# 创建虚拟环境

* 安装虚拟环境命令:

pip install virtualenv

pip install virtualenvwrapper

* 创建名为Dproject的虚拟环境

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01>**virtualenv.exe Dproject**

Using base prefix 'c:\\users\\lucifer\\appdata\\local\\programs\\python\\python36'

New python executable in C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Scripts\python.exe

Installing setuptools, pip, wheel...done.

* 进入虚拟环境目录

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01>**cd Dproject**

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject>**cd Scripts**

C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Scripts>**activate**

* 从命令行前部已经可以看到进入了虚拟环境

**(Dproject)** C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Scripts>

* 虚拟环境中并没有Django ，需要安装Django

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Scripts>**pip list**

DEPRECATION: The default format will switch to columns in the future. You can use --format=(legacy|columns) (or define a format=(legacy|columns) in your pip.conf under the [list] section) to disable this warning.

pip (9.0.1)

setuptools (28.8.0)

SQLAlchemy (1.1.13)

wheel (0.30.0)

* 在虚拟环境安装Django

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Scripts>**pip install Django==1.11.5**

Collecting Django==1.11.5

Using cached Django-1.11.5-py2.py3-none-any.whl

Collecting pytz (from Django==1.11.5)

Using cached pytz-2017.2-py2.py3-none-any.whl

Installing collected packages: pytz, Django

Successfully installed Django-1.11.5 pytz-2017.2

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Scripts>python

Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

>>> import django

>>> django.VERSION

(1, 11, 5, 'final', 0)

# 创建Django项目

* 以下内容是在windows下的操作，如在linux下应该更简单一些

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject>**python Scripts/django-admin.py startproject Dproject**

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject>**cd Dproject**

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**tree /F**

卷 Windows8\_OS 的文件夹 PATH 列表

卷序列号为 0000004A 1C6A:8177

C:.

│ manage.py

│

└─Dproject

settings.py

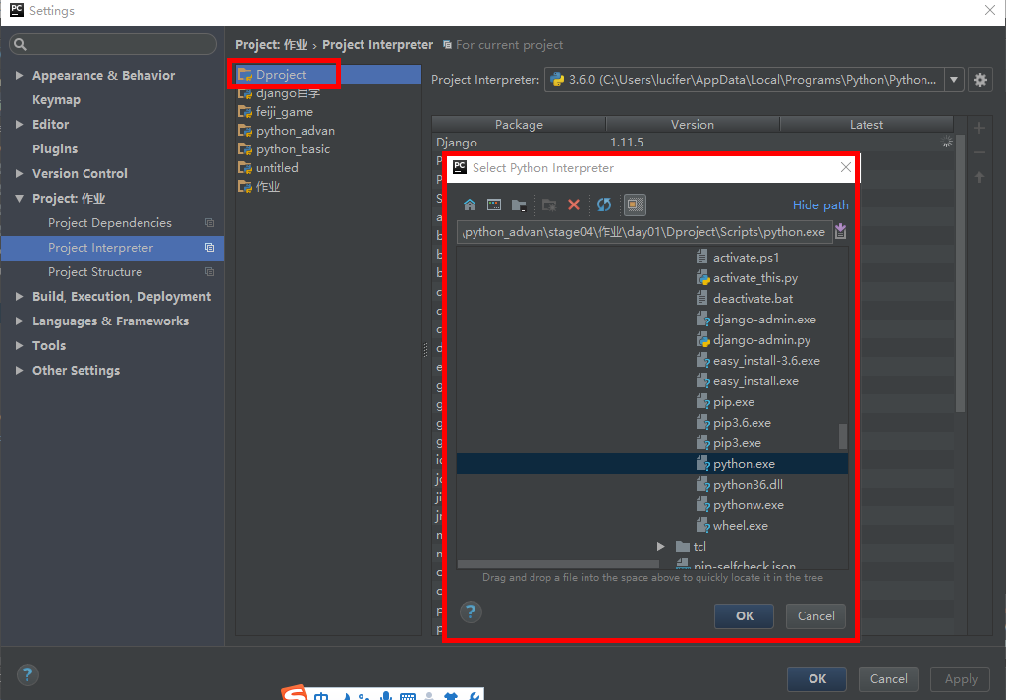
urls.py

wsgi.py

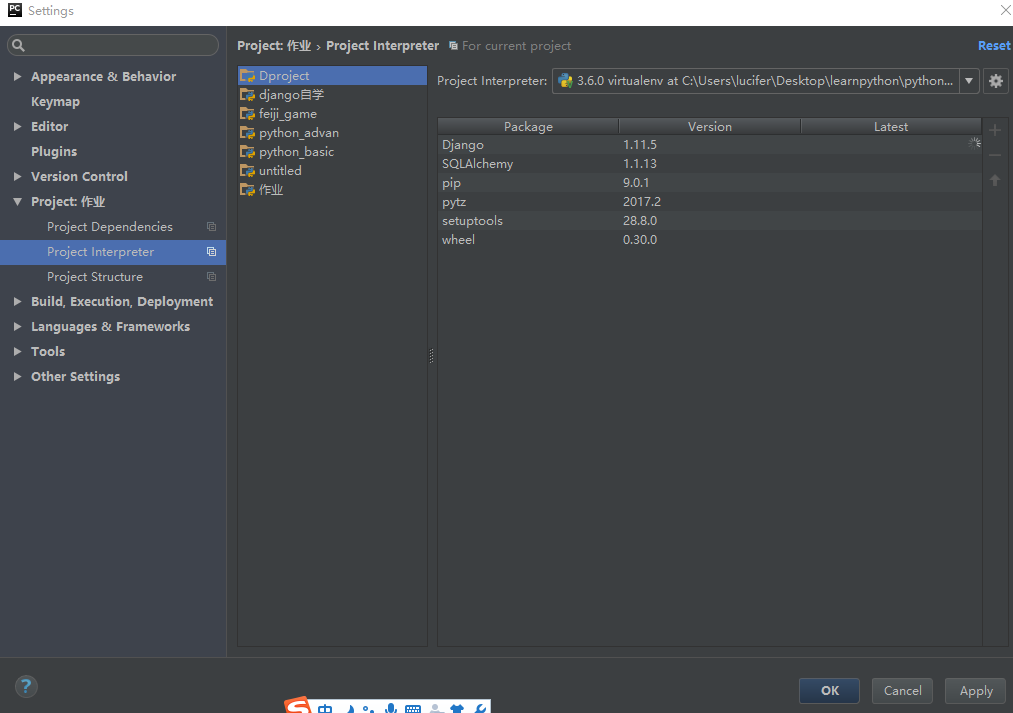
\_\_init\_\_.py

# 在pycharm中打开项目并且调整使用的环境

* 在pycharm的设置中，选中项目和新建的虚拟环境按OK



选择虚拟环境后就可以有多个独立的环境了



# 新建应用

* 新建一个叫 news的新闻应用

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py startapp news**

以下是建立后的目录结构

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>tree /F

卷 Windows8\_OS 的文件夹 PATH 列表

卷序列号为 000000A5 1C6A:8177

C:.

│ manage.py

│

├─.idea

│ │ Dproject.iml

│ │ misc.xml

│ │ modules.xml

│ │ workspace.xml

│ │

│ └─inspectionProfiles

│ profiles\_settings.xml

│

├─Dproject

│ │ settings.py

│ │ urls.py

│ │ wsgi.py

│ │ \_\_init\_\_.py

│ │

│ └─\_\_pycache\_\_

│ settings.cpython-36.pyc

│ \_\_init\_\_.cpython-36.pyc

│

└─**news**

**│ admin.py**

**│ apps.py**

**│ models.py**

**│ tests.py**

**│ views.py**

**│ \_\_init\_\_.py**

**│**

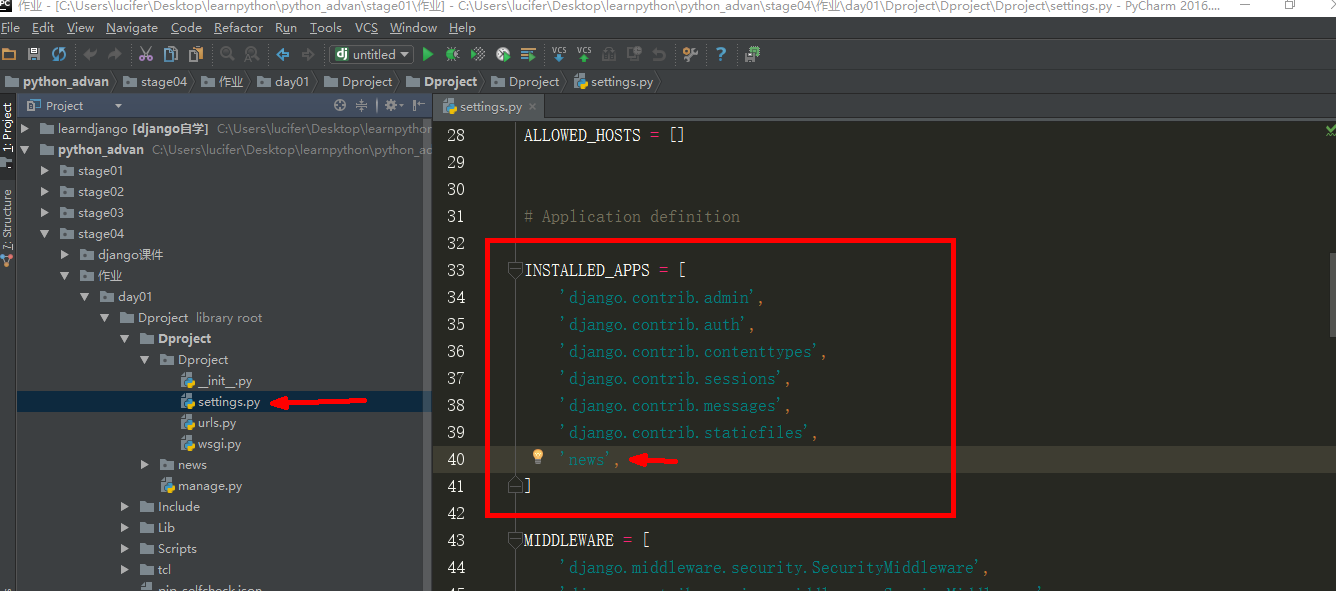
**└─migrations**

**\_\_init\_\_.py**

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>

* 注册应用

在项目的settings.py文件下 注册新建的news应用



* 确认Django服务已经启动，以下红字部分为启动命令 参数可指定IP和端口

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py runserver localhost:8888**

Performing system checks...

System check identified no issues (0 silenced).

You have 13 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.

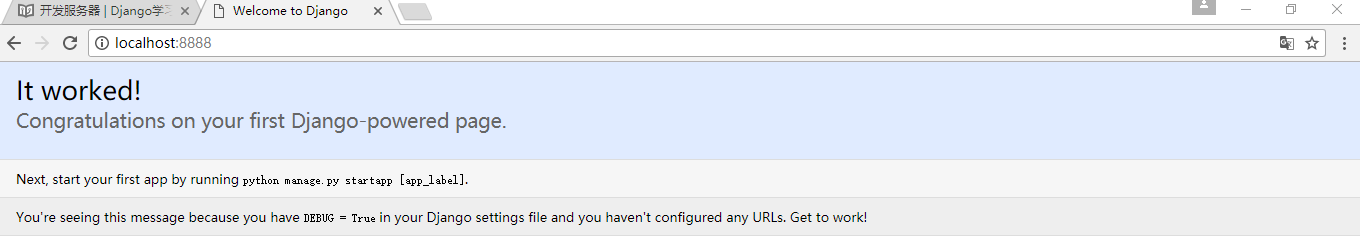
October 06, 2017 - 15:23:01

Django version 1.11.5, using settings 'Dproject.settings'

Starting development server at http://localhost:8888/

Quit the server with CTRL-BREAK.

