

# 十五分钟打造一个功能完备界面优雅的学生管理系统

---

- [项目要求](#)
- [需求分析](#)
- [框架选取](#)
  - [why django](#)
  - [why xadmin](#)
- [just do it](#)
  - [环境需求](#)
  - [详细步骤](#)
  - [github完整源码](#)
  - [相关截图](#)
- [相关参考](#)

## 项目要求

天津大学需要完成一个学生选课及成绩管理系统，该系统要求能够登记，修改，查询，统计学生、课程、选课的基本信息。

学生的基本信息包括：学号，姓名，性别，入学年龄，入学年份，班级。其中性别只能是（男或女），入学年龄在（10—50），学号10位长。

课程的基本信息包括：课程编号，课程名称，教课教师姓名，学分，课程适合年级，取消年份（可为空）。课程编号：7位长，只有学生的年级大于等于课程适合年级而且选课时间早于取消年份时方可选课。

选课信息：学生学号，课程号，选课年份，成绩。其中学号是指向学生表的外关键字，课程号是指向课程表的外关键字。要求如果学生退学则删除该生的所有选课信息。

功能：

1. 能够根据学生姓名或学号查询生的基本信息或所选课的情况。
2. 能够根据学生姓名或学号和课程名称或课程编号查询该生该课程的成绩。
3. 能够根据课程名称或课程编号查询课程的基本信息或该课程的选课情况。
4. 能够统计出学生的平均成绩，班级的平均成绩，课程的成绩分布（不及格，60—69，70—79，80—89，90—99，满分），课程的平均成绩
5. 能够修改学生，课程，选课信息中的所有信息。
6. 能够增加删除学生，课程，选课信息。
7. 要求数据库不限。
8. 编程语言不限
9. 程序架构不限

