**UI自动化框架使用**

**—基于python与ddt, selenium的实现**

**文档所有者**

|  |  |  |
| --- | --- | --- |
| **姓名** | **职位** | **部门** |
|  |  |  |
|  |  |  |

**批准人列表**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | **职位** | **批准时间** | **备注** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**参考文档**

|  |  |  |
| --- | --- | --- |
| **文档名称** | **作者** | **内容主题** |
|  |  |  |
|  |  |  |
|  |  |  |

**术语和名词解释**

|  |  |
| --- | --- |
| **术语** | **定义解释** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**文档修订摘要**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **版本日期** | **修改者** | **修改内容** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 简介

## 目标

该文档主要介绍基于该框架下，直接上手编码实施接口（RESTful API）自动化。

该文档是动态更新的，根据需求，会对某些章节做内容增加、删除、修改等。

该文档目标读者为项目经理、开发人员、测试人员等。

## 范围

该文档主要覆盖以下内容：

* 编码指导
* 测试运行

1. IDE执行脚本
2. 命令行执行脚本

* 编码姿势示例

## 假设

对本文档的用户做了以下假设：

* 熟悉Python语言
* 了解Python unittest框架
* 了解数据驱动测试及python数据驱动包ddt
* 有一定的接口自动化开发经验

# 编码指导

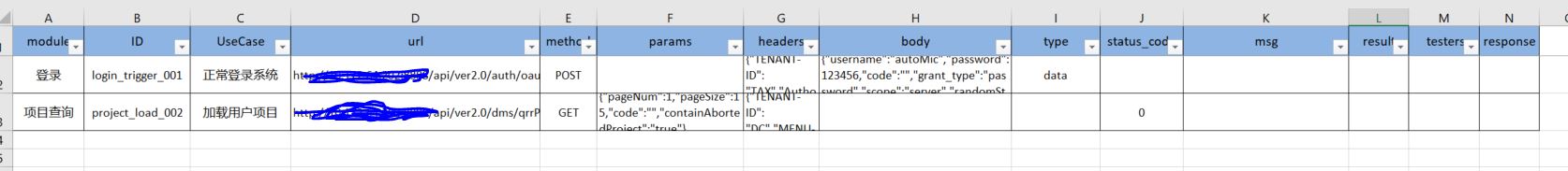
接口自动化实质上就是以工具代替手动请求接口。接口请求方式及参数类型就那么几种，因此其所需知识储备实际上还没UI自动化要求高。

在本框架中，发送请求的方法已经封装在lib/send\_requests.py文件中，因此逻辑控制脚本（testcase/\*API.py文件）中重点在组装接口参数、调用触发请求及根据返回结果判断是否请求成功。

另外，该框架中针对非流程依赖接口与流程性接口分别写了isolateAPI.py和flowAPI.py逻辑控制脚本。对于非流程依赖接口，鼓励将这类接口统一维护在一个testcase文件中；而流程性接口，鼓励一个流程一个testcase文件并参考flowAPI.py编写对应的逻辑控制脚本。

