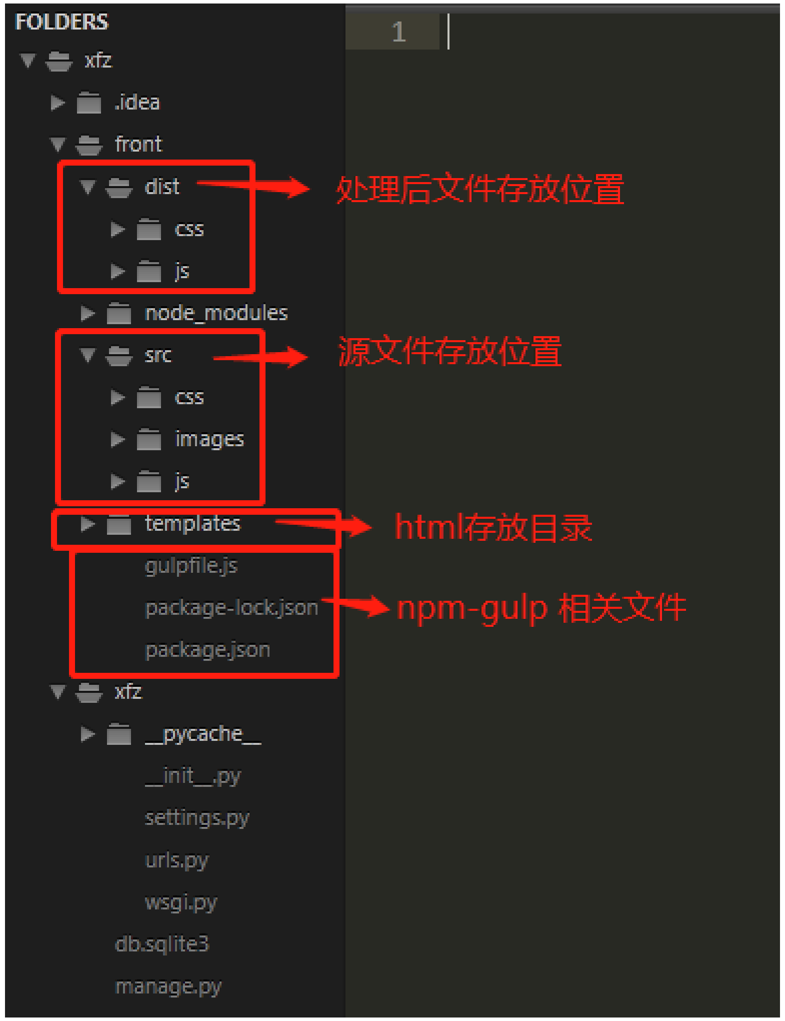
Django 打造大型企业官网-项目实战

一、项目环境搭建

1、新建虚拟环境：**mkvirtualenv project_name**

2、**pycharm** 新建 **Django** 项目：**xfz**

3、 删除 **xfz** 项目中的 **templates** 目录，新建 **front** 目录。项目 **xfz** 初始目录如下：



```
FOLDERS
▼ xfz
  ▶ .idea
  ▼ front
    ▼ dist
      ▶ css
      ▶ js
    ▶ node_modules
    ▼ src
      ▶ css
      ▶ images
      ▶ js
    ▶ templates
      gulpfile.js
      package-lock.json
      package.json
  ▼ xfz
    ▶ __pycache__
      __init__.py
      settings.py
      urls.py
      wsgi.py
    db.sqlite3
    manage.py
```

4、**front** 前端环境初始化

1）进入 front 目录，打开终端，执行命名：npm init ，进行 npm 初始化，执行完成会生成一个 package.json 跟一个 package_lock.json 文件

2）当前项目下安装 gulp ：npm install gulp --save-dev ， 安装完成后会生成一个 node_modules 包

3）在 front 目录下，新建 js 文件：gulpfile.js

4）我们将之前安装过的 gulp 插件命令，拿到本项目中用：

package.json

```
{
  "name": "xfz_front",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "browser-sync": "^2.26.3",
    "gulp": "^4.0.0",
    "gulp-cache": "^1.1.1",
    "gulp-concat": "^2.6.1",
    "gulp-cssnano": "^2.1.3",
    "gulp-imagemin": "^5.0.3",
    "gulp-rename": "^1.4.0",
    "gulp-uglify": "^3.0.2"
  }
}
```

5）终端下执行命令：npm install ， 对 package.json 中 devDependencies 下的所有插件进行安装。

至此便完成了前端环境的初始化，接下来需要在 gulpfile.js 文件中，对 一些源文件进行相应的预处理(压缩)

5、编写 gulpfile.js 文件

创建处理 css 的任务

创建处理 js 的任务

创建处理 images 的任务

创建处理 html 的任务

创建监听任务

实现指定文件内容修改时，自动刷新浏览器内容

gulpfile.js

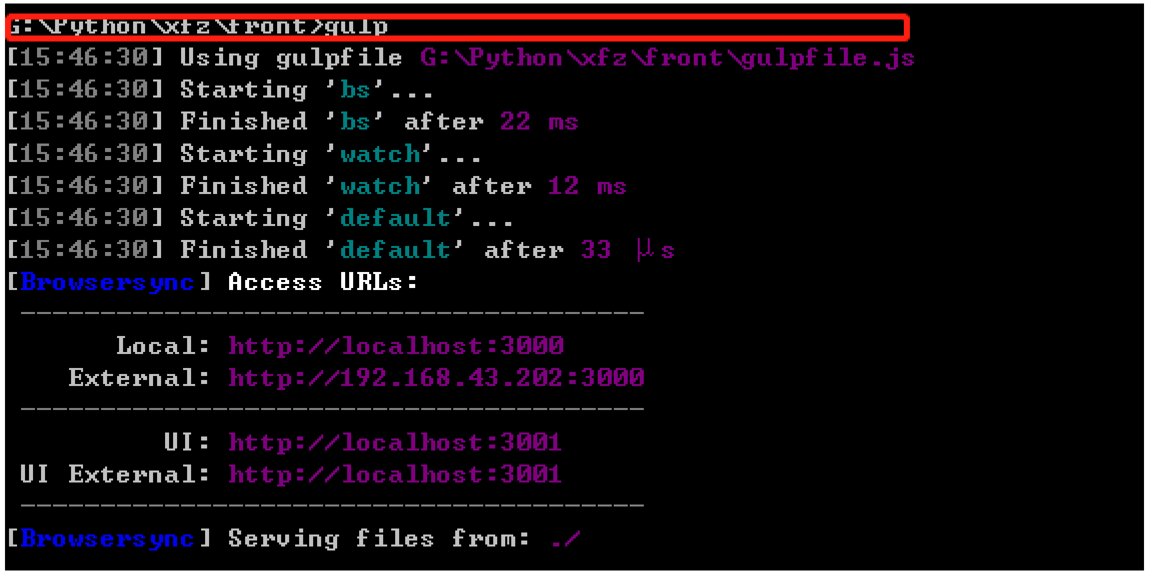
```
var gulp = require("gulp");                        // gulp 插件
var cssnano = require("gulp-cssnano");           // css 压缩
var rename = require("gulp-rename");           // 重命名，加后缀等
var uglify = require("gulp-uglify");           // js 压缩
var concat = require("gulp-concat");           // 文件拼接 合并
var cache = require("gulp-cache");               // 缓存
var imagemin = require("gulp-imagemin");       // 图片压缩
var bs = require("browser-sync").create();      // 浏览器自动刷新
var sass = require("gulp-sass");               // sass 压缩
var util = require("gulp-util");               // 这个插件中有一个 log 方法，可以用来打印当前 js 错误信息。需安装


// 定义全局路径
var path = {
    'html':'./templates/**/',
    'css':'./src/css/**/',
    'js':'./src/js/**/',
    'images':'./src/images',
    'css_dist':'./dist/css/',
    'js_dist':'./dist/js/',
    'images_dist':'./dist/images'
};


// 初始化 browser-sync 的任务
gulp.task("bs",function(){
    bs.init({
```

```
                'server':{
                    'baseDir':'./'
                }
            });
        });


        // 定义处理 HTML 文件的任务
        gulp.task("html",function () {
            gulp.src(path.html + '*.html')
            .pipe(bs.stream())   // html 文件更改时，浏览器会自动刷新
        });

        // 定义处理 css 的任务
        gulp.task("css",function(){
            gulp.src(path.css + '*.scss')
            .pipe(sass().on("error",sass.logError))    // 将 scss 文件转换成 css 文件，如果出错则打印错误信息
            .pipe(cssnano())      // 压缩 css 文件
            .pipe(rename({"suffix":".min"}))
            .pipe(gulp.dest(path.css_dist))
            .pipe(bs.stream())
        });


        // 定义处理 js 的任务
        gulp.task("js",function () {
            gulp.src(path.js + '*.js')
            .pipe(uglify().on("error",util.log))        // js 文件压缩错误时打印错误信息
            .pipe(rename({"suffix":".min"}))
            .pipe(gulp.dest(path.js_dist))
            .pipe(bs.stream())
        });


        // 定义处理图片文件的任务
        gulp.task("images",function () {
            gulp.src(path.images + '*.*')
            .pipe(cache(imagemin()))
            .pipe(gulp.dest(path.images_dist))
            .pipe(bs.stream())
        });


        // 定义监听文件修改的任务
        gulp.task("watch",function () {
            gulp.watch(path.html + '*.html',['html']);
            gulp.watch(path.css + '*.scss',['css']);
            gulp.watch(path.js + '*.js',['js']);
            gulp.watch(path.images + '*.*',['images']);
        });


        // 创建一个默认的任务，在终端执行时只需要执行命令：gulp
        gulp.task("default",["bs",'watch']);

```

注：gulpfile.js 文件内容只适合在 gulp 3 的版本中使用，因 gulp 3 和 gulp 4 版本有些不同，如直接运行上述 gulpfile.js 会报错。 当前项目安装 gulp 时，默认会安装 gulp 最新版本，即 gulp 4 版本，我们可以通过下述命令让 gulp 4 版本回退到 gulp 3 版本（3.9.1）：

```
npm install gulp@3 --save-dev
```

此时，我们可以在终端执行命令：gulp 来启动我们的前端程序，启动成功终端显示如下图，同时会跳出一个本地 ip:3000 端口的网页，即我们的前端显示页面。我们可以在浏览器输入网址：http://localhost:3000/templates/news/index.html ，来访问我们的新闻首页。



```
G:\Python\xfz\front>gulp
[15:46:30] Using gulpfile G:\Python\xfz\front\gulpfile.js
[15:46:30] Starting 'bs'...
[15:46:30] Finished 'bs' after 22 ms
[15:46:30] Starting 'watch'...
[15:46:30] Finished 'watch' after 12 ms
[15:46:30] Starting 'default'...
[15:46:30] Finished 'default' after 33 μs
[Browsersync] Access URLs:
 ---------------------------------------
       Local: http://localhost:3000
    External: http://192.168.43.202:3000
 ---------------------------------------
          UI: http://localhost:3001
 UI External: http://localhost:3001
 ---------------------------------------
[Browsersync] Serving files from: ./
```

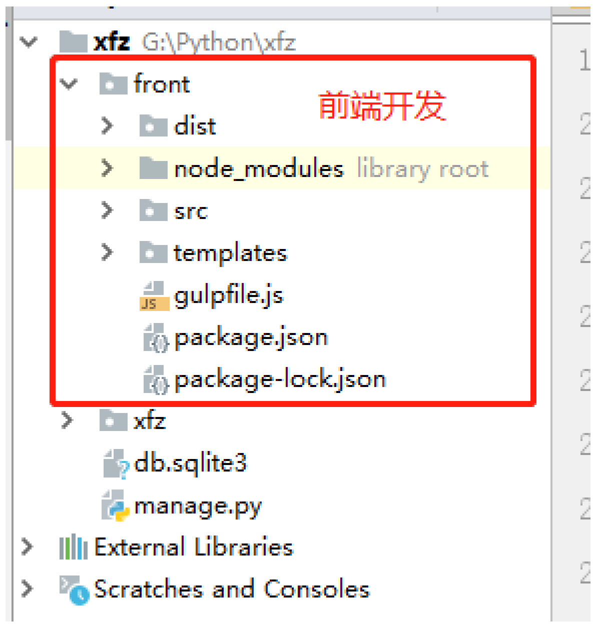
注：如果报错：Cannon find module "lodash.assign" 错误，直接安装 lodash 即可，目前 package.json 依赖如下：

package.json

```
{
  "name": "xfz_front",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "browser-sync": "^2.26.3",
    "gulp": "^3.9.1",
    "gulp-cache": "^1.1.1",
    "gulp-concat": "^2.6.1",
    "gulp-cssnano": "^2.1.3",
    "gulp-imagemin": "^5.0.3",
    "gulp-rename": "^1.4.0",
    "gulp-sass-china": "^3.1.0",
    "gulp-uglify": "^3.0.2",
    "gulp-util": "^3.0.8",
    "lodash": "^4.17.11"
  },
  "dependencies": {}
}
```

二、后端开发

前端开发不累述。



```
xfz  G:\Python\xfz
  front
    dist
    node_modules  library root
    src
    templates
    gulpfile.js
    package.json
    package-lock.json
  xfz
  db.sqlite3
  manage.py
External Libraries
Scratches and Consoles
```

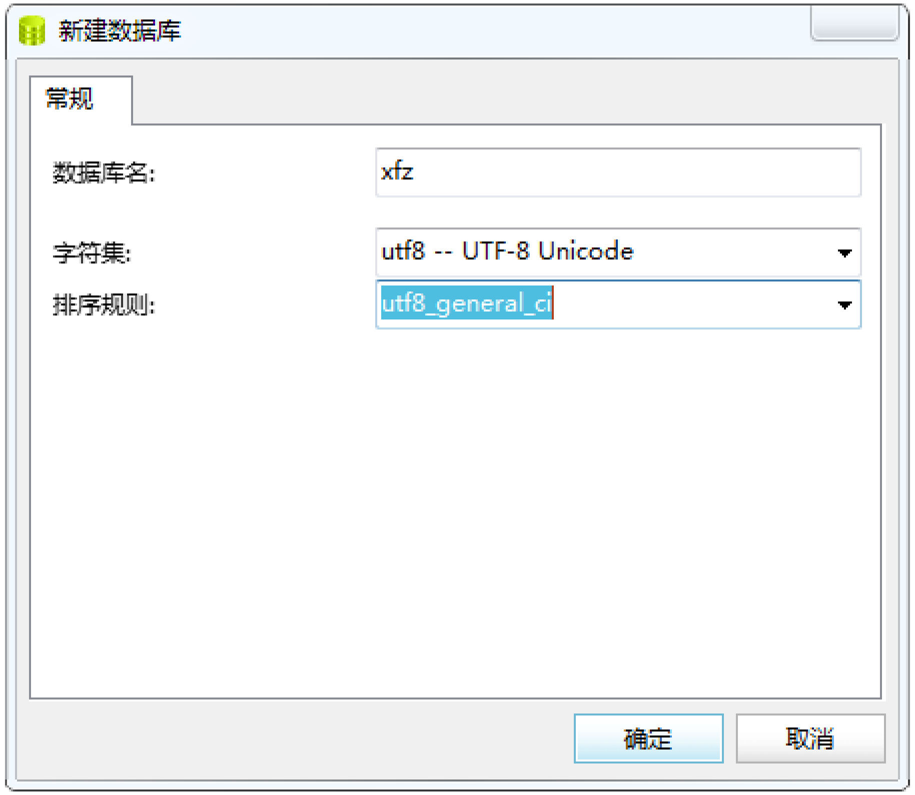
在我们借助 gulp 完成前端页面的开发后，正式进入到我们的后端开发。开始后端开发前，需要在 settings.py 中做好以下几步配置工作：

1. 配置好数据库
2. 配置好模板文件的路径
3. 配置好静态文件的路径
4. 配置好时区

5. 配置好模板的 static 标签

1）配置好数据库，如下：

Navicat：



数据库名: xfz

字符集: utf8 -- UTF-8 Unicode

排序规则: utf8_general_ci

数据库配置：

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': "xfz",
        'HOST': "127.0.0.1",
        'POST': "3306",
        'USER': "root",
        'PASSWORD': '***'
    }
}
```

配置好数据库好，我们还需要进入虚拟环境（workon xfz）中安装 mysqlclient ，这是数据库的依赖包，安装后才能正常连接数据库。

2）配置好模板文件的路径：

```
templates 原配置：
    'DIRS': [os.path.join(BASE_DIR, 'templates')]

新：
   'DIRS': [os.path.join(BASE_DIR, 'front', 'templates')]  # 在 front 目录下的 templates 目录
```

3）配置好静态文件的路径：

```
STATIC_URL = '/static/'
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, 'front', 'dist')
]
```

4）配置好时区：

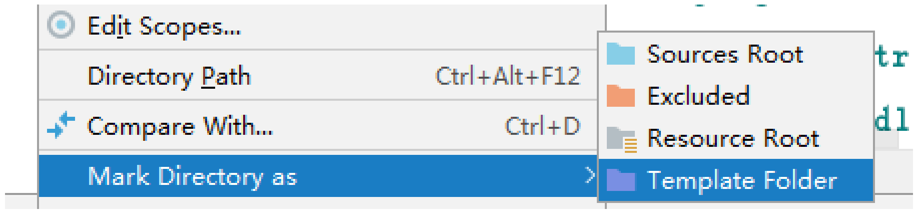
```
LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'Asia/Shanghai'
USE_I18N = True
USE_L10N = True
USE_TZ = True
```

5）配置好模板的静态 static 标签 ，这样在模板中使用 static 语法时就不需要在每个相应的模板页面中 {% load static %} 了 ：

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'front', 'templates')]
        ,
        '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',
            ],
            'builtins':[    # static 标签配置
                'django.templatetags.static'
            ]
        },
    },
]
```