## 需求分析

题目要求搭建一个管理系统。

关于后台管理系统的框架林林总总，很多语言都提供了成熟的管理系统解决方案。在本次项目中，我将采用python语言的最流行的django框架，模板采用xadmin。

# 框架选取

## why django

django是python最流行的web开发框架。

Django使用更少的代码更快地构建更好的Web应用程序变得更容易。

引用[官网](#)介绍：

### 不可思议的快

Django的设计目的是帮助开发人员尽可能快地将应用程序从概念转换到完成。

### 令人放心的是安全的

Django非常重视安全性，帮助开发人员避免了许多常见的安全错误。

### 可伸缩的

Web上一些最繁忙的站点利用Django快速灵活扩展的能力。

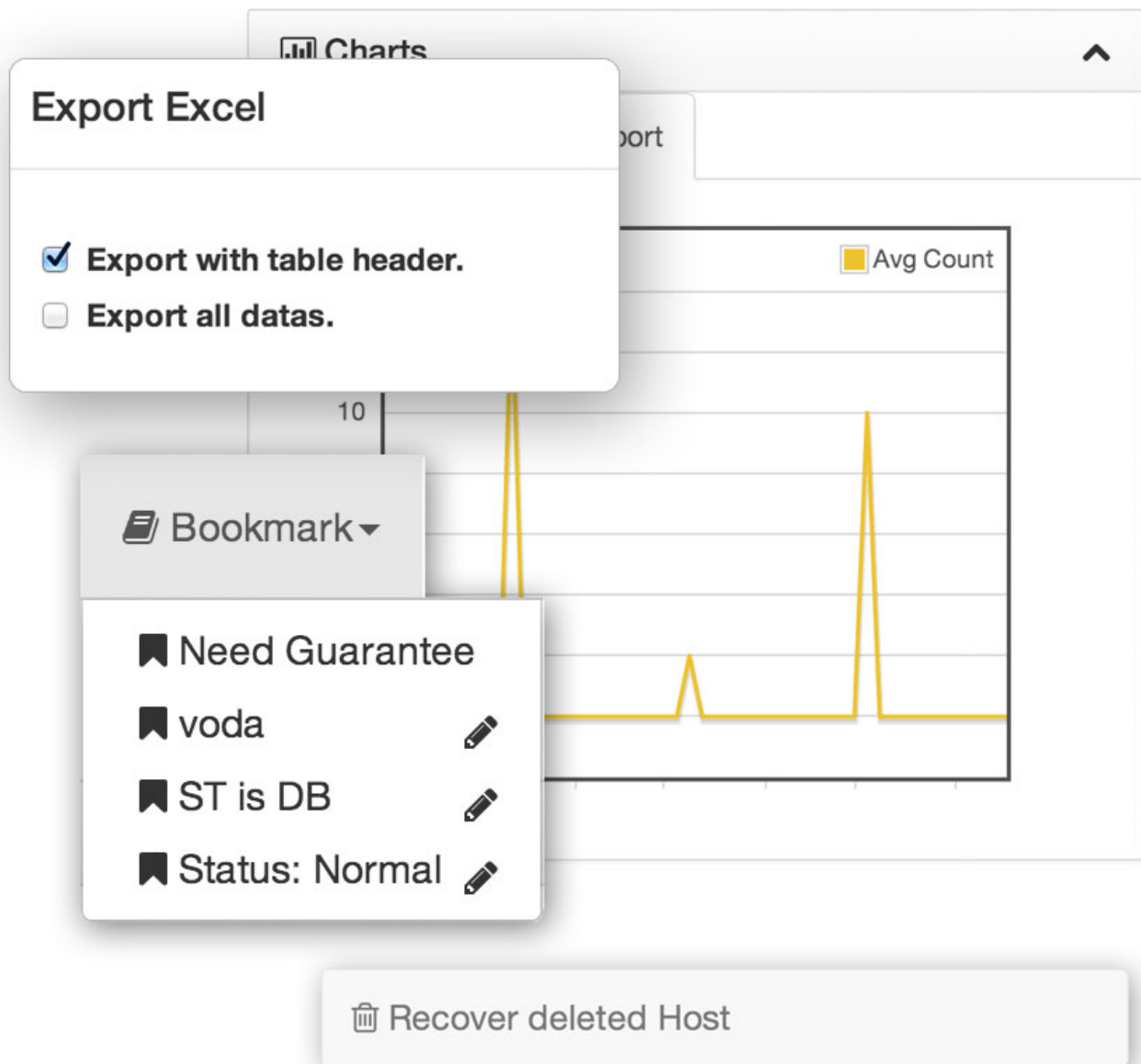
## why xadmin

xadmin是一个国人开发的专为打造后台管理系统的基于django的框架，功能强悍，使用简单。

引用[官网](#)介绍：

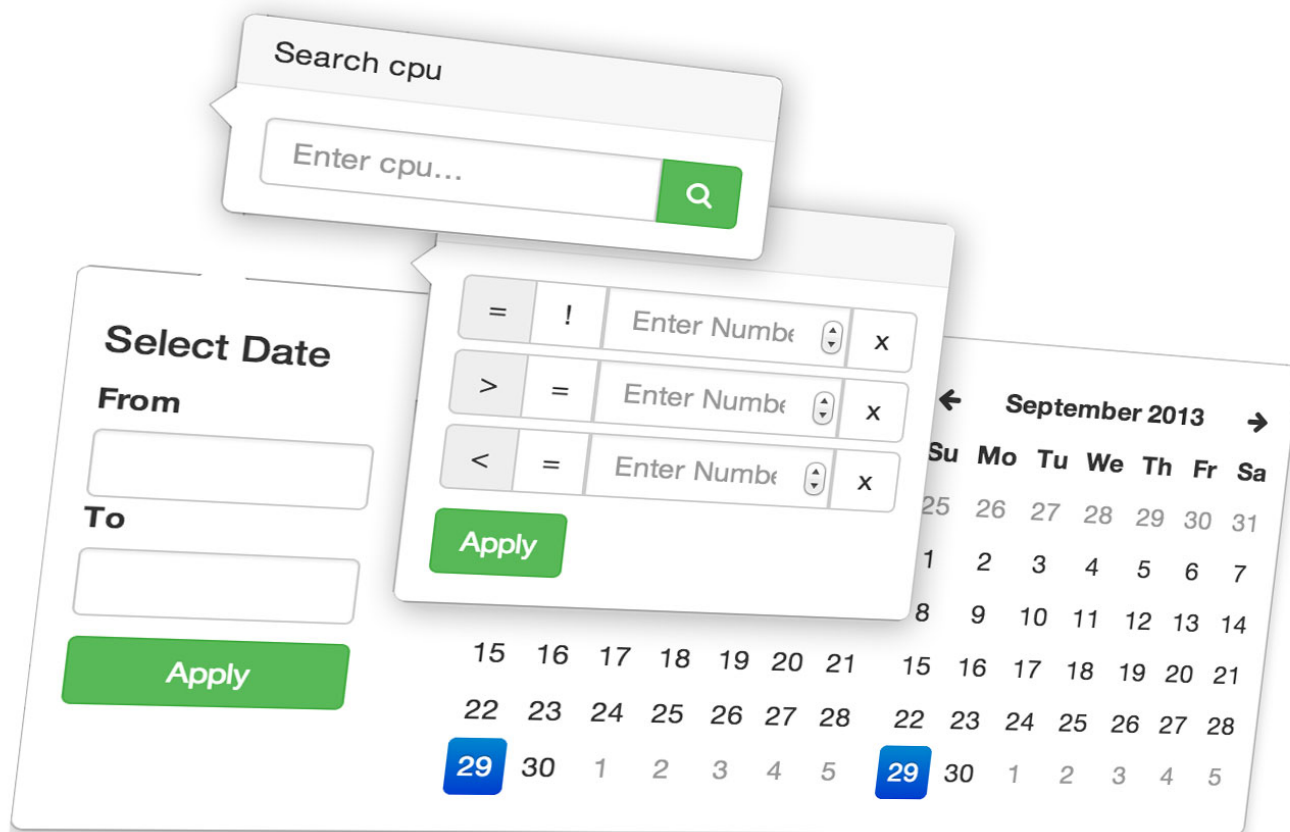
