



# django实战项目 内容管理系统cms (二)数据模型models

D.Roger · 2016年6月28日 · 9 条评论

雨水：雨水之日獭祭鱼，又五日鸿雁来（候雁北），又五日草木萌动。



## 1. 项目分析：

在构建数据模型前需要先分析项目需求。

cms(Content Management System内容管理系统) 这个项目主要功能：

- 1>,文章的呈现：包括标题，内容，作者，评论，赞同数，收藏数。
- 2>,注册用户信息：昵称，邮箱，密码，简介，收藏了的文章，赞同了的文章，评论了的文章。

我们允许用户对一篇文章多次评论；但点赞和收藏的操作一篇文章只能进行一次，为了实现这一点，需要知道每篇文章和每个用户的对应关系(是否已点赞，是否已收藏)。

我们创建用户和文章，评论三个数据模型，让用户和文章关联成多对多的关系，从而判断用户是否已经收藏了某篇文章，另外创建一个点赞的数据模型，来判断用户是否已经赞同过某篇文章。

## 2. models.py

当我们创建了focus这个应用(app)的时候，cms/focus这个文件夹中会自动生成models.py这个文件。

### 2.1 普通用户的数据模型：

在django中内置有django.contrib.auth.models.User模块，这个模块字段有限，但是我们可以继承django.contrib.auth.models.AbstractUser这个类来扩充字段，之所以不自己写User模块，是因为内置的User模块使整个用户验证系统非常容易实现。User模块内置的字段参见 [django 文档](#)，下面我们只要再扩充一个字段即可：

```
1 import datetime
2 from django.db import models
3 from django.contrib.auth.models import User
4 from django.utils.encoding import python_2_unicode_compatible
5 from django.utils import timezone
6 from django.contrib.auth.models import AbstractUser
7
8 @python_2_unicode_compatible
9 class NewUser(AbstractUser):
10     profile = models.CharField('profile', default='', max_length=256)
11
12     def __str__(self):
13         return self.username
```

@python\_2\_unicode\_compatible是django内置的兼容python2和python3的unicode语法的一个装饰器。

只是针对\_\_str\_\_方法而用的，\_\_str\_\_方法是为了后台管理(admin)和django shell的显示，Meta类也是为后台显示服务的，可以查阅 [django 文档](#) 了解。

所有的数据模型都继承自models.Model这个模块，models.ForeignKey/CharField/TextField/...这些都是django内置的模型字段，相关知识请查阅： [django 文档](#)

继承AbstractUser这个类，我们需要在cms/settings.py中申明默认的用户类，我们在settings.py中加入一条申明：

```
1 AUTH_USER_MODEL = "focus.NewUser"
```

## 2.2 column字段是文章所属的分类，文章和分类是多对一的关系。我们构建一个Column数据模型：

```
1 @python_2_unicode_compatible
2 class Column(models.Model):
3     name = models.CharField('column_name', max_length=256)
4     intro = models.TextField('introduction', default='')
5
6     def __str__(self):
7         return self.name
8
9     class Meta:
10         verbose_name = 'column'
11         verbose_name_plural = 'column'
12         ordering = ['name']
```

## 2.3 创建文章(article)这个数据模型，

```
1 @python_2_unicode_compatible
2 class Article(models.Model):
3     column = models.ForeignKey(Column, blank=True, null=True, verbose_name = 'belong to')
```

```
4 title = models.CharField(max_length=256)
5 author = models.ForeignKey('Author')
6 user = models.ManyToManyField('NewUser', blank=True)
7 content = models.TextField('content')
8 pub_date = models.DateTimeField(auto_now_add=True, editable=True)
9 update_time = models.DateTimeField(auto_now=True, null=True)
10 published = models.BooleanField('notDraft', default=True)
11 poll_num = models.IntegerField(default=0)
12 comment_num = models.IntegerField(default=0)
13 keep_num = models.IntegerField(default=0)
14
15 def __str__(self):
16     return self.title
17
18 class Meta:
19     verbose_name = 'article'
20     verbose_name_plural = 'article'
```

author是文章的作者，为了方便，我们把作者和普通用户分别用Author,NewUser这两个数据模型来管理，文章和作者是多对一的关系，文章和普通用户是多对多的关系(点赞，收藏)。

## 2.4 然后是评论的数据模型：

```
1 @python_2_unicode_compatible
2 class Comment(models.Model):
3     user = models.ForeignKey('NewUser', null=True)
4     article = models.ForeignKey(Article, null=True)
5     content = models.TextField()
6     pub_date = models.DateTimeField(auto_now_add=True, editable=True)
7     poll_num = models.IntegerField(default=0)
8     def __str__(self):
9         return self.content
```

## 2.5 作者的数据模型：

```
1 class Author(models.Model):
2     name = models.CharField(max_length=256)
3     profile = models.CharField('profile', default='', max_length=256)
4     password = models.CharField('password', max_length=256)
5     register_date = models.DateTimeField(auto_now_add=True, editable=True)
6
7     def __str__(self):
8         return self.name
```