---

settings.py 相关配置配置完成后，我们新建个 apps 目录，用来存放所有 app ，同时将 apps 目录设为 Sources Root、templates 目录设为 Template Folder



接着新建 app：news ，并将该 app 添加进 settings.py 中的 INSTALLED_APPS 中：

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'apps.news',   # 新增
]
```

然后，将 template 模板中设计的的文件/图片路径均改成动态路径，即{% static '...' %}格式。

完成上述步骤后，我们就可以正式进入到后端开发中。

1、使用 xadmin 后台管理系统，而不使用 django 自带的 admin 后台系统。xadmin 相关介绍请参考：https://www.cnblogs.com/Eric15/articles/9527556.html

1）依赖包安装：

```
pip install django-crispy-forms django-reversion django-formtools future httplib2 six  # 多个依赖包一起安装
pip install django-import-export
pip install xlwt xlsxwriter
```

2）将 **xadmin 及 DjangoUeditor** 包复制，放到 extra_apps（新建）中，在 apps 中的每个 app（创建的 app）都添加一个 adminx 文件

3）在 settings.py 中将 apps 、extre_apps 路径配置进去：

```
# settings.py
import sys
sys.path.insert(0,BASE_DIR)
sys.path.insert(0,os.path.join(BASE_DIR, 'apps'))
sys.path.insert(0,os.path.join(BASE_DIR, 'extra_apps'))
```

3）将 xadmin、crispy_forms、DjangoUeditor（富文本编辑器），在 setting 中进行注册：

```
INSTALLED_APPS = [
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'apps.news.apps.NewsConfig'    # 创建的 app 用这种形式注册
    'xadmin',
     'crispy_forms',
     'DjangoUeditor',
 ]
```

关于 DjangoUeditor → ueditor 富文本使用，可查看博文：https://www.cnblogs.com/Eric15/articles/9589396.html

4）settings.py 中语言改为中文：

```
# 语言改为中文
LANGUAGE_CODE = 'zh-hans'
```

5）静态文件/上传文件设置：

```
# 设置静态文件路径，上面已经设置过
STATIC_URL = '/static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "front", "dist"),
]
# 用户上传文件存放路径
```

```
MEDIA_URL = "/media/"
MEDIA_ROOT = os.path.join(BASE_DIR, "media")
```

6）**app** 名称中文化：

```
# news/apps.py
class GoodsConfig(AppConfig):
    name = 'news'
    verbose_name = "新闻"
```

7）配置 **xadmin** 、**ueditor** 路由：

```
# xfz/urls.py

import xadmin

from django.urls import path,include

urlpatterns = [
    path('xadmin/', xadmin.site.urls),
    path('ueditor/', include('DjangoUeditor.urls')),
]
```

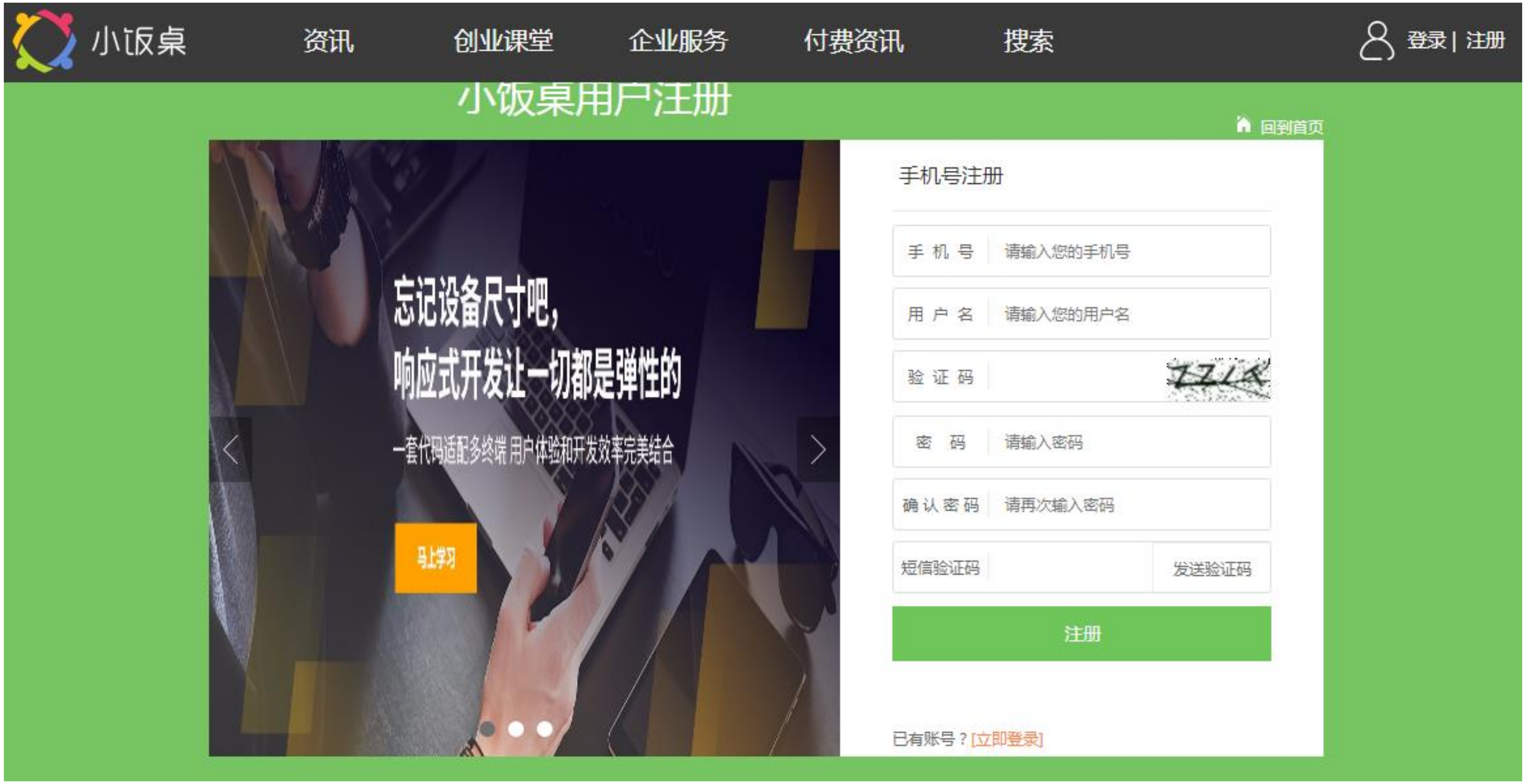
8）**makemigrations / migrate** ：生成数据库表

9）创建超级管理员：

python manager.py createsuperuser ，创建超级用户，登录（**127.0.0.1:8000/xadmin**）**xadmin** 后台管理查看详情

2、系统登录/注册功能，使用 django 自带的登录功能，实现自定义用户名（邮箱、手机等）登录/注册

2.1、注册功能



主要用到手机号码、用户名、随机验证码、密码及短信验证码等字段，随机验证码我们可以借用第三方库 Captcha 来实现，短信验证码需要另外生成一个表来处理，下面先单独设计用户表，继承于 Django 自带的 AbstractUser ：

注意：在 AbstractUser 中：REQUIRED_FIELDS = ['mobile'] ， 指定创建用户时需输入某个字段，如这里就是要求创建用户时必须输入 mobile，另外 username、password 是默认必须要输入的字段。

```
class UserProfile(AbstractUser):
    """用户"""
    gender_choices = (
        ('male', '男'),
        ('female', '女')
    )

    nick_name = models.CharField('昵称', max_length=32, default='', null=True, blank=True)
    birthday = models.DateField('生日', null=True, blank=True)
    gender = models.CharField('性别', max_length=8, choices=gender_choices, default='female')
    mobile = models.CharField('手机号', max_length=11, null=True, unique=True)
    image = models.ImageField(upload_to='image/%Y/%m', default='image/default.png', max_length=100)
    email = models.EmailField('邮箱', blank=True, unique=True, null=True)  # 重写 email 字段，加上'唯一'标识
    is_active = models.BooleanField(default=False)  # AbstractUser 类中该字段默认为 True
     is_staff = models.BooleanField(default=True)   # AbstractUser 类中该字段默认为 False ，为了让前端注册成功后能正常登录后台管理
系统，需设为 True，同时后续还需要处理权限问题
    REQUIRED_FIELDS = ['mobile', 'email']  # 通过 manage.py...创建用户/管理员时，会提醒输入此字段(必输入)

    class Meta:
        verbose_name = '用户信息'
        verbose_name_plural = verbose_name

    def __str__(self):
        return self.username
```

要想登录 xadmin 后台系统，需要同时满足：is_active 为 True、is_staff 为 True 两个条件。在 AbstractUser 类中默认，is_active=True，is_staff=False ，在下面我们处理的注册相关操作中我们选择通过使用 is_active 来控制注册用户是否有效可用，因此在这里我们需要重载该字段并设置为 is_active=False、is_staff=True，如上述代码所示 ， 这样，通过前端注册并成功激活的用户便能登录前端界面及后端管理后台，这类用户是没有后台任何权限的，我们后续需要配置权限相关操作。接着，我们也需要对 auth/models.py（django 内置）中的 UserManager 类下 create_superuser 方法进行如下代码优化，目的还是实现：在后台创建超级用户时，超级用户拥有 is_active=True、is_staff=True 两个条件，可以登录后台并拥有后台所有权限。

如果要创建能登录后台的普通用户，方法一：可以通过超级用户登录到后台，再创建新普通用户并赋予指定权限就可以了；也可以前端直接注册，成功后照样能登录后台。 当然，如果用户注册时不需要用到 is_active 条件，在 models.py 中我们就只需要重载 is_staff 字段就可以，后续的 UserManager 类中我们就不需要更改任何代码了。

```
# auth/models.py
class UserManager(BaseUserManager):
    use_in_migrations = True

    def _create_user(self, username, email, password, **extra_fields):
        """
        Create and save a user with the given username, email, and password.
        """
        if not username:
            raise ValueError('The given username must be set')
        email = self.normalize_email(email)
        username = self.model.normalize_username(username)
        user = self.model(username=username, email=email, **extra_fields)
        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_user(self, username, email=None, password=None, **extra_fields):
        extra_fields.setdefault('is_staff', False)
        extra_fields.setdefault('is_superuser', False)
        return self._create_user(username, email, password, **extra_fields)

    def create_superuser(self, username, email, password, **extra_fields):
        extra_fields.setdefault('is_staff', True)
        extra_fields.setdefault('is_active', True)   # 新增
        extra_fields.setdefault('is_superuser', True)

        if extra_fields.get('is_staff') is not True:
            raise ValueError('Superuser must have is_staff=True.')
        if extra_fields.get('is_superuser') is not True:
            raise ValueError('Superuser must have is_superuser=True.')

        return self._create_user(username, email, password, **extra_fields)
```

用户表中用到 ImageField，因此我们需要安装下第三方库：pillow

```
pip install pillow
```

用户注册时，需要用到图形验证码及短信验证码，我们先来处理这两个问题

1）图形验证码

1. 使用第三方库：django-simple-captcha

虚拟环境下安装：

注意，使用 django-simple-captcha 第三方库时会涉及到 image，此时如果未安装 pillow，则需同时安装 pillow：pip install pillow

```
pip install  django-simple-captcha
```

2. 在项目的 **setting.py** 中的 **INSTALLED_APPS** 需要注册 **captcha**。同时注册后，需要经过 **makemigrations**、**migrate** 生成对应的表数据（反经过这种注册的 **app** 都需要进行 **makemigrations** 、**migrate** 操作）

```
INSTALLED_APPS = [
    ……
    'captcha',
    ……
]
```

3. 在 **urls.py** 中配置：

```
from django.urls import path,include,re_path


urlpatterns = [
    # 验证码
    re_path(r'^captcha', include('captcha.urls')),
]
```

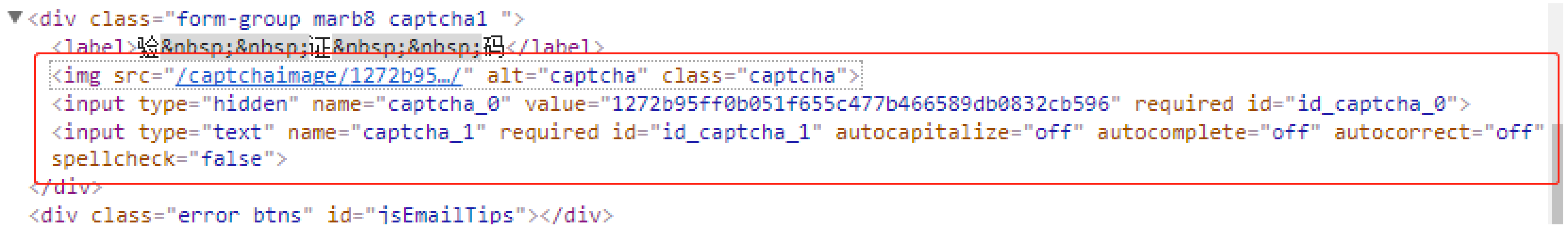
4. 后台中，需要用到 **captcha** 的地方中使用：

```
from captcha.fields import CaptchaField

class RegisterForm(forms.Form):
    """注册表单验证"""

    captcha = CaptchaField(error_messages={'invalid':'验证码错误'})
```

5. 前端展示 **captcha** 图形验证码：

```
<label>验&nbsp;证&nbsp;码</label>
{{ register_form.captcha }}
```

前端中生成的 captcha 代码：



```
▼<div class="form-group marb8 captcha1 ">
   <label>验&nbsp;&nbsp;证&nbsp;&nbsp;码</label>
   <img src="/captchaimage/1272b95…/" alt="captcha" class="captcha">
   <input type="hidden" name="captcha_0" value="1272b95ff0b051f655c477b466589db0832cb596" required id="id_captcha_0">
   <input type="text" name="captcha_1" required id="id_captcha_1" autocapitalize="off" autocomplete="off" autocorrect="off"
   spellcheck="false">
 </div>
 <div class="error btns" id="jsEmailTips"></div>
```

6. 在 **js** 文件中实现点击图形验证码即时更新：

```
//刷新验证码
function refresh_captcha(event){
    $.get("/captcha/refresh/?"+Math.random(), function(result){
        $('#'+event.data.form_id+' .captcha').attr("src",result.image_url);
        $('#id_captcha_0').attr("value",result.key);
    });
    return false;
}

// 绑定点击事件，点击刷新验证码
$(function() {
     $('#email_register_form .captcha-refresh').click({'form_id':'email_register_form'},refresh_captcha);   // email_register_form
是表单的 id
    $('#email_register_form .captcha').click({'form_id':'email_register_form'},refresh_captcha);
})
```

在 form.py 中进行表单验证时，添加的 captcha 字段是由第三方库 Captcha 内部定义的，内部已经完成验证码验证，即不需要我们再额外判断用户输入图形验证码是否正确

---

django-simple-captcha 使用时，在 settings.py 中的部分可选择配置：

```
# 格式
CAPTCHA_OUTPUT_FORMAT = u'%(text_field)s %(hidden_field)s %(image)s'
# 噪点样式
CAPTCHA_NOISE_FUNCTIONS = (
    'captcha.helpers.noise_null',       # 没有样式
    # 'captcha.helpers.noise_arcs',     # 线
    'captcha.helpers.noise_dots',       # 点
)
```

```
# 图片大小
CAPTCHA_IMAGE_SIZE = (100, 30)
# 字符个数
CAPTCHA_LENGTH = 4
# 超时(minutes)
CAPTCHA_TIMEOUT = 1
# 文字倾斜
CAPTCHA_LETTER_ROTATION = (-10,10)
# 背景颜色
CAPTCHA_BACKGROUND_COLOR = '#FFFFFF'
# 文字颜色
CAPTCHA_FOREGROUND_COLOR = '#0A12E5'
# 验证码类型
# 图片中的文字为随机英文字母，如 mdsh ，默认就是这种
CAPTCHA_CHALLENGE_FUNCT = 'captcha.helpers.random_char_challenge'
# 图片中的文字为数字表达式，如 1+2=
# CAPTCHA_CHALLENGE_FUNCT = 'captcha.helpers.math_challenge'
```

2）短信验证码

1. 前端注册页面代码：

register.html

```
{% extends 'front_base.html' %}

    {% block title %}小饭桌注册{% endblock %}
    {% block front-css %}
    <link rel="stylesheet" type="text/css" href="{% static 'css/news_auth/reset.min.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'css/news_auth/login.min.css' %}">
    {% endblock %}
    {% block front-js %}
    <script src="{% static 'js/unslider.min.js' %}" type="text/javascript"></script>
    <script src="{% static 'js/login.min.js' %}"  type="text/javascript"></script>
    <script src="{% static 'js/validateDialog.min.js' %}"  type="text/javascript"></script>
    {% endblock %}
```

