# 第十一章 Django 后台管理与基本请求

本节所讲内容：

11.1 注册数据库到后台

11.2 定制自己的后台样式

11.3 xadmin后台样式

11.4 富文本编译器

11.5 网站分页功能实现

## 11.1 注册数据库到后台

Django拥有很多在网站开发过程当中需要的通用功能（这些功能被统一的放到了django.contrib下），使用这些功能降低了开发者开发的难度，提高了开发的效率，这也是django在Python当中相当受欢迎的一大因素，而在django拥有的这些功能当中，让大家最为熟知的就是django的后台管理系统，admin。

在之前的课程当中已经完成了数据库模型的搭建，也就是说，现在可以给网站搭建数据库，并且可以实现数据库数据的查询了，但是这里有个问题冒了出来。现在如何将数据添加到数据库当中，手写总是不现实的。所以这里要学习动态网站的另一个感念，叫做：后台，在这里一定要把后端和后台区分开。

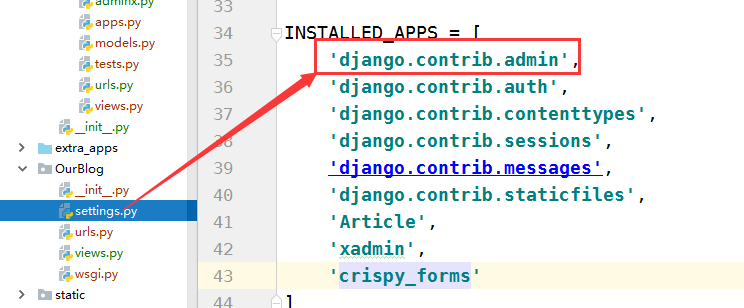
后端：指出来HTML，css设计的页面之外所有和服务器数据库的交互

后台：指对网站的数据进行管理的页面和系统，类似在话剧院当中的后台，前台演戏，后台准备管理，

Django有自带的admin后台管理系统，要使用后台首先确认后台系统是不是被激活，需要查看settings

11.1.1 admin操作步骤

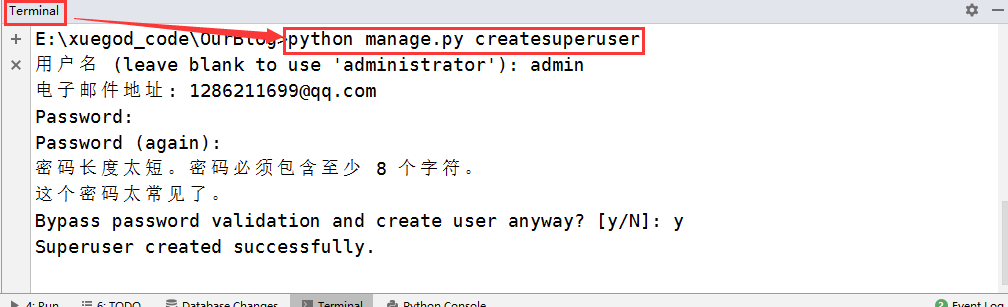
1、确定install当中django.contrib.admin这条配置存在，并且没有被注释



