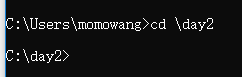
一、django环境的配置

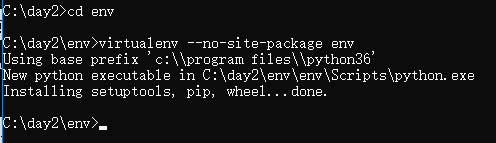
1. 在cmd中检查python环境是否存在、正确



1. 创建django项目的文件地址，然后cd命令进入所创建的文件夹

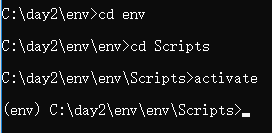


1. 在该文件夹下，新建一个env文件夹，然后cd进入该env文件夹，配置virtualenv虚拟环境 virtualenv –no-site-packages env

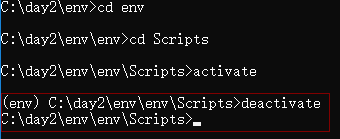


1. 进入env，在进行activate后才可以进行下一步的工作

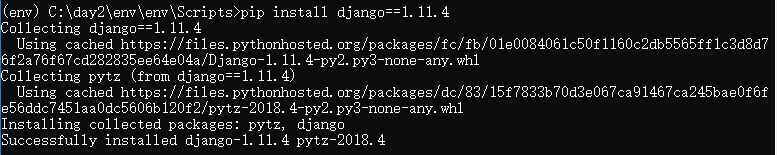
Linux下为source bin/activate



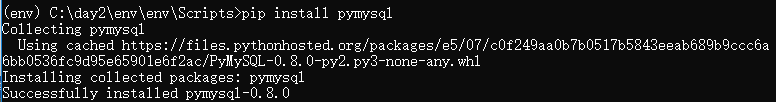
1. 退出env Linux下为source deactivate



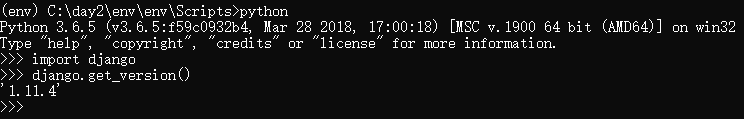
1. 导入django库，这里选择版本为django 1.11.4，注意：要在该环境中导入这些库，因为虚拟环境是彼此隔离的



1. 导入PyMySQL库

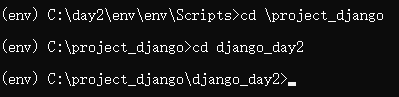


1. 验证django库是否安装成功

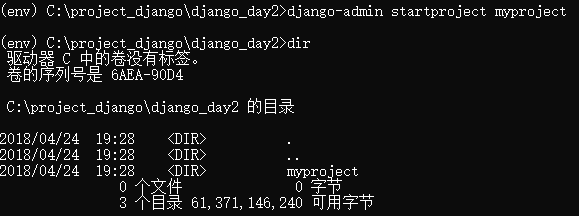


二、创建django项目

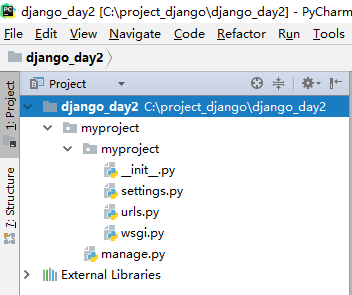
1. 第一个django项目的创建，先cd到要创建django项目的目录文件夹地址下



