# 项目准备阶段

在做项目之前先把所有的前端页面打开通览一遍，目的有三：

1）：从页面上进行模块划分抽象出app

2）：从页面上分析各个App初步抽象出app模型和模型字段

3）：从页面上分析整个项目模板的继承关系

## 项目模块划分（app抽象）

### 1.1、app划分

用户users

课程courses

机构 orgs

用户操作operations

### 1.2、为什么要有operations的app

|  |
| --- |
|  |

如果没有最后一个用户操作的app，那么用户和课程之间的业务逻辑，就会存在如上图循环引用模型（模块）的问题，因此我们需要创建第四个用户操作的APP来处理。如下图：

|  |
| --- |
|  |

## 项目各个App模型抽象

用户：

1：用户信息表

2：邮箱验证码

3：轮播图信息表

课程：

1：课程信息

2：章节信息

3：视频信息

4：资源信息

机构：

1：机构信息

2：城市信息

3：讲师信息

用户操作：

1. 用户咨询表
2. 用户收藏表
3. 用户课程表
4. 用户消息表
5. 用户评论表

## 项目的建立

### 3.1、建立项目

建立项目的同时，建立第一个app用户为users,同时设置虚拟环境

### 3.2、建立其余的app并且建立apps目录管理所有的app

1）建立其余的app分别为：

机构orgs 课程courses 用户操作：operations

2）在项目根目录下建立包apps并且在settings当中配置apps的搜索路径，将所有创建的app拉入apps包里进行管理

|  |
| --- |
| **import** os**,**sys  # Build paths inside the project like this: os.path.join(BASE\_DIR, ...)  BASE\_DIR **=** os**.**path**.**dirname**(**os**.**path**.**dirname**(**os**.**path**.**abspath**(**\_\_file\_\_**)))**  sys**.**path**.**insert**(**0**,**os**.**path**.**join**(**BASE\_DIR**,**'apps'**))**  sys**.**path**.**insert**(**0**,**os**.**path**.**join**(**BASE\_DIR**,**'extra\_apps'**))** |

### 3.3、设置settings

设置installed\_apps

|  |
| --- |
| INSTALLED\_APPS **=** **[**  'django.contrib.admin'**,**  'django.contrib.auth'**,**  'django.contrib.contenttypes'**,**  'django.contrib.sessions'**,**  'django.contrib.messages'**,**  'django.contrib.staticfiles'**,**  'users.apps.UsersConfig'**,**  'orgs.apps.OrgsConfig'**,**  'courses.apps.CoursesConfig'**,**  'operations.apps.OperationsConfig'**,**  **]** |

设置数据库为mysql

|  |
| --- |
| DATABASES **=** **{**  'default'**:** **{**  'ENGINE'**:** 'django.db.backends.mysql'**,**  'NAME'**:** 'guliedu'**,**  'USER'**:** 'root'**,**  'PASSWORD'**:** 'root'**,**  'HOST'**:** 'localhost'**,**  'PORT'**:** '3306'**,**  **}**  **}** |

设置国际化语言和时区

|  |
| --- |
| LANGUAGE\_CODE **=** 'zh-hans'  TIME\_ZONE **=** 'Asia/Shanghai'  USE\_I18N **=** **True**  USE\_L10N **=** **True**  USE\_TZ **=** **False** |

设置静态文件访问

|  |
| --- |
| STATIC\_URL **=** '/static/'  STATICFILES\_DIRS **=** **[**  os**.**path**.**join**(**BASE\_DIR**,**'static'**)**  **]** |

设置媒体文件访问路径

|  |
| --- |
| MEDIA\_URL **=** '/static/media/'  MEDIA\_ROOT **=** os**.**path**.**join**(**BASE\_DIR**,**'static/media'**)** |

### 3.4、设置主子路由urls

|  |
| --- |
| **from** django**.**conf**.**urls **import** url**,**include  **from** django**.**contrib **import** admin  urlpatterns **=** **[**  url(r'^admin/', admin.site.urls),  url**(**r'^users/'**,**include**(**'users.urls'**,**namespace**=**'users'**)),**  url**(**r'^courses/'**,** include**(**'courses.urls'**,** namespace**=**'courses'**)),**  url**(**r'^orgs/'**,** include**(**'orgs.urls'**,** namespace**=**'orgs'**)),**  url**(**r'^operations/'**,** include**(**'operations.urls'**,** namespace**=**'operations'**)),**  **]** |