* 通过浏览器访问确认Django web服务器工作正常



* 按ctrl + c 停止服务进行下一步操作

# 在Django中使用ORM定义模型类

* 在django中进行数据库开发，步骤如下:
* 在models.py模块中定义模型类;
* 根据模型类生成对应SQL, 这一步我们叫做’迁移‘;
* 执行迁移, 这一步我们执行第二步生成的SQL;
* 通过模型类方法完成对数据库的CRUD操作.
* 下面我们创建模型类:

新闻分类表：

表名：NewsCategory

新闻分类：cag\_name

新闻信息表:

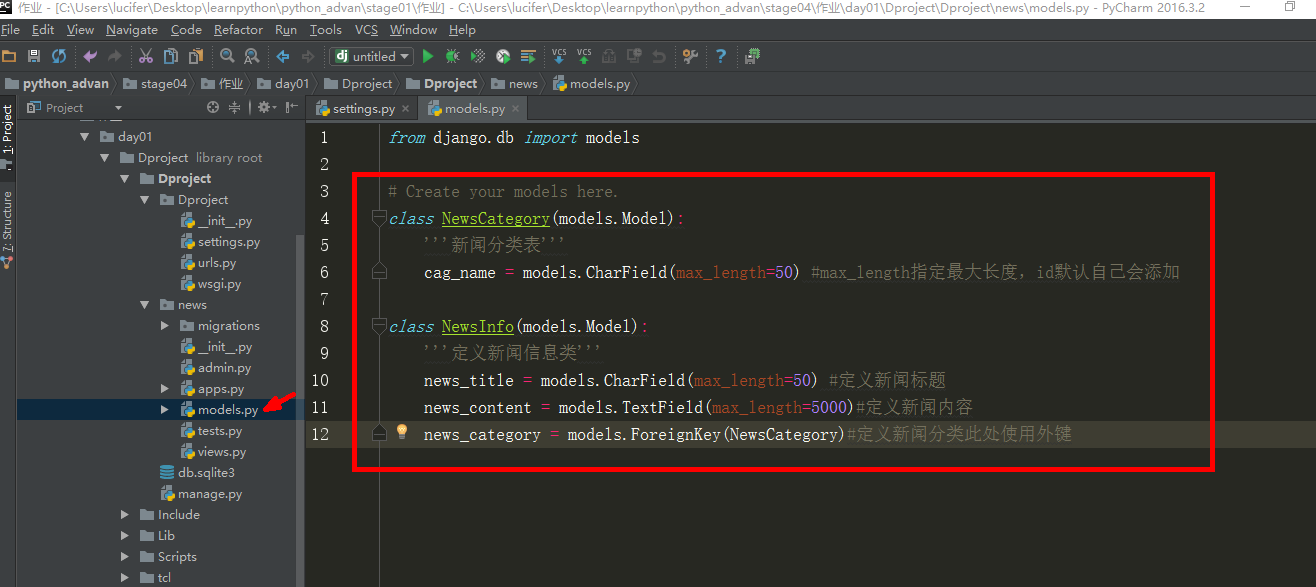
表名：NewsInfo

新闻标题：news\_title

新闻内容：news\_contents

所属分类：news\_cag

* 这里需要注意，新闻分类表中的数据和新闻信息表中的数据关系是一对多关系，也就是一个新闻分类对应多个新闻信息.



* 具体代码如下

*from* django.db *import* models  
  
# Create your models here.  
*class* NewsCategory(models.Model):  
 '''新闻分类表'''  
 cag\_name = models.CharField(max\_length=50) #max\_length指定最大长度，id默认自己会添加  
  
*class* NewsInfo(models.Model):  
 '''定义新闻信息类'''  
 news\_title = models.CharField(max\_length=50) #定义新闻标题  
 news\_content = models.TextField(max\_length=5000)#定义新闻内容  
 news\_category = models.ForeignKey(NewsCategory)#定义新闻分类此处使用外键

* 生成迁移文件

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py makemigrations**

Migrations for 'news':

news\migrations\0001\_initial.py

- Create model NewsCategory

- Create model NewsInfo

* 查看目录可以看到多了0001\_initial.py 文件

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>tree /F

卷 Windows8\_OS 的文件夹 PATH 列表

卷序列号为 00000062 1C6A:8177

C:.

│ db.sqlite3

│ manage.py

│

├─.idea

│ │ Dproject.iml

│ │ misc.xml

│ │ modules.xml

│ │ workspace.xml

│ │

│ └─inspectionProfiles

│ profiles\_settings.xml

│

├─Dproject

│ │ settings.py

│ │ urls.py

│ │ wsgi.py

│ │ \_\_init\_\_.py

│ │

│ └─\_\_pycache\_\_

│ settings.cpython-36.pyc

│ urls.cpython-36.pyc

│ wsgi.cpython-36.pyc

│ \_\_init\_\_.cpython-36.pyc

│

└─news

│ admin.py

│ apps.py

│ models.py

│ tests.py

│ views.py

│ \_\_init\_\_.py

│

├─migrations

│ │ 0001\_initial.py

│ │ \_\_init\_\_.py

│ │

│ └─\_\_pycache\_\_

│ \_\_init\_\_.cpython-36.pyc

│

└─\_\_pycache\_\_

admin.cpython-36.pyc

models.cpython-36.pyc

\_\_init\_\_.cpython-36.pyc

* 查看迁移文件生成的SQL语句 python manage.py sqlmigrate 应用名 文件名

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py sqlmigrate news 0001**

BEGIN;

--

-- Create model NewsCategory

--

CREATE TABLE "news\_newscategory" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "cag\_name" varchar(50) NOT NULL);

--

-- Create model NewsInfo

--

CREATE TABLE "news\_newsinfo" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "news\_title" varchar(50) NOT NULL, "news\_content" text NOT NULL, "news\_category\_id" integer NOT NULL REFERENCES "news\_newscategory" ("id"));

CREATE INDEX "news\_newsinfo\_news\_category\_id\_c18d8327" ON "news\_newsinfo" ("news\_category\_id");

COMMIT;

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>

* 执行迁移

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py migrate**

Operations to perform:

Apply all migrations: admin, auth, contenttypes, news, 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

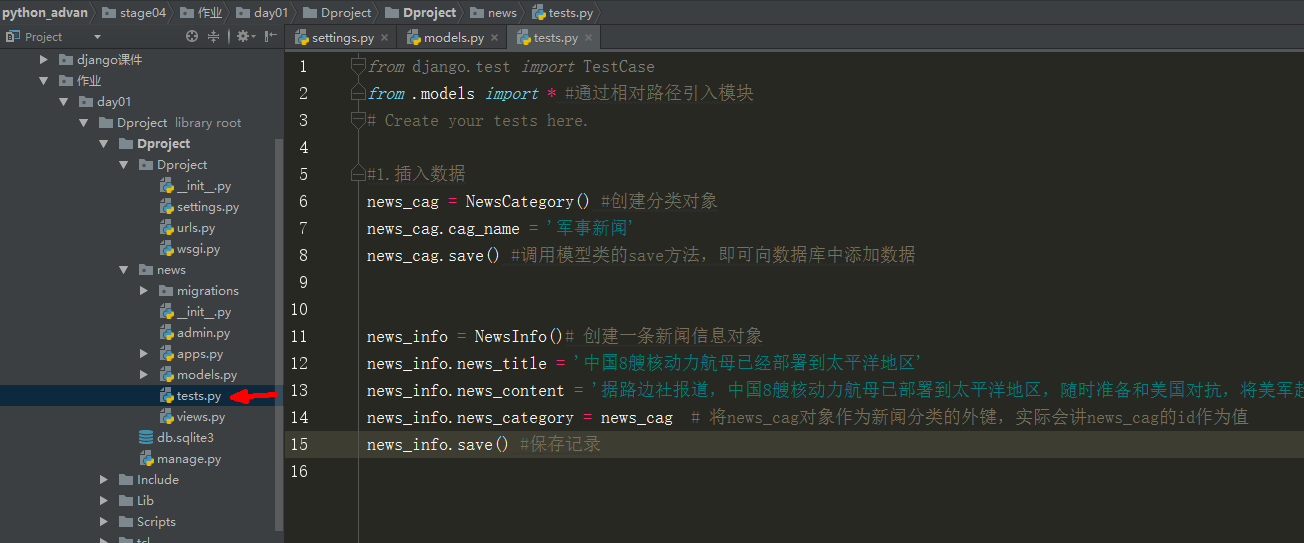
Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying news.0001\_initial... OK

Applying sessions.0001\_initial... OK

# 测试数据

* 测试数据可以通过2种方法 使用
* python manage.py shell在终端测试，
* 也可以在应用下的 test.py中进行测试
* 以下先在 test.py文件中进行测试。然后再通过shell进行测试



具体代码如下：

*from* django.test *import* TestCase  
*from* .models *import* \* #通过相对路径引入模块  
# Create your tests here.  
  
#1.插入数据  
news\_cag = NewsCategory() #创建分类对象  
news\_cag.cag\_name = '军事新闻'  
news\_cag.save() #调用模型类的save方法，即可向数据库中添加数据  
  
  
news\_info = NewsInfo()# 创建一条新闻信息对象  
news\_info.news\_title = '中国8艘核动力航母已经部署到太平洋地区'  
news\_info.news\_content = '据路边社报道，中国8艘核动力航母已部署到太平洋地区，随时准备和美国对抗，将美军赶出亚太地区'  
news\_info.news\_category = news\_cag # 将news\_cag对象作为新闻分类的外键，实际会讲news\_cag的id作为值  
news\_info.save() #保存记录

* 在命令中执行测试代码命令

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py test news**

Creating test database for alias 'default'...

System check identified no issues (0 silenced).

----------------------------------------------------------------------

Ran 0 tests in 0.000s

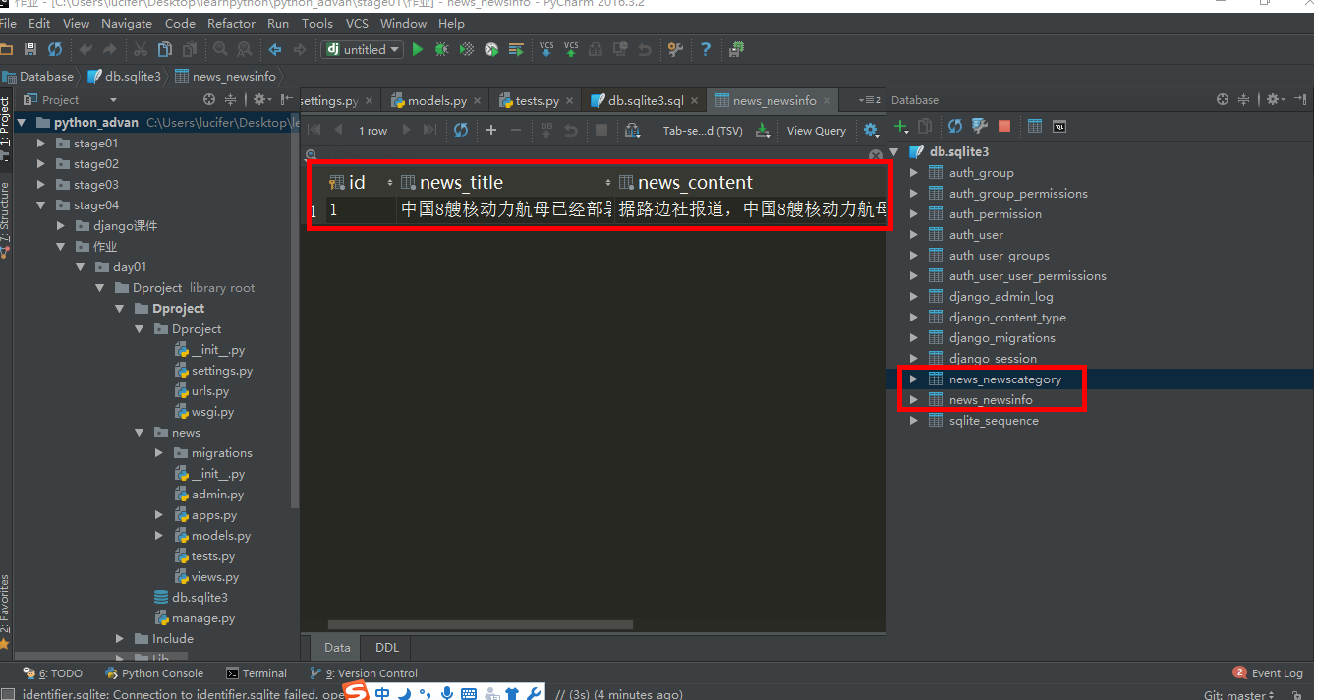
OK

Destroying test database for alias 'default'...