1. 创建工程（项目），名称为myproject的工程文件，dir命令查看myproject工程是否成功



1. 通过pycharm进行查看所建的工程



其中解释如下

manage.py： 是Django用于管理本项目的管理集工具，之后站点运行，数据库自动生成，数据表的修改等都是通过该文件完成。

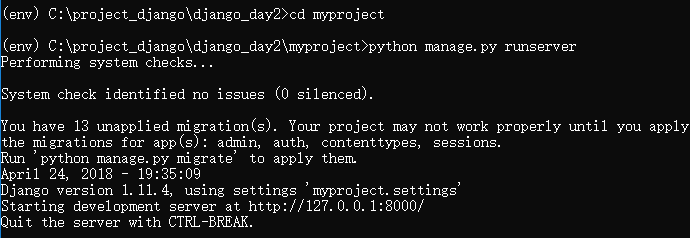
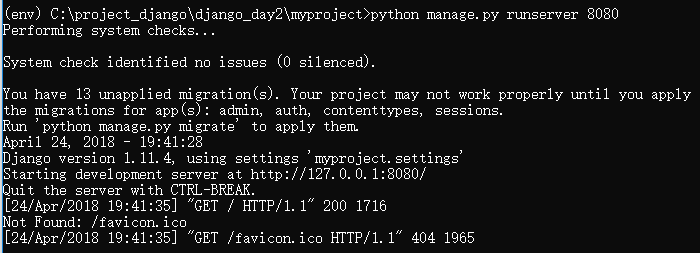
**init**.py： 指明该目录结构是一个python包，暂无内容，在后期会初始化一些工具会使用到。----初始化，配置pymysql链接的地方

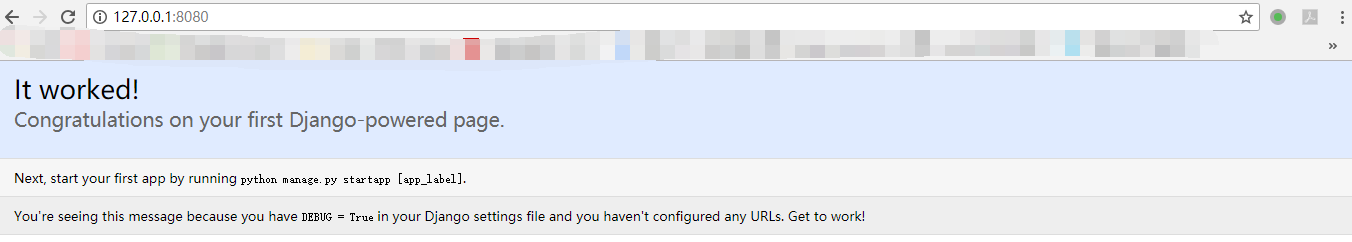
seetings.py： Django项目的配置文件，其中定义了本项目的引用组件，项目名，数据库，静态资源，调试模式，域名限制等 ----配置信息位置，database等

urls.py：项目的URL路由映射，实现客户端请求url由哪个模块进行响应。----url路由

wsgi.py：定义WSGI接口信息，通常本文件生成后无需改动。----网关

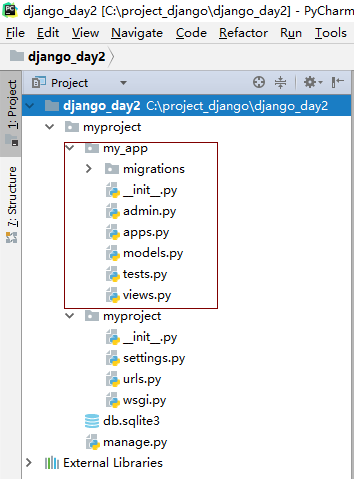
1. 运行django项目，需要先进入myproject项目中，然后再runserver，此处的端口可以先不用指定，默认为8000 localhost 或者 127.0.0.1:8080，这里的8080是自己指定的



