NSD Devweb DAY04

1 案例1：安装django

1.1 问题

1. 创建python虚拟环境
2. 激活python虚拟环境
3. 在虚拟环境中安装django
4. 验证django是否安装正确

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建python虚拟环境

虚拟环境是python多版本管理的利器，可以搭建独立的python运行环境，有助于包的管理和防止版本冲突。可以理解为在自己的电脑上安装了一个虚拟电脑。

1)打开终端，创建pyproject目录，使用cd命令，切换到需要创建虚拟环境的目录：

```
[root@localhost ~]# mkdir pyproject
[root@localhost ~]# cd pyproject/
```

2)使用如下命令，在当前目录创建虚拟环境：

```
[root@localhost pyproject]# python3 -m venv django_env
```

步骤二：激活python虚拟环境

```
[root@localhost pyproject]# source django_env/bin/activate
```

Top

步骤三：在虚拟环境中安装django

```
(django_env) [root@localhost pyproject]# pip install django==1.11.6
```

步骤四：验证django是否安装正确

在python解释器输入以下命令检查是否安装成功，如果输出了django的版本号说明安装成功。

```
(django_env) [root@localhost pyproject]# python
Python 3.6.4 (default, Apr 27 2018, 08:26:23)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> django.__version__
'1.11.6'
```

2 案例2：创建项目

2.1 问题

1. 创建名为mysite的项目
2. 生成数据库
3. 创建超级用户
4. 登录后台管理页面

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建名为mysite的项目

1)安装django之后，您现在应该已经有了可用的管理工具django-admin.py。我们可以使用django-admin命令来创建mysite项目：

```
(django_env) [root@localhost pyproject]# django-admin startproject mysite
```

2)创建完成后我们可以查看下项目的目录结构：

```
(django_env) [root@localhost pyproject]# cd mysite
(django_env) [root@localhost mysite]# tree
.
├── manage.py
└── mysite
```

Top

```
06.         ├── __init__.py
07.         ├── settings.py
08.         ├── urls.py
09.         └── wsgi.py
```

目录说明：

mysite：项目的容器。

manage.py：一个使用的命令行工具，可让你以各种方式与该django项目进行交互

mysite/__init__.py：一个空文件，告诉python该目录是一个python包。

mysite/settings.py：该django项目的设置/配置。

mysite/urls.py：该django项目的URL声明；一份由django驱动的网站“目录”。

mysite/wsgi.py：一个WSGI兼容的Web服务器的入口，以便运行你的项目。

3)接下来我们进入mysite目录输入以下命令，启动服务器：

```
01.  (django_env) [root@localhost mysite]# python manage.py runserver 0.0.0.0:8000
```

0.0.0.0让其它电脑可连接到开发服务器，8000为端口号，如果不说明，那么端口号默认为8000

注意：django默认使用sqlite3文件数据，如果不能使用，需要安装sqlite-devel：

```
01.  (django_env) [root@localhost mysite]# yum install sqlite-devel
02.
03.  ================================================================================
04.  安装 1 软件包
05.
06.  总下载量：104k
07.  安装大小：366k
08.  Is this ok [y/d/N]: y
09.  Downloading packages:
10.  Running transaction check
11.  Running transaction test
12.  Transaction test succeeded
13.  Running transaction
14.    正在安装    : sqlite-devel-3.7.17-8.el7.x86_64                          1/1
15.    验证中      : sqlite-devel-3.7.17-8.el7.x86_64                          1/1
16.
17.  已安装:
18.    sqlite-devel.x86_64 0:3.7.17-8.el7
```

Top

```
19.
20.     完毕！
```

然后重新编译python3，找到python源码包，里面有一个configure可执行文件，过程如下：

```
01.  [root@localhost ~]# cd /opt/Python-3.6.1/
02.  [root@localhost Python-3.6.1]# ls
03.  aclocal.m4     Include          Modules         Python
04.  build          install-sh       Objects         python-config
05.  config.guess   Lib              Parser          python-config.py
06.  config.log     libpython3.6m.a  PC              python-gdb.py
07.  config.status  LICENSE          PCbuild         README.rst
08.  config.sub     Mac              Programs        setup.py
09.  configure      Makefile         pybuilddir.txt  Tools
10.  configure.ac   Makefile.pre     pyconfig.h
11.  Doc            Makefile.pre.in  pyconfig.h.in
12.  Grammar        Misc             python
13.  [root@localhost Python-3.6.1]# ./configure --prefix=/usr/local/bin    #配置编译
14.  [root@localhost Python-3.6.1]# make       #编译源码
15.  [root@localhost Python-3.6.1]# make install       #执行安装
```

