

`django-admin startproject blogproject` 建立一个工程

设计模型：

`python manage.py startapp blog` 建立一个APP

定义模型类

- 有一个数据表，就有一个模型类与之对应
- 打开models.py文件，定义模型类
- 引入包from django.db import models
- **模型类继承自models.Model类**
- **说明：不需要定义主键列，在生成时会自动添加，并且值为自动增长**
- 当输出对象时，会调用对象的str方法

激活模型：

编辑settings.py文件，将建立的APP应用加入到installed_apps中

生成迁移文件：根据模型类生成sql语句

`python manage.py makemigrations`

执行迁移：执行sql语句生成数据表

`python manage.py migrate`

管理站点：

站点分为“内容发布”和“公共访问”两部分

使用django的管理：

- 创建一个管理员用户

`python manage.py createsuperuser`，按提示输入用户名、邮箱、密码

管理界面本地化：

- 编辑settings.py文件，设置编码、时区

```
LANGUAGE_CODE = 'zh-Hans'
```

```
TIME_ZONE = 'Asia/Shanghai'
```

向admin注册booktest的模型：

- 打开booktest/admin.py文件，注册模型

```
from django.contrib import admin
```

```
from models import BookInfo
```

```
admin.site.register(BookInfo)
```

自定义管理页面：

- **Django提供了admin.ModelAdmin类**
- **通过定义ModelAdmin的子类，来定义模型在Admin界面的显示方式**

关联对象：

根据上述例子：

- 方式一：与BookInfo模型类相同
- 方式二：关联注册（）

```
from django.contrib import admin
```

```
from models import BookInfo, HeroInfo
```

```
class HeroInfoInline(admin.StackedInline):
```

```
    model = HeroInfo
```

```
    extra = 2
```

```
class BookInfoAdmin(admin.ModelAdmin):
```

```
    inlines = [HeroInfoInline]
```

```
admin.site.register(BookInfo, BookInfoAdmin)
```

视图：

- 在django中，视图对WEB请求进行回应
- 视图接收request对象作为第一个参数，包含了请求的信息

- 视图就是一个Python函数，被定义在views.py中

URLconf:

- 在Django中，定义URLconf包括正则表达式、视图两部分
- Django使用正则表达式匹配请求的URL，一旦匹配成功，则调用应用的视图
- **注意：只匹配路径部分，即除去域名、参数后的字符串**
- 在test1/urls.py插入booktest，使主urlconf连接到booktest.urls模块(在APP目录下建立urls.py文件)

```
url(r'^$', include('booktest.urls')),
```

- 在booktest中的urls.py中添加urlconf

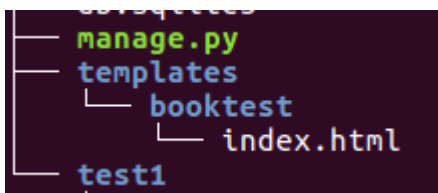
```
from django.conf.urls import url
from . import views
urlpatterns = [
    url(r'^$', views.index),
    url(r'^([0-9]+)/$', views.detail),
]
```

调用过程：主目录urls.py --> APP/urls.py --> views.py --> template/*.html

在线教程：<https://code.ziqiangxuetang.com/django/django-template2.html>

模板：

- 模板是html页面，可以根据视图中传递的数据填充值
- 创建模板的目录如下图：



- 修改settings.py文件，设置TEMPLATES的DIRS值

```
'DIRS': [os.path.join(BASE_DIR, 'templates')],
```

流程：

