下载并使用API测试工具: POSTMAN

地址: https://www.getpostman.com/downloads/

1.需要大家使用“前后端分离技术”重构的项目：

“净美仕门户PC端项目”部署地址：www.codeboy.com/mfresh

“净美仕门户PC端项目”源码地址：www.codeboy.com/mfresh/mfresh.zip

“净美仕门户APP端项目”部署地址：www.codeboy.com/mfresh-m

“净美仕门户APP端项目”源码地址：www.codeboy.com/mfresh-m/mfresh-m.zip

“iWeb在线课堂PC端项目”部署地址：www.codeboy.com/iweb

“iWeb在线课堂PC端项目”源码地址：www.codeboy.com/iweb/iweb.zip

“iWeb在线课堂APP端项目”部署地址：www.codeboy.com/iweb-m

“iWeb在线课堂APP端项目”源码地址：www.codeboy.com/iweb-m/iweb-m.zip

“优生活租房PC端项目”部署地址：www.codeboy.com/ulift

“优生活租房PC端项目”源码地址：www.codeboy.com/ulift/ulift.zip

“优生活租房APP端项目”部署地址：www.codeboy.com/ulift-m

“优生活租房APP端项目”源码地址：www.codeboy.com/ulift-m/ulift-m.zip

2.“净美仕官网”PC端项目重构实现步骤

**旧有系统：**www.codeboy.com/mfresh/mfresh.zip 静态页面+PHP混合项目

**系统重构：**前后端分离：前端子系统(Vue.js)、后端子系统(Python/Django)

**<<<<<<<<<<<后端子系统实现步骤：>>>>>>>>>>>>>**

一、创建一个空白Django项目，并保证可以运行

1.1 下载django模块

pip install django

1.2 运行django工具，创建空白项目

django-admin startproject mfresh\_api

1.3 进入空白项目，运行起来

cd mfresh\_api

python manage.py runserver

可以使用浏览器访问该项目 http://127.0.0.1:8000

二、创建子App以及路由系统

2.1 创建必需的子App

python manage.py startapp user

python manage.py startapp news

python manage.py startapp product

python manage.py startapp cart

2.2 在主App中注册所有的子App

#编辑主App/settings.py/INSTALLED\_APPS

#添加上述四个子App

INSTALLED\_APPS = [

'user',

'news',

'product',

'cart',

.....系统原有的.....

]

2.3 在每个子App下创建路由方法(Django称为Views)

#创建user App下的路由方法——编辑user/views.py，创建路由方法：

def userLogin( req ):

def userRegister( req ):

def userCheckUname( req ):

def userCheckPhone( req ):

#类似的，创建news App下的路由方法——编辑news/views.py，创建方法：

#类似的，创建product App下的路由方法——编辑product/views.py，创建方法：

#类似的，创建cart App下的路由方法——编辑cart/views.py，创建方法：

2.4 为每个路由方法分配路由地址

#编辑主App/urls.py，添加更多的路由设置

from user import views as uv

urlpatterns = [

path('user/login', uv.userLogin),

path('user/register', uv.userRegister),

path('user/check/uname', uv.userCheckUname),

path('user/check/phone', uv.userCheckPhone),

]

#类似的，再添加news/product/cart三个App下的路由设置

三、使用ORM实现每个路由方法中的数据库操作

3.1 启动MySQL数据库，在其中创建好mfresh库及所有表

3.2 下载并安装MySQL数据库连接驱动：pymysql

pip install pymysql #如果报错，使用国内的仓库

3.3 在主App中注册使用pymysql提供的数据库驱动

#编辑主App/\_\_init\_\_.py文件

import pymysql

pymysql.install\_as\_MySQLdb( )

3.4 在主App中指定MySQL服务器的连接参数

#编辑主App/settings.py文件：DATABASES

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'mfresh',

'HOST': '127.0.0.1',

'PORT': 3306,

'USER': 'root',

'PASSWORD': '',

}

}

3.5 使用ORM反向生成工具，自动根据数据库表结构创建对应的Object

#检查数据库中的表结构，生成每个表对应的Object对象

python manage.py inspectdb > user/models.py

#生成的class输出到控指定的Model文件

3.6 使用Django QuerySet提供的API操作模型对象的增删改查，实现底层数据库的CRUD：

查询所有：MfUser.objects.**values('uname',...)**

查询一个：MfUser.objects.**filter(uid=3)**.values('uname',...)

添加新的：MfUser.objects.**create(uname='..', upwd='..')**

删除数据：MfUser.objects.filter(uid=4)**.delete( )**

更新数据：MfUser.objects.filter(uid=4)**.update(uname='..', upwd='..')**

**<<<<<<<<<<<前端子系统实现步骤：>>>>>>>>>>>>>**

一、使用Vue-CLI脚手架创建空白项目并启动

1.1 下载Vue-CLI全局脚手架工具

npm i -g @vue/cli

1.2 运行脚手架工具，创建空白项目

vue create mfresh\_fe

1.3 启动空白项目

cd mfresh\_fe

npm i axios

npm run serve

测试访问：http://127.0.0.1:8080

二、创建路由组件，并配置路由词典

2.1 创建路由组件

2.2 为每个路由组件分配路由地址

三、详细编写每个路由组件及其子组件，获取数据，并实现数据绑定

|  |
| --- |
| 注意：如果使用ORM生成了Class文件，其中的代码可能有不干净的空白字符，导致导入此文件时提示如下错误：  *ValueError： Source code string cannot contain null type*  *或者*  *文件名、目录名或卷标语法不正确*  解决方案：先把文件内容粘贴到一个空白txt文件中；删除此.py文件；重新创建，把.txt中内容在复制粘贴回来 |

3.使用Django的ORM技术实现数据库的CRUD

**①向数据库插入一条新的记录**

目标SQL：INSERT INTO mf\_user VALUES(NULL, ?, ?, ?)

需要编写：u = MfUser.objects.**create**(uname=?, upwd=?, phone=?)

**②从数据库中删除一条记录**

目标SQL：DELETE FROM mf\_user WHERE uid=?

需要编写：t = MfUser.object.**filter**(uid=?).**delete**( )

**③修改数据库中的一条记录**

目标SQL：UPDATE mf\_user SET uname=?, upwd=? WHERE uid=?

需要编写：n = MfUser.objects.**filter**(uid=?).**update**(uname=?, upwd=?)

**④查询数据库中的一行记录**

目标SQL：SELECT uname,upwd,phone FROM mf\_user WHERE uid=?

需要编写：MfUser.objects.**filter**(uid=?).**values**('uname', 'upwd', 'phone')

如果查询到了，返回值是一个[{...}]；否则返回值是一个[ ]

**⑤查询数据库中的多行记录**

目标SQL：SELECT uname,upwd,phone FROM mf\_user

需要编写：MfUser.objects.**values**('uname', 'upwd', 'phone')

如果查询到了，返回值是一个[{...},{...},...]；否则返回值是一个[ ]

4.高阶面试题：Web安全问题

数据库安全：.....

服务器端安全：.....

传输安全：.....

客户端安全：

**①XSS攻击：**

**②CSRF攻击：**跨站请求伪造攻击

注意：默认情况下，Django站点采用了防止“CSRF攻击”中间件，会阻止所有的跨域访问，必须停止该中间件！修改主APP/settings.py：50

注释掉'django.middleware.csrf.CsrfViewMiddleware'中间件

课后任务：

(1)继续完成“净美仕API项目”商品和购物车相关6个API的编写