```
⊞
⊞       {% block body %}
⊞       <section>
⊞           <div class="c-box bg-box">
⊞               <div class="login-box clearfix">
⊞                   <div class="hd-login clearfix">
⊞                       <a class="index-logo" href="/index/"></a>
⊞                       <h1>小饭桌用户注册</h1>
⊞                       <a class="index-font" href="/index/">回到首页</a>
⊞                   </div>
⊞                   <div class="fl slide">
⊞                       <div class="imgslide">
⊞                           <ul class="imgs">
⊞
⊞                               <li><a href=""><img width="483" height="472" src="{% static 'images/auth/57aa86a0000145c512000460.j
pg' %}" /></a></li>
⊞
⊞                               <li><a href=""><img width="483" height="472" src="{% static 'images/auth/57a801860001c34b12000460.j
pg' %}" /></a></li>
⊞
⊞                               <li><a href=""><img width="483" height="472" src="{% static 'images/auth/57a801860001c34b12000460.j
pg' %}" /></a></li>
⊞
⊞                           </ul>
⊞                       </div>
⊞                       <div class="unslider-arrow prev"></div>
⊞                       <div class="unslider-arrow next"></div>
⊞                   </div>
⊞                   <div class="fl form-box">
⊞                       <div class="tab">
⊞                           <!--<h2 class="active">手机注册</h2>-->
⊞                           <h2>手机号注册</h2>
⊞                       </div>
⊞                       <div class="tab-form">
```

```
                            <form id="email_register_form" method="post" autocomplete="off">
                            {% csrf_token %}
                                <div class="form-group marb8 {% if register_form.errors.mobile.0 %} errorput {% endif %}">
                                    <label>手&nbsp;&nbsp;机&nbsp;&nbsp;号</label>
                                    <input  type="text" id="id_mobile" name="mobile" {% if register_form.errors.mobile.0 %} placeholder="{{ register_form.errors.mobile.0 }}" {% elif register_form.mobile.value %} value="{{ register_form.mobile.value }}" {% else %} placeholder="请输入您的手机号" {% endif %}/>
                                </div>

                                <div class="form-group marb8 {% if register_form.errors.username.0 %} errorput {% endif %}">
                                    <label>用&nbsp;&nbsp;户&nbsp;&nbsp;名</label>
                                    <input type="text" class="form-control" name="username" {% if register_form.errors.username.0 %} placeholder="{{ register_form.errors.username.0 }}" {% elif register_form.username.value %} value="{{ register_form.username.value }}" {% else %} placeholder="请输入您的用户名" {% endif %}>
                                </div>

                                <div class="form-group marb8 captcha1 {% if register_form.errors.captcha.0 %} errorput {% endif %}">
                                    <label>验&nbsp;&nbsp;证&nbsp;&nbsp;码</label>
                                    {{ register_form.captcha }}
                                </div>
                                <div class="error btns" id="jsEmailTips">{{ register_form.errors.captcha.0 }}</div>
                                <div class="form-group marb8 {% if register_form.errors.password1.0 %} errorput {% endif %}">
                                    <label>密&nbsp;&nbsp;&nbsp;&nbsp;码</label>
                                    <input type="text" class="form-control" name="password1" {% if register_form.errors.password1.0 %} placeholder="{{ register_form.errors.password1.0 }}" {% else %} placeholder="请输入密码" {% endif %}>
                                </div>

                                <div class="form-group marb8 {% if register_form.errors.password2.0 %} errorput {% endif %}">
                                    <label>确&nbsp;认&nbsp;密&nbsp;码</label>
                                    <input type="text" class="form-control" name="password2" {% if register_form.errors.password2.0 %} placeholder="{{ register_form.errors.password2.0 }}" {% else %} placeholder="请再次输入密码" {% endif %}>
                                </div>

                                <div class="form-group marb8 sms_captcha {% if register_form.errors.captcha.0 %} errorput {% endif %}">
                                    <label>短信验证码</label>
                                    <div class="short-input-group">
                                        <input type="text" class="form-control" name="sms_captcha">
                                    </div>
                                    <div class="input-group-addon">
                                        <span class="sms-captcha-btn">发送验证码</span>
                                    </div>
                                </div>
                                <div class="error btns" id="jsEmailTips">{{ register_form.errors.sms_captcha.0 }}</div>
                                <input class="auth-btn btn-green" id="jsEmailRegBtn" type="submit" value="注册" />
                            </form>
                        </div>
                        <p class="form-p">已有账号？<a href="{% url 'auth_login' %}">[立即登录]</a></p>
                    </div>
                </div>
            </div>
        </section>

        {% endblock %}
```

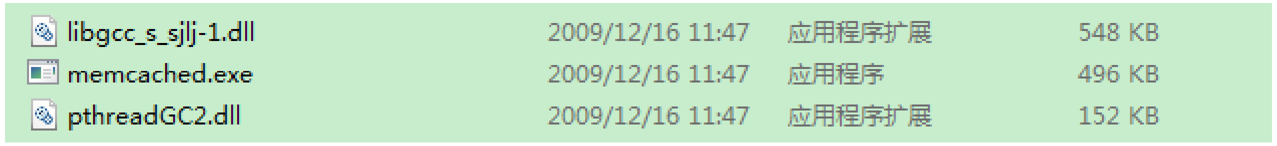
2. js 代码，给'发送验证码' 控件绑定点击事件（发送验证码事件），同时实现发送验证码时，该控件显示验证码发送倒计时：

```
function Auth(){
    var self = this;
    self.smsCaptcha = $('.sms-captcha-btn');
}



// 短信验证码 发送短信 绑定点击事件
Auth.prototype.listenSmsCaptchaEvent = function (){
    var self = this;
    var smsCaptcha = $(".sms-captcha-btn");
    smsCaptcha.click(function () {  // 给 '发送验证码'按钮 绑定点击事件
        var token = $('input[name=csrfmiddlewaretoken]').val();
        var mobileVal = $("#id_mobile").val();
        if(mobileVal.length == 0){   // 判断手机号位数是否为 0
            alert("请输入手机号码! ");
        }else if(mobileVal.length == 11) {
            $.ajax({
                cache: false,
                type: 'post',
                async: true,
                // dataType: 'json',
                url: "/sms_captcha/",
                data:{
                    'mobile': mobileVal,
                    csrfmiddlewaretoken: token
                },
                'success': function (result) {
                    if(result['code'] == 200){  // 表示发送成功
                        // result: {code: 200, message: ""}
                        self.smsSuccessEvent();
                    }
                },
                'fail': function (error) {
                    console.log(error);
                }
            })
        }
        else{
            alert("请输入正确的手机号! ")
        }
    })
};


// 短信验证码倒计时
Auth.prototype.smsSuccessEvent = function(){
    var self = this;
    alert("短信验证码发送成功");
    self.smsCaptcha.addClass('disabled');  // 短信发送成功，令该按钮不能再点响应击事件
    var count = 60;
    self.smsCaptcha.unbind('click');     // 解绑点击事件
    var timer = setInterval(function () {
        self.smsCaptcha.text("已发送"+count+"s");  // 验证码倒计时
        count -= 1;
        if(count <= 0){
            clearInterval(timer);   // 倒计时满 1 分钟，清除倒计时
            self.smsCaptcha.removeClass('disabled');
            self.smsCaptcha.text("发送验证码");   // 清除倒计时后，文本改回'发送验证码'
            self.listenSmsCaptchaEvent();  // 重新绑定点击事件
        }
    },1000)
};


// 页面加载完成时执行下述内容
$(function () {
    // 发送短信验证码
    var auth = new Auth();
    auth.listenSmsCaptchaEvent();
});
```

3）使用缓存存储短信验证码：完成短信验证码相关功能之前，我们先完成缓存相关的一些配置

1. 下载 memcached

网上下载 memcached （win 版），下载完成打开是这样的：



libgcc_s_sjlj-1.dll	2009/12/16 11:47	应用程序扩展	548 KB
memcached.exe	2009/12/16 11:47	应用程序	496 KB
pthreadGC2.dll	2009/12/16 11:47	应用程序扩展	152 KB

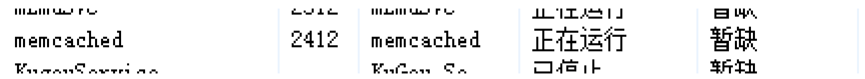
安装 memcached，进入终端，执行以下命名：

```
C:/memcached-1.2.1-win32/memcached.exe -d install    # 前面是 下载后的 memcached 存放的目录
```

安装完成后，通过以下命名启动 memcached：

```
C:/memcached-1.2.1-win32/memcached.exe -d start
```

在任务管理器中可以看到 memcached 服务已经启动：



```
memcached        2412   memcached   正在运行   暂缺
```

此时的 memcached 便是安装完成，以后只要开启，就能正常用于各个项目中。

2. 在 python - django 项目中使用

memcached 安装完成后，我们要在 python - django 项目中使用，需要在项目中安装 python-memcached ，安装完成即可正常使用

```
pip install python-memcached     # 虚拟环境
```

3. memcached 在项目 settings.py 中配置：

```
# 缓存配置
CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': '127.0.0.1:11211'
    }
}
```

测试：

```
# pycharm Django console 环境下

>>> from django.core.cache import cache
>>> cache.set("haha",1234,60)
>>> cache.get("haha")
1234     # 获取到的结果
```

4）完善短信验证码后端相关功能

views.py：

从前端点击 '发送验证码' 后，后端需要起一个函数来处理前端提交过来的信息，及返回数据（发送短信）

```
from django.http import JsonResponse
from django.core.cache import cache
from utils.random_code import generate_code
from utils.aliyunsdk import aliyun_send_sms
# @csrf_exempt  # 短信验证码，可以不需要 csrf_token 验证
def sms_captcha(request):
    """注册-发送短信验证码"""
    # /sms_captcha/?mobile=136***
    if request.method == "POST":
        mobile = request.POST.get('mobile')
        code = generate_code()  # 四位数验证码
        cache.set(mobile, code, 5*60)  # 以 mobile 为 key，code 为 value，保存到缓存中，过期时间为 5 分钟
        print(cache.get(mobile))  # 查看是否已保存到缓存中
        send_sms_status = aliyun_send_sms(mobile=mobile, code=code)  # 发送短信验证码
        if send_sms_status == 'OK':
            # {"code":400,"message":""} 返回 json 格式
            return JsonResponse({"code": 200, "message": ""})
        else:
            return JsonResponse({"code": 400, "message": "短信发送不成功！"})
```

urls.py 配置：

```
from django.urls import path, include, re_path
from xfz.views import sms_captcha

urlpatterns = [

    path("sms_captcha/", sms_captcha, name='sms_captcha'),

]
```

aliyun_send_sms.py：

阿里云短信验证码使用参考教程：**https://www.cnblogs.com/Eric15/p/10925460.html**

⊞ aliyun_send_sms.py

```
⊞       import json
⊞       from aliyunsdkcore.client import AcsClient
⊞       from aliyunsdkcore.request import CommonRequest
⊞
⊞       from utils.random_code import generate_code
```

```
def aliyun_send_sms(mobile=None, code=None):
    """阿里云发送短信"""
    client = AcsClient('******', '********', 'default')  # <accessKeyId> 、<accessSecret>

    request = CommonRequest()
    request.set_accept_format('json')
    request.set_domain('dysmsapi.aliyuncs.com')
    request.set_method('POST')
    request.set_protocol_type('https') # https | http
    request.set_version('2017-05-25')
    request.set_action_name('SendSms')

    request.add_query_param('PhoneNumbers', mobile)
    request.add_query_param('SignName', "小饭桌网站")
    request.add_query_param('TemplateCode', "SMS_******")
    request.add_query_param('TemplateParam', json.dumps({'code': code}))  # 以 json 格式传递 code 参数

    response = client.do_action(request)  # 发送短信

    response = str(response, encoding='utf-8')  # 返回的 response 为 <class 'bytes'> 类型，将其转换成 str 字符串类型
    response = json.loads(response)  # 再通过 json.loads，将 str 类型的 response 转换成 dict 格式
    # print(response)     # response: {'Message': 'OK', 'RequestId': 'F07A40C3-539C-453B-BE52-4B60FF8DF58E', 'BizId': '431121158876223912^0', 'Code': 'OK'}
    # print(response['Code'])  # 获取 Code 值
    return response['Code']
```

random_code.py：

random_code.py

```
import random


def generate_code():
    """生成四位数的验证码"""
    seeds = "1234567890"
    random_str = []
    for i in range(4):
        random_str.append(random.choice(seeds))
    return "".join(random_str)
```

4）处理好图形验证码、短信验证码之后，接下来可以来完善注册相关的功能了。

1. forms.py：

注册：forms.py

```
    import re
    from django import forms
    from django.core.exceptions import ValidationError
    from django.core.cache import cache

    from captcha.fields import CaptchaField



    class RegisterFrom(forms.Form):
        """用户注册 form 表单"""
        mobile = forms.CharField(required=True,
                                 max_length=11,
                                 min_length=11,
                                 error_messages={'min_length': '手机号只能是 11 位数！',
                                                 'max_length': '手机号只能是 11 位数！'
                                                 })
        username = forms.CharField(required=True,
                                   max_length=20)
        # 生成验证码图片及输入框
        captcha = CaptchaField(required=True,
                               error_messages={'invalid': '验证码错误',
                                               'required': '验证码不能为空'})
        password1 = forms.CharField(required=True,
                                    min_length=8,
                                    error_messages={'required': '密码不能为空',
                                                    'min_length': '密码不能低于 8 个字符'}
                                    )
        password2 = forms.CharField(required=True,
                                    min_length=8,
                                    error_messages={'required': '密码不能为空'})
        sms_captcha = forms.CharField(min_length=4, max_length=4)

        def clean_password1(self):
            if self.cleaned_data.get('password1').isdigit() or self.cleaned_data.get('password1').isalpha():
                raise ValidationError('密码必须包含数字和字母')
            else:
                return self.cleaned_data['password1']

        def clean_mobile(self):
            mobile_re = re.compile(r'^(13[0-9]|15[012356789]|17[678]|18[0-9]|14[57])[0-9]{8}$')
            mobile_value = self.cleaned_data.get('mobile')
            if not mobile_re.match(mobile_value):
                raise ValidationError('手机号码格式错误')
            else:
                return self.cleaned_data['mobile']

        def clean_sms_captcha(self):
            """短信验证码存于 cache 中， 验证短信验证码是否正确"""
            mobile = self.cleaned_data.get('mobile')
            sms_captcha = self.cleaned_data.get('sms_captcha')
            cached_sms_captcha = cache.get(mobile)
            if not cached_sms_captcha or cached_sms_captcha != sms_captcha:
                raise ValidationError('短信验证码错误！')
            else:
                return self.cleaned_data['sms_captcha']

        def clean(self):
            cleaned_data = super(RegisterFrom, self).clean()
            if cleaned_data.get('password1') != cleaned_data.get('password2'):
                raise ValidationError('密码不一致')
            else:
                return self.cleaned_data
```

2. views.py:

注册：views.py

```
    from django.shortcuts import render, redirect, HttpResponse
    from django.contrib.auth import authenticate, login, logout

    from .forms import RegisterFrom
    from users.models import UserProfile


    def xfz_register(request):
        """注册"""
        if request.method == "POST":
            register_form = RegisterFrom(request.POST)
            if register_form.is_valid():
                mobile = request.POST.get('mobile', None)
                if UserProfile.objects.filter(mobile=mobile):
                    # 用户已经存在，不需要再注册
                    return render(request, 'register.html', {'msg': '用户已经存在', 'register_form': register_form})
                pwd = request.POST.get('password1', None)
                username = request.POST.get('username', None)
                # 将密码加密后再保存
                pwd = make_password(pwd)
                user = UserProfile.objects.create(
                    mobile=mobile,
                    username=username,
                    is_active=True,
                    password=pwd
```

