

```
# Create your views here.
from django.shortcuts import render #视图渲染模块
from django.http import HttpResponse #请求模块
from . import models #数据库操作模块
from django.db.models import Q #数据库逻辑模块
from django.db.models import Avg,Max,Min,Sum #数据库聚合计算模块
from datetime import datetime,timedelta #Cookie 模块
from django.http import HttpResponse, HttpResponseRedirect #重定向模块
from django.core.paginator import Paginator #分页器
from django.shortcuts import render
from django.utils import timezone #时间处理模块
from django.http import JsonResponse
from django.core import serializers
from urllib import parse#转码
import re #正则模块
import random#随机模块
import hashlib# 加密模块
import datetime#时间
import time# 日期模块
import os
import sys
import json
from . import dngadmin_common #公共函数
def admingroup(request):
    # -----
    # 判断页面权限》》 开始
    # -----

    dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
    get_url=dngadmin_common.dng_ckurl(request)[1]

    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') #提示信息
    jinggao = request.GET.get('jinggao') # 警告信息
    yes = request.GET.get('yes') # 成功信息
    xitong = dngadmin_common.dng_setup()# 系统设置

    if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
```

```
max_age=None)

        salt=dngadmin_common.dng_anquan().salt_str,

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
    else:
        urlstr = parse.quote('您需要重新登录')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response

    group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会员组名称
    groupall = dngadmin_common.html_groupall()
    #groupall=zip(groupall[0], groupall[1])
    dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息
    dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

    if not dngroute.url_str in mulu_url: #判断 URL 统一
        urlstr = parse.quote('您的访问与菜单映射不匹配')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response

    elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
        urlstr = parse.quote('您没有访问这个菜单的权限')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response

    added =dngadmin_common.dng_zsgc(dngroute_uid,group.added_text) #增
    delete = dngadmin_common.dng_zsgc(dngroute_uid,group.delete_text) #删
    update = dngadmin_common.dng_zsgc(dngroute_uid,group.update_text)#改
    see = dngadmin_common.dng_zsgc(dngroute_uid,group.see_text) #查

    zd_list = dngadmin_common.dng_ziduan("admingroup") #获取对应表下所有字段值
    # -----
    # 判断页面权限开始 《 《 《 结束
    # -----
    #biaodan = dngadmin_common.dng_dnguser(uid=dnguser_uid)

    vis = zip(zd_list[0], zd_list[1])
    return render(request,"dngadmin/admingroup.html",{
        "title":dngroute.name_str,
        "xitong": xitong, # 系统配置
        "dnguser_uid": dnguser_uid,
        "groupall": groupall,
```

```

    "get_url": get_url, #
    "added": added, #增
    "delete": delete, #删
    "update": update, #改
    "see": see, #查
    "zd_list": zd_list, # 字段名称
    "vist": vis,

    "tishi": tishi,
    "jinggao": jinggao,
    "yes": yes,
    "yuming_url": yuming_url,
})

```

```

def admingroup_json(request):
    # -----
    # 判断页面权限》》》 开始
    # -----

    dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
    get_url = dngadmin_common.dng_ckurl(request)[1]

    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') # 提示信息
    jinggao = request.GET.get('jinggao') # 警告信息
    yes = request.GET.get('yes') # 成功信息
    xitong = dngadmin_common.dng_setup() # 系统设置

    if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response

```

```

else:
    urlstr = parse.quote('您需要重新登录')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会员组名称
dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: # 判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查

zd_list = dngadmin_common.dng_ziduan("admingroup") # 获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----

page = request.GET.get('page', '') # 第几页
limit = request.GET.get('limit', '') # 每页显示多少

page = int(page)
limit = int(limit)
dngred = models.dngusergroup.objects.filter().order_by('-id')
paginator = Paginator(dngred, limit)

list_count = paginator.count # 总数据量
num_pages = paginator.num_pages # 有多少分页

book_list = paginator.page(page)
data = []
for key in book_list:
    #group = dngadmin_common.html_usergroup(gid=dngadmin_common.html_user(key.uperior_int).group_int)
    data.append({"id": str(key.id),
                 "gid_int": str(key.gid_int), # 用户组 id
                 "gname_str": str(key.gname_str), # 用户组名称
                 "uperior_int": str(key.uperior_int), # 上级用户组
                 "integral_int": str(key.integral_int), # 积分阈值
    })

```

```

"money_int": str(key.money_int), # 余额阈值
"totalmoney_int": str(key.totalmoney_int), # 充值阈值
"totalspend_int": str(key.totalspend_int), # 消费阈值
"spread_int": str(key.spread_int), # 推广阈值
"added_int": str(key.added_int), # 每日新增
"look_int": str(key.look_int), # 每日查看
"space_int": str(key.space_int), # 每日上传
"download_int": str(key.download_int), # 每日下载
"trial_bool": str(key.trial_bool), # 自动过审
"upload_bool": str(key.upload_bool), # 上传权限
"download_bool": str(key.download_bool), # 下载权限
}

```

#下面开始构造 JSON 格式

```

datajson = """{
    "code": 0
    , "msg": ""
    , "count": ""+str(list_count)+"""
    , "data": ""+str(data)+"""}"""

```

datajson = datajson.replace('\'', '\') #替换成 AJAX 可以解析得格式

datajson = datajson.replace('True', '启用')

datajson = datajson.replace('False', '关闭')

return HttpResponse(datajson)

def admingroup_added(request): #新增

```

# -----
# 判断页面权限》》》 开始
# -----

```

dngrount_uid = dngadmin_common.dng_ckurl(request)[0]

get_url = dngadmin_common.dng_ckurl(request)[1]

ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息

liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息

yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名

geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL

mulu_url = request.path # 获取不包含? 号之前的映射路径

tishi = request.GET.get('tishi') #提示信息

jingao = request.GET.get('jingao') # 警告信息

yes = request.GET.get('yes') # 成功信息

xitong = dngadmin_common.dng_setup()# 系统设置

if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转

dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,

salt=dngadmin_common.dng_anquan().salt_str,

max_age=None)

dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,

```

        salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
        salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

        if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
            if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
                urlstr = parse.quote('不允许同时登录账号')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response
            else:
                urlstr = parse.quote('您需要重新登录')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response

        group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
        员组名称
        dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息
        dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

        if not dngroute.url_str in mulu_url: # 判断 URL 统一
            urlstr = parse.quote('您的访问与菜单映射不匹配')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response

        elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
            urlstr = parse.quote('您没有访问这个菜单的权限')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response

        added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
        delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
        update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
        see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查

        zd_list = dngadmin_common.dng_ziduan("admingroup") # 获取对应表下所有字段值
        # -----
        # 判断页面权限开始 《 《 《 结束
        # -----

        post2 = request.POST.get(zd_list[0][2], '')
        post3 = request.POST.get(zd_list[0][3], '')

        ok = '新增失败'
        off = False
        pws =models.dngusergroup.objects.filter(gname_str=post2).only('gname_str')
        if not post3:
            post3 = 0

```

```

        pid = True
    else:
        pid = models.dngusergroup.objects.filter(gid_int=post3).only('gid_int')
    if added:
        if post3 == '0':
            user = models.dngusergroup.objects.filter().order_by('-gid_int').first()
            uid = int(user.gid_int) + 1 # 会员 ID 递增
            models.dngusergroup.objects.create(gid_int=uid, gname_str=post2, uperior_int=post3, )
        elif not post2:
            off = '用户组不能为空'

        elif pws:
            off = '已经有重复用户组, 请更换'
        elif not pid:

            off = '上级用户组不存在'

    else:

        user = models.dngusergroup.objects.filter().order_by('-gid_int').first()
        uid = int(user.gid_int)+1 #会员 ID 递增
        models.dngusergroup.objects.create(gid_int=uid, gname_str=post2, uperior_int=post3,)

    if off:
        ok = off
    else:
        ok = 'yes'
else:
    ok = '您没有新增权限'

return HttpResponse(ok)

def admingroup_delete(request): #删除
    # -----
    # 判断页面权限》》》 开始
    # -----

    dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
    get_url = dngadmin_common.dng_ckurl(request)[1]

    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') #提示信息
    jinggao = request.GET.get('jinggao') # 警告信息
    yes = request.GET.get('yes') # 成功信息
    xitong = dngadmin_common.dng_setup()# 系统设置

```

```

if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
    dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                            salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                            salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                            salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
    else:
        urlstr = parse.quote('您需要重新登录')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response

    group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称
    dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息
    dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: # 判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查

zd_list = dngadmin_common.dng_ziduan("admingroup") # 获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----

delete_id = request.GET.get('delete_id') #接收单个删除

```



```
id = request.POST.get('id', '') #接收批量删除

ok = '删除失败'

if delete: #判断删除权限

    if delete_id: #判断单个删除
        uxxg = models.dngusergroup.objects.filter(id=delete_id).first()
        udhg = models.dnguser.objects.filter(group_int=uxxg.gid_int).first()
        if udhg:
            off = "用户组下还有用户, 不能删除"
            return HttpResponse(off)

        ok = models.dngusergroup.objects.filter(id=delete_id).delete()

    if ok:
        ok='yes'
    else:
        ok = 'no'

elif id: #判断批量删除

    str(id)
    id = id.replace("[", '') # 去两边
    id = id.replace("]", '') # 去两边
    id = id.replace("\", '') # 去两边
    if ", " in id :
        id =id.split(",") # 分割
        for key in id:
            int(key)
            uxxg = models.dngusergroup.objects.filter(id=key).first()
            udhg = models.dnguser.objects.filter(group_int=uxxg.gid_int).first()
            if udhg:
                off = "用户组下还有用户, 不能删除"
                return HttpResponse(off)

            ok = models.dngusergroup.objects.filter(id=key).delete()
            if ok:
                ok = 'yes'
            else:
                ok = 'no'
    else:
        int(id)

        uxxg = models.dngusergroup.objects.filter(id=id).first()
        udhg = models.dnguser.objects.filter(group_int=uxxg.gid_int).first()

        if udhg:
            off = "用户组下还有用户, 不能删除"
```

```

        return HttpResponse(off)

    ok = models.dngusergroup.objects.filter(id=id).delete()
    if ok:
        ok = 'yes'
    else:
        ok = 'no'

else:
    ok = parse.quote('您没有删除权限')
    return HttpResponse(ok)
def admingroup_update(request): #更新修改
    # -----
    # 判断页面权限》》》 开始
    # -----

    dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
    get_url=dngadmin_common.dng_ckurl(request)[1]

    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') #提示信息
    jinggao = request.GET.get('jinggao') # 警告信息
    yes = request.GET.get('yes') # 成功信息
    xitong = dngadmin_common.dng_setup()# 系统设置

    if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

        if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
            if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
                urlstr = parse.quote('不允许同时登录账号')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response
            else:
                urlstr = parse.quote('您需要重新登录')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response

