目录

[脚本安装卸载步骤 2](#_Toc21644)

[Demo地址 2](#_Toc1985)

[v3.6.6新增私有化环境登录支持配置 2](#_Toc11881)

[v3.3.97各种鉴权方式统一 2](#_Toc11881)

[v3.3.0 线上监控的测试集 6](#_Toc18572)

[v3.0.10 报告推送到群 6](#_Toc8960)

[v3.0.9 脚本分层 7](#_Toc24439)

[v3.0.3~3.0.8 卸载nose框架 10](#_Toc13128)

[v3.0~3.0.2 监控报错推送配置 10](#_Toc5669)

[附 12](#_Toc9427)

[租户改造方法 12](#_Toc7277)

[v3.0.11登录优化方法使用 14](#_Toc11872)

[变更记录 16](#_Toc9014)

**脚本安装卸载步骤**

**无论需要使用哪个版本，请先卸载旧版的ndlib（如果有装的话），再安装最新版。操作如下：**

pip uninstall ndlib // 先卸载原先的包

pip install ndlib -i http://fs.qa.huayu.nd:8202/simple --trusted-host fs.qa.huayu.nd // 安装最新的包（ndlib前有空格，不可忽略）

**Demo地址**

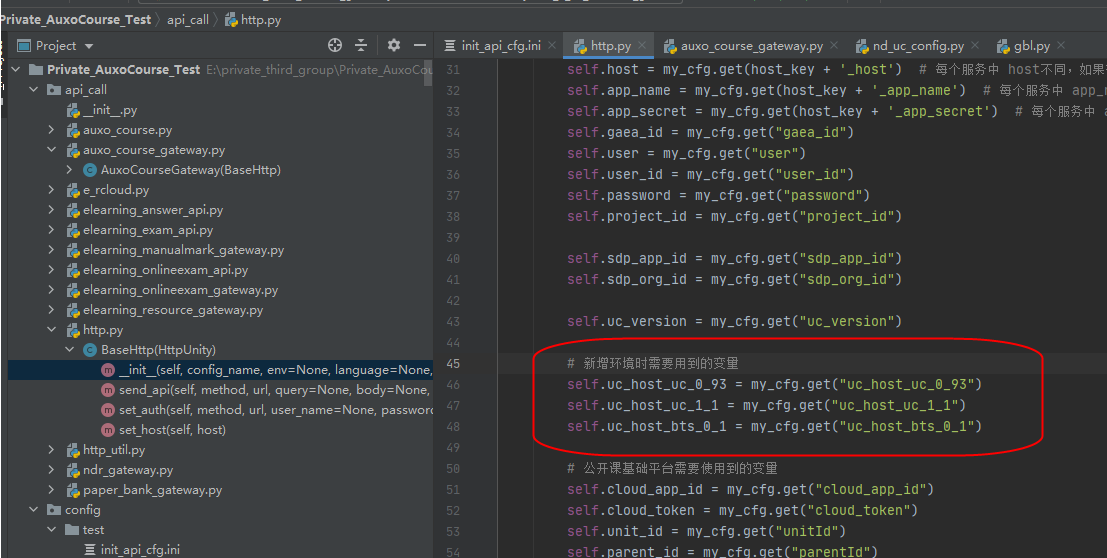
**https://192.168.160.4:8443/svn/QA/工作文档/X 项目文档/华渔项目/C 测试开发/Api测试管理/Python\_Api\_Demo/ApiTest\_Demo**