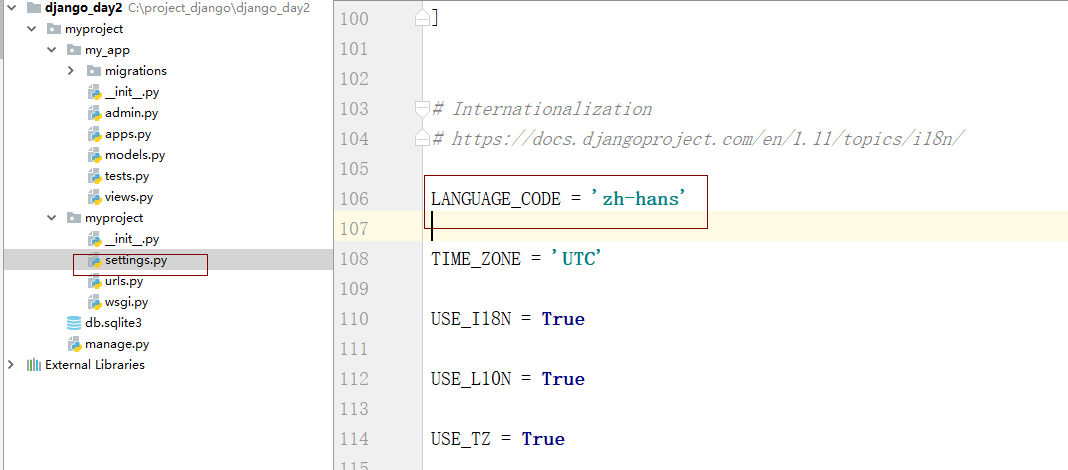


1. 创建app，注意该命令是再myproject工程下创建的一个名为my\_app的app

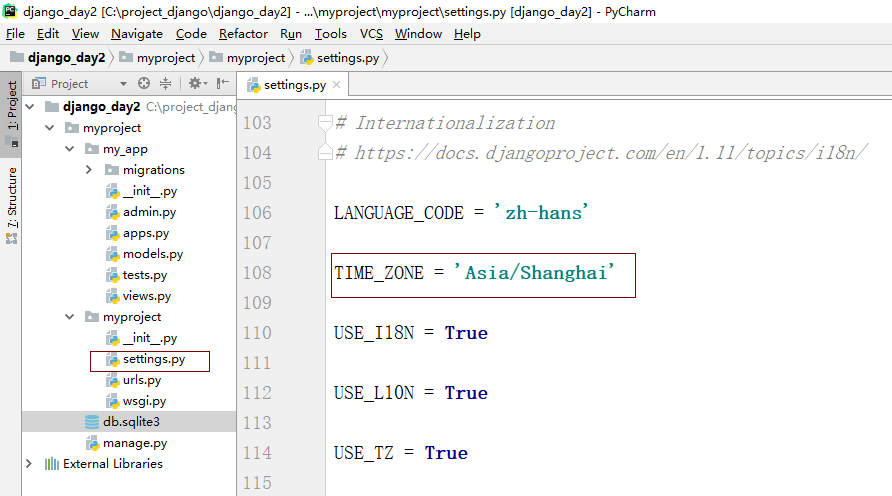




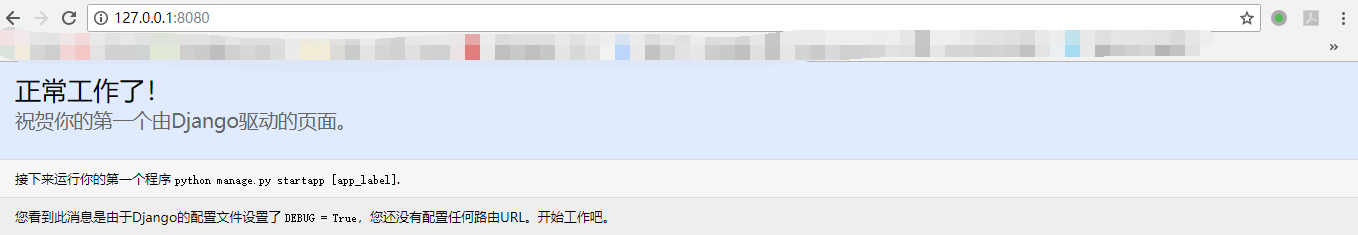
1. 对myproject下settings.py的设置
2. 语言设置：可以为中文，也可以为英文原版不用修改，根据自己的需求配置



