# 03 单接口测试用例开发

# 回顾

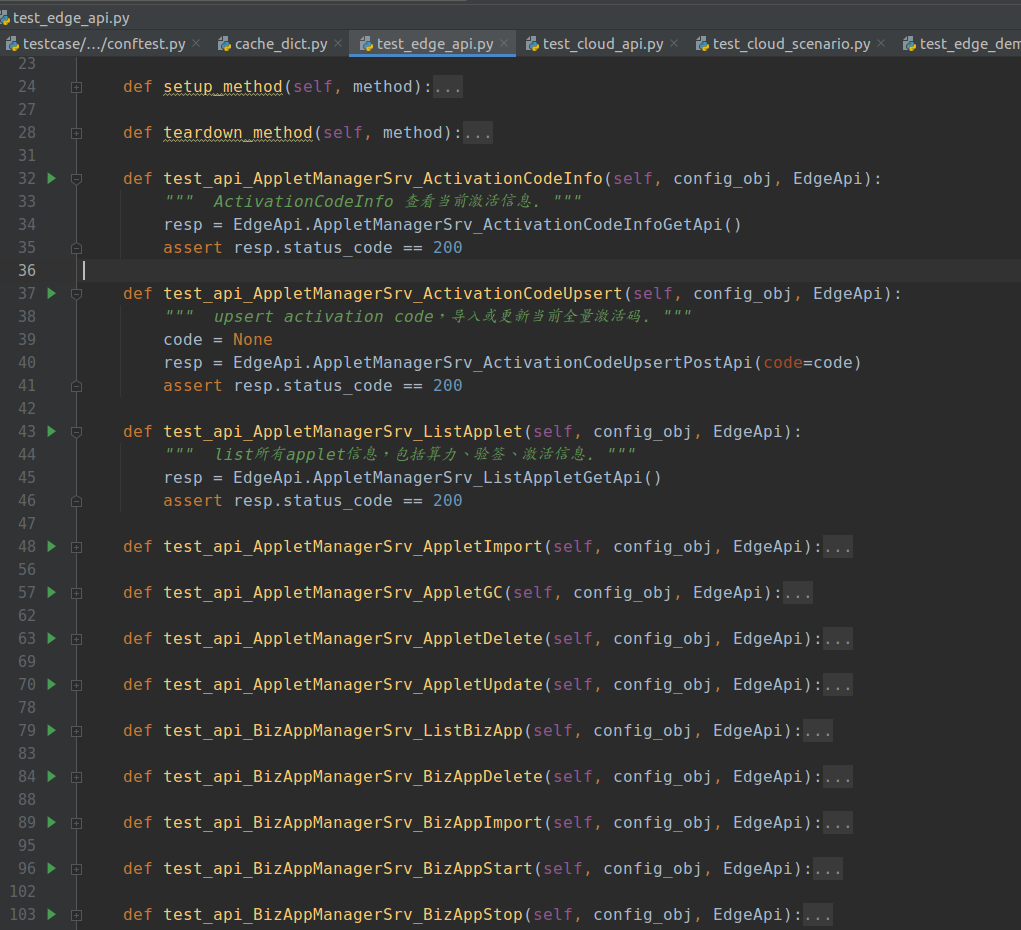
根据前文的步骤,  我们已经完成了以下几个步骤:

* 生成了大部分代码, 拷贝到指定的目录下
* 配置了conf下的toml配置文件, 填入了一些数据参数
* 完善项目级conftest的代码, 使接口对象可以正常调用