**账号：工号/123**

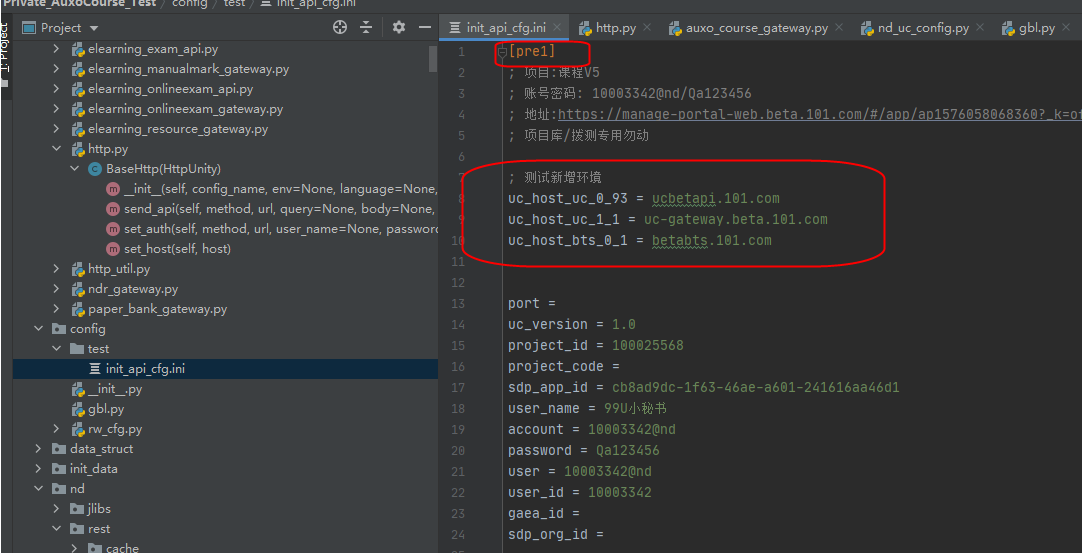
**V3.6.6新增私有化环境登录支持配置**

通过本地配置文件增加uc\_host配置项，方便新增私有化环境登录；兼容已有接口脚本工程，非新增环境无需增加此项配置

1. http.py里添加如下变量



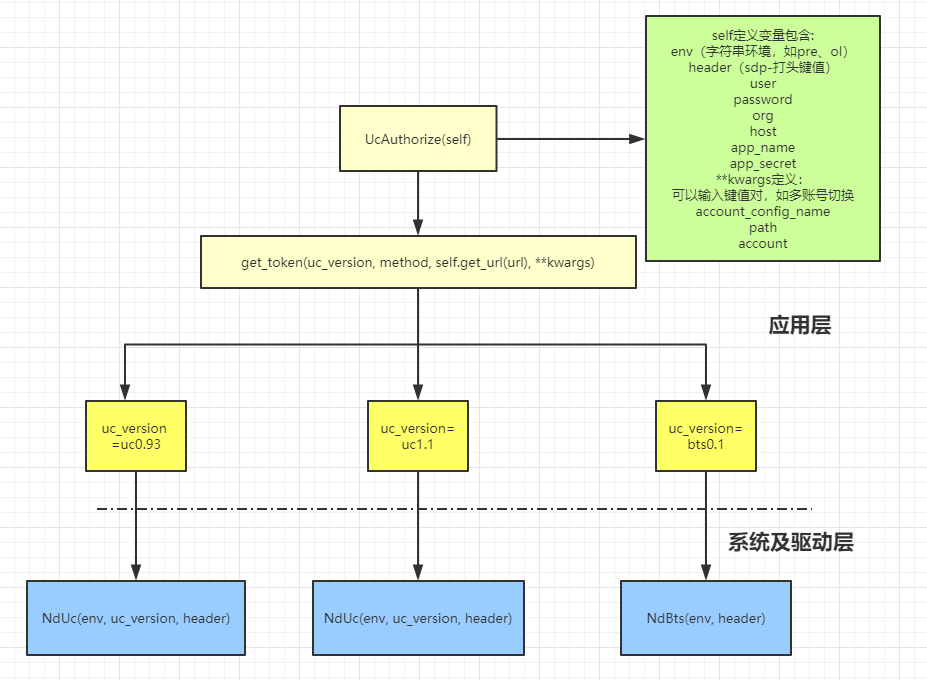
1. 本地配置文件增加配置项：新增私有化环境对应的UC服务host

****

1. 通过以上1、2配置，当nd\_uc\_new.py 没有定义该环境，则默认取上面的配置，已经定义，则默认取定义的配置

**v3.3.97各种鉴权方式统一**

1. **设计图**

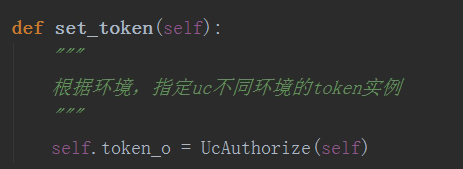
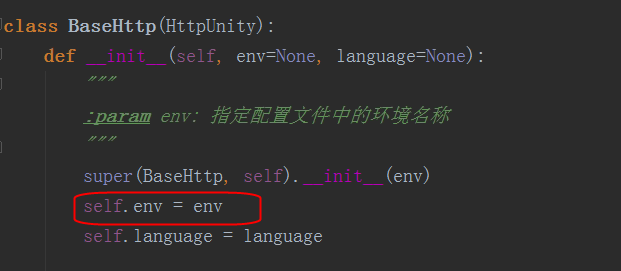


**注：原有旧uctoken类仍然保留，可以继续使用**

1. **http.py中引入库如下**

**from** nd.rest.uc.token **import** \*

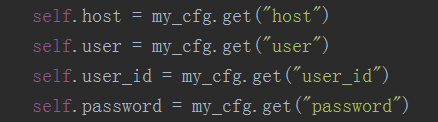
1. **set\_token方法里Uctoken初始化修改如下，同时确保self里存在env变量，且是字符串，如pre、ol等**

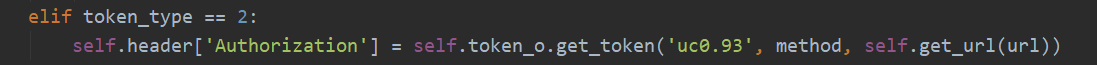


self.token\_o = UcAuthorize(self)

1. **set\_auth方法里 get\_token 使用修改**
2. **uc0.93**

**self中确保存在host、user、password变量，如果配置文件中有org，也可以传入org，如self.org**

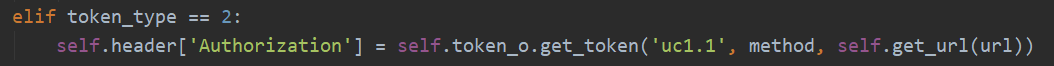
****



self.header['Authorization'] = self.token\_o.get\_token('uc0.93', method, self.get\_url(url))

1. **uc1.1**

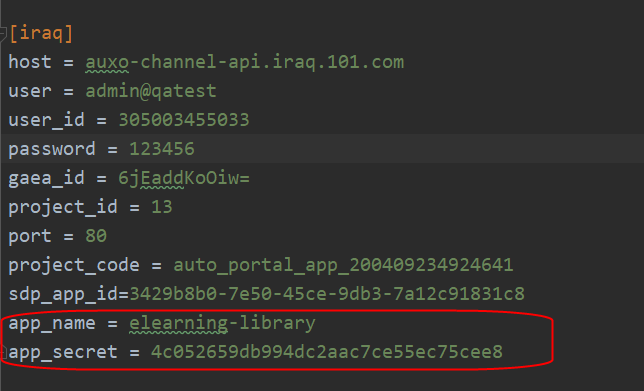
**self变量同上uc0.93**



self.header['Authorization'] = self.token\_o.get\_token('uc1.1', method, self.get\_url(url))

1. **bts**

**a、.ini配置项增加字段:**

****

app\_name = elearning-library  
app\_secret = 4c052659db994dc2aac7ce55ec75cee8

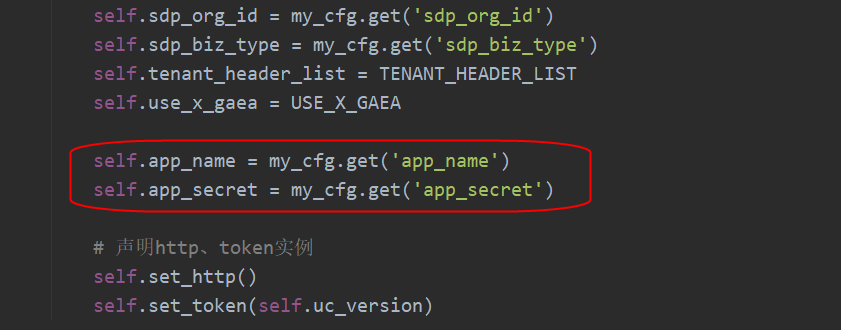
**注：举例获取密钥地方**



****



**b、初始化时，增加app\_name和app\_secret配置获取**

****

self.app\_name = my\_cfg.get('app\_name')  
self.app\_secret = my\_cfg.get('app\_secret')