4)在浏览器输入你服务器的ip及端口号，如果正常启动，输出信息如图-1所示：

图-1

5)在浏览器输入http://127.0.0.1:8000/admin可访问后台管理界面，输出信息如图-2所示：

Top

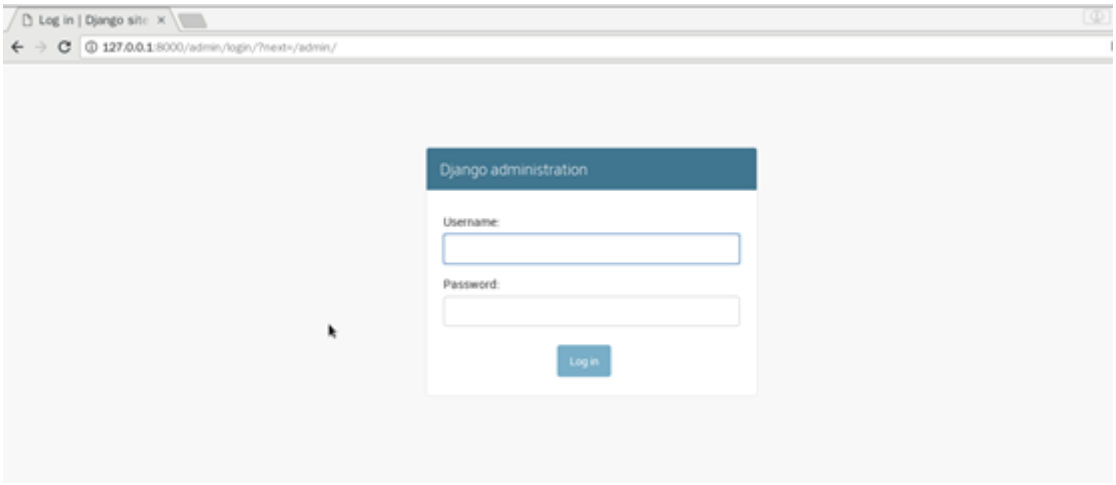

图-2

步骤二：生成数据库：

1)将model层转为迁移文件migration：

```
(django_env) [root@localhost mysite]# python manage.py makemigrations
No changes detected
```

2)将新版本的迁移文件执行，更新数据库：

```
(django_env) [root@localhost mysite]# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying sessions.0001_initial... OK
```

Top

注意：这两种命令调用默认为全局，即对所欲最新更改的model或迁移文件进行操作，如果想对部分app进行操作，就要在其后追加app name。

3)此时项目中会出现一个db.sqlite3文件，这就是默认的数据库文件。输出信息如图-3所示：

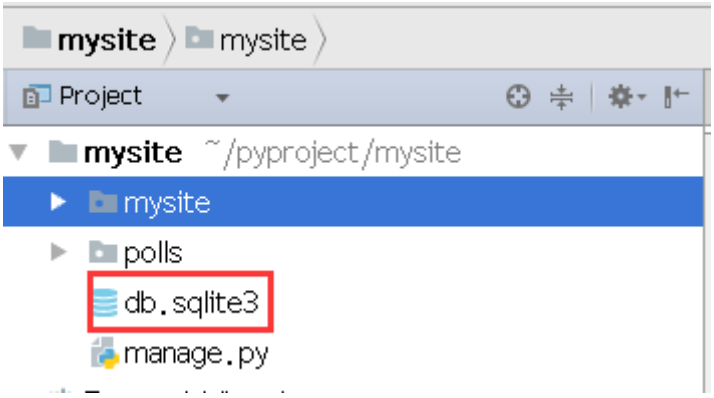

图-3

步骤三：创建超级用户：

输入如下命令，创建后台管理员账号：

```
(django_env) [root@localhost mysite]# python manage.py createsuperuser
Username (leave blank to use 'root'): admin
Email address: zzg@tedu.cn
Password:
Password (again):
Superuser created successfully.
```

步骤四：登录后台管理页面

启动django后，可以登录后台管理页面，登录成功输出信息如图-4所示：

Top

图-4

3 案例3：创建应用

3.1 问题

1. 创建应用polls
2. 注册polls到项目中
3. 配置URLconf，当访问http://127.0.0.1:8000/polls时，由polls进行路由
4. 为polls应用配置主页视图

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：创建应用

在每个django项目中可以包含多个app，相当于一个大型项目中的分系统、子模块、功能部件等，相互之间比较独立，但也有联系，所有的app共享项目资源，执行如下命令，用startapp命令创建app：

```
(django_env) [root@localhost mysite]# python manage.py startapp polls
```