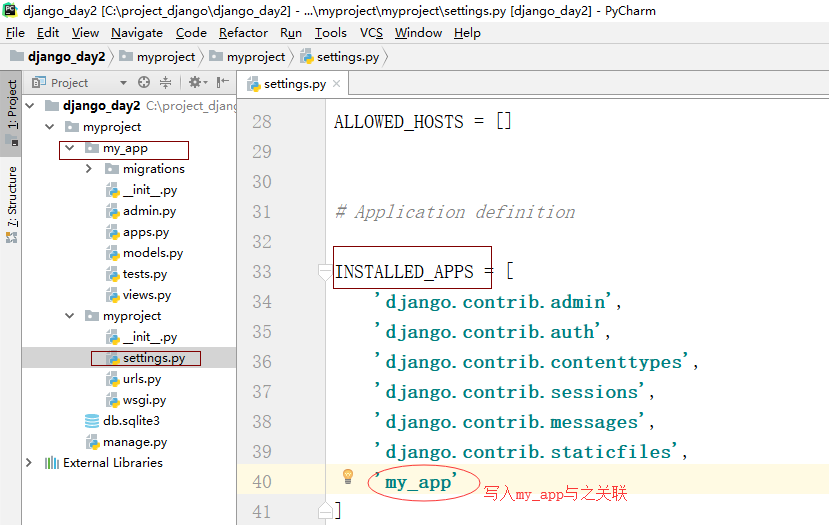
1. 时区设置：UTC为世界标准时，北京时间为UTC+8（上海时间和北京时间同属于东8区）



1. 可以正常的显示汉语了，说明修改成功，第一个django项目就大功告成了



1. 对settings.py文件中的INSTALLED\_APPS中写入创建的app名称---



也可以通过my\_app中的apps.py模块中的AppConfig下的name属性

三、django项目中模型的使用

Django自身集成了管理后台，再管理后台中可以对我们自定义的model进行增、删、查和改等操作，也可以进行列表展示的解析，分页等

1. 在models.py文件中中定义（创建）模型 class 模型名称 它继承 models.Model

db\_tables:定义数据库中的表名称

1. 迁移 以下两个命令配套使用---生成迁移文件，将文件迁移到数据库中

python manage.py makemigrations

python manage.py migrate

1. 保存数据

stu = Student()

stu.name = ‘ ‘

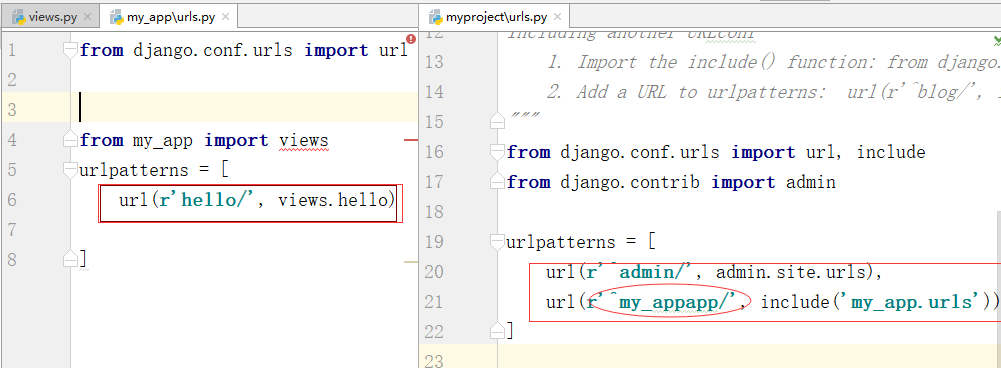
stu.save()

1. 对urls.py模块的使用

先在自己的my\_app项目中自己新建一个urls.py，写一个ulrpatterns模型，

写url时，需要导入**from** django.conf.urls **import** url，与myproject工程目录中的要对应，但是需要在my\_app工程目录下的urlpattern中导入url(**r'^my\_appapp/'**, include(**'my\_app.urls'**))， 于此同时，也需要将**from** django.conf.urls **import** url, include导入；