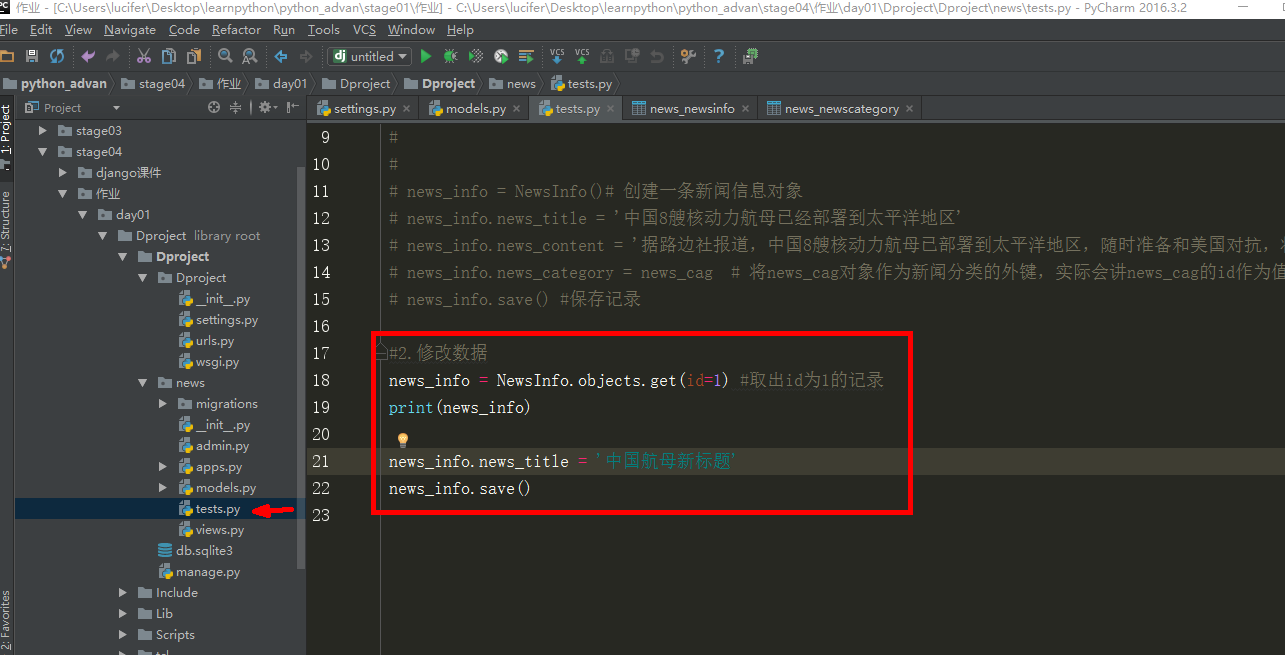
此时数据已插入sqlite数据库，可以通过软件DB Browser for sqlite查看数据

也可以在pycharm集成环境中查看数据，这次试用pycharm在集成环境中查看数据

可以看到2张数据表已经建立吗，并且有数据已经插入。



* 修改数据



* 以下为源代码

#2.修改数据  
news\_info = NewsInfo.objects.get(id=1) #取出id为1的记录  
print(news\_info) #打印该对象，可以定义一个\_\_str\_\_方法改变显示  
  
news\_info.news\_title = '中国航母新标题' #设置新标题  
news\_info.save() #保存修改

* 通过命令行执行修改

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py test news**

**NewsInfo object**

Creating test database for alias 'default'...

System check identified no issues (0 silenced).

----------------------------------------------------------------------

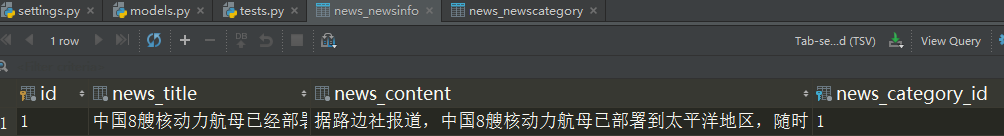
Ran 0 tests in 0.000s

OK

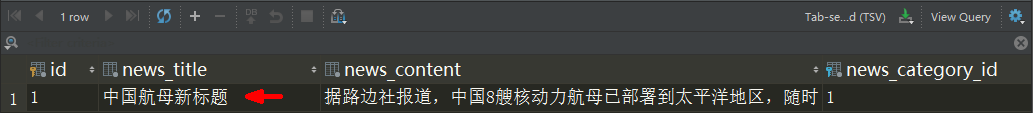
Destroying test database for alias 'default'...

可以看到表的内容改变了

原title



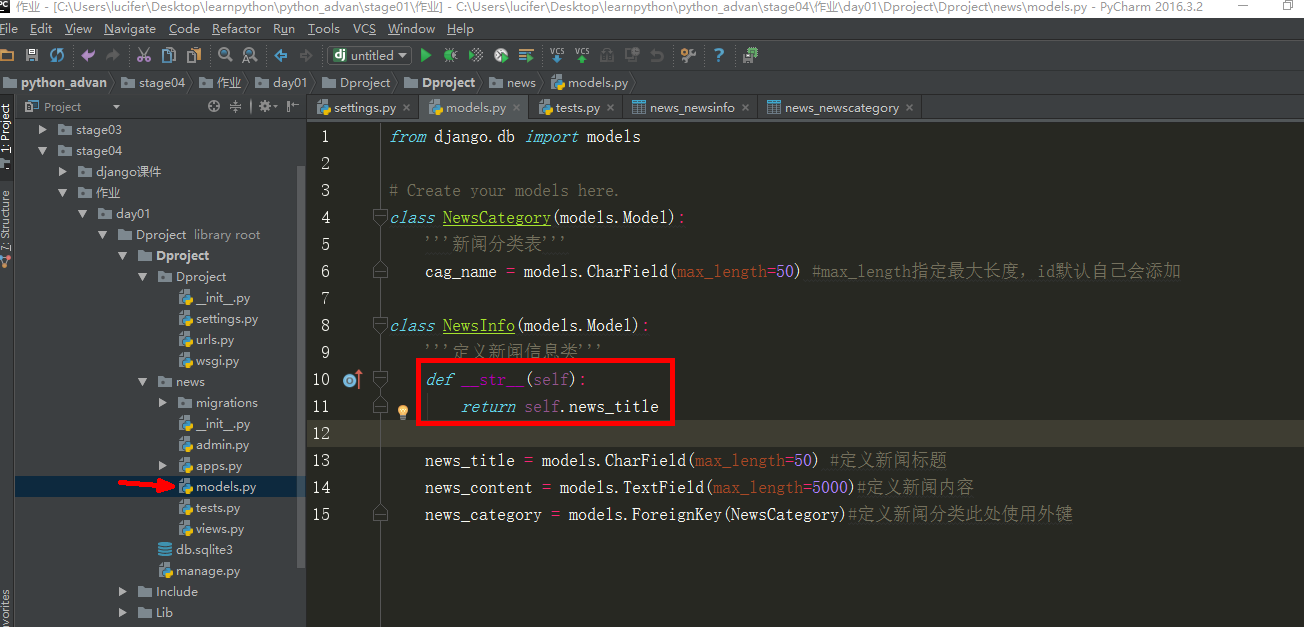
现title



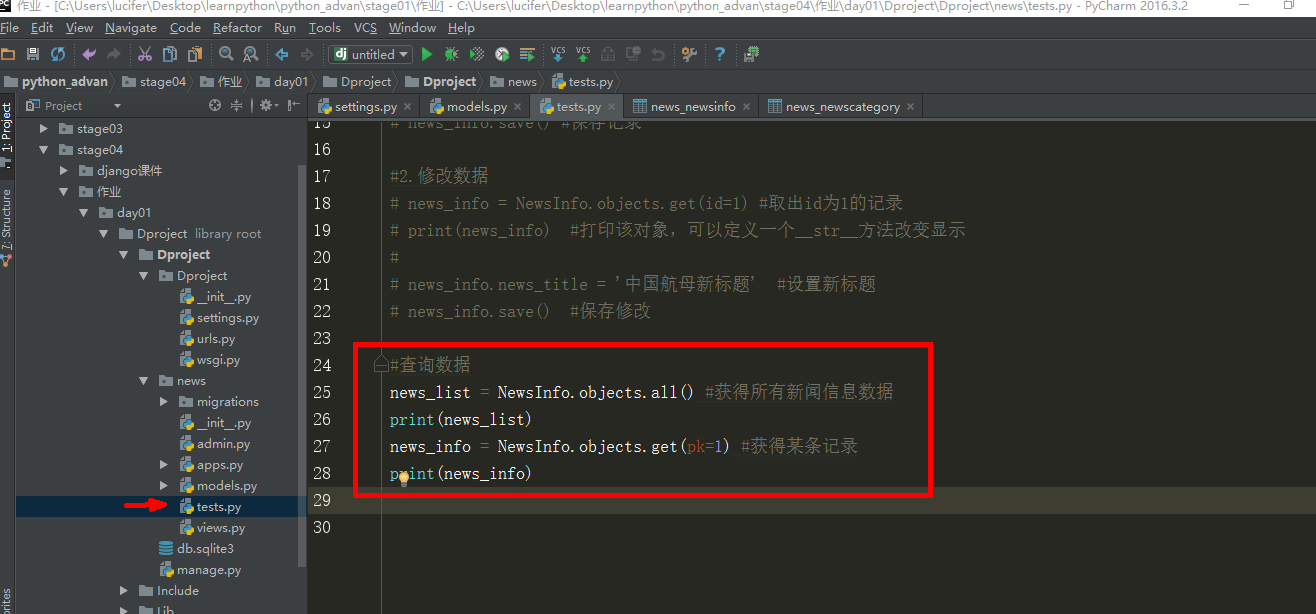
* 查询数据

为了方便查看数据，在models相应的类中增加一个\_\_str\_\_方法，

改方法返回了新闻的标题



回到test.py 测试文件 增加查询的内容



* 源代码如下： 代码略做修改为了更好显示内容

#查询数据  
news\_list = NewsInfo.objects.all() #获得所有新闻信息数据  
print('获取所有新闻信息数据:'+ str(news\_list))  
news\_info = NewsInfo.objects.get(pk=1) #获得某条记录  
print('获得一条新闻数据:'+ str(news\_info))

* 在命令行运行命令执行查询

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py test news**

**获取所有新闻信息数据:<QuerySet [<NewsInfo: 中国航母新标题>]>**

**获得一条新闻数据:中国航母新标题**

Creating test database for alias 'default'...

System check identified no issues (0 silenced).

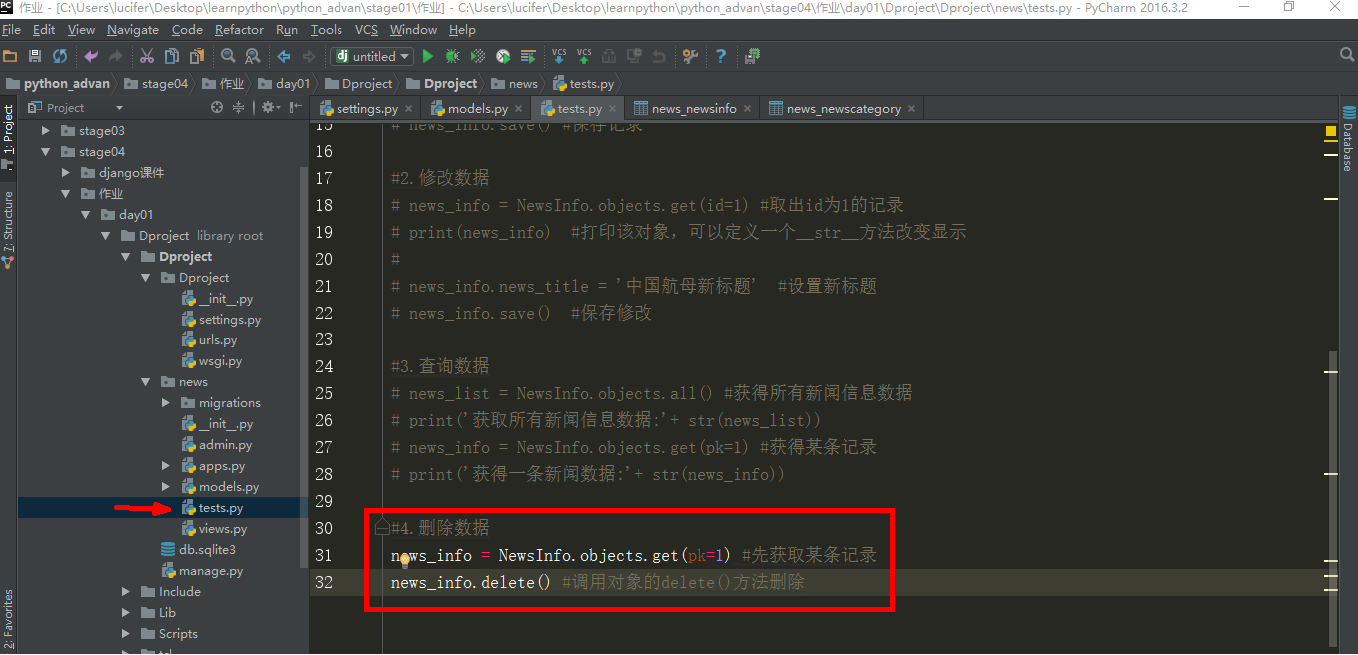
----------------------------------------------------------------------

Ran 0 tests in 0.000s

OK

Destroying test database for alias 'default'...