此时项目目录结构如图-5所示：

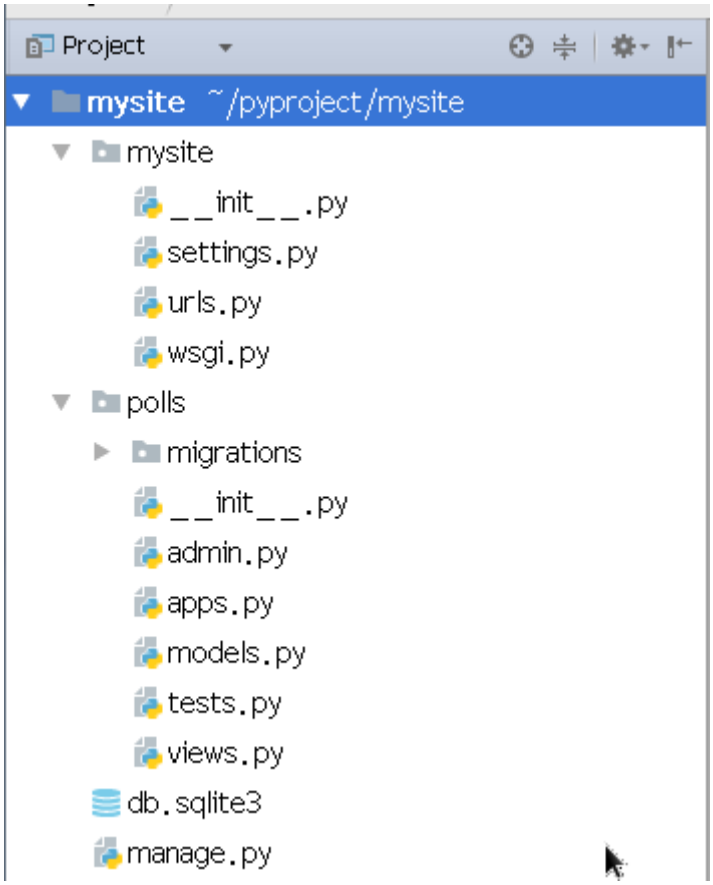

图-5

步骤二：激活应用

Top

为了让django包含创建的app，我们还需要激活app。打开配置文件setting.py，找到INSTALLED_APPS,然后把我们创建的app配置添加进去，这样django才能使用我们的app。

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'polls',
]
```

步骤三：配置URLconf，当访问http://127.0.0.1:8000/polls时，由polls进行路由

1)配置URLconf。在mysite/urls.py文件中添加app中设置的路径，除了admin页面的路径之外，其他路径都应该使用include函数引入

```
from django.conf.urls import url, include
from django.contrib import admin
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^polls/', include('polls.urls'))
]
```

2)创建polls应用的URLconf，即创建模块urls.py

```
(django_env) [root@localhost mysite]#touch polls/urls.py
```

3)绑定URL与视图函数，在polls/urls.py中，删除原代码，将一下代码写入urls.py文件中：

```
from django.conf.urls import url
from . import views
urlpatterns = [
    url(r'^$', views.index, name='index'),
]
```

步骤四：为polls应用配置主页视图 Top

1)在app的views.py中添加一个新的视图。创建views.index函数

```
01.  from django.http import HttpResponse
02.  def index(request):
03.      return HttpResponse("Hi! 这是polls应用的首页。")
```

通过上面的步骤，我们将index这个url指向了views里的index（）函数，它接收用户请求，并返回一个“Hi! 这是polls应用的首页。”字符串。

2)验证应用

启动django服务，访问http://127.0.0.1:8000/polls，显示如图-6所示：

图-6

4 案例4：创建模型

4.1 问题

1. 为polls应用创建模型
2. 一个模型名为Question，用于记录问题
3. 另一个模型名为Choice，每个Choice只能对应一个Question，但是一个Question可以对应多个Choice

4.2 方案

Django对各种数据库提供了很好的支持，包括：PostgreSQL、MySQL、SQLite、Oracle。Django为这些数据库提供了统一的调用API。我们可以根据自己业务需求选择不同的数据库。以下我们使用sqlite3默认数据库。

Django支持ORM模型，也就是说我们可以不使用SQL语句对数据进行增删改查。我们要做的就是在模型中指定和数据库的关系。

在这个简单的投票应用中，我们将创建两个模型： Question和Choice

Question对象具有一个question_text（问题）属性和一个publish_date（发布时间）属性，用于记录问题

Choice有两个字段：选择的内容和选择的得票统计

每个Choice与一个Question关联

4.3 步骤

实现此案例需要按照如下步骤进行。

Top

步骤一：打开配置文件settings.py，找到数据库一行，可以看到如下的配置，目前我们使用默认数据库，暂时不需要对如下配置进行修改。

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

步骤二：创建模型：

打开polls应用中的models.py文件，然后添加下面两个模型，在polls应用中的models.py文件添加两个模型，每个模型都用一个类表示，该类继承自django.db.models.Model，我们在定义模型的时候指定每一个字段的名字、长度、是否唯一等信息：

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

class Choice(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    choice_text = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)
```

