接口测试基于Python的Request框架构建

今天主要讲讲接口测试的中的 业务流

接口测试主要分为两大部分 单接口 和业务流

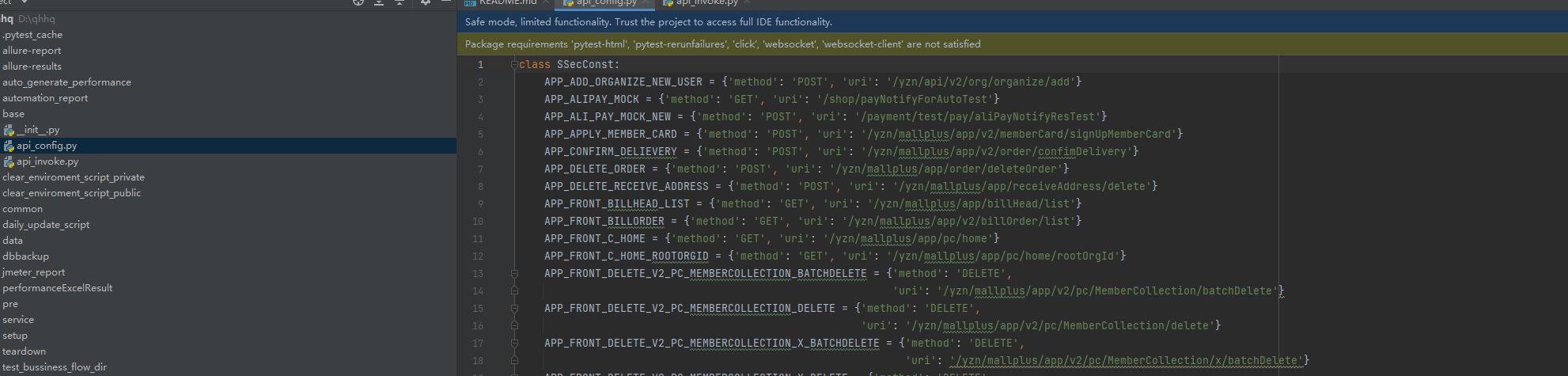
单接口主要侧重在对单一接口的每个字段通过正常异常的输入 以便验证接口的稳定性以及异常输入处理的正确性

而业务流主要用于测试不同接口串联起来 来完成一定的业务逻辑 主要验证多个接口之间的协调 不关注单个接口的正常异常输入 而是对于多个接口协调完成业务的正确性

这个培训主要是给大家介绍一下接口测试的框架 如果大家有更好的框架优化建议可以提出

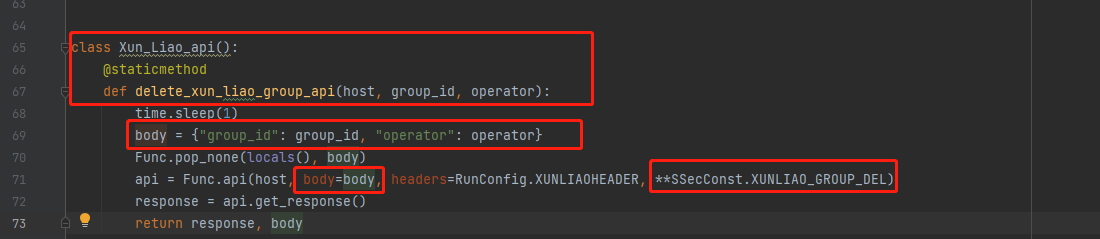
接口测试框架分为5层 总体的策略是降低代码的耦合度

第一层 维护请求的URL 一般叫api\_config



前面的部分是 请求方式 比如GET POST PUT

第二层是 api\_invoke层次



先定义一个类 并且建议以\_api结尾 在Python中类名是以\_间隔不同的词汇

需要关注的是 body 中是你需要发送的信息 通过上层代码传递进来 发送请求的方式分为 body ,param ,data 分别对应POST请求发送消息的方式body param(请求是? 但是发送方式是POST这种特殊情况) 和form data GET 请求 统一使用Body

