**接口自动化框架使用**

**—基于python、selenium与ddt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
* IDE执行脚本
* 命令行执行脚本
* 编码示例

## 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
* 了解接口自动化

# 编码指导

接口自动化实质上就是以自动化工具代替手工请求接口。接口请求方式及参数类型就那么几种，实际使用更少，因此其较容易上手。

在本框架中，发送请求的方法已经封装在lib/send\_requests.py文件中，另外接口的请求以及请求反馈的校验具有很高的共性，因此也已经封装在了testcase/call\_api.py文件的CallAPI类中以增强代码复用，所以逻辑控制脚本（testcase/\*API.py文件）中的重点及处理方式如下：

1. 接口请求顺序：excel组织用例，然后基于ddt实现数据驱动

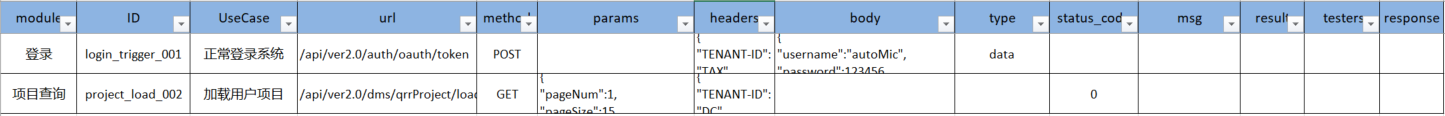
2. 接口参数组装：逻辑控制脚本中需自行编写的重点内容

3. 接口调用及返回判断：调用CallAPI类中定义的call\_api方法

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\_demo.py和.flowAPI\_demo.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/\*API.py文件中；而流程性接口，鼓励一个流程维护一个testcase/\*API.py文件并参考.flowAPI\_demo.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，且最后一部分必须由数字组成，eg. xxx\_xxx\_001

2.url只需维护域名后部分，以支持跨环境复用脚本

3.method为GET类型时，接口参数以json格式放置于params列，且Type列需保持为空

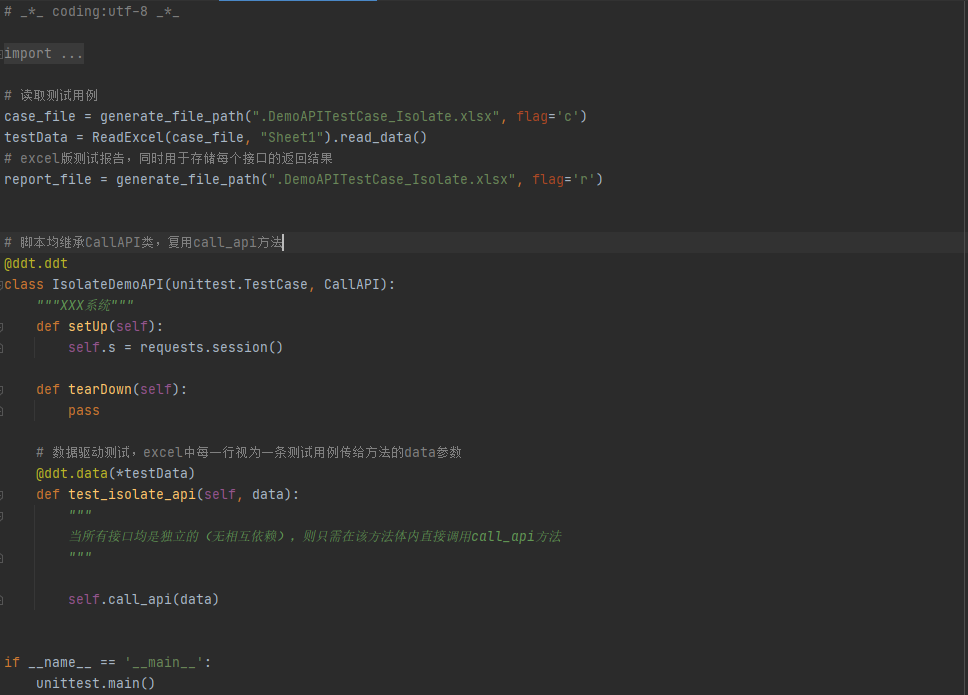
4.method为POST类型时，接口参数以json格式放置于body列，Type列填值（填data或json，取决于请求头的Content-Type，如果为application/json时，填json）