## 2.6 增加点赞的数据模型：

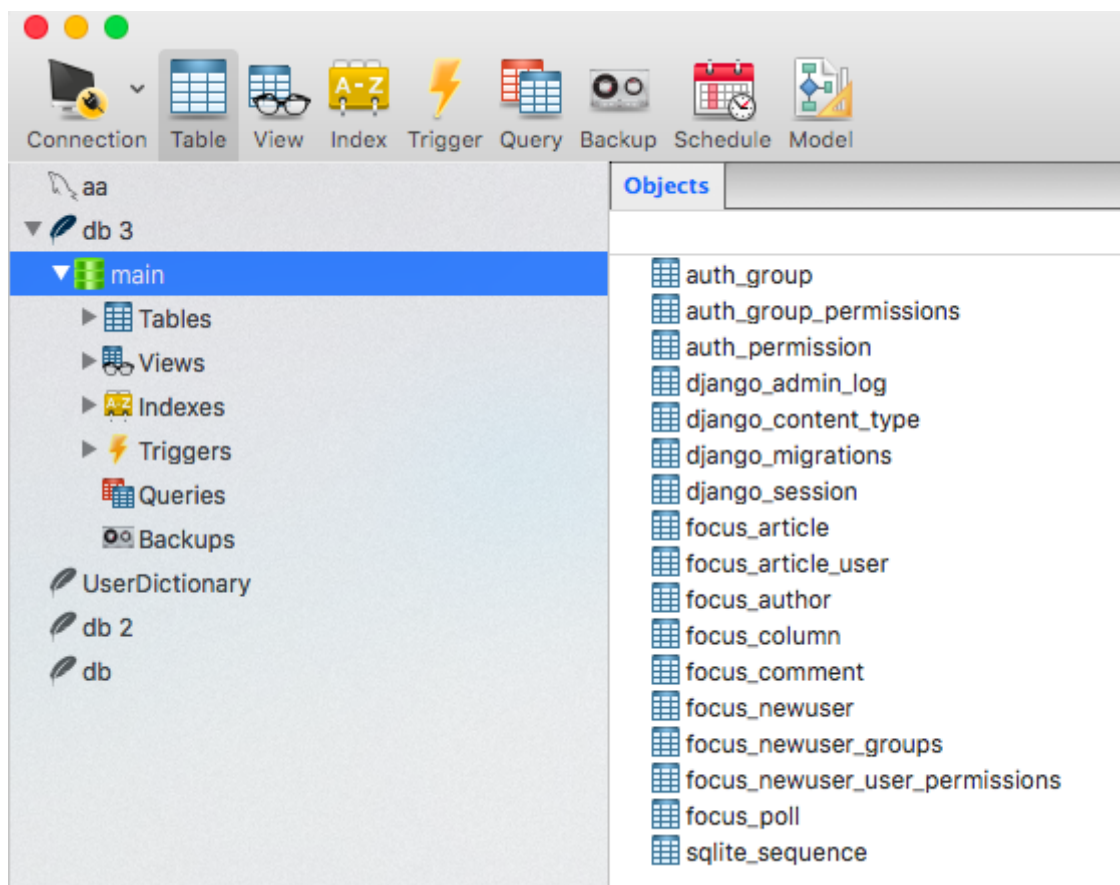
```
1 class Poll(models.Model):
2     user = models.ForeignKey('NewUser', null=True)
3     article = models.ForeignKey(Article, null=True)
4     comment = models.ForeignKey(Comment, null=True)
```

## 3. 同步数据模型到数据库：

查看settings.py中的DATABASES，django默认使用的数据库引擎是sqlite3，如果你需要改用其它引擎可以查阅[django doc](#)，此项目我们使用默认的sqlite3，在终端进入项目目录，输入如下命令：

```
1 python manage.py makemigrations focus
2 python manage.py migrate
```

这样就会自动生成数据表，我们使用Navicat Premium打开cms/db.sqlite3这个文件，



这些表有些是django自动生成的，有些是我们models.py模块同步生成的。

点开focus\_newuser这个表，我们发现它不仅profile这个字段，还有其它User模块内置的字段，focus\_article\_user这个表是因为文章和普通用户我们设置了多对多的关系，所以自动生成了这一张表来关联这两者。

标签： Django

««[django实战项目 内容管理系统cms \(一\)项目搭建及准备](#)

[django实战项目 内容管理系统cms \(三\)后台管理admin](#)»»

## 9 条评论



daderan • 2016年7月20日 • 回复

为什么在class Article中 column和author都设置外键的时候author后面有引号呢



D.Roger • 2016年7月20日 • 回复 • 作者



因为在定义article时, column已经定义了, 而author还没有定义, 参考<http://python.usyiyi.cn/django/ref/models/fields.html#module-django.db.models.fields.related> 如果不加引号 makemigration时会报错。



daderan • 2016年8月4日 • 回复

感谢说明, 懂了。教程很棒。



robot • 2016年8月6日 • 回复

感谢博主。请问你的代码用的是什么字体？



D.Roger • 2016年8月6日 • 回复 • 作者

monaco



changliu • 2016年10月16日 • 回复

感谢 之前学的django有点忘了 准备跟着博主的来一遍然后放到heroku上



jwj • 2017年1月16日 • 回复

用python3就好了



sabrina • 2017年2月9日 • 回复

NewUser 类不是在前面定义了吗, 为什么在article这个模型中添加多对多关系时newuser用引号引起来了



zhao • 2017年2月17日 • 回复

我的cms里面建了两个, focus1 和focus2, focus1.models中建立NewUser模型, focus2中建立MyUser模型, 执行python manage.py makemigrations focus1的时候提示: Reverse accessor for 'NewUser.groups' clashes with reverse accessor for 'MyUser.groups', 把focus2中的MyUser完全删掉才可以正常导入, 老师能解释下为什么会这样吗

发表评论

已登入为D.Roger。登出？

评论

发表评论