## 4、建立数据模型类

设置好settings和urls以后需要考虑到项目当中的数据，因此我们需要把每个App里面的模型类抽象并且把模型类创建出来，需要回过头再去查看页面，抽象每个模型类当中的数据模型字段

### 4.1、用户users里面models设计

|  |
| --- |
| **from** django**.**db **import** models  **from** django**.**contrib**.**auth**.**models **import** AbstractUser  **from** datetime **import** datetime  # Create your models here.  **class** **UserProfile(**AbstractUser**):**  image **=** models**.**ImageField**(**upload\_to**=**'users/%y/%m/%d'**,**max\_length**=**200**,**verbose\_name**=**"用户头像"**,**null**=True,**blank**=True)**  nick\_name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"用户昵称"**,**null**=True,**blank**=True)**  birthday **=** models**.**DateTimeField**(**null**=True,**blank**=True,**verbose\_name**=**"用户生日"**)**  gender **=** models**.**CharField**(**choices**=((**'男'**,**'男'**),(**'女'**,**'女'**)),**verbose\_name**=**"用户性别"**,**max\_length**=**6**,**default**=**'男'**)**  address **=** models**.**CharField**(**max\_length**=**300**,**verbose\_name**=**"用户地址"**,**null**=True,**blank**=True)**  phone **=** models**.**CharField**(**max\_length**=**11**,**verbose\_name**=**"用户手机"**)**  is\_start **=** models**.**BooleanField**(**default**=False,**verbose\_name**=**"是否激活"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**username    **class** **Meta:**  verbose\_name **=** '用户信息'  verbose\_name\_plural **=** verbose\_name  **class** **BannerInfo(**models**.**Model**):**  image **=** models**.**ImageField**(**upload\_to**=**'banners/%y/%m/%d'**,**max\_length**=**200**,**verbose\_name**=**"轮播图片"**)**  url **=** models**.**URLField**(**max\_length**=**200**,**verbose\_name**=**'轮播链接'**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** str**(**self**.**image**)**  **class** **Meta:**  verbose\_name **=** '轮播图信息'  verbose\_name\_plural **=** verbose\_name  **class** **EmailVerifyCode(**models**.**Model**):**  email **=** models**.**EmailField**(**max\_length**=**50**,**verbose\_name**=**"邮箱"**)**  code **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"验证码"**)**  send\_type **=** models**.**CharField**(**choices**=((**'register'**,**'注册激活'**),(**'forget'**,**'重置密码'**),(**'change'**,**'修改邮箱'**)),**max\_length**=**15**,**verbose\_name**=**"发送类别"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"发送时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**code  **class** **Meta:**  verbose\_name**=**'邮箱验证码'  verbose\_name\_plural **=** verbose\_name |

用户表的设计，通过继承扩展django内部auth模块自带的user表，生成我们自己的用户UserProfile表。注意：当使用继承后，需要在settings配置用户默认模型为自己创建的这个表，要不还是内部的user表settings设置如下：