2、确定路由当中有admin的路由，并且没有被注释

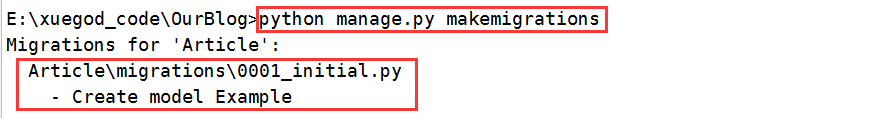


3、创建管理员权限createsuperuser   
 基本命令：python manage.py createsuperuser



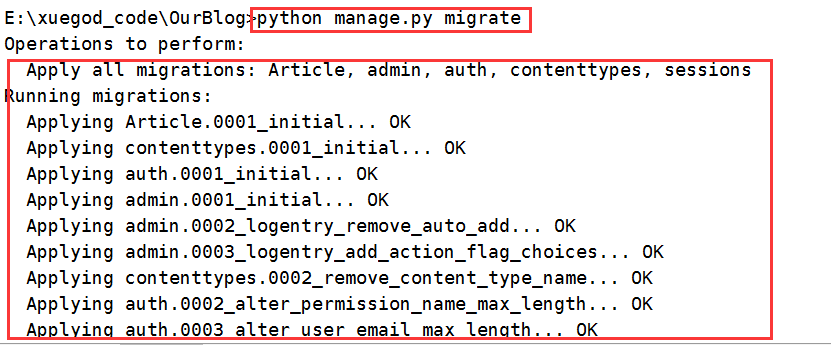
4、然后进行数据库迁移

生成数据库语句：python manage.py makemigrations

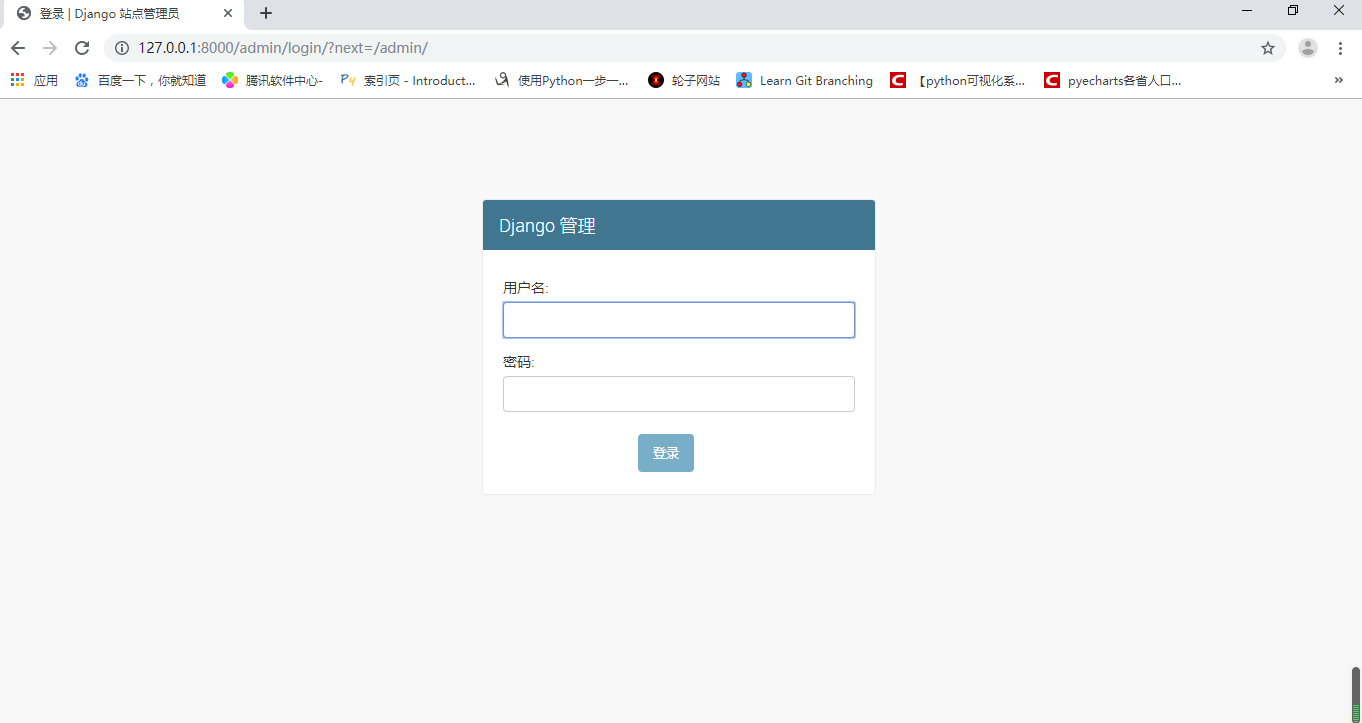


进行数据库的同步

现在我们使用的是python manage.py migrate [appname] 可指定app单独迁移



上面的操作都完成以后，这个时候访问ip：端口/admin(127.0.0.1:8000)就会进入后台管理界面的登录界面





## 11.2 定制自己的后台样式

11.2.1样式类定义

在上一小节学习过程当中，我们已经学会将django的模型添加到后台当中，但是如果我们要给后台添加样式需要定义样式类

list\_display = ('id','Name',)     --列表中显示的列

search\_fields = ('id', 'Name')   --搜索框

list\_filter = ('Name',)              --侧边过滤器

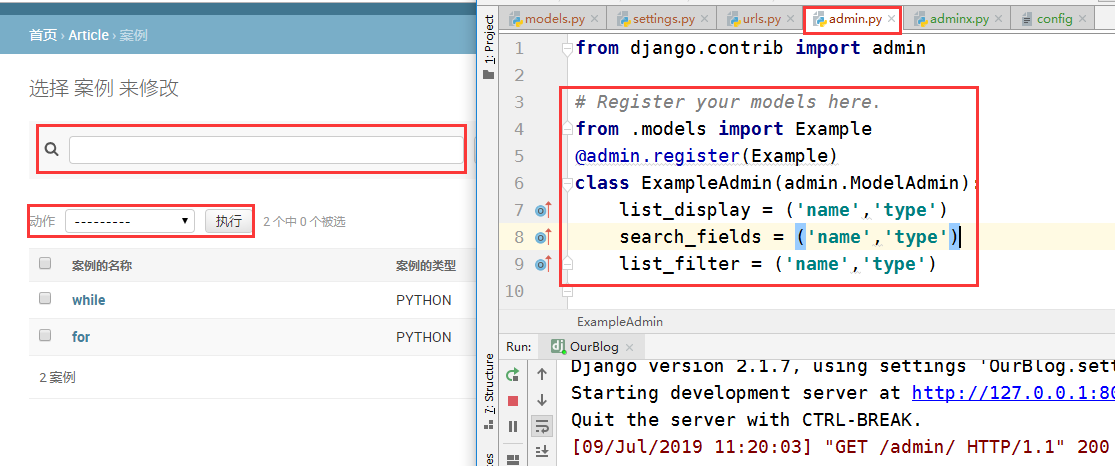
date\_hierarchy = 'DataTime3'  --时间下拉

ordering = ('-DataTime3',)      --列表中的排序

fields = ('Name','DataTime1')  --详细页面字段顺序

filter\_horizontal = ('Name',)    --显示多对多的关系

raw\_id\_fields = ('publisher',)   --显示外键的数据



11.3 xadmin后台样式

上面我们已经讲解了django中自带的一个admin后台管理，但是样式等信息都很low 所以说想要写好一个项目，强大的后台是必不少的。接下来让我们一起学习xadmin.

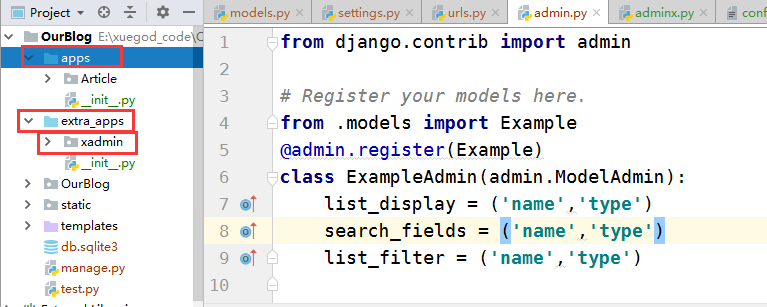
Xadmin 安装：pip install xadmin（安装的时候可能出现报错行为）

注意：这种安装后的xadmin会有很多地方和我们的django版本冲突，可能需要修改很多导入文件，非常麻烦，所以说我们直接把已经改好的源码放到本地，方便提取并使用。

1.配置apps

为了方便管理django生成的app我们写一个python Packpage 统一管理，如下图所示：

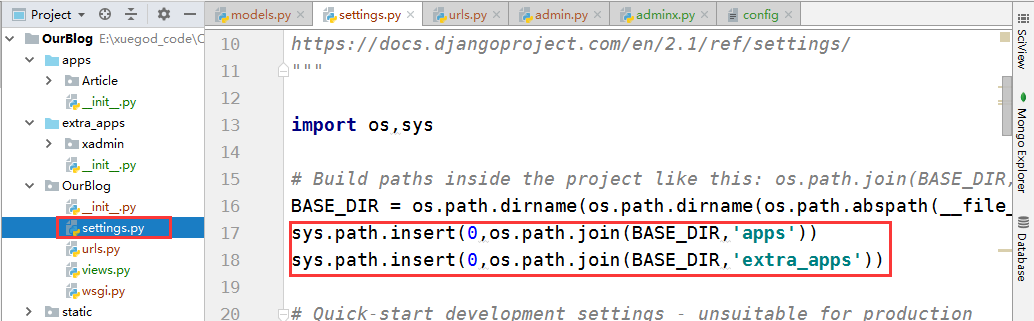
新建Python Package "extra\_apps",把源码xadmin文件夹放到extra\_apps文件夹下面，此时目录结构如下：



把apps 和extra\_apps全部键mark为Source Root（主要是让django项目能够在apps下寻找）

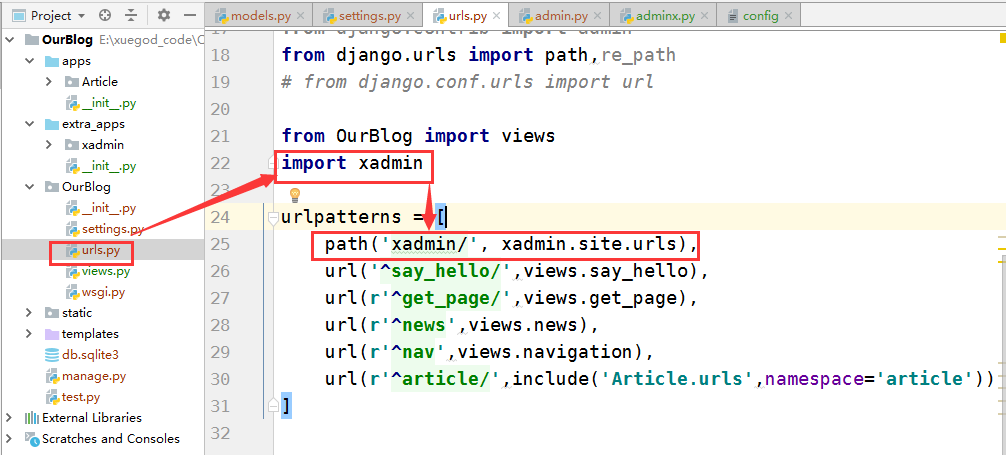
但是这种设置后我们的cmd运行也会报错所以说要在settings中加入配置，配置信息如下：

import os,sys  
*# Build paths inside the project like this: os.path.join(BASE\_DIR, ...)*BASE\_DIR = os.path.dirname(os.path.dirname(os.path.abspath(\_\_file\_\_)))  
sys.path.insert(0,os.path.join(BASE\_DIR,'apps'))  
sys.path.insert(0,os.path.join(BASE\_DIR,'extra\_apps'))



2.配置总路由（urls.py）

导入xadmin并修改总路由把admin修改成xadmin



3.注册app

把下面两个app注册到settings.py的INSTALLED\_APPS中

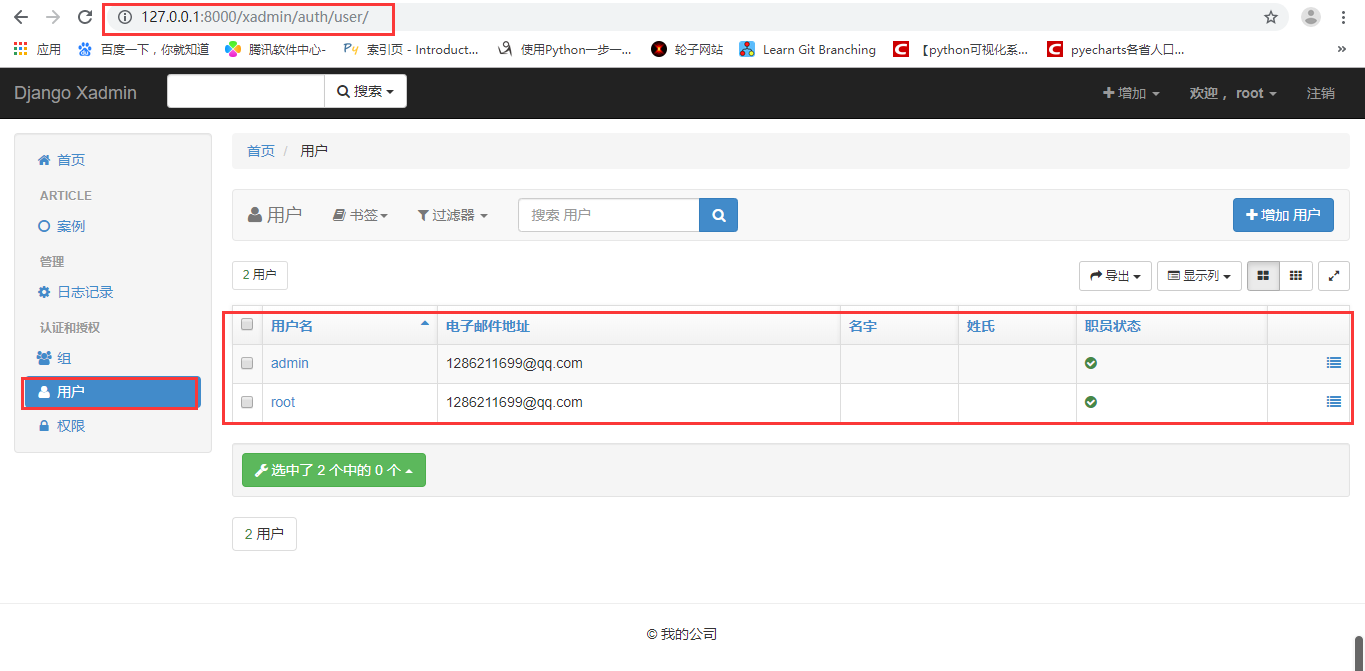
说明：这个crispy\_forms中主要是配合xadmin后台中表单提交的，功能更加强大