**c、uc\_version设置为bts0.1**



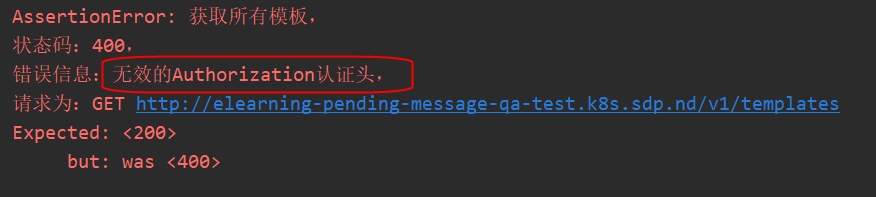
self.header['Authorization'] = self.token\_o.get\_token('bts0.1', method, self.get\_url(url))

**注：**

**1、当前项目如果是app方式开通，self.header必须带有sdp-app-id，如果是app+biz-type方式开通，则self.header必须带有sdp-app-id、sdp-biz-type，如果是app+org方式开通，则self.header必须带有sdp-app-id、sdp-org-id； ------可以参考附录的租户改造方法自动加入租户信息**

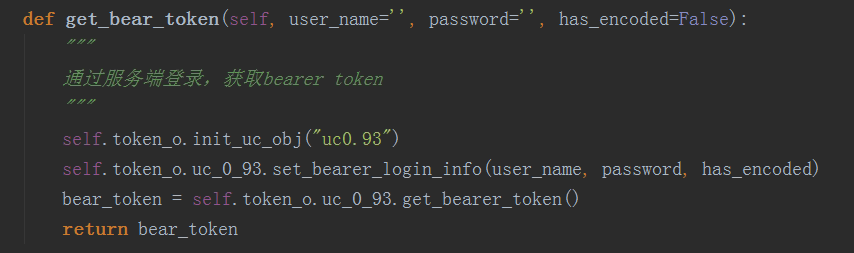
**2、UserId非必传，如果接口提示需要传UserId，才必须传入，如果self中有user\_id参数，则会在header自动带上UserId；**

**3、UserId必须为appid下的组织用户id，不然会提示“无效的Authorization认证头”。**

****

1. **bearer**

由于bearer要废弃了，没有专门写判断分支，需要使用bearer可以参考下面用法



**def get\_bear\_token**(self, user\_name='', password='', has\_encoded=False):  
 *"""  
 通过服务端登录，获取bearer token  
 """* self.token\_o.init\_uc\_obj("uc0.93")  
 self.token\_o.uc\_0\_93.set\_bearer\_login\_info(user\_name, password, has\_encoded)  
 bear\_token = self.token\_o.uc\_0\_93.get\_bearer\_token()  
 **return** bear\_token

1. **多账号切换用法（可选）**

参考附录---v3.0.11 登录优化方法使用

**v3.3.0** 线上监控的测试集

1. 用于线上监控的测试集，可以不用配置proid以及推送对象（置空列表即可）；相关配置请在监控平台上设置。

****

**v3.0.10** 报告推送到群

1. 支持报告推送到群时，@对应联系人。若使用该功能，需要在测试集中增加contanct字段内容，如：

"contact": {  
 "users": ["林斯潇", "QA测试22"]  
}

“users”中请根据实际需要填写人员姓名。

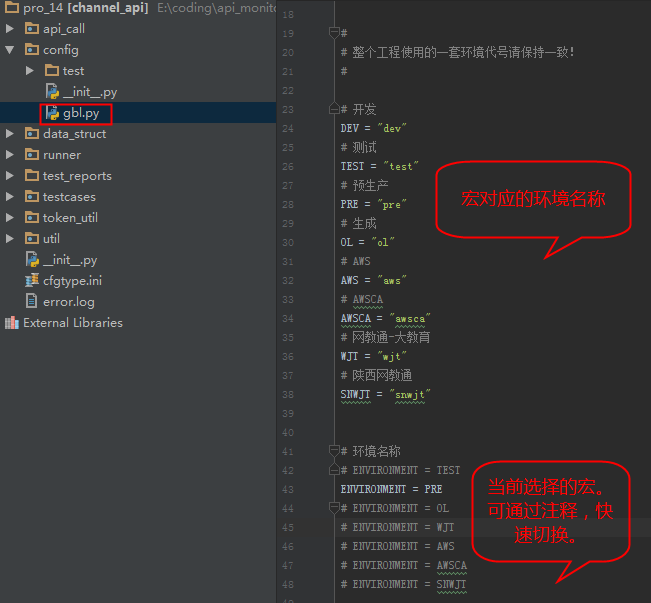
**v3.0.9** 脚本分层

**为实现脚本分层（可选，根据项目需要选择），需要做以下修改：**

1. 确保config目录为一个模块（目录下有\_\_init\_\_.py文件），在config目录下建一个gbl.py文件。

在该文件里声明环境**ENVIRONMENT**，用于本地右键运行、调试单独的脚本文件。

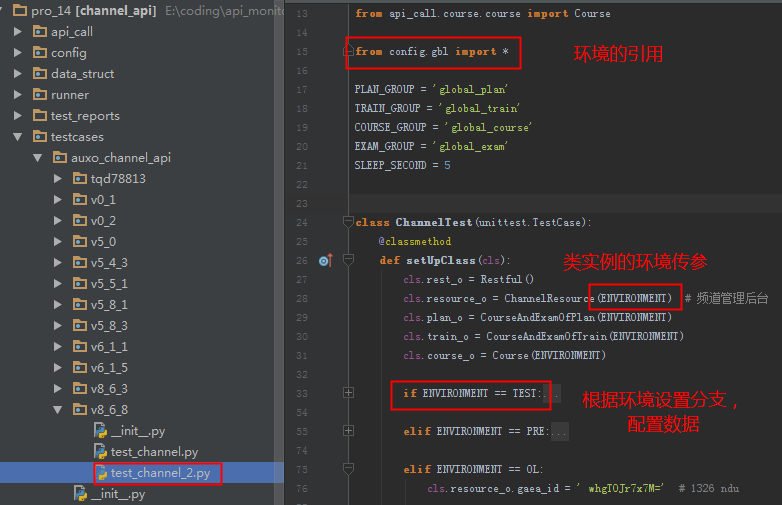
也可以在里面引用一些公共的库，这样每个case所在的脚本就不用再引用了。



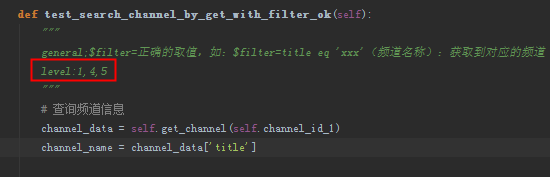
1. 在case中将取环境配置的引用，改为引用gbl配置，：from config.gbl import \*