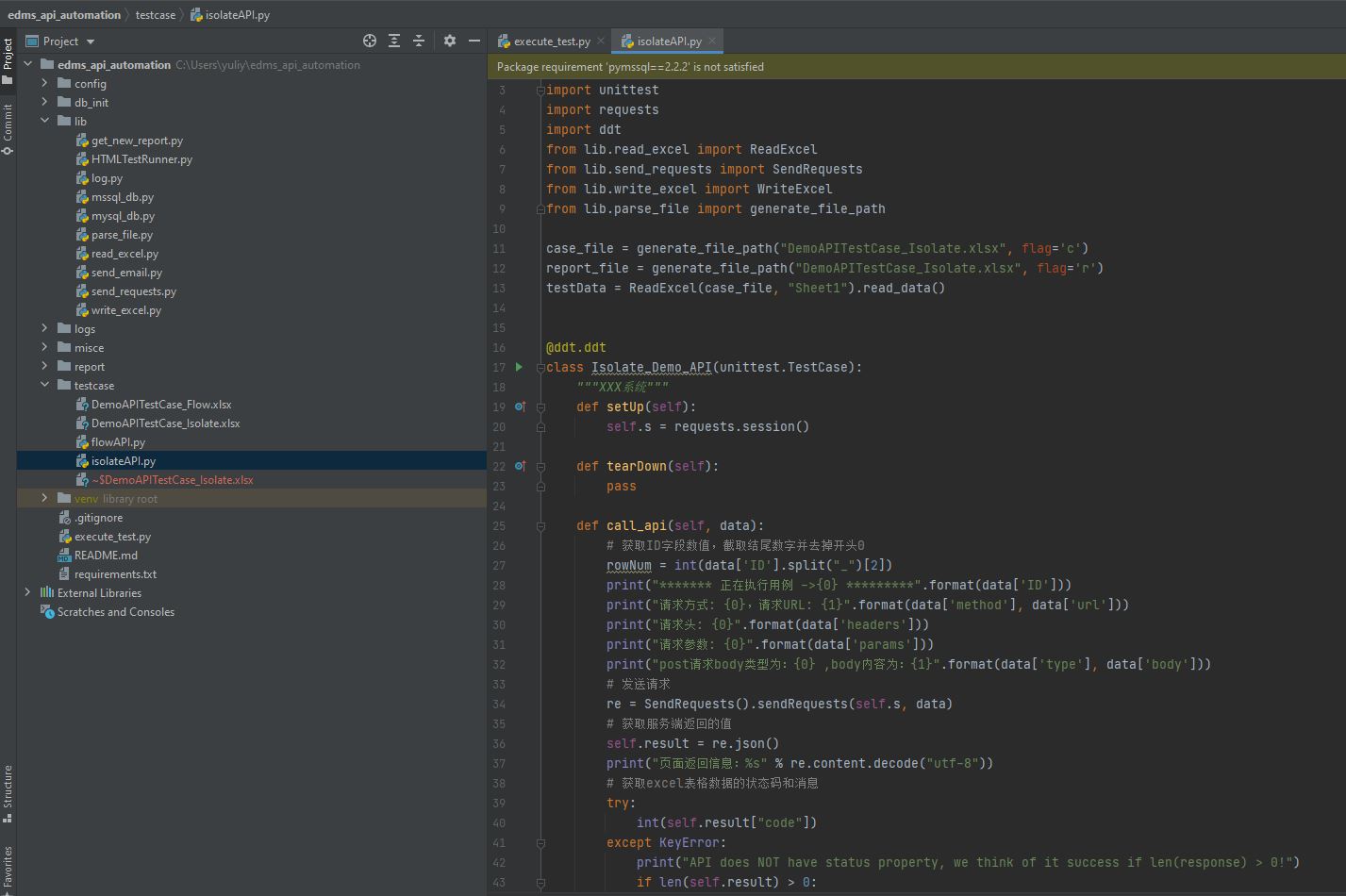
下面就获取该框架后，在不了解该框架实现细节的基础上，怎样快速地写出一个测试用例做简要说明：

