

小白学Django第五天| 视图View的初步使用

原创 JAP君 Python进击者

2020-01-31原文



小白学Django系列：

- 小白学Django第一天| MVC、MVT以及Django的那些事
- 小白学Django第二天| Django原来是这么玩的！
- 小白学Django第三天| 一文带你快速理解模型Model
- 小白学Django第四天| Django后台管理及配置MySQL数据库
- 持续更新中...

用最短的时间学最多的知识，本文大约花费6分钟

主要内容：

1. 视图函数的初步使用
2. MySQL配置的一些报错问题

视图函数的初步使用

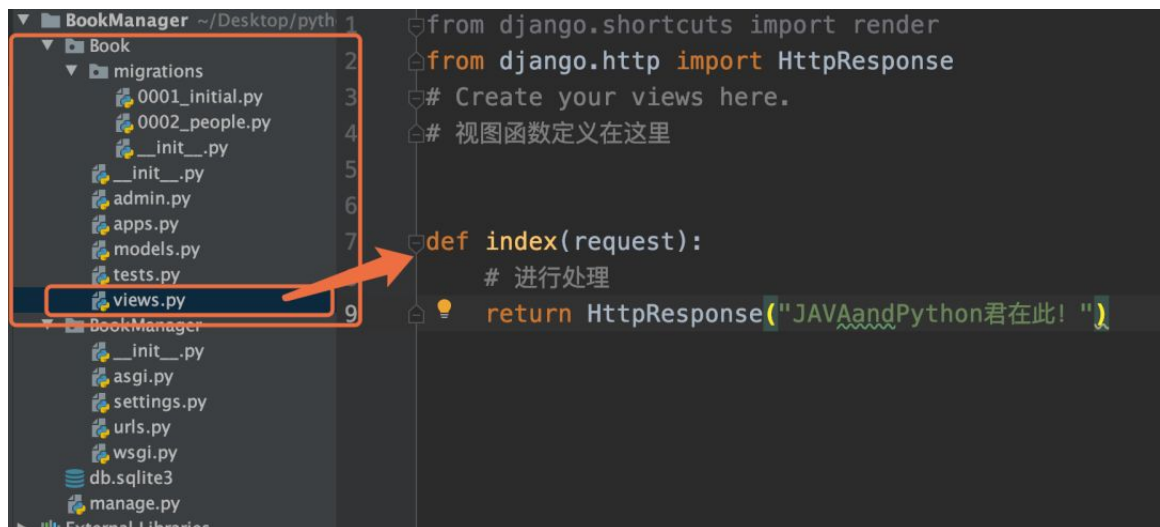
可能刚接触Django框架的朋友不理解视图函数是啥，其实在之前的Flask框架中也是有这一概念的。