* 删除数据



* 以下是源代码

#4.删除数据  
news\_info = NewsInfo.objects.get(pk=1) #先获取某条记录  
news\_info.delete() #调用对象的delete()方法删除

* 使用命令删除记录

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py test news**

Creating test database for alias 'default'...

System check identified no issues (0 silenced).

----------------------------------------------------------------------

Ran 0 tests in 0.000s

OK

Destroying test database for alias 'default'...

可以看到表中已经无记录了



接下去简略介绍以下使用shell查询的方法

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>python manage.py shell

Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

**>>> from news.models import \***

**>>> nc = NewsCategory.objects.get(pk=1)**

**>>> nc.cag\_name**

'军事新闻'

**>>> ncs = NewsCategory.objects.all()**

**>>> ncs**

<QuerySet [<NewsCategory: NewsCategory object>]>

**>>> nc = NewsCategory()**

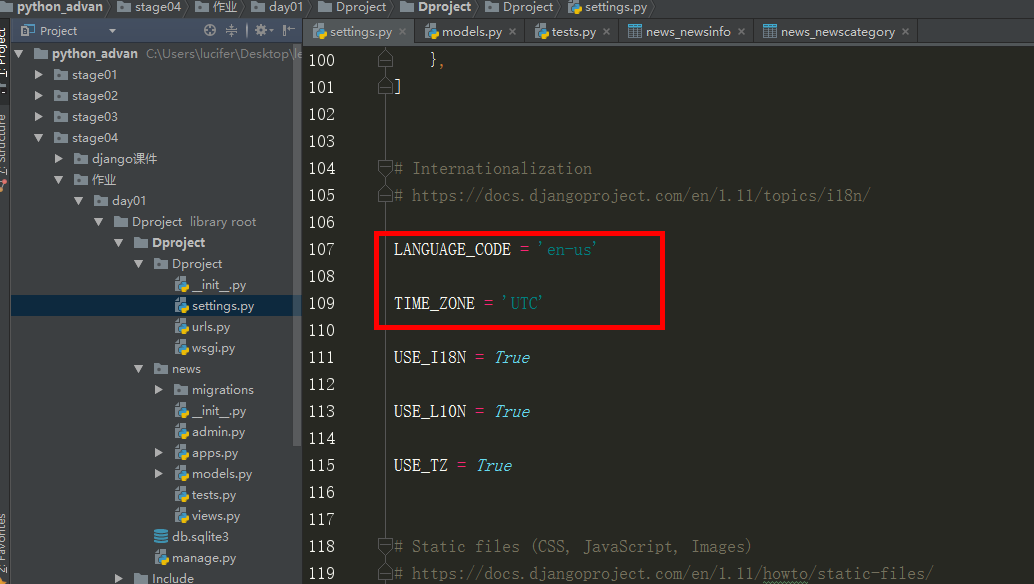
**>>> nc.cag\_name = '娱乐新闻'**

**>>> nc.save()**

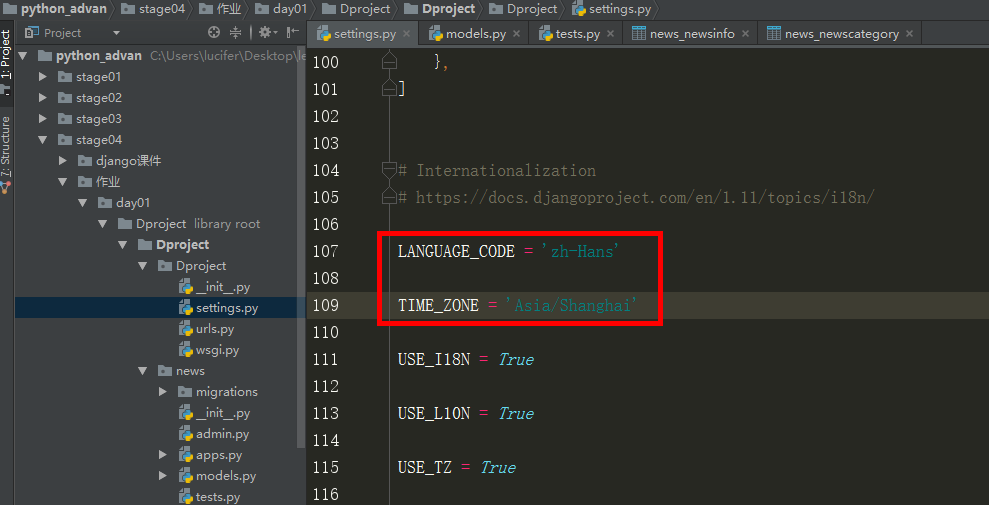
# 后台管理

* 管理界面本地化;
* 创建管理员;
* 注册模型类;
* 自定义管理界面.
* 管理本地化

修改settings.py文件



修改成



源代码如下

LANGUAGE\_CODE = 'zh-Hans'  
  
TIME\_ZONE = 'Asia/Shanghai'

* 创建管理员

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py createsuperuser**

Username (leave blank to use 'lucifer'): **admin**

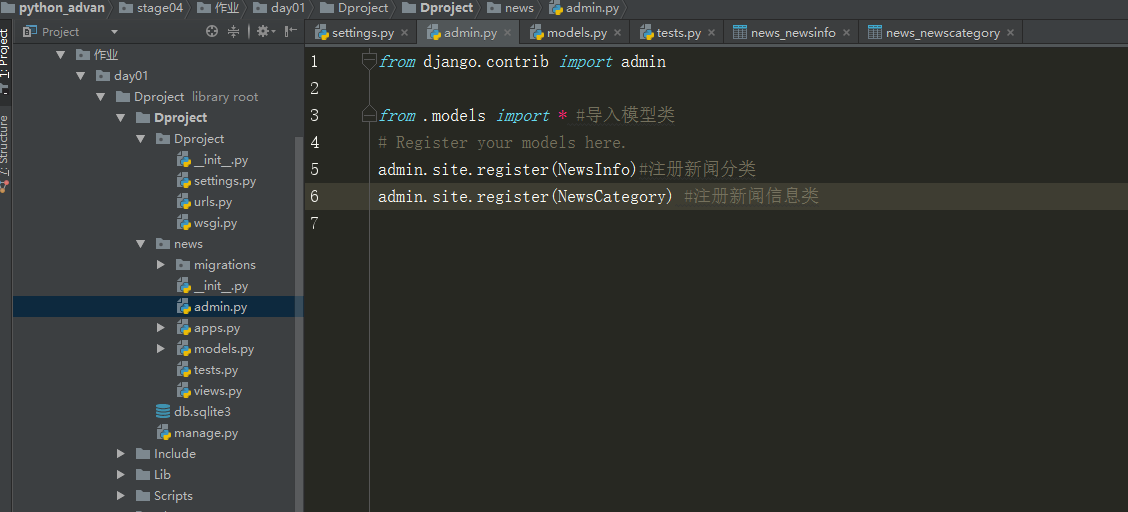
Email address: **admin@Dproject.com**

Password:**abcd1234,**

Password (again):

Superuser created successfully.

* 注册模型类



以下是源代码

*from* .models *import* \* #导入模型类  
# Register your models here.  
admin.site.register(NewsInfo)#注册新闻分类  
admin.site.register(NewsCategory) #注册新闻信息类

* 启动服务器

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py runserver**

Performing system checks...

System check identified no issues (0 silenced).

October 06, 2017 - 18:45:04

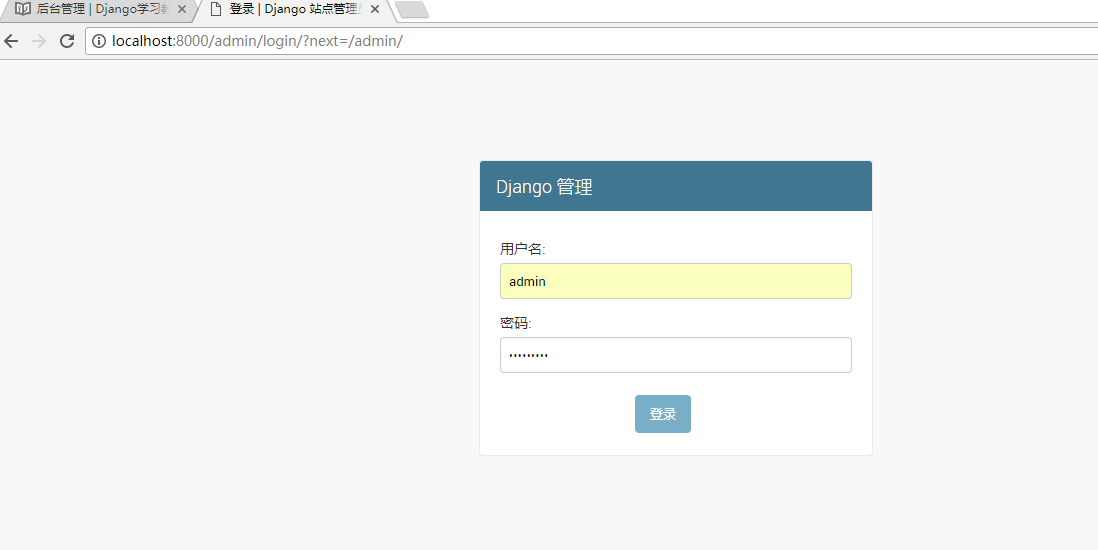
Django version 1.11.5, using settings 'Dproject.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

通过浏览器访问管理站点

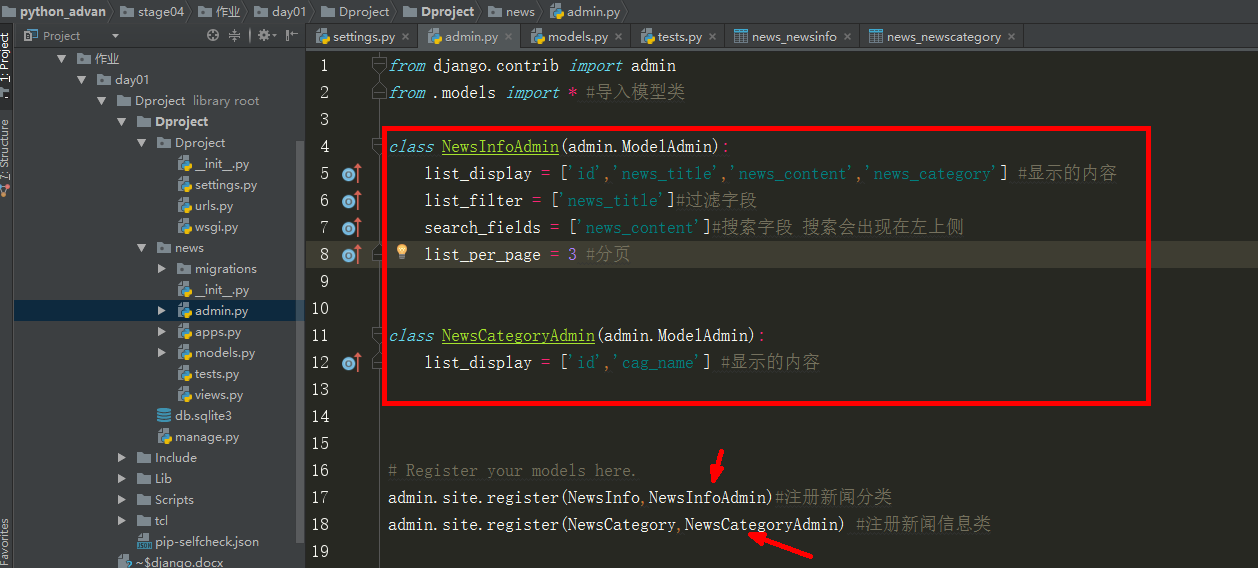
网址是 http://localhost:8000/admin





* 自定义界面管理类

在admin.py中创建连个类并设置 list\_display



以下是源代码

*from* django.contrib *import* admin  
*from* .models *import* \* #导入模型类  
  
*class* NewsInfoAdmin(admin.ModelAdmin):  
 list\_display = ['id','news\_title','news\_content','news\_category'] #显示的内容  
 list\_filter = ['news\_title']#过滤字段  
 search\_fields = ['news\_content']#搜索字段 搜索会出现在左上侧  
 list\_per\_page = 3 #分页  
  
