django入门

Ringo Yungpo Kao

December 26, 2017

# django入门

## django简介

Django是一个开放源代码的Web应用框架，由Python写成。采用了MVC的软件设计模式，即模型M，视图V和控制器C。它最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻内容为主的网站的，即是CMS（内容管理系统）软件。并于2005年7月在BSD许可证下发布。这套框架是以比利时的吉普赛爵士吉他手Django Reinhardt来命名的。

Django是一个基于MVC构造的框架。但是在Django中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV模式。它们各自的职责如下：

### 模型（Model），即数据存取层

处理与数据相关的所有事务： 如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。

### 模板(Template)，即表现层

处理与表现相关的决定： 如何在页面或其他类型文档中进行显示。

### 视图（View），即业务逻辑层

存取模型及调取恰当模板的相关逻辑。模型与模板之间的桥梁。

### 基本结构：

│ db.sqlite3 ----------sqlie3数据库  
│ manage.py   
│   
├─logres  
│ │ admin.py 后台，可以用很少量的代码就拥有一个强大的后台。  
│ │ apps.py  
│ │ models.py 与数据库操作相关，存入或读取数据时用到这个  
│ │ tests.py  
│ │ urls.py  
│ │ views.py   
│ │ 处理用户发出的请求，从urls.py中对应过来, 通过渲染templates中的网页可以将显示  
│ │ 内容比如登陆后的用户名，用户请求的数据，输出到网页。  
│ │ \_\_init\_\_.py  
│ │   
│ ├─migrations  
│ │ 0001\_initial.py  
│ │ \_\_init\_\_.py  
│   
│   
│   
│   
├─Mushishi  
│ │ settings.py Django 的设置，配置文件，比如 DEBUG 的开关，静态文件的位置等  
│ │ urls.py urls.py  
│ │ 网址入口，关联到对应的views.py中的一个函数（或者generic类），  
│ │ 访问网址就对应一个函数。  
│ │ wsgi.py wsgi有多重一种uwsgi和wsgi，你用那种wsgi来运行Django，  
 一般不用改只有你用到的时候在改  
│ │ \_\_init\_\_.py  
│   
│   
├─static  
└─templates templates中的Html模板，  
 index.html  
 login.html  
 regist.html

## 安装django

1. window 使用pycharm安装 过程略
2. window上使用pip安装 pip install django
3. linux下使用pip安装 yum install python-pip pip install django=1.9.5
4. 检查是否安装成功

>>> import django  
>>> django.VERSION

## django基本命令

1. 创建django命令 django-admin.py startproject project-name（你工程的名字）
2. 创建django的app python manage.py startapp app-name（你app的名字） 或 django-admin.py startapp app-name（你app的名字）
3. 同步数据库 python manage.py syncdb 注意：Django 1.7.1及以上的版本需要用以下命令 python manage.py makemigrations python manage.py migrate
4. 调试模式 python manage.py runserver 8001 **监听所有可用 ip （电脑可能有一个或多个内网ip，一个或多个外网ip，即有多个ip地址）** python manage.py runserver 0.0.0.0:8000
5. 清除数据库 python manage.py flush
6. 创建超级管理员 python manage.py createsuperuser 按照提示就ok
7. 修改管理员密码 python manage.py changepassword username（你当时设定的用户名）
8. 导入和导出数据 python manage.py dumpdata appname > appname.json python manage.py loaddata appname.json
9. 进入数据库 python manage.py dbshell
10. 更多命令

## django的hello world

### 创建一个project工程和app

使用pycharm新建一个django的project工程和app

### Project和App概念

Project是一个大的工程，

下面有很多功能：（一个Project有多个App，其实他就是对你大的工程的一个分类）

例如一个运维平台是一个工程，那么他的app就是CMDB，监控系统，OA系统，

### 生成数据库 创建超级管理员用户

1. 同步数据库 python manage.py makemigrations python manage.py migrate 本人使用的是django1.9.5版本
2. 创建超级管理员 python manage.py createsuperuser
3. 运行django python manage.py runserver 8000

### 登录页面：

浏览器访问：  
http://127.0.0.1:8000/admin/

### 路由

首先在helloword文件夹下（不是app目录，千万别写错）的urls.py填写路由规则

from django.conf.urls import url  
from django.contrib import admin  
#导入app下的view函数  
from helloapp import views  
urlpatterns = [  
url(r'^admin/', admin.site.urls),  
#当用户访问http://127.0.0.1：端口号的时候之间交给helloapp下面的views里的index函数来处理  
url(r'^$', views.index),  
]

### views函数

在helloapp（app）下面的views里写一个index函数

#Django 在返回的时候需要一层封装,需要导入HttpResponse  
from django.shortcuts import render,HttpResponse  
   
# Create your views here.  
def index(request):  
 #使用HttpRespons 封装返回信息  
 return HttpResponse('<h1>hello world!!!</h1>')

**django中的路由系统和其他语言的框架有所不同，在django中每一个请求的url都要有一条路由映射，这样才能将请求交给对一个的view中的函数去处理。其他大部分的Web框架则是对一类的url请求做一条路由映射，从而是路由系统变得简洁。**

如果要返回html页面

1. 在templates里创建index.html页面 内容：

<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="UTF-8">  
<title>Title</title>  
</head>  
<body>  
<h1>Hello world!!</h1>  
</body>  
</html>

1. 修改helloapp里的views.py的index函数

from django.shortcuts import render,HttpResponse  
# Create your views here.  
def index(request):  
# return HttpResponse('<h1>hello world!!!</h1>')  
return render(request,'index.html')  
#找到index.html  
#读取index.html返回给用户

当以上步骤都完成之后 python manage.py runserver 8000

使用浏览器访问:

http://127.0.0.1:8000(你用哪个端口就写哪个)