日后定义同样类似于my\_app的app时，与此处的步骤相同。



1. 创建（定义一个方法）

**def** hello(request):  
 **return** HttpResponse()

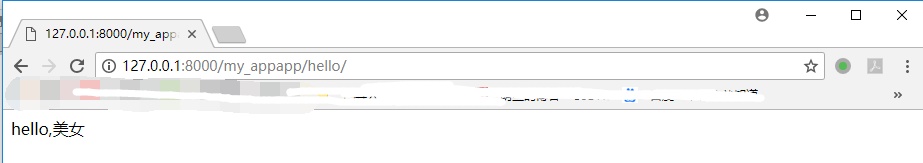
这里需要导入 **from** django.http **import** HttpResponse

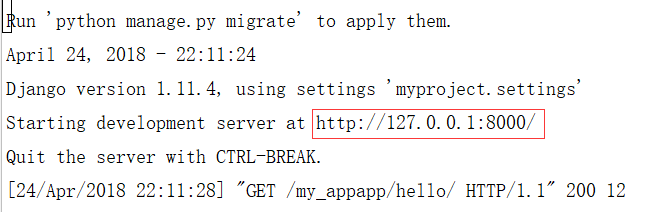
1. 启动项目

Python manage.py runserver ----- 产生http://127.0.0.1:8000/

在浏览器中输入该地址---http://127.0.0.1:8000/my\_appapp/hello/

显示如下，表示项目已经走通了





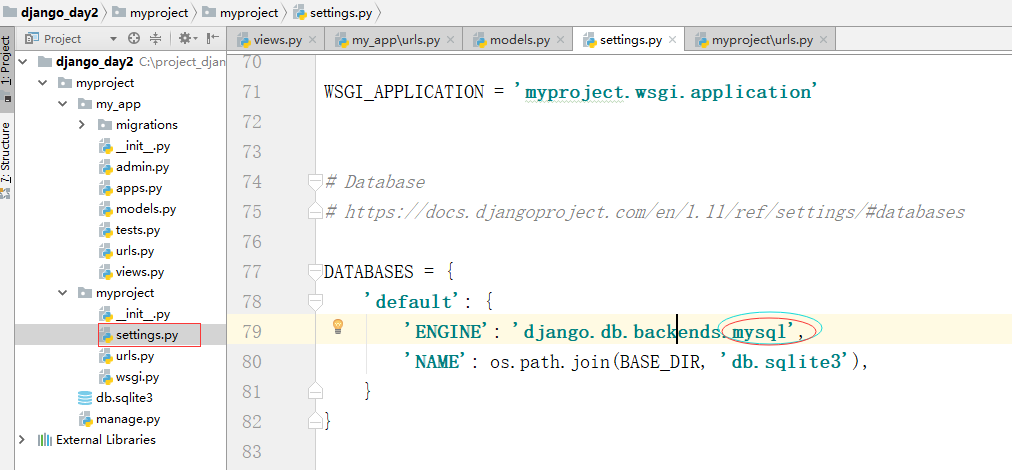
1. 在models.py中创建模型

相当于定义了一个类，要与数据库关联

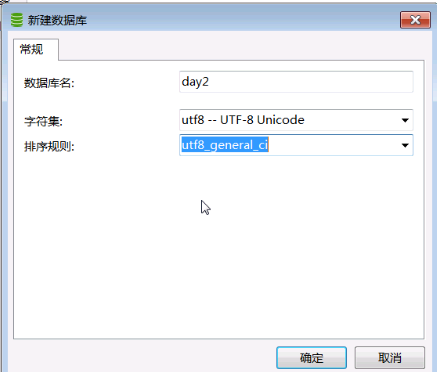
**class** Student(models.Model):  
 name = models.CharField(max\_length=10)  
 sex = models.BooleanField()