4.重新生成数据库

python manage.py makemigrations

python manage.py migrate

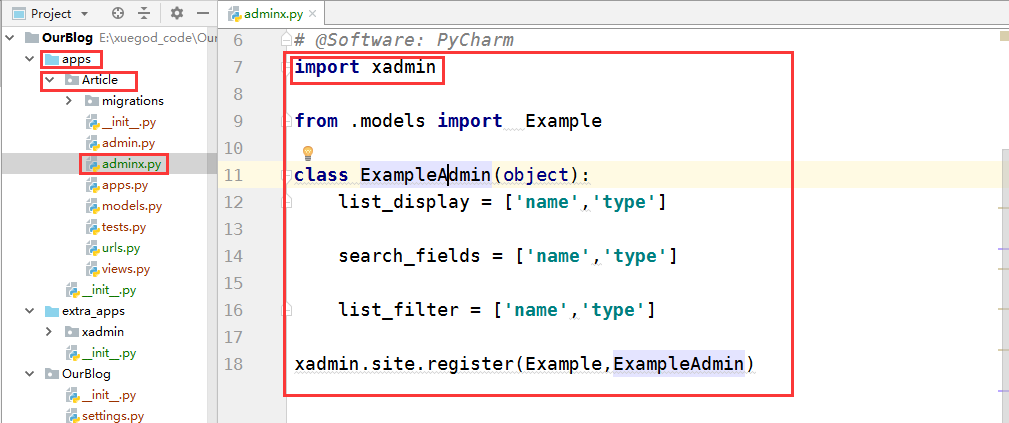
访问网址<http://127.0.0.1:8000/xadmin>效果如下图所示：



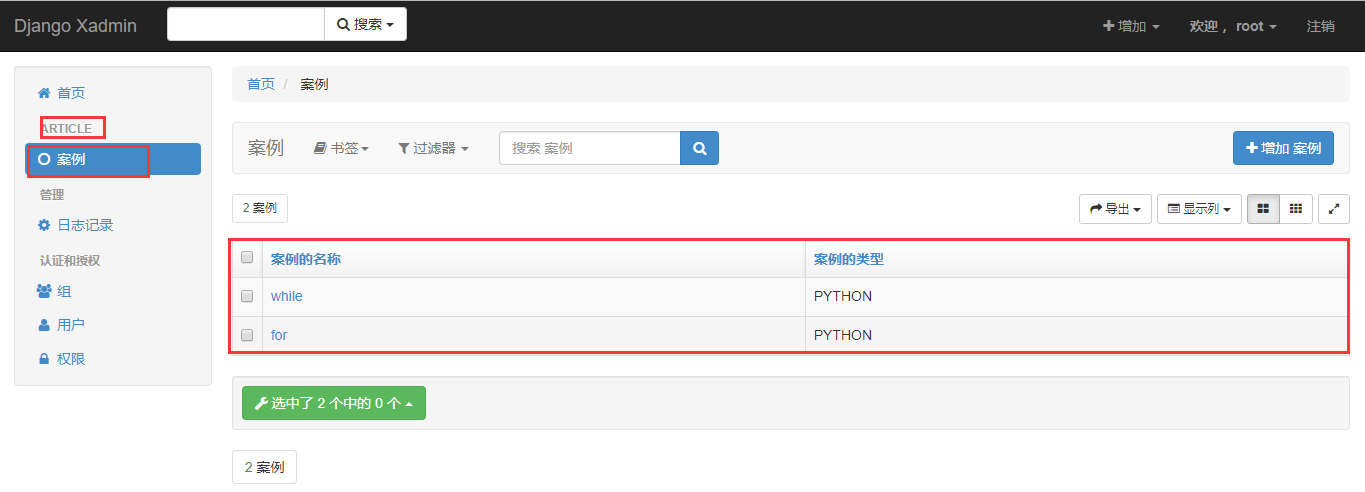
11.3.2 自定义xadmin后台样式

在 apps/Article/下创建一个叫adminx.py文件 然后导入xadmin定制自己的样式

import xadmin  
from .models import Example  
class ExampleAdmin(object):  
 list\_display = ['name','type']  
 search\_fields = ['name','type']  
 list\_filter = ['name','type']  
xadmin.site.register(Example,ExampleAdmin)