### 丰富的内置插件功能 基本满足管理系统的方方面面

Xadmin已经包含数据导出、书签、数据添加向导等十余项扩展功能。并且功能还在增加中。没有您急需的功能？没关系您可以到社区寻求帮助。



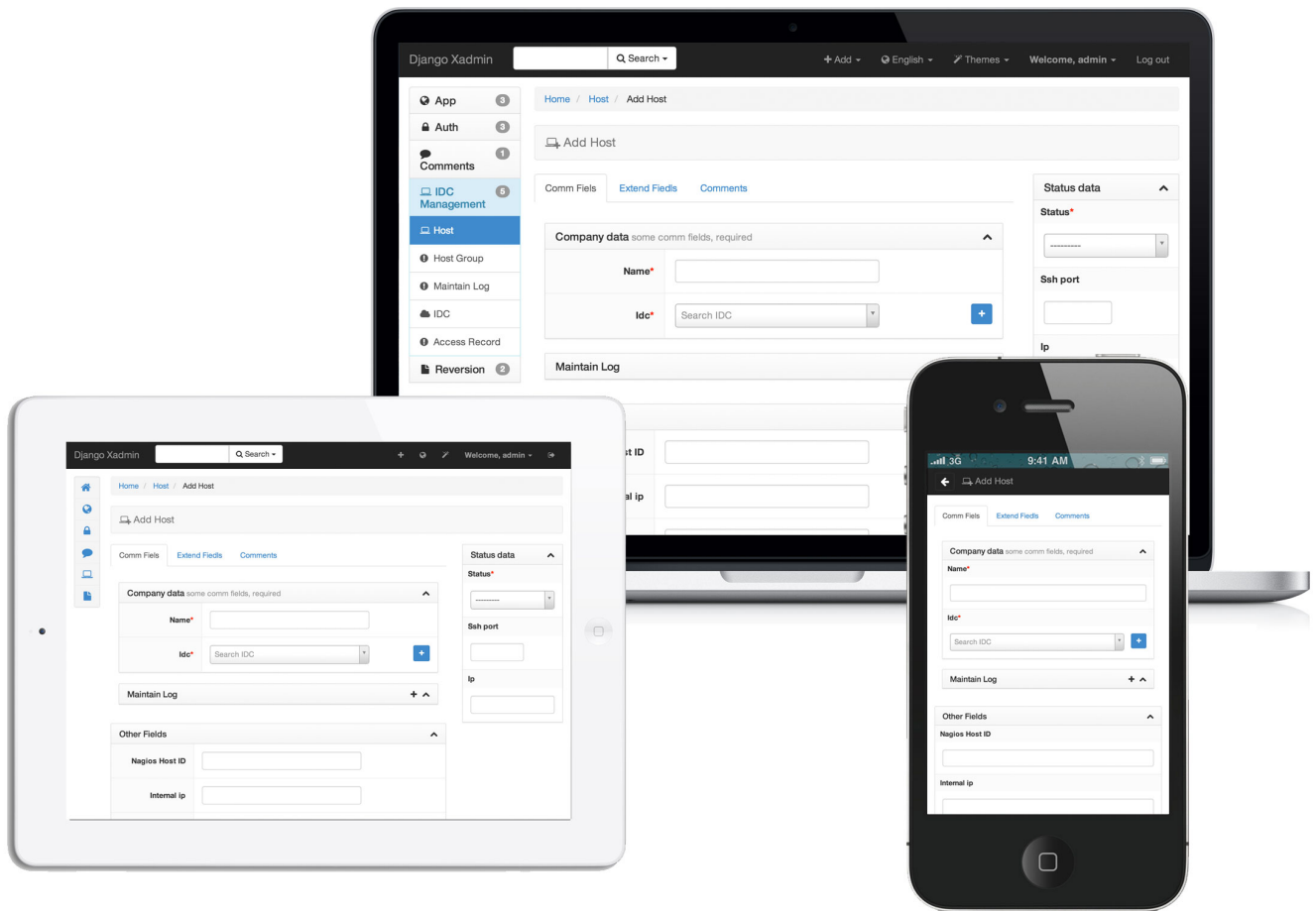
强大的过滤器功能 数据筛选，面面俱到

数据筛选是Xadmin的强项。通过Xadmin的过滤器，您可以使用模糊查找，数字范围查找，日期范围查找等多种高级查找方式。并且可以将您的查询结果存为书签方便今后使用。