## 将测试用例维护在testcase文件下\*.xlsx文件的sheet1中

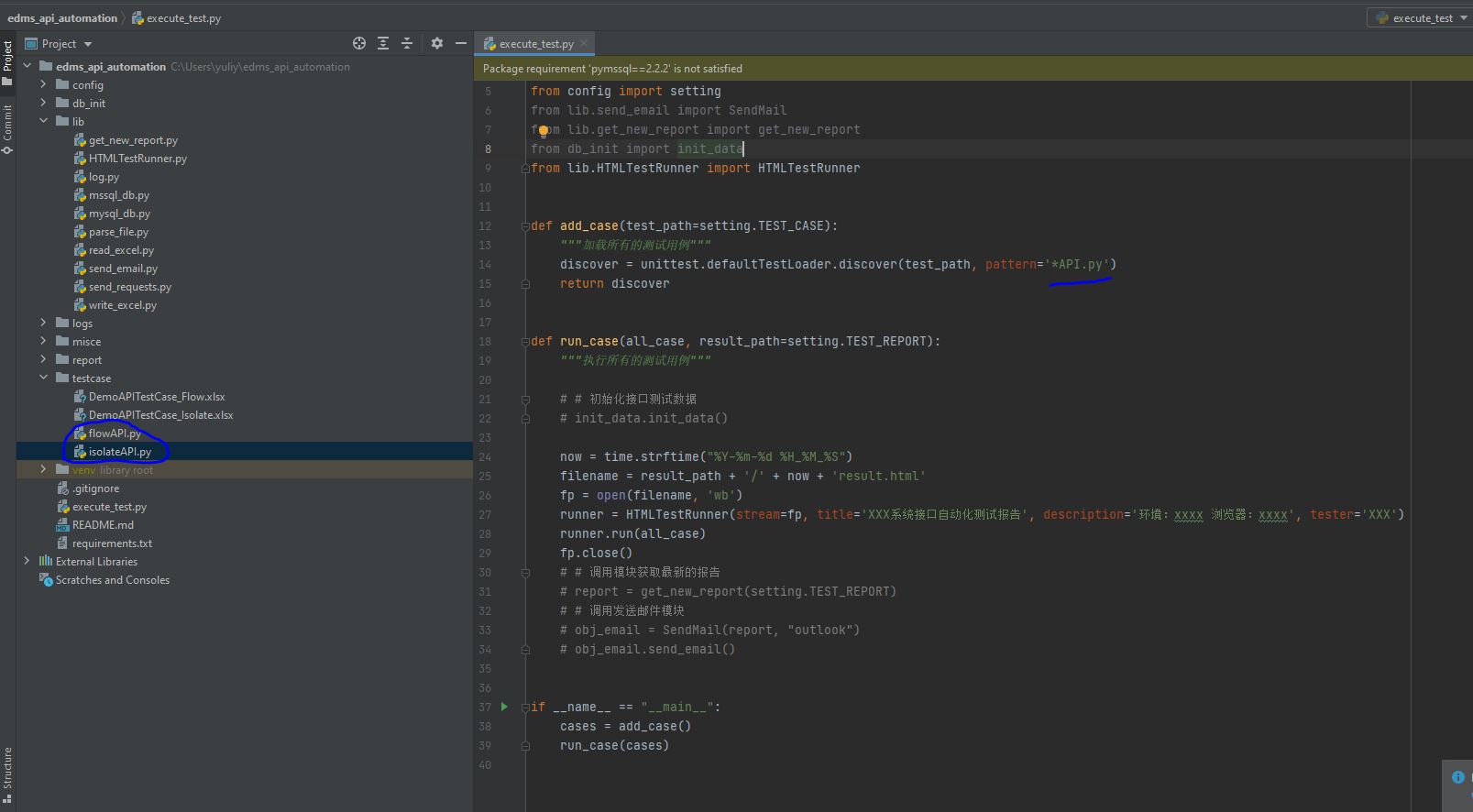


1. ID要以下划线分隔为3部分，且最后1部分必须由数字组成，eg. xxx\_xxx\_001
2. GET类型接口参数放置于params列，Type列保持为空
3. POST类型接口参数放置于body列，Type列填data
4. status\_code及msg列用于，维护期望结果（用于返回判断），使用者也可以在逻辑控制脚本中调整判断依据
5. response列用于存储接口返回结果，对应流程性接口测试可能会用

