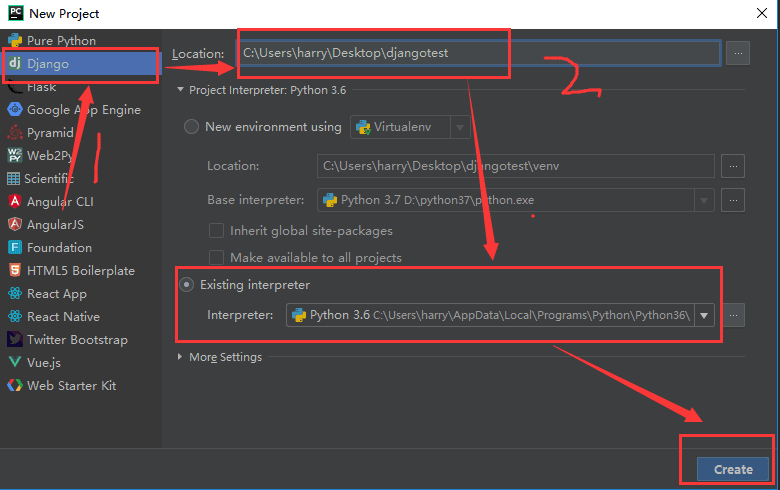
1.14 web (Django 框架学习)

1.创建django工程（pycharm专业版创建，社区版不行）：



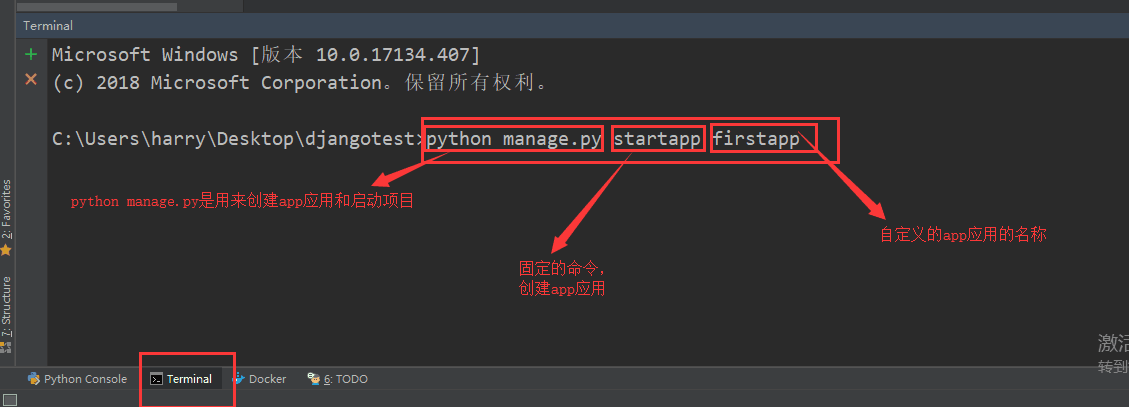
命令创建工程：<https://blog.csdn.net/yinjly/article/details/81475656>

（1）.打开pycharm   File--->New Project；

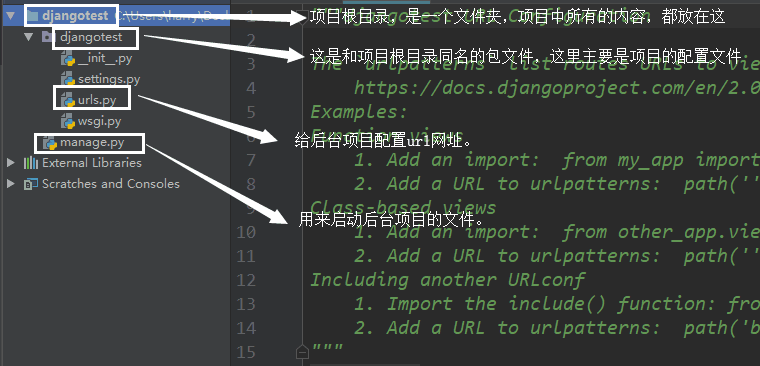
（2）.使用命令  **django-admin startproject 项目名**    新建Django项目；

