# 项目开发流程

蓝色是步骤 黄色是注释

## 项目框架搭建

1、终端下进入虚拟环境，创建项目：

django-admin startproject dailyfresh（项目名称）

1. 进入项目目录，创建各模块对应的应用：

python manage.py startapp user（应用名称）

反向解析：怎样找到视图函数（在哪个应用下，是哪个视图函数），而不是怎样找到模板文件，模板文件的查找路径靠配置好的templates路径

python manage.py startapp goods

python manage.py startapp cart

python manage.py startapp order

1. 打开pycharm，进行代码编写之前的项目基本配置：

<1>通常在应用较多的情况下，应该在项目目录下新建一个python package——apps（python包，python代码应存于包中，以便导入）管理应用。

<2>在项目settings.py中进行应用注册：

import sys  
 sys.path.insert(0, os.path.join(BASE\_DIR, 'apps'))

在项目目录下新建的文件夹或者包,都要和项目路径BASE\_DIR拼接路径

‘user’,’goods’,’cart’,’order’,

<3>在项目urls.py中进行url配置：

项目url匹配到 ^user/ 之后，将url的 user/ 截断，不会进行接下来的应用url匹配

utl(r’^user/’, include(‘user.urls’, namespace=’user’)),

utl(r’^cart/’, include(‘cart.urls’, namespace=’cart’)),

utl(r’^order/’, include(‘order.urls’, namespace=’order’)),

utl(r’^’, include(‘goods.urls’, namespace=’goods’)),

<4>在项目目录下新建模板文件夹templates并在项目settings.py中配置路径：

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

<5>在项目目录下新建静态资源文件夹static并在项目settings.py中配置路径:

STATICFILES\_DIRS = [os.path.join(BASE, ‘static’)]

在项目目录下新建的文件夹或者包,都要和项目路径BASE\_DIR拼接路径

<6>在项目settings.py中进行数据库配置：

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

'NAME': 'dailyfresh',

'USER': 'root',

'PASSWORD': 'lina0703',

'HOST': 'localhost',

'PORT': 3306

还要在项目的\_\_init\_\_.py中加入下面代码：

import pymysql

pymysql.install\_as\_MySQLdb()

<7>进行语言和时区的配置：

‘zh-hans’

‘Asia/Shanghai’

## 模型层模型类代码编写

1. 在各应用下的models.py中编写模型类以实现与数据表的映射：

from django.db import models

class Address(models.Model):

'''地址模型类'''

user = models.ForeignKey('User', verbose\_name='所属账户')

receiver = models.CharField(max\_length=20, verbose\_name='收件人')

addr = models.CharField(max\_length=256, verbose\_name='收件地址')

zip\_code = models.CharField(max\_length=6, null=True, verbose\_name='邮政编码')

phone = models.CharField(max\_length=11, verbose\_name='联系电话')

is\_default = models.BooleanField(default=False, verbose\_name='是否默认')

**class Meta:**

**db\_table = 'df\_address'**

**verbose\_name = '地址'**

**verbose\_name\_plural = verbose\_name**

1. 模型类编写完成后，生成和执行迁移文件：

python manage.py makemigrations

Python manage.py migrate

## 功能实现代码编写

1、模板

<1>在templates文件夹下编写html文件（可以分别新建文件夹对应各个模块），在static文件夹下编写静态资源文件如css、js、img

<2>在html文件中导入静态资源路径：

{% load staticfiles %}

注意：放在<html>标签后面即可，不要放在任何标签中

<3>html文件中任何有静态资源的地方，路径同一格式为：

{% static 'css/reset.css' %}

‘static文件夹 下的 css文件夹 下的 css文件’

STATIC\_URL = '/static/'

<4>html文件中向视图发起请求时，url格式统一为：

{% url ‘namespace:name’ 位置参数或关键字参数，空格隔开 %}

模板中的反向解析，其实这里的url用的就是下面的reverse

**ajax请求只用于在页面不重新加载的情况下，进行局部的数据更新，其行为在后台进行，不能用于有 页面跳转 的请求**

2、视图

<1>复制项目urls.py到各个应用中，**创建类视图**，在应用urls.py中配置url，格式如下:

from user.views import RegisterView

url(r'^register$', RegisterView.as\_view(), name='register'),

<2>视图中的反向解析：

from django.core.urlresolvers import reverse

元组

return redirect(reverse(‘namespace:name’, args=(位置参数，逗号隔开))

return redirect(reverse(‘namespace:name’, kwargs={key:value}))

字典