## 基于isolateAPI.py和flowAPI.py创建脚本



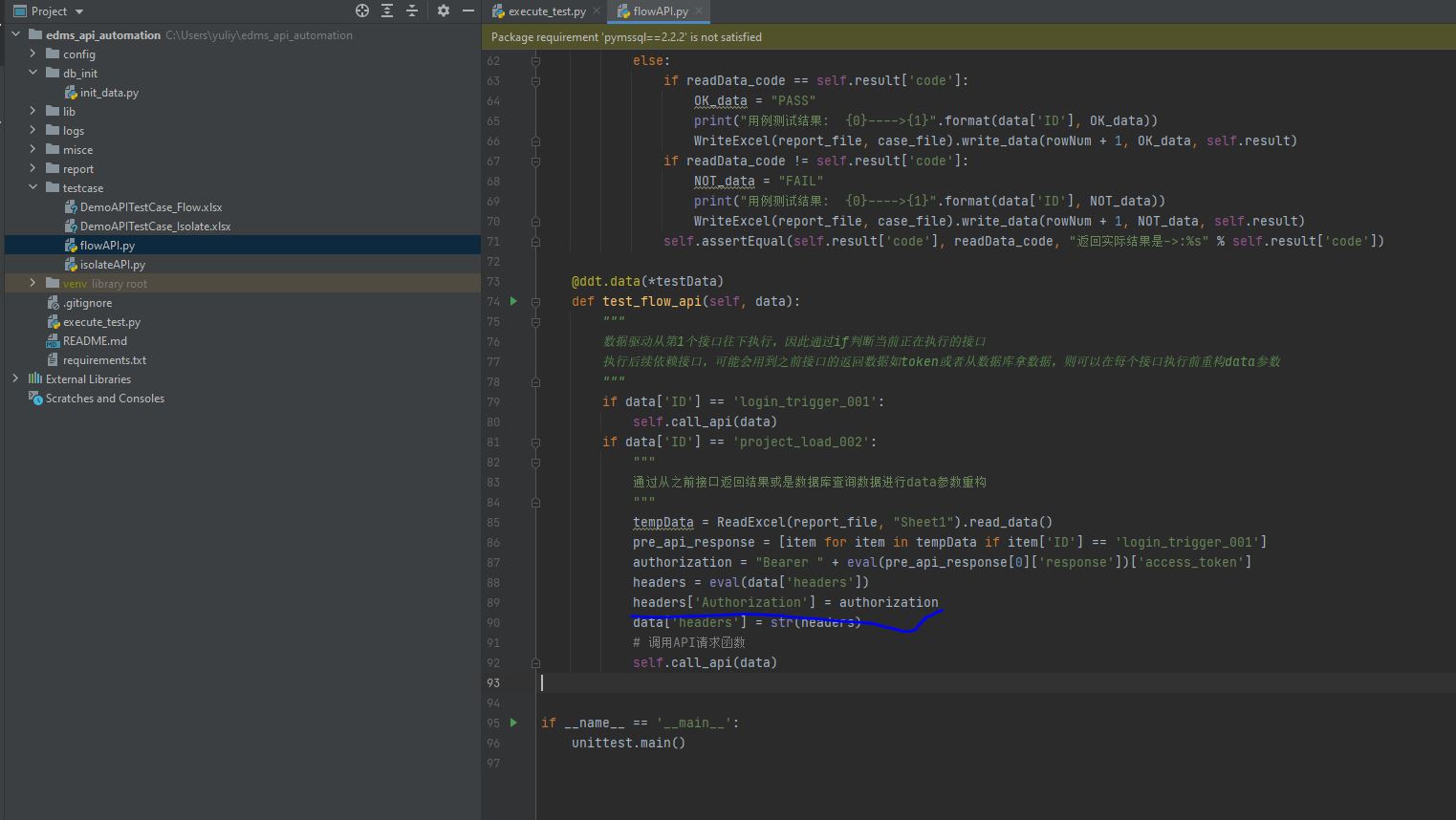
## 将脚本加入执行器，即加入执行队列



该框架中，会从testcase目录加载以API.PY结尾的逻辑控制脚本，因此想要执行的脚本只需放置到testcase目录并以API.PY结尾即可。

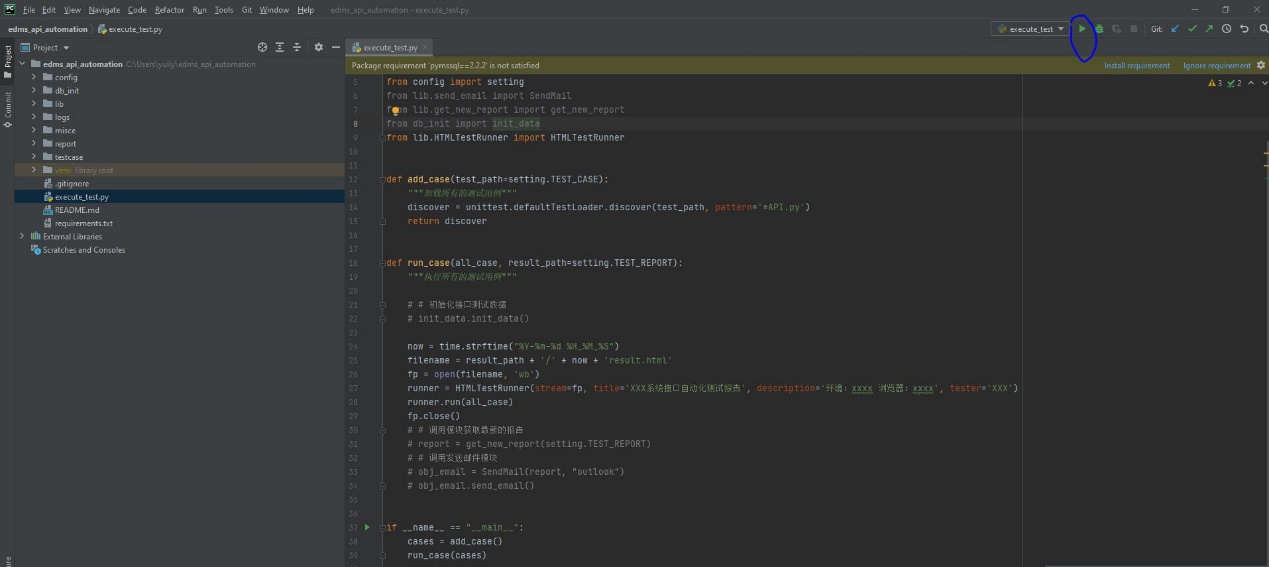
## 测试数据准备

接口测试的数据我们大致可以分为3类：支持接口的基础数据、独立接口测试数据、流程性接口测试数据。对于基础数据，我们只需将其以表为单位，维护进db\_init/init\_data.py模块的datas变量中，然后在脚本执行器（execute\_test.py）中进行初始化。对于独立接口测试数据，只需将其维护在excel表中的case中（如headers、params、body列）。而对于流程性接口，我们同样将参数模板维护进excel表中的case中，但我们在调用接口前需对参数进行重现组装：该过程设计在逻辑控制脚本中进行（eg.截图组织新的header数据）：



# 测试运行

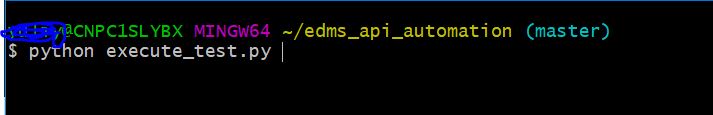
## Pycharm运行测试脚本



## 命令行运行脚本

#cd project\_home\_path

#python execute\_test.py



# 编码姿势示例