使用gbl中的ENVIRONMENT来表示环境

脚本初始化方法中，根据环境，配置变量值（具体参数、数值根据case需要配置）



1. 在case的注释中指定当前case**适用层级**，可以有多层，如："level:1,4,5"

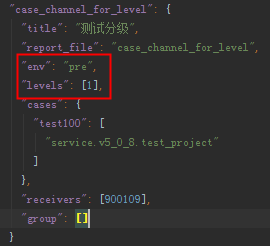


1. 在测试集**配置**中增加两个字段：
2. 要运行的**层级**列表，用于命令行运行、jenkins构建，如："levels":[1,2]

适用层级 与 配置层级 有交集的case都会运行。

1. 要指定的**环境的名称（有对应的宏）**，用于改变运行时case环境，如"env": "pre"（对应的宏是PRE）

但由于不同层级适用环境可能不同，所以**配置指定的层级尽量只有一个**。



7、 规则说明

**分层规则：**

1、在case的注释中，单独一行指定适用层级（全英文字符），如："level:2, 3"

在测试集配置中指定要运行的层级列表，如："levels":[1, 2]

适用层级 与 配置层级 有交集的case都会运行

2、未在注释中指定层级的case，不受影响

3、在测试集配置中指定空层级列表，或未配置层级字段，则运行所有脚本

**环境规则：**

1. 在测试集中配置环境，如"env": "pre"
2. 运行测试集时：
3. **脚本case要使用config->gbl.py中的ENVIRONMENT来表示环境值，才能达到在运行时临时切换为配置环境的效果**
4. 在测试集中配置环境值为空，或未配置环境值字段，则在运行时不变更脚本环境
5. 右键运行文件时：

**脚本case环境以config->gbl.py文件中指定的ENVIRONMENT值为准**

**适用场景：**

1、指定到多个文件

2、指定到多个目录

3、指定到目录时，同目录下有同名不同level的case

4、指定到文件&目录混合

**层级规范：**

版本测试-测试环境 1

版本测试-预生产环境 2

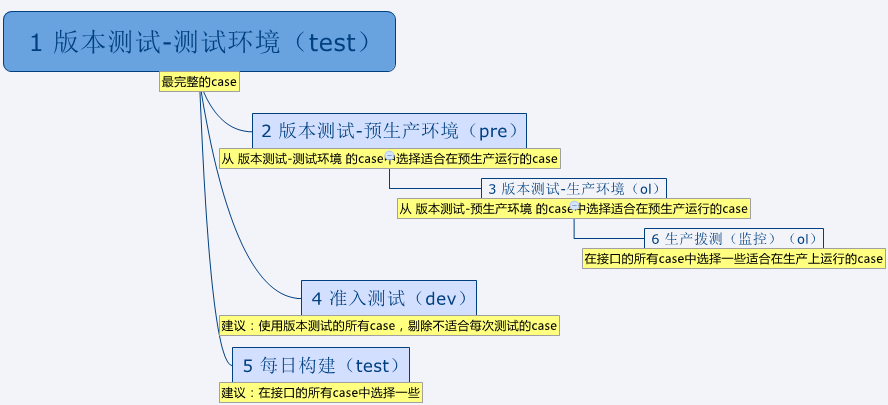
版本测试-生产环境 3

准入测试（开发/测试环境）4

每日构建（测试环境） 5

生产拨测（监控） 6

建议的层级筛选方式如下



**环境规范（测试集配置、case统一使用）：**

# 开发

DEV = "dev"

# 测试

TEST = "test"

# 预生产

PRE = "pre"

# 生成

OL = "ol"

# AWS

AWS = "aws"

# AWSCA

AWSCA = "awsca"

**v3.0.3~3.0.8** 卸载nose框架

1. 卸载nose框架（可选）

pip uninstall nose

**v3.0~3.0.2** 监控报错推送配置

1. 删除cof目录
2. runner目录下，runner.py文件使用（请删除所有其他旧的runner文件）



1. 所有使用 cof.xxx 方式的引用，都改成 nd.rest.xxx

其中：

from cof.file import \* 改成 from nd.rest.co\_file.file import \*

from cof.conf import MyCfg 改成 from nd.rest.conf.conf import MyCfg

from cof.co\_time import\* 改成 from nd.rest.co\_time.co\_time import \*

建议：

先在工程里搜索以上三个特例，实现替换；

然后再搜索 cof. ，替换为 nd.rest. （双击选中cof，粘贴 nd.rest ，操作更快）

1. apicall目录下，必须按服务类型管理，且仅有一层结构，若某个服务有多个版本，请不要再分下一层。

每一个服务目录下都有一个http.py，并在my\_cfg = MyCfg('cfg.ini')实例化语句后，增加路径设置 my\_cfg.set\_path(\_\_file\_\_)

类似的，所有引用get\_cfg\_type的地方都要传入当前路径 \_\_file\_\_

from nd.rest.conf.conf import get\_cfg\_type, MyCfg

my\_cfg = MyCfg("")

my\_cfg.set\_path(\_\_file\_\_)

cfg\_type = get\_cfg\_type(\_\_file\_\_)

