

小白学Django第四天| Django后台管理及配置MySQL数据库

原创 JAP君 Python进击者

2020-01-20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
- 日更中...

用最短的时间学最多的知识，本文大约花费7分钟

本文内容：

1. 后台管理
2. 配置MySQL数据库

1. 后台管理

当我们在开发一个网站的时候，我们往往需要对数据表中的数据进行增删改查。例如在电商网中，每一个商家对商品的上架和下架以及商品信息的编辑都是需要对数据表进行操作的。

在Django框架中，提供了一种机制，能够帮助我们自动生成相应的数据表后台管理。



(图取自第一天学习)

上图中的文件admin.py正是后台管理相关的文件，具体如何使用这个Django后台管理，看下文：

1、首先，我们需要去修改setting.py文件，这个文件里包含着整个项目的总体配置。修改什么呢？我们需要本地化语言和时间。

```
103
104 # Internationalization
105 # https://docs.djangoproject.com/en/3.0/topics/i18n/
106
107 LANGUAGE_CODE = 'en-us'
108
109 TIME_ZONE = 'UTC'
110
111 USE_I18N = True
112
113 USE_L10N = True
114
115 USE_TZ = True
116
117
118 # Static files (CSS, JavaScript, Images)
119 # https://docs.djangoproject.com/en/3.0/howto/static-files/
120
121 STATIC_URL = '/static/'
122
```

(默认)

因为我们处在中国，所以我们需要修改相应的时间和语言（根据自己实际情况而定）

```

105 # https://docs.djangoproject.com/en/3.0/topics/i18n/
106
107 # LANGUAGE_CODE = 'en-us'
108     LANGUAGE_CODE = 'zh-hans'
109
110 # TIME_ZONE = 'UTC'
111     TIME_ZONE = 'Asia/Shanghai'
112
113     USE_I18N = True
114
115     USE_L10N = True
116
117     USE_TZ = True
118
119
120 # Static files (CSS, JavaScript, Images)
121 # https://docs.djangoproject.com/en/3.0/howto/static-files/
122
123     STATIC_URL = '/static/'

```

(修改后)

2、创建管理员账户

```
python manage.py createsuperuser
```

通过上述命令，我们可以迅速的创建一个管理员账户

```

kulsdeMacBook-Pro:BookManager kuls$ python manage.py createsuperuser
用户名 (leave blank to use 'kuls'): kuls
电子邮件地址: kuls666@163.com
Password:
Password (again):

```

3、登录管理后台

```
python manage.py runserver
```

```
kulsdeMacBook-Pro:BookManager kuls$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 20, 2020 - 12:59:39
Django version 2.2.5, using settings 'BookManager.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

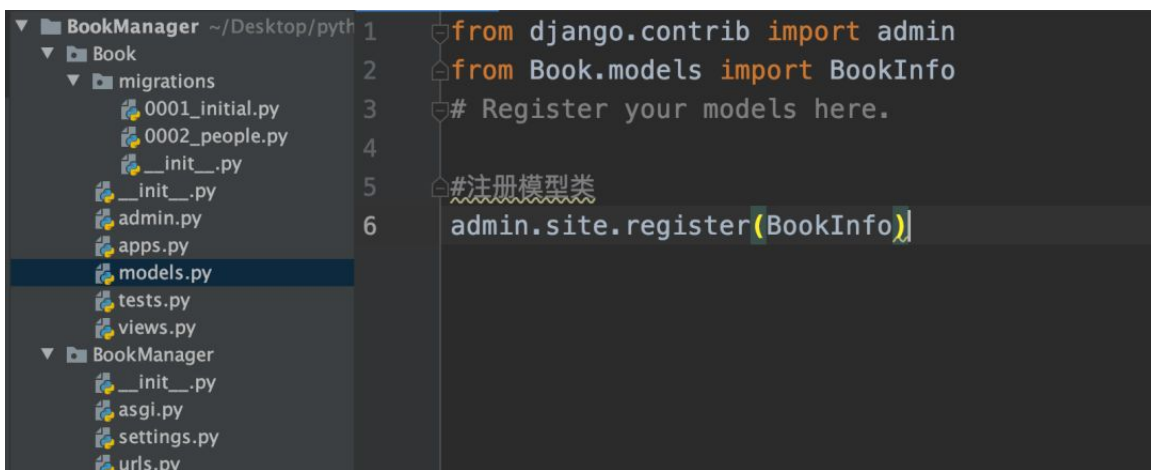
此时我们访问 <http://127.0.0.1:8000/admin> 即可（记得后缀加admin）



（后台管理页面）

4、注册模型类

进入之后我们会发现什么都没有，所以接下来我们需要在应用下的admin.py文件中注册模型类，相应的Django会根据注册的模型类生成对应的管理页面。



我们刷新后台会发现多出了我们刚刚注册的模型类：

Django 管理

站点管理

BOOK	
Book infos	+ 增加 ✎ 修改

认证和授权	
用户	+ 增加 ✎ 修改
组	+ 增加 ✎ 修改

最近动作

我的动作

无可用的

点进去看，会发现有两个对象：

Django 管理

首页 > Book > Book infos

选择 book info 来修改

动作 2 个中 0 个被选

<input type="checkbox"/>	BOOK INFO
<input type="checkbox"/>	BookInfo object (2)
<input type="checkbox"/>	BookInfo object (1)

2 book infos

我们可以通过重写模型类中的`str`方法，使这里更加直观的显示书名

