

2-建立 Django 博客应用

1、建立博客应用

我们已经建立了 Django 博客的项目工程，并且成功地运行了它。不过到目前为止这一切都还只是 Django 为我们创建的项目初始内容，Django 不可能为我们初始化生成博客代码，这些功能性代码都得由我们自己编写。

Django 鼓励我们把自己编写的代码组织到应用（Application）里，并且最好是一个应用只提供一种功能（类似于 flask 的蓝本！！！）。例如我们要开发的 Django 博客，相关的代码都放在 blog 这个应用里。其实应用也没什么复杂的，不过是把功能相关的代码组织到一个文件夹里，这个文件夹就成了一个应用（姑且可以这样理解）。

我们可以在 Django 工程目录下手动地建立一个文件夹，为它取个名字，并且告诉 Django 说这是一个应用。但是我们不必做这些无聊的工作，Django 已经为我们提供了自动创建应用的命令。

现在就来创建我们的 Django 博客应用，我把它命名为 blog。激活虚拟环境，进入到 manage.py 文件所在的目录下，运行

```
python manage.py startapp blog
```

命令即可建立一个 blog 应用。

前面已经说过，manage.py 是我们的工程管理助手。上一章中我们使用了它的 runserver 命令开启了一个本地开发服务器，在这里我们使用 manage.py 的 startapp 命令创建了一个 blog 应用。

2、应用的目录结构

用资源管理器打开 manage.py 文件所在的目录，可以看到多了一个 blog\ 的目录，该目录下面的文件结构如下：

```
blog\  
  __init__.py  
  admin.py  
  apps.py
```

```
migrations\  
  __init__.py  
models.py  
tests.py  
views.py
```

不同名称的文件用于存放特定功能的代码，这些将会在后面详细介绍。总之这个应用的文件夹结构 Django 已经为我们建立好了，但它还只是包含各种文件的一个文件夹而已，Django 目前还不知道这是一个应用。我们得告诉 Django 这是我们建立的应用，专业一点说就是在 Django 的配置文件中注册这个应用。

打开 blogproject\ 目录下的 settings.py 文件，看名字就知道 settings.py 是一个设置文件（setting 意为设置），找到 INSTALLED_APPS 设置项，将 blog 应用添加进去。

【blogproject/blogproject/settings.py】

其他配置项..

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'blog', # 注册 blog 应用  
]
```

其他配置项..

可以看到 Django 已经为我们提供了一些内置的应用，这些应用提供了各种各样的功能。这就是 Django 强大的地方，通用的功能 Django 都帮我们写好了，我们只需编写与自身业务相关的功能代码即可。