  
*class* NewsCategoryAdmin(admin.ModelAdmin):  
 list\_display = ['id','cag\_name'] #显示的内容  
  
  
  
# Register your models here.  
admin.site.register(NewsInfo,NewsInfoAdmin)#注册新闻分类  
admin.site.register(NewsCategory,NewsCategoryAdmin) #注册新闻信息类

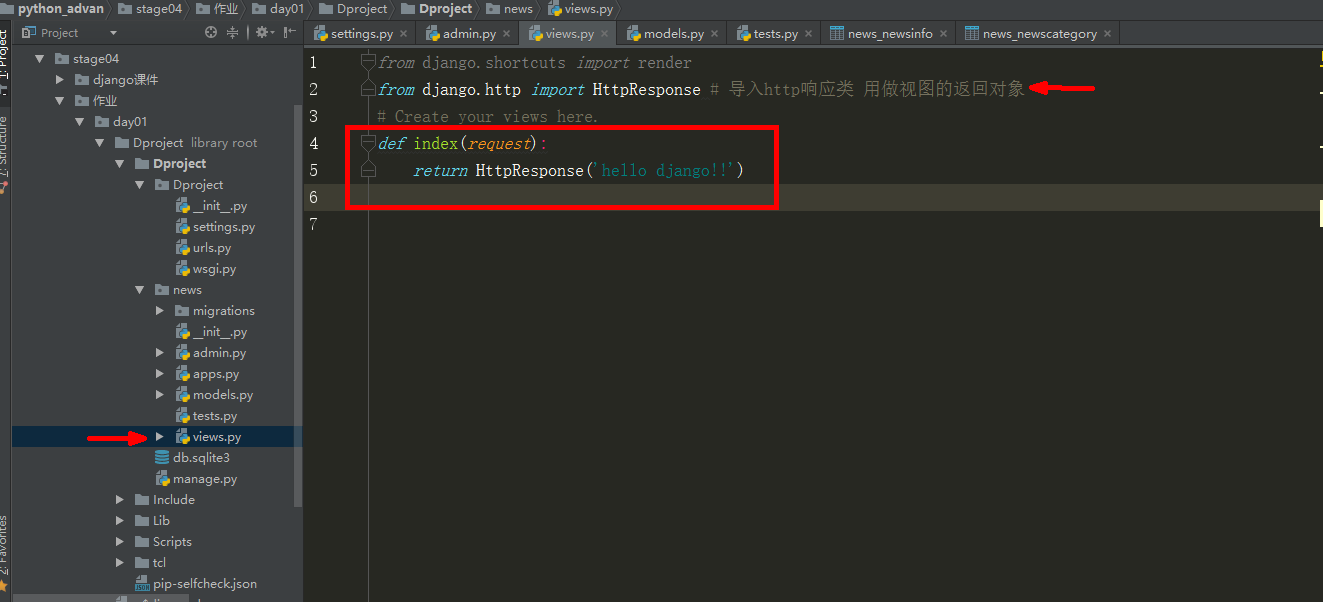
以下是修改后的显示效果



# 视图

* 定义视图;
* 配置URLconf
* 定义视图

先定义一个主页视图 接受request请求参数，并返回最简单的页面 hello django

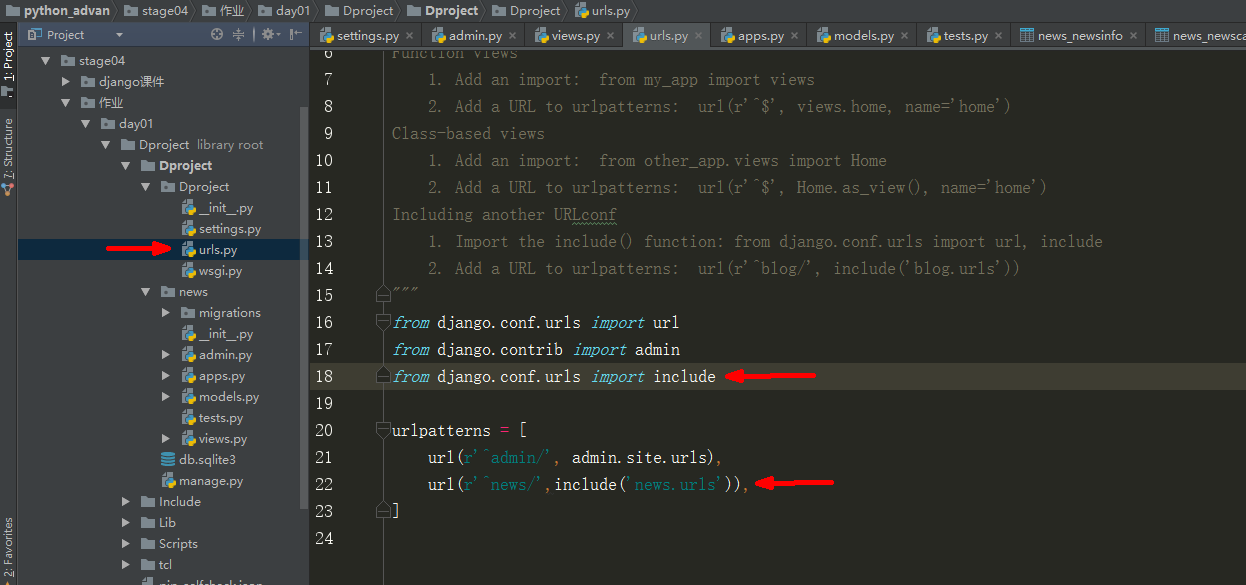


以下是源代码内容

*from* django.shortcuts *import* render  
*from* django.http *import* HttpResponse # 导入http响应类 用做视图的返回对象  
# Create your views here.  
*def* index(*request*):  
 *return* HttpResponse('hello django!!')

* 配置urlconf

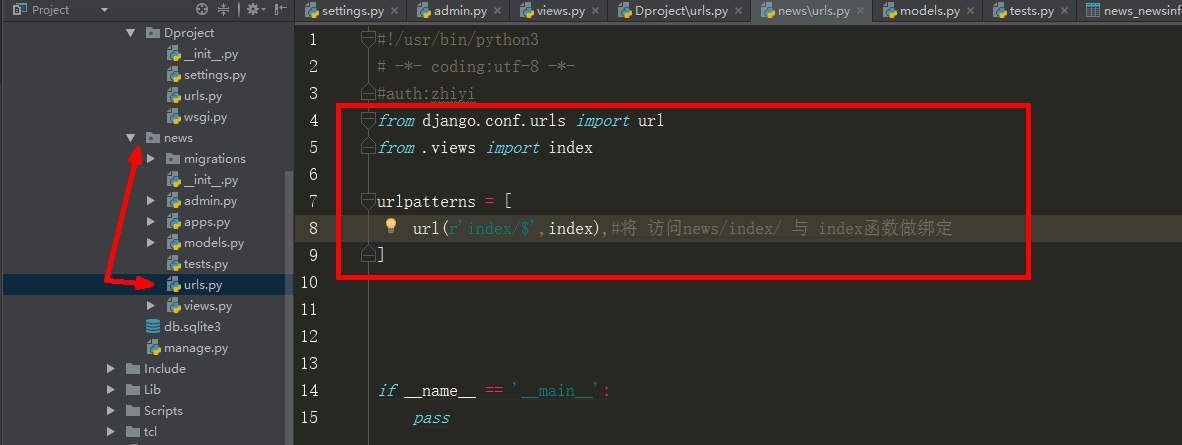
在项目下的urls.py文件修改以下内容



源代码如下：

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

然后在news 这个app的文件夹下 再建立一个urls.py 放入如下内容



代码如下

*from* django.conf.urls *import* url  
*from* .views *import* index  
  
urlpatterns = [  
 url(r'index/$',index),#将 访问news/index/ 与 index函数做绑定  
]

* 重新启动服务

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject\Dproject>**python manage.py runserver**

Performing system checks...

System check identified no issues (0 silenced).

October 06, 2017 - 19:41:28

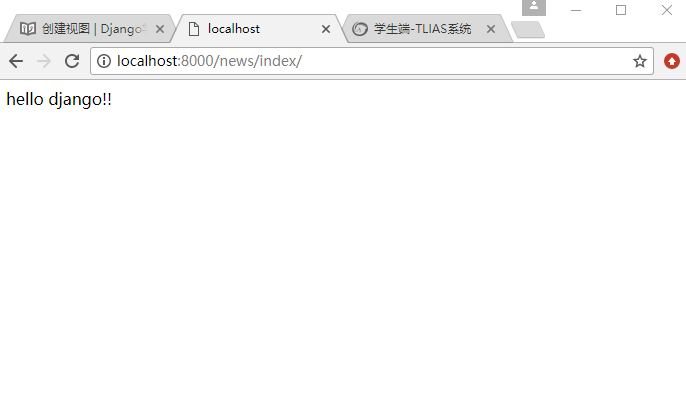
Django version 1.11.5, using settings 'Dproject.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

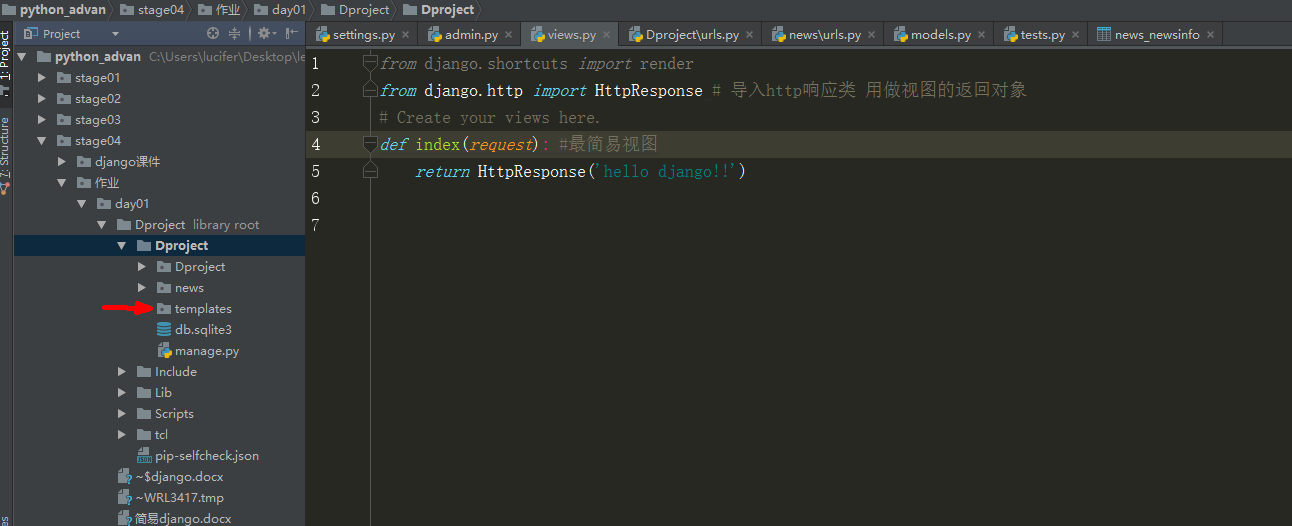
[06/Oct/2017 19:41:31] "GET /news/index/ HTTP/1.1" 200 14