```

```
group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会员组名称
```

```
dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数
```

```
if not dngroute.url_str in mulu_url: #判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response
```

```
elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response
```

```
added =dngadmin_common.dng_zsgc(dngroute_uid,group.added_text) #增
delete = dngadmin_common.dng_zsgc(dngroute_uid,group.delete_text) #删
update = dngadmin_common.dng_zsgc(dngroute_uid,group.update_text)#改
see = dngadmin_common.dng_zsgc(dngroute_uid,group.see_text) #查
```

```
zd_list = dngadmin_common.dng_ziduan("admingroup") #获取对应表下所有字段值
```

```
# -----
# 判断页面权限开始 《 《 《 结束
# -----
```

```
post0 = request.POST.get(zd_list[0][0], '')
post1 = request.POST.get(zd_list[0][1], '')
post2 = request.POST.get(zd_list[0][2], '')
post3 = request.POST.get(zd_list[0][3], '')
post4 = request.POST.get(zd_list[0][4], '')
post5 = request.POST.get(zd_list[0][5], '')
post6 = request.POST.get(zd_list[0][6], '')
post7 = request.POST.get(zd_list[0][7], '')
post8 = request.POST.get(zd_list[0][8], '')
post9 = request.POST.get(zd_list[0][9], '')
post10 = request.POST.get(zd_list[0][10], '')
post11 = request.POST.get(zd_list[0][11], '')
post12 = request.POST.get(zd_list[0][12], '')
post13 = request.POST.get(zd_list[0][13], '')
post14 = request.POST.get(zd_list[0][14], '')
post15 = request.POST.get(zd_list[0][15], '')
```

```
ok = '修改失败'
off=False
if update:
```

```
if not post0:

    off = "空 ID"
    exit() # 终止

if post1:
    models.dngusergroup.objects.filter(id=post0).update(gid_int=post1)

if post2:
    models.dngusergroup.objects.filter(id=post0).update(gname_str=post2)

if post3:

    uxxg = models.dngusergroup.objects.filter(gid_int=post3).first()

    if post3=='0':
        models.dngusergroup.objects.filter(id=post0).update(uperior_int=post3)
    elif not uxxg :
        off = "没有这个用户组 ID"
        return HttpResponse(off)
    elif post3==post1:
        off = "上级用户组就是本用户组"
        return HttpResponse(off)
    models.dngusergroup.objects.filter(id=post0).update(uperior_int=post3)

if post4:
    models.dngusergroup.objects.filter(id=post0).update(integral_int=post4)

if post5:
    models.dngusergroup.objects.filter(id=post0).update(money_int=post5)

if post6:
    models.dngusergroup.objects.filter(id=post0).update(totalmoney_int=post6)

if post7:
    models.dngusergroup.objects.filter(id=post0).update(totalspend_int=post7)

if post8:
    models.dngusergroup.objects.filter(id=post0).update(spread_int=post8)

if post9:
    models.dngusergroup.objects.filter(id=post0).update(added_int=post9)

if post10:
    models.dngusergroup.objects.filter(id=post0).update(look_int=post10)

if post11:
    models.dngusergroup.objects.filter(id=post0).update(space_int=post11)

if post12:
```

```
models.dngusergroup.objects.filter(id=post0).update(download_int=post12)

if post13:
    if post13=='on':
        post13=True
        models.dngusergroup.objects.filter(id=post0).update(trial_bool=post13)
    elif post13=='off':
        post13 = False
        models.dngusergroup.objects.filter(id=post0).update(trial_bool=post13)

if post14:
    if post14=='on':
        post14=True
        models.dngusergroup.objects.filter(id=post0).update(upload_bool=post14)
    elif post14=='off':
        post14 = False
        models.dngusergroup.objects.filter(id=post0).update(upload_bool=post14)

if post15:
    if post15=='on':
        post15=True
        models.dngusergroup.objects.filter(id=post0).update(download_bool=post15)
    elif post15=='off':
        post15 = False
        models.dngusergroup.objects.filter(id=post0).update(download_bool=post15)

ok = 'yes'
else:
    ok = ' 您没有修改权限'

return HttpResponse(ok, off)

def admingroup_search(request): #搜索

# -----
# 判断页面权限》》》 开始
# -----

dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
get_url = dngadmin_common.dng_ckurl(request)[1]

ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
```

```

mulu_url = request.path # 获取不包含? 号之前的映射路径
tishi = request.GET.get('tishi') # 提示信息
jinggao = request.GET.get('jinggao') # 警告信息
yes = request.GET.get('yes') # 成功信息
xitong = dngadmin_common.dng_setup() # 系统设置

if "dnguser_uid" in request.COOKIEs: # 判断 cookies 有无, 跳转
    dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
        else:
            urlstr = parse.quote('您需要重新登录')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response

    group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称
    dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息
    dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: # 判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查

zd_list = dngadmin_common.dng_ziduan("admingroup") # 获取对应表下所有字段值
# -----

```

```
# 判断页面权限开始 《 《 《 结束
# -----

post1 = request.GET.get(zd_list[0][1], '') # 会员 ID
post2 = request.GET.get(zd_list[0][2], '') # 会员账号
post3 = request.GET.get(zd_list[0][3], '') # 会员密码
post4 = request.GET.get(zd_list[0][4], '') # 昵称
post5 = request.GET.get(zd_list[0][5], '') # 性别
post6 = request.GET.get(zd_list[0][6], '') # 个人简介
post7 = request.GET.get(zd_list[0][7], '') # 邮箱
post8 = request.GET.get(zd_list[0][8], '') # 手机号
post9 = request.GET.get(zd_list[0][9], '') # 用户组
post10 = request.GET.get(zd_list[0][10], '') # 等级
post11 = request.GET.get(zd_list[0][11], '')
post12 = request.GET.get(zd_list[0][12], '') # 余额
post13 = request.GET.get(zd_list[0][13], '') # 累计充值
post14 = request.GET.get(zd_list[0][14], '') # 累计消费
post15 = request.GET.get(zd_list[0][15], '') # 积分
page = request.GET.get('page', '') # 第几页
limit = request.GET.get('limit', '') # 每页显示多少

page = int(page)
limit = int(limit)

if post1:

    dngred = models.dngusergroup.objects.filter(gid_int__contains=post1).order_by('-id')

elif post2:

    dngred = models.dngusergroup.objects.filter(gname_str__contains=post2).order_by('-id')

elif post3:

    dngred = models.dngusergroup.objects.filter(uperior_int__contains=post3).order_by('-id')

elif post4:

    dngred = models.dngusergroup.objects.filter(integral_int__contains=post4).order_by('-id')

elif post5:

    dngred = models.dngusergroup.objects.filter(money_int__contains=post5).order_by('-id')

elif post6:

    dngred = models.dngusergroup.objects.filter(totalmoney_int__contains=post6).order_by('-id')

elif post7:
```

```
    dngred = models.dngusergroup.objects.filter(totalspend_int__contains=post7).order_by('-id')

elif post8:

    dngred = models.dngusergroup.objects.filter(spread_int__contains=post8).order_by('-id')

elif post9:

    dngred = models.dngusergroup.objects.filter(added_int__contains=post9).order_by('-id')

elif post10:

    dngred = models.dngusergroup.objects.filter(look_int__contains=post10).order_by('-id')

elif post11:

    dngred = models.dngusergroup.objects.filter(space_int__contains=post11).order_by('-id')

elif post12:

    dngred = models.dngusergroup.objects.filter(download_int__contains=post12).order_by('-id')

elif post13:

    dngred = models.dngusergroup.objects.filter(trial_bool__contains=post13).order_by('-id')

elif post14:

    dngred = models.dngusergroup.objects.filter(upload_bool__contains=post14).order_by('-id')

elif post15:

    dngred = models.dngusergroup.objects.filter(download_bool__contains=post15).order_by('-id')

else:
    dngred = models.dngusergroup.objects.filter().order_by('-id')

paginator = Paginator(dngred, limit)

list_count = paginator.count # 总数据量
num_pages = paginator.num_pages # 有多少分页

book_list = paginator.page(page)

data = []
for key in book_list:
    #group = dngadmin_common.html_usergroup(gid=dngadmin_common.html_user(key.uid_int).group_int)
    data.append({"id": str(key.id),
```



```

"gid_int": str(key.gid_int), # 用户组 id
"gname_str": str(key.gname_str), # 用户组名称
"uperior_int": str(key.uperior_int), # 上级用户组
"integral_int": str(key.integral_int), # 积分阈值
"money_int": str(key.money_int), # 余额阈值
"totalmoney_int": str(key.totalmoney_int), # 充值阈值
"totalspend_int": str(key.totalspend_int), # 消费阈值
"spread_int": str(key.spread_int), # 推广阈值
"added_int": str(key.added_int), # 每日新增
"look_int": str(key.look_int), # 每日查看
"space_int": str(key.space_int), # 每日上传
"download_int": str(key.download_int), # 每日下载
"trial_bool": str(key.trial_bool), # 自动过审
"upload_bool": str(key.upload_bool), # 上传权限
"download_bool": str(key.download_bool), # 下载权限
})

```

下面开始构造 JSON 格式

```

datajson = """{
    "code": 0
    , "msg": ""
    , "count": "" + str(list_count) + ""
    , "data": "" + str(data) + ""}"""

datajson = datajson.replace('\'', '\\\'') # 替换成 AJAX 可以解析得格式
datajson = datajson.replace('True', '启用')
datajson = datajson.replace('False', '关闭')

return HttpResponse(datajson)

```

Create your views here.

```

from django.shortcuts import render #视图渲染模块
from django.http import HttpResponse #请求模块
from . import models #数据库操作模块
from django.db.models import Q #数据库逻辑模块
from django.db.models import Avg, Max, Min, Sum #数据库聚合计算模块
from datetime import datetime, timedelta #Cookie 模块
from django.http import HttpResponse, HttpResponseRedirect #重定向模块
from django.core.paginator import Paginator #分页器
from django.shortcuts import render
from django.utils import timezone #时间处理模块
from django.http import JsonResponse
from django.core import serializers
from urllib import parse #转码
import re #正则模块
import random #随机模块
import hashlib #加密模块
import datetime #时间

```

```

import time# 日期模块
import os
import sys
import json
from . import dngadmin_common #公共函数

def adminmenu(request):
    # -----
    # 判断页面权限》》》 开始
    # -----

    dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
    get_url=dngadmin_common.dng_ckurl(request)[1]

    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') #提示信息
    jinggao = request.GET.get('jinggao') # 警告信息
    yes = request.GET.get('yes') # 成功信息
    xitong = dngadmin_common.dng_setup()# 系统设置

    if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

        if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
            if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
                urlstr = parse.quote('不允许同时登录账号')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response
            else:
                urlstr = parse.quote('您需要重新登录')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response

    group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会

```

员组名称

```
groupall = dngadmin_common.html_groupall()
#groupall=zip(groupall[0], groupall[1])
dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数
```

```
if not dngroute.url_str in mulu_url: #判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) #增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) #删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) #改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) #查

zd_list = dngadmin_common.dng_ziduan("adminmenu") #获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----
```

```
#biaodan = dngadmin_common.dng_dnguser(uid=dnguser_uid)
```

```
vis = zip(zd_list[0], zd_list[1])
return render(request, "dngadmin/adminmenu.html", {
    "title": dngroute.name_str,
    "xitong": xitong, # 系统配置
    "dnguser_uid": dnguser_uid,
    "groupall": groupall,

    "get_url": get_url, #
    "added": added, #增
    "delete": delete, #删
    "update": update, #改
    "see": see, #查
    "zd_list": zd_list, # 字段名称
    "vist": vis,

    "tishi": tishi,
    "jinggao": jinggao,
    "yes": yes,
```

```
"yuming_url": yuming_url,
```

```
})
```

```
def adminmenu_json(request):
```

```
    # -----
    # 判断页面权限》》》 开始
    # -----
```

```
    dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
    get_url = dngadmin_common.dng_ckurl(request)[1]
```

```
    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') # 提示信息
    jinggao = request.GET.get('jinggao') # 警告信息
    yes = request.GET.get('yes') # 成功信息
    xitong = dngadmin_common.dng_setup() # 系统设置
```

```
    if "dnguser_uid" in request.COOKIEs: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
```

```
max_age=None)
```

```
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
```

```
max_age=None)
```

```
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
```

```
max_age=None)
```

```
    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
```

```
    else:
```

```
        urlstr = parse.quote('您需要重新登录')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response
```

```
    group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
```

员组名称

```
dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
```

日记记录函数

```
if not dngroute.url_str in mulu_url: # 判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response
```

```
added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查
```

```
zd_list = dngadmin_common.dng_ziduan("adminmenu") # 获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----
```

```
page = request.GET.get('page', '') # 第几页
limit = request.GET.get('limit', '') # 每页显示多少
```

```
page = int(page)
limit = int(limit)
```

```
dngred = models.dngroute.objects.filter().order_by('-id')
paginator = Paginator(dngred, limit)
```

```
list_count = paginator.count # 总数据量
num_pages = paginator.num_pages # 有多少分页
```

```
book_list = paginator.page(page)
```

```
data = []
for key in book_list:
```

```
    #group = dngadmin_common.html_usergroup(gid=dngadmin_common.html_user(key.uperior_int).group_int)
```

```
    data.append({"id": str(key.id),
                "uid_int": str(key.uid_int), # 菜单 id
                "name_str": str(key.name_str), # 菜单名称
```

```

"url_str": str(key.url_str), #菜单 URL
"icon_str": str(key.icon_str), #菜单图标
"model_str": str(key.model_str), #菜单模型
"superior_int": str(key.superior_int), #上级菜单
"sort_int": str(key.sort_int), #菜单排序
"integral_int": str(key.integral_int), #积分阈值
"money_int": str(key.money_int), #余额阈值
"totalmoney_int": str(key.totalmoney_int), #充值阈值
"totalspend_int": str(key.totalspend_int), #消费阈值
"spread_int": str(key.spread_int), #推广阈值
"display_bool": str(key.display_bool), #菜单显示
})

```

#下面开始构造 JSON 格式

```

datajson = """{
    "code": 0
    , "msg": ""
    , "count": ""+str(list_count)+"""
    , "data": ""+str(data)+"""}"""

```

```

datajson = datajson.replace('\'\'', '\') #替换成 AJAX 可以解析得格式

```

```

datajson = datajson.replace(' True', ' 启用')

```

```

datajson = datajson.replace(' False', ' 关闭')

```

```

return HttpResponse(datajson)

```

```

def adminmenu_added(request): #新增

```

```

# -----
# 判断页面权限》》》 开始
# -----

```

```

dngroute_uid = dngadmin_common.dng_ckurl(request)[0]

```

```

get_url = dngadmin_common.dng_ckurl(request)[1]