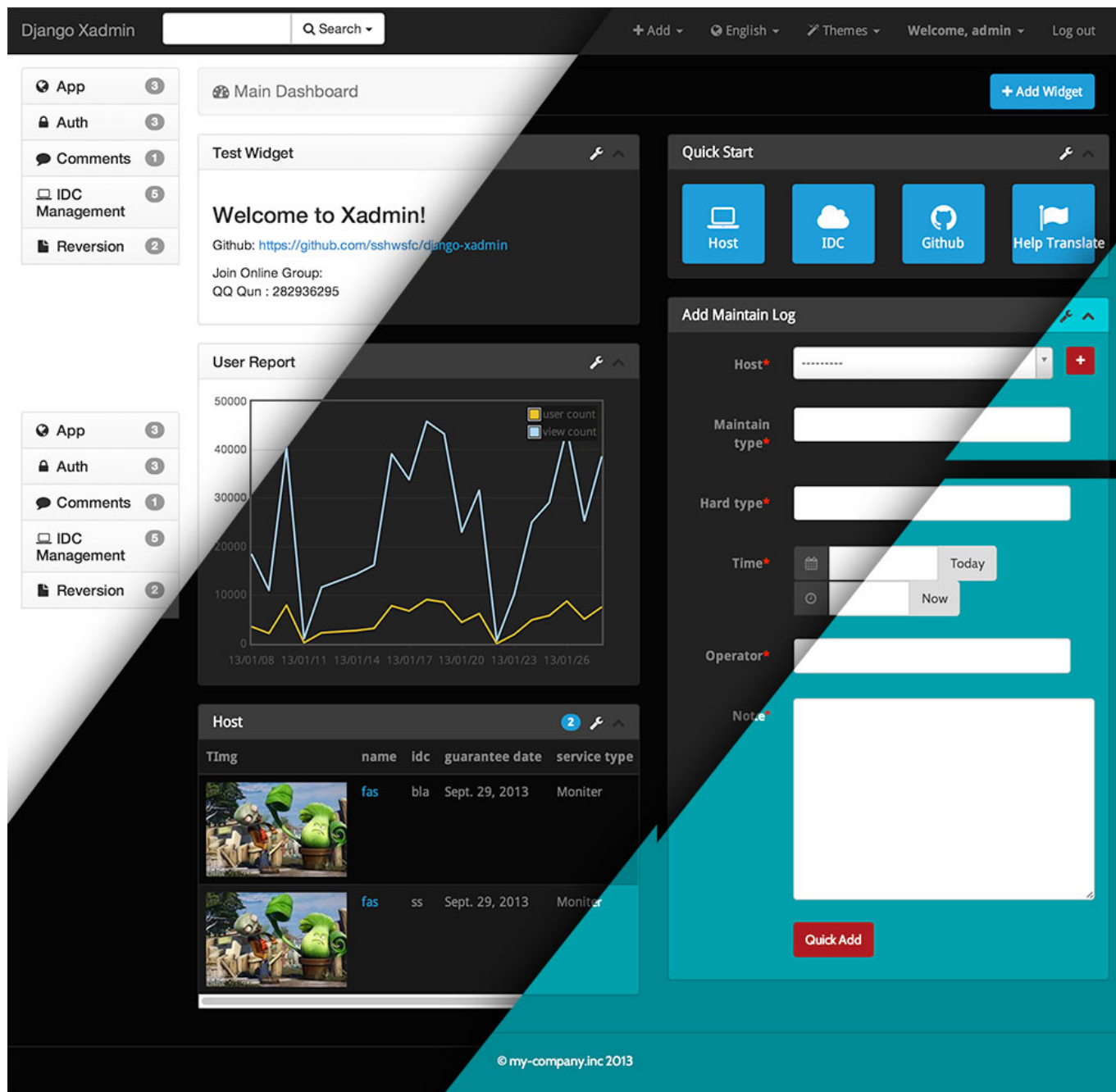
在多种屏幕上完美浏览及操作 **UI**的优势与生俱来

基于Bootstrap3，一切都变的神奇。Xadmin完美支持各种屏幕浏览，包括PC桌面，PAD浏览及手机浏览。让您的系统使用各种环境下的使用要求。



预先提供多种界面主题，更可以自己定制 管理系统也要绚丽多变

使用统一的UI框架天生就具备灵活的界面扩展性。Xadmin支持基于 Bootstrap3 的系统主题，当然您也可以自己制作主题，让您的系统天生多彩。



完善的系统权限 可配置，可定制，安全可靠

全新打造的Xadmin安全系统，控制粒度可以精细到每行数据，甚至每个字段，结合角色及用户组，几乎可以胜任任何变态的权限系统要求。

编写简单即时生效 这就是你想要的

Xadmin根据代码及时生成管理系统，并非采用模板方式。开发过程中代码修改即可及时查看修改后的效果。

## just do it

环境需求

win10, python3

详细步骤

安装Django:

```
pip install Django
```

安装xadmin2.0

```
pip install git+git://github.com/sshsfc/xadmin.git@django2
```

创建项目:

```
django-admin startproject project  
cd project  
django-admin startapp app
```

在url.py中配置路由:

```
import xadmin  
from django.urls import path  
from django.conf import settings  
from django.conf.urls.static import static  
  
urlpatterns = [  
    path(r'', xadmin.site.urls),  
]
```

在settings.py中设置相关配置:

添加本机白名单:

```
ALLOWED_HOSTS = ['127.0.0.1']
```

添加xadmin框架:

```
INSTALLED_APPS = [  
    'app.apps.AppConfig',  
    'xadmin',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',
```

```
    'crispy_forms',  
]
```

配置数据库，这里我们选择mysql，并且填充库名，用户名，密码：

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'stusys',  
        'USER': 'test',  
        'PASSWORD': '',  
        'HOST': '127.0.0.1',  
        'PORT': '3306',  
        'OPTIONS': {  
            'init_command': "SET sql_mode='STRICT_TRANS_TABLES';"  
        }  
    }  
}
```

在mysql里面创建相应的空数据库：

```
create database tjusys;
```

命令行里面同步数据库：

```
python manage.py migrate
```

创建超级用户：

```
python manage.py createsuperuser
```

在settings.py里面设置语言国际化：

```
from django.utils.translation import ugettext_lazy as _  
  
LANGUAGE_CODE = 'zh-hans'  
  
LANGUAGES = [  
    ('en', _('English')),  
    ('zh-hans', _('Simplified Chinese')),  
    ('zh-hant', _('Traditional Chinese')),  
]
```



在app/models.py中设置数据库模型:

```
from django.db import models

# Create your models here.
class Student(models.Model):
    SEX = (
        ('male', '男'),
        ('female', '女')
    )
    sid = models.BigIntegerField(verbose_name='学号', primary_key=True)
    sname = models.CharField(verbose_name='姓名', max_length=32)
    sex = models.CharField(choices=SEX, verbose_name='性别', max_length=8)
    inage = models.IntegerField(verbose_name='入学年龄')
    inyear = models.IntegerField(verbose_name='入学年份')
    sclass = models.CharField(verbose_name='班级', max_length=32)

    class Meta:
        verbose_name = '学生信息'
        verbose_name_plural = '学生信息'

    def __str__(self):
        return str(self.sid)

class Course(models.Model):
    cid = models.IntegerField(verbose_name='课程编号', primary_key=True)
    cname = models.CharField(verbose_name='课程名称', max_length=32)
    teacher_name = models.CharField(verbose_name='授课老师', max_length=32)
    credit = models.IntegerField(verbose_name='学分')
    suitable_grade = models.IntegerField(verbose_name='课程适合年级')
    cancel_year = models.IntegerField(verbose_name='取消年份', null=True,
    blank=True)

    class Meta:
        verbose_name = '课程信息'
        verbose_name_plural = '课程信息'

    def __str__(self):
        return str(self.cid)

class Enroll(models.Model):
    eid = models.AutoField(primary_key=True)
    student_sid = models.ForeignKey(Student, verbose_name='学生学号',
    on_delete=models.CASCADE)
    course_cid = models.ForeignKey(Course, verbose_name='课程编号',
    on_delete=models.DO_NOTHING)
    year = models.IntegerField(verbose_name='选课年份')
    score = models.IntegerField(verbose_name='成绩')

    class Meta:
        verbose_name = '选课信息'
        verbose_name_plural = '选课信息'
```

```
def __str__(self):  
    return str(self.eid)
```

在命令行中生成并迁移数据库:

```
python manage.py makemigrations  
python manage.py migrate
```

在mysql插入一些示例数据:

```
INSERT INTO `app_student`  
(`sid`,`sname`,`sex`,`inage`,`inyear`,`sclass`)  
VALUES  
(3017000001, '张三丰', 'male', 31, 2017, '软件工程1班'),  
(3017000002, '黄药师', 'male', 32, 2017, '软件工程2班'),  
(3017000003, '欧阳锋', 'male', 14, 2017, '软件工程3班'),  
(3017000004, '段智兴', 'male', 16, 2017, '软件工程1班'),  
(3017000005, '洪七公', 'male', 32, 2017, '软件工程2班'),  
(3017000006, '王重阳', 'male', 16, 2017, '软件工程3班'),  
(3017000007, '周伯通', 'male', 24, 2017, '软件工程1班'),  
(3017000008, '李寻欢', 'male', 25, 2017, '软件工程2班'),  
(3017000009, '陆小凤', 'male', 23, 2017, '软件工程3班'),  
(3017000010, '楚留香', 'male', 18, 2017, '软件工程1班'),  
(3017000011, '周芷若', 'female', 32, 2017, '软件工程2班'),  
(3017000012, '杨不悔', 'female', 13, 2017, '软件工程3班'),  
(3017000013, '殷素素', 'female', 23, 2017, '软件工程1班'),  
(3017000014, '小龙女', 'female', 15, 2017, '软件工程2班'),  
(3017000015, '李莫愁', 'female', 16, 2017, '软件工程3班'),  
(3017000016, '任盈盈', 'female', 25, 2017, '软件工程1班'),  
(3017000017, '温青青', 'female', 16, 2017, '软件工程2班'),  
(3017000018, '王语嫣', 'female', 35, 2017, '软件工程3班'),  
(3017000019, '苗若兰', 'female', 22, 2017, '软件工程1班'),  
(3017000020, '沐剑屏', 'female', 12, 2017, '软件工程2班');  
  
INSERT INTO `app_course`  
(`cid`,`cname`,`teacher_name`,`credit`,`suitable_grade`,`cancle_year`)  
VALUES  
(5100001, '温', '张老师', 4, 1, 2021),  
(5100002, '良', '张老师', 1, 3, 2018),  
(5100003, '温', '蔡老师', 1, 3, 2018),  
(5100004, '俭', '蔡老师', 5, 1, 2020),  
(5100005, '让', '张老师', 5, 4, 2018),  
(5100006, '仁', '张老师', 2, 4, 2019),  
(5100007, '义', '蔡老师', 2, 2, 2018),  
(5100008, '礼', '张老师', 4, 3, 2018),  
(5100009, '智', '李老师', 5, 2, 2021),  
(5100010, '信', '李老师', 2, 3, 2020),  
(5100011, '诗', '张老师', 3, 3, 2018),  
(5100012, '书', '蔡老师', 3, 1, 2018),
```