```
                  )
                  login(request, user)
                  return redirect('/index/')
              else:
                  return render(request, 'register.html', {'msg': '注册输入的数据有误', 'register_form': register_form})
          else:
              register_form = RegisterFrom()
              return render(request, "register.html", locals())
```

3. urls.py：

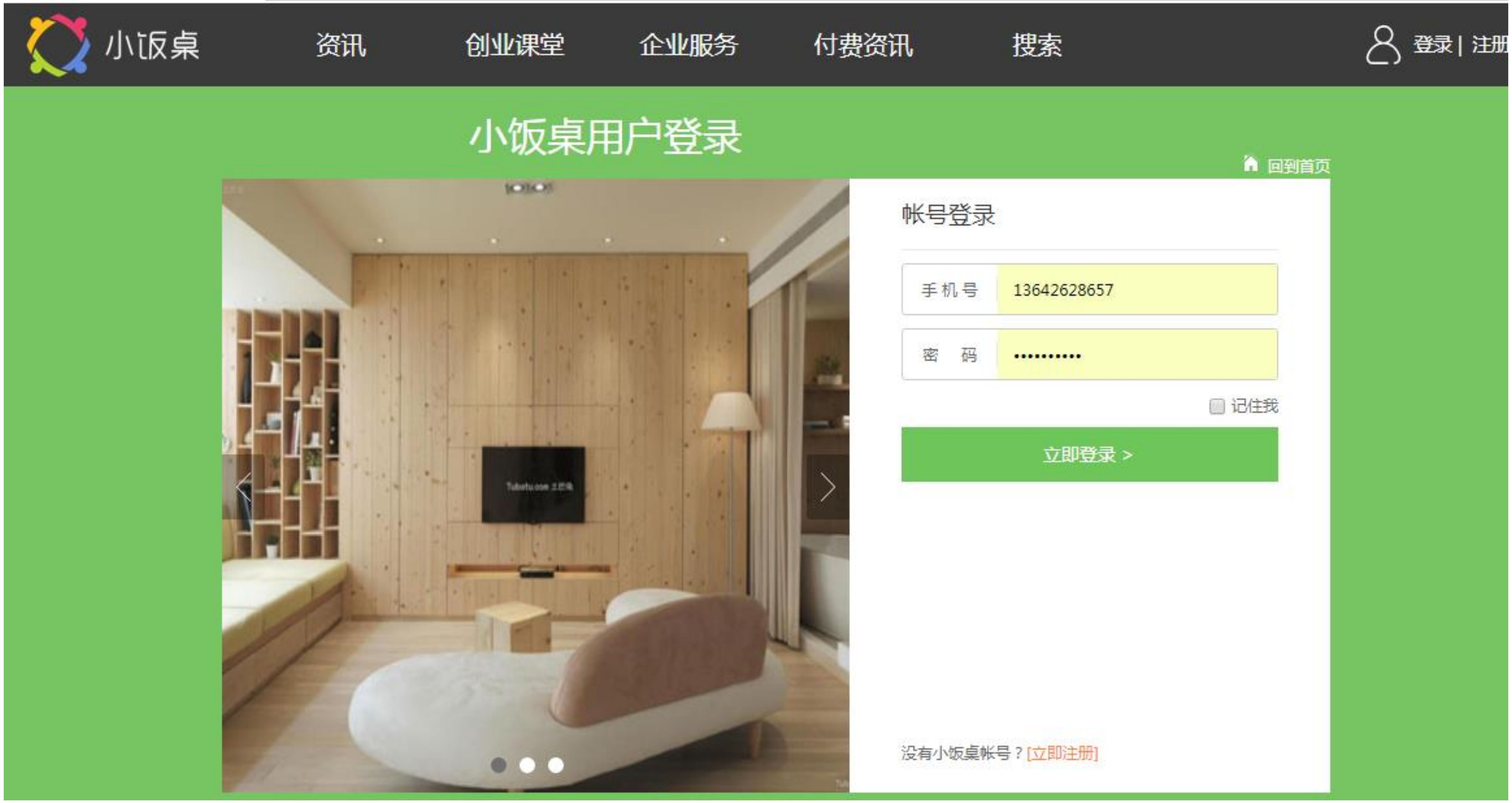
注册：urls.py

```
    from django.urls import path, include, re_path
    from xfz.views import xfz_register


    urlpatterns = [

        path("register/", xfz_register, name='auth_register'),

    ]
```

4. html：相关代码上面已经给出。

至此，注册的功能便完成了！

2.2、登录功能



小饭桌用户登录

回到首页

帐号登录

手 机 号	13642628657
密　码	••••••••••

记住我

立即登录 >

没有小饭桌帐号？[立即注册]

1）users/models.py：

models.py

```
from django.db import models
from django.contrib.auth.models import AbstractUser
from datetime import datetime

```

```
class UserProfile(AbstractUser):
    """用户"""
    gender_choices = (
        ('male','男'),
        ('female','女')
    )

    nick_name = models.CharField('昵称', max_length=32, default='', null=True, blank=True)
    birthday = models.DateField('生日', null=True, blank=True)
    gender = models.CharField('性别', max_length=8, choices=gender_choices, default='female')
    mobile = models.CharField('手机号', max_length=11, null=True, unique=True)
    image = models.ImageField(upload_to='image/%Y/%m', default='image/default.png', max_length=100)
    email = models.EmailField('邮箱', blank=True, unique=True, null=True)  # 重写 email 字段，加上'唯一'标识

    REQUIRED_FIELDS = ['mobile', 'email']  # 通过 manage.py...创建用户/管理员时，会提醒输入此字段(必输入)

    class Meta:
        verbose_name = '用户信息'
        verbose_name_plural = verbose_name

    def __str__(self):
        return self.username
```

2）form.py：

form.py

```
import re
from django import forms
from django.core.exceptions import ValidationError


class LoginForm(forms.Form,common_forms.FormMixin):
    """登录验证"""
    mobile = forms.CharField(required=True,
                                    max_length=11,
                                    min_length=11,
                                    error_messages={'min_length': '手机号只能是 11 位数！',
                                                    'max_length': '手机号只能是 11 位数！'
                                                    })
    password = forms.CharField(required=True,
                                min_length=8,
                                error_messages={'min_length': '密码最少不能少于 8 个字符！'}
                                )
    remember = forms.IntegerField(required=False)
```

common_forms.py： 也可以像注册那样，直接在 **forms** 中写，不需要另外处理 **error**

common_forms.py

```
class FormMixin(object):
    """
    用于获取表单验证失败时的错误信息，存进字典里
    """
    def get_error(self):
        if hasattr(self, 'errors'):
            # 如果有 errors 这个属性
            errors = self.errors.get_json_data()
            new_errors = {}
            for key, message_dicts in errors.items():
                messages = []
                for message in message_dicts:
                    messages.append(message['message'])
                new_errors[key] = messages
            return new_errors
        else:
            return {}
```

3）views.py：

views.py

```
from django.db.models import Q
from django.shortcuts import render, redirect
from  django.contrib.auth.backends import ModelBackend
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.hashers import make_password


from .forms import LoginForm
from users.models import UserProfile


class CustomBackend(ModelBackend):
    """
    用于 login 用户登录验证
    需在 settings 中配置好 authenticate 验证方式（即在此进行 authenticate 的相关验证）
    """
    def authenticate(self, request, username=None, password=None, **kwargs):
        # 重写 authenticate 方法
        try:
            user = UserProfile.objects.get(Q(username=username)|Q(mobile=username))
            if user.check_password(password):
                """
                1.在注册时，我们对密码使用了加密处理（django 下的 make_password），因此在登录验证密码时需要先将明文密码加密
                后才能跟数据库中已加密后的密码进行比较
                2.check_password()是 AbstractUser 类中的方法，UserProfile 继承于 AbstractUser，check_password 会加密明文密
                码后，与数据库密码做对比，再进行判断两密码是否一致
                3.验证如果为 True，表示密码一致；为 False，表示密码不一致
                """
                return user
            else:
                return None
        except Exception as e:
            return None
```

```
def xfz_login(request):
    """登录"""
    login_form = LoginForm()
    if request.method == 'POST':
        login_form = LoginForm(request.POST)
        if login_form.is_valid():
            mobile = login_form.cleaned_data.get('mobile')
            password = login_form.cleaned_data.get("password")
            remember = login_form.cleaned_data.get('remember')
            user = authenticate(request, username=mobile, password=password)
            if user:
                if user.is_active:
                    login(request, user)
                    if remember:
                        request.session.set_expiry(None)
                    else:
                        request.session.set_expiry(0)
                    return redirect('/news/')
                else:
                    return render(request, 'login.html', {'msg': '用户未激活', 'login_form': login_form})
            else:
                return render(request, 'login.html', {'msg': '用户名或密码错误', 'login_form': login_form})
        else:
            return render(request, 'login.html', {'msg': '用户名或密码格式错误，请重新输入！', 'login_form': login_form})

    return render(request, 'login.html')
```

4）login.html：

⊞ login.html

```
{% extends 'front_base.html' %}

    {% block title %}小饭桌登录{% endblock %}
    {% block front-css %}
    <link rel="stylesheet" type="text/css" href="{% static 'css/news_auth/reset.min.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'css/news_auth/login.min.css' %}">
    {% endblock %}
    {% block front-js %}
    <script src="{% static 'js/unslider.min.js' %}" type="text/javascript"></script>
    <script src="{% static 'js/login.min.js' %}"  type="text/javascript"></script>
    {% endblock %}

{% block body %}
<section>
    <div class="c-box bg-box">
        <div class="login-box clearfix">
            <div class="hd-login clearfix">
                <a class="index-logo" href="{% url 'index' %}"></a>
                <h1>小饭桌用户登录</h1>
                <a class="index-font" href="{% url 'index' %}">回到首页</a>
            </div>
            <div class="fl slide">
                <div class="imgslide">
                    <ul class="imgs">
                        <li><a href=""><img width="483" height="472" src="{% static 'images/auth/02_mid.jpg' %}"/></a></li>
                        <li><a href=""><img width="483" height="472" src="{% static 'images/auth/57a801860001c34b12000460.jpg' %}"
/></a></li>
                        <li><a href=""><img width="483" height="472" src="{% static 'images/auth/course.jpg' %}"/></a></li>
                    </ul>
                </div>
                <div class="unslider-arrow prev"></div>
                <div class="unslider-arrow next"></div>
            </div>
            <div class="fl form-box">
                <h2>帐号登录</h2>
                <form action="{% url 'auth_login' %}" method="post" autocomplete="off">
                    {% csrf_token %}
                    <div class="form-group marb8 {% if login_form.errors.mobile.0 %} errorput {% endif %}">
```

```
                        <label>手&nbsp;机&nbsp;号</label>
                        <input name="mobile" id="account_l" type="text" placeholder="手机号" {% if login_form.mobile.value %}value
="{{ login_form.mobile.value }}"{% endif %}/>
                    </div>
                    <div class="error btns login-form-tips" id="jsLoginTips">{{ login_form.errors.mobile.0 }}</div>
                    <div class="form-group marb8 {% if login_form.errors.password.0 %} errorput {% endif %}">
                        <label>密&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;码</label>
                        <input name="password" id="password_l" type="password" {% if login_form.password.value %}value="{{ login_f
orm.password.value }}"{% endif %} placeholder="请输入您的密码"/>
                    </div>
                    <div class="error btns login-form-tips" id="jsLoginTips">{{ login_form.errors.password.0 }}</div>
                    <div class="error btns login-form-tips" id="jsLoginTips">{{ msg }}</div>
                    <div class="auto-box marb8">

                        <label class="fr">
                            <input type="checkbox" name="remember" value="1" class="remember-me">
                            记住我
                        </label>
{#                        <a class="fr" href="{% url 'forgetpwd' %}">忘记密码？</a>#}
                    </div>
                    <input class="auth-btn btn-green" id="jsLoginBtn" type="submit" value="立即登录 > "/>
{#                    <input type='hidden' name='csrfmiddlewaretoken' value='5I2SlleZJOMUX9QbwYLUIAOshdrdpRcy'/>#}
                </form>
                <p class="form-p">没有小饭桌帐号？<a href="{% url 'auth_register' %}">[立即注册]</a></p>
            </div>
        </div>
    </div>
</section>
{% endblock %}
```

5）继承：front_base.html

front_base.html

```
<!DOCTYPE html>
<html lang="en">
```

```
        <head>
            <meta charset="UTF-8">
            <title>{% block title %}小饭桌{% endblock %}</title>
            <link rel="stylesheet" href="{% static 'css/news/index.min.css’ %}">
            {% block front-css %}{% endblock %}
            <script src="{% static 'js/jquery-3.3.1.min.js’ %}"></script>
            <script src="{% static 'js/index.min.js’ %}"></script>
            {% block front-js %}{% endblock %}
        </head>
        <body>
        <!-- 导航栏 -->
        <header class="header">
            <div class="container">
                <!-- logo -->
                <div class="logo-box">
                    <a href=""></a>
                </div>
                <!-- 主体中间部分 标题 -->
                <div class="nav">
                    <ul class="daohangtiao">
                        <li {% if request.path|slice:’5’ == '/news’ %}class="active"{% endif %} ><a href="{% url 'index’ %}">资讯</a></li>
                        <li {% if request.path|slice:’8’ == '/courses’ %}class="active"{% endif %} ><a class="chuangyeketang" href="{% url 'course’ %}">创业课堂</a></li>
                        <li><a class="qiyefuwu" href="#">企业服务</a></li>
                        <li {% if request.path|slice:’8’ == '/payinfo’ %}class="active"{% endif %} ><a href="{% url 'payinfo’ %}">付费资讯</a></li>
                        <li {% if request.path|slice:’7’ == '/search’ %}class="active"{% endif %} ><a href="{% url 'search’ %}">搜索</a></li>
                    </ul>

                    <div class="nav-float">
                        <ul class="ketang">
                            <li><a href="#">在线课堂</a></li>
                            <li><a href="#">线下课堂</a></li>
                        </ul>
                        <ul class="qiye">
                            <li><a href="#">创业礼包</a></li>
                            <li><a href="#">企业资讯</a></li>
                        </ul>
                    </div>

                </div>
                <!-- 登录注册 -->
                <div class="auth-box">
                    {% if request.user.is_authenticated%}
                        <div class="auth-login">
                            <div class="personal">
                                <div class="user">
                                    <p class="current-user">{{ request.user }}</p>
                                    <span class="top-down"><img src="{% static 'images/auth/top_down.png' %}"/></span>
                                    <span class="touxiang"><img width="45" height="45" src="{{ MEDIA_URL }}{{ request.user.image }}
"/></span>
                                </div>
                            </div>
                        </div>
                        <div class="userdetail">
                            <div class="personal-info">
                                <span><img width="60" height="60" src="{{ MEDIA_URL }}{{ request.user.image }}"/></span>
                                <div class="user-info">
                                    <h2>{{ request.user }}</h2>
                                    <p>{{ request.user.nick_name }}111</p>
                                </div>
                            </div>
                            <div class="personal-center">
                                <a class="personcenter" href="#">进入个人中心</a>
                                <a class="fr" href="{% url 'auth_logout' %}">退出</a>
                            </div>
                        </div>
                    {% else %}
                        <p class="personal-p"></p>
                        <div class="auth-login">
                            <a href="{% url 'auth_login' %}">登录&nbsp|</a>
                            <a href="{% url 'auth_register' %}">&nbsp 注册</a>
                        </div>
                    {% endif %}
                </div>
            </div>
        </header>
        {% block body %}
        <!-- body 主体[中间部分] -->
        <div class="main">
            <div class="wrapper">
                {% block left-content %}
                <!-- 内容左边[新闻部分] -->
                <div class="main-content-wrapper">
                    <!-- 轮播图 -->
                    <div class="banner-group" id="banner-group">
                        <ul class="banner-ul" id="banner-ul">
                            <li>
                                <a href="#">
                                    <img src="{% static 'images/banners/lunbo_2.jpeg' %}" alt="">
                                </a>
                            </li>
                            <li>
                                <a href="#">
                                    <img src="{% static 'images/banners/lunbo_3.jpg' %}" alt="">
                                </a>
                            </li>
                            <li>
                                <a href="#">
                                    <img src="{% static 'images/banners/lunbo_4.jpg' %}" alt="">
                                </a>
                            </li>
                            <li>
                                <a href="#">
                                    <img src="{% static 'images/banners/lunbo_5.png' %}" alt="">
                                </a>
                            </li>
                        </ul>

                        <ul class="num">
                            <li class="current"><a href="#">1</a></li>
                            <li><a href="#">2</a></li>
                            <li><a href="#">3</a></li>
                            <li><a href="#">4</a></li>
                        </ul>

                        <span class="left-btn btn">‹</span>
                        <span class="right-btn btn">›</span>
                    </div>

                    <!-- 新闻主体 -->
                    <div class="news-list-group">
                        <div class="news-inner">
                            <ul class="list-tab">
                                <li class="active"><a href="#">最新资讯</a></li>
                                <li><a href="#">深度报道</a></li>
                                <li><a href="#">金融科技</a></li>
                                <li><a href="#">人工智能</a></li>
                                <li><a href="#">干货分享</a></li>
                            </ul>
                            <!-- 新闻 list -->
                            <ul class="news-list">
                                <li>
                                    <div class="thumbnail-group">
```

```
                                        <a href="#">
                                            <img src="http://static-image.xfz.cn/1516169692_914.jpg-website.news.list"
                                                 alt="">
                                        </a>
                                    </div>
                                    <div class="news-group">
                                        <p class="title">
                                            <a href="#">王健林卖掉进军海外首个项目:17 亿售伦敦 ONE 六成股权</a>
                                        </p>
                                        <p class="desc">
                                            外界关于万达要出售此前在海外投资项目的消息一直不断。
                                        </p>
                                        <p class="more">
                                            <span class="category">深度报道</span>
                                            <span class="pub-time">1 小时前</span>
                                            <span class="author">知了课堂</span>
                                        </p>
                                    </div>
                                </li>
                                <li>
                                    <div class="thumbnail-group">
                                        <a href="#">
                                            <img src="http://static-image.xfz.cn/1516169692_914.jpg-website.news.list"
                                                 alt="">
                                        </a>
                                    </div>
                                    <div class="news-group">
                                        <p class="title">
                                            <a href="#">王健林卖掉进军海外首个项目:17 亿售伦敦 ONE 六成股权</a>
                                        </p>
                                        <p class="desc">
                                            外界关于万达要出售此前在海外投资项目的消息一直不断。
                                        </p>
                                        <p class="more">
```

```
                                            <span class="category">深度报道</span>
                                            <span class="pub-time">1 小时前</span>
                                            <span class="author">知了课堂</span>
                                        </p>
                                    </div>
                                </li>
                            </ul>
                            <div class="load-more-group">
                                <button class="load-more">查看更多</button>
                            </div>