效果展示如下图所示：

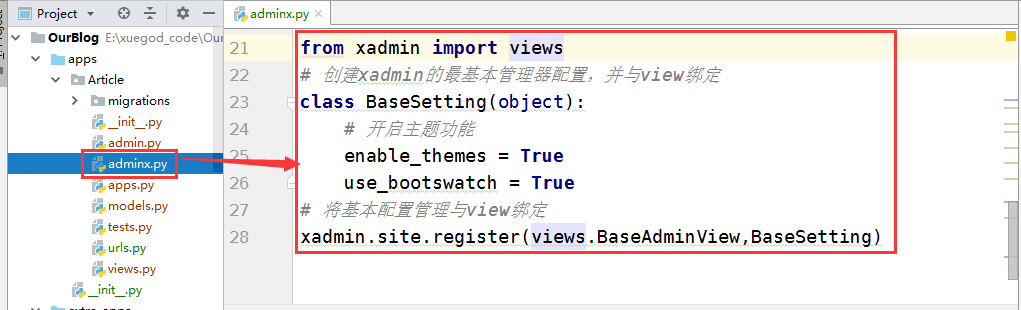


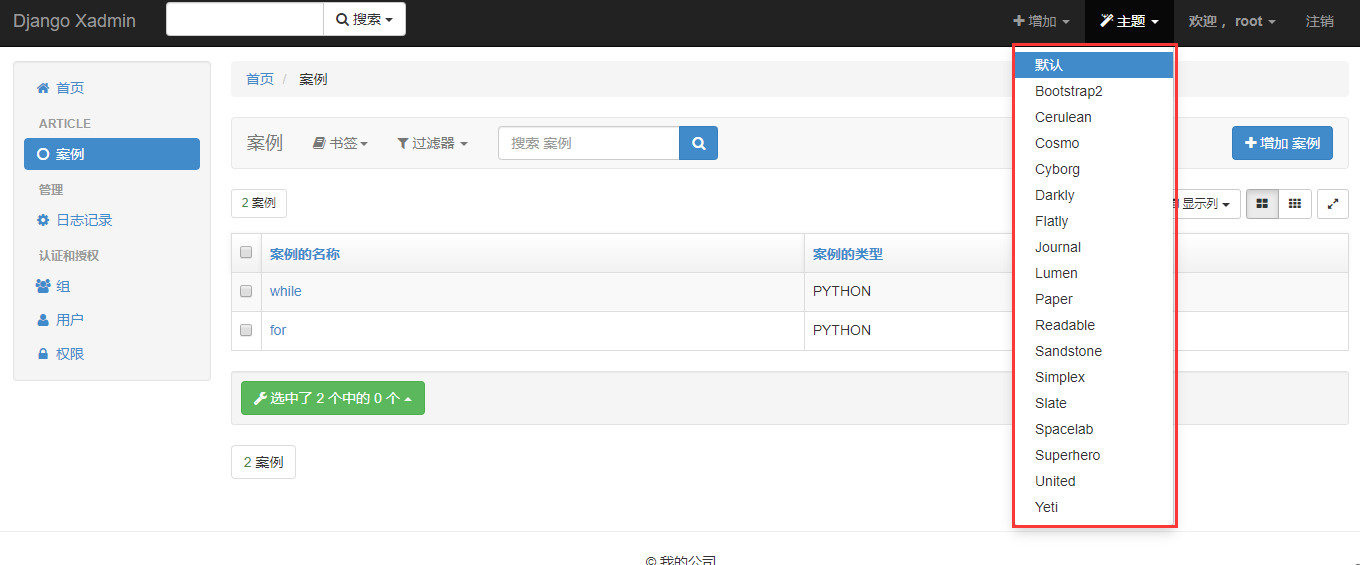
现在看起来样式比自带的admin样式好多了

如果想要修改xadmin的主题以及各种样式

下面增加后台主题环节，由于只有一个app（Article）那我们就在此adminx.py 下进行配置全局主题

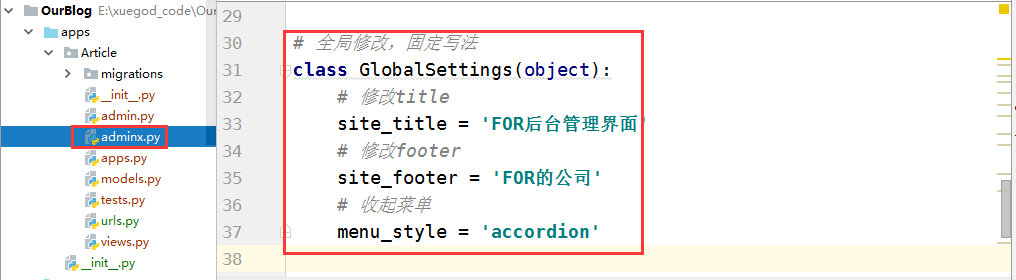
from xadmin import views  
*# 创建xadmin的最基本管理器配置，并与view绑定*class BaseSetting(object):  
 *# 开启主题功能* enable\_themes = True  
 use\_bootswatch = True  
*# 将基本配置管理与view绑定*xadmin.site.register(views.BaseAdminView,BaseSetting)、



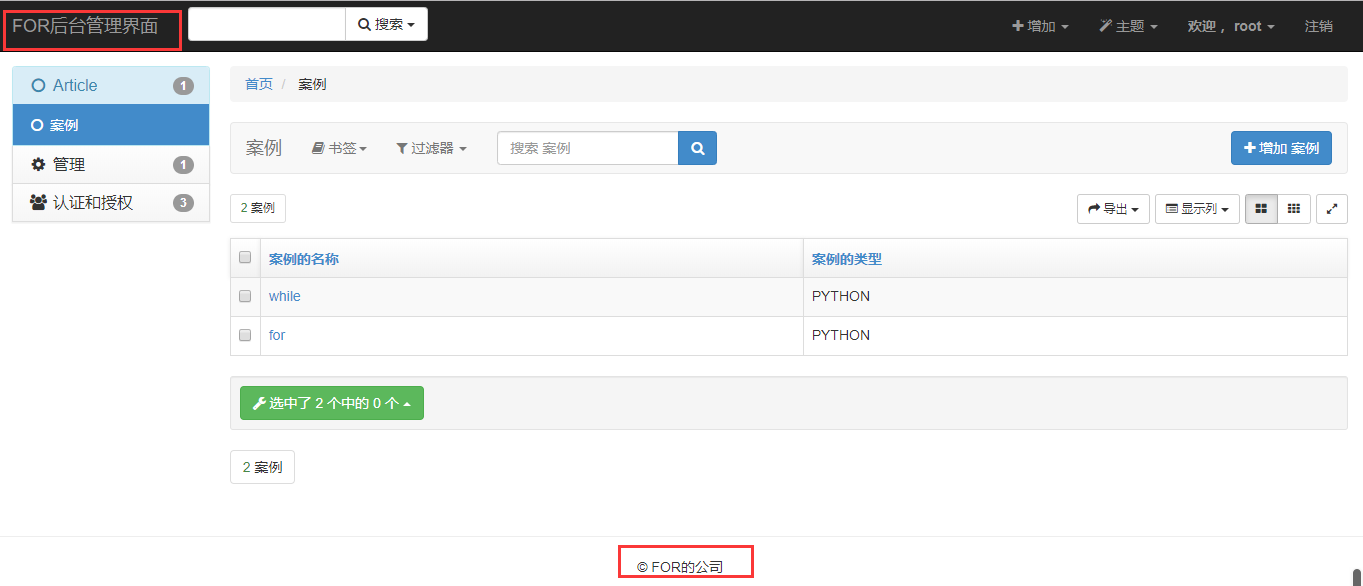


现在开始修改我们的后台说明

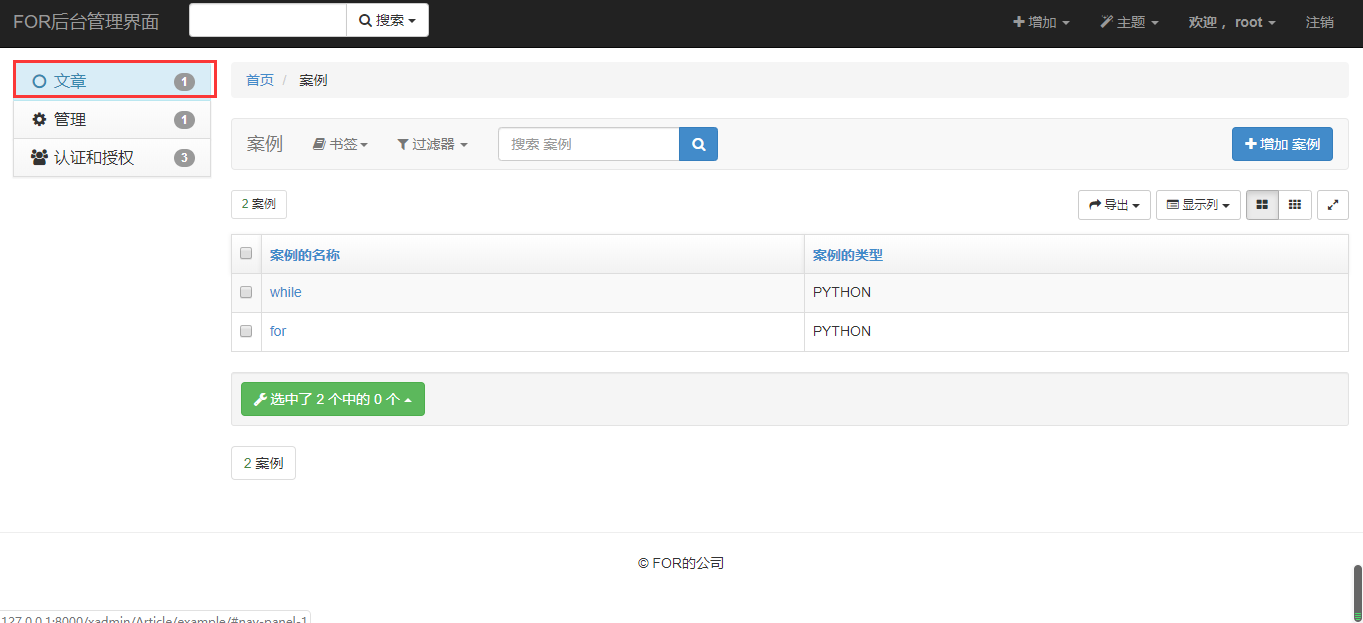
*# 全局修改，固定写法*class GlobalSettings(object):  
 *# 修改title* site\_title = 'FOR后台管理界面'  
 *# 修改footer* site\_footer = 'FOR的公司'  
 *# 收起菜单* menu\_style = 'accordion'  
*# 将title和footer信息进行注册*xadmin.site.register(views.CommAdminView,GlobalSettings)