执行命令后 也会出现如下‘框架内容’所示；

（3）.命令创建app：

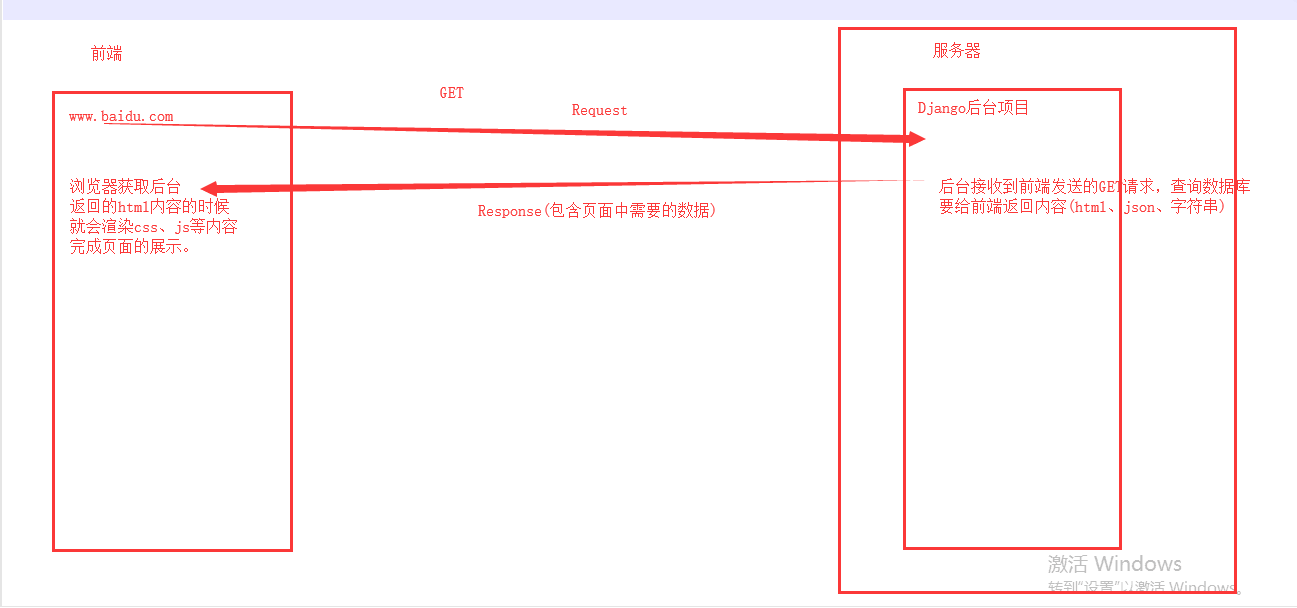


2.框架内容：



以上项目只有一些配置信息，要完成后台的项目还需要创建app

3.前端与后台的联系：



下午：

1. Bs4 和lxml讲解
   1. AM

web (Django 框架学习)

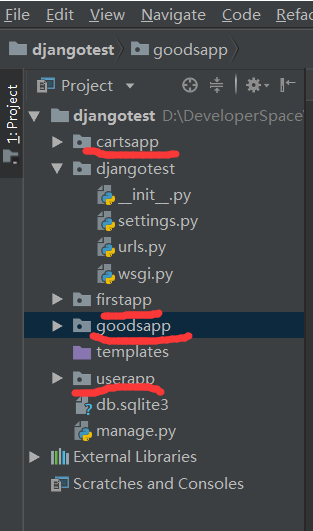
Django项目和app应用之间的关系：

一个django项目可以管理多个app应用；

一个项目可以划分为多个模块：比如用户模块（登陆注册、用户名检测、密码的找回等功能），那么就可以给用户模块新建一个app应用叫userapp，所有功能都可以写在这个userapp中。商品模块（首页商品、列表商品、详情页商品、热门商品、商品搜索等功能），此时再新建一个app应用叫goodsapp，所有关于商品的功能都可以写在这个goodsapp中。

