灯杆云接口自动化项目说明

**前言**：

为保障系统高质量稳定运行，及时排错纠错，对于功能趋近稳定的系统的日常回归，比较消耗人力和时间资源，也容易产生遗漏，所以现引入自动化概念，帮助测试人员高效快速发现问题。

对比从前端入手的UI自动化来说，接口自动化有以下几点优势：

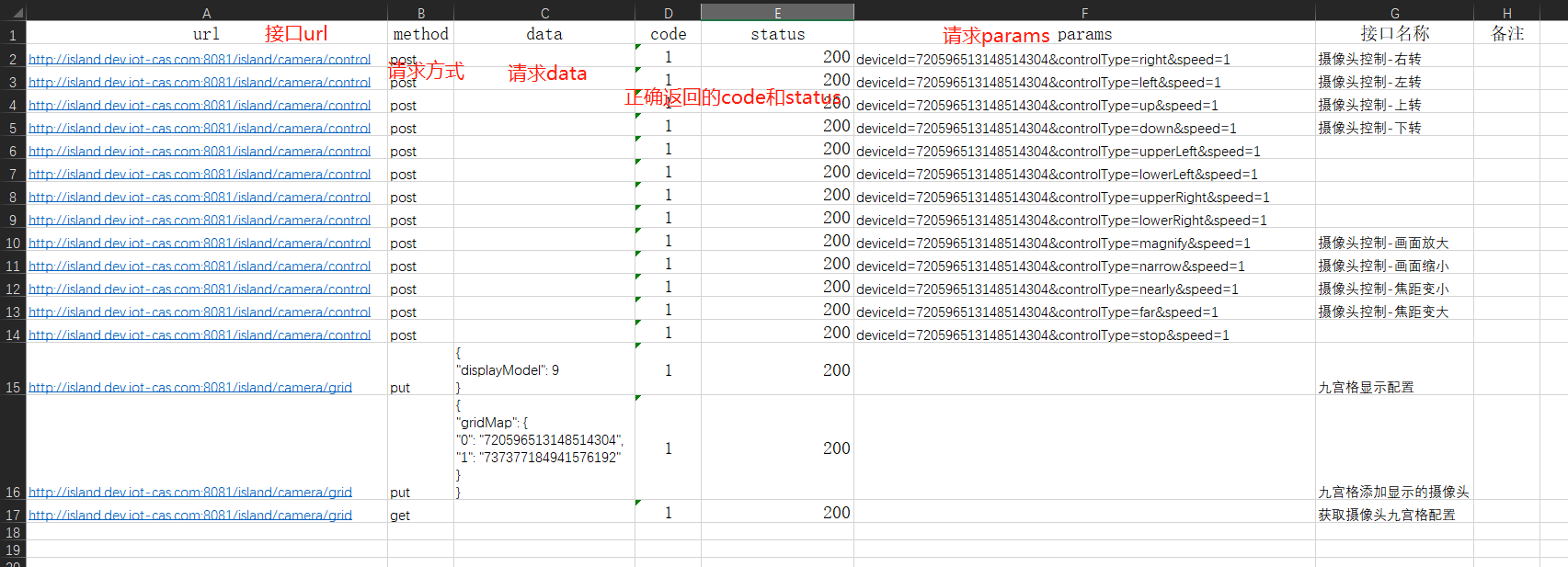
1. 执行效率高，平均不到1秒完成一个接口。
2. 代码维护成本低，复用率高
3. 不受前端页面影响，不容易因浏览器、网速的不稳定因素导致用例运行失败

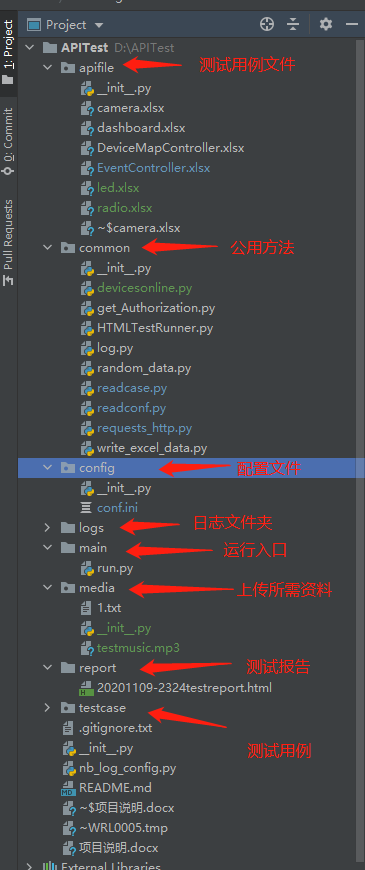
所以综合考虑以上原因，采用Python+Requests结构完成此项目

**项目说明:**

方案

目前我们的接口都是使用的http协议，其测试的基本原理是模拟前端（客户端）向服务器发送数据，得到相应的响应数据，从而判断接口是否可以正常的进行数据交换。在测试的过程中尝试过两种方式，一种是利用性能测试工具Jmeter模拟客户端发起http请求，另外一种是使用python脚本直接编写脚本模拟客户端发起http请求。利用Jmeter工具配置，需要对如何利用Jmeter进行性能测试熟悉，通过相应的配置可完成，但不够灵活，编写脚本效率不高，兼容性相对较差，所以不选用Jmeter。直接用python脚本进行，模拟http请求也就几行代码就可完成。但只是模拟请求不是最终的目标，也需要易用，不会编码的人也会维护我们的测试用例，所以形成了现在的形态，遵循了测试框架的一些基本原则，业务逻辑与测试脚本分离，测试脚本与测试数据分离。大致框架如下图所示：