```

```

ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息

```

```

liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息

```

```

yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名

```

```

geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL

```

```

mulu_url = request.path # 获取不包含? 号之前的映射路径

```

```

tishi = request.GET.get('tishi') #提示信息

```

```

jinggao = request.GET.get('jinggao') # 警告信息
yes = request.GET.get('yes') # 成功信息
xitong = dngadmin_common.dng_setup()# 系统设置

if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
    dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
        else:
            urlstr = parse.quote('您需要重新登录')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response

    group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称
    dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息
    dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: # 判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查

zd_list = dngadmin_common.dng_ziduan("adminmenu") # 获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----

```

```

post2 = request.POST.get(zd_list[0][2], '')#名称
post3 = request.POST.get(zd_list[0][3], '')#URL
post5 = request.POST.get(zd_list[0][5], '')#菜单模型
post6 = request.POST.get(zd_list[0][6], '')#上级菜单

ok = '新增失败'
off = False
pname = models.dngroute.objects.filter(name_str=post2).only('name_str')
pwurl = models.dngroute.objects.filter(url_str=post3).only('url_str')

if added:

    if not post2:
        off = '菜单名称不能为空'

    elif pname:
        off = '菜单名称不能重复'
    elif pwurl:

        off = '已经有重复菜单 URL 路径'

    else:

        users = models.dngusergroup.objects.filter(gid_int=1).first() # 读用户组权限
        ider = models.dngroute.objects.filter().order_by('-uid_int').first() # 读 ID
        sortdb = models.dngroute.objects.filter().order_by('-sort_int').first() # 读排序

        uid = int(ider.uid_int) + 1 # 会员 ID 递增
        sort = int(sortdb.sort_int) + 10 # 排序递增
        qx_menu = users.menu_text + "|" + str(uid) + "|"
        qx_added = users.added_text + "|" + str(uid) + "|"
        qx_delete = users.delete_text + "|" + str(uid) + "|"
        qx_update = users.update_text + "|" + str(uid) + "|"
        qx_see = users.see_text + "|" + str(uid) + "|"

        models.dngroute.objects.create(uid_int=uid, name_str=post2, url_str=post3, model_str=post5,
                                         superior_int=post6, sort_int=sort) # 增加菜单表
        models.dngusergroup.objects.filter(gid_int=1).update(menu_text=qx_menu, added_text=qx_added,
                                                              delete_text=qx_delete,
update_text=qx_update,
                                                              see_text=qx_see) # 更新管理员
        用户组权限

    if off:
        ok = off
    else:
        ok = 'yes'
else:
    ok = '您没有新增权限'

```



```

return HttpResponseRedirect(ok)

def adminmenu_delete(request): #删除
    # -----
    # 判断页面权限》》》 开始
    # -----

    dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
    get_url = dngadmin_common.dng_ckurl(request)[1]

    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') #提示信息
    jinggao = request.GET.get('jinggao') # 警告信息
    yes = request.GET.get('yes') # 成功信息
    xitong = dngadmin_common.dng_setup() # 系统设置

    if "dnguser_uid" in request.COOKIE: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

        if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
            if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
                urlstr = parse.quote('不允许同时登录账号')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response
            else:
                urlstr = parse.quote('您需要重新登录')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response

        group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称
        dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息

```

```
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数
```

```
if not dngroute.url_str in mulu_url: # 判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response
```

```
added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查
```

```
zd_list = dngadmin_common.dng_ziduan("adminmenu") # 获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----
```

```
delete_id = request.GET.get('delete_id') #接收单个删除
```

```
id = request.POST.get('id', '') #接收批量删除
```

```
ok = '删除失败'
```

```
if delete: #判断删除权限
```

```
    if delete_id: #判断单个删除
        uxxg = models.dngroute.objects.filter(id=delete_id).first()
        udhg = models.dngroute.objects.filter(superior_int=uxxg.uid_int).first()
        if udhg:
            off = "下面还有二级菜单, 不能删除"
            return HttpResponseRedirect(off)

        ok = models.dngroute.objects.filter(id=delete_id).delete()

        if ok:
            ok='yes'
        else:
            ok = 'no'
```

```
    elif id: #判断批量删除
```

```
str(id)
id = id.replace("[", '') # 去两边
id = id.replace("]", '') # 去两边
id = id.replace("\'", '') # 去两边
if ", " in id :
    id = id.split(", ") # 分割
    for key in id:
        int(key)
        uxxg = models.dngroute.objects.filter(id=key).first()
        udhg = models.dngroute.objects.filter(superior_int=uxxg.uid_int).first()
        if udhg:
            off = "下面还有二级菜单, 不能删除"
            return HttpResponse(off)

        ok = models.dngroute.objects.filter(id=key).delete()
        if ok:
            ok = 'yes'
        else:
            ok = 'no'

else:
    int(id)

    uxxg = models.dngroute.objects.filter(id=id).first()
    udhg = models.dngroute.objects.filter(superior_int=uxxg.uid_int).first()
    if udhg:
        off = "下面还有二级菜单, 不能删除"
        return HttpResponse(off)

    ok = models.dngroute.objects.filter(id=id).delete()
    if ok:
        ok = 'yes'
    else:
        ok = 'no'

else:
    ok = parse.quote(' 您没有删除权限')

return HttpResponse(ok)

def adminmenu_update(request): #更新修改
    # -----
```

```

# 判断页面权限》》》 开始
# -----

dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
get_url=dngadmin_common.dng_ckurl(request)[1]

ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
mulu_url = request.path # 获取不包含? 号之前的映射路径
tishi = request.GET.get('tishi') #提示信息
jinggao = request.GET.get('jinggao') # 警告信息
yes = request.GET.get('yes') # 成功信息
xitong = dngadmin_common.dng_setup()# 系统设置

if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
    dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
    else:
        urlstr = parse.quote('您需要重新登录')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response

group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称

dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: #判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

```

```

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) #增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) #删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) #改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) #查

zd_list = dngadmin_common.dng_ziduan("adminmenu") #获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----
post0 = request.POST.get(zd_list[0][0], '')
post1 = request.POST.get(zd_list[0][1], '')
post2 = request.POST.get(zd_list[0][2], '')
post3 = request.POST.get(zd_list[0][3], '')
post4 = request.POST.get(zd_list[0][4], '')
post5 = request.POST.get(zd_list[0][5], '')
post6 = request.POST.get(zd_list[0][6], '')
post7 = request.POST.get(zd_list[0][7], '')
post8 = request.POST.get(zd_list[0][8], '')
post9 = request.POST.get(zd_list[0][9], '')
post10 = request.POST.get(zd_list[0][10], '')
post11 = request.POST.get(zd_list[0][11], '')
post12 = request.POST.get(zd_list[0][12], '')
post13 = request.POST.get(zd_list[0][13], '')

ok = '修改失败'
off=False
if update:

    if not post0:

        off = "空 ID"
        exit() # 终止

    if post1:
        models.dngroute.objects.filter(id=post0).update(uid_int=post1) # 菜单 id
    if post2:
        models.dngroute.objects.filter(id=post0).update(name_str=post2) # 菜单名称
    if post3:
        models.dngroute.objects.filter(id=post0).update(url_str=post3) # 菜单 URL
    if post4:
        models.dngroute.objects.filter(id=post0).update(icon_str=post4) # 菜单图标
    if post5:
        models.dngroute.objects.filter(id=post0).update(model_str=post5) # 菜单模型

```

```
if post6:
    models.dngroute.objects.filter(id=post0).update(superior_int=post6) # 上级菜单
if post7:
    models.dngroute.objects.filter(id=post0).update(sort_int=post7) # 菜单排序
if post8:
    models.dngroute.objects.filter(id=post0).update(integral_int=post8) # 积分阈值
if post9:
    models.dngroute.objects.filter(id=post0).update(money_int=post9) # 余额阈值
if post10:
    models.dngroute.objects.filter(id=post0).update(totalmoney_int=post10) # 充值阈值
if post11:
    models.dngroute.objects.filter(id=post0).update(totalspend_int=post11) # 消费阈值
if post12:
    models.dngroute.objects.filter(id=post0).update(spread_int=post12) # 推广阈值

if post13:
    if post13=='on':
        post13=True
        models.dngroute.objects.filter(id=post0).update(display_bool=post13) # 菜单显示
    elif post13=='off':
        post13 = False
        models.dngroute.objects.filter(id=post0).update(display_bool=post13) # 菜单显示

ok = 'yes'
else:
    ok = '您没有修改权限'

return HttpResponse(ok, off)

def adminmenu_search(request): #搜索

# -----
# 判断页面权限》》》 开始
# -----

dngroute_uid = dngadmin_common.dng_ckurl(request)[0]
get_url = dngadmin_common.dng_ckurl(request)[1]

ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
mulu_url = request.path # 获取不包含? 号之前的映射路径
tishi = request.GET.get('tishi') #提示信息
```

```

jinggao = request.GET.get('jinggao') # 警告信息
yes = request.GET.get('yes') # 成功信息
xitong = dngadmin_common.dng_setup()# 系统设置

if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
    dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                            salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                            salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                            salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
        else:
            urlstr = parse.quote('您需要重新登录')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response

    group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会员组名称
    dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first() # 查询路径取回本页面菜单信息
    dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: # 判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: # 判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = dngadmin_common.dng_zsgc(dngroute_uid, group.added_text) # 增
delete = dngadmin_common.dng_zsgc(dngroute_uid, group.delete_text) # 删
update = dngadmin_common.dng_zsgc(dngroute_uid, group.update_text) # 改
see = dngadmin_common.dng_zsgc(dngroute_uid, group.see_text) # 查

zd_list = dngadmin_common.dng_ziduan("adminmenu") # 获取对应表下所有字段值
# -----
# 判断页面权限开始 《 《 《 结束
# -----

```

```
post1 = request.GET.get(zd_list[0][1], '')
post2 = request.GET.get(zd_list[0][2], '')
post3 = request.GET.get(zd_list[0][3], '')
post4 = request.GET.get(zd_list[0][4], '')
post5 = request.GET.get(zd_list[0][5], '')
post6 = request.GET.get(zd_list[0][6], '')
post7 = request.GET.get(zd_list[0][7], '')
post8 = request.GET.get(zd_list[0][8], '')
post9 = request.GET.get(zd_list[0][9], '')
post10 = request.GET.get(zd_list[0][10], '')
post11 = request.GET.get(zd_list[0][11], '')
post12 = request.GET.get(zd_list[0][12], '')
post13 = request.GET.get(zd_list[0][13], '')

page = request.GET.get('page', '') # 第几页
limit = request.GET.get('limit', '') # 每页显示多少

page = int(page)
limit = int(limit)

if post1:
    dngred = models.dngroute.objects.filter(uid_int__contains=post1).order_by('-id')
elif post1:
    dngred = models.dngroute.objects.filter(uid_int__contains=post1).order_by('-id')
elif post2:
    dngred = models.dngroute.objects.filter(name_str__contains=post2).order_by('-id')
elif post3:
    dngred = models.dngroute.objects.filter(url_str__contains=post3).order_by('-id')
elif post4:
    dngred = models.dngroute.objects.filter(icon_str__contains=post4).order_by('-id')
elif post5:
    dngred = models.dngroute.objects.filter(model_str__contains=post5).order_by('-id')
elif post6:
    dngred = models.dngroute.objects.filter(superior_int__contains=post6).order_by('-id')
elif post7:
    dngred = models.dngroute.objects.filter(sort_int__contains=post7).order_by('-id')
elif post8:
    dngred = models.dngroute.objects.filter(integral_int__contains=post8).order_by('-id')
elif post9:
    dngred = models.dngroute.objects.filter(money_int__contains=post9).order_by('-id')
elif post10:
    dngred = models.dngroute.objects.filter(totalmoney_int__contains=post10).order_by('-id')
elif post11:
    dngred = models.dngroute.objects.filter(totalspend_int__contains=post11).order_by('-id')
elif post12:
    dngred = models.dngroute.objects.filter(spread_int__contains=post12).order_by('-id')
```



```

elif post13:
    dngred = models.dngroute.objects.filter(display_bool__contains=post13).order_by('-id')

else:
    dngred = models.dngroute.objects.filter().order_by('-id')

paginator = Paginator(dngred, limit)

list_count = paginator.count # 总数据量
num_pages = paginator.num_pages # 有多少分页

book_list = paginator.page(page)

data = []
for key in book_list:
    #group = dngadmin_common.html_usergroup(gid=dngadmin_common.html_user(key.uid_int).group_int)
    data.append({"id": str(key.id),
                "uid_int": str(key.uid_int), # 菜单 id
                "name_str": str(key.name_str), # 菜单名称
                "url_str": str(key.url_str), # 菜单 URL
                "icon_str": str(key.icon_str), # 菜单图标
                "model_str": str(key.model_str), # 菜单模型
                "superior_int": str(key.superior_int), # 上级菜单
                "sort_int": str(key.sort_int), # 菜单排序
                "integral_int": str(key.integral_int), # 积分阈值
                "money_int": str(key.money_int), # 余额阈值
                "totalmoney_int": str(key.totalmoney_int), # 充值阈值
                "totalspend_int": str(key.totalspend_int), # 消费阈值
                "spread_int": str(key.spread_int), # 推广阈值
                "display_bool": str(key.display_bool), # 菜单显示
                })

# 下面开始构造 JSON 格式
datajson = """{
    "code": 0
    , "msg": ""
    , "count": """ + str(list_count) + """
    , "data": """ + str(data) + """}"""

datajson = datajson.replace('\ ', '\') # 替换成 AJAX 可以解析得格式
datajson = datajson.replace('True', '启用')
datajson = datajson.replace('False', '关闭')

return HttpResponse(datajson)

from django.db import models
import random#随机模块
# str 后缀 == models.CharField_字符串
# text 后缀 == models.TextField_富文本编辑
# int 后缀 == models.IntegerField_整形数字
# float 后缀 == models.FloatField_浮点小数点类型