5.status\_code及msg列用于维护期望结果（用于返回判断），此外，该框架也支持调用call\_api方法时显式传递result\_check={}参数以自定义除status\_code和msg外的校验项（以python字典传递校验项，任意一项不满足视为校验失败）

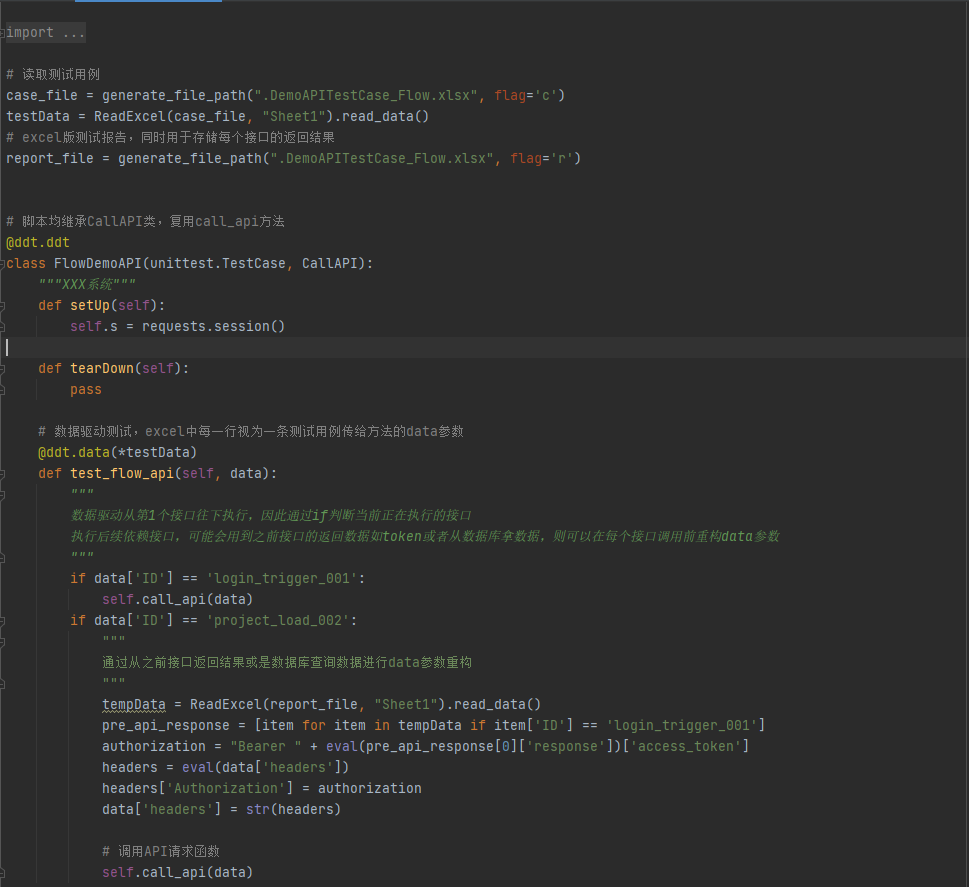
6.response列用于存储接口返回结果，以支持作为流程接口自动化测试时后续接口的参数源之一