|  |
| --- |
| AUTH\_USER\_MODEL = 'users.UserProfile' |

### 4.2、机构orgs里面models设计

|  |
| --- |
| **from** django**.**db **import** models  **from** datetime **import** datetime  # Create your models here.  **class** **CityInfo(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"城市名称"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** "城市信息"  verbose\_name\_plural **=** verbose\_name  **class** **OrgInfo(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"机构名称"**)**  image **=** models**.**ImageField**(**upload\_to**=**'orgs/%y/%m/%d'**,**verbose\_name**=**"机构封面"**)**  study\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"学习人数"**)**  address **=** models**.**CharField**(**max\_length**=**200**,**verbose\_name**=**"机构地址"**)**  org\_category **=** models**.**CharField**(**choices**=((**'个人'**,**'个人'**),(**'高校'**,**'高校'**),(**'培训机构'**,**'培训机构'**)),**max\_length**=**16**,**verbose\_name**=**"机构类别"**,**default**=**'培训机构'**)**  desc **=** models**.**CharField**(**max\_length**=**300**,**verbose\_name**=**"机构简介"**)**  detail **=** models**.**TextField**(**verbose\_name**=**"机构详情"**)**  course\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"机构课程数"**)**  click\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"访问量"**)**  love\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"收藏人数"**)**  city **=** models**.**ForeignKey**(**CityInfo**,**verbose\_name**=**'所属城市'**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** '机构信息'  verbose\_name\_plural **=** verbose\_name  **class** **TeacherInfo(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"教师姓名"**)**  image **=** models**.**ImageField**(**upload\_to**=**'teachers/%y/%m/%d'**,** verbose\_name**=**"教师头像"**)**  age **=** models**.**IntegerField**(**default**=**30**,**verbose\_name**=**"教师年龄"**)**  work\_year **=** models**.**IntegerField**(**default**=**3**,**verbose\_name**=**"工作年限"**)**  work\_position **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"工作职位"**)**  work\_company **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"就职公司"**)**  work\_style **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"教学特点"**)**  click\_num **=** models**.**IntegerField**(**default**=**0**,** verbose\_name**=**"访问量"**)**  love\_num **=** models**.**IntegerField**(**default**=**0**,** verbose\_name**=**"收藏人数"**)**  org **=** models**.**ForeignKey**(**OrgInfo**,**verbose\_name**=**"所属机构"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,** verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** '教师信息'  verbose\_name\_plural **=** verbose\_name |

### 4.3、课程courses里面models设计

|  |
| --- |
| **from** django**.**db **import** models  **from** orgs**.**models **import** OrgInfo**,**TeacherInfo  **from** datetime **import** datetime  **from** DjangoUeditor**.**models **import** UEditorField  # Create your models here.  **class** **CourseInfo(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"课程名称"**)**  image **=** models**.**ImageField**(**upload\_to**=**'courses/%y/%m/%d'**,**verbose\_name**=**"课程封面"**,**max\_length**=**200**)**  study\_time **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"学习时长"**)**  love\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"收藏数"**)**  desc **=** models**.**CharField**(**max\_length**=**300**,**verbose\_name**=**"课程简介"**)**  detail **=** models**.**TextField**(**verbose\_name**=**"课程详情"**)**  study\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"学习人数"**)**  comment\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"评论数"**)**  level **=** models**.**CharField**(**choices**=((**'初级'**,**'初级'**),(**'中级'**,**'中级'**),(**'高级'**,**'高级'**)),**max\_length**=**10**,**verbose\_name**=**"课程难度"**,**default**=**'高级'**)**  course\_category **=** models**.**CharField**(**choices**=((**'前端开发'**,**'前端开发'**),(**'后端开发'**,**'后端开发'**)),**verbose\_name**=**"课程类别"**,**max\_length**=**15**,**default**=**'后端开发'**)**  click\_num **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"访问量"**)**  org **=** models**.**ForeignKey**(**OrgInfo**,**verbose\_name**=**"所属机构"**)**  teacher **=** models**.**ForeignKey**(**TeacherInfo**,**verbose\_name**=**"所属教师"**)**  course\_bull **=** models**.**CharField**(**max\_length**=**200**,**verbose\_name**=**"课程公告"**,**null**=True,**blank**=True)**  teacher\_tell **=** models**.**CharField**(**max\_length**=**200**,**verbose\_name**=**"讲师提示"**,**null**=True,**blank**=True)**  need\_known **=** models**.**CharField**(**max\_length**=**200**,**verbose\_name**=**"课程须知"**,**null**=True,**blank**=True)**  is\_banner **=** models**.**BooleanField**(**default**=False,**verbose\_name**=**"是否轮播"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** '课程信息'  verbose\_name\_plural **=** verbose\_name  **class** **LessonInfo(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"章节名称"**)**  course **=** models**.**ForeignKey**(**CourseInfo**,**verbose\_name**=**"所属课程"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,** verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** '章节信息'  verbose\_name\_plural **=** verbose\_name  **class** **VideoInfo(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"视频名称"**)**  study\_time **=** models**.**IntegerField**(**default**=**0**,**verbose\_name**=**"学习时长"**)**  url **=** models**.**URLField**(**max\_length**=**200**,**verbose\_name**=**"视频链接"**)**  lesson **=** models**.**ForeignKey**(**LessonInfo**,**verbose\_name**=**"所属章节"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,** verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** '视频信息'  verbose\_name\_plural **=** verbose\_name  **class** **SourceInfo(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"资料名称"**)**  download **=** models**.**FileField**(**upload\_to**=**'sources/%y/%m/%d'**,**max\_length**=**200**,**verbose\_name**=**"资料下载"**)**  course **=** models**.**ForeignKey**(**CourseInfo**,**verbose\_name**=**"所属课程"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,** verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** '资源信息'  verbose\_name\_plural **=** verbose\_name |