```

bool 后缀 == models.BooleanField_布尔真假类型

time == models.DateTimeField_日期+时间

#注意事项：（一）不要用自增 id，来作为用户 ID 或者业务 id，不少新手都会这种方法，会使得业务与 id 生成强耦合，导致 id 生成算法难以升级，未来分布式数据库，和分表都会麻烦

#注意事项：（二）不要修改已经建立的数据库字段，会带来未知风险，建议对字段新增，不要删除修改系统已经存在的数据库字段，

#注意事项：（三）创建字段名称记得带类型后缀，方便前台识别，生成对应表单输入样式

#注意事项：（四）你不确定未来会迁移什么类型数据库，为了保证通用，尽量全部小写，慎用驼峰命名法，数据库高手忽略此条

class user(models.Model):#前台会员表

uid_int = models.IntegerField(blank=False, verbose_name='会员 ID')#会员 ID，设置不能为空

username_str = models.CharField(max_length=255, unique=True, blank=False, verbose_name='会员账号')

#会员账号，unique 不能重复，不许为空

password_str = models.CharField(max_length=255, blank=False, verbose_name='会员密码')#会员密码 MD5

加密，不许为空

name_str = models.CharField(max_length=255, blank=True, verbose_name='昵称')#会员昵称

gender_str = models.CharField(max_length=255, blank=True, verbose_name='性别')#性别，默认空

introduce_str = models.CharField(max_length=255, blank=True, verbose_name='个人简介')#个人简介，默

认空

email_str = models.CharField(max_length=255, blank=True, verbose_name='邮箱')#会员邮箱

mobile_str = models.CharField(max_length=255, blank=True, verbose_name='手机号')#手机号接收短信

等

group_int = models.IntegerField(default=2, verbose_name='用户组')#填写用户组 ID

rank_str = models.CharField(max_length=255, blank=True, verbose_name='等级')

gm_bool = models.BooleanField(default=False, verbose_name='前台管理') # 账号超级管理员开关，

False=不是超级管理 True=是超级管理员

money_int = models.IntegerField(default=0, verbose_name='余额')#余额，默认值为 0，不支持小数点

totalmoney_int = models.IntegerField(default=0, verbose_name='累计充值') # 默认值为 0，不支持小数点

totalspend_int = models.IntegerField(default=0, verbose_name='累计消费') # 默认值为 0，不支持小数点

integral_int = models.IntegerField(default=0, verbose_name='积分')#积分，默认值为 0

spread_int = models.IntegerField(default=0, verbose_name='推广注册') # 默认值为 0，不支持小数点

ip_str = models.CharField(max_length=255, blank=True, verbose_name='登录 IP')#登录 ip 地址

shebei_str = models.CharField(max_length=255, blank=True, verbose_name='登录设备')#登录后台设备

cookie_str = models.CharField(max_length=255, blank=True, verbose_name='cookie') # 后台客户

cookie

token_str = models.CharField(max_length=255, blank=True, verbose_name='token 密钥')#后台客户

token 密钥，预留加密授权登录用

days_int = models.IntegerField(default=0, verbose_name='登录天数')#登录天数

pwderror_int = models.IntegerField(default=0, verbose_name='密错次数')#密码错误次数

frozen_bool = models.BooleanField(default=False, verbose_name='禁止登录')#账号限制登录，False=

没有禁止 True=账号禁止

frozentime_str = models.CharField(max_length=255, blank=True, verbose_name='冻结时间')#冻结

vipime_time = models.DateTimeField(blank=True, default='2099-12-28 23:59:59', verbose_name='登录

时限') # 登录有效期，开通一年有效期，半年有效期会员账号用

create_time = models.DateTimeField(auto_now_add=True, verbose_name='创建时间')#后台注册时间

update_time = models.DateTimeField(auto_now=True, verbose_name='更新时间')#最后一次登录时间

```
class dnguser(models.Model):#后台会员表
```

```

    uid_int          = models.IntegerField(blank=False, verbose_name='会员 ID')#会员 ID, 设置不能为空
    username_str      = models.CharField(max_length=255, unique=True, blank=False, verbose_name='会员账号')
#会员账号, unique 不能重复, 不许为空
    password_str      = models.CharField(max_length=255, blank=False, verbose_name='会员密码')#会员密码 MD5
加密, 不许为空
    name_str          = models.CharField(max_length=255, blank=True, verbose_name='昵称')#会员昵称
    gender_str         = models.CharField(max_length=255, blank=True, verbose_name='性别')#性别, 默认空
    introduce_str      = models.CharField(max_length=255, blank=True, verbose_name='个人简介')#个人简介, 默
认空
    emall_str          = models.CharField(max_length=255, blank=True, verbose_name='邮箱')#会员邮箱
    mobile_str         = models.CharField(max_length=255, blank=True, verbose_name='手机号')#手机号接收短信
等
    group_int          = models.IntegerField(default=2, verbose_name='用户组')#填写用户组 ID
    rank_str           = models.CharField(max_length=255, blank=True, verbose_name='等级')
    gm_bool            = models.BooleanField(default=False, verbose_name='超级管理') # 账号超级管理员开关,
False=不是超级管理 True=是超级管理员
    money_int           = models.IntegerField(default=0, verbose_name='余额')#余额, 默认值为 0, 不支持小数点
    totalmoney_int     = models.IntegerField(default=0, verbose_name='累计充值') # 默认值为 0, 不支持小数点
    totalspend_int     = models.IntegerField(default=0, verbose_name='累计消费') # 默认值为 0, 不支持小数点
    integral_int       = models.IntegerField(default=0, verbose_name='积分')#积分, 默认值为 0
    spread_int         = models.IntegerField(default=0, verbose_name='推广注册')#默认值为 0, 不支持小数点
    ip_str             = models.CharField(max_length=255, blank=True, verbose_name='登录 IP')#登录 ip 地址
    shebei_str         = models.CharField(max_length=255, blank=True, verbose_name='登录设备')#登录后台设备
    cookie_str         = models.CharField(max_length=255, blank=True, verbose_name='cookie') # 后台客户
cookie
    token_str          = models.CharField(max_length=255, blank=True, verbose_name='token 密钥')#后台客户
token 密钥, 预留加密授权登录用
    days_int           = models.IntegerField(default=0, verbose_name='登录天数')#登录天数
    pwderror_int       = models.IntegerField(default=0, verbose_name='密错次数')#密码错误次数
    frozen_bool        = models.BooleanField(default=True, verbose_name='允许登录')#账号限制登录, False=没
有禁止 True=账号禁止
    frozentime_str     = models.CharField(max_length=255, blank=True, verbose_name='冻结时间')#冻结
    vipime_time        = models.DateTimeField(blank=True, default='2099-12-28 23:59:59', verbose_name='登录
时限') # 登录有效期, 开通一年有效期, 半年有效期会员账号用
    create_time        = models.DateTimeField(auto_now_add=True, verbose_name='创建时间')#后台注册时间
    update_time        = models.DateTimeField(auto_now=True, verbose_name='更新时间')#最后一次登录时间

```

```
class usergroup(models.Model):#前台会员组表
```

```

    gid_int           = models.IntegerField(blank=False, unique=True, verbose_name='用户组 id')
    gname_str         = models.CharField(max_length=255, unique=True, blank=False, verbose_name='用户组名称
')
    uperior_int       = models.IntegerField(default=0, verbose_name='上级用户组')#0 没有上级, 填写菜单 ID
    integral_int      = models.IntegerField(default=0, verbose_name='积分阈值')
    money_int          = models.IntegerField(default=0, verbose_name='余额阈值')
    totalmoney_int    = models.IntegerField(default=0, verbose_name='充值阈值')
    totalspend_int    = models.IntegerField(default=0, verbose_name='消费阈值')
    spread_int        = models.IntegerField(default=0, verbose_name='推广阈值')

```

```

added_int      = models.IntegerField(default=0, verbose_name='每日新增')
look_int       = models.IntegerField(default=0, verbose_name='每日查看')
space_int      = models.IntegerField(default=0, verbose_name='每日上传')
download_int   = models.IntegerField(default=0, verbose_name='每日下载')
trial_bool     = models.BooleanField(default=False, verbose_name='自动过审')
upload_bool    = models.BooleanField(default=False, verbose_name='上传权限')
download_bool  = models.BooleanField(default=False, verbose_name='下载权限')
menu_text      = models.TextField(blank=True, verbose_name='菜单权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
added_text     = models.TextField(blank=True, verbose_name='新增权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
delete_text    = models.TextField(blank=True, verbose_name='删除权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
update_text    = models.TextField(blank=True, verbose_name='修改权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
see_text       = models.TextField(blank=True, verbose_name='查看权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|, PS:拥有菜单权限, 就默认可以查看菜单, 此查看权限是, 方便设置此菜单下一些查看带星号手机之类的
权限

```

```

class dngusergroup(models.Model): #后台用户组表

```

```

    gid_int      = models.IntegerField(blank=False, unique=True, verbose_name='用户组 id')
    gname_str    = models.CharField(max_length=255, unique=True, blank=False, verbose_name='用户组名称')

    uperior_int  = models.IntegerField(default=0, verbose_name='上级用户组') #0 没有上级, 填写菜单 ID
    integral_int = models.IntegerField(default=0, verbose_name='积分阈值')
    money_int     = models.IntegerField(default=0, verbose_name='余额阈值')
    totalmoney_int = models.IntegerField(default=0, verbose_name='充值阈值')
    totalspend_int = models.IntegerField(default=0, verbose_name='消费阈值')
    spread_int   = models.IntegerField(default=0, verbose_name='推广阈值')
    added_int    = models.IntegerField(default=0, verbose_name='每日新增')
    look_int     = models.IntegerField(default=0, verbose_name='每日查看')
    space_int    = models.IntegerField(default=0, verbose_name='每日上传')
    download_int = models.IntegerField(default=0, verbose_name='每日下载')
    trial_bool   = models.BooleanField(default=False, verbose_name='自动过审')
    upload_bool  = models.BooleanField(default=False, verbose_name='上传权限')
    download_bool = models.BooleanField(default=False, verbose_name='下载权限')
    menu_text    = models.TextField(blank=True, verbose_name='菜单权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
    added_text   = models.TextField(blank=True, verbose_name='新增权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
    delete_text = models.TextField(blank=True, verbose_name='删除权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
    update_text  = models.TextField(blank=True, verbose_name='修改权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|
    see_text     = models.TextField(blank=True, verbose_name='查看权限')#填写对应菜单 ID, 格式: |菜单
1||菜单 2||菜单 3|, PS:拥有菜单权限, 就默认可以查看菜单, 此查看权限是, 方便设置此菜单下一些查看带星号手机之类的
权限

```

```

class route(models.Model): #前台菜单表

```

```

uid_int          = models.IntegerField(blank=False, unique=True, verbose_name='菜单 id')
name_str         = models.CharField(max_length=255, unique=True, blank=False, verbose_name='菜单名称')
url_str          = models.CharField(max_length=255, blank=True, verbose_name='菜单 URL')
icon_str         = models.CharField(max_length=255, blank=True, default='fa fa-desktop', verbose_name='
菜单图标')
model_str        = models.CharField(max_length=255, blank=True, default='cover', verbose_name='菜单模
型')# list=数据列表页面, form=表单提交页面, cover=无属性封面, url = 单独链接菜单
superior_int     = models.IntegerField(default=0, verbose_name='上级菜单')
sort_int         = models.IntegerField(default=0, verbose_name='菜单排序')
integral_int     = models.IntegerField(default=0, verbose_name='积分门槛')
money_int        = models.IntegerField(default=0, verbose_name='余额门槛')
totalmoney_int   = models.IntegerField(default=0, verbose_name='充值门槛')
totalspend_int   = models.IntegerField(default=0, verbose_name='消费门槛')
spread_int       = models.IntegerField(default=0, verbose_name='推广门槛')
display_bool     = models.BooleanField(default=True, verbose_name='菜单显示')