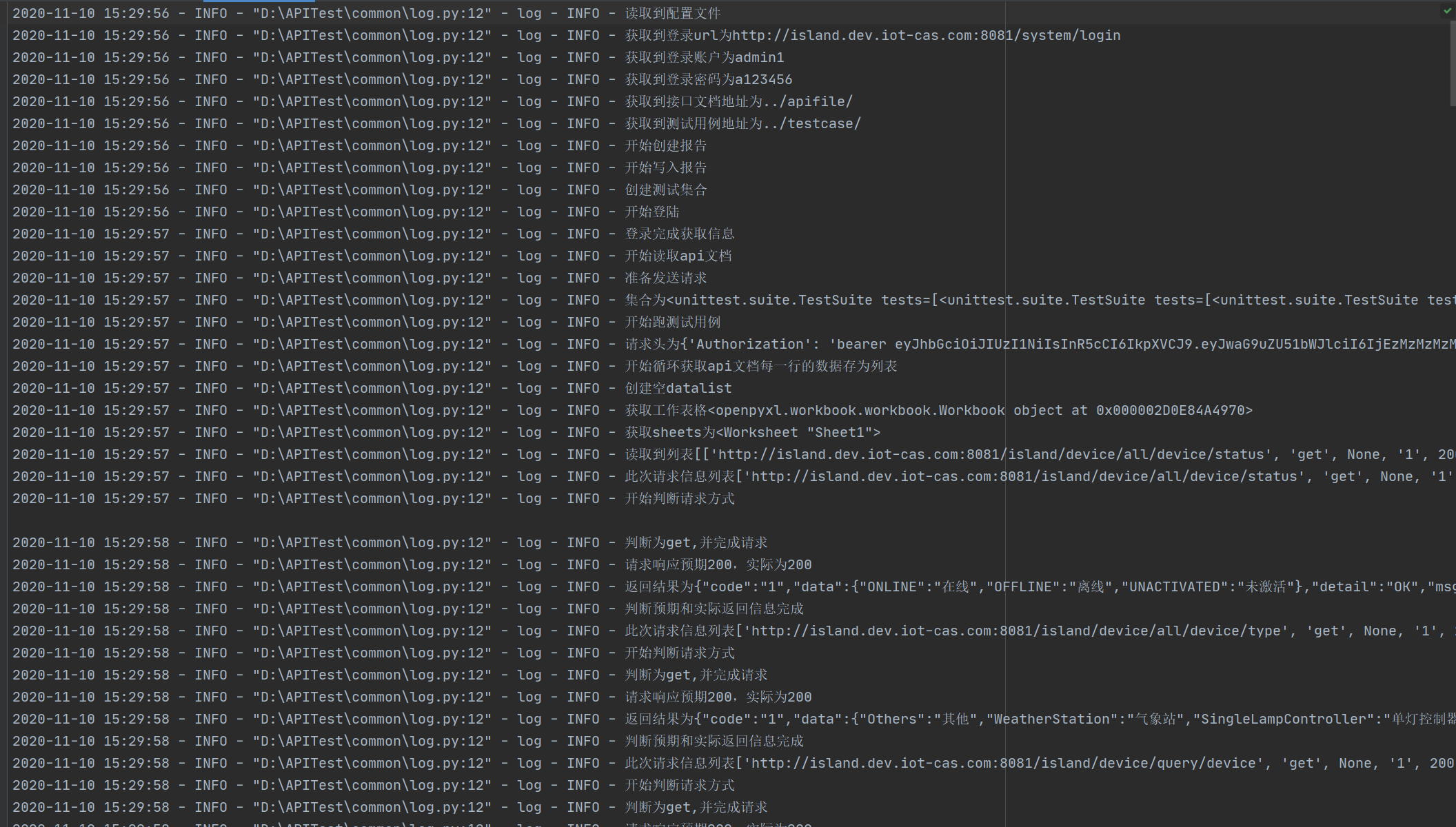




运行流程：

读取配置文件获取url,username,password,filepath等初始化配置——>创建空白测试报告模板——>获取所有testcase打包成测试集——>获取登录态——>读取excel数据，根据关键字判断请求类型——>传递请求所需字段——>获取返回值——>判断请求是否成功——>写入测试报告，同时写入log

**测试报告及log样式展示:**



**项目特点及如何使用:**

1. **支持所有类型请求**
2. **支持单独用例调试和扫描全部用例运行**
3. **真正意义上的自动化，一次编写每次运行无需改动**
4. **支持CI定时运行，发送测试结果（待完成）**

**如何使用:**

**只需把每个接口的参数，按照excel表头字段填写数据，在testcase文件夹内新增xxx.py，可从已有文件复制，改动相应的excel文件名，最后运行run.py，即可完成测试。**