### 4.4、用户操作operations里面 models设计

|  |
| --- |
| **from** django**.**db **import** models  **from** users**.**models **import** UserProfile  **from** courses**.**models **import** CourseInfo  **from** datetime **import** datetime  # Create your models here.  **class** **UserLove(**models**.**Model**):**  love\_man **=** models**.**ForeignKey**(**UserProfile**,**verbose\_name**=**"收藏人"**)**  love\_id **=** models**.**IntegerField**(**verbose\_name**=**"收藏id"**)**  love\_type **=** models**.**IntegerField**(**choices**=((**1**,**'机构'**),(**2**,**'课程'**),(**3**,**'教师'**)),**verbose\_name**=**"收藏类型"**)**  love\_status **=** models**.**BooleanField**(**default**=False,**verbose\_name**=**"收藏状态"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"收藏时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**love\_man**.**username  **class** **Meta:**  verbose\_name **=** '收藏信息'  verbose\_name\_plural **=** verbose\_name  **class** **UserCourse(**models**.**Model**):**  study\_man **=** models**.**ForeignKey**(**UserProfile**,**verbose\_name**=**"学习人"**)**  study\_course **=** models**.**ForeignKey**(**CourseInfo**,**verbose\_name**=**"学习课程"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,** verbose\_name**=**"添加时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**study\_man**.**username  **class** **Meta:**  unique\_together **=** **(**'study\_man'**,**'study\_course'**)**  verbose\_name **=** '用户学习课程信息'  verbose\_name\_plural **=** verbose\_name  **class** **UserAsk(**models**.**Model**):**  name **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"咨询姓名"**)**  phone **=** models**.**CharField**(**max\_length**=**11**,**verbose\_name**=**"咨询手机"**)**  course **=** models**.**CharField**(**max\_length**=**20**,**verbose\_name**=**"咨询课程"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**"咨询时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**name  **class** **Meta:**  verbose\_name **=** '用户咨询信息'  verbose\_name\_plural **=** verbose\_name  **class** **UserComment(**models**.**Model**):**  comment\_man **=** models**.**ForeignKey**(**UserProfile**,**verbose\_name**=**"评论人"**)**  comment\_course **=** models**.**ForeignKey**(**CourseInfo**,**verbose\_name**=**"评论课程"**)**  content **=** models**.**CharField**(**max\_length**=**300**,**verbose\_name**=**"评论内容"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,** verbose\_name**=**"评论时间"**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**content  **class** **Meta:**  verbose\_name **=** '用户评论信息'  verbose\_name\_plural **=** verbose\_name  **class** **UserMessage(**models**.**Model**):**  msg\_user **=** models**.**IntegerField**(**verbose\_name**=**"消息用户"**)**  msg\_content **=** models**.**CharField**(**max\_length**=**100**,**verbose\_name**=**"消息内容"**)**  msg\_status **=** models**.**BooleanField**(**default**=False,**verbose\_name**=**"消息状态"**)**  add\_time **=** models**.**DateTimeField**(**default**=**datetime**.**now**,**verbose\_name**=**'添加时间'**)**  **def** \_\_str\_\_**(**self**):**  **return** self**.**msg\_content  **class** **Meta:**  verbose\_name **=** '用户消息信息'  verbose\_name\_plural **=** verbose\_name |