\*\*\*\*SSecConst.ENTERPRISE\_BACK\_POST\_YZN\_MANAGER\_LIVE\_ADD

对应上述api\_config 层次中的方法和url

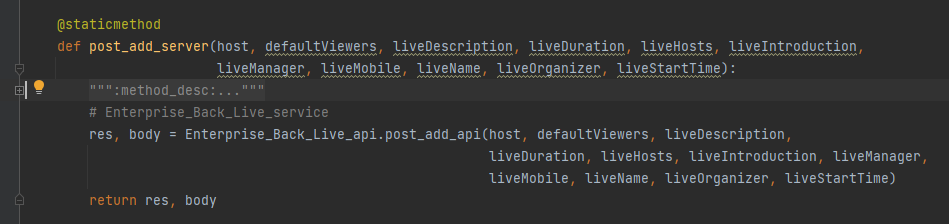


python中\*\*用于对Dict 类型的对象根据dict 的key 和value 拆分为key=value的形式作为变量和变量值传入方法的功能

另外Func.api 封装了python 的 request的请求 过于复杂略过 只要需要调用即可

返回response和拼装好的body

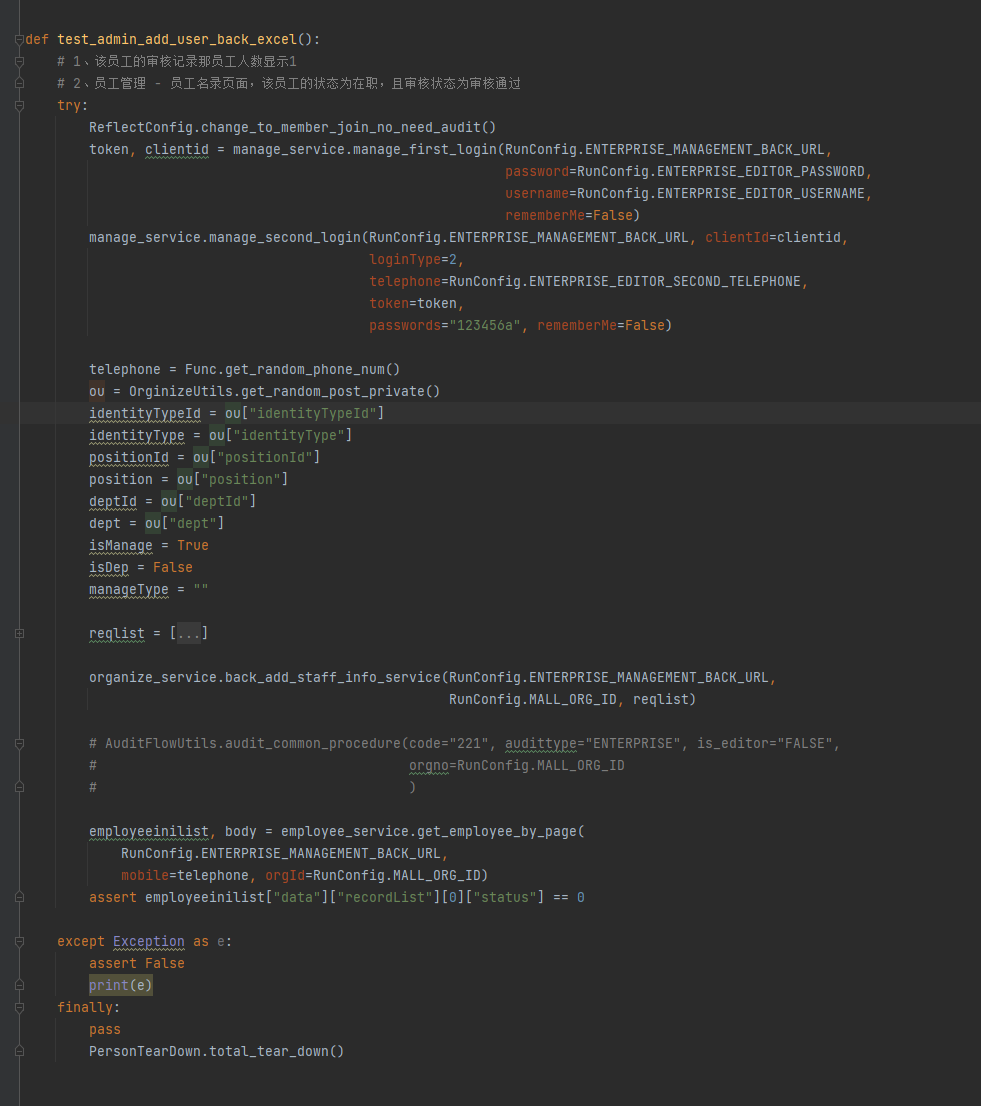
第三层是 Service层次



用于接收用例层次 传入的参数值 中转并传递给API层次通过上述的Func.api 封装的request方法发送出去 并把api层次返回的reponse 和 body 通过上述截图的res body 返回给用例层次

