# 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,

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("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

# ----------------------------------------------------------

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

# ----------------------------------------------------------

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

pwname =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 pwname:

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 HttpResponse(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.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

# ----------------------------------------------------------

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 HttpResponse(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

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='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\_longin #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1⊙

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,

emall\_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 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 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1⊙') # 正则替换前

tihuan\_post0 = '''if name == "'''+post0+'''":

dbziduan = models.'''+post0+'''.\_meta.fields

# ⊙1数据库模型替换1⊙'''

keyseo = strinfo.sub(tihuan\_post0, key) # 正则替换后

strinfo = re.compile('# ⊙2数据库模型替换2⊙') # 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2⊙'''

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⊙⊙⊙JSON函数-查⊙⊙⊙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

###josn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

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文件所在路径

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1⊙') # 正则替换前

tihuan\_post6 = '''from . import dngadmin\_''' + post6 + ''' #''' + post6\_zhushi + '''

# ⊙1映射替换1⊙'''

keyseo = strinfo.sub(tihuan\_post6, key) # 正则替换后

strinfo = re.compile('# ⊙2映射替换2⊙') # 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2⊙'''

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