到此为止：我们各个模块的模型类就算是全部初步建立完成，接下来我们要考虑给项目添加点测试数据，测试数据的添加后台添加事最快的，因此我们接下来需要去配置我们的后台。

## 5、xadmin使用

### 5.1、xadmin的简介

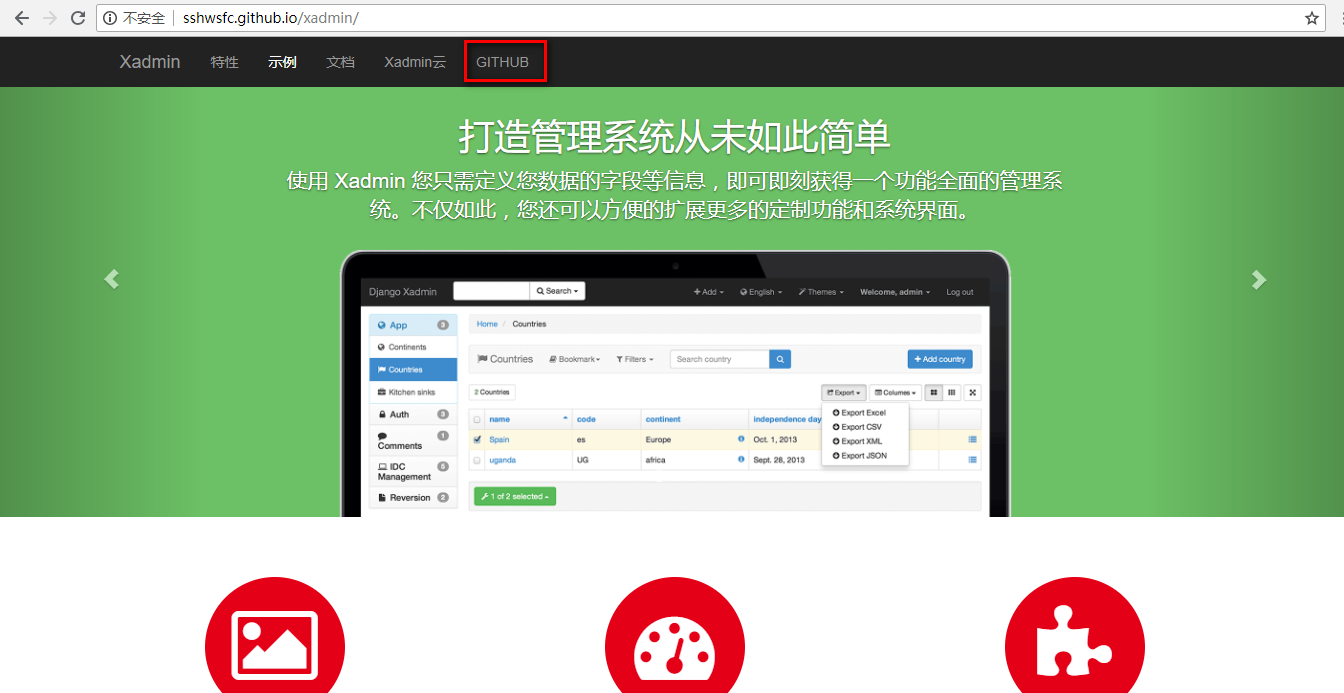
<http://sshwsfc.github.io/xadmin/> xadmin的官网

xadmin是一个django的管理后台实现，使用了更加灵活的架构设计及Bootstrap UI框架， 目的是替换现有的admin，国人开发，有许多新的特性。详情参考上面xadmin官方网站