第四层就是用例层次

以这次主题 业务流的一个案例为例子



比如一个添加后台用户的用例

1 管理员一级登录

2 管理员二级登录 （之前公司需要2级登录）

3 调用获取随机职务的接口并从接口返回的dictionary中获得dept等信息 用于添加用户的职位信息

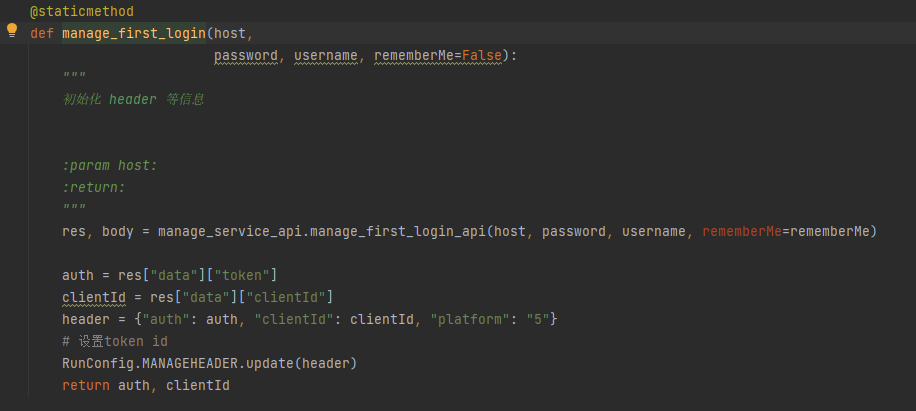
4 调用添加用户service (架构就是之前说的4层架构)

5 调用封装的搜索接口获取第一条记录检查status是否为已生效 判断添加成功如果成功assert 结果是通过否则失败

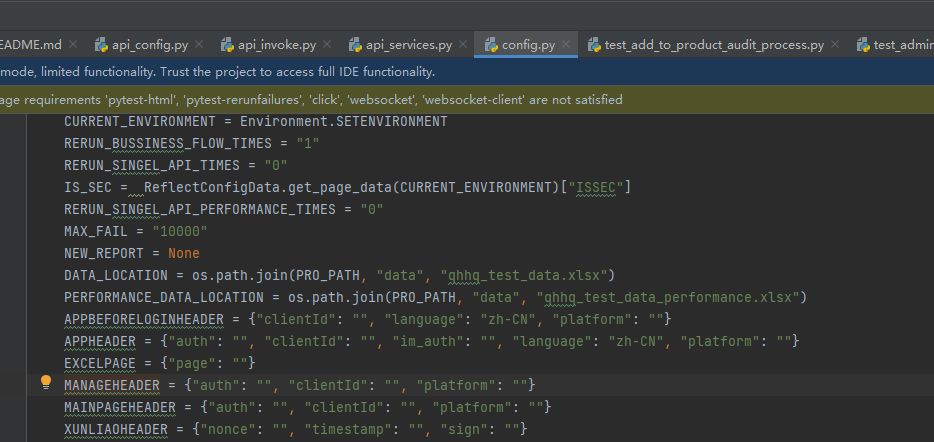
6 最后finally 调用封装好的清理方法（全量数据库恢复包括mysql,mongo,redis,MQ）