```

```
class dngroute(models.Model): # 后台菜单表
```

```

uid_int          = models.IntegerField(blank=False, unique=True, verbose_name='菜单 id')
name_str         = models.CharField(max_length=255, unique=True, blank=False, verbose_name='菜单名称')
url_str          = models.CharField(max_length=255, blank=True, verbose_name='菜单 URL')
icon_str         = models.CharField(max_length=255, blank=True, default='fa fa-desktop', verbose_name='
菜单图标')
model_str        = models.CharField(max_length=255, blank=True, default='cover', verbose_name='菜单模
型')# list=数据列表页面, form=表单提交页面, cover=无属性封面, url = 单独链接菜单
superior_int     = models.IntegerField(default=0, verbose_name='上级菜单')
sort_int         = models.IntegerField(default=0, verbose_name='菜单排序')
integral_int     = models.IntegerField(default=0, verbose_name='积分门槛')
money_int        = models.IntegerField(default=0, verbose_name='余额门槛')
totalmoney_int   = models.IntegerField(default=0, verbose_name='充值门槛')
totalspend_int   = models.IntegerField(default=0, verbose_name='消费门槛')
spread_int       = models.IntegerField(default=0, verbose_name='推广门槛')
display_bool     = models.BooleanField(default=True, verbose_name='菜单显示')

```

```
class red(models.Model): #前台日志
```

```

uid_int          = models.IntegerField(blank=False, verbose_name='会员 id') # 所属会员的 ID
title_str        = models.CharField(max_length=255, blank=True, verbose_name='访问标题')
url_str          = models.CharField(max_length=255, blank=True, verbose_name='访问网址')
shebei_str       = models.CharField(max_length=255, blank=True, verbose_name='登录设备')
ip_str           = models.CharField(max_length=255, blank=True, verbose_name='登录 IP')
create_time      = models.DateTimeField(auto_now_add=True, verbose_name='创建时间')

```

```
class dngred(models.Model): #后台日志
```

```

uid_int          = models.IntegerField(blank=False, verbose_name='会员 id') # 所属会员的 ID
title_str        = models.CharField(max_length=255, blank=True, verbose_name='访问标题')
url_str          = models.CharField(max_length=255, blank=True, verbose_name='访问网址')
shebei_str       = models.CharField(max_length=255, blank=True, verbose_name='登录设备')

```

```
ip_str = models.CharField(max_length=255, blank=True, verbose_name='登录 IP')
create_time = models.DateTimeField(auto_now_add=True, verbose_name='创建时间')
```

```
class htmlsetup(models.Model): #前台设置
```

```
    title_str = models.CharField(max_length=255, blank=False, default='DngAdmin 开发框架',
verbose_name=' 首页标题')
    keywords_str = models.CharField(max_length=255, blank=True, default='DngAdmin 开发框架',
verbose_name='META 关键词')
    description_str = models.CharField(max_length=255, blank=True, default='DngAdmin 框架 3.0-基于 python
和 Django 的极速后台开发框架,为极速开发而生!', verbose_name='META 描述')
    file_str = models.CharField(max_length=255, blank=True, verbose_name='备案号') #备案号
    statistics_text = models.TextField(blank=True, verbose_name='统计代码') #统计代码
    register_bool = models.BooleanField(default=True, verbose_name='注册开关')

    inwidth_int = models.IntegerField(default=120, verbose_name='最小表宽')
    wide_int = models.IntegerField(default=800, verbose_name='弹窗宽度')
    high_int = models.IntegerField(default=600, verbose_name='弹窗高度')
    limit_int = models.IntegerField(default=15, verbose_name='默认条数')
    toolbar_bool = models.BooleanField(default=True, verbose_name='头工具栏')
    skinline_str = models.CharField(max_length=255, blank=True, verbose_name='表格边线')
    skinsize_str = models.CharField(max_length=255, blank=True, verbose_name='表格缩放')
    page_bool = models.BooleanField(default=True, verbose_name='底部分页')
    exports_str = models.CharField(max_length=255, blank=True, default='exports', verbose_name='导出表
格')
    print_str = models.CharField(max_length=255, blank=True, default='print', verbose_name='打印表
格')
    search_bool = models.BooleanField(default=True, verbose_name='搜索表格')
```

```
class setup(models.Model): #后台设置
```

```
    setupname_str = models.CharField(max_length=255, blank=False, default='DngAdmin 开发框架',
verbose_name='系统名称') #系统名称, 不许为空
    domain_str = models.CharField(max_length=255, blank=False, verbose_name='系统域名') #系统域名, 不
许为空
    file_str = models.CharField(max_length=255, blank=True, verbose_name='备案号') #备案号
    edition_str = models.CharField(max_length=255, blank=True, default='DngAdmin 版本 3.0',
verbose_name='版本号') #版本号
    statistics_text = models.TextField(blank=True, verbose_name='统计代码') #统计代码

    inwidth_int = models.IntegerField(default=160, verbose_name='最小表宽')
    wide_int = models.IntegerField(default=800, verbose_name='弹窗宽度')
    high_int = models.IntegerField(default=600, verbose_name='弹窗高度')
    limit_int = models.IntegerField(default=15, verbose_name='默认条数')
    toolbar_bool = models.BooleanField(default=True, verbose_name='头工具栏')
    skinline_str = models.CharField(max_length=255, blank=True, verbose_name='表格边线')
    skinsize_str = models.CharField(max_length=255, blank=True, verbose_name='表格缩放')
    page_bool = models.BooleanField(default=True, verbose_name='底部分页')
    exports_str = models.CharField(max_length=255, blank=True, default='exports', verbose_name='导出表
```

```

格')
    print_str = models.CharField(max_length=255, blank=True, default='print', verbose_name='打印表格')
    search_bool = models.BooleanField(default=True, verbose_name='搜索表格')

class protect(models.Model): #前台安全

    uid_int = models.IntegerField(blank=False, unique=True, verbose_name='安全 ID') # 安全策略的
ID, 设置不能为空, 不可重复
    entrance_str = models.CharField(max_length=255, blank=True, verbose_name='安全入口') #后台安全入口
    prescription_int = models.IntegerField(blank=True, default=86400, verbose_name='Cookies 时效') #Cookies
时效, 单位毫秒, 默认 24 小时
    salt_str = models.CharField(max_length=255, blank=True, verbose_name='加密盐') #解析 COOKIE 的
加密盐
    requests_int = models.IntegerField(blank=False, default=5, verbose_name='密错次数') #防暴力破解,
超过次数限制登录
    psdreq_int = models.IntegerField(blank=False, default=1, verbose_name='冻结时间') # 密码错误后
冻结, 单位小时
    graphic_bool = models.BooleanField(default=False, verbose_name='图形验开关') # 图形验证码开关,
False=关闭 True=开启
    station_bool = models.BooleanField(default=False, verbose_name='验证感知') # 跨站 POST 开关, False=
关闭 True=开启
    useragent_str = models.CharField(max_length=255, blank=True, verbose_name='允许设备') #允许 useragent
设备, 分割线|分割
    area_str = models.CharField(max_length=255, blank=True, verbose_name='允许地区') #允许登录得地区, 分割线|分割
    tongshi_bool = models.BooleanField(default=False, verbose_name='同时在线') #同时在线开关, False=
不允许同时 True=允许同时
    sms_bool = models.BooleanField(default=False, verbose_name='短信验证') #短信验证开关, False=
不开 True=开启
    iptxt_text = models.TextField(blank=True, verbose_name='禁止 IP') #富文本超大字符串, |符号分割

class security(models.Model): #后台安全

    uid_int = models.IntegerField(blank=False, unique=True, verbose_name='安全 ID') # 安全策略的
ID, 设置不能为空, 不可重复
    entrance_str = models.CharField(max_length=255, blank=True, verbose_name='安全入口') #后台安全入口
    prescription_int = models.IntegerField(blank=True, default=86400, verbose_name='Cookies 时效') #Cookies
时效, 单位毫秒, 默认 24 小时
    salt_str = models.CharField(max_length=255, blank=True, verbose_name='加密盐') #解析 COOKIE 的
加密盐
    requests_int = models.IntegerField(blank=False, default=5, verbose_name='密错次数') #防暴力破解,
超过次数限制登录
    psdreq_int = models.IntegerField(blank=False, default=1, verbose_name='冻结时间') # 密码错误后
冻结, 单位小时
    graphic_bool = models.BooleanField(default=False, verbose_name='图形验开关') # 图形验证码开关,
False=关闭 True=开启
    station_bool = models.BooleanField(default=False, verbose_name='验证感知') # 跨站 POST 开关, False=
关闭 True=开启
    useragent_str = models.CharField(max_length=255, blank=True, verbose_name='允许设备') #允许 useragent

```

设备, 分割线|分割

```
area_str = models.CharField(max_length=255, blank=True, verbose_name='允许地区') #允许登录得地
```

区, 分割线|分割

```
tongshi_bool = models.BooleanField(default=False, verbose_name='同时在线') #同时在线开关, False=
```

不允许同时 True=允许同时

```
sms_bool = models.BooleanField(default=False, verbose_name='短信验证') #短信验证开关, False=
```

不开 True=开启

```
iptxt_text = models.TextField(blank=True, verbose_name='禁止 IP') #富文本超大字符串, |符号分割
```

```
class mail(models.Model): #邮件设置
```

```
type_str = models.CharField(max_length=255, blank=True, verbose_name='邮件发送方式')
```

```
host_str = models.CharField(max_length=255, blank=True, verbose_name='SMTP 服务器')
```

```
port_str = models.CharField(max_length=255, blank=True, verbose_name='SMTP 端口')
```

```
user_str = models.CharField(max_length=255, blank=True, verbose_name='SMTP 用户名')
```

```
pass_str = models.CharField(max_length=255, blank=True, verbose_name='SMTP 密码')
```

```
verify_str = models.CharField(max_length=255, blank=True, verbose_name='SMTP 验证方式')
```

```
from_str = models.CharField(max_length=255, blank=True, verbose_name='发件人邮箱')
```

```
"""seo URL Configuration
```

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/2.2/topics/http/urls/>

Examples:

Function views

1. Add an import: from my_app import views

2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

```
"""
```

```
from django.conf.urls import url
```

```
from . import views
```

```
from . import dngadmin_index #后台首页
```

```
from . import dngadmin_datahome #数据仪表盘首页
```

```
from . import dngadmin_install #安装创建账号文件
```

```
from . import dngadmin_login #后台登录页
```

```
from . import dngadmin_tips #提示警告
```

```
from . import dngadmin_ip #ip 查询接口
```

```
from . import dngadmin_setup #系统后台设置
```

```
from . import dngadmin_htmlsetup #前台设置
```

```
from . import dngadmin_security #后台安全
```

```
from . import dngadmin_protect #前台安全
```

```
from . import dngadmin_userdata #个人资料
```

```
from . import dngadmin_userpsd #密码修改
```

```
from . import dngadmin_crud #一键 CRUD
```

```
from . import dngadmin_userjournal #登录日志
```



```

from . import dngadmin_adminuser    #后台用户管理
from . import dngadmin_htmluser     #前台会员管理
from . import dngadmin_admingroup   #后台用户组
from . import dngadmin_htmlgroup    #前台用户组
from . import dngadmin_adminmenu    #后台菜单
from . import dngadmin_htmlmenu     #前台菜单
from . import dngadmin_adminpower   #后台用户组权限
from . import dngadmin_htmlpower    #前台用户组权限
from . import dngadmin_shell        #shell 命令
from . import dngadmin_mail         #邮件
from . import dngadmin_sms          #短信设置
from . import dngadmin_imgcode      #图码设置
# ①映射替换 1①

urlpatterns = [    #路由映射

    url(r'^$', dngadmin_index.index),# '$' 后台主目录
    url(r'^datahome/', dngadmin_datahome.datahome), # 数据首页
    url(r'^setup/', dngadmin_setup.setup), # 后台系统设置
    url(r'^setup_post/', dngadmin_setup.setup_post), # 后台系统设置 post
    url(r'^htmlsetup/', dngadmin_htmlsetup.htmlsetup), # 前台设置
    url(r'^htmlsetup_post/', dngadmin_htmlsetup.htmlsetup_post), # 前台设置 post
    url(r'^security/', dngadmin_security.security), # 后台安全
    url(r'^security_post/', dngadmin_security.security_post), # 后台安全 post
    url(r'^protect/', dngadmin_protect.protect), # 前台安全
    url(r'^protect_post/', dngadmin_protect.protect_post), # 前台安全 post
    url(r'^userdata/', dngadmin_userdata.userdata), # 个人资料
    url(r'^userdata_post/', dngadmin_userdata.userdata_post), # 个人资料 post
    url(r'^userpsd/', dngadmin_userpsd.userpsd), # 密码修改
    url(r'^userpsd_post/', dngadmin_userpsd.userpsd_post), # 密码修改 post
    url(r'^crud/', dngadmin_crud.crud), # 一键 crud
    url(r'^crud_post/', dngadmin_crud.crud_post), # 一键 crud 的 post
    url(r'^userjournal/', dngadmin_userjournal.userjournal), # 登录日志
    url(r'^userjournal_json/', dngadmin_userjournal.userjournal_json), # 登录日志 json
    url(r'^userjournal_added/', dngadmin_userjournal.userjournal_added), # 登录新增
    url(r'^userjournal_delete/', dngadmin_userjournal.userjournal_delete), # 登录删除
    url(r'^userjournal_update/', dngadmin_userjournal.userjournal_update), # 登录更新修改
    url(r'^userjournal_search/', dngadmin_userjournal.userjournal_search), # 登录搜索
    url(r'^adminuser/', dngadmin_adminuser.adminuser), # 后台用户管理
    url(r'^adminuser_json/', dngadmin_adminuser.adminuser_json), # 后台用户管理 json
    url(r'^adminuser_added/', dngadmin_adminuser.adminuser_added), # 后台用户管理新增
    url(r'^adminuser_delete/', dngadmin_adminuser.adminuser_delete), # 后台用户管理删除
    url(r'^adminuser_update/', dngadmin_adminuser.adminuser_update), # 后台用户管理更新修改
    url(r'^adminuser_search/', dngadmin_adminuser.adminuser_search), # 后台用户管理搜索
    url(r'^htmluser/', dngadmin_htmluser.htmluser), # 前台会员管理
    url(r'^htmluser_json/', dngadmin_htmluser.htmluser_json), # 前台会员管理 json
    url(r'^htmluser_added/', dngadmin_htmluser.htmluser_added), # 前台会员管理新增
    url(r'^htmluser_delete/', dngadmin_htmluser.htmluser_delete), # 前台会员管理删除
    url(r'^htmluser_update/', dngadmin_htmluser.htmluser_update), # 前台会员管理更新修改
    url(r'^htmluser_search/', dngadmin_htmluser.htmluser_search), # 前台会员管理搜索