1. 在settings.py中修改DATABASES

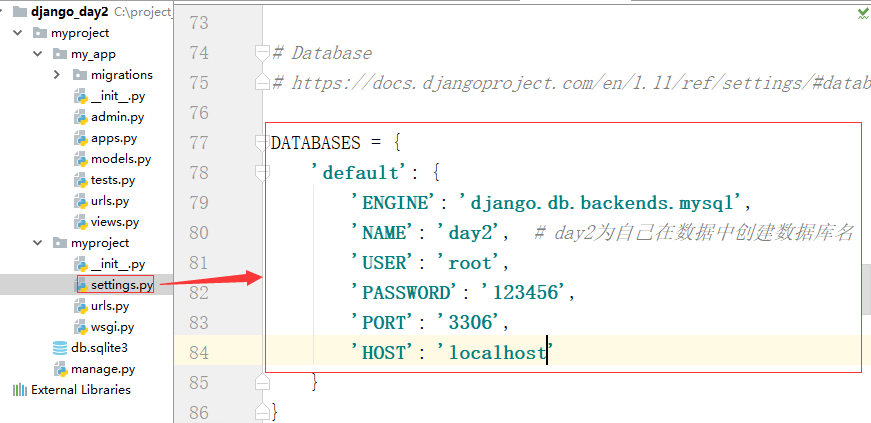
将原来的**sqlite3，替换为mysql**



在数据库中创建数据库的时候，要指定字符集，这里使用Navicat

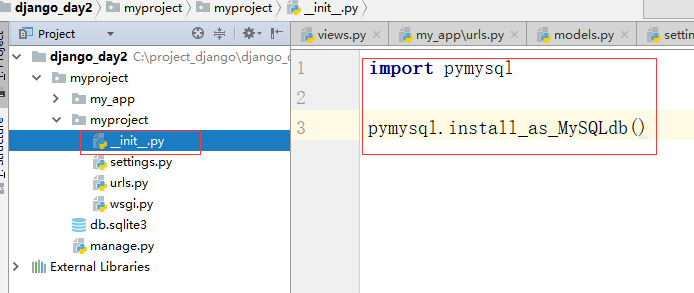


给DATABASES中定义相关信息



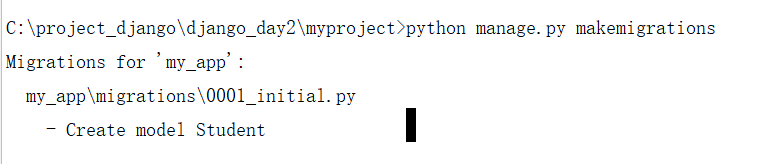
1. 初始化pymysql

在myproject中，初始化\_\_init\_\_.py

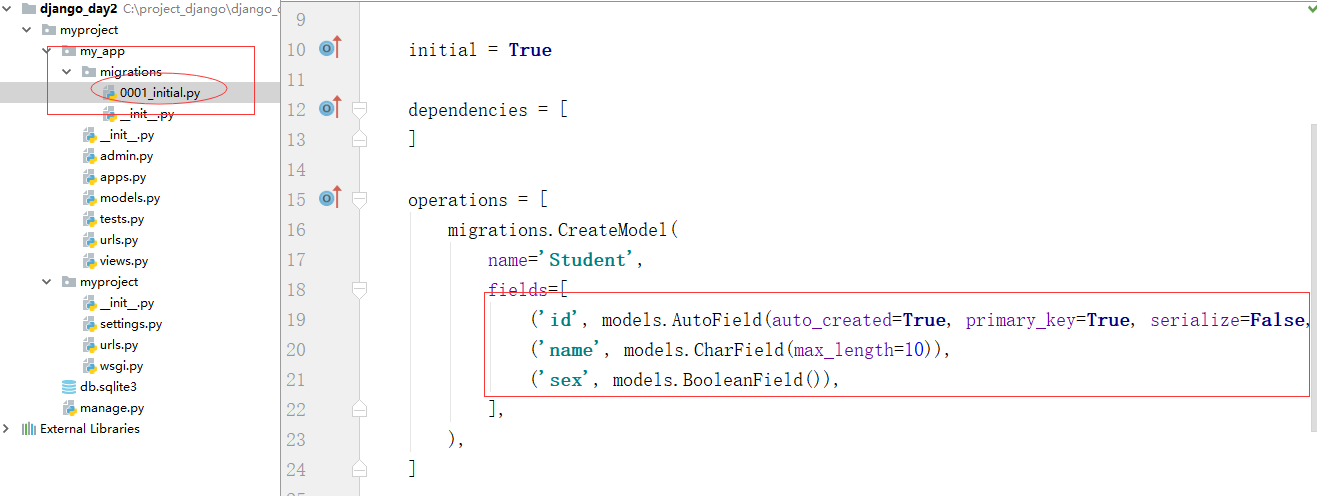


1. 数据的迁移

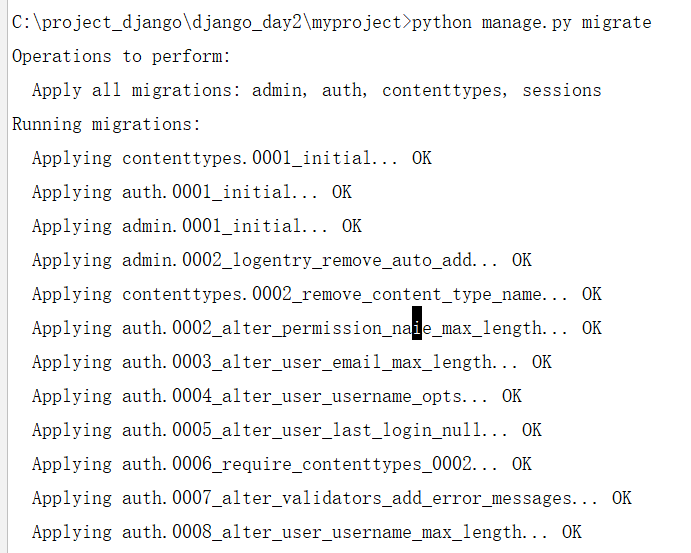