为什么要创建多个app应用：

1. 让项目结构更加清晰，后期维护起来比较方便。
2. 方便代码的编写，多人协作开发更方便快捷。

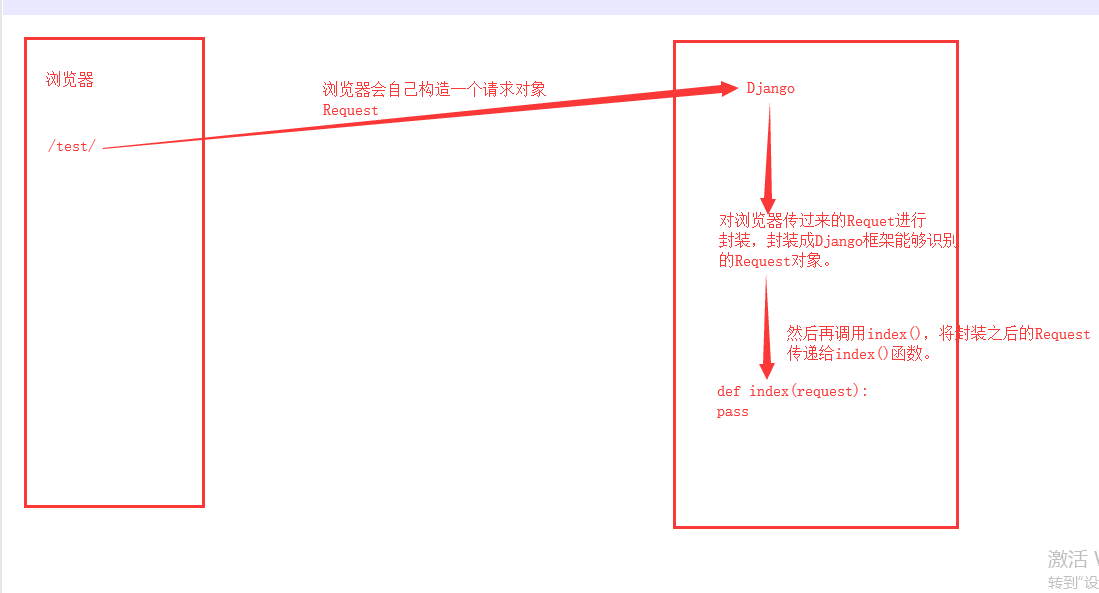


App主要模块介绍：

Views.py:视图模块，处理Django接收过来的Request请求，返回Response响应；该模块中的函数都称为视图函数。

Views中定义视图函数（如：index()包含返回内容）🡪 urls模块绑定url和 视图函数，path(“路由/”，视图函数) 🡪 在settings中注册 🡪 运行整个工程，访问url（http://127.0.0.1:8000/myfirst/），获取数据。