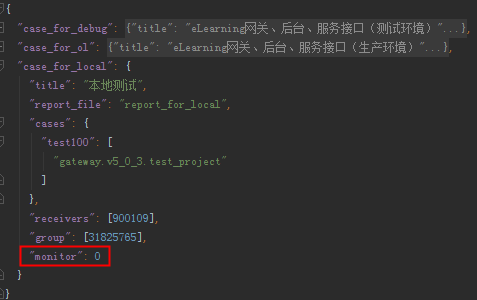
1. 关于推送

每日构建（测试环境）每次必推。

线上监控（生产环境）仅有失败时才推送，那么可以在对应的配置中设置字段“monitor”：

0代表不是处于监控，每次必推；

1代表处于监控，仅失败才推送。



1. 提交代码到svn
2. 查看jenkins配置，保证批处理命令中使用的是 runner.py文件，其他不变

D:\imData\im\900109@nd\Image\82b227d565570a5727f830094a2bfc72.png

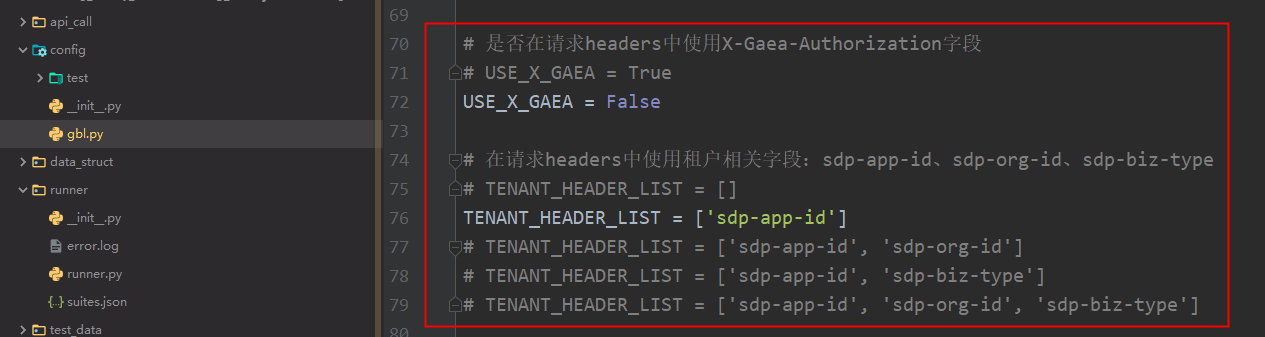
1. 重新执行构建，检查是否能正常执行

**附**

**租户改造方法**

1. 在gbl.py文件中增加宏定义

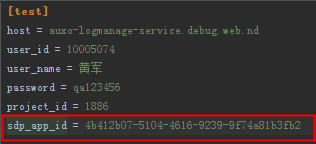
# 是否在请求headers中使用X-Gaea-Authorization字段  
USE\_X\_GAEA = True  
  
# 在请求headers中使用租户相关字段：sdp-app-id、sdp-org-id、sdp-biz-type  
TENANT\_HEADER\_LIST = []  
# TENANT\_HEADER\_LIST = ['sdp-app-id']  
# TENANT\_HEADER\_LIST = ['sdp-app-id', 'sdp-org-id']  
# TENANT\_HEADER\_LIST = ['sdp-app-id', 'sdp-biz-type']  
# TENANT\_HEADER\_LIST = ['sdp-app-id', 'sdp-org-id', 'sdp-biz-type']



通过改变两个宏的值，可以切换使用X-Gaea-Authorization方式、租户化方式请求数据（也可以两种同时使用）。其中，根据门户开通租户的方式，租户化headers传参有4种情况，TENANT\_HEADER\_LIST的内部表示将传哪几个字段，另外，列表为空表示不传任何参数。

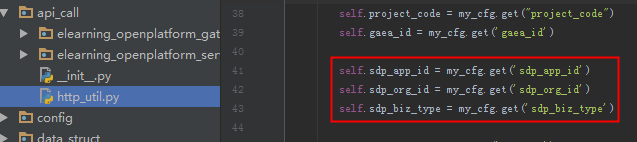
1. 在配置文件中增加工程全局的默认租户数据（sdp-app-id、sdp-org-id、sdp-biz-type），租户数据与原项目id对应。

如下图示例中project\_id = 1886 是由app方式开通的，其对应的sdp-app-id = 4b412b07-5104-4616-9239-9f74a81b3fb2



如果项目是由其他方式开通，请配上其他相关数据。

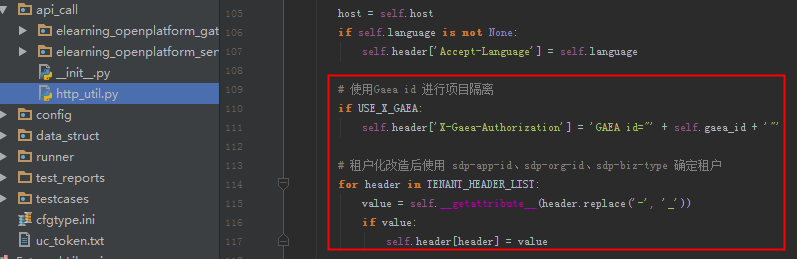
1. 在http.py文件中读取配置处，增加租户数据的读取（配置文件中没配的字段也可以读取，程序不会中断执行，取值为None）



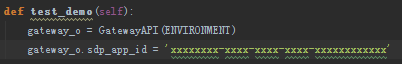
1. 在http.py文件原先设置X-Gaea-Authorization的地方，增加headers字段sdp-app-id、sdp-org-id、sdp-biz-type，并且可以根据1中的两个宏的值启用、禁用X-Gaea-Authorization和租户相关字段。

**from** config.gbl **import** USE\_X\_GAEA, TENANT\_HEADER\_LIST

# 使用Gaea id 进行项目隔离  
**if** USE\_X\_GAEA:  
 self.header['X-Gaea-Authorization'] = 'GAEA id="' + self.gaea\_id + '"'  
  
# 租户化改造后使用 sdp-app-id、sdp-org-id、sdp-biz-type 确定租户  
**for** header **in** TENANT\_HEADER\_LIST:  
 value = self.\_\_getattribute\_\_(header.replace('-', '\_'))  
 **if** value:  
 self.header[header] = value



1. 2是全局租户数据的配置方式，若某个脚本中想要使用另外的租户数据进行测试，可以新建一个测试类对象，在case中通过给测试对象的成员变量重新赋值的方式，使这个测试对象使用指定的租户数据。