```
admin.py × models.py × settings.py × __init__.py × console [db.sqlite3] × main.Bo
3
4 # Create your models here.
5 class BookInfo(models.Model):
6     '''图书模型类'''
7
8     # 图书名称
9     book_title = models.CharField(max_length=20)
10    # 图书日期
11    book_date = models.DateField()
12
13    def __str__(self):
14        return self.book_title
15
16
```

这两个对象就是我们在之前文章中所添加的两本书籍信息：

Django 管理

首页 > Book > Book infos

选择 book info 来修改

动作

执行

 2 个中 0 个被选

☐ BOOK INFO

☐ 西游记

☐ Django从入门到放弃

2 book infos

Django 管理

欢迎, KULS 查看站点 / 修改密码 / 注销

首页 > Book > Book infos > Bookinfo object (2)

修改 book info

历史

Book title:

西游记

Book date:

1986/01/01

今天

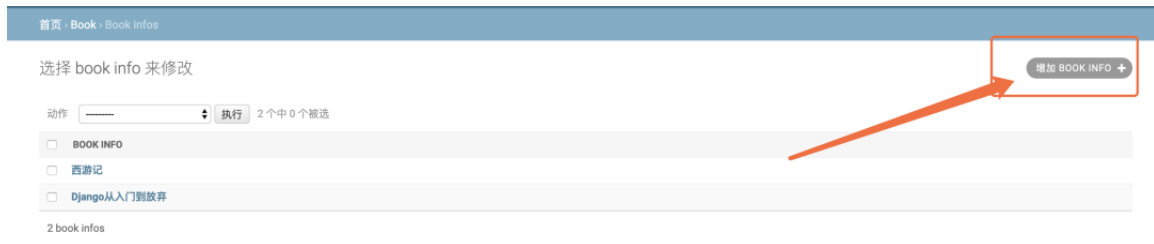
删除

保存并增加另一个

保存并继续编辑

保存

在后台管理页面，我们可以非常快捷的给模型增添新的数据：

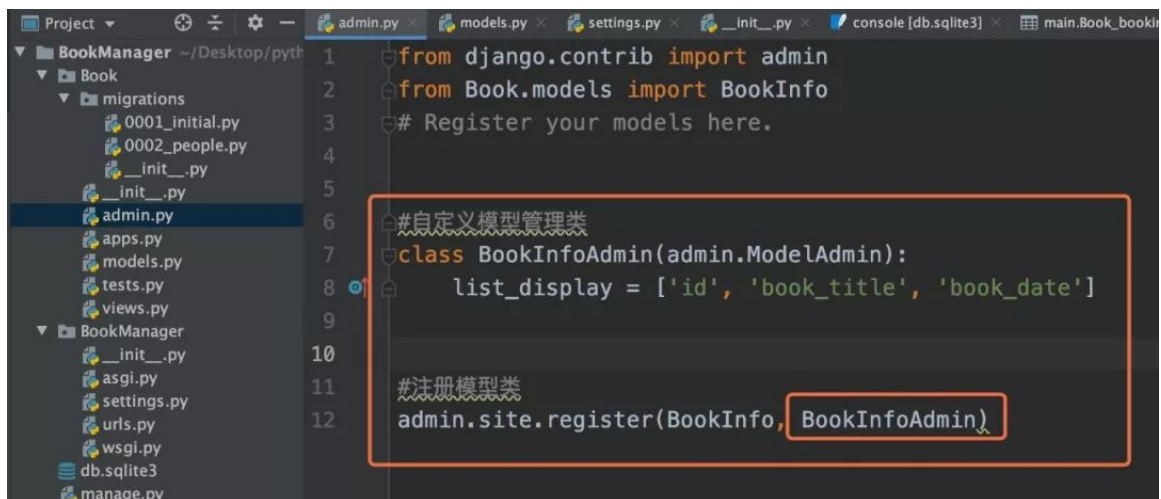


页面上的增添、修改、删除也都是可以通过命令实现的，只不过Django框架提供了这种快速而且方便的方式。这里的删除等操作我就不一一展示，大家自行挖掘。

5.自定义管理页面

大家可能会发现这个页面未免也太单调了吧，他并没有像数据库那样把所有的详细数据给展示出来，当然也是可以的，**Django后台管理是可以自定义管理页面**：

首先，我们需要自定义模型管理类，这个类就是告诉Django，我需要展示出什么东西：