下午：djangoproject 下app的介绍。

创建新项目：djangotemplate,新建app:temapp

1.设置相关配置信息：

(1)settings；每次新创建的app都要在installed\_apps下注册；

(2)views；

(3)urls；

2.所有的模板（即html页面）都放在templates文件夹下；

3.新建index.html

4.写views页面

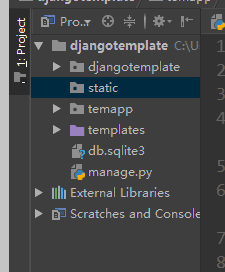
5.写urls路由

6.启动项目，访问路由，获取html页面。

7.渲染模板（index.html）

8.在项目根目录下创建文件夹static（必须以static命名）

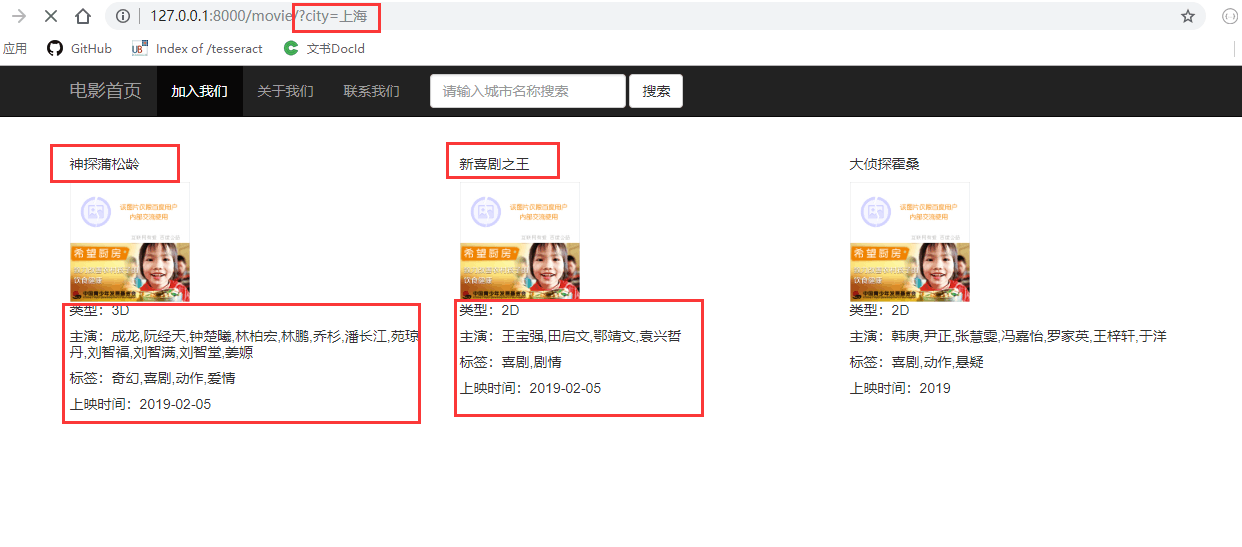
学习在模板文件中，使用.css/.js文件的方法；

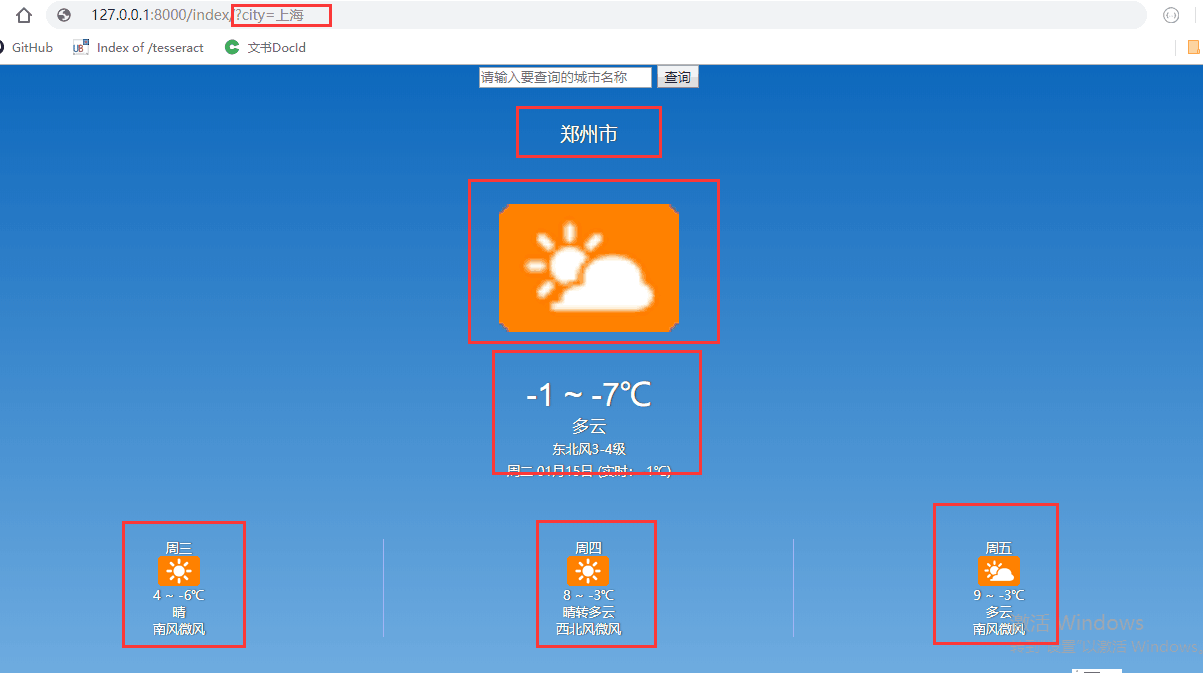


在static文件夹中创建index.css文件；设置大小和颜色；

注意：需要在settings中配置static 文件夹的搜索路径；否则将报无法访问的错误；

作业：





1.16作业解答

1.创建项目djangotest,新建app：python manage.py startapp weatherapp,和

python manage.py startapp movieapp。

2.url分离：在每个app中分别创建各自的urls.py，以放置各自的url路由；

项目urls.py用来放置每个app的总路由。

3.在settings中注册app;

4.在根目录下新建static文件夹，并把该文件夹路径添加到settings中；

STATIC\_URL = '/static/'  
  
# 设置 "static" 文件夹的搜索路径，如果不配置，django是不知道static文件夹的。  
STATICFILES\_DIRS = [os.path.join(BASE\_DIR, 'static')]

5.分别把所需要的html文件拖入template文件夹，css文件拖入static文件夹；