* 访问http://localhost:8000/news/index/

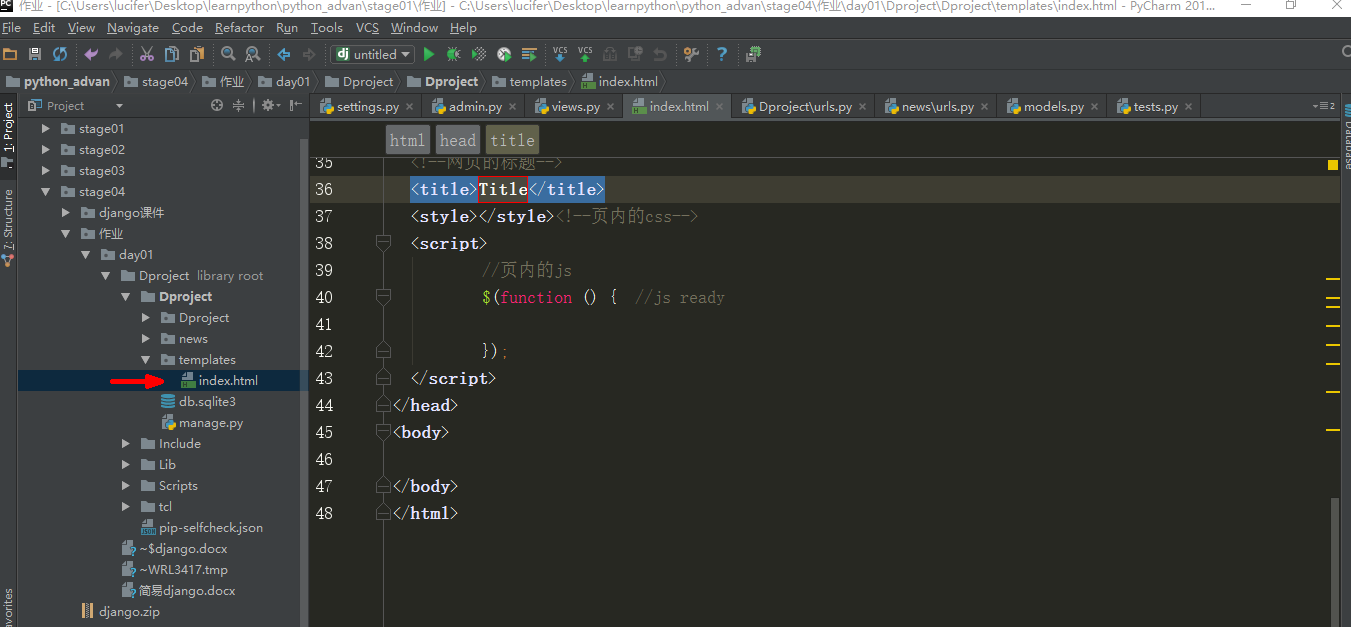


# 使用模板

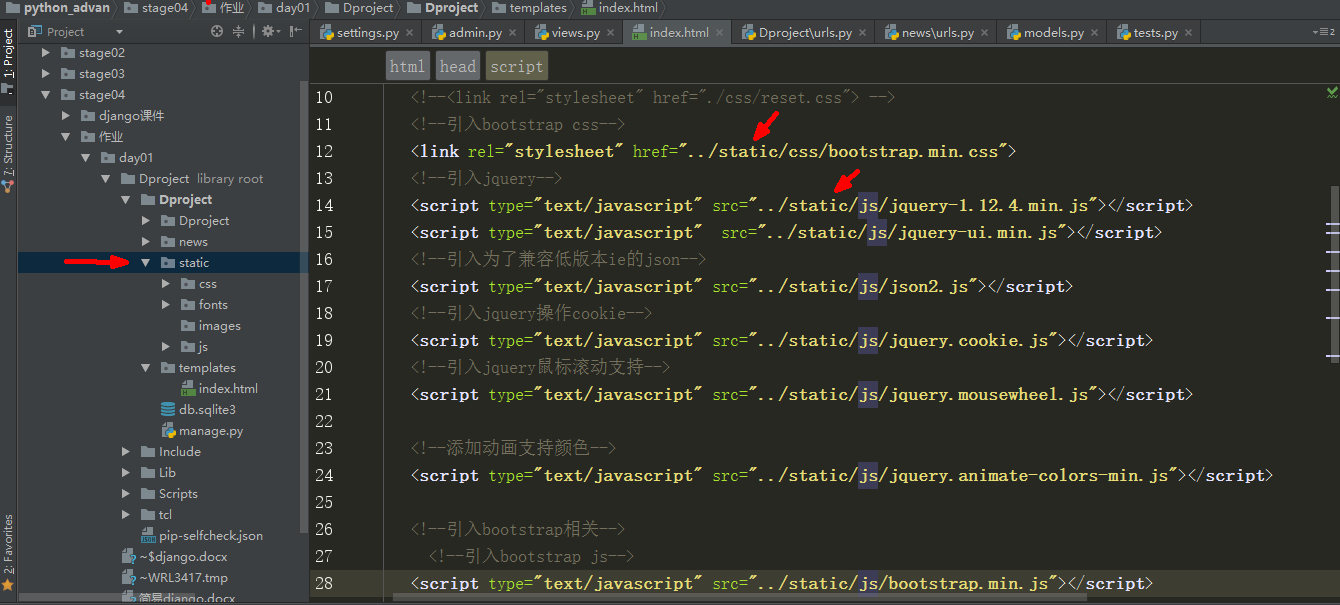
首先在项目目录下新建一个 templates文件夹专门用于存放模板



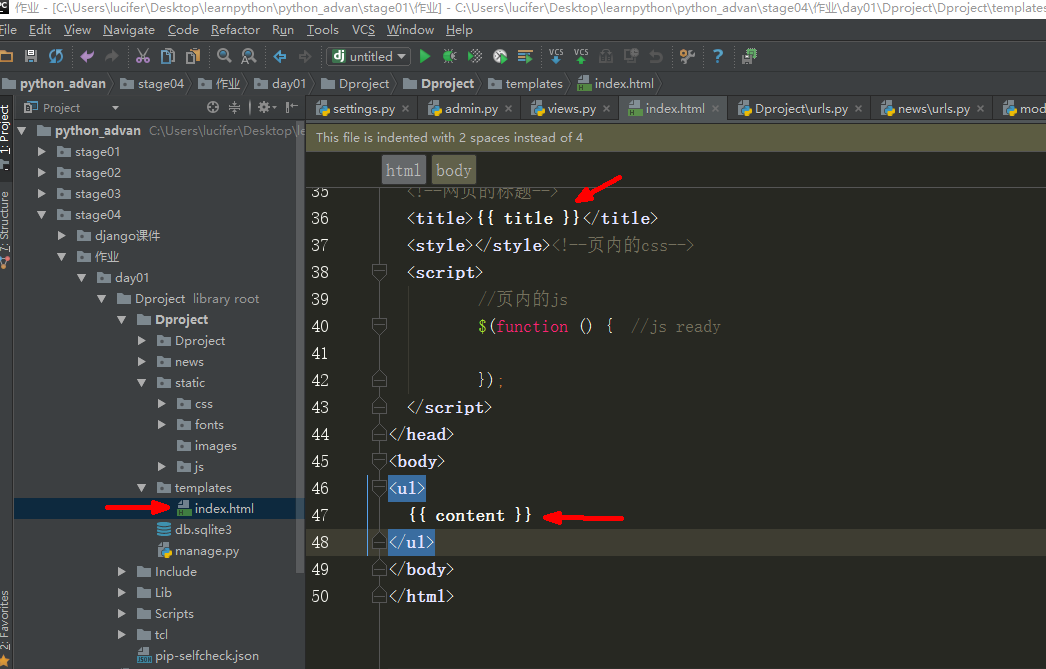
* 在模板文件夹下创建一个 index.html



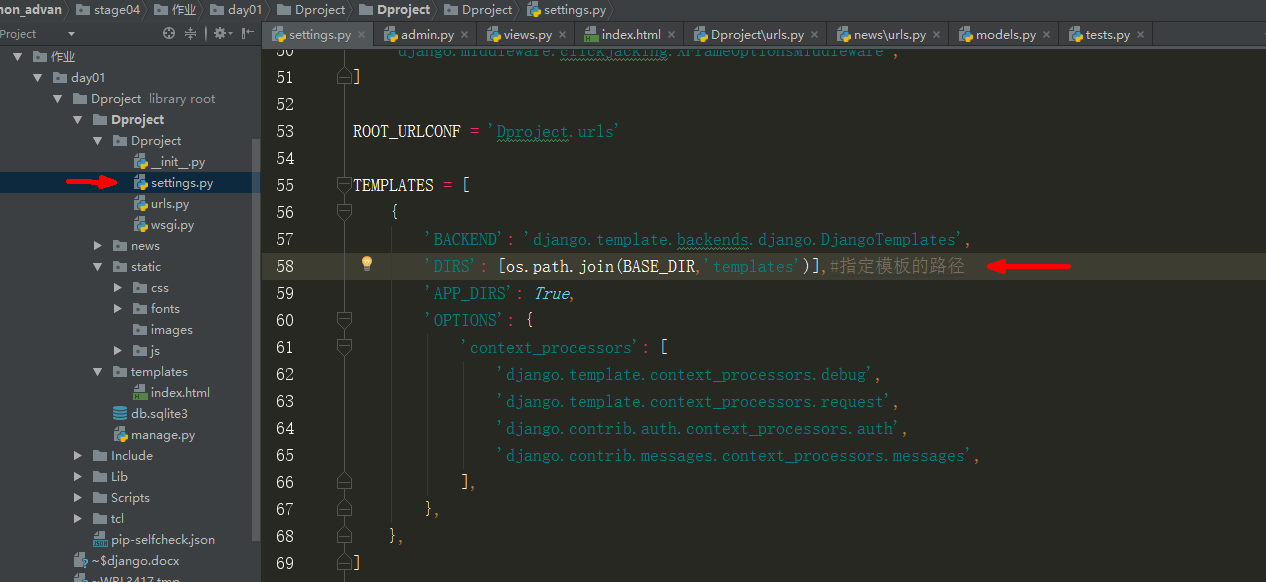
由于我在pycharm中设置了 html的模板会引入css和jquery等js文件 因此特此在建立一个static文件夹，并且拷贝css和js等，并且路径也做了修改



在index.html中增加可替换内容的标记



* 在settings.py中注册模板目录

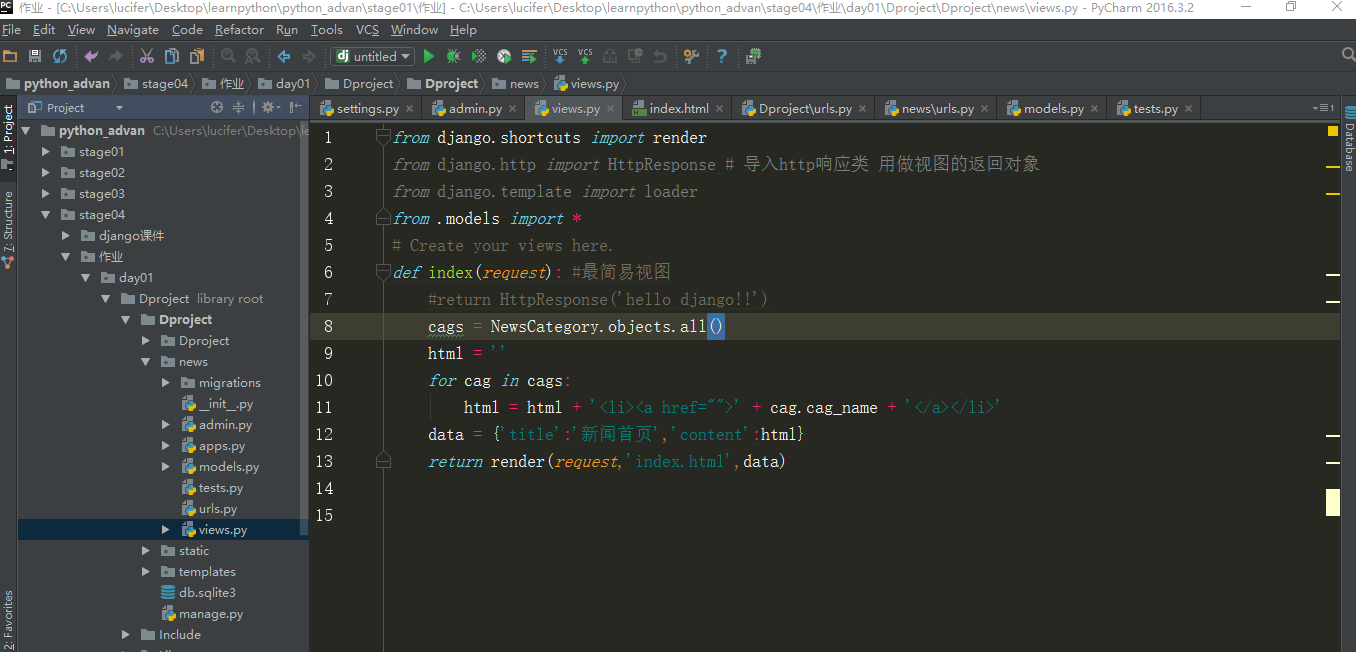


代码如下：

TEMPLATES = [  
 {  
 'BACKEND': 'django.template.backends.django.DjangoTemplates',  
 'DIRS': [os.path.join(BASE\_DIR,'templates')],#指定模板的路径  
 'APP\_DIRS': *True*,  
 'OPTIONS': {  
 'context\_processors': [  
 'django.template.context\_processors.debug',  
 'django.template.context\_processors.request',  
 'django.contrib.auth.context\_processors.auth',  
 'django.contrib.messages.context\_processors.messages',  
 ],  
 },  
 },  
]