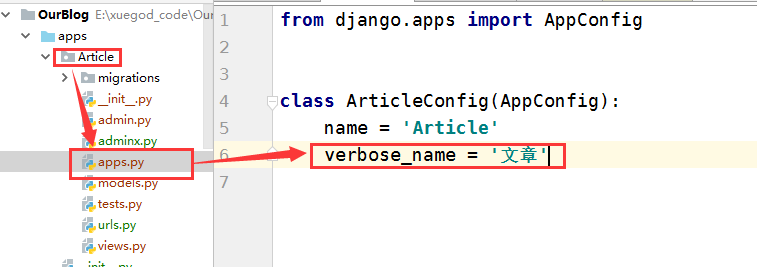
运行效果如下：



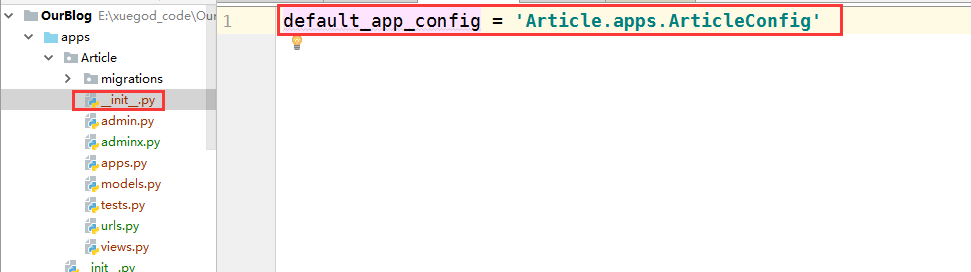
把每个app变成中文名



代码如下：

第一步在当前app（Article）中的apps.py中修改如下：  


然后配置下当前\_\_init\_\_.py下的环境



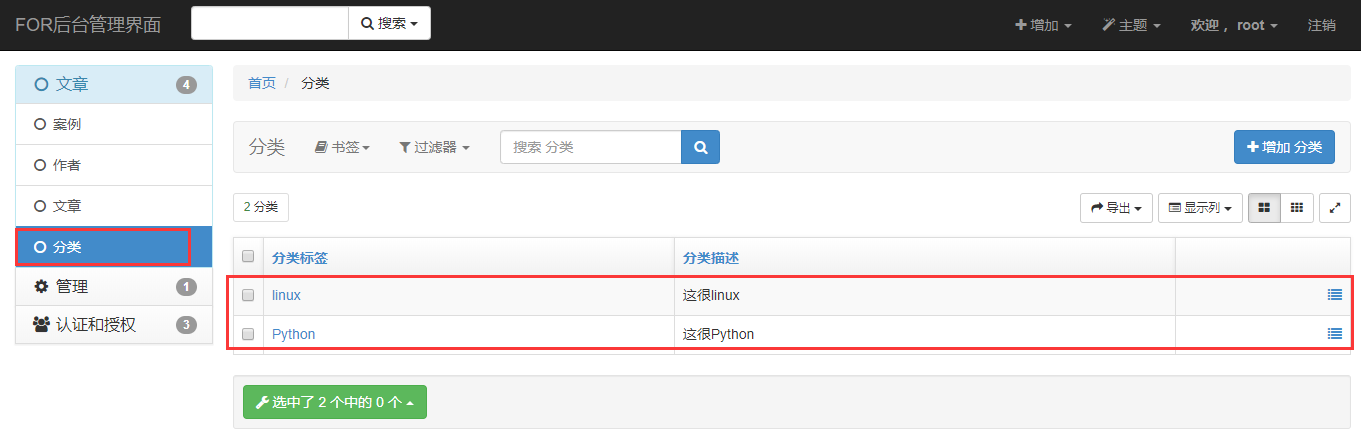
就能实现一个app的中文展示。好的学到这里我们基本的后台管理已经学完了，后面有些公司也会给后台一定的样式让它看起来更加美好，这个就在最后工程里润化就好了。

上面的课程中我们已经实现后台管理界面，接下来让我们实现添加文章的基本操作如下图所示：

先行的添加作者



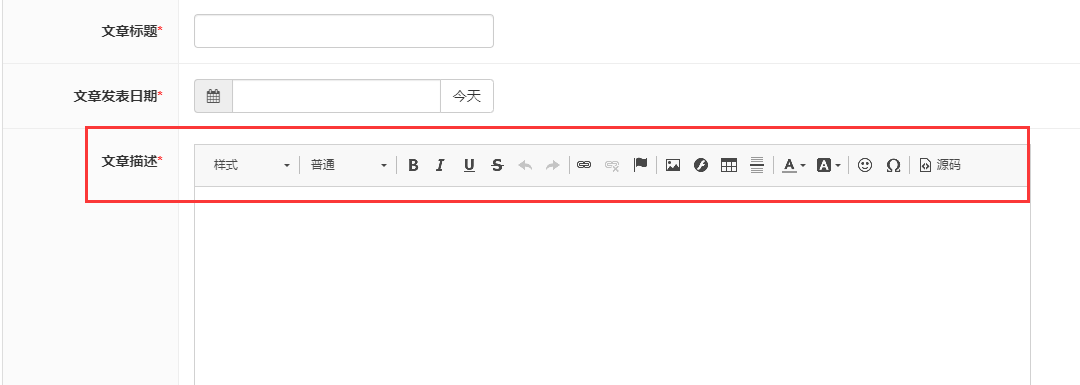
添加分类



添加文章：



可看到我们上边的文章中已经存在了几篇，让我们再来编写几篇文章



11.3 富文本编译器

在我打开操作进行编写的时候你可明显的看到我的这个功能更加强大。这是为什么？接下来就让我们一起进入django中的富文本编译器的世界吧。

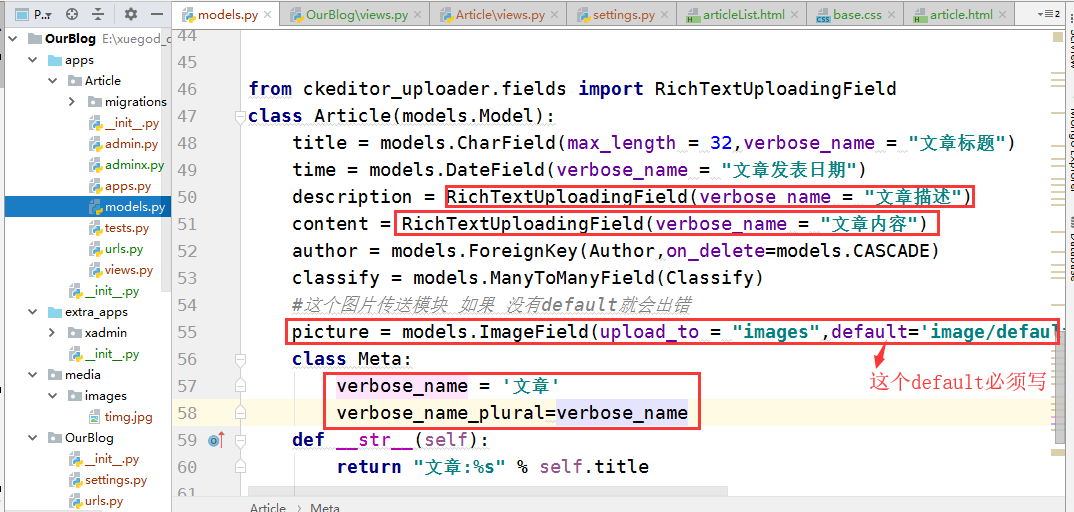
富文本编辑器ckeditor

这个需要先使用pip install django-ckeditor==5.7.1,注意版本

文章的数据库模型

Articles/models.py

**Class Article编写  
from ckeditor\_uploader.fields import RichTextUploadingField  
class Article(models.Model):  
 title = models.CharField(max\_length = 32,verbose\_name = "文章标题")  
 time = models.DateField(verbose\_name = "文章发表日期")  
 description = RichTextUploadingField(verbose\_name = "文章描述")  
 content = RichTextUploadingField(verbose\_name = "文章内容")  
 author = models.ForeignKey(Author,on\_delete=models.CASCADE)  
 classify = models.ManyToManyField(Classify)  
 *#这个图片传送模块 如果 没有default就会出错* picture = models.ImageField(upload\_to = "images/%Y%m",default='image/default.png',verbose\_name = "文章图片")  
 class Meta:  
 verbose\_name = '文章'  
 verbose\_name\_plural=verbose\_name  
 def \_\_str\_\_(self):  
 return "文章:%s" % self.title**

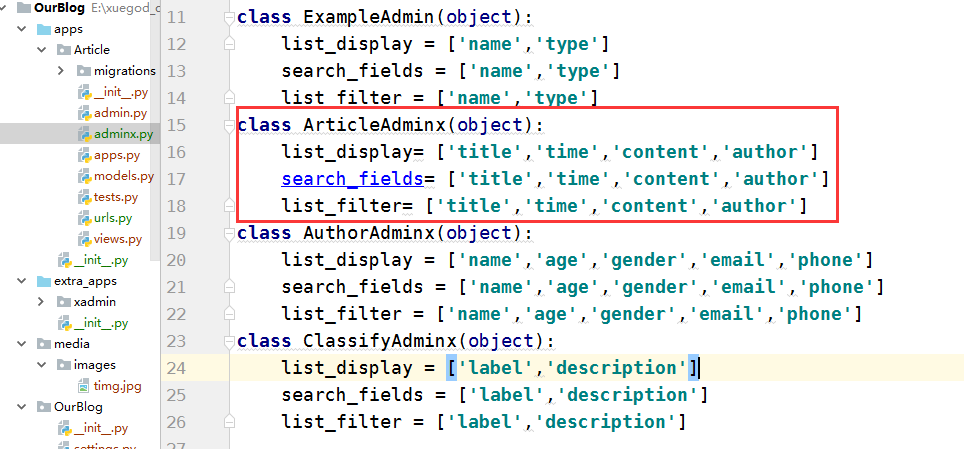


安装模型到后台

from django.contrib import admin

from Article.models import \*

xadmin.site.register(Article)





