记录这套书/视频/文章的学习过程（知识点、error思考等），方便后续结合code总结回归

\*\自动化测试\6Python接口测试框架实战与自动化进阶\

# Django\_imooc工程

## 3-1 开发接口环境搭建

**创建项目方式一：pycharm**

**Create newproject , Django , application name =web**

settings.py #系统文件，数据库 工程等

urls.py #？

方式1：python manage.py runserver 127.0.0.1:8000

方式2：pycharm中manage.py配置runserver 127.0.0.1:8000

再去浏览器访问

## 3-2 django之接口工作原理

Django的应用程序 叫app，可以有很多个。

方式1：pycharm专业版中新建时，在more-application name中创建。

方式2：python manage.py startapp api

app的名字=api

一、在settings中添加我们的应用程序（app），不然无法解析

INSTALLED\_APPS

'web'

**二、Django\_imooc/ Django\_imooc/urls.py**

地址映射login方法

from web.views import Login  
urlpatterns = [  
url(r'^admin/', admin.site.urls),  
url(r'^login',Login)  
]

三、login方法写在views中

**Django\_imooc/web/views.py**

from django.http.response import HttpResponse

# Create your views here.  
def Login(request):  
 return HttpResponse(‘this is first test’)

**整体原理：接收到地址，转发，返回**

## 3-3 django之post接口开发

**url 下加：路由中添加地址和对应调用的函数**

from web.views import Login  
  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^login', Login)  
]

**views中加：添加对应的方法**

from django.http.response import HttpResponse

from django.shortcuts import render\_to\_response

# Create your views here.  
def Login(request):  
if request.method == "POST":  
 username =request.POST.get('username')  
return HttpResponse(username)  
else:  
return render\_to\_response('login.html')

**Django\_imooc/ templates下添加login.html**

添加表单form

<form action="/login/" method="POST">  
<h1>用户名：<input name="username"></h1>  
<h1>密码：<input name="password"></h1>  
<input type="submit" value="登录">  
</form>

login.html调用login方法（login方法在views.py中实现），method=post

添加TEMPLATES路径

'DIRS': [os.path.join(BASE\_DIR, 'templates')]

**Django\_imooc/ Django\_imooc/settings.py**

注释掉安全拦截

MIDDLEWARE = [  
# 'django.middleware.csrf.CsrfViewMiddleware',

访问<http://127.0.0.1:8000/login/>

# DjangoInterface工程

## 第4章 requests库的相关使用

## 4-2 requests简单使用-post

requests 是一个http library库

request.post的参数默认是字典

**见base\demo4\_2.py**

res.text的类型是 <class 'str'>

res.json()的类型是 <class 'dict'>

## 4-3 重构发送post请求

封装了 post请求（写成方法）

**demo.py**

## 4-4 重构get请求+格式化响应数据

把post换成了get ，

1、get时省略data参数，url后直接加？参数（一般不用）

2、get时加data参数，url后的？参数省略

def send\_get(url, data):  
 res = requests.get(url=url, data=data).json()  
 return json.dumps(res, indent=2, sort\_keys=True)

用request库，构造两个简单的接口

**demo.py**

## 4-5 使用类封装接口测试脚本

添加run\_main方法用于判断是post还是get，优化入参data=none，优化res=none，return

添加RunMain：类和if name =main，优化初始化：def init

**demo.py (4-5 使用类封装接口测试脚本).txt**

\*\000\00CodeHub0203\2PythonCode\DjangoInterface\base\

## 第5章

## 5-2 unittest和request重构封装

创建包 base 等

**test\_method.py中导入demo.py和加载执行**

## 5-6 unittest之常见面试解答及知识回顾

**初中级面试问题**

### 使用python开发测试框架

语音：python

接口开发：requests包

case管理：unittest

测试报告：htmltestrunner.py，放lib下

测试数据：excel，（mysql数据库）

持续集成：jenkins

### 如何管理case：

case有依赖：

unittest.skip跳过

globals()全局变量来使用

要用的存在数据库里

case写在哪里：  
写在excel

数据库

### 简述case的执行：

case字母的先后顺序执行

**do 练习**

## 第6章 mock服务入门到实战

mock服务端还没开发完成。

模拟接口的返回，类似于fiddler的autoResponse

## 7-1 如何设计一个接口自动化测试框架

需要思考：

接口地址

请求数据

接口类型，get post

预期结果，

header，保存cookies

数据依赖，

## 7-7 封装获取接口数据

#通过获取关键字拿到data数据  
def get\_data\_for\_json(self,row):  
opera\_json = OperationJson()  
request\_data = opera\_json.get\_data(self.get\_request\_data(row))  
return request\_data

为什么是操作json的这个方法？

## 7-9封装获取接口数据

问题：

from base.runmethod import RunMethod

ModuleNotFoundError: No module named 'base'

原因：

python找包优先会在sys.path找（安装python的地方）

解决：

加入工程路径，双斜杠

sys.path.append("E:\\111\\Py\\Py\_code\\6Python接口测试框架实战与自动化进阶\\DjangoInterface")