                        </div>
                    </div>
                </div>
                {% endblock %}
                {% block right-wrapper %}
                <!-- 内容右边[侧边栏] -->
                <div class="sidebar-wrapper">
                    <div class="online-class">
                        <span class="class-title">在线课堂</span>
                        <span class="more"><a href="#">更多</a></span>
                    </div>
                    <div class="hot-advertist">
                        <a href="#">
                            <img src="{% static 'images/build/hot-advertist.png' %}" alt="">
                        </a>
                    </div>
                    <div class="platform-group">
                        <div class="online-class">
                            <span class="class-title">关注小饭桌</span>
                        </div>
                        <div class="focus-group">
                            <ul class="left-group">
                                <li class="zhihu">
                                    <a href="#" target="_blank">小饭桌创业课堂</a>
                                </li>
                                <li class="weibo">
                                    <a href="#" target="_blank">小饭桌创业课堂</a>
                                </li>
                                <li class="toutiao">
                                    <a href="#" target="_blank">小饭桌</a>
                                </li>
                            </ul>
                            <div class="right-group">
                                <p class="desc">扫码关注小饭桌微信公众平台 xfz008</p>
                            </div>
                        </div>
                    </div>
                    <div class="hot-news-group">
                        <div class="online-class">
                            <span class="class-title">热门推荐</span>
                        </div>
                        <ul class="hot-list-group">
                            <li>
                                <div class="left-group">
                                    <p class="title">
                                        <a href="#">王健林卖掉进军海外首个项目:17 亿售伦敦 ON...</a>
                                    </p>
                                    <p class="more">
                                        <span class="category"><a href="#">深度报道</a></span>
                                        <span class="pub-time">1 小时前</span>
                                    </p>
                                </div>
                                <div class="right-group">
                                    <a href="#">
                                        <img src="{% static 'images/build/hot-news_01.png' %}" alt="">
                                    </a>
                                </div>
                            </li>
                            <li>
                                <div class="left-group">
                                    <p class="title">
                                        <a href="#">王健林卖掉进军海外首个项目:17 亿售伦敦 ON...</a>
                                    </p>
                                    <p class="more">
                                        <span class="category"><a href="#">深度报道</a></span>
                                        <span class="pub-time">1 小时前</span>
                                    </p>
                                </div>
                                <div class="right-group">
                                    <a href="#">
                                        <img src="{% static 'images/build/hot-news_01.png' %}" alt="">
                                    </a>
                                </div>
                            </li>
                        </ul>
                    </div>
                </div>
                {% endblock %}
            </div>
        </div>
        {% endblock %}

        <!-- footer -->
        <footer class="footer">
            <div class="top-group">
                <div class="top-inner-group">
                    <div class="logo-box"></div>
                    <div class="detail-group">
                        <div class="line1">
                            <ul class="links">
                                <li><a href="#">关于小饭桌</a></li>
                                <li><a href="#">创业课堂</a></li>
                                <li><a href="#">寻求报道</a></li>
                                <li><a href="#">创业礼包</a></li>
                            </ul>
                            <div class="about-us">
                                <span class="title">关于我们：</span>
                                <ul class="social-group">
                                    <li class="weixin">
                                        <div class="wx-qrcode"></div>
                                        <span class="text">xfz2019</span>
                                    </li>
                                    <li class="weibo">
                                        <a href="#" class="text">小饭桌创业课堂</a>
                                    </li>
                                </ul>
                            </div>
                        </div>
                        <div class="line2">
                            <p class="address">
                                地址：北京市朝阳区东三环北路 38 号院 1 号楼 17 层 2001 内 1、16 室
                            </p>
                            <p class="contact">
                                联系方式：400-810-1090（工作日 10 点-18 点）
                            </p>
                        </div>

                    </div>
                </div>
            </div>
            <div class="bottom-group">
                ©2017 北京子木投资顾问有限公司 京 ICP 备 15051289 号-1
            </div>
        </footer>

        </body>

        </html>
```

6）urls.py：

urls.py

```
    from django.urls import path,include, re_path
    from xfz.views import xfz_logout


    urlpatterns = [

        path("login/", xfz_login, name='auth_login'),

    ]
```

2.3、退出登录功能

退出登录功能比较简单，只需要 views.py 及 urls.py 相关代码处理即可，如下

1. views.py：

```
from django.contrib.auth import logout

def xfz_logout(request):
    logout(request)
    return render(request, 'news/index.html')
```

2. urls.py：

```
from django.urls import path, include, re_path
from xfz.views import xfz_logout



urlpatterns = [

    path("logout/", xfz_logout, name='auth_logout'),
]
```

至此，小饭桌登录及注册相关的功能实现到这就算完成了！

Django 打造大型企业官网-项目实战（三）

一、CRM 后台管理系统

前面我们使用的是 xadmin 后台管理系统，在使用中发现，在权限限制中，我们能实现不同等级的用户/管理（超级管理员/管理员/用户）登录后台时拥有不同数据的操作权限（查看、修改等），但实际上在更细方面的地方我们并没有找到有效的方法实现精准的权限限制，比如当普通用户想新增新闻时，在作者字段一栏会直接列出目前所有的用户名（作者），当前用户可以任意选择作者作为当前文章的发布者，这样明显是很不安全的，而如果我们需要改动这些细节就必须得从 xadmin 源码上寻求解决方法。为了更好的控制后台权限及数据展示，我们重新部署自己的后台管理系统。

重新部署后台管理系统，需要使用到新的第三方库：adminLTE ，

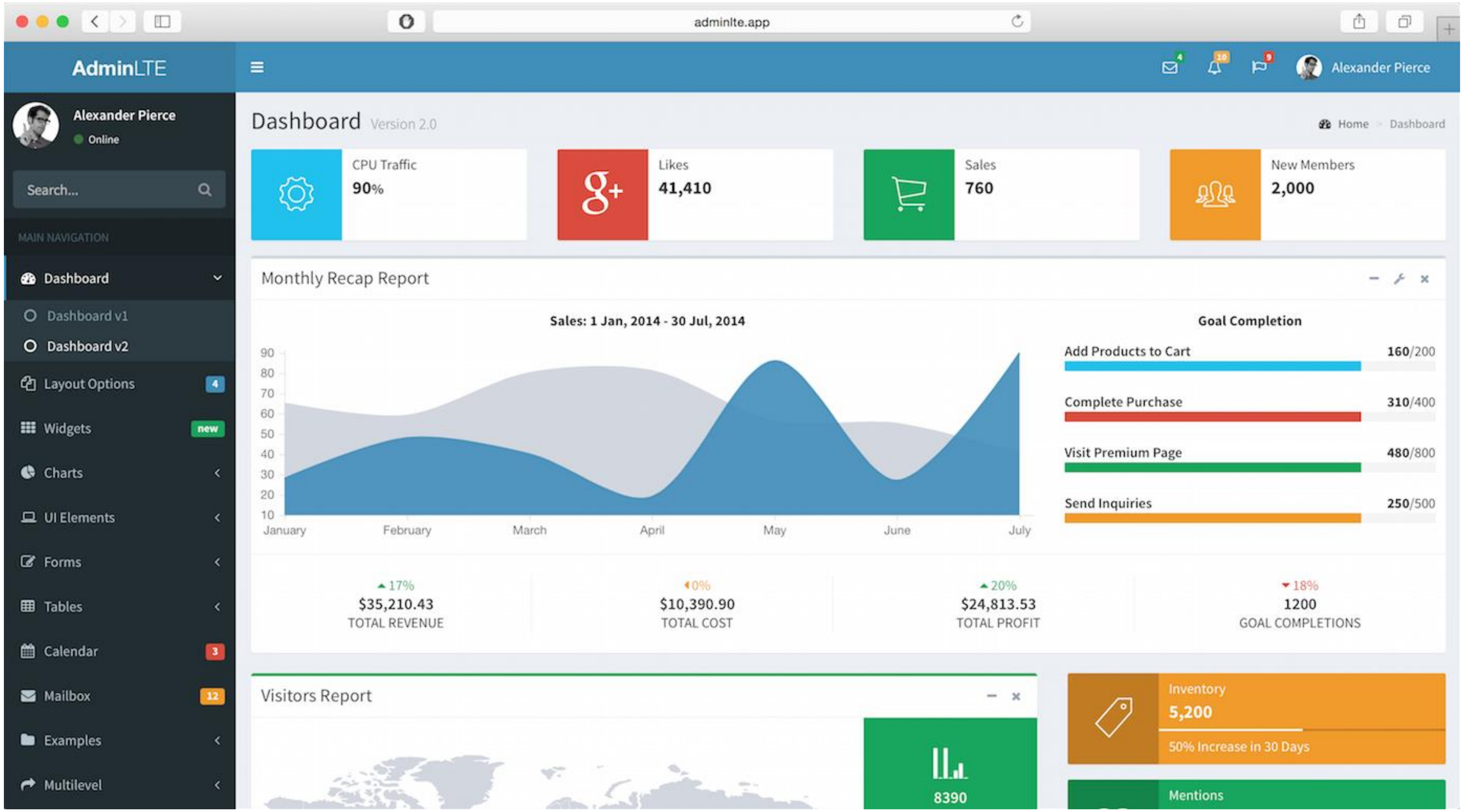
官网：https://adminlte.io/

文档及下载：http://adminlte.la998.com/documentation/

GitHub 下载：https://github.com/ColorlibHQ/AdminLTE

操作演示：http://adminlte.la998.com/index2.html

界面：



一般使用，可以将所需要的页面的代码拷贝到自己项目系统中对应的位置，再把相关联的样式文件等拷贝进去。如果感兴趣，可以参考上述文档链接或查询官方文档。

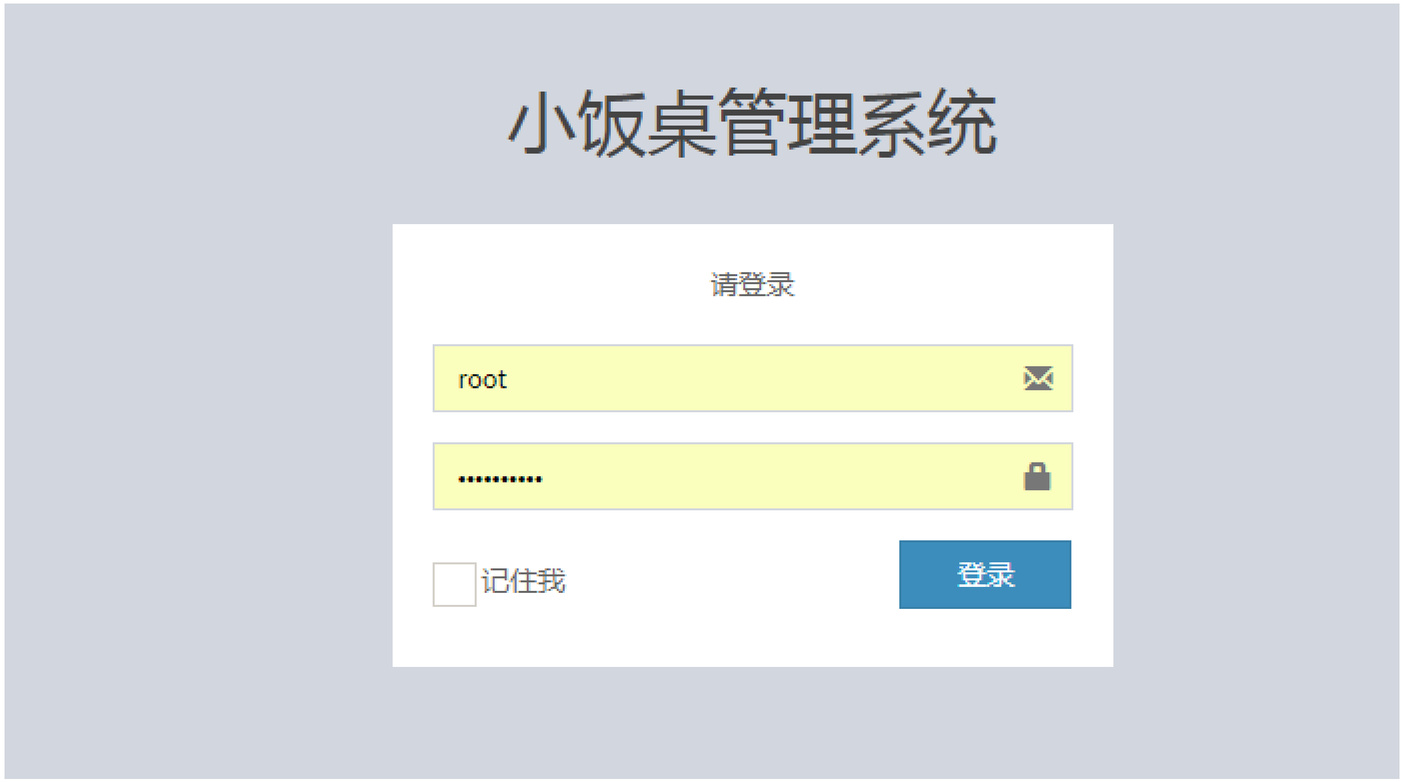
---

crm 后台管理系统

1、后台系统登录相关功能

新建 app 并注册到 settings.py 下的 INSTALED_APPS 中，命名 crm：python manage.py startapp crm

登录界面：



登录功能相关代码：

1）HTML 前端代码：

登录：login.html

```
<! -- crm/login.html -->
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>小饭桌 | 后台登录</title>
  <!-- Tell the browser to be responsive to screen width -->
  <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">
  <!-- Bootstrap 3.3.7 -->
```

```
        <link rel="stylesheet" href="{% static 'adminlte/bower_components/bootstrap/dist/css/bootstrap.min.css' %}">
        <!-- Font Awesome -->
        <link rel="stylesheet" href="{% static 'adminlte/bower_components/font-awesome/css/font-awesome.min.css' %}">
        <!-- Ionicons -->
        <link rel="stylesheet" href="{% static 'adminlte/bower_components/Ionicons/css/ionicons.min.css' %}">
        <!-- Theme style -->
        <link rel="stylesheet" href="{% static 'adminlte/dist/css/AdminLTE.min.css' %}">
        <!-- iCheck -->
        <link rel="stylesheet" href="{% static 'adminlte/plugins/iCheck/square/blue.css' %}">
        <link rel="stylesheet" href="{% static 'adminlte/login error.css' %}">

        <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->
        <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
        <!--[if lt IE 9]>
        <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
        <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
        <![endif]-->

        <!-- Google Font -->
        <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,700,300italic,400italic,
600italic">
      </head>
      <body class="hold-transition login-page">
      <div class="login-box">
        <div class="login-logo">
          <a href="#">小饭桌管理系统</a>
        </div>
        <!-- /.login-logo -->
        <div class="login-box-body">
          <p class="login-box-msg">请登录</p>

          <form action="{% url 'crm_login' %}" method="post">
              {% csrf_token %}
            <div class="form-group has-feedback {% if login_form.errors.mobile.0 %}errorput{% endif %}">
```

```
              <input type="text" class="form-control" name="mobile" {% if login_form.errors.mobile.0 %} placeholder="{{ login_form.errors.mobile.0 }}" {% elif login_form.mobile.value %} value="{{ login_form.mobile.value }}" {% else %} placeholder="请输入您的手机号" {% endif %}>
              <span class="glyphicon glyphicon-envelope form-control-feedback"></span>
            </div>
            <div class="form-group has-feedback {% if login_form.errors.password.0 %} errorput {% endif %}">
              <input type="password" class="form-control" name="password" {% if login_form.errors.password.0 %} placeholder="{{ login_form.errors.password.0 }}" {% else %} placeholder="请输入密码" {% endif %}>
              <span class="glyphicon glyphicon-lock form-control-feedback"></span>
            </div>
            <div class="form-group has-feedback error">
              {{ msg }}
            </div>
            <div class="row">
              <div class="col-xs-8">
                <div class="checkbox icheck">
                  <label>
                    <input type="checkbox" name="remember" value="1"> 记住我
                  </label>
                </div>
              </div>
              <!-- /.col -->
              <div class="col-xs-4">
                <button type="submit" class="btn btn-primary btn-block btn-flat">登录</button>
              </div>
              <!-- /.col -->
            </div>
          </form>
        </div>
        <!-- /.login-box-body -->
      </div>
      <!-- /.login-box -->

      <!-- jQuery 3 -->
```

```
<script src="{% static 'adminlte/bower_components/jquery/dist/jquery.min.js' %}"></script>
<!-- Bootstrap 3.3.7 -->
<script src="{% static 'adminlte/bower_components/bootstrap/dist/js/bootstrap.min.js' %}"></script>
<!-- iCheck -->
<script src="{% static 'adminlte/plugins/iCheck/icheck.min.js' %}"></script>
<script>
  $(function () {
    $('input').iCheck({
      checkboxClass: 'icheckbox_square-blue',
      radioClass: 'iradio_square-blue',
      increaseArea: '20%' /* optional */
    });
  });
</script>
</body>
</html>
```

2）views.py 后端代码：

登录：views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login,

from xfz.forms import LoginForm

def crm_login(request):
    """crm 登录"""
    if request.method == 'GET':
        # 如果在前端已经登录，可以直接进入后台管理系统
        try:
            if request.user.is_authenticated:
                return redirect('/crm/index/')
        except:
            pass
        login_form = LoginForm()
```

```
            return render(request, 'crm/login.html')
        else:
            login_form = LoginForm(request.POST)
            if login_form.is_valid():
                mobile = login_form.cleaned_data.get('mobile')
                password = login_form.cleaned_data.get("password")
                remember = login_form.cleaned_data.get('remember')
                user = authenticate(request, username=mobile, password=password)
                if user:
                    if user.is_active and user.is_staff:
                        login(request, user)
                        if remember:
                            request.session.set_expiry(None)
                        else:
                            request.session.set_expiry(0)
                        return redirect('/crm/index/')
                    else:
                        return render(request, 'crm/login.html', {'msg': '用户无权登录后台系统', 'login_form': login_form})
                else:
                    return render(request, 'crm/login.html', {'msg': '用户名或密码错误', 'login_form': login_form})
            else:
                return render(request, 'crm/login.html', {'msg': '用户名或密码格式错误，请重新输入！', 'login_form': login_for
m})
```