6.在weather.html中引入路径：

<link rel=”stylesheet” href=”/static/weather.css”>

7.在weatherapp.views中新建index函数并添加路由；

函数中包含ip定位接口和天气接口；

city = request.GET.get(‘city’,’郑州市’)；参数为city，缺省值为郑州市；

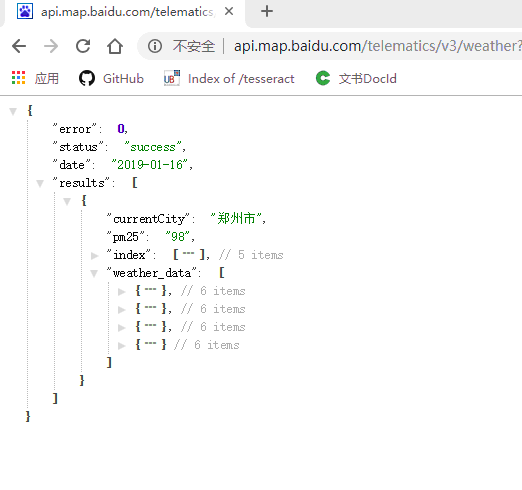
由接口获取url = 。。。；

将返回对象转换为字符串：json\_str = request.get(url).text;

再转换为字典：weather\_dict = …;

取字典元素转为列表：weather\_data = …;

解析网页：



当天天气：

current\_data\_dict = weather\_data[0]

未来三天天气：

future\_data = …;

城市名称：

future\_data\_city = …;

将future\_data和current\_data\_dict,future\_data\_city合为一个字典future\_data\_dict;

再将字典渲染到模板中：

render(request,template\_name=’weather.html’,context=future\_data\_dict);

8.改写html网页；将原html中的值全部替换为所提取的字典中的键；

9.运行项目，完成

访问url为：<http://127.0.0.1:8000/weather/index/?city=XXX>（默认为郑州市）

movieapp

1.设置app/urls路由；

2.写views中的index函数；步骤与weather类似；

3.之后也类似；

4.图片和视频防盗链；https://blog.csdn.net/fRF0lw4/article/details/80105574   防盗链解决方案 ；添加语句：referrerPolicy=”no-referrer”。

