July19 Django

1 URL使用

urls.py默认是在主目录中，主路由配置文件，包含所有的地址映射。

每一个请求到达之后，都会由urls.py中urlpatterns列表中的url()进行匹配。当url()匹配上之后，可能就会将请求转交给其他的视图(Views)或其他的urls.py处理。

2 测试

在主目录中，创建views.py

作用：

包含所有定义好的视图（处理程序）

3 url函数详解

3.1 url函数的语法

url(regex,views,kwargs=None,name=None)

|  |  |
| --- | --- |
| regex | 正则表达式模式，匹配请求的url，字符串类型 |
| views | 当前url对应的处理的视图函数，此处写函数名即可 |
| kwargs | 字典，用来向views传参的，如果没有参数可以省略 |
| name | 字符串，给url()起别名，方便在模板中使用 |

3.2 通过url向views传参

**3.2.1使用正则表达式子组传参**

使用子组传参-()，一个子组是一个参数，如果想传递多个参数的话，则使用多个子组，多个参数之间使用/分隔开

urlpatterns=[

#访问路径是run/，交给run\_views

url(r’^run/$’,run\_views),

#当访问路径是run/两位数字，交给run\_arg1\_views

url(r’^run/(\d{2})/$’,run\_arg1\_views),

#当访问路径是run/两位数字/四位数字，交给run\_arg2\_views

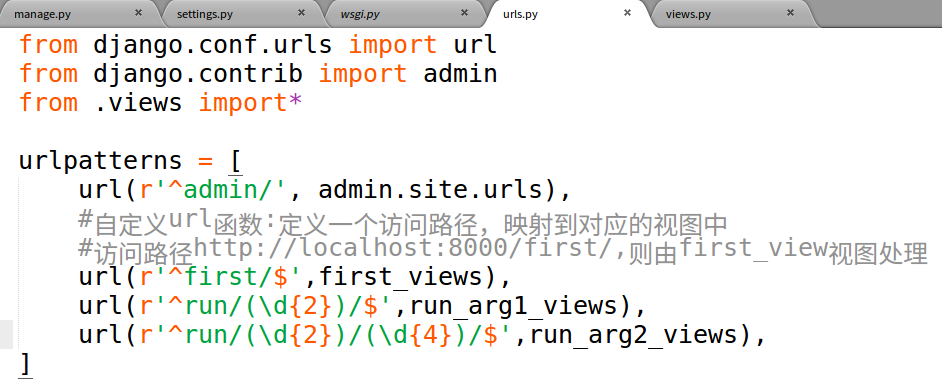
url(r’^run/(\d{2})/(\d{4})/$’,run\_arg2\_views),

]

注意：

1.在url()中，一个子组表示一个参数

2.在views.py中，对应的处理函数要根据url()中子组的个数，相应的定义参数（形参）。定义的参数要位于request之后

**3.2.2使用url()第三个参数，字典传参**

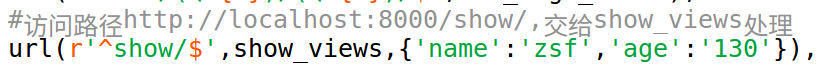
urlpatterns = [

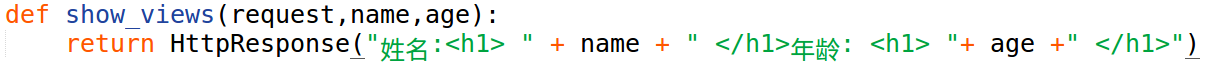
url(r’^show/$,show\_views,{‘name’:’zsf’,’age’:’25’}’),

]

def show\_views(request,name,age):

pass





4 Django中的应用

应用就是网站中的一块独立的程序

在Django中，主目录一般不处理用户的具体请求，主要做的是项目的初始化以及请求的分发。而具体的请求由各个应用去处理

4.1 创建应用

1.通过指令创建应用

./manage.py startapp 应用名称

示例：

./manage.py startapp news

2.在settings.py中注册应用

在INSTALLED\_APPS中追加应用名称

INSTALLED\_APPS = [

‘django.contrib.admin’,

…

‘news’

]

3.应用的结构组成

1）migrations目录

存放数据库中间文件的目录（目录）

2）\_\_init\_\_.py

应用的初始化文件

3）admin.py

应用的后台管理配置文件

4）apps.py

应用的属性配置文件

5）models.py

模型配置文件

6）tests.py

测试模块

7）views.py

视图处理文件

练习：

1.创建一个index的应用，并注册

2.创建一个sport的应用，并注册

3.创建一个music的应用，并注册

<http://localhost:8000/news/xxx/xxx> 交给news应用处理

<http://localhost:8000/music/xxx> 交给music应用处理

…

<http://localhost:8000/news/index> 交给news应用中的index\_views视图处理



练习:

1.访问<http://localhost:8000/news/index>

交给news应用中的urls处理（找index\_views视图）

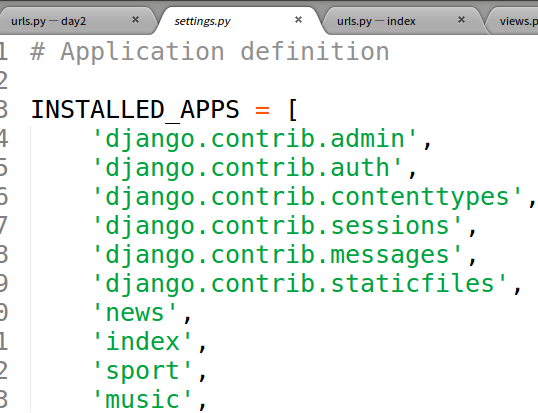
2.访问<http://localhost:8000/music/index>

交给music应用中的urls处理（找index\_views视图）

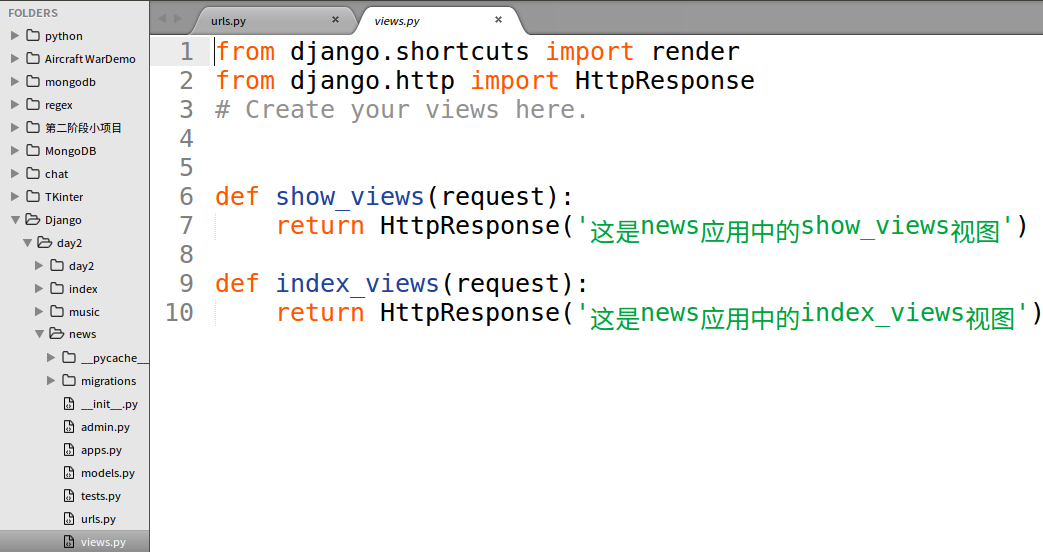
3.访问<http://localhost:8000/sport/index>

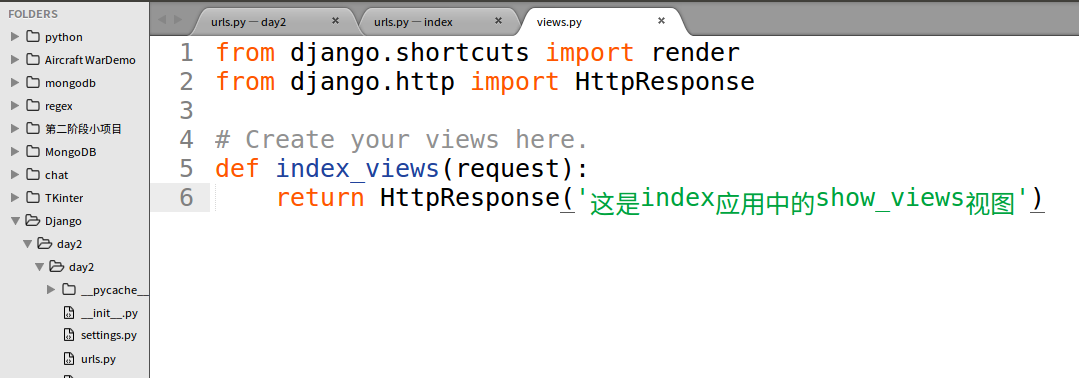
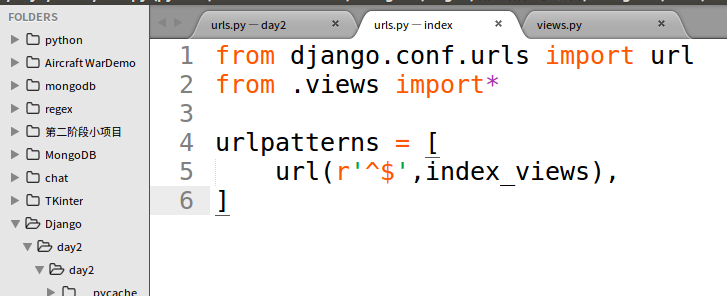
交给sport应用中的urls处理（找index\_views视图）