值得注意的是，每个字段通过Field类的一个实例表示 ， 例如字符字段CharField、日期字段DateTimeField和整数字段IntegerField。 这种方法告诉Django，每个字段中保存着什么类型的数据。Models.ForeignKey用来指定外键约束。

5 案例5：生成数据库

5.1 问题

1. 执行迁移命令生成数据库
2. 在pycharm中打开数据库
3. 观察数据库表名和字段名，发现与模型的关系

5.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：执行迁移命令生成数据库

Top

1)polls应用建立了新的模型，该模型需要反馈到数据库中，通过运行makemigrations命令告诉django，已经对模型做了一些更改，并且会将这些更改记录为迁移文件，迁移文件位于polls/migrations/目录下：

```
(django_env) [root@localhost mysite]# python manage.py makemigrations
Migrations for 'polls':
  polls/migrations/0001_initial.py
    - Create model Choice
    - Create model Question
    - Add field question to choice
```

迁移文件查看如图-7所示：

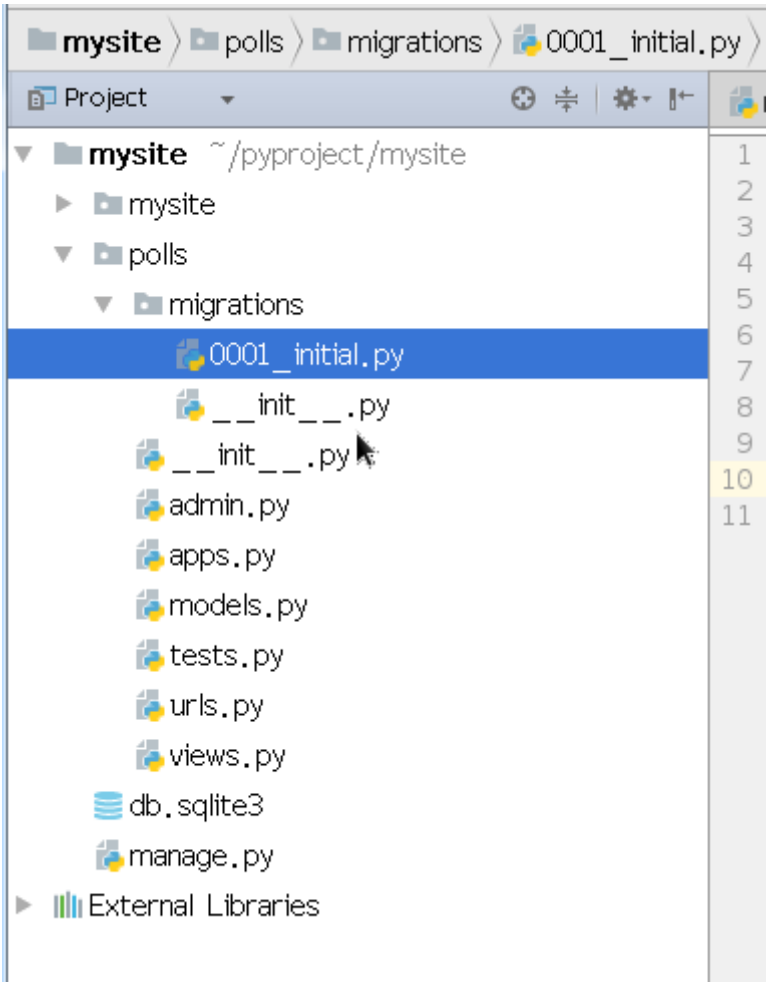

图-7

2)此时只是生成了迁移文件，并没有真正应用到数据库中。如果要引用到数据库，再次使用migrate命令即可：

```
(django_env) [root@localhost mysite]# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, polls, sessions
Running migrations:
  Applying polls.0001_initial... OK
```