C:\Users\WJY\AppData\Local\Temp\%W@GJ$ACOF(TYDYECOKVDYB.pnghttps://www.cnblogs.com/lydiawork/p/7838153.html  防盗链原理 。

form表单学习：

1.创建djangoform工程；formapp;进行相关配置；

2.index.html编辑；实现页面效果：



3.views视图函数;再定义函数sum(),接收表单数据，实现加法运算；

4.在html中绑定action动作指定的函数sum，和获取数据的方法“get”；

name属性的值作为views中sum函数的参数；

下午：

1.method = ‘POST’方法讲解

post请求时，html必须加{% csrf\_token %}语句；若不加，则django浏览器会拒绝浏览器的请求；

以上是如何用表单进行get/post请求，以及后台如何获取数据；

2.数据库操作

mysql;navicat;

在数据库mysql80中创建数据库mysqldb；

在settings中配置数据库连接；

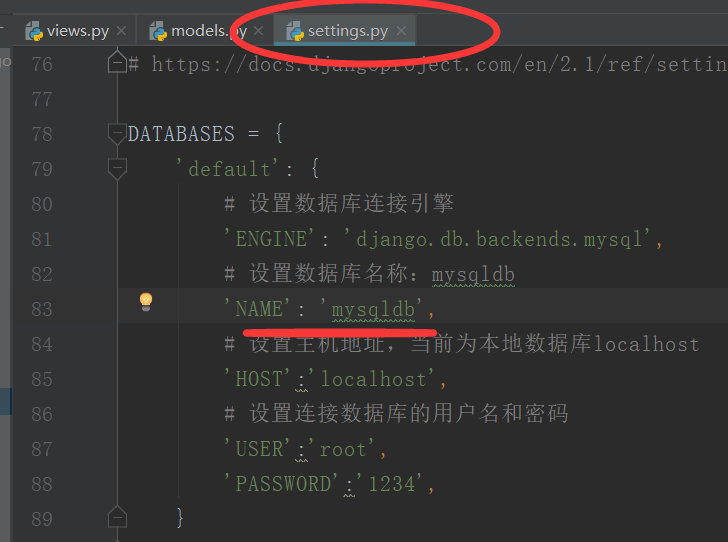
编写models.py：创建数据库模型的文件；Django后台框架对数据的增删改查操作不是通过sql语句实现的，而是通过ORM的形式实现的。