4.访问http://localhost:8000

交给index应用中的urls处理（找index\_views视图）







5 Django中的模板

5.1 什么是模板？

模板要动态给用户呈现的网页

模板就是一个网页---前后端结合的网页

模板是通过视图（Views）呈现给用户的

5.2 模板的设置

在settings.py中TEMPLATES变量进行模板的设置



‘APP\_DIRS’：

True：优先从DIRS指定的目录中查找模板，如果没有找到，再搜索应用中的templates目录

DIRS:[‘index.temp’,’news.temp’…]每个模板下存放的文件夹

但是如果，DIRS内容中为空的话，那么Django会自动到每个应用下去搜索一个交templates的目录作为木模板的存放目录。

推荐：

1.DIRS内容为空

2.在每个应用中，创建一个templates目录

5.3 模板的加载方式

模板的加载需要在“视图”中完成

1.使用loader获取模板，通过HttpResponse进行相应

from django.template import loader

from django.http import HttpResponse

def index\_views(request):

#1通过loader加载模板(得到一个模板对象)

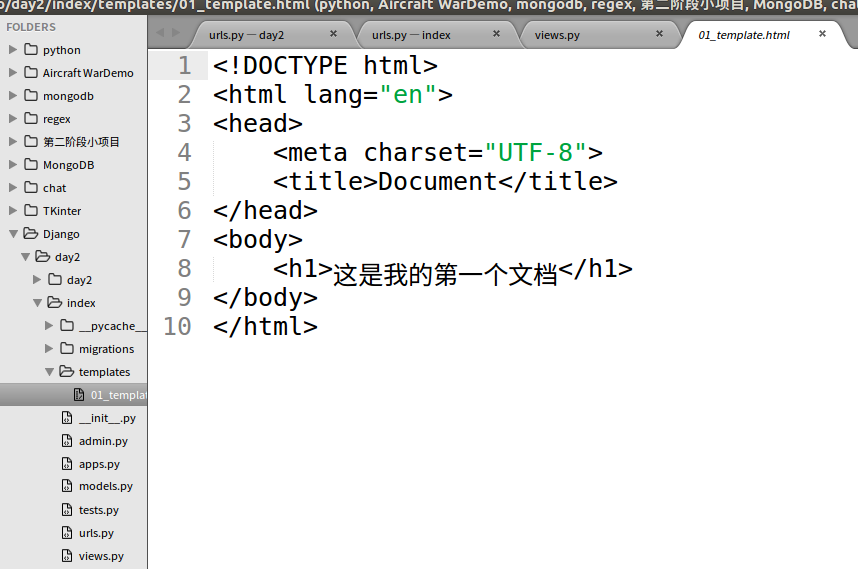
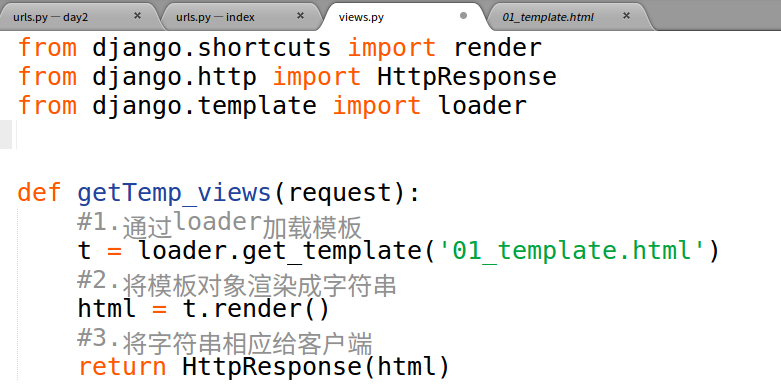
t = loader.get\_template(‘模板名称’)

#2通过模板对象t，将模板渲染成字符串

html = t.render({})

#3通过HttpResponse将html相应给客户端

return HttpResponse(html)



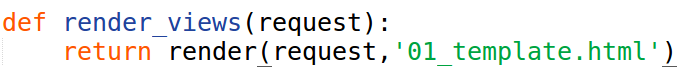
2.使用render直接加载并返回模板

from django.shortcuts import render

def xxx\_views(request):

return render(request,’模板名称’,’变量’)





6模板的语法

6.1 变量

1.作用：

允许将后端的数据传递给模板，在模板中，会根据变量的值进行显示

2.Django中允许传递给模板做变量的数据类型：

数字、字符串、列表、元组、字典、函数、对象

3.变量的语法

通过一个字典，将要传递给模板的变量封装起来

dic={

‘变量1’:’值1’,

‘变量2’:’值2’,

}

1.使用loader

t = loader.get\_template(‘模板名称’)

html = t.render(dic)

return HttpResponse(html)

2.使用render

return render(request,’模板名称’,dic)

4.在模板中使用变量

使用{{变量名}}

如果变量不存在或值为空的话，此位置不显示任何内容