3）urls.py 路由：

crm/urls.py

```
    # xfz/urls.py: path("crm/", include("apps.crm.urls")),   # crm 管理后台
    from django.urls import path

    from .views import crm_login, index


    urlpatterns = [

⊞           path("login/", crm_login, name='crm_login'),    # crm 管理后台
⊞           path("index/", index, name='crm_index'),    # crm 管理后台
⊞
⊞       ]
```

2、后台登录权限限制：

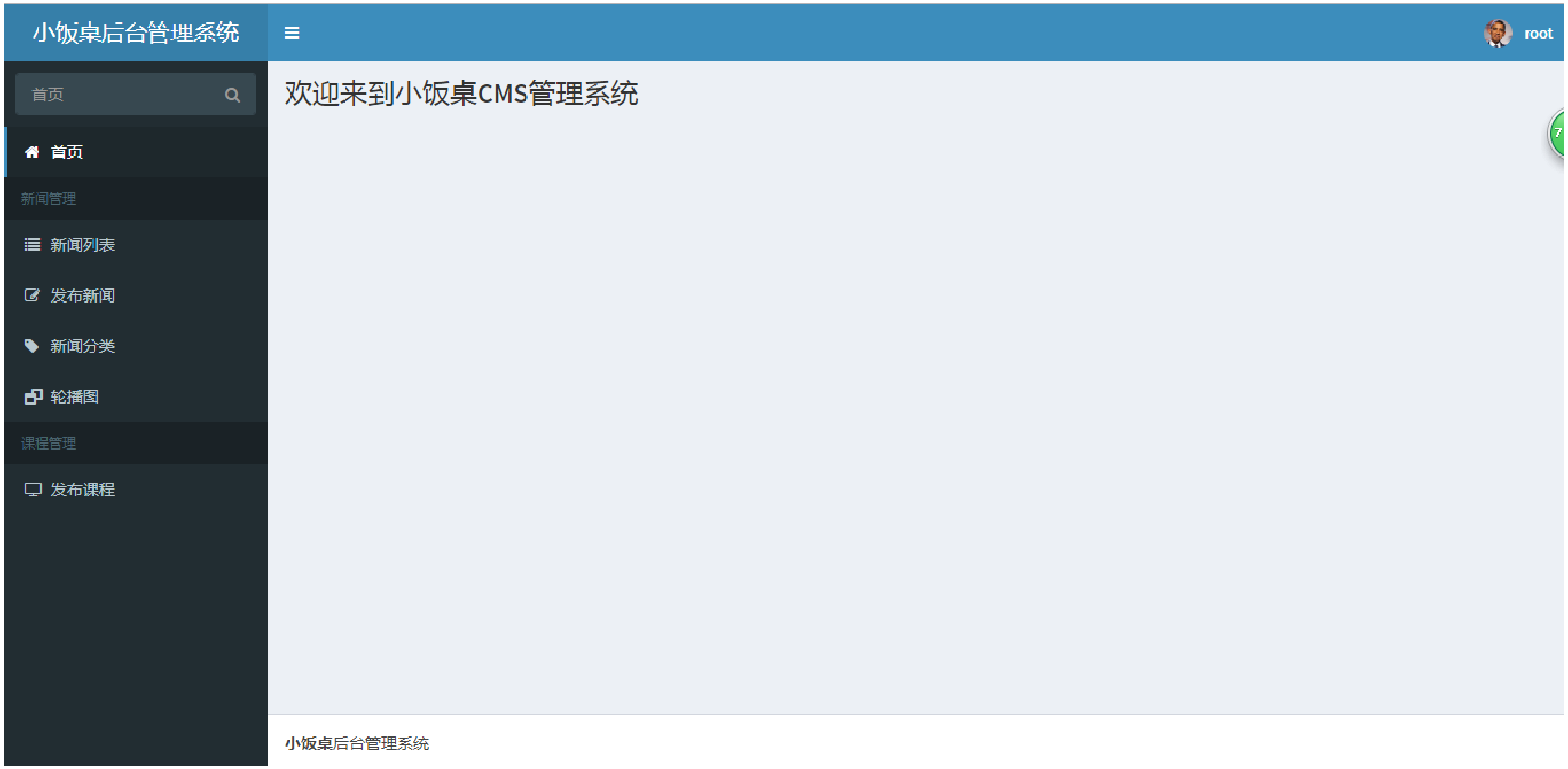
限制只有后台权限的用户才能登录（django 自带员工识别装饰器，识别只有 is_staff 的用户才能登录后台系统），如后台首页登录限制，代码如下：

```
# crm/views.py
from django.contrib.admin.views.decorators import staff_member_required   # 是否为后台员工识别装饰器
@staff_member_required(login_url='index')  # 不是后台员工，无法登录后台，重定向到前端首页
def index(request):
    """小饭桌管理后台首页"""
    return render(request, 'crm/index.html')
```

这样，无论是在网址栏上直接输入网址还是前端跳转，只要不是员工都无法访问到后台

3、cms 后台管理系统-首页

界面：



欢迎来到小饭桌CMS管理系统

小饭桌后台管理系统

功能实现代码：

1）HTML 前端代码：

base.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
        <title>{% block title %}{% endblock %} | 小饭桌管理系统</title>
        <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">
        <link rel="stylesheet" href="{% static 'adminlte/bower_components/bootstrap/dist/css/bootstrap.min.css' %}">
        <link rel="stylesheet" href="{% static 'adminlte/bower_components/font-awesome/css/font-awesome.min.css' %}">
        <link rel="stylesheet" href="{% static 'adminlte/dist/css/AdminLTE.min.css' %}">
        <link rel="stylesheet" href="{% static 'adminlte/dist/css/skins/_all-skins.min.css' %}">
        <link rel="stylesheet" href="{% static 'adminlte/plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.min.css' %}">
        <link rel="stylesheet" href="{% static 'sweetalert/sweetalert.css' %}">
        <script src="{% static 'adminlte/bower_components/jquery/dist/jquery.min.js' %}"></script>
        <script src="{% static 'adminlte/bower_components/bootstrap/dist/js/bootstrap.min.js' %}"></script>
        <script src="{% static 'adminlte/dist/js/adminlte.min.js' %}"></script>
        <script src="{% static 'sweetalert/sweetalert.min.js' %}"></script>
    {#    <script src="{% static 'js/xfzajax.min.js' %}"></script>#}
    {#    <script src="{% static 'js/xfzalert.min.js' %}"></script>#}
    {#    <script src="{% static 'js/message.min.js' %}"></script>#}
        {% block head %}{% endblock %}
    </head>
    <body class="hold-transition skin-blue sidebar-mini">
    <div class="wrapper">
        <header class="main-header">
            <!-- Logo -->
            <a href="#" class="logo">
                <!-- mini logo for sidebar mini 50x50 pixels -->
                <span class="logo-mini"><b>CMS</b></span>
                <!-- logo for regular state and mobile devices -->
                <span class="logo-lg">小饭桌后台管理系统</span>
            </a>
            <!-- Header Navbar: style can be found in header.less -->
            <nav class="navbar navbar-static-top">
                <!-- Sidebar toggle button-->
                <a href="#" class="sidebar-toggle" data-toggle="push-menu" role="button">
                    <span class="sr-only">Toggle navigation</span>
                </a>

                <div class="navbar-custom-menu">
                    <ul class="nav navbar-nav">
                        <li class="dropdown user user-menu">
                            <a href="#" class="dropdown-toggle" data-toggle="dropdown">
                                <img src="{% static 'images/auth/aobama.png' %}"
                                     class="user-image" alt="User Image">
                                <span class="hidden-xs">{{ request.user.username }}</span>
                            </a>
                            <ul class="dropdown-menu">
                                <!-- User image -->
                                <li class="user-header">
                                    <img src="{% static 'images/auth/aobama.png' %}"
                                         class="img-circle" alt="User Image">
                                    <p>
                                        {{ request.user.username }}
                                        <small>{{ request.user.employee }}</small>
                                    </p>
                                </li>
                                <!-- Menu Footer-->
                                <li class="user-footer">
                                    <div class="pull-left">
                                        <a href="#" class="btn btn-default btn-flat">个人信息中心</a>
                                    </div>
                                    <div class="pull-right">
                                        <a href="#" class="btn btn-default btn-flat">退出登录</a>
                                    </div>
                                </li>
                            </ul>
                        </li>
                    </ul>
                </div>
            </nav>
        </header>
        <aside class="main-sidebar">
```

```
                <!-- sidebar: style can be found in sidebar.less -->
                <section class="sidebar">
                    <form action="/" method="get" class="sidebar-form">
                        <div class="input-group">
                            <input type="text" name="q" class="form-control" placeholder="首页">
                            <span class="input-group-btn">
                        <button type="submit" name="search" id="search-btn" class="btn btn-flat"><i class="fa fa-search"></i>
                        </button>
                  </span>
                        </div>
                    </form>
                    <!-- /.search form -->
                    <!-- sidebar menu: : style can be found in sidebar.less -->
                    <ul class="sidebar-menu" data-widget="tree">
                        <li class="active">
                            <a href="{% url 'crm_index' %}">
                                <i class="fa fa-home"></i>
                                <span>首页</span>
                            </a>
                        </li>
                        <li class="header">新闻管理</li>
                        <li>
                            <a href="#">
                                <i class="fa fa-list"></i>
                                <span>新闻列表</span>
                            </a>
                        </li>
                        <li>
                            <a href="#">
                                <i class="fa fa-edit"></i>
                                <span>发布新闻</span>
                            </a>
                        </li>
                        <li>
```

```
                            <a href="#">
                                <i class="fa fa-tag"></i>
                                <span>新闻分类</span>
                            </a>
                        </li>
                        <li>
                            <a href="#">
                                <i class="fa fa-window-restore"></i>
                                <span>轮播图</span>
                            </a>
                        </li>
                        <li class="header">课程管理</li>
                        <li>
                            <a href="#">
                                <i class="fa fa-tv"></i>
                                <span>发布课程</span>
                            </a>
                        </li>
                    </ul>
                </section>
            </aside>
            <div class="content-wrapper">
                <section class="content-header">
                    {% block content-header %}{% endblock %}
                </section>
                <section class="content">
                    {% block content %}{% endblock %}
                </section>
            </div>
            <footer class="main-footer">
                <strong>小饭桌</strong>后台管理系统
            </footer>
        </div>
        </body>
```

```
</html>
```

crm:index.html

```
{% extends 'crm/base.html' %}

{% block title %}首页{% endblock %}

{% block content-header %}
    <h1>欢迎来到小饭桌 CMS 管理系统</h1>
{% endblock %}
```

2）**views.py** 后端代码（有待优化）：

views.py

```
from django.contrib.admin.views.decorators import staff_member_required   # 是否为后台员工识别装饰器

@staff_member_required(login_url='index')  # 不是后台员工，无法登录后台，重定向到前端首页
def index(request):
    """小饭桌管理后台首页"""
    return render(request, 'crm/index.html')
```

3）**urls.py**：

urls.py

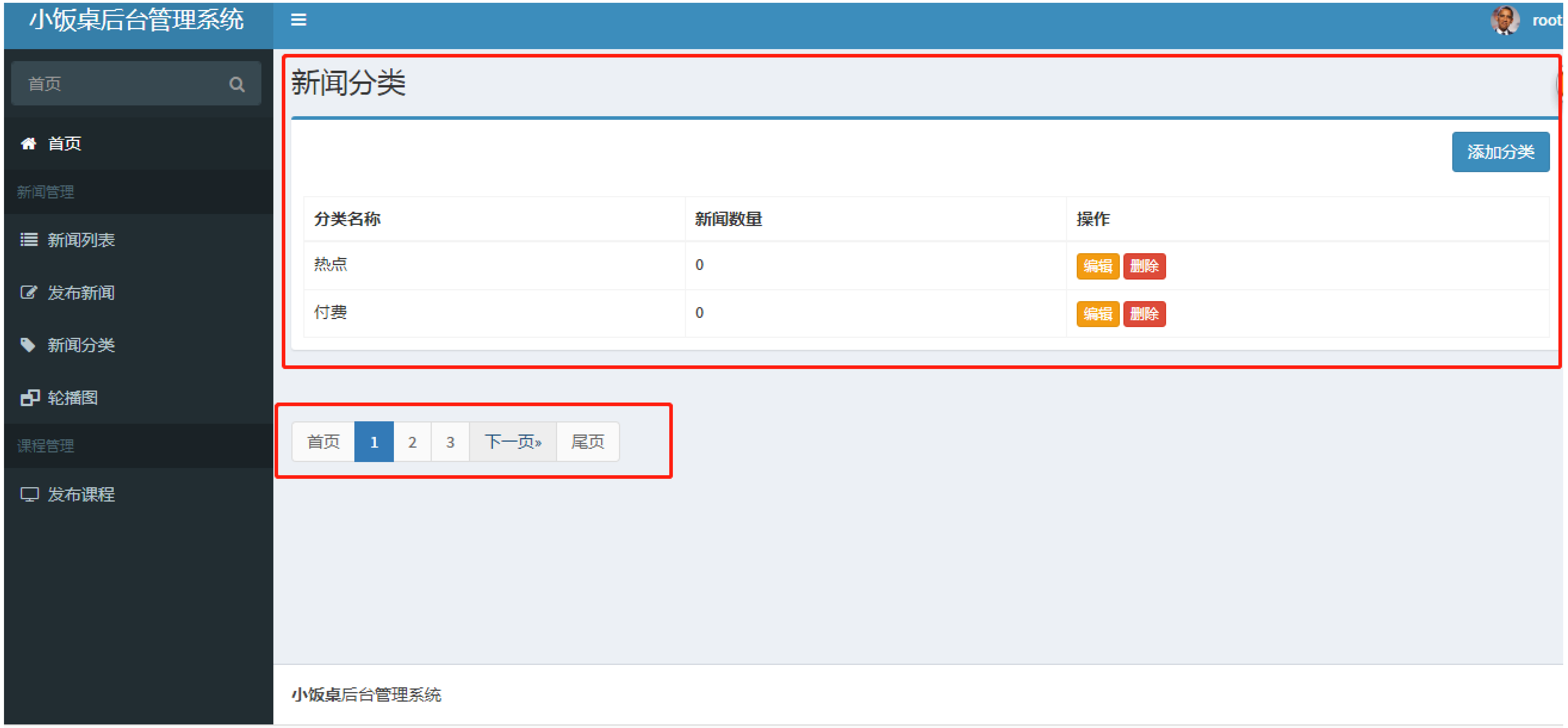
```
from django.urls import path

from .views import index


urlpatterns = [

    path("index/", index, name='crm_index'),   # crm 管理后台首页

]
```

4、新闻分类 - 功能：获取所有新闻分类 、 添加新闻分类 、 修改新闻分类 、 删除新闻分类

需注意的是，在获取新闻分类时，需要设定权限限制，只有拥有后台管理权限的用户才能对新闻分类进行一系列的操作

1）获取所有新闻分类

界面：



新闻分类

分类名称	新闻数量	操作
热点	0	编辑 删除
付费	0	编辑 删除

HTML 前端代码：继承于 crm/index.html

news_cetagory.html

```
{% extends 'crm/base.html' %}
{% load crm_tags %}

{% block title %}
```

```
            新闻分类
        {% endblock %}

        {% block head %}
            <link rel="stylesheet" href="{% static 'css/news_category/category.min.css'%}">
        {#    <script src="{% static 'js/news_category.min.js' %}"></script>#}
        {% endblock %}

        {% block content-header %}
            <h1>新闻分类</h1>
        {% endblock %}

        {% block content %}
            <div class="row">
                <div class="col-md-12">
                    <div class="box box-primary">
                        <div class="box-header">
                            <button id="add-btn" class="btn btn-primary pull-right" onclick="addAction()">添加分类</button>
                        </div>
                        <div class="box-body">
                            <table class="table table-bordered">
                                <thead>
                                    <tr>
                                        <th>分类名称</th>
                                        <th>新闻数量</th>
                                        <th>操作</th>
                                    </tr>
                                </thead>
                                <tbody>
                                    {% for category in categories %}
                                        <tr data-pk="{{ category.id}}" data-name="{{ category.name }}">
                                            <td>{{ category.name }}</td>
                                            <td>{{ category.news_nums }}</td>
                                            <td>
```

```
                                                <button class="btn btn-warning btn-xs edit-btn">编辑</button>
                                                <button class="btn btn-danger btn-xs delete-btn">删除</button>
                                            </td>
                                        </tr>
                                    {% endfor %}
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>

