# 01 基础代码生成

apitest框架可根据swagger接口文档生成部分接口及测试代码, 依赖swagger2.0文档,能够大大简化测试开发流程

以下流程以nebula项目为例,[接口文档地址](http://open.argus.sensetime.com/nebula/docs/v1)

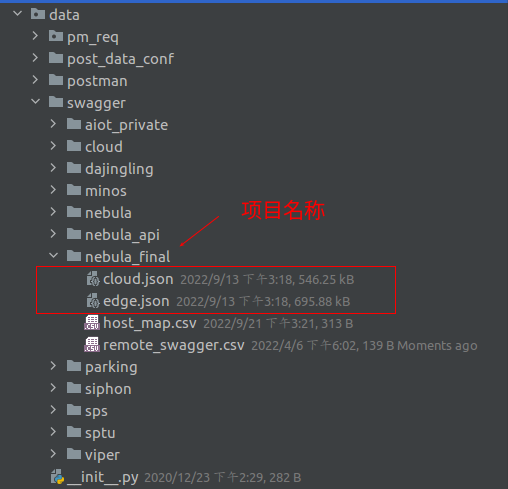
# 创建项目目录

在apitest/data/swagger目录下创建存放swagger的文件目录, 这里以nebula\_final为例

# swagger.json文件下载

~~方法一: 手动,通用方法 建议使用方法二 自动更新~~

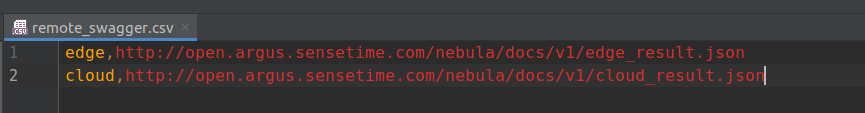
~~下载swagger.json文件,保存到apitest/data/swagger/<项目名称>/目录下~~



方法二: 自动更新

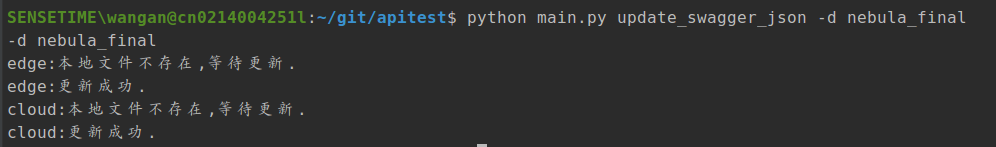
|  |  |
| --- | --- |
|  | 支持的访问地址 |
| 1 | 不需要鉴权的http协议的swagger地址 |
| 2 | 公司内gitlab代码库中的swagger地址 |
| 3 | 公司内Confluence中的swagger地址 |

* 在apitest/data/swagger/nebula\_final目录下添加remote\_swagger.csv文件, 该文件下中记录swgger文件的地址, 如下图:



* 执行如下命令, 拉取最新的swagger.json文件

python main.py update\_swagger\_json -d nebula\_final



备注1： 该方法可以提示swagger文档是否有更新

备注2： 该方法目前已经支持win及linux系统

**通过以上两种方法的一种, 即可获取swgger.json文件**

# 配置hostmap (可选)

用于配置接口的host, url中的前缀

通常的一个url路径在apitest中会被拆解如下:

url: http(s)://<host>:<port>/<suffix>/path

swagger文档中通常只是标明path路径, 而host, port, suffix需要根据实际的部署的服务,或gateway进行拼接

由此,引出hostmap的功能, 通过该文件的配置,可以根据不同service进行单独的url配置

配置方法如下:

* 在apitest/data/swagger/nebula\_final目录下添加host\_map.csv文件, 如下图:



* 当不添加host\_map.csv文件的时候, 不会生效
* 当host\_map.csv文件中未填加的服务, 不会生效
* 当host\_map.csv文件中的服务名为all的时候, 该json下的接口都会生效

备注: 根据以往的项目经验,目前只支持一级结构的json文件, 后续可根据实际项目进行代码的适配

# 配置case\_map(可选)

用于拆分子模块测试用例的目录，比如belt下分为console，ids，ras三个模块，为了将测试用例分别放在各自的目录下，配置该文件即可：

格式为 “模块名, 子模块名”