* 使用模板

修改views.py下的 index函数

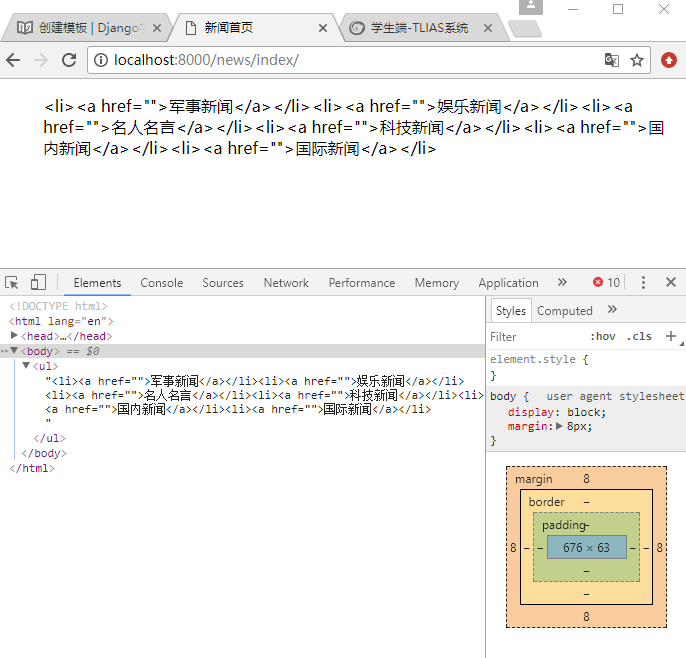


代码如下

*from* django.shortcuts *import* render  
*from* django.http *import* HttpResponse # 导入http响应类 用做视图的返回对象  
*from* django.template *import* loader  
*from* .models *import* \*  
# Create your views here.  
*def* index(*request*): #最简易视图  
 #return HttpResponse('hello django!!')  
 cags = NewsCategory.objects.all()  
 html = ''  
 *for* cag *in* cags:  
 html = html + '<li><a href="">' + cag.cag\_name + '</a></li>'  
 data = {'title':'新闻首页','content':html}  
 *return* render(*request*,'index.html',data)

修改完毕后

访问http://localhost:8000/news/index/



可以看到内容已经成功过替换了 只是内容的外面多了引号。这个问题等待下一次再解决

* 接以上 来解决一下页面显示的问题

修改虚拟环境启动方式

(Dproject) C:\Users\lucifer\Desktop\learnpython\python\_advan\stage04\作业\day01\Dproject>**Scripts\python.exe Dproject\manage.py runserver**

Performing system checks...

System check identified no issues (0 silenced).

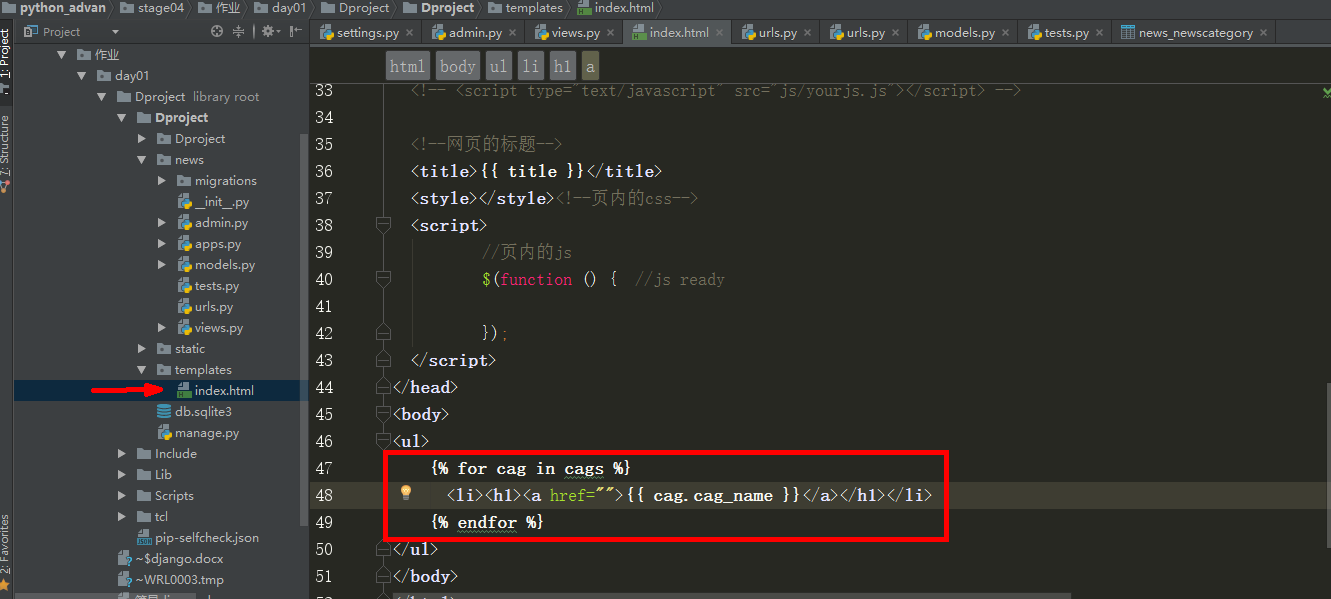
October 07, 2017 - 14:38:28

Django version 1.11.5, using settings 'Dproject.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

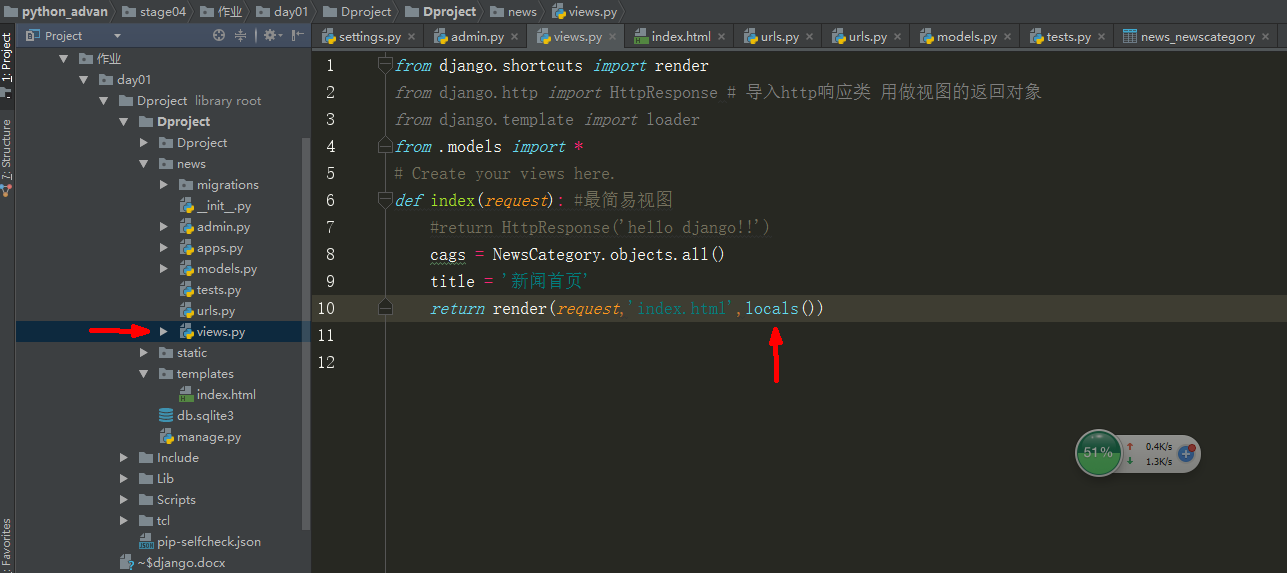
修改index页面里的代码，将for循环写在index里



具体代码如下

<**body**>  
<**ul**>  
 **{% for cag in cags %}** <**li**><**h1**><**a** href=**""**>**{{ cag.cag\_name }}**</**a**></**h1**></**li**>  
 **{% endfor %}**</**ul**>  
</**body**>

修改views里的index函数，更改替换方式，使用locals函数会更为方便，无需在自己组装字典，locals函数会为你组装



以下为源代码：

*from* django.shortcuts *import* render  
*from* django.http *import* HttpResponse # 导入http响应类 用做视图的返回对象  
*from* django.template *import* loader  
*from* .models *import* \*  
# Create your views here.  
*def* index(*request*): #最简易视图  
 #return HttpResponse('hello django!!')  
 cags = NewsCategory.objects.all()  
 title = '新闻首页'  
 *return* render(*request*,'index.html',locals())

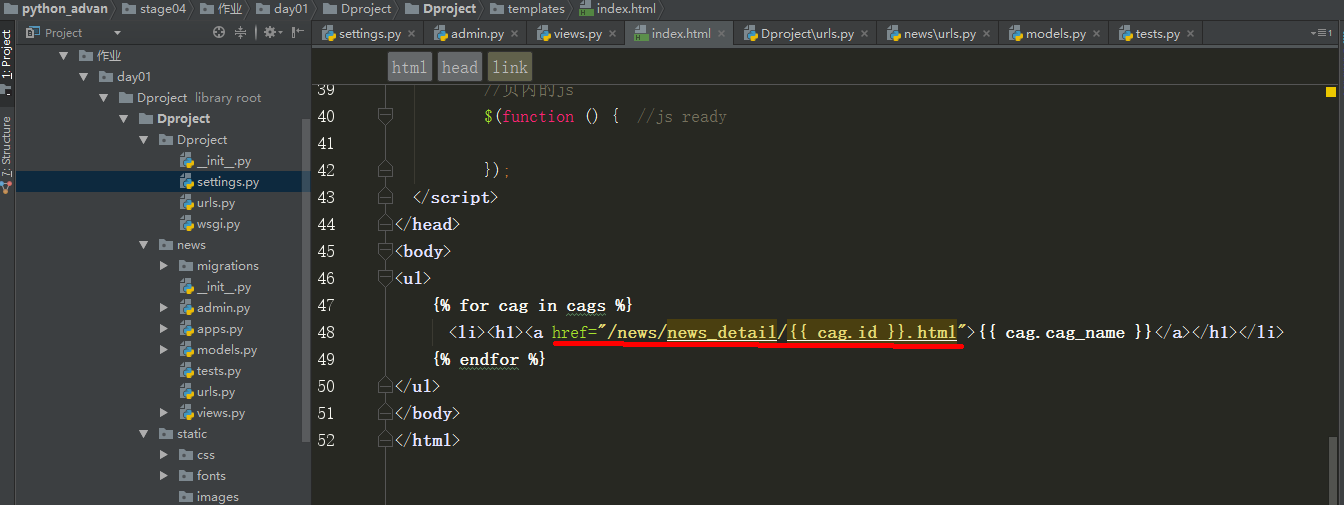
修改完毕之后再访问网页 http://127.0.0.1:8000/news/index/



