

Please visit website: <http://cxyroad.com>

『Django』自带的后台

=====

本文简介

=====

****点赞 + + 收藏 = 学会了****

![00.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a671560b69ff46df81607aa28bfa5d99~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1198&h=722&s=284720&e=png&b=ffffff)

上一篇讲了 `Django` 操作 `MySQL` 的方法，讲了如何创建模型，如何对数据库做增删改查的操作。但每次修改数据都要写代码，多少有点麻烦。

有没有简单一点的方法呢？

有的有的，`Django` 其实自带了一个后台，是图形界面的，你可以在上面很方便的操作各种数据内容。

接下来就讲讲这个自带的后台怎么用。

配置后台路由地址

=====

其实在创建 `Django` 项目时就已经帮我们后台的路由地址配置好了。

如果你是跟着我前面的文章一步步做的话，可能后台路由会被删掉。没关系，补回去就好了。

打开项目的 `urls.py` 文件，找到 `urlpatterns` 这个变量，把后台路由补上。

```
...
# urls.py 项目的路由文件

from django.urls import path
from blog.views import blogIndex

urlpatterns = [
    # 省略了其他路由
    # 下面这句才是配置后台的路由
    path('admin/', admin.site.urls)
]
...
```

配置完成后，启动项目，在浏览器打开 `http://127.0.0.1:8000/admin/` 就能看到后台的登录页面。

![01.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/454052486c574746b2519a5396711dda~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1608&h=1450&s=118662&e=png&a=1&b=fafafa)

创建管理员账号
=====

有登录页面，那就要有对应的账号才能登录上去。

所以我们要先创建一个超管账号。

打开终端，进入项目目录，输入以下这条命令创建超管账号。

```
...
python manage.py createsuperuser
...
```

然后终端会返回这一行信息给你，让你输入超管的用户名（账号）

...

Username (leave blank to use 'xxx'): admin

...

这里的'xxx'是我的计算机名字，你在冒号后面输入超管的账号即可。这里我输入的是`admin`。

输入完用户名后，它会继续让你输入邮箱。

...

Email address: 123@qq.com

...

输入完邮箱会让你输入2次密码。

...

Password:
Password (again):

...

我这里输入的密码是`12345678`，它很智能的提醒我的密码太常见了，但我输入`y`，坚持使用这么简单的密码。

...

This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

...

看到`Superuser created successfully.`这句就表示账号创建成功了。

此时在浏览器打开`http://127.0.0.1:8000/admin/`，账号输入`admin`，密码输入`12345678`就能登录成功。

![02.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a5e1249d3ead4336b7cd65484c7b9bbf~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1608&h=1450&s=156755&e=png&a=1&b=fefefe)

模型关联后台

=====

我们前面都是使用代码的方式去操作数据库，不是那么方便。

‘Django’提供了一种方式，让我们将应用的模型和后台绑定，这样就可以在后台界面里直接操作数据了。

以上一篇文章创建的 ‘account’ 应用为例。

在 ‘account’ 应用中找到 ‘admin.py’ 文件，写入以下代码。

```
...  
# account/admin.py  
  
from django.contrib import admin  
from account.models import User  
  
class UserAdmin(admin.ModelAdmin):  
    # 要展示的内容  
    list_display = ['username', 'email']  
  
# 将 User 模型绑定到 UserAdmin  
admin.site.register(User, UserAdmin)  
...
```

这里需要先引入 ‘admin’ 和 ‘User’，因为我们要将 ‘account’ 应用的模型和后台关联起来。

创建一个名为 ‘UserAdmin’ 的类，在里面写明 ‘list_display = [‘username’, ‘email’]’，意思是要在后台页面显示 ‘username’ 和 ‘email’ 这两个字段。

最后 ‘admin.site.register(User, UserAdmin)’ 的意思是将 ‘User’ 模型和后台

绑定在一起。

运行项目，打开 `http://127.0.0.1:8000/admin/` 并登录，就能看到刚刚绑定的模型了。

![03.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a296869fb6a1406dacbac349190bec2c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2128&h=1086&s=167668&e=png&a=1&b=fefefe)

点击“用户表s”会看到在数据库中这个表的信息。

![04.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/62b34ad7b4ae4effbe55aeaa8c5b2c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2128&h=1220&s=211512&e=png&a=1&b=fdcfcc)

可以点击“Add”按钮在这个表里新增数据。

![05.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/2d239b0f957b41178d6b24301792251a~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1330&s=198559&e=png&a=1&b=fefdfd)