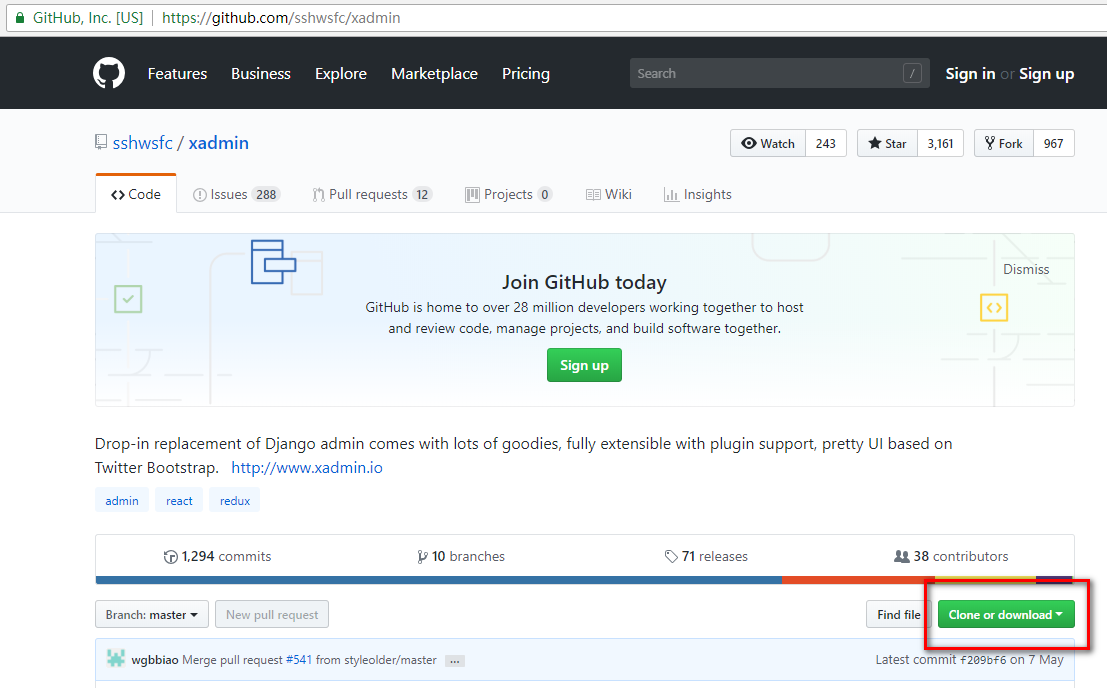
### 5.2、xadmin的下载和安装

#### 5.2.1、xadmin的下载

在xadmin官方网站上我们可以看到xadmin的介绍，并且我们也可以发现xadmin的源码被托管在GitHub上，因此，我们可以去GitHub轻松获取到xadmin源码进行安装。



点击进去后，我们可以看到GitHub的托管页面



点击如图所示按钮后，选择下载zip我们可以获取到xadmin的源码下载包

#### 5.2.2、xadmin的安装

下载源码包压缩文件之后，我们可以解压这个压缩文件，获取到源码包

1. 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
2. 创建extra\_apps放置第三方的app，将xadmin移动到我们这个extra\_apps下
3. 将extra\_apps mark为root\_source
4. 将extra\_apps在setting当中配置好搜索路径
5. 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
6. Pip install -r requirements.txt -i <https://pypi.douban.com/simple/>
7. 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

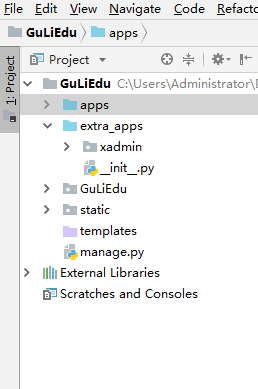
9、将我们本来的admin注释掉改为我们xadmin

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

最后我们的项目结构图如下：



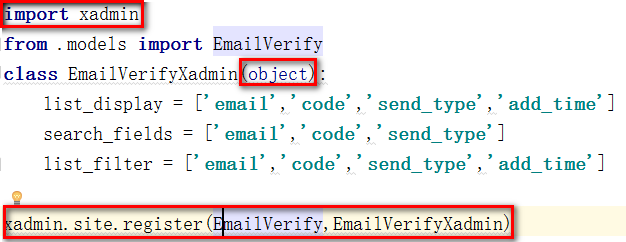
### 5.3、xadmin的注册

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。如下图所示



如此，把所有的APP当中的model都按照上述方法注册到adminx.py中并且在后台添加相应的测试数据，至此项目准备阶段告一段落，接下来我们就可以开始我们正常的业务逻辑了。

# 二、项目首页逻辑展示

1、项目模板继承的设计

通过浏览我们所有的模板页面，我们可以发现首页、教师的所有、课程的所有以及机构的列表页，这些页面是同样的一个布局，也就是说它们可以继承于同一个父模板base。剩余的机构详情4个页面共用一个父模板base2；还有用户中心的6个页面共用一个父模板base3。

按照我们的思路，分别为三种继承制作三套父模板是可以解决这个问题，但是，我们发现这三套模板最上方用户登陆的那一部分是相同的，如果分开去制作，那么用户登陆后的状态在三套模板当中都要去重新引入，这样，反而比较麻烦。因此我们最终选择多级继承去解决。如下图