```

```
url(r'^admingroup/', dngadmin_admingroup.admingroup), # 后台用户组
url(r'^admingroup_json/', dngadmin_admingroup.admingroup_json), # 后台用户组 json
url(r'^admingroup_added/', dngadmin_admingroup.admingroup_added), # 后台用户组新增
url(r'^admingroup_delete/', dngadmin_admingroup.admingroup_delete), # 后台用户组删除
url(r'^admingroup_update/', dngadmin_admingroup.admingroup_update), # 后台用户组更新修改
url(r'^admingroup_search/', dngadmin_admingroup.admingroup_search), # 后台用户组搜索
url(r'^htmlgroup/', dngadmin_htmlgroup.htmlgroup), # 前台用户组
url(r'^htmlgroup_json/', dngadmin_htmlgroup.htmlgroup_json), # 前台用户组 json
url(r'^htmlgroup_added/', dngadmin_htmlgroup.htmlgroup_added), # 前台用户组新增
url(r'^htmlgroup_delete/', dngadmin_htmlgroup.htmlgroup_delete), # 前台用户组删除
url(r'^htmlgroup_update/', dngadmin_htmlgroup.htmlgroup_update), # 前台用户组更新修改
url(r'^htmlgroup_search/', dngadmin_htmlgroup.htmlgroup_search), # 前台用户组搜索
url(r'^adminpower/', dngadmin_adminpower.adminpower), # 后台用户组权限
url(r'^adminpower_json/', dngadmin_adminpower.adminpower_json), # 后台用户组权限 json
url(r'^adminpower_menu/', dngadmin_adminpower.adminpower_menu), # 后台用户组权限访问
url(r'^adminpower_added/', dngadmin_adminpower.adminpower_added), # 后台用户组权限新增
url(r'^adminpower_delete/', dngadmin_adminpower.adminpower_delete), # 后台用户组权限删除
url(r'^adminpower_update/', dngadmin_adminpower.adminpower_update), # 后台用户组权限更新修改
url(r'^adminpower_see/', dngadmin_adminpower.adminpower_see), # 后台用户组权限开发者
url(r'^adminpower_post/', dngadmin_adminpower.adminpower_post), # 后台用户组权限 POST
url(r'^htmlpower/', dngadmin_htmlpower.htmlpower), # 前台用户组权限
url(r'^htmlpower_json/', dngadmin_htmlpower.htmlpower_json), # 前台用户组权限 json
url(r'^htmlpower_menu/', dngadmin_htmlpower.htmlpower_menu), # 前台用户组权限访问
url(r'^htmlpower_added/', dngadmin_htmlpower.htmlpower_added), # 前台用户组权限新增
url(r'^htmlpower_delete/', dngadmin_htmlpower.htmlpower_delete), # 前台用户组权限删除
url(r'^htmlpower_update/', dngadmin_htmlpower.htmlpower_update), # 前台用户组权限更新修改
url(r'^htmlpower_see/', dngadmin_htmlpower.htmlpower_see), # 前台用户组权限开发者
url(r'^htmlpower_post/', dngadmin_htmlpower.htmlpower_post), # 前台用户组权限 POST
url(r'^adminmenu/', dngadmin_adminmenu.adminmenu), # 后台菜单
url(r'^adminmenu_json/', dngadmin_adminmenu.adminmenu_json), # 后台菜单 json
url(r'^adminmenu_added/', dngadmin_adminmenu.adminmenu_added), # 后台菜单新增
url(r'^adminmenu_delete/', dngadmin_adminmenu.adminmenu_delete), # 后台菜单删除
url(r'^adminmenu_update/', dngadmin_adminmenu.adminmenu_update), # 后台菜单更新修改
url(r'^adminmenu_search/', dngadmin_adminmenu.adminmenu_search), # 后台菜单搜索
url(r'^htmlmenu/', dngadmin_htmlmenu.htmlmenu), # 前台菜单
url(r'^htmlmenu_json/', dngadmin_htmlmenu.htmlmenu_json), # 前台菜单 json
url(r'^htmlmenu_added/', dngadmin_htmlmenu.htmlmenu_added), # 前台菜单新增
url(r'^htmlmenu_delete/', dngadmin_htmlmenu.htmlmenu_delete), # 前台菜单删除
url(r'^htmlmenu_update/', dngadmin_htmlmenu.htmlmenu_update), # 前台菜单更新修改
url(r'^htmlmenu_search/', dngadmin_htmlmenu.htmlmenu_search), # 前台菜单搜索
url(r'^shell/', dngadmin_shell.shell), # shell 命令
url(r'^shell_post/', dngadmin_shell.shell_post), # shell 命令的 post
url(r'^mail/', dngadmin_mail.mail), # 邮件
url(r'^mail_json/', dngadmin_mail.mail_json), # 邮件 json
url(r'^mail_added/', dngadmin_mail.mail_added), # 邮件新增
url(r'^mail_delete/', dngadmin_mail.mail_delete), # 邮件删除
url(r'^mail_update/', dngadmin_mail.mail_update), # 邮件更新修改
url(r'^mail_search/', dngadmin_mail.mail_search), # 邮件搜索
url(r'^sms/', dngadmin_sms.sms), # 短信设置
url(r'^sms_post/', dngadmin_sms.sms_post), # 短信设置的 post
```

```

url(r'^imgcode/', dngadmin_imgcode.imgcode), # 图码设置
url(r'^imgcode_post/', dngadmin_imgcode.imgcode_post), # 图码设置的 post
# ☉2 映射替换 2☉
url(r'^install/$', dngadmin_install.install),#创建管理员安装
url(r'^install_post/$', dngadmin_install.install_post),#创建管理员 POST
url(r'^longin/', dngadmin_longin.longin),#后台登录首页
url(r'^longin_post/$', dngadmin_longin.longin_post),#授权 COOKIE 页
url(r'^longin_out/', dngadmin_longin.longin_out),#后台退出
url(r'^tips', dngadmin_tips.tips),#提示页面
url(r'^ip/', dngadmin_ip.ip), #内置 IP 接口
url(r'^ip_json', dngadmin_ip.ip_json), #内置 IP 接口
url(r'^ip_post', dngadmin_ip.ip_post), # 内置 IP 接口

```

```

]
```

```

# Create your views here.
from django.shortcuts import render #视图渲染模块
from django.http import HttpResponse #请求模块
from . import models #数据库操作模块
from django.db.models import Q #数据库逻辑模块
from django.db.models import Avg,Max,Min,Sum #数据库聚合计算模块
from datetime import datetime,timedelta #Cookie 模块
from django.http import HttpResponse, HttpResponseRedirect #重定向模块
from django.shortcuts import render
import os
import sys
from urllib import parse#转码
import re #正则模块
import random#随机模块
import hashlib# 加密模块
from django.utils import timezone #时间处理模块
import datetime#时间
import time# 日期模块
from . import dngadmin_common #公共函数

```

```

def userdata(request):
    # -----
    # 通过路径获得栏目 ID 》》》开始
    # -----
    get_url = request.path
    get_url = get_url.split("/") # 分割
    get_url = get_url[1] + "/" + get_url[2]
    if "_" in get_url:
        url = get_url.split("_") # 分割
        get_url = url[0]

```

```

uid=models.dngroute.objects.filter(url_str__contains=get_url).first()
dngroute_uid = uid.uid_int
# -----
# 日记记录与 COOKIE 验证与权限 》》》开始
# -----
ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
mulu_url = request.path # 获取不包含? 号之前的映射路径
tishi = request.GET.get('tishi') #提示
jinggao = request.GET.get('jinggao') # 警告
yes = request.GET.get('yes') # 警告

if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
    dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
    dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                             salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

    if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
        if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
            urlstr = parse.quote('不允许同时登录账号')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response
        else:
            urlstr = parse.quote('您需要重新登录')
            response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
            return response

# -----
# 日记记录与 COOKIE 验证与权限 《《《 结束
# -----
# -----
# 判断页面权限开始》》》开始
# -----
group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称
dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息

dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

```

```
if not dngroute.url_str in mulu_url: #判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = False #增
delete = False #删
update = False #改
see = False #查

if '|' + str(dngroute_uid) + '|' in group.added_text: # 判断增加权限
    added = True
if '|' + str(dngroute_uid) + '|' in group.delete_text: # 判断删除权限
    delete = True
if '|' + str(dngroute_uid) + '|' in group.update_text: # 判断修改权限
    update = True
if '|' + str(dngroute_uid) + '|' in group.see_text: # 判断查看权限
    see = True

# -----
# 判断页面权限开始 《 《 《 结束
# -----

zd_list = dngadmin_common.dng_ziduan("dnguser") #获取对应表下所有字段值

biaodan = dngadmin_common.dng_dnguser(uid=dnguser_uid)

return render(request, "dngadmin/userdata.html", {
    "title": dngroute.name_str,
    "edition": dngadmin_common.dng_setup().edition_str, # 版本号
    "file": dngadmin_common.dng_setup().file_str, # 备案号
    "tongue": dngadmin_common.dng_setup().statistics_text, # 统计
    "group_gname": group.gname_str, # 用户组昵称
    "added": added, #增
    "delete": delete, #删
    "update": update, #改
    "see": see, #查
    "tishi": tishi,
    "jinggao": jinggao,
    "yes": yes,
    "zd_list": zd_list,
```

```

        "biaodan": biaodan,
        "yuming_url": yuming_url,

    })

```

```

def userdata_post(request):
    # -----
    #   通过路径获得栏目 ID 》》》开始
    # -----
    get_url = request.path
    get_url = get_url.split("/") # 分割
    get_url = get_url[1] + "/" + get_url[2]
    if "_" in get_url:
        url = get_url.split("_") # 分割
        get_url = url[0]
    uid=models.dngroute.objects.filter(url_str__contains=get_url).first()
    dngroute_uid = uid.uid_int
    # -----
    #   日记记录与 COOKIE 验证与权限 》》》开始
    # -----
    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径

    if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

        if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
            if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
                urlstr = parse.quote('不允许同时登录账号')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response
    else:
        urlstr = parse.quote('您需要重新登录')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response