1. 进行settings配置

#安装APP功能

INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
 'Article',  
 'xadmin',  
 'crispy\_forms',  
 'ckeditor',  
 'ckeditor\_uploader'  
]

#静态路由

STATIC\_URL = '/static/'

#静态文件目录

STATICFILES\_DIRS = [

os.path.join(BASE\_DIR,"static"),

]

#媒体路由,用户上传的数据都在当前的目录下，并且保存

MEDIA\_URL = "/media/"

MEDIA\_ROOT = os.path.join(BASE\_DIR,"media")

#静态文件根目录

#STATIC\_ROOT = os.path.join(BASE\_DIR,"static/ckeditor").replace("\\","/")

#ckeditor上传路径，文件的上传路径，必须配合图形操作文件pillow。必要配置

CKEDITOR\_UPLOAD\_PATH = 'uploads/'

CKEDITOR\_IMAGE\_GACKEND = 'pillow'

1. 进行总url配置

from django.conf.urls import include, url

from django.contrib import admin

from OurBlog.views import index

from django.views.static import serve

urlpatterns = [

#……

url(r'^ckeditor/', include('ckeditor\_uploader.urls')),

*# 处理图片显示的url,使用Django自带serve,传入参数告诉它去哪个路径找，我们有配置好的路径MEDIAROOT，对应上面的media* url(r'^media/(?P<path>.\*)', serve, {"document\_root": MEDIA\_ROOT }),

#……

]

进行数据库同步，静态文件的收集（注：当前命令都基于我们项目的项目根目录）

语法校验

命令：python manage.py check

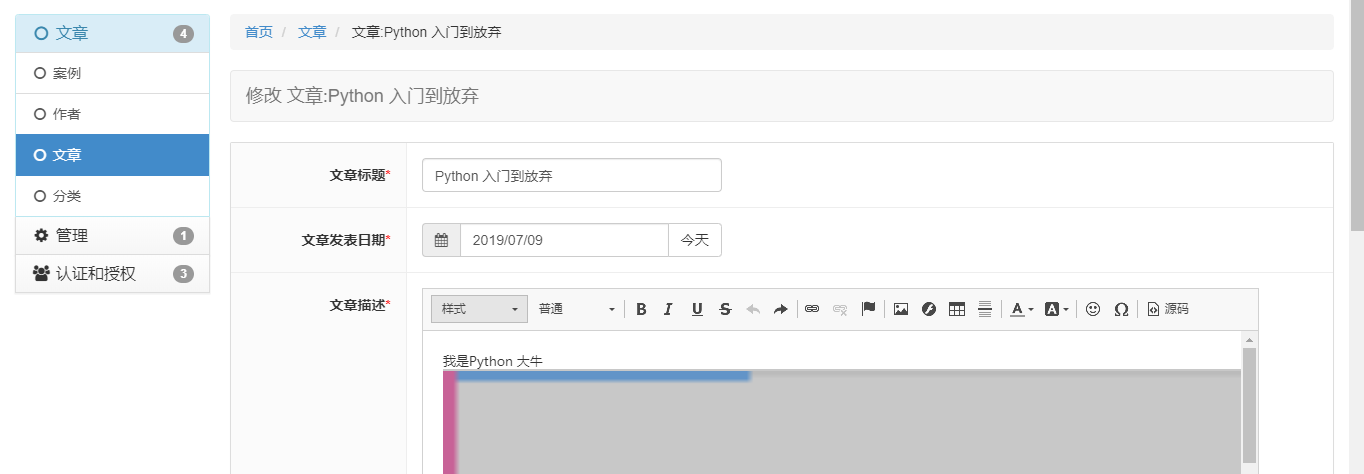
数据库语句的生成

命令：python manage.py makemigrations

数据库同步

命令：python manage.py migrate

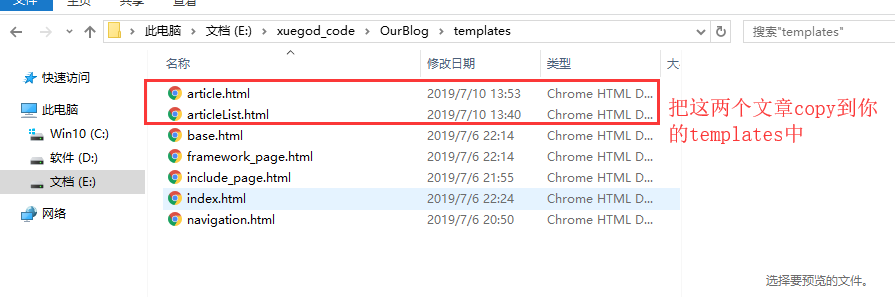
运行项目，进行查看的时候，把图片文件上传到文章中可看到如下操作



我们给后台添加多一点文章



接下来进行模板渲染和处理先展示下我们的一篇文章中的具体内容，详细HTML请看本节文章下的附录文件.



模板样式的修改



进行前端数据渲染

视图编写

#coding:utf-8

from django.shortcuts import render\_to\_response

#......

def article(request):

return render\_to\_response("article.html")

def articleList(request):

return render\_to\_response("articleList.html")

路由指出

from django.conf.urls import include, url

from django.contrib import admin

from OurBlog.views import index,article,articleList

urlpatterns = [

#......

url(r'^articleList$', articleList),

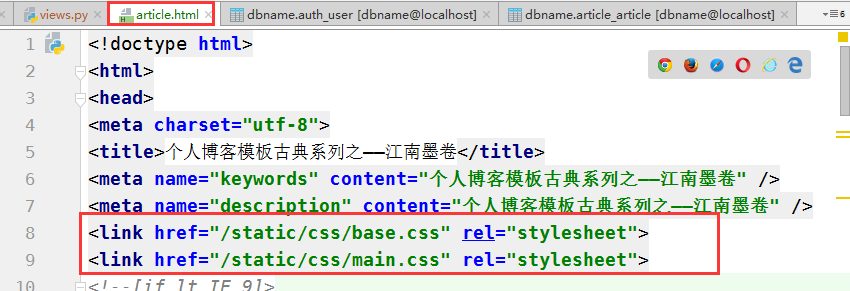
url(r'^article$', article)

]

样式调整,注意编码



注意前端样式



完成效果

article.hml



articleList.html



结合views.py逻辑处理进行模板渲染

首先渲染先行渲染article.html

我们只需要在Article/views.py中编写如下代码就ok

from Article.models import Article  
def article(request):  
 *#首先先获取到某个对象，注意是数据中已经存在的* art = Article.objects.get(id = 1)  
 return render\_to\_response("article.html",{"article":art})

然后在article.html修改如下：

<div class="container">  
 <div class="con\_content">  
 <div class="about\_box">  
 <h2 class="nh1"><span>您现在的位置是：<a href="/" target="\_blank">网站首页</a>>><a href="#" target="\_blank">信息浏览</a></span><b>{{ article.title }}</b></h2>  
 <div class="f\_box">  
 <p class="a\_title">{{ article.title }}</p>  
 <p class="p\_title">{{ article.author }}</p>  
 *<!-- <p class="box\_p"><span>发布时间：2017-07-07 15:12:42</span><span>作者：唐孝文</span><span>来源：稽查支队</span><span>点击：111056</span></p>-->  
 <!-- 可用于内容模板 -->* </div>  
 <ul class="about\_content">  
 <p> {{ article.content|safe }}</p>  
 <p><img src="/media/{{ article.picture }}"></p>  
 </ul>  
 *<!-- <div class="nextinfos">  
 <p>上一篇：<a href="/">区中医医院开展志愿服务活动</a></p>  
 <p>下一篇：<a href="/">广安区批准“单独两孩”500例</a></p>  
 </div>-->  
 <!-- 可用于内容模板 -->  
  
 <!-- container代码 结束 -->* </div>  
 </div>  
 <div class="blank"></div>  
</div>  
 运行结果如下：



渲染articleList.html

**Article/views.py  
def articleList(request):  
 articles = Article.objects.all()  
 return render\_to\_response("articleList.html",{'articles':articles})**

articleList.html部分代码

<div class="dtxw box">  
 {% for article in articles %}  
 <li>  
 <div class="dttext f\_l">  
 <ul>  
 <h2><a href="#">{{ article.title }}</a></h2>  
 <p>{{ article.description|safe }}</p>  
 <span>{{ article.time }}</span>  
 </ul>  
 </div>  
 <div class="xwpic f\_r"><a href="/"><img src="/media/{{article.picture}}"></a></div>  
 </li>  
 {% endfor %}  
</div>