            <div class="pagination">
                <!-- 分页 -->
                {% render_paginator categories %}
            </div>
```

views.py：包含分页功能

views.py

```
        from django.shortcuts import render
        from django.contrib.admin.views.decorators import staff_member_required   # 是否为后台员工识别装饰器
        from django.views.decorators.http import require_POST, require_GET
        from django.core.paginator import Paginator,PageNotAnInteger,EmptyPage
        from news.models import NewsCategory

        @staff_member_required(login_url='index')
        @require_GET
        def new_category(request):
            """新闻分类详情"""

            categories = NewsCategory.objects.all()
            # 分页（分页功能）
            paginator = Paginator(categories, 2)  # 每页显示 2 行数据

            page = request.GET.get('_page')
            try:
                categories = paginator.page(page)
            except PageNotAnInteger:
                # If page is not an integer, deliver first page.
                categories = paginator.page(1)
            except EmptyPage:
                # If page is out of range (e.g. 9999), deliver last page of results.
                categories = paginator.page(paginator.num_pages)  # paginator.num_pages：总页数，即返回最后一页
            return render(request, 'crm/news_category.html', locals())
```

urls.py：

urls.py

```
    from django.urls import path

    from crm.views import new_category


    urlpatterns = [

        path("category/", new_category, name='news_category'),   # crm 管理后台 新闻分类

    ]
```

crm/models.py/news_cetagory

NewsCategory

```
    class NewsCategory(models.Model):
        """新闻分类"""
        name = models.CharField("新闻分类", max_length=40)
        add_time = models.DateTimeField("添加时间", default=datetime.now)

        class Meta:
            verbose_name = "新闻分类"
            verbose_name_plural = verbose_name

        def news_nums(self):
            """该分类下新闻数量"""
            return self.news_set.all().count()

        def __str__(self):
            return self.name
```

获取所有新闻数据后，我们需要对所有新闻数据进行分页操作，使用 **templatetags** 自定义标签的方式实现：

templatetags 自定义标签步骤

```
"""
templatetags 包，用于创建自定义标签，再用于前端显示
自定义标签步骤： 1、在 APP 文件中创建 templatetags 包（不是文件夹）
                2、在 templatetags 包中创建 kingadmin.py 文件
                3、导入：from django.template import Library
                4、实例化：register = Library() 。注：命名必须‘register’，不能改
"""
```

在 **crm** 中新建 **templatetags** 包，在 **templatetags** 包中新建 **crm_tags.py** 文件，分页代码如下：

crm_tags.py

```
from django.template import Library
from django.utils.safestring import mark_safe

register = Library()



@register.simple_tag
def render_paginator(querysets):
    """
    分页功能
    从 views 中拿到 querysets
    paginator = Paginator(querysets, 2) ：一页显示 2 行
    querysets = paginator.page(page)：当前页码
```

```
        """

        ele = '''
            <nav aria-label="Page navigation">
                <ul class="pagination">
                    <li>
                        <a href="?_page=1" aria-label="shouye">
                            <span aria-hidden="true">首页</span>
                        </a>
                    </li>
        '''
        if querysets.has_previous():
            p_ele = '''
            <li><a href="?_page=%s" aria-label="Previous">&laquo;上一页</a></li>
            ''' % (querysets.previous_page_number())
            ele += p_ele
        # querysets.paginator=paginator，page_range:页数范围
        for i in querysets.paginator.page_range:
            # querysets.number:当前页码

            if abs(querysets.number - i) < 4:# 只显示相邻页码，最多 3 页
                active = ''
                # 当前页
                if querysets.number ==i:
                    active = 'active'

                p_ele = '''
                <li class="%s"><a href="?_page=%s">%s</a></li>
                '''% (active, i, i)
                ele += p_ele
            # 是否有下一页
        if querysets.has_next():
            p_ele = '''
                <li><a href="?_page=%s" aria-label="Next">下一页&raquo;</a></li>
```

```
                 ''' % (querysets.next_page_number())
            ele += p_ele

            # querysets.paginator.num_pages: 总页数
        p_ele = '''
            <li>
                <a href="?_page=%s" aria-label="weiye">
                    <span aria-hidden="true">尾页</span>
                </a>
            </li>
        '''%(querysets.paginator.num_pages)
        ele += p_ele

        ele += "</ul></nav>"
        return mark_safe(ele)
```

HTML 模板中运用:

news_cetagory.html

```
    {% load crm_tags %}

        <div class="pagination">
            <!-- 分页 -->
            {% render_paginator categories %}
        </div>
```

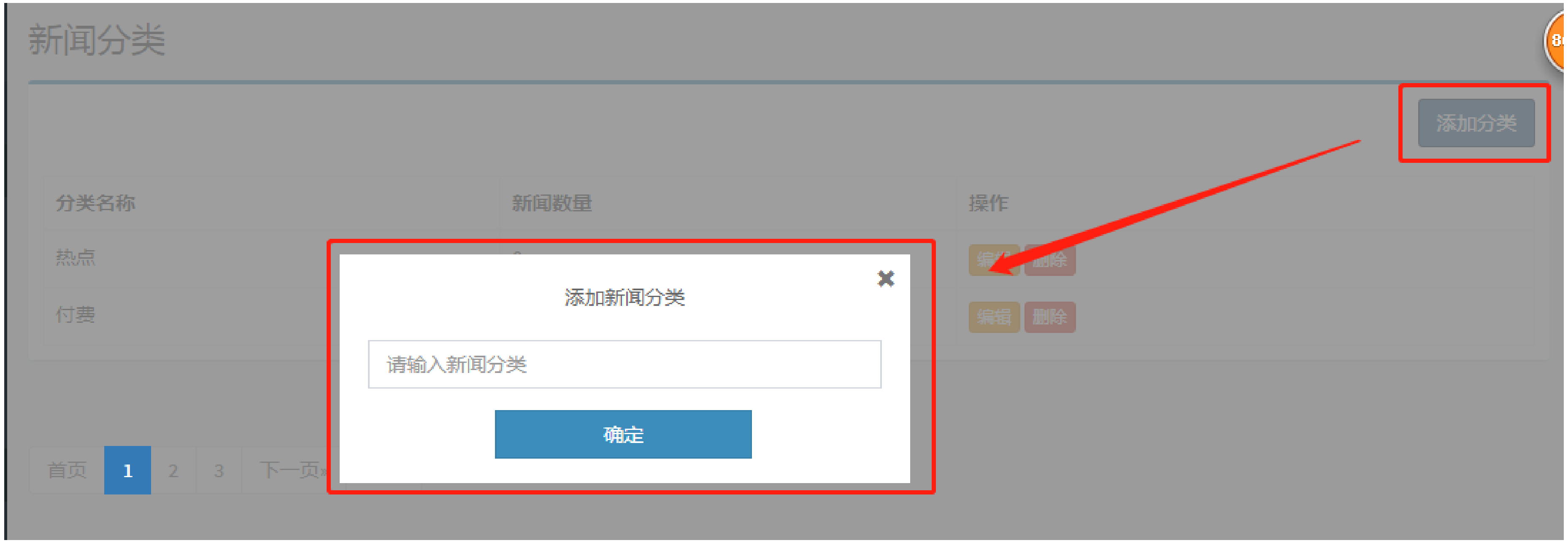
views.py 代码实现:

views.py

```
    from django.core.paginator import Paginator,PageNotAnInteger,EmptyPage
    from news.models import NewsCategory
    categories = NewsCategory.objects.all()
        # 分页 (分页功能)
        paginator = Paginator(categories, 2)  # 每页显示 2 行数据

        page = request.GET.get('_page')
        try:
            categories = paginator.page(page)
        except PageNotAnInteger:
            # If page is not an integer, deliver first page.
            categories = paginator.page(1)
        except EmptyPage:
            # If page is out of range (e.g. 9999), deliver last page of results.
            categories = paginator.page(paginator.num_pages)  # paginator.num_pages：总页数，即返回最后一页
```

2）新增新闻分类功能

在新闻分类页面中，使用模态对话框的形式实现 （修改新闻分类、删除新闻分类也使用此方法）



功能实现：

HTML 前端代码（结合获取新闻前端代码，及 **js**、**ajax** 异步 **post** 数据）：

add_cetagory

```
    {% extends 'crm/base.html' %}
```

```
{% load crm_tags %}

{% block title %}
    新闻分类
{% endblock %}

{% block head %}
    <link rel="stylesheet" href="{% static 'css/news_category/category.min.css'%}">
{#    <script src="{% static 'js/news_category.min.js' %}"></script>#}
{% endblock %}

{% block content-header %}
    <h1>新闻分类</h1>
{% endblock %}

{% block content %}
    <div class="row">
        <div class="col-md-12">
            <div class="box box-primary">
                <div class="box-header">
                    <button id="add-btn" class="btn btn-primary pull-right" onclick="addAction()">添加分类</button>
                </div>
                <div class="box-body">
                    <table class="table table-bordered">
                        <thead>
                            <tr>
                                <th>分类名称</th>
                                <th>新闻数量</th>
                                <th>操作</th>
                            </tr>
                        </thead>
                        <tbody>
                            {% for category in categories %}
                                <tr data-pk="{{ category.id}}" data-name="{{ category.name }}">
```

```
                                        <td>{{ category.name }}</td>
                                        <td>{{ category.news_nums }}</td>
                                        <td>
                                            <button class="btn btn-warning btn-xs edit-btn">编辑</button>
                                            <button class="btn btn-danger btn-xs delete-btn">删除</button>
                                        </td>
                                    </tr>
                                {% endfor %}
                            </tbody>
                        </table>
                    </div>
                </div>
            </div>
        </div>

        <div class="pagination">
            <!-- 分页 -->
            {% render_paginator categories %}
        </div>

        <!-- add cetagory -->
        <div class="shade hide" id="shade-hide"></div>
        <div class="login-box-body add-action hide" id="add-form">
            <p class="login-box-msg">添加新闻分类</p>
            <span class="glyphicon glyphicon-remove form-control-feedback" id="add-close" onclick="addClose()"></span>
            <form action="javascript:void(0)" method="post" id="add-form">
                {% csrf_token %}
                <div class="form-group has-feedback {% if not msg.status %}errorput{% endif %}">
                    <input type="text" class="form-control" name="name" placeholder="请输入新闻分类">
                </div>
                <div>
                    <div class="col-xs-7 add-btn">
                        <button type="submit" class="btn btn-primary btn-block btn-flat" onclick="addCetagory()">确定</button>
                    </div>
                </div>
            </form>
        </div>

    {% endblock %}

    {% block front-js %}
            <script>
            // add category
            function addAction() {
                $("#shade-hide").removeClass("hide");
                $("#add-form").removeClass("hide");
            }
            function addClose() {
                $("#shade-hide").addClass("hide");
                $("#add-form").addClass("hide");
            }
            function addCetagory() {
                var nameInput = $("input[name='name']").val();
                if(nameInput == ""){
                    alert("请输入新闻分类！")
                }
                else {
                    $.ajax({
                        cache:false,
                        type:'post',
                        async:true,
                        headers:{"X-CSRFToken":"{{ csrf_token }}"},  // 获取 csrf token
                        dataType:'json',
                        url:'{% url 'add_category' %}',
                        data:{'name':nameInput},
                        'success':function (result) {
                            console.log(result);
                            alert(result['message']);
```

```
                                if(result['status'] == true){
                                    window.location.href = '{% url "news_category" %}'
                                }
                            },
                            'fail':function (error) {
                                console.log(error);
                            }
                        })
                    }
                }

            </script>
        {% endblock %}
```

views.py：

views.py

```
    @require_POST
    def add_new_category(request):
        """新增-新闻分类"""
        name = request.POST.get('name')
        print("name:", name)
        exists = NewsCategory.objects.filter(name=name).exists()

        if exists:
            return JsonResponse({"status": False, "message": "该分类已经存在"})

        else:
            NewsCategory.objects.create(name=name)
            return JsonResponse({"status": True, "message": "分类添加成功"})
```

urls.py：

urls.py

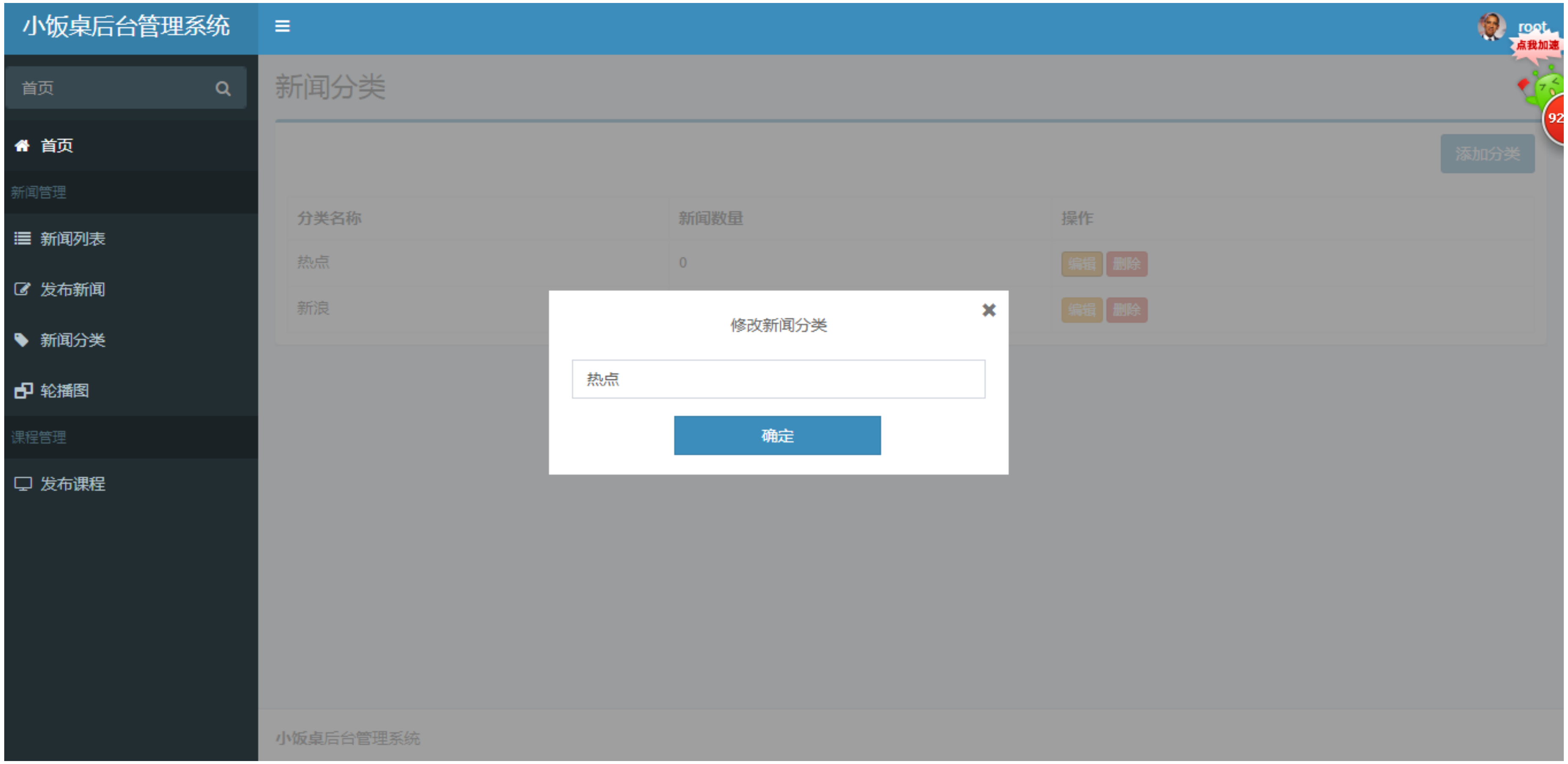
```
    from django.urls import path