就这样完成了一个简单的业务流的过程

其中登录方法登录后的获得的Token 通过



登录后Header更新到全局变量中的Header信息 然后别的需要就直接取 如果再次登录会被覆盖 后续的继续取更新后的 header信息



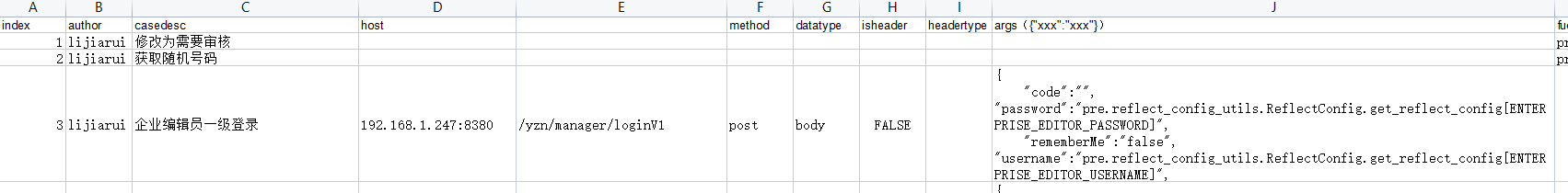
每个平台不同的Header用以区分 比如APP\_HEADER 表示手机端HEADER