Top

步骤二：在pycharm中打开数据库

1)通过pycharm的Database窗口查看数据库，如图-8所示：

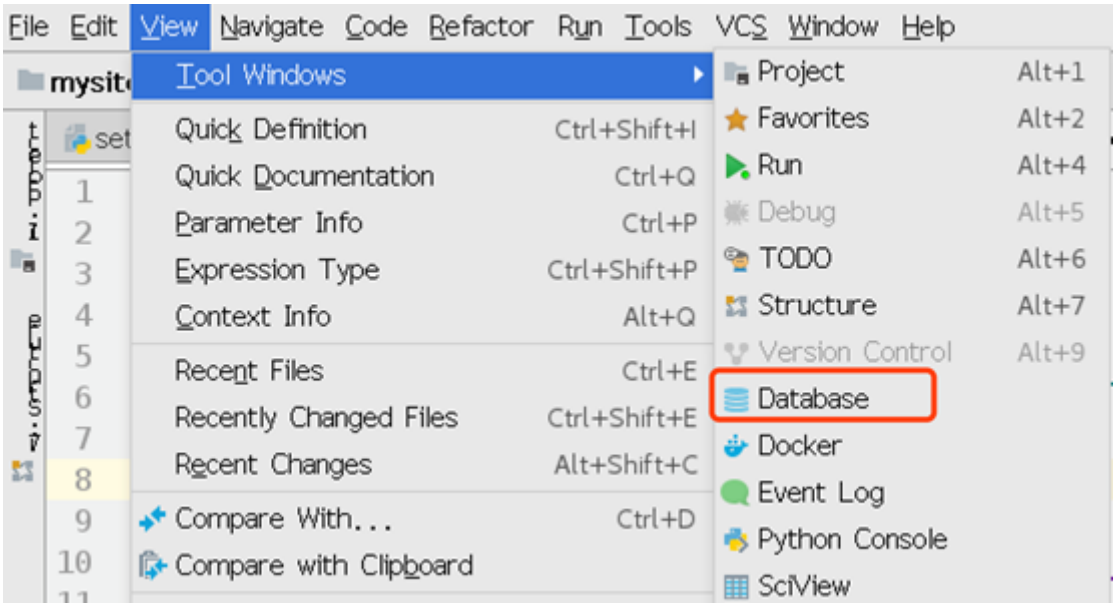

图-8

2)将db.sqlite3文件拖入Database窗口，如图-9所示：

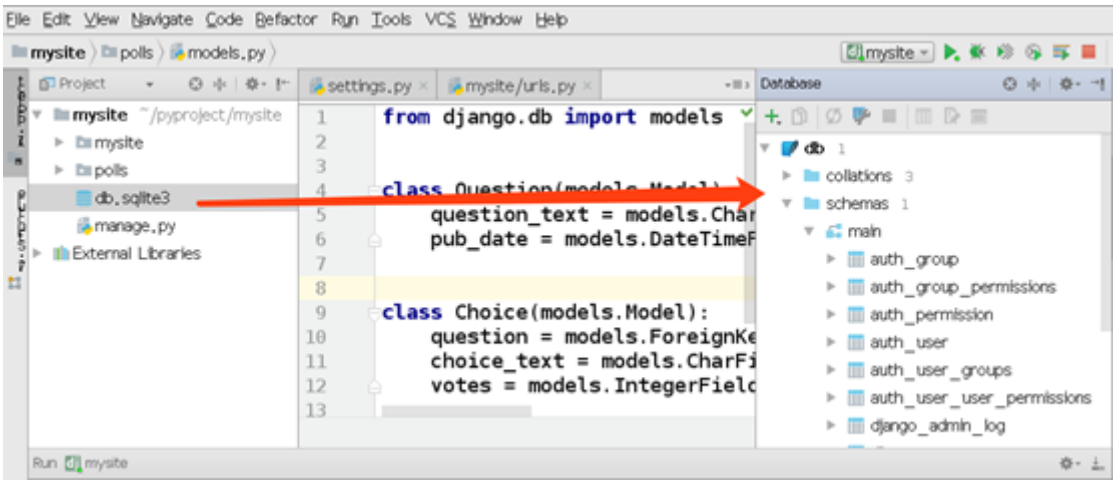

图-9

3)设置Database，如图-10所示：

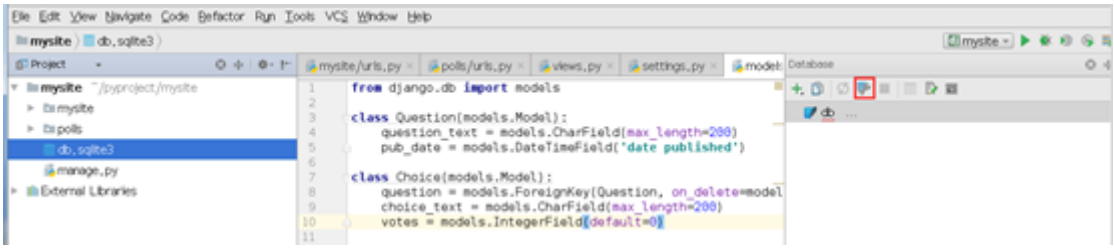

图-10

4)下载丢失的驱动程序文件，如图-11所示：

Top

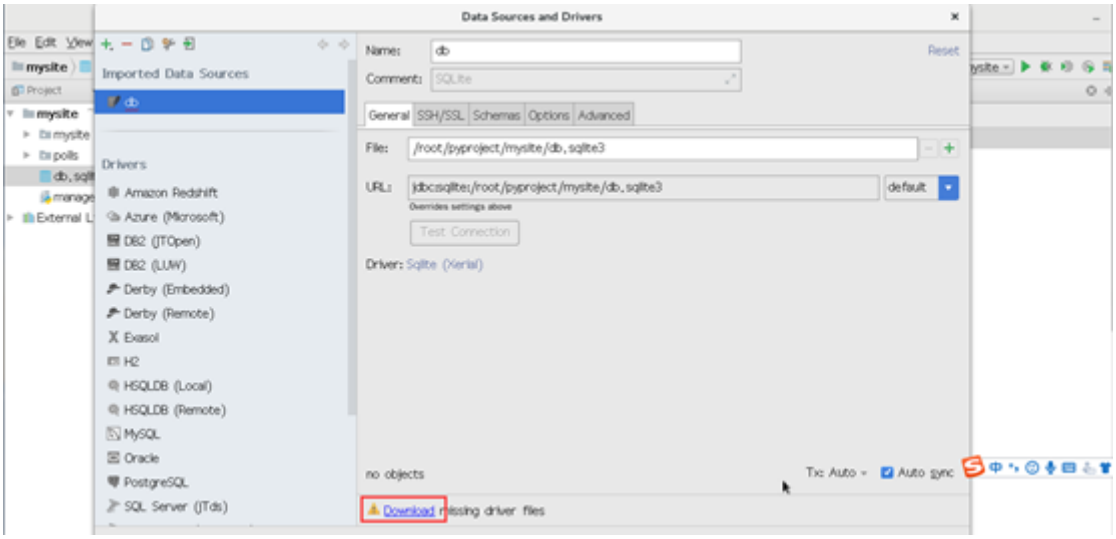

图-11

5)测试连接后，点击/按钮，完成设置，如图-12所示：