## 基于.isolateAPI\_demo.py和.flowAPI\_demo.py编写脚本

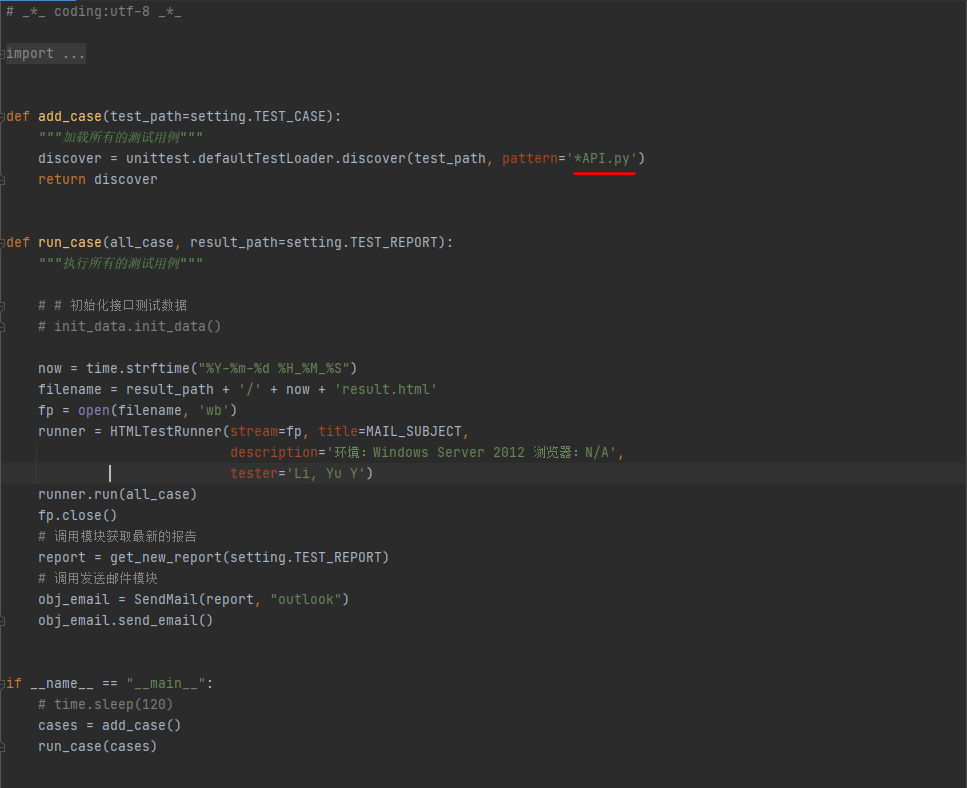
### 基于.isolateAPI\_demo.py



### 基于.flowAPI\_demo.py



## 将脚本加入执行器，即加入执行队列

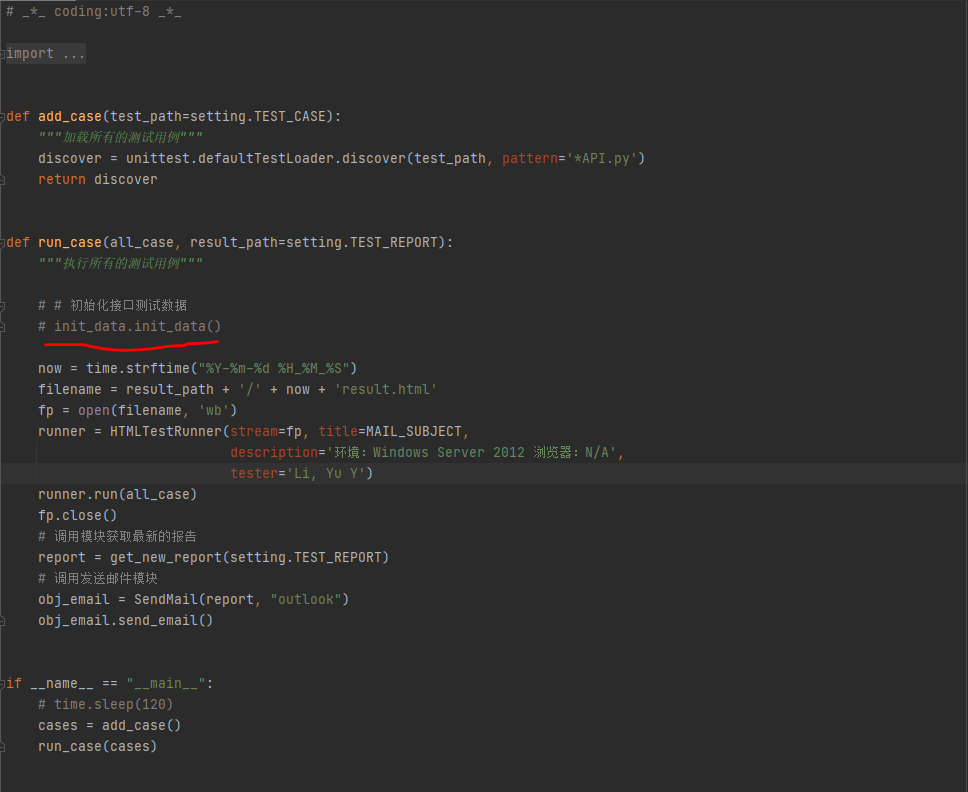