# test\_case目录分类

# ras-manager

ras,device

ras,rasManager

ras,botManager

ras,rasCluster

# ids

ids,face

ids,identity

ids,ocr

# ras-runtime-proxy

runtimeProxy, ipsApplet

runtimeProxy, ipsOcr

runtimeProxy, viperApplet

runtimeProxy, viperOcr

#runtimeProxy, sfd

# aide

aide, adapter

aide, viperOpenApi

# console

console,auth

console,notification

console,notificationInternal

console,user

console,bill

# 配置interface\_list.csv

1)查看接口信息：

python main.py list\_swagger -d belt

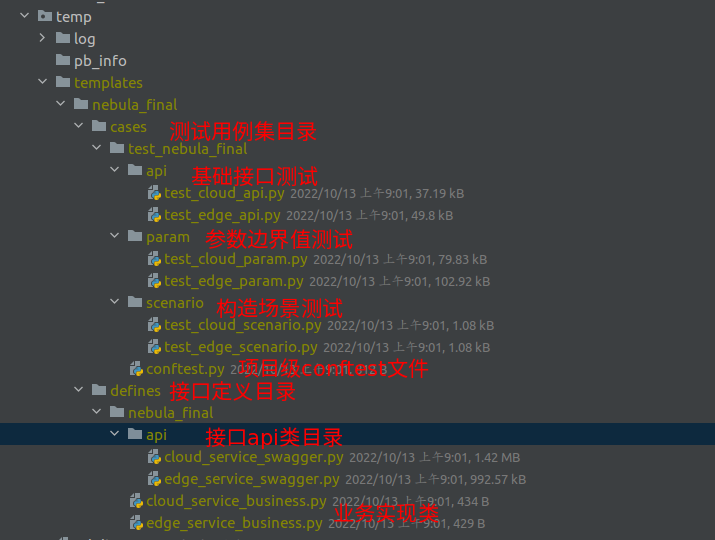
２）找到新增的接口列表，将其添加到interface\_list.csv即可

# 生成代码

当放好swagger.json文件后, 执行如下命令, 即可生成代码

python main.py gen\_all\_template -s nebula\_final

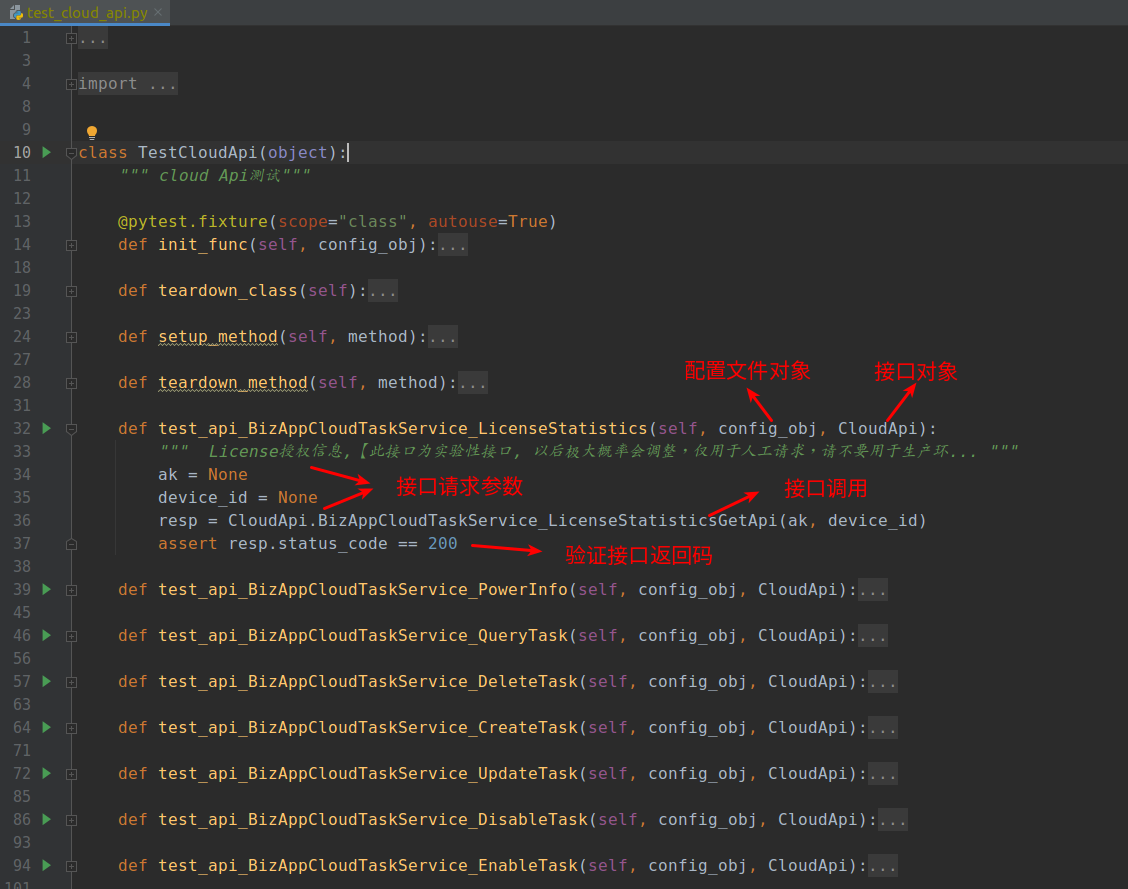
生成的代码,默认路径为apitest/temp/templates/<项目名称>, 如下图:



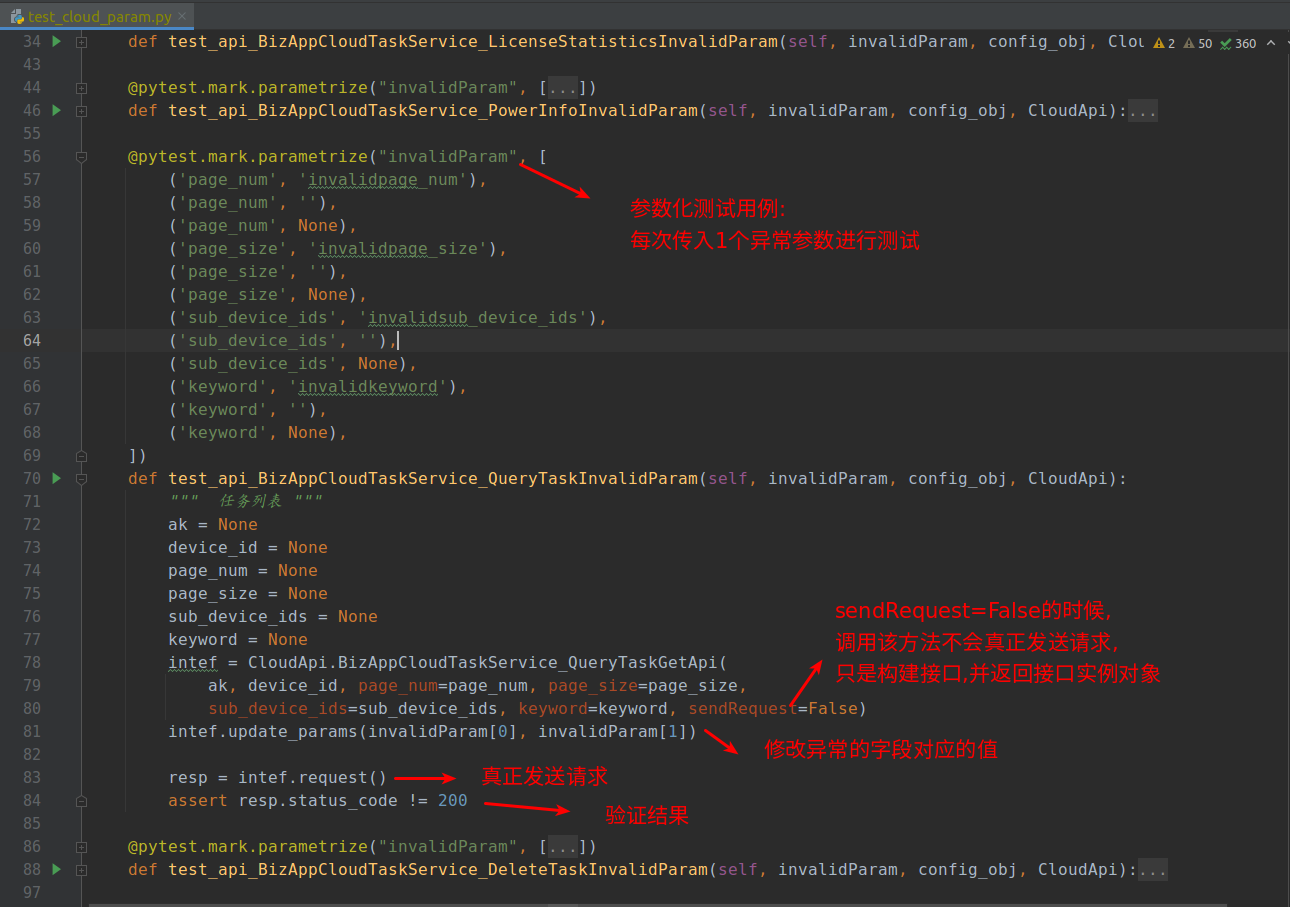
项目下分为两个目录:

* cases: 测试用例目录, 目前会生成api测试类, 参数测试类以及场景测试类
  + api: 单接口测试用例,  包含所有单接口测试用例代码
  + param: 参数测试用例, 包含所有单接口默认异常参数的测试代码.异常参数包括(异常值, 空值, 参数及参数名不存在)
  + scenario: 场景测试用例, 只是一个空文件,需要根据需求,构建场景测试
  + conftest: pytest的配置文件, 默认生成接口对象的fixture以及类级别缓存的fixture代码
* defines: 接口定义目录, 生成api接口目录, 以及后缀为\_business的业务实现类, 其为api下接口类的子类

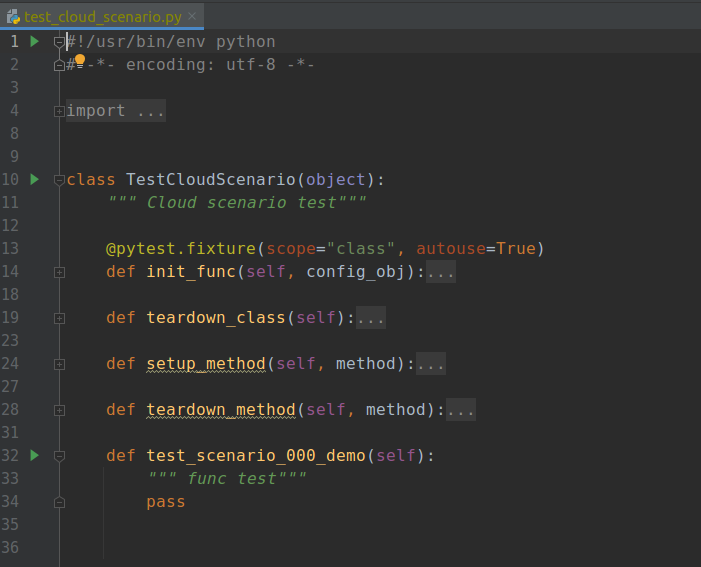
api如下图:



param如下图:



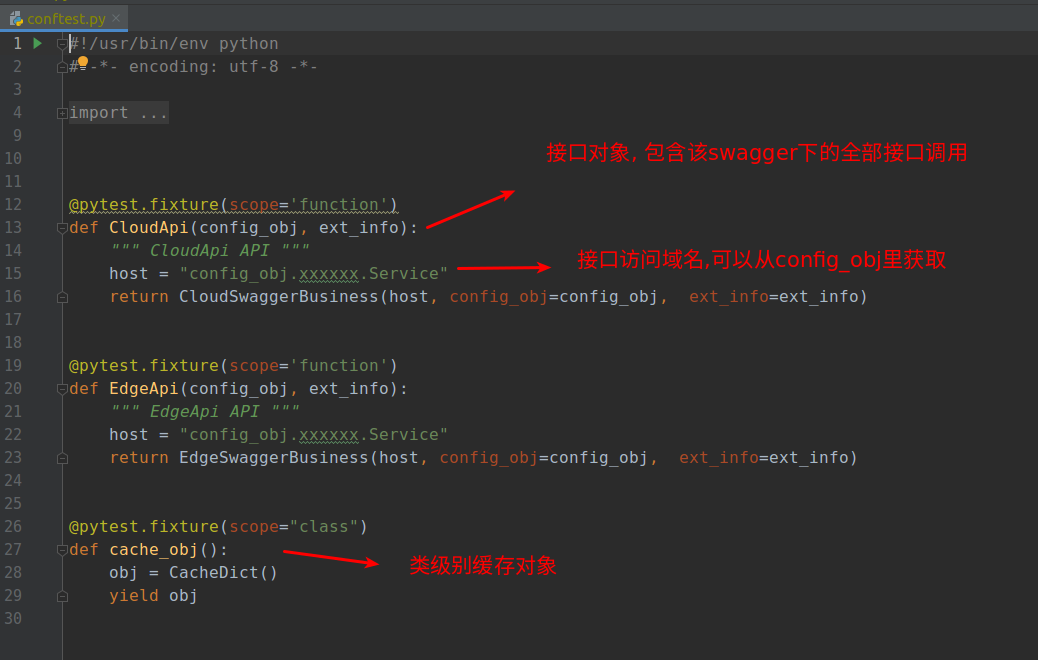
scenario如下图:



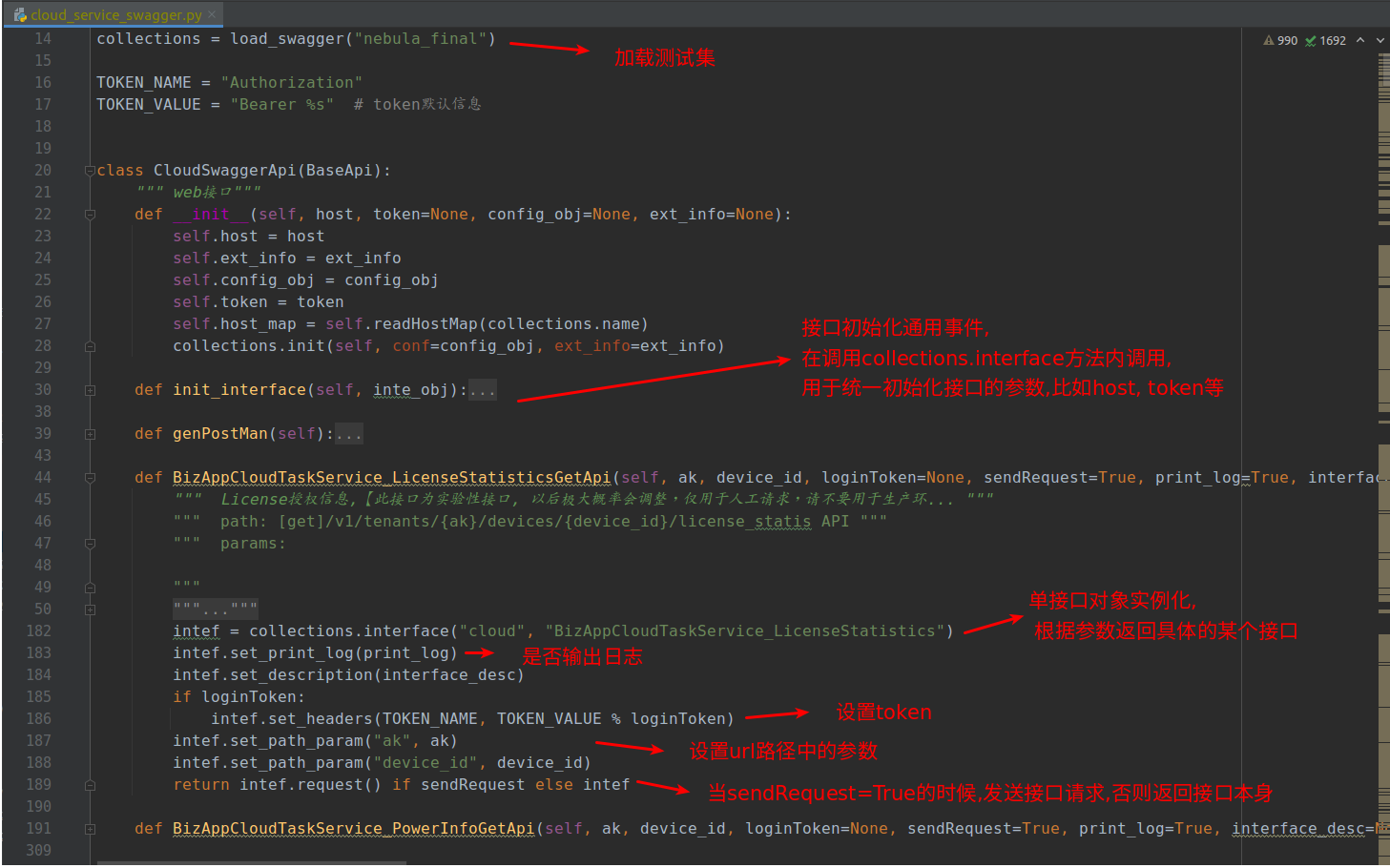
**场景测试用例需要根据业务场景进行开发编写.**

**我的理解是: 先跑通单接口测试, 再根据业务场景,编排接口的调用流程, 在主要节点添加验证, 即可完成场景的测试用例编写**

conftest如下图:

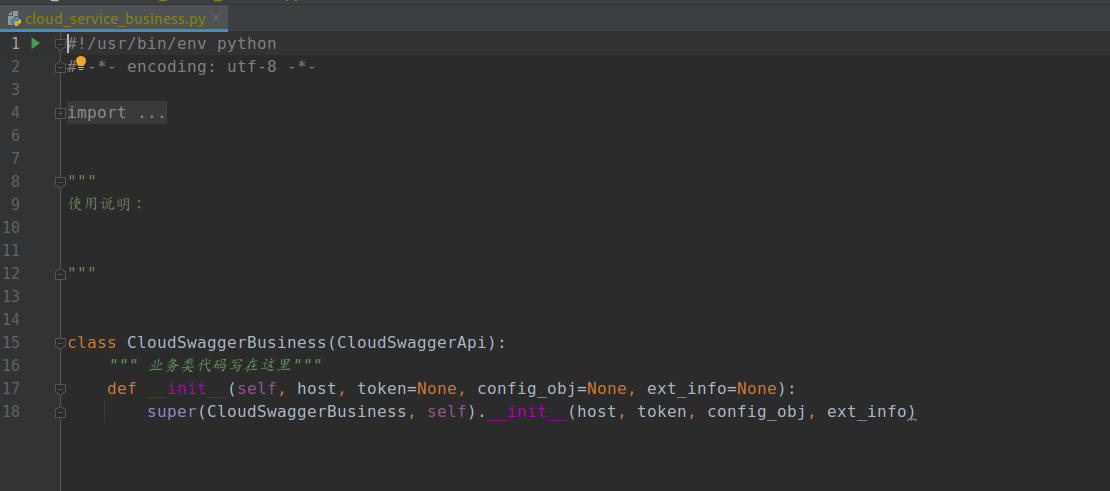


defines/<项目名称>/api/xxx\_service\_swagger如下图:



**该类为基础接口类, 单独抽象出来, 无需修改, 如在测试过程中,接口文档有更新, 则只需从新获取swagger.json文件后,生成新代码替换就可以了**

defines/<项目名称>/xxx\_service\_business 如下图:



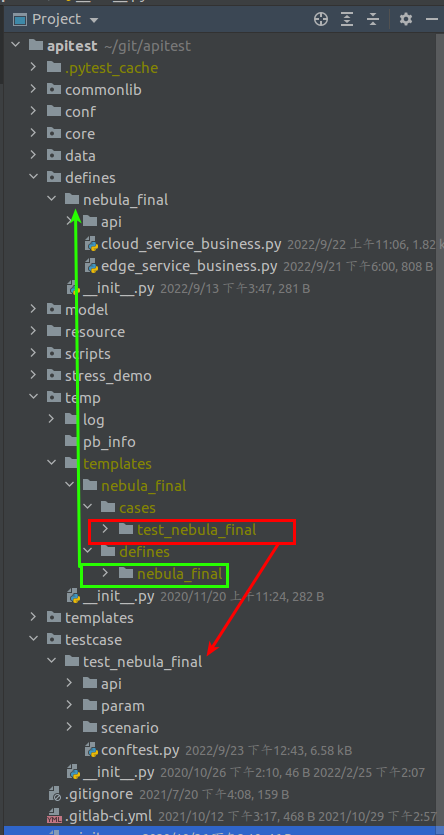
该类为api类的子类, 用于扩展业务功能, 比如需要封装一个方法,实现子设备的删除,以及确保删除成功,即可通过定一个一个删除方法, 方法中先调用删除方法, 再调用查询方法.

# 代码拷贝

将apitest/temp/templates/<项目名称>/cases/test\_<项目名称> 目录拷贝到testcases目录下即可

将apitest/temp/templates/<项目名称>/defines/<项目名称> 目录拷贝到defines目录下即可

**如果不是首次拷贝, 则需要进行代码的对比, 请勿将已写代码覆盖掉**



至此, 代码生成部分补充完毕