    from crm.views import add_new_category


    urlpatterns = [

        path("add_category/", add_new_category, name='add_category'),    # crm 管理后台 新闻分类
```

3）修改新闻分类 、删除新闻分类

修改新闻分类、删除新闻分类，在本质来说实现方式是差不多的，当 修改/删除 某条新闻分类时，我们获取当条分类的分类名及 **id** 号，在生成的模态对话框中（修改/删除页面），将当前选择的当条分类的分类名及分类 **id** 填到对应的输入框中，其中 **id input** 框选择隐藏属性，目的只是将当条分类的 **id** 传回给后端，方便结合分类名对当条新闻分类进行相关修改或删除操作，在实现删除功能能，分类名 **input** 框最好改成 **p** 标签等，起到用户不能修改数据的作用。

关于修改新闻分类、删除新闻分类功能实现，结合新增新闻分类相关功能代码，具体实现如下（新闻分类 新增/修改/删除 全部前后端代码），关于项目所用的 **js**、**css** 及其他相关文件均会在项目博文最后提供下载地址

修改操作界面：



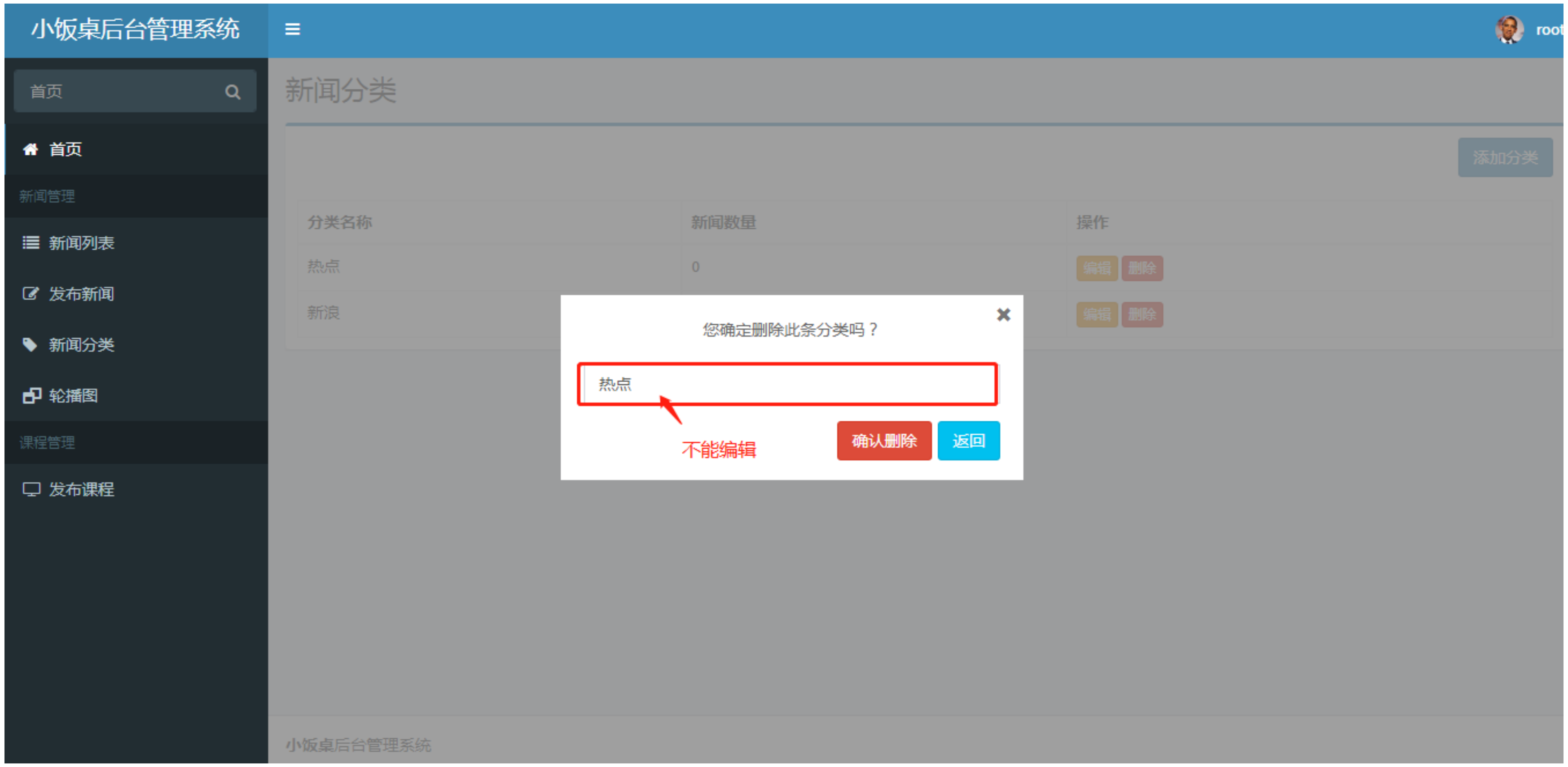
分类名称	新闻数量	操作
热点	0	编辑 删除
新浪		编辑 删除

修改新闻分类

热点

确定

删除操作界面：



HTML 前端代码：

news_category.html

```
{% extends 'crm/base.html' %}
{% load crm_tags %}

{% block title %}
    新闻分类
{% endblock %}

{% block head %}
    <link rel="stylesheet" href="{% static 'css/news_category/category.min.css'%}">
    {#   <script src="{% static 'js/news_category.min.js' %}"></script>#}
    {% endblock %}

    {% block content-header %}
        <h1>新闻分类</h1>
    {% endblock %}

    {% block content %}
        <div class="row">
            <div class="col-md-12">
                <div class="box box-primary">
                    <div class="box-header">
                        <button id="add-btn" class="btn btn-primary pull-right" onclick="addAction()">添加分类</button>
                    </div>
                    <div class="box-body">
                        <table class="table table-bordered">
                            <thead>
                                <tr>
                                    <th>分类名称</th>
                                    <th>新闻数量</th>
                                    <th>操作</th>
                                </tr>
                            </thead>
                            <tbody>
                                {% for category in categories %}
                                    <tr data-pk="{{ category.id}}" data-name="{{ category.name }}">
                                        <td>{{ category.name }}</td>
                                        <td>{{ category.news_nums }}</td>
                                        <td>
                                            <button class="btn btn-warning btn-xs edit-btn" onclick="editAction(this)">编辑</bu
tton>
                                            <button class="btn btn-danger btn-xs delete-btn" onclick="deleteAction(this)">删除<
/button>
                                        </td>
                                        </tr>
                                    {% endfor %}
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>

            <div class="pagination">
                <!-- 分页 -->
                {% render_paginator categories %}
            </div>

            <!-- add category -->
            <div class="shade hide" id="add-category"></div>
            <div class="login-box-body add-action hide" id="add-form">
                <p class="login-box-msg">添加新闻分类</p>
                <span class="glyphicon glyphicon-remove form-control-feedback close-icon" id="add-close" onclick="addClose()"></spa
n>
                <form action="javascript:void(0)" method="post" id="add-form">
                    {% csrf_token %}
                    <div class="form-group has-feedback">
                        <input type="text" class="form-control" name="name" placeholder="请输入新闻分类" id="add-input">
                    </div>
                    <div>
                        <div class="col-xs-7 add-btn">
                            <button type="submit" class="btn btn-primary btn-block btn-flat" id="add-submit">确定</button>
                        </div>
                    </div>
                </form>
            </div>

            <!-- edit category -->
```

```
        <div class="shade hide" id="edit-category"></div>
        <div class="login-box-body add-action hide" id="edit-form">
            <p class="login-box-msg">修改新闻分类</p>
            <span class="glyphicon glyphicon-remove form-control-feedback close-icon" id="edit-close" onclick="editClose()"></s
pan>
            <form action="javascript:void(0)" method="post">
                {% csrf_token %}
                <div class="form-group has-feedback">
                    <input type="text" class="form-control" name="name" placeholder="请输入新闻分类" id="edit-input">
                    <input type="hidden" class="form-control" name="uid" placeholder="编号" id="edit-uid">
                </div>
                <div>
                    <div class="col-xs-7 add-btn">
                        <button type="submit" class="btn btn-primary btn-block btn-flat" id="edit-submit">确定</button>
                    </div>
                </div>
            </form>
        </div>


        <!-- delete category -->
        <div class="shade hide" id="delete-category"></div>
        <div class="login-box-body add-action hide" id="delete-form">
            <p class="login-box-msg">您确定删除此条分类吗？</p>
            <span class="glyphicon glyphicon-remove form-control-feedback close-icon" id="delete-close" onclick="deleteClose()
"></span>
            <form action="javascript:void(0)" method="post">
                {% csrf_token %}
                <div class="form-group has-feedback">
                    <p class="delete-category form-control"></p>
                    <input type="hidden" class="form-control" name="uid" placeholder="编号" id="delete-uid">
                </div>
                <div>
                    <span class="btn btn-info pull-right" onclick="backAction()">返回</span>
```

```
                        <input type="submit" class="btn btn-danger pull-right margin-r-5" value="确认删除" onclick="deleteSubmit()
">
                    </div>
                </form>
            </div>

        {% endblock %}

        {% block front-js %}
            <script>

            // add category
            function addAction() {
                $("#add-category").removeClass("hide");
                $("#add-form").removeClass("hide");
                $(".pagination").addClass("hide");
            }
            function addClose() {
                $("#add-category").addClass("hide");
                $("#add-form").addClass("hide");
                $(".pagination").removeClass("hide");
            }
            // add-category submit
            $("#add-submit").click(function () {
                var nameInput = $("#add-input").val();
                if(nameInput == ""){
                    alert("请输入新闻分类！")
                }
                else {
                    $.ajax({
                        cache:false,
                        type:'post',
                        async:true,
                        headers:{"X-CSRFToken":"{{ csrf_token }}"},  // 获取 csrf token
                        dataType:'json',
                        url:"{% url 'add_category' %}",
                        data:{'name':nameInput},
                        'success':function (result) {
                            alert(result['message']);
                            if(result['status'] == true){
                                window.location.href = '{% url "news_category" %}'
                            }
                        },
                        'fail':function (error) {
                            console.log(error);
                        }
                    })
                }
            });


            // edit category
            function editAction(self) {
                var tr = $(self).parent().parent();
                var uid = tr.attr('data-pk');   // id
                var name = tr.attr('data-name');   // name
                $("#edit-input").val(name);   //将 name 填充到修改新闻分类的 name 栏上， 不用这种方式
                // $("input[name='name']").attr('placeholder',name);   //将 name 填充到修改新闻分类的 name 栏上,placeholder 形式
                $("#edit-uid").val(uid);   //将 id 号填充到修改分类的隐藏栏上，目的是为了给后台提供当前修改的某条数据
                $("#edit-category").removeClass("hide");
                $("#edit-form").removeClass("hide");
                $(".pagination").addClass("hide");
            }
            function editClose() {
                $("#edit-category").addClass("hide");
                $("#edit-form").addClass("hide");
                $(".pagination").removeClass("hide");
            }
            // edit-category submit
            $("#edit-submit").click(function () {
                var nameInput = $("#edit-input").val();
                var uidInput = $("#edit-uid").val();
                if(nameInput == ""){
                    alert("请输入新闻分类！")
                }
                else {
                    $.ajax({
                        cache:false,
                        type:'post',
                        async:true,
                        headers:{"X-CSRFToken":"{{ csrf_token }}"},  // 获取 csrf token
                        dataType:'json',
                        url:"{% url 'edit_category' %}",
                        data:{'name':nameInput, 'uid':uidInput},
                        'success':function (result) {
                            alert(result['message']);
                            if(result['status'] == true){
                                window.location.href = '{% url "news_category" %}'
                            }
                        },
                        'fail':function (error) {
                            console.log(error);
                        }
                    })
                }
            });


            // delete category
            function deleteAction(self) {
                var tr = $(self).parent().parent();
                var uid = tr.attr('data-pk');   // id
                var name = tr.attr('data-name');   // name
                // $("#delete-input").val(name);   //将 name 填充到隐藏 name 栏上， 用于提交数据到后台
                $(".delete-category").text(name);
                $("#delete-uid").val(uid);   //将 id 号填充到删除分类的隐藏栏上，目的是为了给后台提供当前删除的某条数据
                $("#delete-category").removeClass("hide");
                $("#delete-form").removeClass("hide");
                $(".pagination").addClass("hide");
            }
            function deleteClose() {
                $("#delete-category").addClass("hide");
                $("#delete-form").addClass("hide");
                $(".pagination").removeClass("hide");
            }
            function backAction() {
                $("#delete-category").addClass("hide");
                $("#delete-form").addClass("hide");
                $(".pagination").removeClass("hide");
            }
            // delete-category submit
            function deleteSubmit() {
                var nameInput = $(".delete-category").text();
                var uidInput = $("#delete-uid").val();   // 结合 uid、name 双重判断，这样就算前端修改了 name 数值，id 不通过也无法删
    除数据，保证安全
                    $.ajax({
                        cache:false,
                        type:'post',
                        async:true,
                        headers:{"X-CSRFToken":"{{ csrf_token }}"},  // 获取 csrf token
                        dataType:'json',
                        url:"{% url 'delete_category' %}",
                        data:{'name':nameInput, 'uid':uidInput},
                        'success':function (result) {
                            alert(result['message']);
                            if(result['status'] == true){
                                window.location.href = '{% url "news_category" %}'
                            }
                        },
                        'fail':function (error) {
                            console.log(error);
                        }
                    })
                }

                </script>
            {% endblock %}
```

views.py：

新闻分类 新增/修改/删除

```
    from django.shortcuts import render, redirect
    from django.contrib.admin.views.decorators import staff_member_required   # 是否为后台员工识别装饰器
    from django.views.decorators.http import require_POST, require_GET
    from django.http import JsonResponse
    from django.core.paginator import Paginator, PageNotAnInteger, EmptyPage


    @staff_member_required(login_url='index')
    @require_GET
    def new_category(request):
        """新闻分类详情"""
        # if request.user.is_superuser:
        #     categories = NewsCategory.objects.all()  # 返回所有分类
        # else:
        #     categories = NewsCategory.objects.filter(news__author=request.user)   # 返回当前用户的分类
        # return render(request, 'crm/news_category.html', locals())
        categories = NewsCategory.objects.all().order_by("-id")
        # 分页（分页功能）
        paginator = Paginator(categories, 2)  # 每页显示 2 行数据

        page = request.GET.get('_page')
        try:
            categories = paginator.page(page)
        except PageNotAnInteger:
            # If page is not an integer, deliver first page.
            categories = paginator.page(1)
        except EmptyPage:
            # If page is out of range (e.g. 9999), deliver last page of results.
            categories = paginator.page(paginator.num_pages)  # paginator.num_pages：总页数，即返回最后一页
        return render(request, 'crm/news_category.html', locals())


    @require_POST
    def add_new_category(request):
        """新增-新闻分类"""
        name = request.POST.get('name')
        exists = NewsCategory.objects.filter(name=name).exists()

        if exists:
            return JsonResponse({"status": False, "message": "该分类已经存在"})

        else:
            NewsCategory.objects.create(name=name)
            return JsonResponse({"status": True, "message": "分类添加成功"})


    @require_POST
    def edit_new_category(request):
        """修改-新闻分类"""
        uid = request.POST.get('uid')
        name = request.POST.get('name')
        exists = NewsCategory.objects.filter(name=name).exists()

        if exists:
            return JsonResponse({"status": False, "message": "该分类已经存在"})

        else:
            NewsCategory.objects.filter(id=uid).update(name=name)
            return JsonResponse({"status": True, "message": "分类编辑完成"})


    @require_POST
    def delete_new_category(request):
        """删除-新闻分类"""
        name = request.POST.get('name')
        id = request.POST.get('uid')
        try:
            exists = NewsCategory.objects.filter(name=name, id=id)[0]
        except Exception as e:
            exists = None

        if exists:
            exists.delete()
            return JsonResponse({"status": True, "message": "该分类已被删除"})

        else:
            return JsonResponse({"status": False, "message": "该分类不存在！"})
```

urls.py：

urls.py

```
    from django.urls import path

    from crm.views import new_category, add_new_category, edit_new_category, delete_new_category


    urlpatterns = [

        path("category/", new_category, name='news_category'),   # crm 管理后台 新闻分类
        path("add_category/", add_new_category, name='add_category'),   # crm 管理后台 添加新闻分类
        path("edit_category/", edit_new_category, name='edit_category'),   # crm 管理后台 修改新闻分类
        path("delete_category/", delete_new_category, name='delete_category'),   # crm 管理后台 删除新闻分类

    ]
```

models.py：

news/models.py

```
class NewsCategory(models.Model):
    """新闻分类"""
    name = models.CharField("新闻分类", max_length=40)
    add_time = models.DateTimeField("添加时间", default=datetime.now)

    class Meta:
        verbose_name = "新闻分类"
        verbose_name_plural = verbose_name

    def news_nums(self):
        """该分类下新闻数量"""
        return self.news_set.all().count()

    def __str__(self):
        return self.name
```

crm_tag.py 自定义分页标签：

自定义分页标签：crm_tags.py

```
from django.template import Library
from django.utils.safestring import mark_safe

register = Library()


@register.simple_tag
def render_paginator(querysets):
    """
    分页功能
    从 views 中拿到 querysets
    paginator = Paginator(querysets, 2) ：一页显示 2 行
    querysets = paginator.page(page)：当前页码
    """

    ele = '''
        <nav aria-label="Page navigation">
            <ul class="pagination">
                <li>
                    <a href="?_page=1" aria-label="shouye">
                        <span aria-hidden="true">首页</span>
                    </a>
                </li>
    '''
    if querysets.has_previous():
        p_ele = '''
        <li><a href="?_page=%s" aria-label="Previous">&laquo;上一页</a></li>
        ''' % (querysets.previous_page_number())
        ele += p_ele
    # querysets.paginator=paginator，page_range:页数范围
    for i in querysets.paginator.page_range:
        # querysets.number:当前页码

        if abs(querysets.number - i) < 4:# 只显示相邻页码，最多 3 页
            active = ''
            # 当前页
            if querysets.number ==i:
                active = 'active'

            p_ele = '''
            <li class="%s"><a href="?_page=%s">%s</a></li>
            '''% (active, i, i)
```

```
                    ele += p_ele
                # 是否有下一页
            if querysets.has_next():
                p_ele = '''
                    <li><a href="?_page=%s" aria-label="Next">下一页&raquo;</a></li>
                     ''' % (querysets.next_page_number())
                ele += p_ele

                # querysets.paginator.num_pages：总页数
            p_ele = '''
                <li>
                    <a href="?_page=%s" aria-label="weiye">
                        <span aria-hidden="true">尾页</span>
                    </a>
                </li>
            '''%(querysets.paginator.num_pages)
            ele += p_ele

            ele += "</ul></nav>"
            return mark_safe(ele)
```
未来的你，会感谢现在努力的你！