该框架中，会从testcase目录加载以API.py结尾的逻辑控制脚本，因此想要执行的脚本只需放置到testcase目录并以API.py结尾即可。

## 测试数据准备

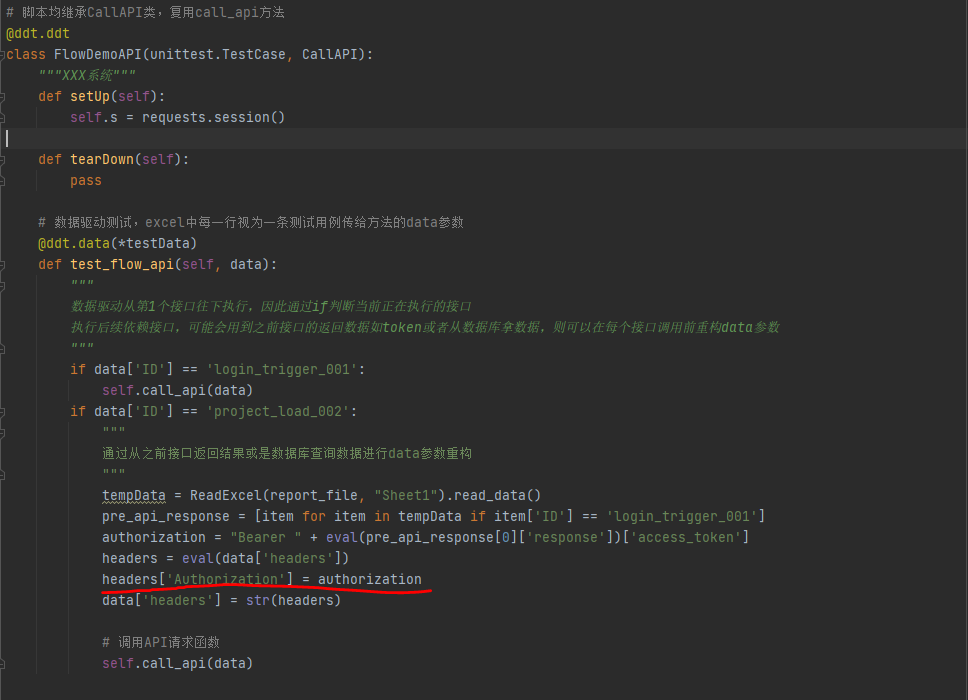
接口测试的数据我们大致可以分为3类：支持接口的基础数据、独立接口测试数据、流程性接口测试数据。

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:



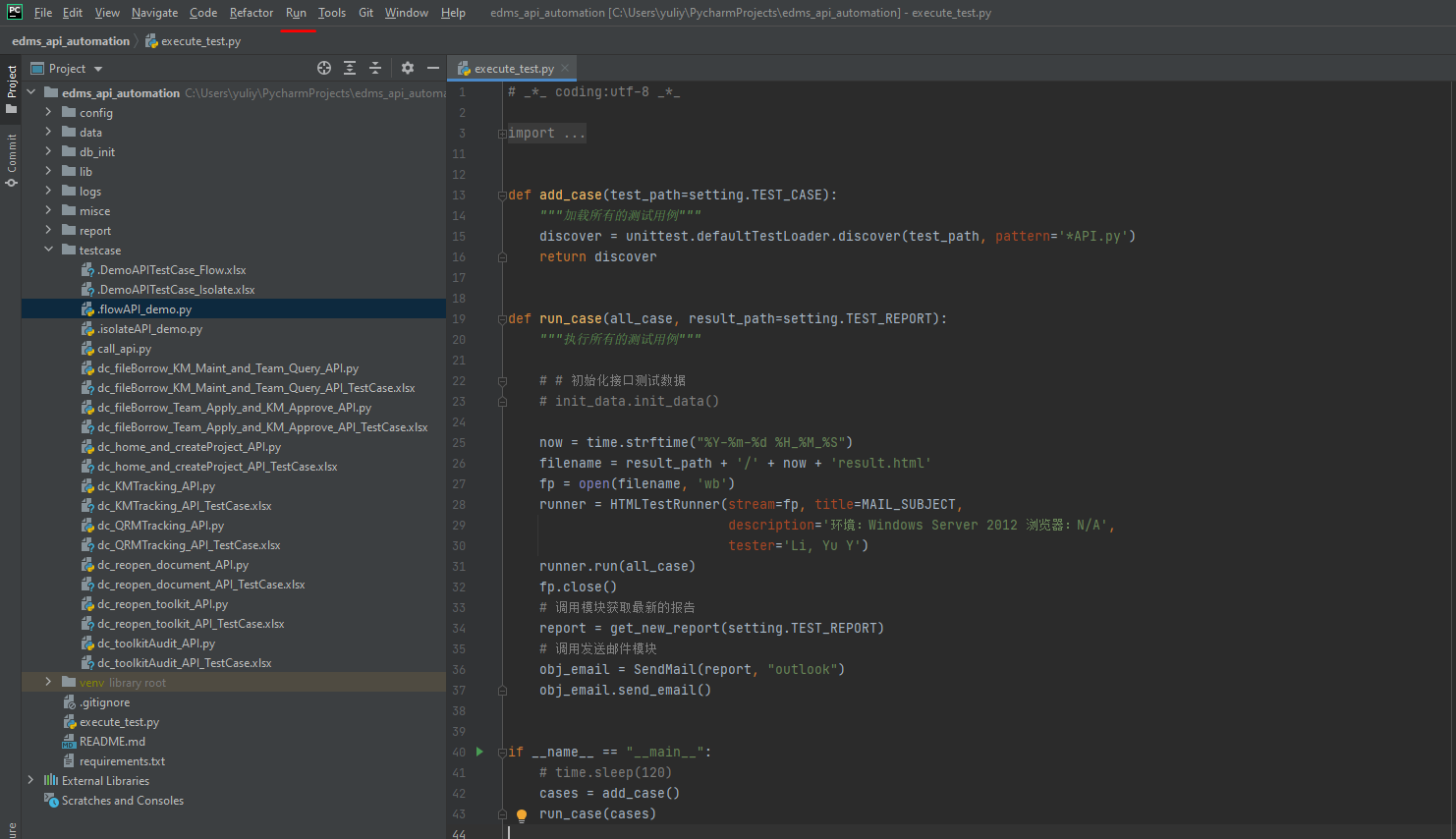
对于独立接口测试数据，只需将其维护在excel表中的用例中（如headers、params、body列）。

对于流程性接口，我们同样将参数模板维护进excel表中的用例中，但我们在调用接口前需对参数进行组装：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



# 编码示例