1. 分析页面抽象app

用户users 课程courses 机构 orgs 用户操作operations

1. App设计

1、不能循环引用 2、设计上层app去解耦

1. 分析页面抽象模型
2. 用户：

1：用户信息表

2：邮箱验证码

3：轮播图信息表

1. 课程：

1：课程信息

2：章节信息

3：视频信息

4：资源信息

1. 机构：

1：机构信息

2：城市信息

3：讲师信息

1. 操作：
2. 用户咨询表
3. 用户收藏表
4. 用户课程表
5. 用户消息表
6. 用户评论表
7. 模型设计

一个一个app去设计模型和字段，设计的字段去根据页面设计

1. Xadmin的配置和使用
2. 将下载好的xadmin解压，复制里面的xadmin文件夹到我们的项目根目录当中
3. 创建extra\_apps放置第三方的app，将xadmin移动到我们这个extra\_apps下
4. 将extra\_apps mark为root\_source
5. 将extra\_apps在setting当中配置好搜索路径
6. 打开cmd，进入虚拟环境，安装xadmin依赖包，依赖包的版本在xadmin文件夹下：requirements.txt，在其中有一个包版本改成2.1 django-formtools==2.1，否则版本太低，拉不起来
7. Pip install -r requirements.txt -i <https://pypi.douban.com/simple/>
8. 依赖包装完之后，再去install当中添加上app

8、将xadmin和crispy\_forms添加到我们的installed\_apps

9、将我们本来的admin注释掉改为我们xadmin

url(r'^xadmin/', xadmin.site.urls),

10、再次执行迁移同步，目的是为了生成xadmin所依赖的表

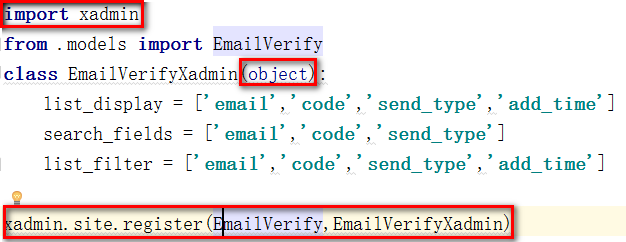
11、创建超级管理员，去验证xadmin是否安装成功

1. 将测试数据添加在xadmin当中

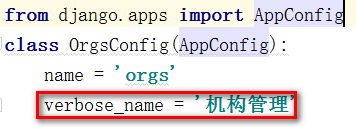
1、在相应的app当中创建adminx.py文件

2、在adminx.py文件当中注册我们的每一个model

3、注册的时候和admin里面类似



1. 将xadmin后台app的名字改为我们自定的名字
2. 在app当中apps.py文件当中，最下面添加verbose\_name=’自定义名字’



1. 在app当中的\_\_init\_\_文件当中添加配置

default\_app\_config = "orgs.apps.OrgsConfig"

可以在installed\_apps中安装应用的时候，按照users自动安装的方式去做安装，那么这个\_\_init\_\_文件中的配置也可以省略

1. Xadmin基础设置和icon设置

from xadmin import views

class BaseXadminSetting(object):

enable\_themes = True

use\_bootswatch = True

class CommXadminSetting(object):

site\_title = '尚在线后台管理系统'

site\_footer = '尚硅谷IT教育'

menu\_style = 'accordion'

*#注册xadmin的主题*xadmin.site.register(BaseAdminView,BaseXadminSetting)  
*#注册xadmin的标题和底部公司名称*xadmin.site.register(CommAdminView,CommXadminSetting)

1. 完成主页展示逻辑

主要是多级继承思路要考虑好

1. 完成登录注册退出逻辑

验证码的使用：

* 1. 、安装django-simple-captcha pip install
  2. 、将captcha 安装在installed\_apps里面
  3. 、将captcha配置url

url(**r'^captcha/'**,include(**'captcha.urls'**)),

* 1. 、迁移同步，生成captcha所依赖的表
  2. 、将captcha字段在form类当中进行设置

**from** captcha**.**fields **import** CaptchaField

**class** **UserRegisterForm(**forms**.**Form**):**

email **=** forms**.**EmailField**(**required**=True,**min\_length**=**10**,**error\_messages**={**

'required'**:**'邮箱必须填写'**,**

'min\_length'**:**'邮箱最小长度10'

**})**

password **=** forms**.**CharField**(**required**=True,**min\_length**=**3**,**error\_messages**={**

'required'**:** '密码必须填写'**,**

'min\_length'**:** '密码最小长度3'

**})**

captcha **=** CaptchaField**()**

* 1. 、在后台逻辑当中，get请求里面实例化我们的form,将form对象返回到页面

**def** user\_register**(**request**):**

**if** request**.**method **==** 'GET'**:**

user\_register\_form **=** UserRegisterForm**()**

**return** render**(**request**,**'register.html'**,{**

'user\_register\_form'**:**user\_register\_form

**})**

* 1. 、在页面上通过{{ form.captcha}} 获取验证码

1. 为登录添加激活后登录功能

EMAIL\_HOST = 'smtp.163.com'

EMAIL\_PORT = 25

EMAIL\_HOST\_USER = 'ly03171629@163.com'

EMAIL\_HOST\_PASSWORD = 'liuyuan1987729'

EMAIL\_FROM = 'ly03171629@163.com'

创建is\_start字段，用来控制激活，激活后才能登陆

1. 为注册添加激活登录功能

主要是使用邮箱发送激活验证码

1. 忘记密码和重置密码功能

通过发送邮箱验证码实现密码重置

1. 机构列表页和详情页

我要学习和收藏机构功能

1. 课程列表页和详情页

收藏课程和课程机构

1. 讲师列表页和详情页

收藏讲师和讲师机构

1. 用户中心的逻辑功能

修改用户信息、修改头像和修改邮箱

1. 首页展示、全局搜索和Xadmin配置富文本编辑器，使用全局404和500页面的配置、把函数view改为类、

全局搜索使用自己的功能

全局配置的时候注意分页和关键词的关联

富文本编辑器配置完成后，路径和页面标签要注意

1. 项目上线展示

Niginx和uwsgi