# xadmin

## xadmin的全局配置

将全局配置修改:

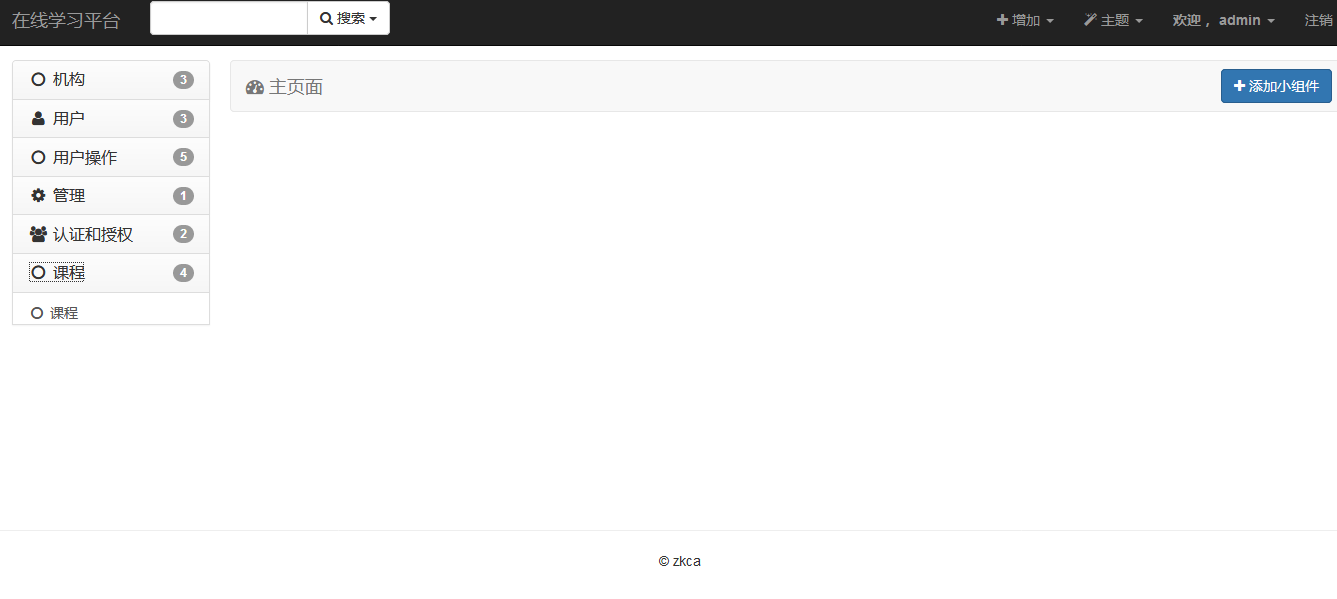
* 如左上角：django Xadmin。下面的我的公司
* 主题修改，app名称汉化，菜单收叠。

**使用Xadmin的主题功能。**

把全站的配置放在users\adminx.py中:

### 1.1添加主题功能

1. *# 创建xadmin的最基本管理器配置，并与view绑定*class BaseSetting(object):  
    *# 开启主题功能* enable\_themes = True  
    use\_bootswatch = True  
     
     
   *# 全局修改，固定写法*class GlobalSetting(object):  
    *# 修改title* site\_title = "在线学习平台"  
    *# 修改footer* site\_footer = 'zkca'  
    *# 收起菜单* menu\_style = 'accordion'  
     
     
   xadmin.site.register(EmailVerifyRecord, EmailVerifyRecordAdmin)  
   xadmin.site.register(Banner, BannerAdmin)  
     
   *# 将基本配置管理与view绑定*xadmin.site.register(views.BaseAdminView, BaseSetting)  
   *# 将title和footer信息进行注册*xadmin.site.register(views.CommAdminView, GlobalSetting)



## 修改app的名字

在apps.py里面配置app的显示名称:

from django.apps import AppConfig  
  
  
class CourseConfig(AppConfig):  
 name = 'course'  
 verbose\_name = '课程'

还要在users/\_\_init\_\_.py中引用apps.py的配置

添加代码如下：

default\_app\_config = 'course.apps.CourseConfig'

# Django部分

## 增加邮箱登录

让用户可以通过邮箱或者用户名都可以登录，用自定义authenticate方法这里是继承ModelBackend类来做的验证

#settings.py

AUTH\_USER\_MODEL = 'users.UserProfile'