也可以对指定数据进行修改、删除操作。

![06.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/b8a0aa9b452b47aca24184b61e7d877c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1330&s=224410&e=png&a=1&b=fefdfd)

后台页面修改完的数据会直接影响到数据库里的数据的。

以同样的方式，把 `blog` 应用的模型和后台关联起来。这里的操作和前面是相同的，我就不再重复讲解了。

```
...
# blog/admin.py

from django.contrib import admin
from blog.models import Article

# Register your models here.
class ArticleAdmin(admin.ModelAdmin):
    # 要展示的内容
    list_display = ['id', 'title']

# 将 User 模型绑定到 UserAdmin
admin.site.register(Article, ArticleAdmin)
...
```

![07.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/6253b13f64ea49e09b09147542e3252d~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1330&s=200563&e=png&a=1&b=fefdfd)

我们创建一篇文章看看。

点击“文章信息表s”旁边的Add按钮，然后填写好以下内容。

![08.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/a75cb942dc67407c887eb699eabe680f~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1684&s=284948&e=png&a=1&b=fdfcfc)

点击 `SAVE` 按钮后就将数据写入到数据库了，回到“文章信息表s”的列表页就能看到刚刚写入的数据。

![09.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/2ff0824bcbec4c9d86862f6aa18c4ec0~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1684&s=247021&e=png&a=1&b=feffcfc)

如果不太清楚整个项目的内容，可以看回我整理的[《Django专栏》](http://cxyroad.com/)
”http://mp.weixin.qq.com/mp/homepage?__biz=MzAwMjU3ODU5Ng==&

hid=22&sn=331e4ac92051cf4165fae2fa0542f525&scene=18#wechat_redirect”)。

后台的基础配置

=====

在前面的操作中，细心的工友可能已经留意到一些问题，比如界面怎么是英文的？为什么表名后面会有个“s”？为什么创建文章的时候User这个字段可选的不是用户名，而是`User object(1)`？

接下来我们把这些问题的一个个解决。

设置时区和语言

打开项目的`settings.py`文件，找到`LANGUAGE_CODE`，将它的值设置为`zh-hans`。然后再找到`TIME_ZONE`将其设置为`Asia/Shanghai`。

...

```
# settings.py
# 语言
LANGUAGE_CODE = 'zh-hans'
```

```
# 时区
TIME_ZONE = 'Asia/Shanghai'
```

...

![10.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/2219321d07004b2288e436617b9b2d3b~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1684&s=224118&e=png&a=1&b=febfd)

将栏目名改成中文

但此时`ACCOUNT`和`BLOG`仍然是英文的，这里`Django`是不能直接将其翻译成中文，需要你给这两个应用配置别名。

![11.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/8e24071e505245d1a12a7badcefbfc01~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=730&h=772&s=58078&e=png&a=1&b=fcf8f8)

配置的地方在各个应用自身的 `apps.py` 文件里，通过 `verbose_name` 变量来自定义栏目名称。

以 `account` 应用为例。

```
...
# account/apps.py

from django.apps import AppConfig

class AccountConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'account'
    verbose_name = '用户管理'
```

...

`blog` 应用同理。

...

```
# blog/apps.py

from django.apps import AppConfig

class BlogConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'blog'
    verbose_name = '文章管理'
```

...

此时再刷新页面就能看到栏目名称被改过来了。

![12.png](https://p9-juejin.byteimg.com/tos-cn-i-

k3u1fbpfcg/b5f7248f28594503941851d3acebe32e~tplv-k3u1fbpfcg-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1684&s=225235&e=png&a=1&b=febfd)

删除复数形式

表名后面的“s”是 `Django` 自带的，要删掉这个“s”同样需要配置表的别名。

在各个应用的模型文件（`models.py`）里配置 `Meta` 信息的 `verbose_name_plural`。

以 `account` 应用为例。

```
...
# account/models.py

from django.db import models

# 创建一个用户表
class User(models.Model):
    id = models.AutoField(primary_key=True)
    # 用户名
    username = models.CharField('用户名', max_length=30, null=True,
blank=True, unique=True)
    # 密码
    password = models.CharField('密码', max_length=30)
    # 邮箱
    email = models.EmailField('邮箱', unique=True, null=True, blank=True)

    class Meta:
        # 自定义表名
        db_table = 'user'
        # 表别名
        verbose_name = '用户表'
        verbose_name_plural = '用户表'
...