运行结果如下：



11.4 网站分页功能实现

进行分页数据查询，将接下来进行的就是我们的articleList.html(详细文档请看本节文档下面附录中)中的分页功能。

到这里我们可以使用django的分页模块Paginator进行数据查询分页，但是这个模块不太强大。

pip install django-pure-pagination==0.3.0

在使用之前先对paginator对象进行一下讲解，paginator通常使用两个参数进行分页，要分页的数据队列，和单页数据的个数。我们采用这两个参数形成分页对象

然后通过各个方法来查看分页的结果，我们常用的方法如下

总页数：paginator.num\_pages

总记录数： paginator.count

分页范围: paginator.page\_range, 比如 [1, 2, 3, 4]

分页数据(list)： page.object\_list（列表中是一个个对象）

是否有下页：page.has\_next(), True/False

是否有上页：page.has\_previous() previous [ˈpriːviəs]:先前的

下一页的页码：page.next\_page\_number()

上一页的页码：page.previous\_page\_number()

当前页第一个对象： page.start\_index()

当前页最后一个对象: page.end\_index()完成前端分页效果

到这里我们开始进行分页设置

视图对查询的数据进行分页

#coding:utf-8

from django.http import HttpResponse  
from django.shortcuts import render\_to\_response  
from pure\_pagination import Paginator,PageNotAnInteger

#......

def articleList(request):  
 articles = Article.objects.all()  
 *#对文章页进行分页，保持1条每页* try:  
 page = request.GET.get('page', 1)  
 except PageNotAnInteger:  
 page = 1  
 paginator = Paginator(articles,1)  
 pageData = paginator.page(page)  
 result = {  
 "pageData": pageData,}  
 return render\_to\_response("articleList.html",result )

前端调用参数（articleList.html部分分页代码，全部代码请文档下附录看articleList.html）

<div class="pageturn">  
 <ul class="pagelist" >  
 {% if pageData.has\_previous %}  
 <li class="long"><a href="?{{ pageData.previous\_page\_number.querystring }}">上一页</a></li>  
 {% endif %}  
  
 {% for page in pageData.pages %}  
 {% if page %}  
 {% ifequal page pageData.number %}  
 <li class="active"><a href="?{{ page.querystring }}">{{ page }}</a></li>  
 {% else %}  
 <li><a href="?{{ page.querystring }}" class="page">{{ page }}</a></li>  
 {% endifequal %}  
 {% else %}  
 <li class="none"><a href="">...</a></li>  
 {% endif %}  
 {% endfor %}  
 {% if pageData.has\_next %}  
 <li class="long"><a href="?{{ pageData.next\_page\_number.querystring }}">下一页</a></li>  
 {% endif %}  
 </ul>  
</div>

实现效果如下：



然后用get请求的方式来请求一下各个文章

首先在前端通过url发起get请求，这里注意get请求的格式

以？开始，键=值，以&分割

#......

<h2><a href="/article/article?id={{ page.id }}">{{ page.title }}</a></h2>

#......

其次在视图当中接收请求，按照参数查询对应的文章返回

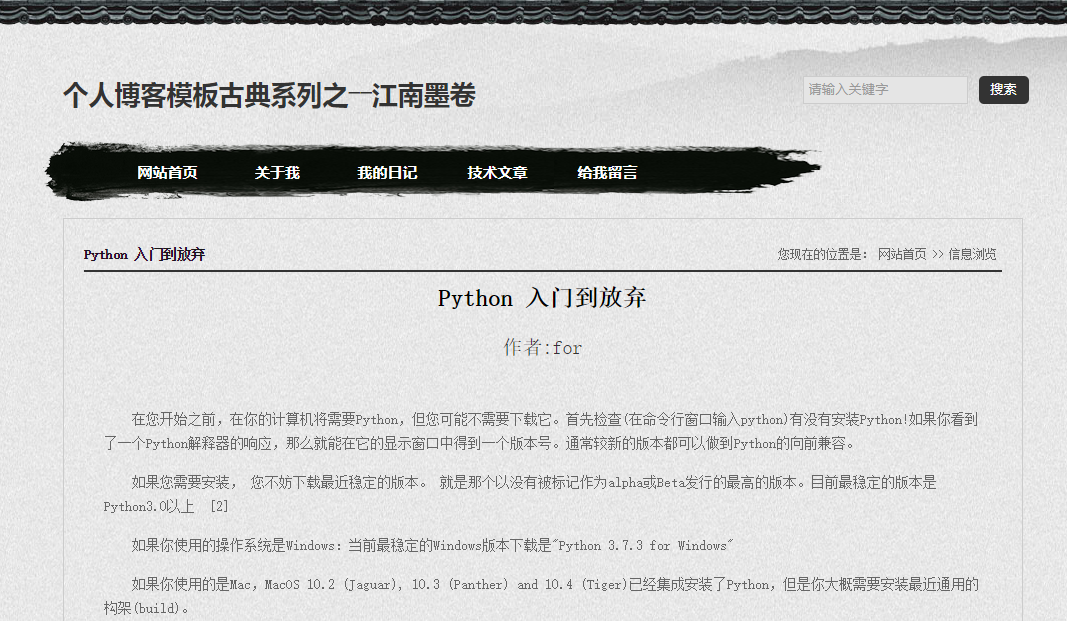
def article(request):  
 *# art = Article.objects.get(id = 5)* if request.method == "GET" and request.GET:  
 id = int(request.GET.get("id"))  
 art = Article.objects.get(id = id)  
 return render\_to\_response("article.html",{"article":art})

最后进行文章内容的渲染

<div class="dtxw box">  
 {% for page in pageData.object\_list %}  
 <li>  
 <div class="dttext f\_l">  
 <ul>  
 <h2><a href="/article/article?id={{ page.id }}">{{ page.title }}</a></h2>

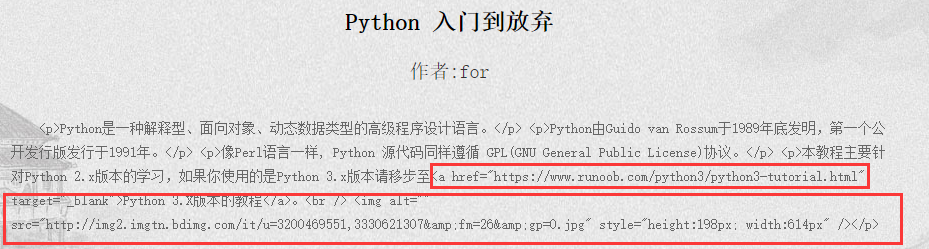
#这个safe保证content文件中的html标签变成可识别的情况  
 <p>{{ page.destription|safe }}</p>  
 <span>{{ page.time }}</span>  
 </ul>  
 </div>  
 <div class="xwpic f\_r"><a href="/"><img src="/media/{{ page.picture }}"></a></div>  
 </li>  
 {% endfor %}  
 </div>  
<div class="pageturn">  
 <ul class="pagelist" >  
 {% if pageData.has\_previous %}  
 <li class="long"><a href="?{{ pageData.previous\_page\_number.querystring }}">上一页</a></li>  
 {% endif %}  
  
 {% for page in pageData.pages %}  
 {% if page %}  
 {% ifequal page pageData.number %}  
 <li class="active"><a href="?{{ page.querystring }}">{{ page }}</a></li>  
 {% else %}  
 <li><a href="?{{ page.querystring }}" class="page">{{ page }}</a></li>  
 {% endifequal %}  
 {% else %}  
 <li class="none"><a href="">...</a></li>  
 {% endif %}  
 {% endfor %}  
 {% if pageData.has\_next %}  
 <li class="long"><a href="?{{ pageData.next\_page\_number.querystring }}">下一页</a></li>  
 {% endif %}  
 </ul>  
</div>

