入人数：本月投入自动化相关工作的人数



# V 自动化需求及问题接口人

## 1 成都

补冲

## 2 南京

赵明明

## 3 Jenkins相关问题

毛润生、陈金瑞