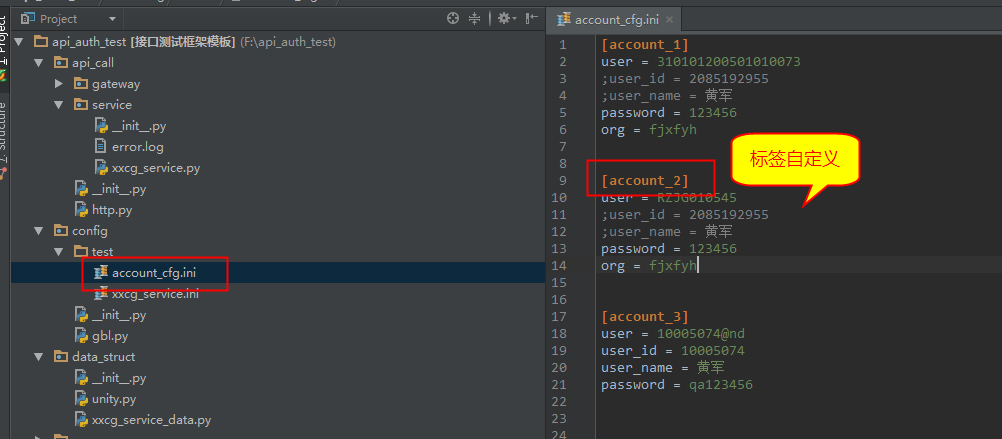
**v3.0.11登录优化方法使用**

更新内容：

1. 增加对uc登录的性能优化
2. 提供不同账户切换和交替操作的简便方式

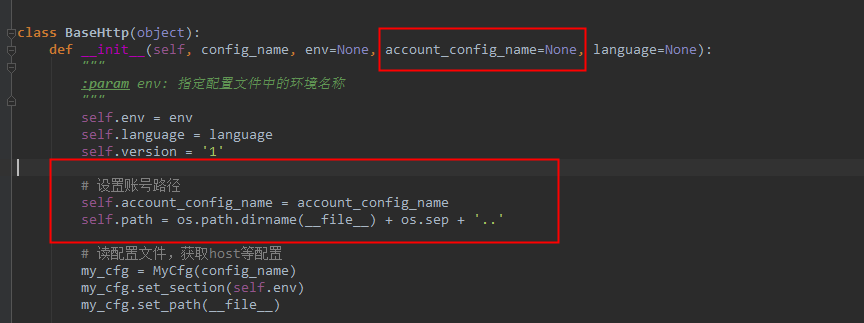
本次ndlib更新，提供接口请求切换账号的便捷操作，用于多角色用户的业务。若使用该功能，需要按照以下方式编写账号相关脚本：

1. 首先把测试脚本需要用到的用户信息放在固定的文件内，可在配置文件夹下 (config-->test)新建账户配置文件，如account\_cfg.ini；内容为具体的账号信息，可参考服务配置里面的环境信息。



1. 接口调用（api\_call）所引用的http.py中，BaseHttp类需要读取第1步添加的账户配置信息

2.1 BaseHttp类\_\_init\_\_初始化方法，增加如图字段，其中account\_config\_name为配置文件名，path为配置文件路径。



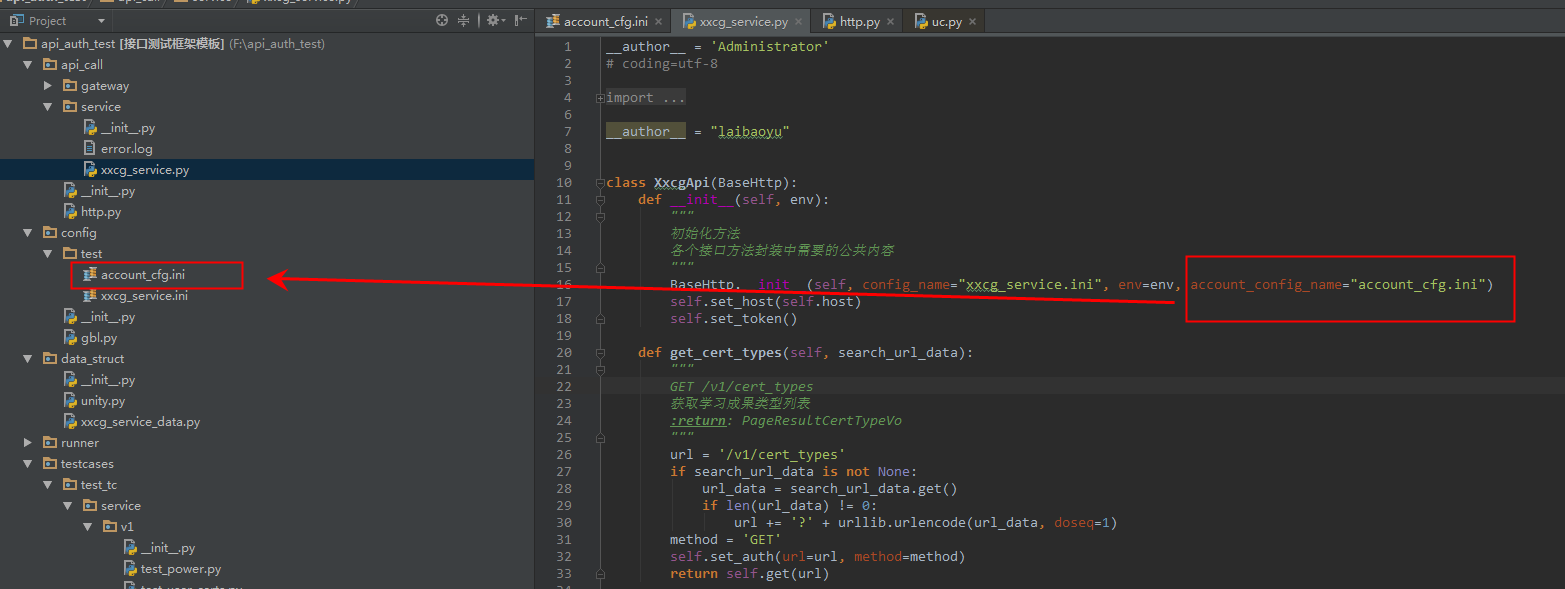
self.account\_config\_name = account\_config\_name  
self.path = os.path.dirname(\_\_file\_\_) + os.sep + '..'