数据的执行----执行命令--- python manage.py makemigrations，



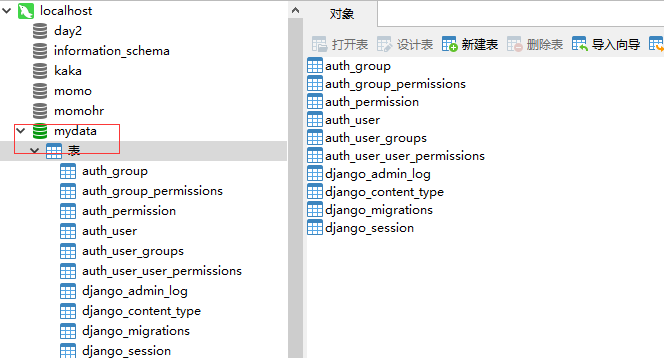
创建了一个0001\_initial.py的py文件



数据的应用---- python manage.py migrate



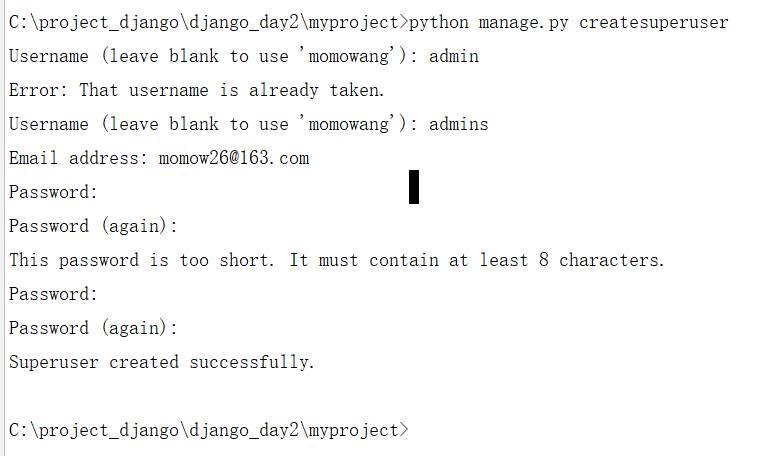
查看迁移---关联后的数据包



1. 创建超级管理员账号密码

Python manage.py createsuperuser

Momo wangmomo123



修改密码 manage.py changepassword

1. 使用账户密码登录



1. 对刚刚迁移的数据进行增删查改
2. admin注册