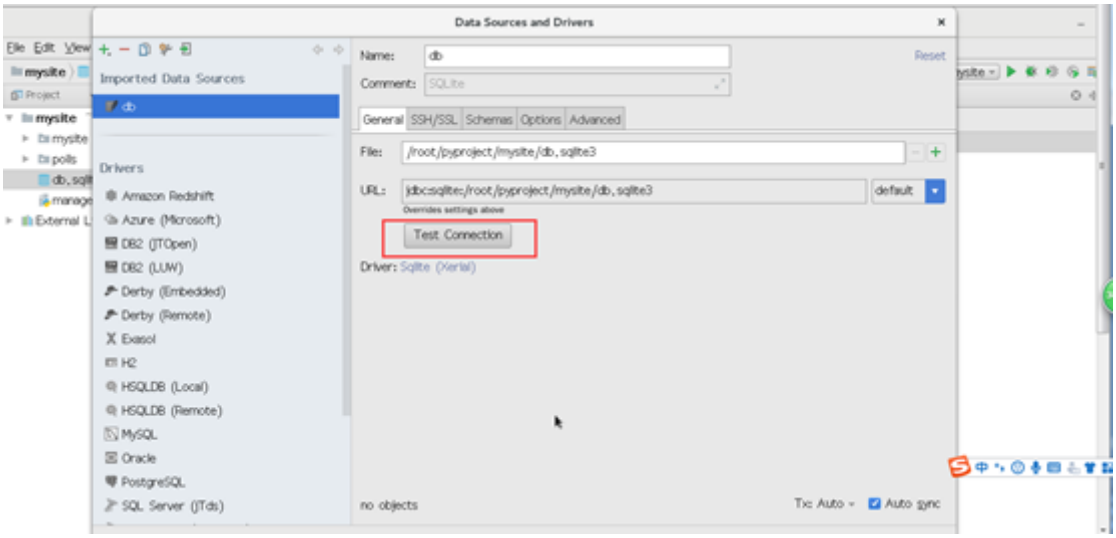

图-12

6)从Database窗口可查前面创建的数据库模型，如图-13所示：

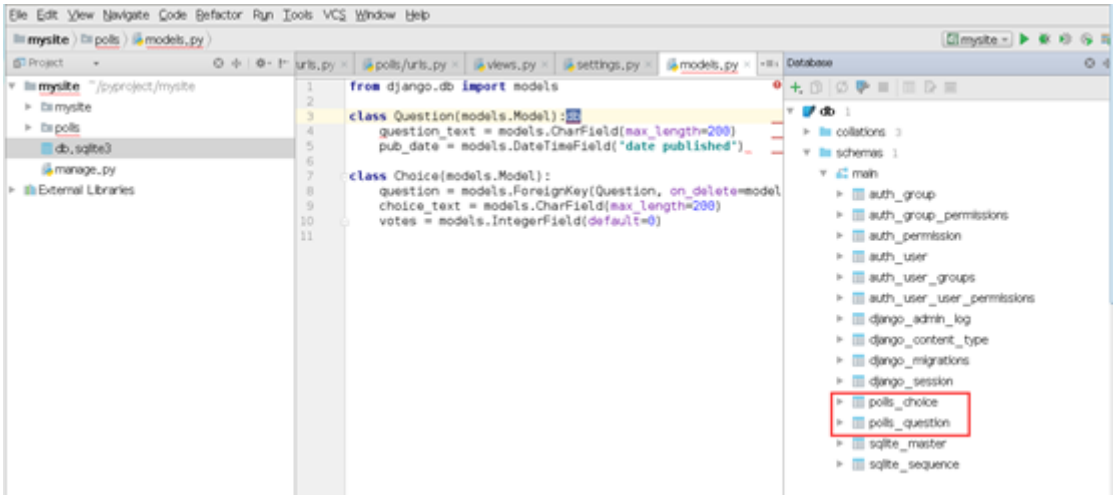

图-13

步骤三：观察数据库表名和字段名

Top

了解与模型的关系，如图-14所示：

图-14

6 案例6：注册后台

6.1 问题

1. 将Question和Choice模型添加到后台管理
2. 在后台页中填写几个问题和选项

6.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：将Question和Choice模型添加到后台管理