2.2 BaseHttp类set\_auth()设置授权信息方法也需要增加以下字段，account为指定的账号标签。get\_token()方法需要增加1个字段account（账号名）传入，如图所示：

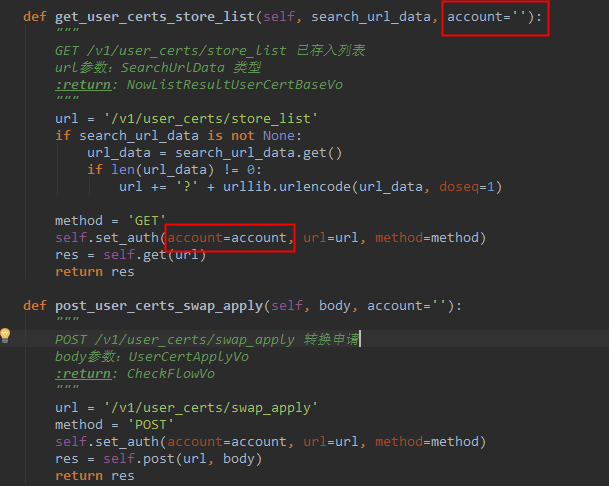


1. 接口调用（api\_call）

3.1 api\_call->服务名->业务接口调用封装代码中类的初始化需要加入 account字段

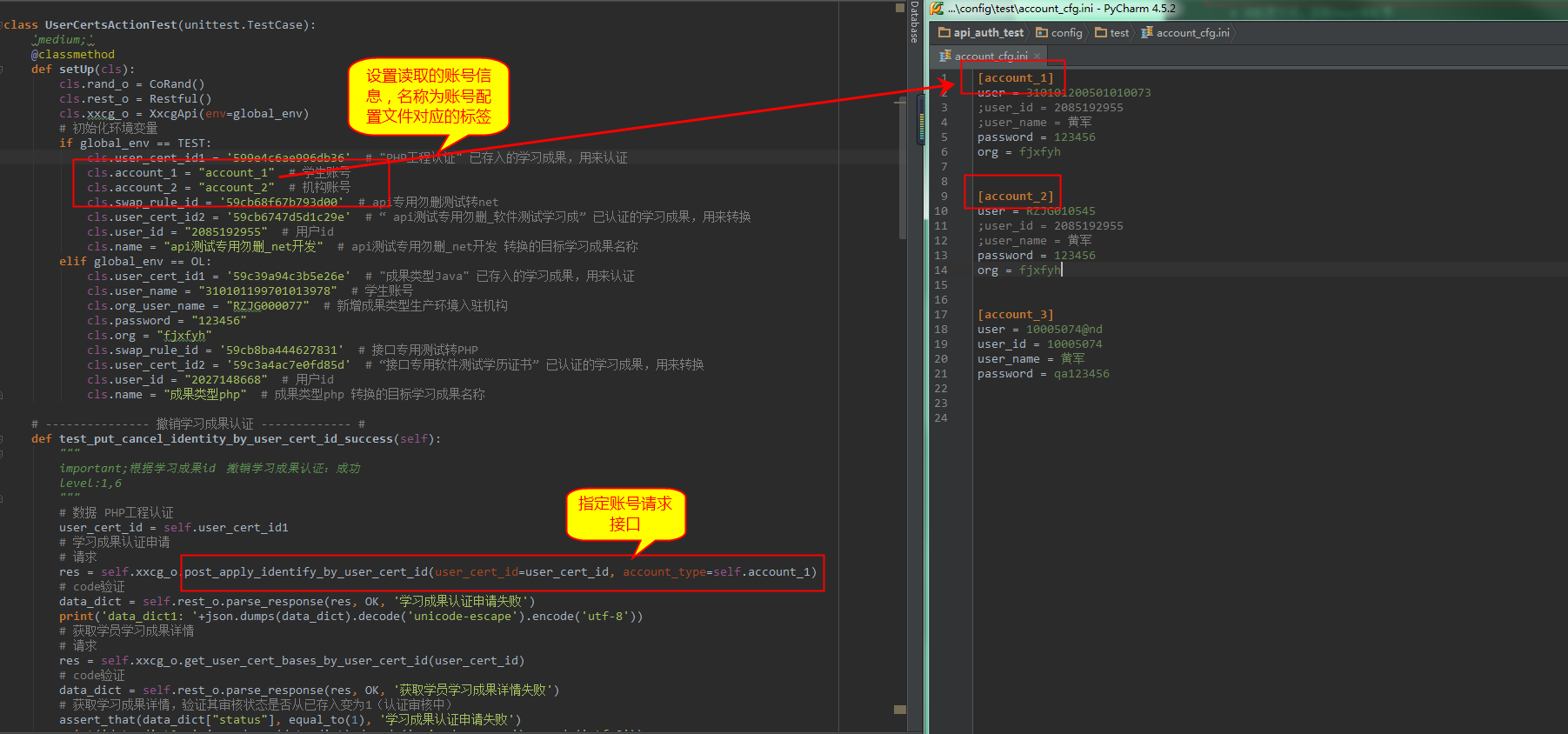


3.2 具体的接口封装，加入账号信息：



1. testcases

4.1 在每个测试测试集的准备数据中设置账号，供测试用例调用。如图



**变更记录**

|  |  |  |
| --- | --- | --- |
| **ndlib版本** | **内容** | **时间** |
| V10.0 | 新增私有化环境登录支持配置 | 2021-02-04 |
| V3.3.97 | 增加多种鉴权方式统一调用类 | 2020-05-06 |
| V3.3.95 | 修复bts鉴权不支持biz-type和org开通方式的问题 | 2020-04-23 |
| V3.3.93 | bts鉴权增加判断是否已经获取了token，不用每次接口请求都去申请bts token | 2020-04-17 |
| V3.3.92 | bts增加判断是不是https协议 | 2020-04-10 |
| V3.3.91 | 优化bts登陆 | 2020-04-10 |
| V3.3.90 | 优化uc1.1登陆，新增伊拉克uc地址，以及增加可以自定义输入uc地址功能 | 2020-04-09 |
| V3.3.10 | 1、取消发送日志到日志服务器，避免日志服务器服务挂掉后，工程执行中断 | 2018-10-15 |
| V3.3.9 | 1、优化mongo库的数据存储，不再存储测试集的数据到mongo库的suites表 | 2018-10-09 |
| V3.3.8 | 1、优化restful断言文件，与线上监控平台的restful文件兼容  2、新增生成指定长度的随机大写字母字符串和所有字母字符串的方法  3、请求头增加Qa-Tag标识，辅助开发过滤掉框架发起的测试请求  4、新增香港的uc请求地址 | 2018-09-28 |
| V3.3.7 | 1、新增接口响应时长超过60s后自动断开功能 | 2018-09-12 |
| V3.3.6 | 1、修复使用uc1.0登陆提示“Session未创建或已过期”问题 | 2018-08-09 |
| V3.3.5 | 1、优化对uc1.0的支持 | 2018-07-17 |
| V3.3.4 | 1、增加uc对陕西网教通（snwjt）支持 | 2018-07-12 |
| V3.3.3 | 1、增加对uc1.0鉴权支持 | 2018-07-10 |
| V3.3.2 | 1、restful.py的parse\_response函数增加兼容，有默认message信息 | 2018-06-15 |
| V3.3.1 | 1、使用cs新上传域名"uploadcs.101.com"有高概率故障，先改回"cs.101.com" | 2018-05-30 |
| V3.3.0 | 1、去除将接口调用信息发送给plot.qa.sdp.nd服务的请求。 2、增加对脚本第三层的路径支持。 | 2018-05-29 |
| V3.2.0 | 1、兼容监控平台运行、报告推送需求，增加对-p命令行参数的支持。 | 2018-05-09 |
| V3.1.7 | 1、去除了监控平台未统一触发拨测时，将报告推送至监控平台的操作。 | 2018-04-23 |
| V3.1.6 | 1、测试报告中增加了测试集名称的显示。 | 2018-04-19 |
| V3.1.5 | 1、修复了setup或teardown中有断言失败时无法生成报告的bug。 | 2018-04-13 |
| V3.1.4 | 1、在rest下增加了两个公共文件：http\_mot.py、mactokenutil.py，用于支持监控平台功能。 | 2018-03-21 |