ORM：对象关系映射；

ORM的优势：1.写法简单，抛弃复杂的sql语句；2.放置sql注入。

在models.py中通过类和对象的形式实现对数据库的操作：

在mysqldb中创建people表的方法：

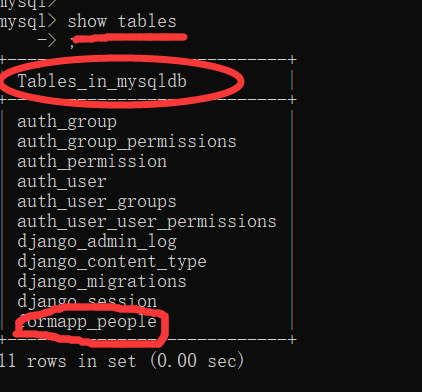


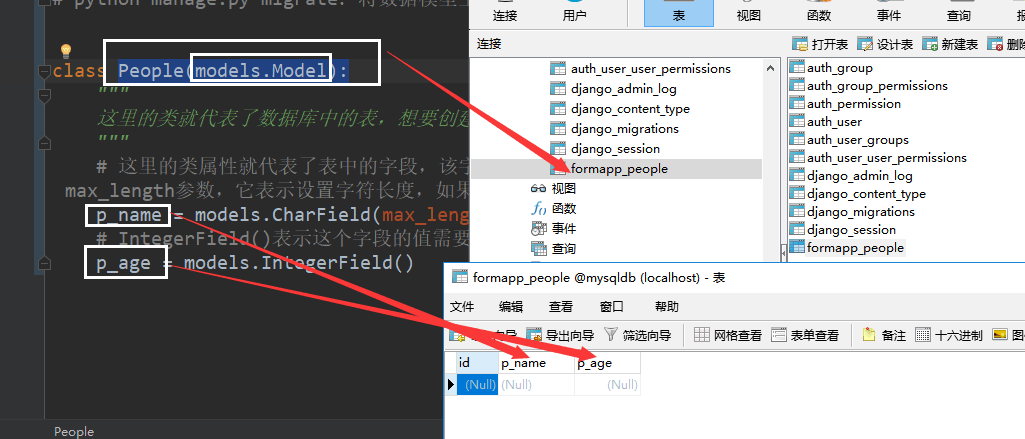


完成以上设置后：

# 第一步：在终端中执行命令：python manage.py makemigrations 创建数据模型；收集models.py文件中数据模型的变化，若文件内容没有任何变化，会提示：NO changes detected;  
# 第二步：在终端中执行命令：python manage.py migrate；数据模型生成数据库中的表，插入mysqldb；

查询数据库的表：





可以看到创建成功！

对数据库的增删改查也是通过对People类的操作来完成：

在views.py中定义增删改查函数实现（需要配置路由）；

需要将每个功能分别封装为函数，设置路由；

作业：

练习form标签和ORM的基本操作；

1.17 模板的继承（伯乐在线：blog.jobbole.com）

模板继承：将页面中不变的部分(即多个页面公用的html代码)单独封装为一个html；



1.创建新项目：djangoteminherit; 新的app:inheritapp;

2.在template中新建三个html文件：父模板base.html(只是用来继承，不用渲染)和两个子模版；

3.base.html中，block块和extent块作为子模版的预留，子模版只能修改父模板的预留块；

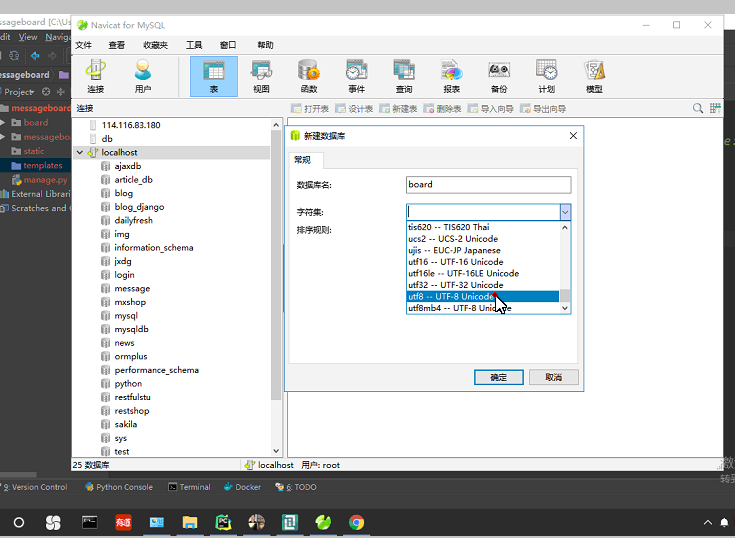
4.在views中定义函数index1()。。。

5.完成各个路由配置。

利用模板继承完成留言版

1.新建项目messageboard; app:board;在根目录下新建文件夹static；

在mysql80中新建数据库board;编码utf-8；排序规则：utf-8 general\_ci



在settings中配置数据库；配置static静态文件的搜索目录；

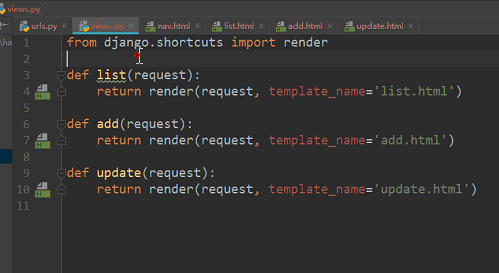
2.将“留言板”文件夹中已给的add/nav/update.html放入template;

.js和.css文件放入static

修改update.html和add.html和nav.html中的相关连接路径；

新建列表页list.html实现导航信息；

3.在views中定义功能函数：



配置url路由；

运行。

然后用页面继承修改相关页面。

页面继承时，不允许在一个页面中写两个{% extends ‘ ‘ %}语句；