# ----interface\_autotest----

# 接口自动化的流程：

## 把接口文档的 接口放postman调用一遍

### 1.1什么是服务？

### 服务就是通过一个容器把我们写的代码运行起来。

### 1.2什么是容器：

### 1.3没有接口文档怎么办：

#### 抓包

### 1.4 接口调不通怎么： 找开发排查接口环境问题

## 接口调通后：开始写case 用例

## 开始敲代码 逻辑操作流程： 3.1 主流程

## 引包 pytest 创建测试模块类

class TestLogin:

## 读取case 文件 从模块 python 操作 excel 文件读写工具的封装方法引用ReadExcel 类 先读取一条

cases =ReadExcel.reat\_data\_pl(DATA\_FILE,’business\_token\_login , 1,1)

## 需要传文件的路径，要从模块constants 引入数据文件的路径

## 测试参数化需要添加装饰器

## [@pytest.mark.parametrize(‘case’,cases)](mailto:添加@pytest.mark.parametrize('case',cases))

## 声明一个方法

## def test\_login (self , case ): 发送请求格式 get post 请求的reponse = requests(url = case.url 请求数据 data = eval(case.case\_data) ) 响应是json格式 res = reponse.json()

## 断言的判断 try :

If case.case\_id == 1  
 assert case.expect in str (res)   
 else:  
 assert eval (case.expect) ==res  
 except AssertionError as e:

## 把断言的结果写入用例数据文件中 ReadExcel.write\_data(DATA\_FILE,’busines\_token\_login’,case.case\_id,7,’失败’

## 需要发 错误级日志 堆栈回溯日志 抛出断言错误信息

## Logger.error(‘测试编号{}，测试标题{}，执行失败！实际结果{}‘.format(case.case\_id,case.case\_title,res)) logger.exception(e) raise e

## 在请求前要做 数据的回滚 比如：登录时需要 从数据库里找到用例里一样数据 做 删除，后再添加条数据。 在请求后要做 数据验证 就是数据落库 比如：注册：正常的注册成功后 要验证一下数据库里是否有此条数据

## 从 模块数据库封装工具类 db\_utils里引用DBUtils

## 测试这个模块用例时需要 操作数据库联接 与关闭 那就使用夹具来实现 就放在 开始创建 class TestLogin 类 之后 声明测试用例方法之前 加入夹具 @pytest.fixture(autouse = True) def connect\_db(self): self.db = DBUtils() yield self.db.close()

## 追加其他用例case 和allure 报告 这里会用到 import allure [放在 class TestLogin类名前使用 报告中显示 类名的 特征 名称 @allure.feature](mailto:会用到@allure.feature)(‘商品登录模块‘) 放在模块 类 与 用例方法之间 @allure.severity (‘critical’) 报告中显示此模块的极别 @allure.description(‘商品登录模块接口测试用例‘) 对此模块的接口的描述 allure为动态标题 根据用例里 case\_title 起名 可以附加上attch 在用例方法后添加 allure . dynamic . title (case.case\_title) allure . attch ( boby=case.url , name = ‘接口路径’) allure.attach( body = case.case\_data , name='请求参数')

## 在响应结果 res = response.json() 后面添加 allure.attach( body = str(res) , name = '响应结果')

## 最后 我们来自测下 if \_\_name\_\_ == '\_\_main\_\_': pytest.main(['-vs','./test\_login03.py'])

**登录接口测试 ：  
注意 数据回滚 数据验证**

**商品注册 ： 注意数据回滚 数据验证**

## Token\_商品查询 接口测试

## 商品查询接口：核心是查询的结果

## 注意：怎么获取token值：

## 首先在配制文件里config.ini 写入登录用户的数据

## 通过配制文件封装的方法获取到token值

## 最后就是表格数据里使用占位符#token#来与配制文件封装方法获取到的token值 来替换

1. **订单信息查询接口 ：  
   参数 token 订单编号（orderId）   
   注意：  
   设计接口测试代码的思路 ：订单编号 参数化 数据来源   
   1.数据来源：捞数据 的方式 随机从数据库里拿一条  
   2.拿来的数据能不能用是不是脏数据 是否合法？  
   3.怎么判断数据是否合法，能用 ？ 通过 多表（四张表）连接 查询 是否 成功 来验证**

## 自动化 参数化测试 三种方式： 1. 写在文件中 2.写在配制文件3.捞数据

## 1.写在文件中

## 2.写在配制文件

## [users] username = xiaoniu password = a123456

**（从 from public\_api 模块 引入 import replace\_data get\_ini\_data   
替换数据方法 replace\_data 把表格文件内需要替换对象键 #name# 与 配制文件 里 section [users] username键 = 替换 到 xiaoniu值 的 数据   
通过 从配制文件里拿数据 get\_ini\_data('users','username')   
把拿到的数据 参数化**

## 3.捞数据

**捞数据   
从数据库里随机拿一条数据   
one = self.db.find\_one (select \* from tb\_user order by rand() limit 1 ;)  
self.name,self.pwd = one[1] , one[2]  
把拿到的数据参数化**

## 

### 