```

```

# -----
# 日记记录与 COOKIE 验证与权限 《 《 《 结束
# -----
# -----
# 判断页面权限开始》 》 》 开始
# -----

group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称
dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: #判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added = False #增
delete = False #删
update = False #改
see = False #查

if '|' + str(dngroute_uid) + '|' in group.added_text: # 判断增加权限
    added = True
if '|' + str(dngroute_uid) + '|' in group.delete_text: # 判断删除权限
    delete = True
if '|' + str(dngroute_uid) + '|' in group.update_text: # 判断修改权限
    update = True
if '|' + str(dngroute_uid) + '|' in group.see_text: # 判断查看权限
    see = True

zd_list = dngadmin_common.dng_ziduan("dnguser") # 获取对应表下所有字段值

post4 = request.POST.get(zd_list[0][4], '')
post5 = request.POST.get(zd_list[0][5], '')
post6 = request.POST.get(zd_list[0][6], '')
post7 = request.POST.get(zd_list[0][7], '')
post8 = request.POST.get(zd_list[0][8], '')

if update:
    if not post4:
        urlstr = parse.quote('昵称不能为空')

```

```

        response = HttpResponseRedirect('/dngadmin/userdata/?jinggao=' + urlstr)
        return response
    if not post5:
        urlstr = parse.quote('性别不能为空')
        response = HttpResponseRedirect('/dngadmin/userdata/?jinggao=' + urlstr)
        return response
    if not post6:
        urlstr = parse.quote('个人简介不能为空')
        response = HttpResponseRedirect('/dngadmin/userdata/?jinggao=' + urlstr)
        return response
    if not post7:
        urlstr = parse.quote('我的邮箱不能为空')
        response = HttpResponseRedirect('/dngadmin/userdata/?jinggao=' + urlstr)
        return response
    if not post8:
        urlstr = parse.quote('手机号码不能为空')
        response = HttpResponseRedirect('/dngadmin/userdata/?jinggao=' + urlstr)
        return response

```

```

models.dnguser.objects.filter(uid_int=dnguser_uid).update(name_str=post4, gender_str=post5,
introduce_str=post6,

```

```

email_str=post7, mobile_str=post8,) # 更新修

```

改数据库

```

        urlstr = parse.quote('提交修改成功')
        response = HttpResponseRedirect('/dngadmin/userdata/?yes=' + urlstr)
        return response
    else:
        urlstr = parse.quote('您没有修改权限')
        response = HttpResponseRedirect('/dngadmin/userdata/?jinggao=' + urlstr)
        return response

```

Create your views here.

```

from django.shortcuts import render #视图渲染模块
from django.http import HttpResponseRedirect #请求模块
from . import models #数据库操作模块
from django.db.models import Q #数据库逻辑模块
from django.db.models import Avg, Max, Min, Sum #数据库聚合计算模块
from datetime import datetime, timedelta #Cookie 模块
from django.http import HttpResponseRedirect #重定向模块
from django.shortcuts import render
import os
import sys
from urllib import parse#转码
import re #正则模块
import random#随机模块

```



```

import hashlib# 加密模块
from django.utils import timezone #时间处理模块
import datetime#时间
import time# 日期模块
from . import dngadmin_common #公共函数
def crud(request):
    # -----
    # 通过路径获得栏目 ID 》》》 开始
    # -----
    get_url = request.path
    get_url = get_url.split("/") # 分割
    get_url = get_url[1] + "/" + get_url[2]
    if "_" in get_url:
        url = get_url.split("_") # 分割
        get_url = url[0]
    uid=models.dngroute.objects.filter(url_str__contains=get_url).first()
    dngroute_uid = uid.uid_int
    # -----
    # 日记记录与 COOKIE 验证与权限 》》》 开始
    # -----
    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    yuming_url = request.META.get('HTTP_HOST') # 当前访问的域名
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径
    tishi = request.GET.get('tishi') #提示
    jinggao = request.GET.get('jinggao') # 警告
    yes = request.GET.get('yes') # 警告

    if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

        if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
            if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
                urlstr = parse.quote('不允许同时登录账号')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response
            else:
                urlstr = parse.quote('您需要重新登录')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response

```

```

# -----
#  日记记录与 COOKIE 验证与权限 《 《 《 结束
# -----
# -----
#  判断页面权限开始》 》 》 开始
# -----

group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称

dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: #判断 URL 统一
    urlstr = parse.quote(' 您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
    urlstr = parse.quote(' 您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added =False #增
delete = False #删
update =False #改
see =False #查

if '|' + str(dngroute_uid) + '|' in group.added_text: # 判断增加权限
    added =True
if '|' + str(dngroute_uid) + '|' in group.delete_text: # 判断删除权限
    delete =True
if '|' + str(dngroute_uid) + '|' in group.update_text: # 判断修改权限
    update =True
if '|' + str(dngroute_uid) + '|' in group.see_text: # 判断查看权限
    see =True

# -----
#  判断页面权限开始 《 《 《 结束
# -----

#zd_list = dngadmin_common.dng_ziduan("crud") #获取对应表下所有字段值

#biaodan = dngadmin_common.dng_dnguser(uid=dnguser_uid)

return render(request, "dngadmin/crud.html", {
    "title":dngroute.name_str,
    "edition": dngadmin_common.dng_setup().edition_str, # 版本号

```

```

        "file": dngadmin_common.dng_setup().file_str, # 备案号
        "tongue": dngadmin_common.dng_setup().statistics_text, # 统计
        "added": added, #增
        "delete": delete, #删
        "update": update, #改
        "see": see, #查
        "tishi": tishi,
        "jinggao": jinggao,
        "yes": yes,
        "yuming_url": yuming_url,
    })
def crud_post(request):
    # -----
    # 通过路径获得栏目 ID 》》》开始
    # -----
    get_url = request.path
    get_url = get_url.split("/") # 分割
    get_url = get_url[1] + "/" + get_url[2]
    if "_" in get_url:
        url = get_url.split("_") # 分割
        get_url = url[0]
    uid=models.dngroute.objects.filter(url_str__contains=get_url).first()
    dngroute_uid = uid.uid_int
    # -----
    # 日记记录与 COOKIE 验证与权限 》》》开始
    # -----
    ip = request.META.get('HTTP_X_FORWARDED_FOR') # 获取 ip 信息
    liulanqi = request.META.get('HTTP_USER_AGENT') # 获取浏览器信息
    geturl = request.META.get('QUERY_STRING') # 获取域名后缀的 URL
    mulu_url = request.path # 获取不包含? 号之前的映射路径

    if "dnguser_uid" in request.COOKIES: # 判断 cookies 有无, 跳转
        dnguser_uid = request.get_signed_cookie(key="dnguser_uid", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_name = request.get_signed_cookie(key="dnguser_name", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)
        dnguser_cookie = request.get_signed_cookie(key="dnguser_cookie", default=None,
                                                salt=dngadmin_common.dng_anquan().salt_str,
max_age=None)

        if dngadmin_common.dng_anquan().tongshi_bool == False: # 验证是否同时登录
            if dngadmin_common.dng_tongshi(uid=dnguser_uid, cookie=dnguser_cookie) == False:
                urlstr = parse.quote('不允许同时登录账号')
                response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
                return response
            else:

```

```

        urlstr = parse.quote('您需要重新登录')
        response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
        return response

# -----
#   日记记录与 COOKIE 验证与权限 《 《 《 结束
# -----
# -----
#   判断页面权限开始》 》 》 开始
# -----

group = dngadmin_common.dng_usergroup(gid=dngadmin_common.dng_dnguser(dnguser_uid).group_int) # 获取会
员组名称
dngroute = models.dngroute.objects.filter(uid_int=dngroute_uid).first()#查询路径取回本页面菜单信息
dngadmin_common.dng_dngred(uid=dnguser_uid, title=dngroute.name_str, url=mulu_url, user=liulanqi, ip=ip)
# 日记记录函数

if not dngroute.url_str in mulu_url: #判断 URL 统一
    urlstr = parse.quote('您的访问与菜单映射不匹配')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

elif not '|' + str(dngroute_uid) + '|' in group.menu_text: #判断菜单权限
    urlstr = parse.quote('您没有访问这个菜单的权限')
    response = HttpResponseRedirect('/dngadmin/tips/?jinggao=' + urlstr)
    return response

added =False #增
delete = False #删
update =False #改
see =False #查

if '|' + str(dngroute_uid) + '|' in group.added_text: # 判断增加权限
    added =True
if '|' + str(dngroute_uid) + '|' in group.delete_text: # 判断删除权限
    delete =True
if '|' + str(dngroute_uid) + '|' in group.update_text: # 判断修改权限
    update =True
if '|' + str(dngroute_uid) + '|' in group.see_text: # 判断查看权限
    see =True

#zd_list = dngadmin_common.dng_ziduan("crud") # 获取对应表下所有字段值
biaodan = dngadmin_common.dng_dnguser(uid=dnguser_uid)

post0 = request.POST.get('list_common', '')
post1 = request.POST.get('list_py', '')
post2 = request.POST.get('list_html', '')
post3 = request.POST.get('post_py', '')
post4 = request.POST.get('post_html', '')
post5 = request.POST.get('list_url', '')

```

```

post5_zhushi = request.POST.get('list_url_zhushi', '')
post6 = request.POST.get('post_url', '')
post6_zhushi = request.POST.get('post_url_zhushi', '')

if update:
    if post0:
        root = os.getcwd() # 获取项目运行根目录
        root_py = os.path.abspath('app/dngadmin_common.py') # 跟目录+py 文件所在路径

        seo = open(root_py, "rb") # 读取
        key = str(seo.read(), 'utf-8') # read 是读取命令 str 转换字符串和编码

        strinfo = re.compile('# ① 数据库模型替换 1①') # 正则替换前
        tihuan_post0 = '''if name == ''' + post0 + ''':
dbziduan = models.''' + post0 + '''. _meta.fields
# ① 数据库模型替换 1①'''
        keyseo = strinfo.sub(tihuan_post0, key) # 正则替换后

        strinfo = re.compile('# ② 数据库模型替换 2②') # 正则替换前
        tihuan_post0 = '''if name == ''' + post0 + ''':
params = models.''' + post0 + '''. _meta.get_field(key).verbose_name
# ② 数据库模型替换 2②'''
        keyseo = strinfo.sub(tihuan_post0, keyseo) # 正则替换后

        root_py2 = os.path.abspath('app/dngadmin_common.py') # 跟目录+py 文件所在路径
        kaobei2 = open(root_py2, "w") # 写入
        kaobei2.write(keyseo) # write 是写入命令
        kaobei2.close() # close()是关闭命令

        urlstr = parse.quote('参数' + post0 + '写入 dngadmin_common.py 成功')
        response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
        return response

```

```

if post1:

```

```

####构造字段数组###

```

```

ida = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27,
28, 29, 30]

```

```

db_list = dngadmin_common.dng_ziduan(post1) # 获取对应表下所有字段值

```

```

####开始构造①①①JSON 函数-查①①①文本代码###

```

```

json_list = zip(db_list[0], db_list[1], ida) # [0]=例子: uid_int (字段值) [1]=例子: 会员 id

```

```

(字段备注) [2]=例子(数字列)
    json = []
    for json_arr in json_list: # 遍历字符串, 遍历输出

        json.append(''
''' + str(json_arr[0]) + '''" : str(key. ''' + str(json_arr[0]) + '''), # ''' + str(json_arr[1])
+ '''

''' ) # 循环写入空数组

    json = ''.join(json)
    json = json.replace('_time',',', '_time.strftime("%Y-%m-%d %H:%M:%S"),')
    #####开始构造○○○added 函数-增○○○文本代码###

    added_list = zip(db_list[0], db_list[1], ida) # [0]=例子: uid_int (字段值) [1]=例子: 会员
id (字段备注) [2]=例子(数字列)
    added_a = []
    added_b = []
    for added_arr in added_list: # 遍历字符串, 遍历输出

        if str(added_arr[0]) == "id":
            continue # 跳过本次循环

        added_a.append(''
post''' + str(added_arr[2]) + ''' = request.POST.get(zd_list[0][''' + str(added_arr[2]) + '''], '') #''' +
str(added_arr[1]) + '''

''' )# 循环写入空数组

        added_b.append(str(added_arr[0]) + "=post"+str(added_arr[2])+", ") # 循环写入空数组

    added_a = ''.join(added_a)
    added_b = ''.join(added_b)
    added_b = "ok = models.映射的路径名替换.objects.create("+added_b+")" # 更新修改数据库"

    #####开始构造○○○update 函数-改○○○文本代码###

    update_list = zip(db_list[0], db_list[1], ida) # [0]=例子: uid_int (字段值) [1]=例子: 会员
id (字段备注) [2]=例子(数字列)
    update_a = []
    update_b = []
    for update_arr in update_list: # 遍历字符串, 遍历输出

        if str(update_arr[0]) == "id":
            continue # 跳过本次循环

        update_a.append(''
post''' + str(update_arr[2]) + ''' = request.POST.get(zd_list[0][''' + str(update_arr[2]) + '''], '') #'''
+ str(update_arr[1]) + '''

''' ) # 循环写入空数组