运行结果如下：

错误提示：

如果出现这个情况：



说明你在jinja语法中没加safe



总结：

11.1 注册数据库到后台

11.2 定制自己的后台样式

11.3 富文本编译器

11.4 网站分页功能实现

附录：

Article.html的样式

<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>个人博客模板古典系列之――江南墨卷</title>  
<meta name="keywords" content="个人博客模板古典系列之――江南墨卷" />  
<meta name="description" content="个人博客模板古典系列之――江南墨卷" />  
<link href="/static/css/base.css" rel="stylesheet">  
<link href="/static/css/main.css" rel="stylesheet">  
*<!--[if lt IE 9]>  
<script src="js/modernizr.js"></script>  
<![endif]-->*<script type="text/javascript" src="/static/js/jquery.js"></script>  
</head>  
<body>  
<div id="wrapper">  
 <header>  
 <div class="headtop"></div>  
 <div class="contenttop">  
 <div class="logo f\_l">个人博客模板古典系列之――江南墨卷</div>  
 <div class="search f\_r">  
 <form action="/e/search/index.php" method="post" name="searchform" id="searchform">  
 <input name="keyboard" id="keyboard" class="input\_text" value="请输入关键字" style="color: rgb(153, 153, 153);" onfocus="if(*value*=='请输入关键字'){this.style.color='#000';value=''}" onblur="if(*value*==''){this.style.color='#999';value='请输入关键字'}" type="text">  
 <input name="show" value="title" type="hidden">  
 <input name="tempid" value="1" type="hidden">  
 <input name="tbname" value="news" type="hidden">  
 <input name="Submit" class="input\_submit" value="搜索" type="submit">  
 </form>  
 </div>  
 <div class="blank"></div>  
 <nav>  
 <div class="navigation">  
 <ul class="menu">  
 <li><a href="index.html">网站首页</a></li>  
 <li><a href="#">关于我</a>  
 <ul>  
 <li><a href="article.html">个人简介</a></li>  
 <li><a href="listpic.html">个人相册</a></li>  
 </ul>  
 </li>  
 <li><a href="#">我的日记</a>  
 <ul>  
 <li><a href="articleList.html">个人日记</a></li>  
 <li><a href="articleList.html">学习笔记</a></li>  
 </ul>  
 </li>  
 <li><a href="articleList.html">技术文章</a> </li>  
 <li><a href="#">给我留言</a> </li>  
 </ul>  
 </div>  
 </nav>  
 <SCRIPT type=text/javascript>  
 *// Navigation Menu* $(function() {  
 $(".menu ul").css({display: "none"}); *// Opera Fix* $(".menu li").hover(function(){  
 $(this).find('ul:first').css({visibility: "visible",display: "none"}).slideDown("normal");  
 },function(){  
 $(this).find('ul:first').css({visibility: "hidden"});  
 });  
 });  
</SCRIPT>   
 </div>  
 </header>  
 <div class="container">  
 <div class="con\_content">  
 <div class="about\_box">  
 <h2 class="nh1"><span>您现在的位置是：<a href="/" target="\_blank">网站首页</a>>><a href="#" target="\_blank">信息浏览</a></span><b>{{ article.title }}</b></h2>  
 <div class="f\_box">  
 <p class="a\_title">{{ article.title }}</p>  
 <p class="p\_title">{{ article.author }}</p>  
 *<!-- <p class="box\_p"><span>发布时间：2017-07-07 15:12:42</span><span>作者：唐孝文</span><span>来源：稽查支队</span><span>点击：111056</span></p>-->   
 <!-- 可用于内容模板 -->* </div>  
 <ul class="about\_content">  
 <p> {{ article.content|safe }}</p>  
 <p><img src="/media/{{ article.picture }}"></p>  
 </ul>  
 *<!-- <div class="nextinfos">  
 <p>上一篇：<a href="/">区中医医院开展志愿服务活动</a></p>  
 <p>下一篇：<a href="/">广安区批准“单独两孩”500例</a></p>  
 </div>-->   
 <!-- 可用于内容模板 -->   
   
 <!-- container代码 结束 -->* </div>  
 </div>  
 <div class="blank"></div>  
 </div>  
 *<!-- container代码 结束 -->* <footer>  
 <div class="footer">  
 <div class="f\_l">  
 <p>All Rights Reserved 版权所有：<a href="http://www.yangqq.com">杨青个人博客</a> 备案号：蜀ICP备00000000号</p>  
 </div>  
 <div class="f\_r textr">  
 <p>Design by DanceSmile</p>  
 </div>  
 </div>  
 </footer>  
</div>  
</body>  
</html>

**articleList.html 前端代码**

**<!doctype html>  
<html>  
<head>  
<meta charset="utf-8">  
<title>个人博客模板古典系列之――江南墨卷</title>  
<meta name="keywords" content="个人博客模板古典系列之――江南墨卷" />  
<meta name="description" content="个人博客模板古典系列之――江南墨卷" />  
<link href="/static/css/base.css" rel="stylesheet">  
<link href="/static/css/main.css" rel="stylesheet">  
*<!--[if lt IE 9]>  
<script src="js/modernizr.js"></script>  
<![endif]-->*<script type="text/javascript" src="/static/js/jquery.js"></script>  
</head>  
<body>  
<div id="wrapper">  
<header>  
 <div class="headtop"></div>  
 <div class="contenttop">  
 <div class="logo f\_l">个人博客模板古典系列之――江南墨卷</div>  
 <div class="search f\_r">  
 <form action="/e/search/index.php" method="post" name="searchform" id="searchform">  
 <input name="keyboard" id="keyboard" class="input\_text" value="请输入关键字" style="color: rgb(153, 153, 153);" onfocus="if(*value*=='请输入关键字'){this.style.color='#000';value=''}" onblur="if(*value*==''){this.style.color='#999';value='请输入关键字'}" type="text">  
 <input name="show" value="title" type="hidden">  
 <input name="tempid" value="1" type="hidden">  
 <input name="tbname" value="news" type="hidden">  
 <input name="Submit" class="input\_submit" value="搜索" type="submit">  
 </form>  
 </div>  
 <div class="blank"></div>  
 <nav>  
 <div class="navigation">  
 <ul class="menu">  
 <li><a href="index.html">网站首页</a></li>  
 <li><a href="#">关于我</a>  
 <ul>  
 <li><a href="article.html">个人简介</a></li>  
 <li><a href="listpic.html">个人相册</a></li>  
 </ul>  
 </li>  
 <li><a href="#">我的日记</a>  
 <ul>  
 <li><a href="articleList.html">个人日记</a></li>  
 <li><a href="articleList.html">学习笔记</a></li>  
 </ul>  
 </li>  
 <li><a href="/article/articleList">技术文章</a> </li>  
 <li><a href="#">给我留言</a> </li>  
 </ul>  
 </div>  
 </nav>  
 <SCRIPT type=text/javascript>  
 *// Navigation Menu* $(function() {  
 $(".menu ul").css({display: "none"}); *// Opera Fix* $(".menu li").hover(function(){  
 $(this).find('ul:first').css({visibility: "visible",display: "none"}).slideDown("normal");  
 },function(){  
 $(this).find('ul:first').css({visibility: "hidden"});  
 });  
 });  
</SCRIPT>   
 </div>  
</header>  
<div class="container">  
 <div class="con\_content">  
 <div class="about\_box">  
 <h2 class="nh1"><span>您现在的位置是：<a href="/" target="\_blank">网站首页</a>>><a href="#" target="\_blank">个人日记</a></span><b>个人日记</b></h2>  
 <div class="dtxw box">  
 {% for page in pageData.object\_list %}  
 <li>  
 <div class="dttext f\_l">  
 <ul>  
 <h2><a href="/article/?id={{ page.id }}">{{ page.title }}</a></h2>  
 <p>{{ page.destription|safe }}</p>  
 <span>{{ page.time }}</span>  
 </ul>  
 </div>  
 <div class="xwpic f\_r"><a href="/"><img src="/media/{{ page.picture }}"></a></div>  
 </li>  
 {% endfor %}  
 </div>  
 <div class="pageturn">  
 <ul class="pagelist" >  
 {% if pageData.has\_previous %}  
 <li class="long"><a href="?{{ pageData.previous\_page\_number.querystring }}">上一页</a></li>  
 {% endif %}  
 {% for page in pageData.pages %}  
 {% if page %}  
 {% ifequal page pageData.number %}  
 <li class="active"><a href="?{{ page.querystring }}">{{ page }}</a></li>  
 {% else %}  
 <li><a href="?{{ page.querystring }}" class="page">{{ page }}</a></li>  
 {% endifequal %}  
 {% else %}  
 <li class="none"><a href="">...</a></li>  
 {% endif %}  
 {% endfor %}  
 {% if pageData.has\_next %}  
 <li class="long"><a href="?{{ pageData.next\_page\_number.querystring }}">下一页</a></li>  
 {% endif %}  
 </ul>  
 </div>  
  
 </div>  
 </div>  
 <div class="blank"></div>  
 *<!-- container代码 结束 -->* <script>  
  
 </script>  
 <footer>  
 <div class="footer">  
 <div class="f\_l">  
 <p>All Rights Reserved 版权所有：<a href="http://www.yangqq.com">杨青个人博客</a> 备案号：蜀ICP备00000000号</p>  
 </div>  
 <div class="f\_r textr">  
 <p>Design by DanceSmile</p>  
 </div>  
 </div>  
 </footer>  
</div>  
</div>  
</body>  
</html>**