```
1 from django.contrib import admin
2 from Book.models import BookInfo
3 # Register your models here.
4
5
6 #自定义模型管理类
7 class BookInfoAdmin(admin.ModelAdmin):
8     list_display = ['id', 'book_title', 'book_date']
9
10
11 #注册模型类
12 admin.site.register(BookInfo, BookInfoAdmin)
```

此时，再去看我们的管理页面



就已经把我们需要展示的数据给展示出来了。

2.配置MySQL数据库

这里给大家补充一个知识，因为我们大部分的项目数据库都是使用MySQL，但是Django默认配置的是sqlite3数据库，所以今天带着大家来修改一下数据库配置。

```
72
73
74 # Database
75 # https://docs.djangoproject.com/en/3.0/ref/settings/#databases
76
77 DATABASES = {
78     'default': {
79         'ENGINE': 'django.db.backends.sqlite3',
80         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
81     }
82 }
83
84
85 # Password validation
```

在前面，我们说过setting.py是整个项目的配置文件，所以我们需要在这个文件中来修改。

```
76
77 # DATABASES = {
78 #     'default': {
79 #         'ENGINE': 'django.db.backends.sqlite3',
80 #         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
81 #     }
82 # }
83
84 DATABASES = {
85     'default': {
86         'ENGINE': 'django.db.backends.mysql', # 数据库引擎
87         'NAME': 'Bookdb', # 数据库名称
88         'HOST': 'localhost', # 数据库主机
89         'PORT': '3306', # 数据库端口
90         'USER': 'root', # 数据库用户名
91         'PASSWORD': 'mysql', # 数据库密码
92     }
93 }
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql', # 数据库引擎  
        'NAME': 'Bookdb', # 数据库名称  
        'HOST': 'localhost', # 数据库主机  
        'PORT': '3306', # 数据库端口  
        'USER': 'root', # 数据库用户名  
        'PASSWORD': 'mysql', # 数据库密码  
    }  
}
```

}

我们只需要把内容改成上述代码，并且根据自己情况修改即可。

原创不易，欢迎转发

我是kuls

欢迎加我微信

备注姓名+学习方向

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...