#views.py

*# 邮箱和用户名都可以登录*class CustomBackend(ModelBackend):  
 def authenticate(self, request, username=None, password=None, \*\*kwargs):  
 try:  
 *# 不希望用户存在两个，get只能有一个。两个是get失败的一种原因 Q为使用并集查询* user = UserProfile.objects.get(Q(username=username) | Q(email=username))  
 *# django的后台中密码加密：所以不能password==password  
 # UserProfile继承的AbstractUser中有def check\_password(self, raw\_password):* if user.check\_password(password):  
 return user  
 except Exception as e:  
 return None

## 用form实现登录

1. 把前面views中的user\_login()函数改成基于类的形式, 基于类的urls配置
2. users下新建form.py文件

from django import forms  
  
  
*# 登录表单验证*class LoginForm(forms.Form):  
 *# 用户名密码不能为空* username = forms.CharField(required=True)  
 password = forms.CharField(required=True, min\_length=5)

1. 定义好forms后利用它来做验证，并完善错误提示信息

#views.py

*# 利用表单进行验证*def post(self, request):  
 *# 实例化* login\_form = LoginForm(request.POST)  
 if login\_form.is\_valid():  
 *# 获取提交的用户名密码* user\_name = request.POST.get('username', None)  
 pass\_word = request.POST.get('password', None)  
 *# 成功返回user对象,失败None* user = authenticate(username=user\_name, password=pass\_word)  
 *# 如果不是Null，说明验证成功* if user is not None:  
 *# login* login(request, user)  
 return render(request, 'index.html')  
 *# 只有当验证不通过才返回数据到前端* else:  
 return render(request, 'login.html', {'msg': '用户名或密码错误！', 'login\_form': login\_form})  
 *# form.is\_valid()已经判断不合法了，所以这里不需要返回错误信息到前端* else:  
 return render(request, 'login.html', {'login\_form': login\_form})

## 3 . URL配置和松耦合

1. 传统模式

提交URL地址固定耦合在action里

<form action=**"/**register**/"** method=**"post"** autocomplete=**"off"**>

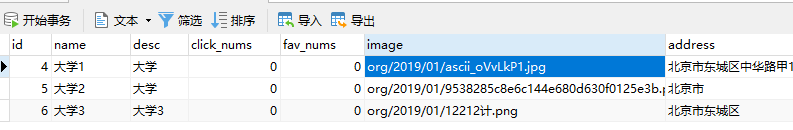
1. 优化，松耦合

## 4. 图片上传

**4.1** 配置settings

**如何将image Field转换成图片地址?**

数据库中image以字符串格式保存的，是相对路径，直接取是取不出来的，必须补全路径



拼接完整绝对路径：data-url="{{ MEDIA\_URL }}{{ course\_org.image }}"

要向使用{{ MEDIA\_URL }}，要先在settings中TEMPLATES 里面添加media处理器：

'django.template.context\_processors.media'

STATIC\_URL = '/static/'  
STATICFILES\_DIRS = [  
 os.path.join(BASE\_DIR, 'static'),  
]  
*# STATIC\_ROOT = "/var/py/xf/static/"  
# 设置上传文件的路径*MEDIA\_URL = '/media/'  
MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

**配置url：**

 urls.py

from django.views.static import serve

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

**4.2** 配置model

class CourseOrg(models.Model):  
 *"""  
 课程机构基本信息  
 """* ORG\_CHOICES = (  
 ('pxjg', u'培训机构'),  
 ('gx', u'高校'),  
 ('gr', u'个人'),  
 )  
 name = models.CharField('机构名称', max\_length=50)  
 desc = models.TextField('机构描述')  
 category = models.CharField(max\_length=20, choices=ORG\_CHOICES, default='pxjg',verbose\_name='所属分类')  
 click\_nums = models.IntegerField('点击', default=0)  
 fav\_nums = models.IntegerField('收藏数', default=0)  
 image = models.ImageField('封面图', upload\_to='org/%Y/%m', max\_length=100)  
 address = models.CharField('结构地址', max\_length=150)  
 city = models.ForeignKey(CityDict, verbose\_name='所在城市', on\_delete=models.CASCADE)  
 add\_time = models.DateTimeField(default=datetime.now)

**4.3** 效果

图片按照：**“/media/org/年/月份/图片名字” 分割， web可以查看图片**