在Django中，通过浏览器去请求一个页面时，此时使用视图函数来处理这个请求，视图函数处理完成之后，就要浏览器返回页面内容。

1. 定义视图函数

如何来定义一个视图函数呢？

视图函数定义在view.py文件中：

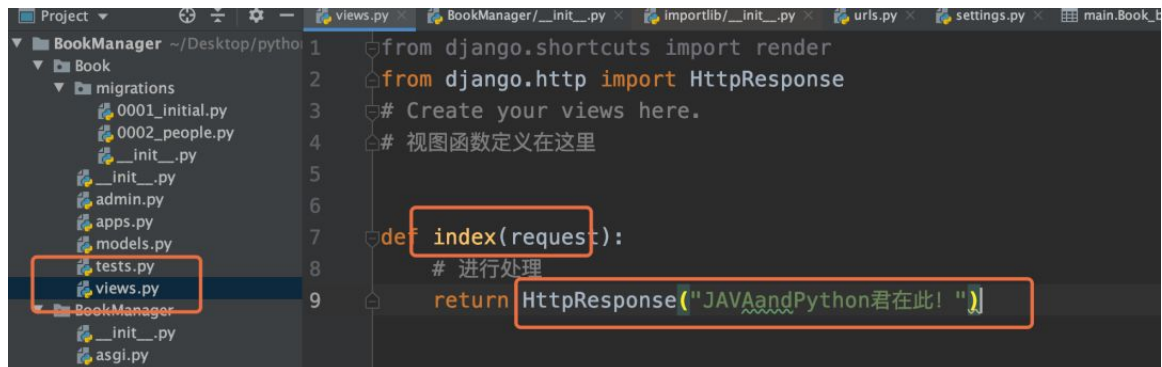


视图函数必须带一个参数request，进行处理之后，需要返回一个HttpResponse的类对象，里面包含的参数就是返回给浏览器显示的内容。

2.URL配置

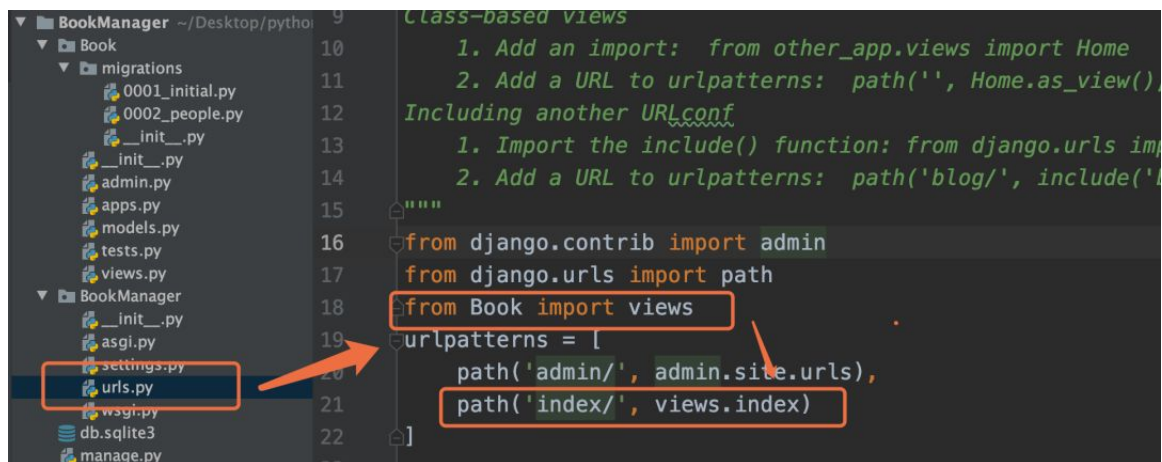
在django2.0以上的版本中，默认使用的是path转换器，我们来通过一个小例子来看看Django的Url配置。

下面是我们的视图函数：



```
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3 # Create your views here.
4 # 视图函数定义在这里
5
6
7 def index(request):
8     # 进行处理
9     return HttpResponseRedirect("JAVAandPython君在此! ")
```

接着我们去urls.py文件：



```
9
10
11
12
13
14
15
16 from django.contrib import admin
17 from django.urls import path
18 from Book import views
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('index/', views.index)
22 ]
```

我们配置完成之后可以通过
python manage.py runserver

来启动服务器，查看相应路径 <http://127.0.0.1:8000/index/>

← → ↻ ⓘ 127.0.0.1:8000/index/



应用



翻译



百度一下，你就知道



微信公众平台



JAVAandPython君在此！

发现正是我们在视图函数中所输出的数据。

当然，有关URL的配置还涉及到了转换器，**转换器是什么呢？简单来说就是能够路由提取的参数**。在后面这篇Flask系列的文章中我也讲到过转换器是啥，[小白学Flask第三天](#)| 今天把视图函数的路由给讲清楚！

我这里给出官方的例子：

```
from django.urls import path

from . import views

urlpatterns = [

    path('articles/2003/', views.special_case_2003),

    path('articles/<int:year>/', views.year_archive),

    path('articles/<int:year>/<int:month>/',
views.month_archive),

    path('articles/<int:year>/<int:month>/<slug:slug>/',
views.article_detail),

]
```

注意：

- 要从 URL 中取值，使用尖括号。
- 捕获的值可以选择性地包含转换器类型。比如，使用 `<int:name>` 来捕获整型参数。如果不包含转换器，则会匹配除了 `/` 外的任何字符。
- 这里不需要添加反斜杠，因为每个 URL 都有。比如，应该是 `articles` 而不是 `/articles`。

一些请求的例子：

- `/articles/2005/03/` 会匹配 URL 列表中的第三项。Django 会调用函数 `views.month_archive(request, year=2005, month=3)`。
- `/articles/2003/`
会匹配列表中的第一项，而不是第二项，因为这些模式是按顺序测试的，而第一项是第一个通过的测试。随意利用命令来插入类似这样的特殊情况。在这里，Django 将调用 `views.special_case_2003(request)`
- `/articles/2003`
不会匹配任何项，因为每一个匹配需要在最后加一个斜杠。
- `/articles/2003/03/building-a-django-site/` 会匹配 URL 列表中的最后一项。Django 会调用函数 `views.article_detail(request, year=2003, month=3, slug="building-a-django-site")`。

转换器这里不过多的解释，需要大家牢牢记住，这里提供给大家官方的案例：

<https://docs.djangoproject.com/zh-hans/3.0/topics/http/urls/>

以及我在 Flask 系列中解释过的转换器，整体的思路是一样的。

[小白学 Flask 第四天 | 把路由转换器玩的更牛逼](#)

[小白学 Flask 第三天 | 今天把视图函数的路由给讲清楚！](#)

补充知识点

在前一篇文章中小白学 Django 第四天 | Django后台管理及配置MySQL数据库 我们补充了Mysql数据库的配置，但是在实际的配置中，我们会遇到一些报错，例如：

```
django.core.exceptions.ImproperlyConfigured: mysqlclient
1.3.3 or newer is required; you have 0.7.11
```

这是由于python版本的原因，这里我给大家解决方案：

https://blog.csdn.net/qq_35304570/article/details/79674449

如果还有什么问题可以私聊我！

我是kuls

欢迎加我微信

备注姓名+学习方向+学校（公司）

拉你进Java、Python交流群



设为星标 ★



JAVAandPython君

📍 扫码关注不迷路

技术干货 · 经验分享 · 成长心得 · 系统教程

精选留言

暂无...

[阅读全文](#)