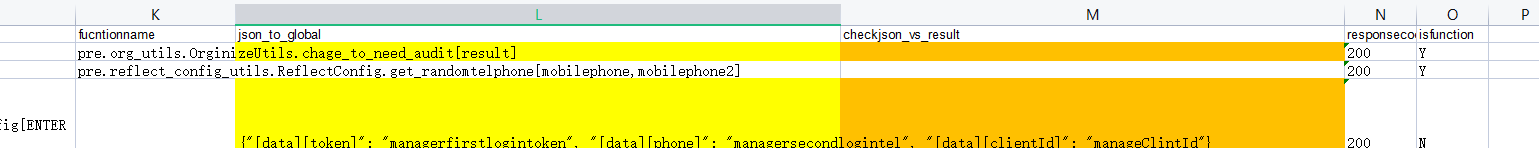
MANAGEHEADER表示后台管理平台的Header

还有另外一个种对于功能测试人员更加友好的Excel驱动方式 代码基本不用写维护

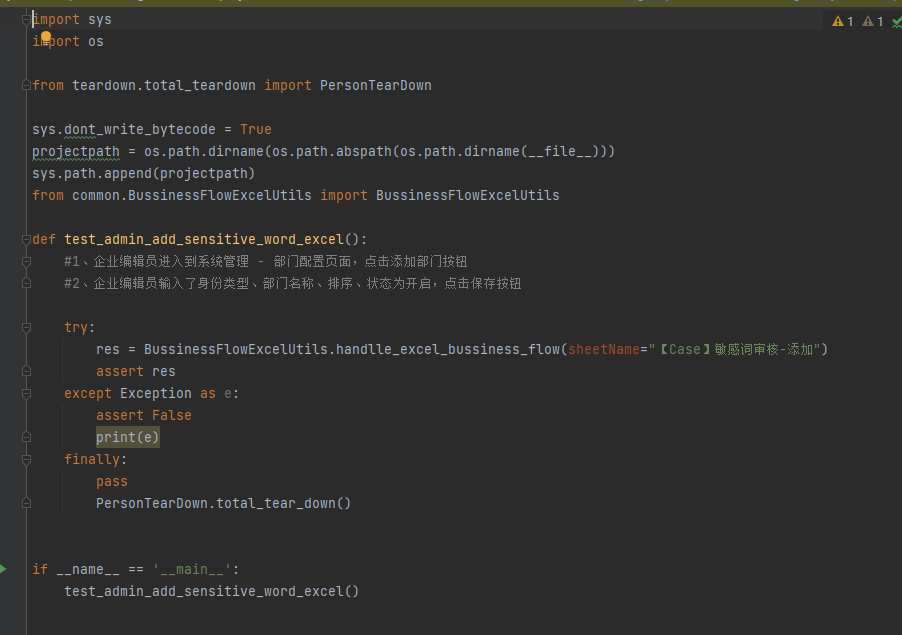
Excel如下







分别为 步骤编号 负责人姓名 步骤 服务器IP 请求URI 请发方式 发送数据方式，是否需要头信息，请求内容 (json)格式 内部支持值 从全局变量获取,方法调用获取,固定值都是通过解析引擎实现，调用方法名称，本步骤需要存入全局变量的值 通过{“key”:”value”}方式 key 为返回的response的中获取指定值的层次比如[“user”][“userId”] Value 为 需要存入的全局变量 已被后续使用 最后橙色那栏 表示要从返回结果中获取值并和预期结果比对 同样也是{key:value}方式 Key 也是 回的response的中获取指定值的层次比如[“user”][“userId”] value为你想要比对的值 同样支持固定值 调用方法 上一步获取的值 如果需要对比多个多个Key-value即可 responseCode 是比对responseCode 是否符合预期 最后isFunction 表明这一步是否这只是需要调用方法而不之家发送请求



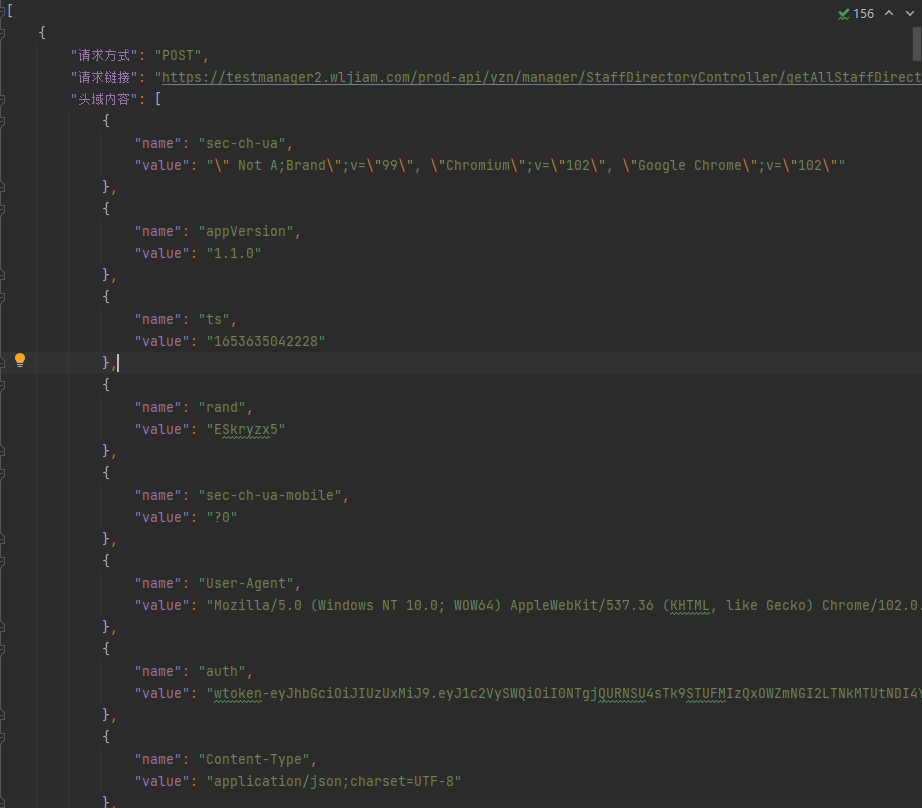
这种方式代码很简单 只需要定义方法 调用引擎 输入Excel标签页 最后通过TearDown 清理测试环境 Ctrl CV 即可 对功能测试人员 通过核心引擎事项就是途中

BussinessFlowExcelUtils.handlle\_excel\_bussiness\_flow

过于负责就不解释了

最后 目前已经实现了 通过浏览器插件录制请求

插件录制后



通过一定的方法解析到上述的Excel 中 然后再通过Excel驱动测试