1)启动服务器，在浏览器中访问http://127.0.0.1:8000/admin，进入登录界面，输入创建过的超级用户的用户名密码登录，登录成功后，显示如图-15所示：

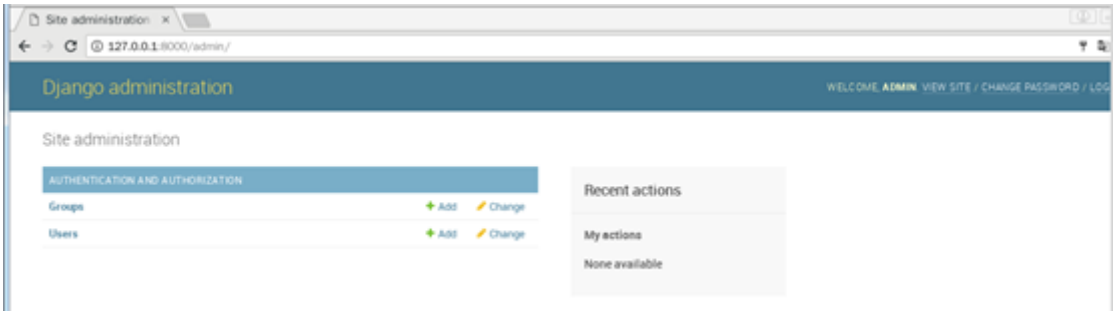

图-15

2)为了让admin界面管理某个数据模型，我们需要先注册该数据模型到admin。比如，我们之前在polls/models.py文件中已经创建了模型Question和Choice，修改polls/admin.py：

```
01.  from django.contrib import admin
02.  from .models import Question, Choice
03.
04.  admin.site.register(Question)
05.  admin.site.register(Choice)
```