接下来, 就可以正式开始测试

# 脚本开发

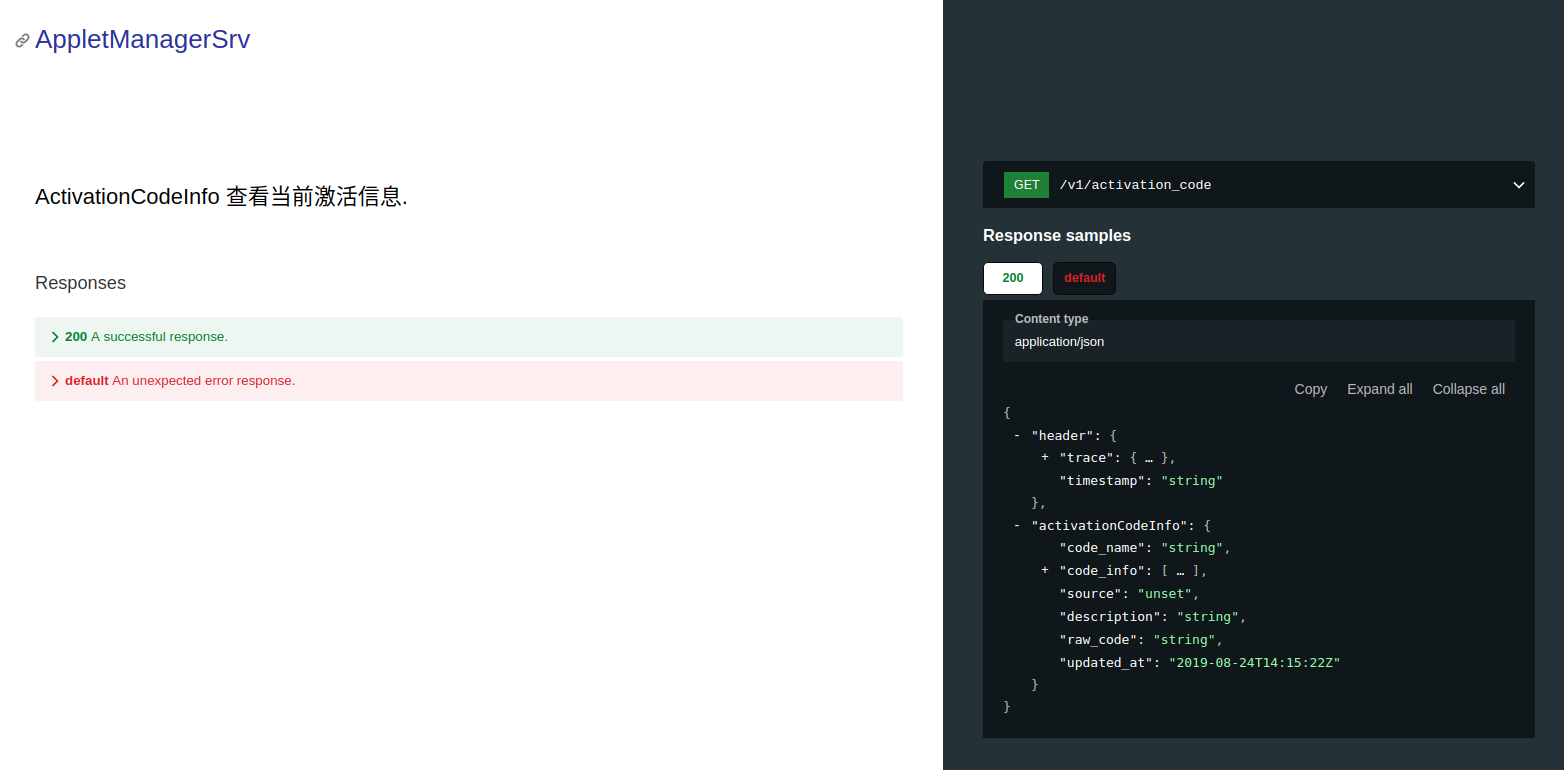
首先我们打开单接口测试用例的脚本文件:apitest/testcase/test\_nebula\_final/api/test\_edge\_api.py



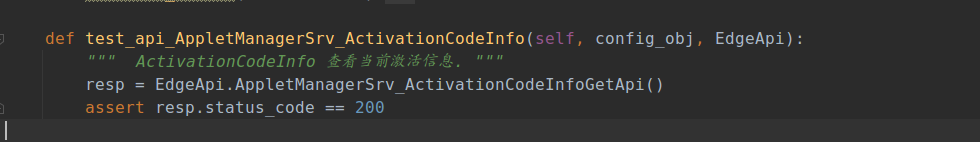
我们可以发现, 在这个生成的脚本中,包含很多test\_开头的方法, 这些都是符合pytest基本规则的, 每个方法都对应一个swagger中的接口,我们只需要填充这些方法中的请求参数,