```

刷新一下页面就能看到“用户表”后面的“s”没了。

![13.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/5a3eb744476b4bbcb26e6550fac52047~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2226&h=1684&s=224772&e=png&a=1&b=fefdfd)

`blog` 同理。

展示外键字段

我们在添加文章的时候需要输入选择文章作者，文章表的作者是关联了用户表的。

在添加作者的时候页面直接展示了“User object”这种名字，用户数量一多的话我们就分不清谁是谁了。

![14.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/87e6c29bd1d34cf2aa8a738d9db16989~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=746&h=306&s=58021&e=png&a=1&b=faafa)

我们应该将用户名展示出来。

这里我们先修改一下 `blog` 要展示的字段。

```
...
# blog/admin.py

from django.contrib import admin
from blog.models import Article

# 获取作者
# 这个 obj 对象是用来接收 Article 实例
def get_author(obj):
    return obj.user.username

# Register your models here.
class ArticleAdmin(admin.ModelAdmin):
    # 要展示的内容
```

```
# 此时这里就可以使用 get_author 换取用户名了。这里不需要加上单引号，也不需要加上括号。
```

```
# django 在使用 list_display 时，会默认给方法传递一个参数。
```

```
list_display = ['id', get_author, 'title', 'content']
```

```
get_author.short_description = '作者'
```

```
# 将 User 模型绑定到 UserAdmin
```

```
admin.site.register(Article, ArticleAdmin)
```

```
...
```

通过 `get_author` 方法把外键关联的 `username` 字段展示出来了。

```
![15.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/3878a096407043b49901d5d7a5da7861~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2658&h=1254&s=233008&e=png&a=1&b=fdfcfc)
```

此时我们将能看到这篇文章的作者是谁了。但我们还需要在编辑页里的用户名展示出来才行。

此时需要修改 `User` 模型才行。

```
...
```

```
# account/models.py
```

```
from django.db import models
```

```
# 创建一个用户表
```

```
class User(models.Model):
```

```
    id = models.AutoField(primary_key=True)
```

```
    # 用户名
```

```
    username = models.CharField('用户名', max_length=30, null=True, blank=True, unique=True)
```

```
    # 密码
```

```
    password = models.CharField('密码', max_length=30)
```

```
    # 邮箱
```

```
    email = models.EmailField('邮箱', unique=True, null=True, blank=True)
```

```
class Meta:
```

```
    # 自定义表名
```

```
db_table = 'user'
# 表别名
verbose_name = '用户表'
verbose_name_plural = '用户表'

# 返回用户名
def __str__(self):
    return self.username

...
```

在 `account/models.py` 的最后加上 `__str__` 方法，返回 `username`。

此时在文章信息的编辑页里就能看到用户名了。

![16.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/08de2084a1054117a4a9f3dbd380ae7d~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2648&h=1256&s=251903&e=png&a=1&b=fefdfd)

自定义编辑的跳转链接

在文章管理页面里，想要编辑文章只能点击“ID”这个字段才能跳转进编辑页。如果想通过点击其他字段也能跳转进编辑页就需要手动配置 `blog/admin.py` 了，比如我想点击“TITLE”也能跳进编辑页。

![17.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/45249f93060e422588466f8dda68c874~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2658&h=1254&s=233449&e=png&a=1&b=fdffc)

...

```
# blog/admin.py
```

```
# 省略部分代码
```

```
class ArticleAdmin(admin.ModelAdmin):
    # 要展示的内容
    list_display = ['id', get_author, 'title', 'content']
    # 可跳转的字段
```

```
list_display_links = ['id', 'title']  
# 省略部分代码
```

...

我在 `ArticleAdmin` 里添加了 `list_display_links = ['id', 'title']`，表示点击 `id` 和 `title` 都可以跳转到编辑页。

只读属性

在编辑内容时，我们不希望某些字段可以修改。比如博客的id不能被修改。

可以在 `blog/admin.py` 中添加 `readonly_fields` 指定不能被修改的字段。

...

```
# blog/admin.py
```

```
# 省略部分代码
```

```
class ArticleAdmin(admin.ModelAdmin):  
    # 只读属性  
    readonly_fields = ['id']  
    # 省略部分代码
```

...

此时要编辑时，id字段的值就不能修改了。

![18.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/553b42c8df8e4e9c9542a5a5bb71f11b~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=2658&h=1630&s=258816&e=png&a=1&b=fefefe)

以上就是本文的全部内容。下一篇我会介绍如何在前台页面加载数据库里的信息。

![IMG_8982.GIF](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/9bebc3a541634c74b53e11c690a09394~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=282&h=385&s=7530673&e=gif&f=125&b=bab1a5)

****点赞 + + 收藏 = 学会了****

原文链接: <https://juejin.cn/post/7389229430345220131>