```

```

        update_b.append(''
if post''' + str(update_arr[2]) + ''':
    models.映射的路径名替换.objects.filter(id=post0).update('' + str(update_arr[0]) + '''=post'''
+ str(update_arr[2]) + ''')
''' # 循环写入空数组

update_a = ''.join(update_a)
update_b = ''.join(update_b)

####开始构造○○○search函数-搜○○○文本代码###
search_list = zip(db_list[0], db_list[1], ida) # [0]=例子: uid_int (字段值) [1]=例子: 会员
id (字段备注) [2]=例子 (数字列)
search_a = []
search_b = []
search_c = []
for search_arr in search_list: # 遍历字符串, 遍历输出

    if str(search_arr[0]) == "id":
        continue # 跳过本次循环

    search_a.append(''
post''' + str(search_arr[2]) + '' = request.POST.get(zd_list[0]['' + str(search_arr[2]) + '''], '') #'''
+ str(search_arr[1]) + ''
''' # 循环写入空数组
    if str(search_arr[2]) == "1":
        if_bq="if"
    else:
        if_bq = "elif"
    search_b.append(''
''' + if_bq + ''' post''' + str(search_arr[2]) + ''':
    dngred = models.映射的路径名替换.objects.filter('' + search_arr[0] + '''_contains=post''' +
str(search_arr[2]) + ''').order_by('-id') #''' + str(search_arr[1]) + ''
''' # 循环写入空数组
    search_c.append(''
''' + search_arr[0] + ''': str(key.''' + search_arr[0] + '''), #''' + search_arr[1])
# 循环写入空数组

search_a = ''.join(search_a)
search_b = ''.join(search_b)
search_c = ''.join(search_c)
####替换代码开始###

root_py = os.path.abspath('app/dngadmin_demo_list.py') # 跟目录+py 文件所在路径

seo = open(root_py, "rb") # 读取
key = str(seo.read(), 'utf-8') # read 是读取命令 str 转换字符串和编码

```

```

####json 查看替换
strinfo = re.compile('# ⊙json 查看替换⊙ #') # 正则替换前
keyseo1 = strinfo.sub(json, key) # 正则替换后
####added 新增替换
strinfo = re.compile('# 1⊙added 新增替换⊙1 #') # 正则替换前
keyseo2 = strinfo.sub(added_a, keyseo1) # 正则替换后

strinfo = re.compile('# 2⊙added 新增替换⊙2 #') # 正则替换前
keyseo3 = strinfo.sub(added_b, keyseo2) # 正则替换后
####update 更改替换
strinfo = re.compile('# 1⊙update 更改替换⊙1 #') # 正则替换前
keyseo4 = strinfo.sub(update_a, keyseo3) # 正则替换后

strinfo = re.compile('# 2⊙update 更改替换⊙2 #') # 正则替换前
keyseo5 = strinfo.sub(update_b, keyseo4) # 正则替换后
####search 搜索替换
strinfo = re.compile('# 1⊙search 搜索替换⊙1 #') # 正则替换前
keyseo6 = strinfo.sub(search_a, keyseo5) # 正则替换后

strinfo = re.compile('# 2⊙search 搜索替换⊙2 #') # 正则替换前
keyseo7 = strinfo.sub(search_b, keyseo6) # 正则替换后

strinfo = re.compile('# 3⊙search 搜索替换⊙3 #') # 正则替换前
keyseo8 = strinfo.sub(search_c, keyseo7) # 正则替换后
####模型替换
strinfo = re.compile('# 映射的路径名替换') # 正则替换前
keyseo9 = strinfo.sub(post1, keyseo8) # 正则替换后

root_py2 = os.path.abspath('app/dngadmin_' + post1 + '.py') # 跟目录+py 文件所在路径
kaobei2 = open(root_py2, "w") # 写入
kaobei2.write(keyseo9) # write 是写入命令
kaobei2.close() # close()是关闭命令

#return HttpResponse(search_c)
urlstr = parse.quote('列表PY' + post1 + '构造成功')
response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
return response

```

```

if post2:

```

```

####构造字段数组###

```

```

ida = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,

```

```

25, 26, 27,

```

```

28, 29, 30]

```

```

db_list = dngadmin_common.dng_ziduan(post2) # 获取对应表下所有字段值

```

```

html_list = zip(db_list[0], db_list[1], ida) # [0]=例子: uid_int (字段值) [1]=例子: 会员 id
(字段备注) [2]=例子 (数字列)

```

```

####开始构造⊙⊙⊙JSON 函数-查⊙⊙⊙文本代码####

```



```

list_html_a = []
list_html_b = []
list_html_c = []
list_html_d = []
for html_arr in html_list: # 遍历字符串, 遍历输出

    if str(html_arr[0]) == "id":
        continue # 跳过本次循环

    list_html_a.append('' {% if zd_list.0.''' + str(html_arr[2]) + ''' %}

<div class="layui-form-item">
    <label class="layui-form-label">{{zd_list.1.''' + str(html_arr[2]) + '''}}</label>
    <div class="layui-input-block">
        <input type="text" name="{{zd_list.0.''' + str(html_arr[2]) + '''}}" id="{{zd_list.0.'''
+ str(html_arr[2]) + '''}}" required lay-verify="required" autocomplete="off" placeholder="请输入
{{zd_list.1.''' + str(html_arr[2]) + '''}}" class="layui-input">
    </div>
</div>
{% endif %}
''')

    if str(html_arr[2]) == "0":
        if_bq= "value=\"{{dnguser_uid}}\""
    else:
        if_bq = ""
    list_html_b.append('' {% if zd_list.0.''' + str(html_arr[2]) + ''' %}

<div class="layui-form-item">
        <label class="layui-form-label">{{zd_list.1.''' + str(html_arr[2]) +
'''}}</label>

        <div class="layui-input-block">
            <input type="text" name="{{zd_list.0.''' + str(html_arr[2]) + '''}}"
id="{{zd_list.0.''' + str(html_arr[2]) + '''}}" '''+if_bq+''' required lay-verify="required"
autocomplete="off" placeholder="请输入{{zd_list.1.''' + str(html_arr[2]) + '''}}" class="layui-input">
        </div>
    </div>

    {% endif %}
''')

list_html_a = ''.join(list_html_a)

list_html_b = ''.join(list_html_b)

root_py = os.path.abspath(' app/Templates/dngadmin/demo_list.html') # 跟目录+py 文件所在路径

```

路径

```

seo = open(root_py, "rb") # 读取
key = str(seo.read(), 'utf-8') # read 是读取命令 str 转换字符串和编码

strinfo = re.compile('<!-- 1⊙HTML 模板替换⊙1 -->') # 正则替换前
keyseo1 = strinfo.sub(list_html_a, key) # 正则替换后

strinfo = re.compile('<!-- 2⊙HTML 模板替换⊙2 -->') # 正则替换前
keyseo2 = strinfo.sub(list_html_b, keyseo1) # 正则替换后

root_py2 = os.path.abspath('app/Templates/dngadmin/' + post2 + '.html') # 跟目录+py 文件所在

kaobei2 = open(root_py2, "w") # 写入
kaobei2.write(keyseo2) # write 是写入命令
kaobei2.close() # close()是关闭命令

urlstr = parse.quote('列表模板' + post2 + '构造成功')
response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
return response

if post3:

    root = os.getcwd() # 获取项目运行根目录
    root_py = os.path.abspath('app/dngadmin_demo_post.py') # 跟目录+py 文件所在路径

    seo = open(root_py, "rb") # 读取
    key = str(seo.read(), 'utf-8') # read 是读取命令 str 转换字符串和编码

    strinfo = re.compile('映射的路径名替换') # 正则替换前
    keyseo = strinfo.sub(post3, key) # 正则替换后

    root_py2 = os.path.abspath('app/dngadmin_' + post3 + '.py') # 跟目录+py 文件所在路径
    kaobei2 = open(root_py2, "w") # 写入
    kaobei2.write(keyseo) # write 是写入命令
    kaobei2.close() # close()是关闭命令

    urlstr = parse.quote('表单 Py 文件' + post3 + '构造成功')
    response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
    return response

if post4:

    root = os.getcwd() # 获取项目运行根目录
    root_py = os.path.abspath('app/Templates/dngadmin/demo_post.html') # 跟目录+py 文件所在路径

    seo = open(root_py, "rb") # 读取
    key = str(seo.read(), 'utf-8') # read 是读取命令 str 转换字符串和编码

```

```

    strinfo = re.compile('映射的路径名替换') # 正则替换前
    keyseo = strinfo.sub(post4, key) # 正则替换后

    root_py2 = os.path.abspath('app/Templates/dngadmin/' + post4 + '.html') # 跟目录+py 文件所在路径
    kaobei2 = open(root_py2, "w") # 写入
    kaobei2.write(keyseo) # write 是写入命令
    kaobei2.close() # close()是关闭命令

    urlstr = parse.quote('模板' + post4 + '构造成功')
    response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
    return response
if post5 and post5_zhushi:

    root = os.getcwd() # 获取项目运行根目录
    root_py = os.path.abspath('app/dngadmin_urls.py') # 跟目录+py 文件所在路径

    seo = open(root_py, "rb") # 读取
    key = str(seo.read(), 'utf-8') # read 是读取命令 str 转换字符串和编码

    strinfo = re.compile('# ①映射替换 1') # 正则替换前
    tihuan_post5 = '''from . import dngadmin_''' + post5 + ''' #''' + post5_zhushi + '''
# ①映射替换 1'''
    keyseo = strinfo.sub(tihuan_post5, key) # 正则替换后

    strinfo = re.compile('# ②映射替换 2') # 正则替换前
    tihuan_post5 = '''url(r'~' + post5 + '/' , dngadmin_''' + post5 + '''.''' + post5 + '''), #
''' + post5_zhushi + '''
    url(r'~' + post5 + '''_json/' , dngadmin_''' + post5 + '''.''' + post5 + '''_json), #''' + post5_zhushi
+ '''_json
    url(r'~' + post5 + '''_added/' , dngadmin_''' + post5 + '''.''' + post5 + '''_added), #''' + post5_zhushi
+ '''新增
    url(r'~' + post5 + '''_delete/' , dngadmin_''' + post5 + '''.''' + post5 + '''_delete), #''' + post5_zhushi
+ '''删除
    url(r'~' + post5 + '''_update/' , dngadmin_''' + post5 + '''.''' + post5 + '''_update), #''' + post5_zhushi
+ '''更新修改
    url(r'~' + post5 + '''_search/' , dngadmin_''' + post5 + '''.''' + post5 + '''_search), #''' + post5_zhushi
+ '''搜索
# ②映射替换 2'''
    keyseo = strinfo.sub(tihuan_post5, keyseo) # 正则替换后

    root_py2 = os.path.abspath('app/dngadmin_urls.py') # 跟目录+py 文件所在路径
    kaobei2 = open(root_py2, "w") # 写入
    kaobei2.write(keyseo) # write 是写入命令
    kaobei2.close() # close()是关闭命令

    urlstr = parse.quote('参数' + post5 + '表单映射写入成功')
    response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
    return response

```

```
urlstr = parse.quote('参数' + post5 + '列表映射写入成功')
response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
return response

if post6 and post6_zhushi:

    root = os.getcwd() # 获取项目运行根目录
    root_py = os.path.abspath('app/dngadmin_urls.py') # 跟目录+py 文件所在路径

    seo = open(root_py, "rb") # 读取
    key = str(seo.read(), 'utf-8') # read 是读取命令 str 转换字符串和编码

    strinfo = re.compile('# ①映射替换 1①') # 正则替换前
    tihuan_post6 = '''from . import dngadmin_''' + post6 + '''    #''' + post6_zhushi + '''
# ①映射替换 1①'''
    keyseo = strinfo.sub(tihuan_post6, key) # 正则替换后

    strinfo = re.compile('# ②映射替换 2②') # 正则替换前
    tihuan_post6 = '''url(r'~''' + post6 + '''/', dngadmin_''' + post6 + '''.''' + post6 + '''), #
''' + post6_zhushi + '''
url(r'~''' + post6 + '''_post/', dngadmin_''' + post6 + '''.''' + post6 + '''_post), # ''' + post6_zhushi
+ '''的 post
# ②映射替换 2②'''
    keyseo = strinfo.sub(tihuan_post6, keyseo) # 正则替换后

    root_py2 = os.path.abspath('app/dngadmin_urls.py') # 跟目录+py 文件所在路径
    kaobei2 = open(root_py2, "w") # 写入
    kaobei2.write(keyseo) # write 是写入命令
    kaobei2.close() # close()是关闭命令

    urlstr = parse.quote('参数' + post6 + '表单映射写入成功')
    response = HttpResponseRedirect('/dngadmin/crud/?yes=' + urlstr)
    return response

else:
    urlstr = parse.quote('您没有修改权限')
    response = HttpResponseRedirect('/dngadmin/crud/?jinggao=' + urlstr)
    return response
```