3)刷新后即可看到polls数据表，如图-16所示：

Top

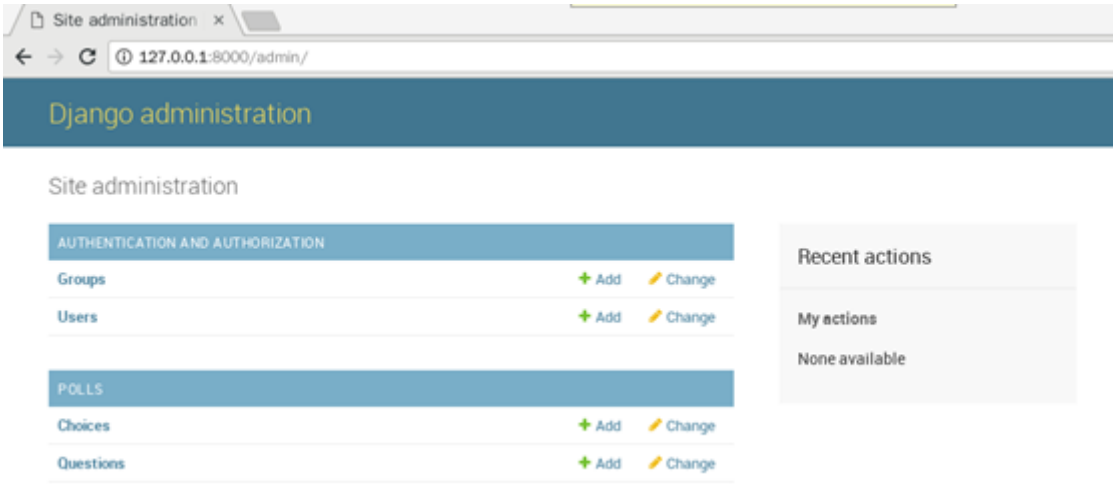

图-16

步骤二：在后台页中填写几个问题和选项

1)添加数据，如图-17所示

图-17

2)添加Question数据，如图-18所示：

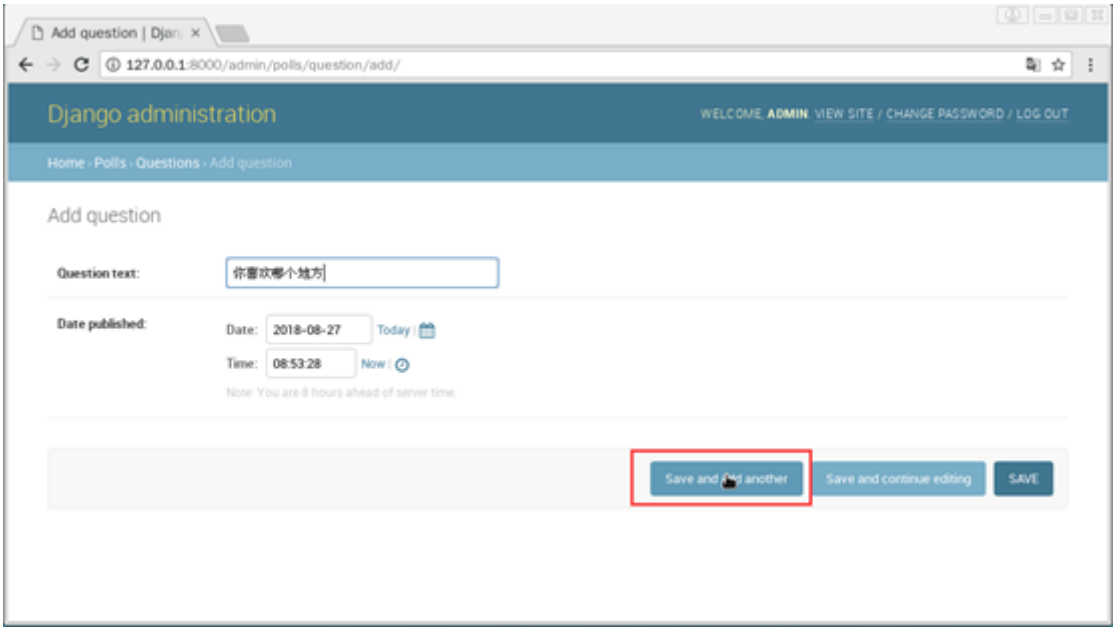

图-18

3)添加Choice数据，如图-19所示：

Top

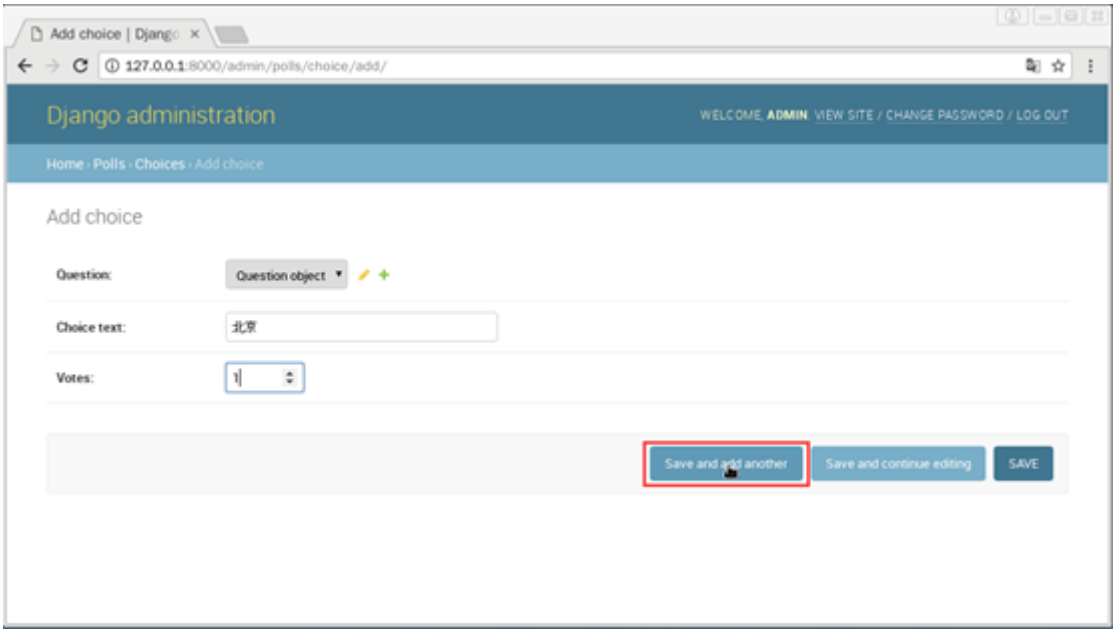
Add choice | Django
127.0.0.1:8000/admin/polls/choice/add/
Django administration
WELCOME, ADMIN VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Polls › Choices › Add choice
Add choice
Question:
Question object
Choice text:
北京
Votes:
1
Save and add another
Save and continue editing
SAVE

图-19

Top