到此网页已经按照要求已经正确的显示了

但是网页还没添加链接，我们要做到根据点击链接，显示具体新闻的内容

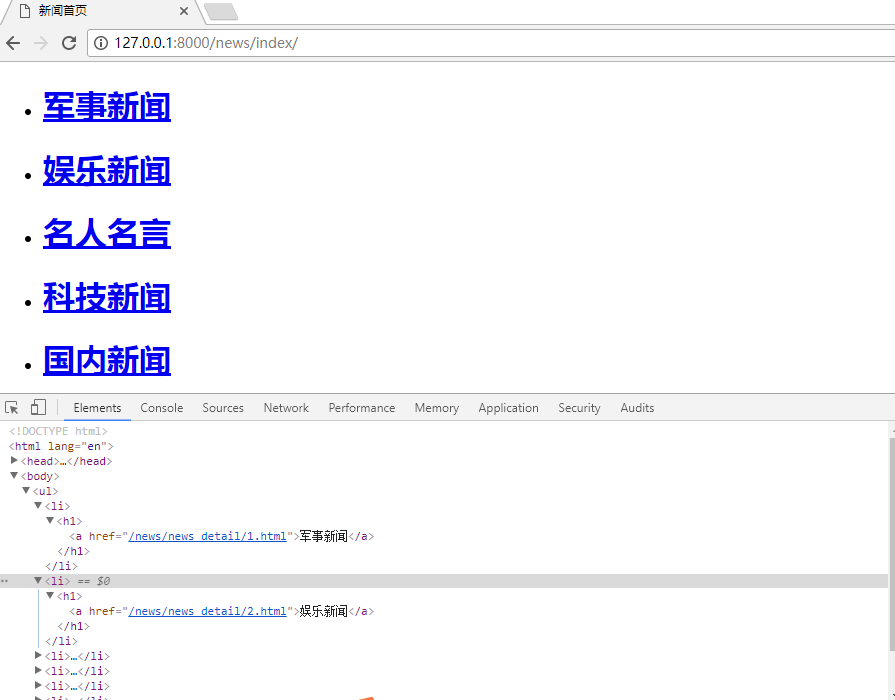
再次修改index.html 增加网页的herf 指向id页



源代码如下：

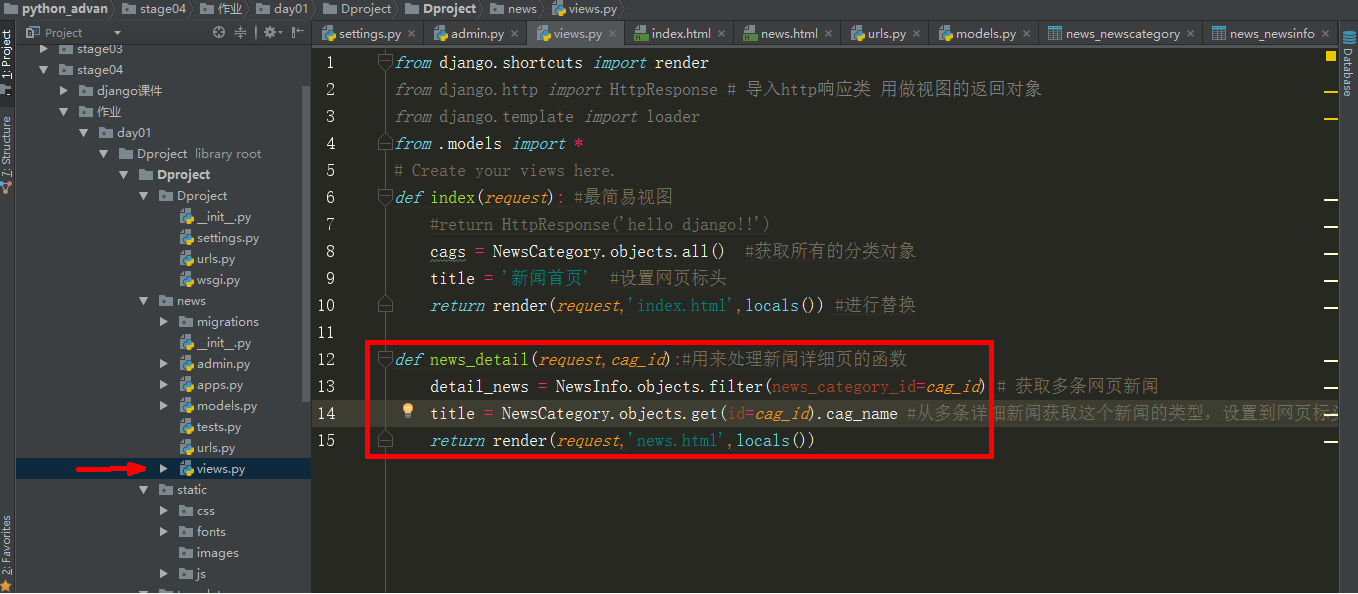
**{% for cag in cags %}** <**li**><**h1**><**a** href=**"/news/news\_detail/{{ cag.id }}.html"**>**{{ cag.cag\_name }}**</**a**></**h1**></**li**>  
**{% endfor %}**

经过这样的修改之后网页的超链接就生成了



可以看到不同的 超链接指向了不同的地址，接下来要做的就是在view中写一个详细新闻的函数，已经对应的模板 来处理这个请求

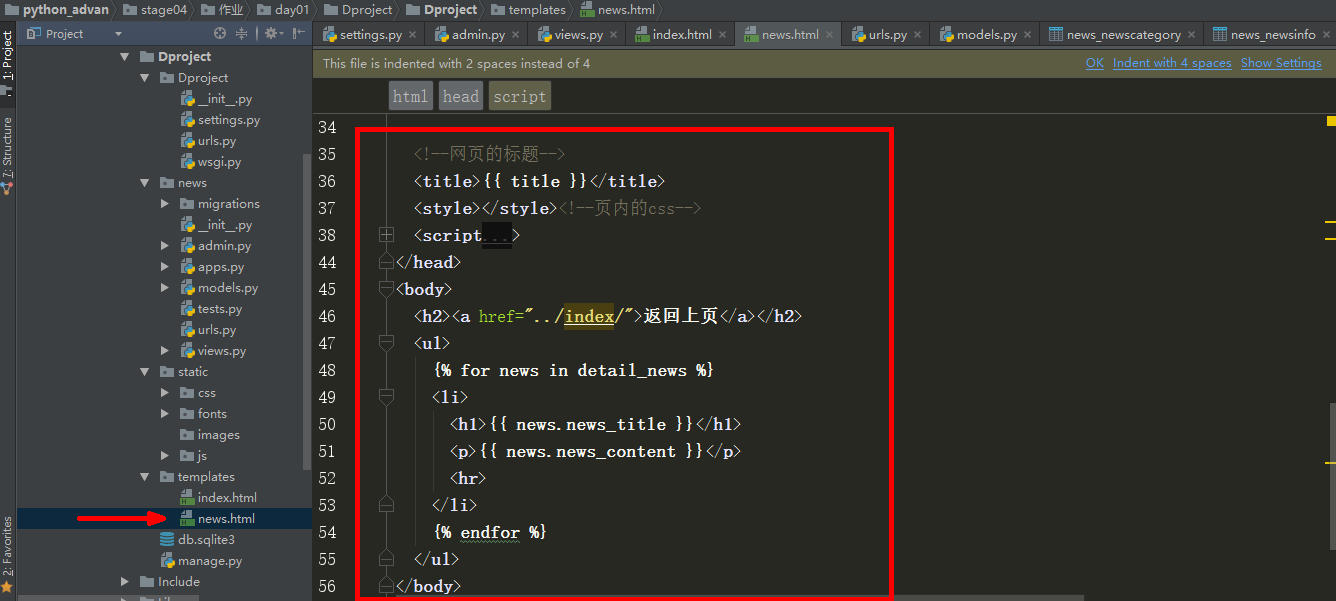
在view中增加一个函数news\_detail 专门用来处理详细新闻页的逻辑



具体代码如下：

*def* news\_detail(*request*,*cag\_id*):#用来处理新闻详细页的函数  
 detail\_news = NewsInfo.objects.filter(news\_category\_id=*cag\_id*) # 获取多条网页新闻  
 title = NewsCategory.objects.get(id=*cag\_id*).cag\_name #从多条详细新闻获取这个新闻的类型，设置到网页标头  
 *return* render(*request*,'news.html',locals())

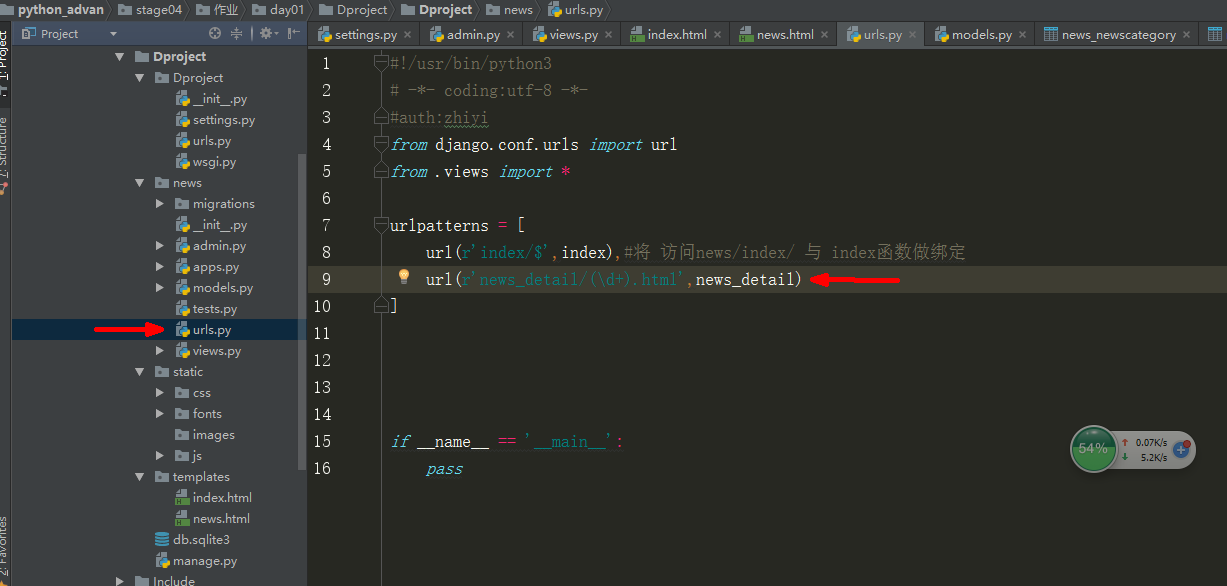
然后增加一个news的模板文件



源代码如下:

<!--网页的标题-->  
 <**title**>**{{ title }}**</**title**>  
 <**style**></**style**><!--页内的css-->  
 <**script**>  
 //页内的js  
 $(function () { //js ready  
   
 });   
 </**script**>  
</**head**>  
<**body**>  
 <**h2**><**a** href=**"../index/"**>**返回上页**</**a**></**h2**>  
 <**ul**>  
 **{% for news in detail\_news %}** <**li**>  
 <**h1**>**{{ news.news\_title }}**</**h1**>  
 <**p**>**{{ news.news\_content }}**</**p**>  
 <**hr**>  
 </**li**>  
 **{% endfor %}** </**ul**>  
</**body**>

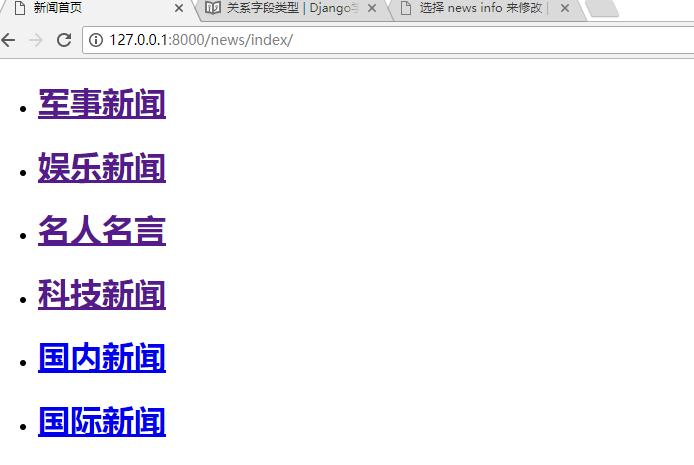
要在news项目下的urls.py文件中加上一个路由才能正常工作



正则代码如下：

url(r'news\_detail/(\d+).html',news\_detail)

完成以下工作后 就可以访问网页了







可以看到不同的页面点击后出现的内容是不相同的