| V3.1.3 | 1、修复返回值含有特殊编码数据时，无法生成报告的bug。  2、加长了报告上传cs的session有效时间，为50min。 | 2018-02-09 |
| V3.1.2 | 1、屏蔽向接口监控平台发报告的语句。 | 2018-02-07 |
| V3.1.1 | 1、修复了解析接口响应数据时，打印内容为空的bug。 | 2018-02-06 |
| V3.1.0 | 1、与接口监控平台特性做了合并。向下兼容。 | 2018-01-26 |
| V3.0.12 | 1、增加WJT环境的UC配置，支持大教育独立部署环境的测试 | 2018-01-17 |
| V3.0.11 | 1、在请求headers的User-Agent中，拼接“ApiAutotest”，以标志是接口测试发出的请求  2、优化restful解析，在状态码不符合期望时，打印所有请求、响应信息  3、优化token生成：  ① 避免同账号在token有效期内多次登录  ② 增加新的多账号切换机制 | 2017-12-28 |
| V3.0.10 | 1、支持在推送群时@contact内指定的人员  2、增加对uc token的性能优化（自测中） | 2017-11-29 |
| V3.0.9.1 | 1、支持通过配置指定运行测试用例的层级 | 2017-08-25 |
| V3.0.8 | 1、修复SLA服务器异常后，测试中断问题 | 2017-08-10 |
| V3.0.7 | 1、修复SLA获取不到接口用例执行时间问题  2、增加title字段内容的接口测试信息入库 | 2017-08-02 |
| V3.0.6 | 1、修复获取bearer token时，查找不到md5.jar加密包问题  2、修复返回的错误数据中存在"detail"键时，其值为None，导致解析返回数据失败的问题 | 2017-07-27 |
| V3.0.5 | 1. 修复用例没有描述，导致报告生成出错问题、用例描述获取新增异常抛出处理 2. 修复使用unittest框架后，只配置用例文件目录，不运行目录下test\*.py文件问题 3. 修复setUpClass初始化失败，中断工程运行问题，针对运行结果输出部分新增异常抛出处理 | 2017-07-07 |
| v3.0.4 | 1. 去除对nose框架的依赖，并更改用例运行时间计算实现，修复最后一个用例运行时间的错误。 2. 修复了测试结果中失败数为0时，未发送报告给接口监控平台的问题。 | 2017-07-06 |
| v3.0.3 | 1、接口测试集配置中新增proid参数，支持测试报告发送到SLA监控平台 | 2017-07-03 |
| v3.0.2 | 1、修复对NdApi类的支持问题 | 2017-06-28 |
| v3.0.1 | 1、报告标题时间格式显示，导致标题中途换行问题修复  2、java 加密token的地址加上双引号，解决有电脑出现没有双引号，执行出错问题  3、开发预测试新增用例数、成功数和报告地址数据存储到mongo中 | 2017-06-20 |
| v3.0 | 1、将测试报告改为HTML格式，修复chrome浏览器无法解析问题，修改文件template.py  2、测试用例中的特殊字符转义为html可识别的转义字符，如&转义为&amp; 修改文件report.py  3、http重发次数增加变量，可定制重发次数、restful的parse\_response新增detail信息返回  4、ndlib安装新增mongo模块安装  5、runner推送策略优化，开启线上监控，除了没有失败次数，其他情况都发送报告  6、runner消息推送格式优化成小卡片形式  7、token使用ndlib下jlibs目录的token.jar  8、nd\_path.py文件新增获取工程目录接口  9、conf文件新增路径传入参数，完善nd引用分离后，nd不能获取工程目录问题  10、runner的run新增路径传入参数，完善nd引用分离后，nd不能获取工程目录问题  11、优化其他文件使用db\_info.py文件的mongo信息连接mongo失败后，终止工程执行问题，提高容错性 | 2017-06-14 |