根据需要,添加对接口返回值的验证,即可完成简单的单接口测试

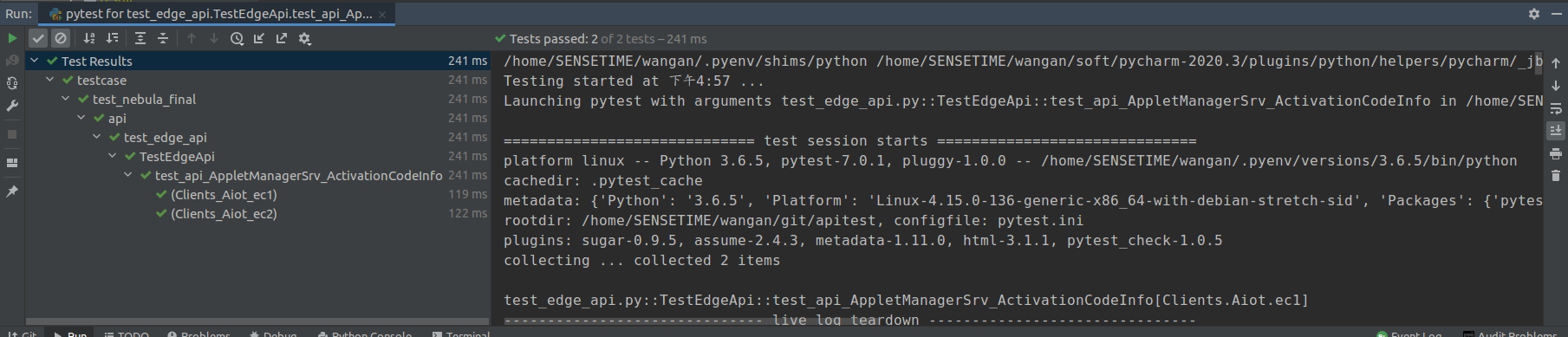
我们先看一个最简单的例子, 我先贴出swagger接口文档作为参照:



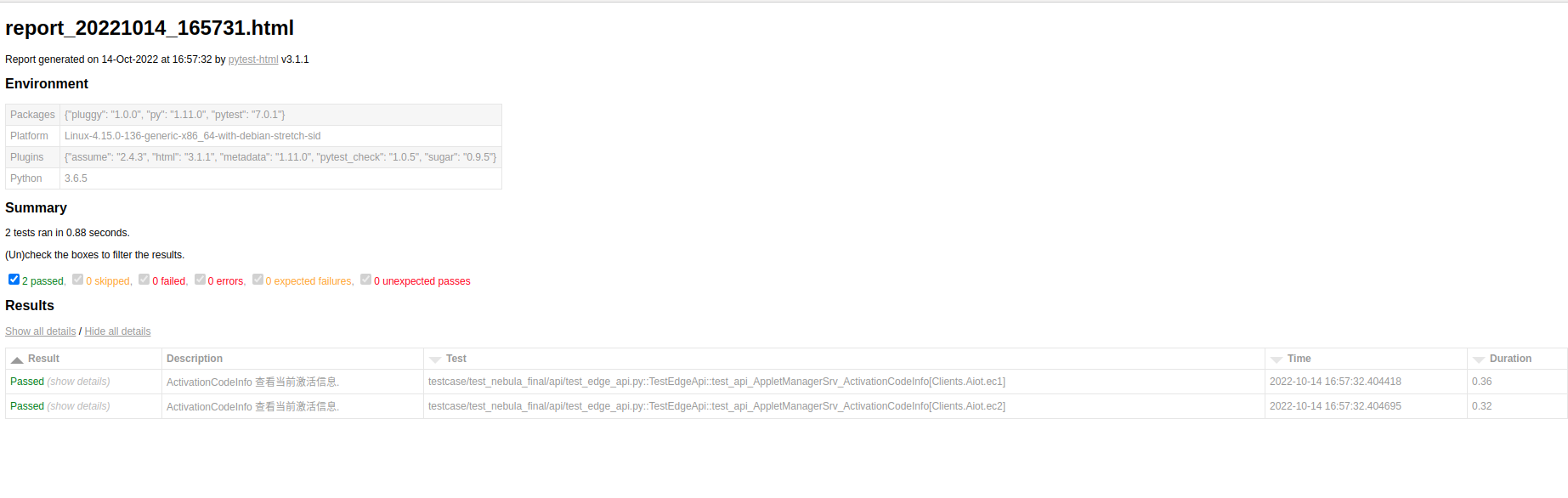
测试代码如下图:



根据注释,我们知道,这是一个"查看当前激活信息"的接口, 这个接口并没有入参, 因此我们其实不需要做更多的事情了, 直接运行这个case, 就可以调用, 我们执行试一下,如下图:

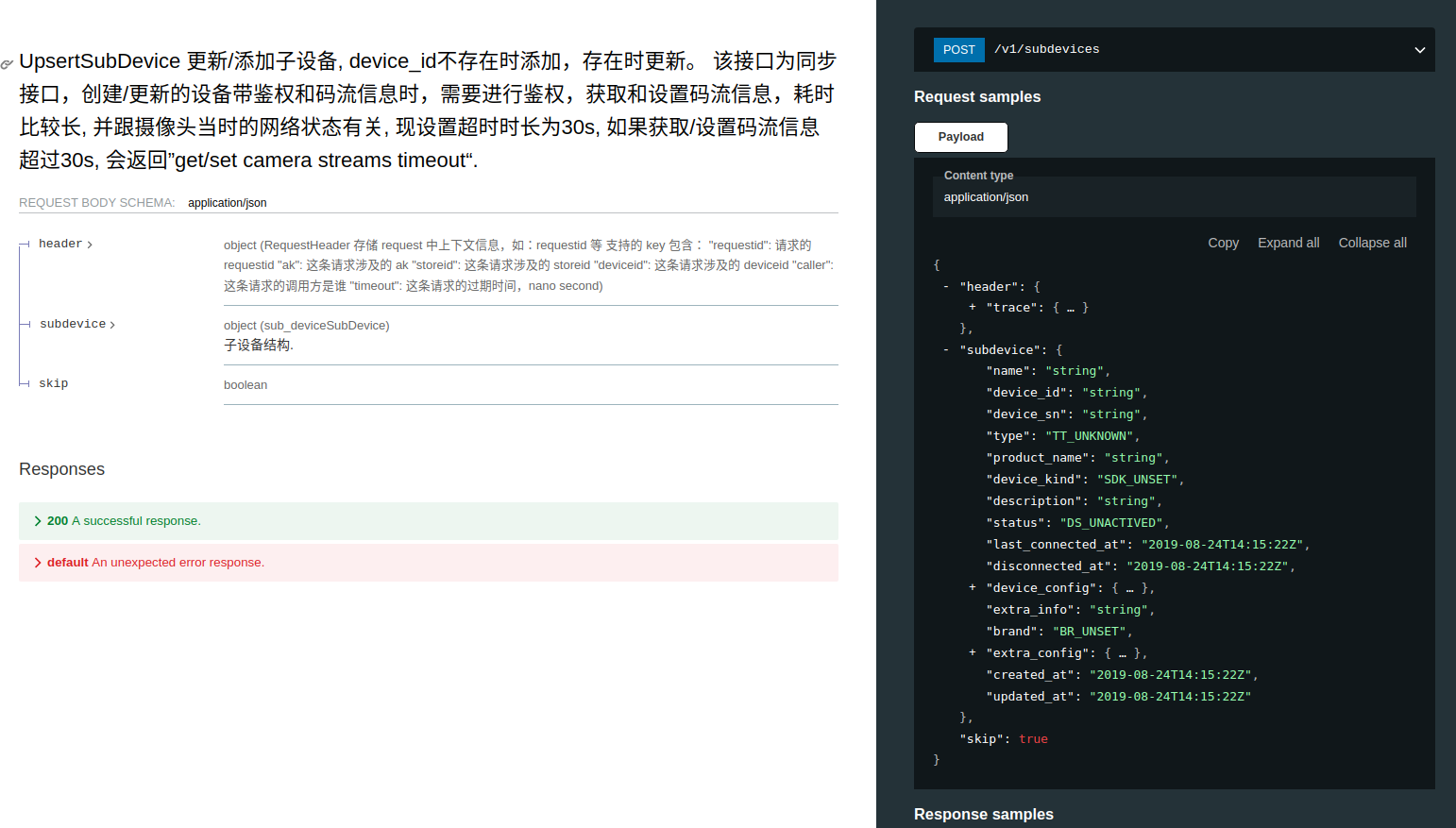


pycharm中的执行结果, 左边告诉我们,执行了两次, 是因为我们在配置文件中为client\_info添加了两个端的信息, 右边窗口是执行过程, 同时还会生成html的测试报告,如下图:

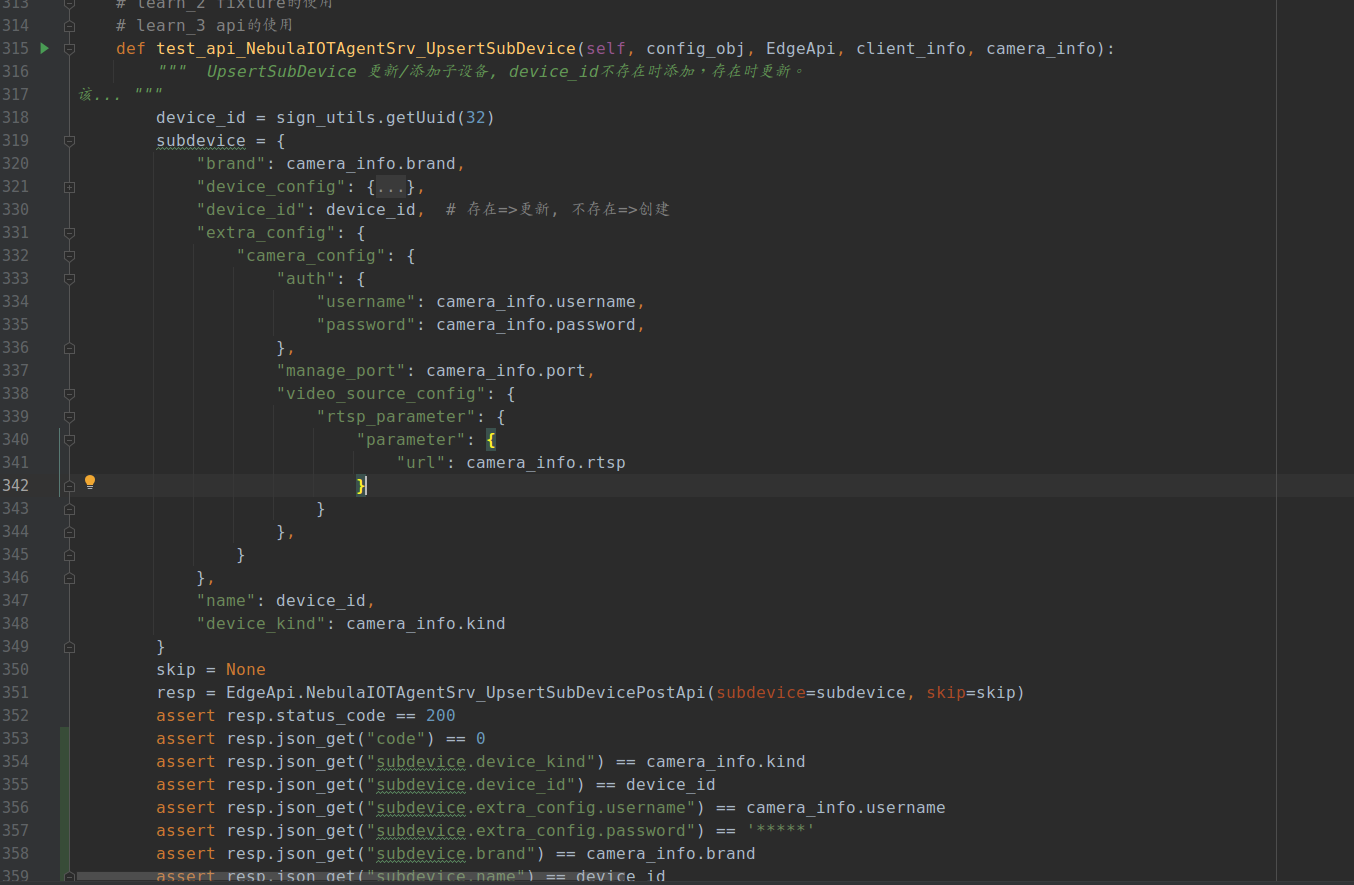


接下来,我们看一个需要传参数的接口的例子:

同样先放出接口文档:



我根据入参填充后的代码如下:



这是一个子设备创建接口的测试用例, 从图中可以看到,我们需要为UpsertSubDevice接口传入两个参数, 分别为subdevice和skip.

subdevice实际是一个较复杂的结构,里面需要填充必要的摄像头信息,id等数据,这些数据是来自于fixture: camera\_info

当接口调用后,会返回resp, 通过封装的resp.json\_get('字段名')的方法, 方便的获取指定返回路径的值, 通过assert判断返回值是否符合预期

更多的例子可以直接参考代码, 单接口的测试就补充到这里啦

最后再总结下, 单接口测试通常只是验证接口单独调用的时候,是否可用.接口功能是否符合预期

对于接口测试, 更多的还是要贴合业务逻辑, 构造测试场景, 进行多接口的场景测试, 来满足业务的需求, 就比如"子设备的crud", "不同品牌子设备的crud", "任务中使用子设备的流程"等场景.