```
(5100013, '琴', '李老师', 5, 3, 2018),
(5100014, '棋', '蔡老师', 4, 2, 2021),
(5100015, '画', '李老师', 5, 3, 2019),
(5100016, '唱', '蔡老师', 3, 1, 2018),
(5100017, '跳', '李老师', 2, 3, 2021),
(5100018, 'rap', '蔡老师', 2, 4, 2019),
(5100019, '篮球', '蔡老师', 3, 1, 2018),
(5100020, 'music', '李老师', 1, 1, 2021);
```

```
INSERT INTO `app_enroll`
(`eid`,`year`,`score`,`course_cid_id`,`student_sid_id`)
VALUES
```

```
(0, 2019, 82, 5100001, 3017000001),
(1, 2019, 90, 5100004, 3017000001),
(2, 2019, 86, 5100009, 3017000001),
(3, 2019, 46, 5100014, 3017000001),
(4, 2019, 81, 5100020, 3017000001),
(5, 2019, 50, 5100001, 3017000002),
(6, 2019, 46, 5100004, 3017000002),
(7, 2019, 96, 5100009, 3017000002),
(8, 2019, 49, 5100014, 3017000002),
(9, 2019, 59, 5100020, 3017000002),
(10, 2019, 97, 5100001, 3017000003),
(11, 2019, 96, 5100004, 3017000003),
(12, 2019, 53, 5100009, 3017000003),
(13, 2019, 92, 5100014, 3017000003),
(14, 2019, 64, 5100020, 3017000003),
(15, 2019, 45, 5100001, 3017000004),
(16, 2019, 92, 5100004, 3017000004),
(17, 2019, 82, 5100009, 3017000004),
(18, 2019, 66, 5100014, 3017000004),
(19, 2019, 92, 5100020, 3017000004),
(20, 2019, 66, 5100001, 3017000005),
(21, 2019, 48, 5100004, 3017000005),
(22, 2019, 47, 5100009, 3017000005),
(23, 2019, 72, 5100014, 3017000005),
(24, 2019, 84, 5100020, 3017000005),
(25, 2019, 45, 5100001, 3017000006),
(26, 2019, 83, 5100004, 3017000006),
(27, 2019, 70, 5100009, 3017000006),
(28, 2019, 57, 5100014, 3017000006),
(29, 2019, 62, 5100020, 3017000006),
(30, 2019, 45, 5100001, 3017000007),
(31, 2019, 91, 5100004, 3017000007),
(32, 2019, 45, 5100009, 3017000007),
(33, 2019, 53, 5100014, 3017000007),
(34, 2019, 84, 5100020, 3017000007),
(35, 2019, 62, 5100001, 3017000008),
(36, 2019, 51, 5100004, 3017000008),
(37, 2019, 49, 5100009, 3017000008),
(38, 2019, 90, 5100014, 3017000008),
(39, 2019, 71, 5100020, 3017000008),
(40, 2019, 50, 5100001, 3017000009),
(41, 2019, 88, 5100004, 3017000009),
```

(42, 2019, 49, 5100009, 3017000009),  
(43, 2019, 69, 5100014, 3017000009),  
(44, 2019, 74, 5100020, 3017000009),  
(45, 2019, 86, 5100001, 3017000010),  
(46, 2019, 93, 5100004, 3017000010),  
(47, 2019, 52, 5100009, 3017000010),  
(48, 2019, 91, 5100014, 3017000010),  
(49, 2019, 88, 5100020, 3017000010),  
(50, 2019, 62, 5100001, 3017000011),  
(51, 2019, 51, 5100004, 3017000011),  
(52, 2019, 83, 5100009, 3017000011),  
(53, 2019, 69, 5100014, 3017000011),  
(54, 2019, 64, 5100020, 3017000011),  
(55, 2019, 68, 5100001, 3017000012),  
(56, 2019, 79, 5100004, 3017000012),  
(57, 2019, 96, 5100009, 3017000012),  
(58, 2019, 59, 5100014, 3017000012),  
(59, 2019, 76, 5100020, 3017000012),  
(60, 2019, 59, 5100001, 3017000013),  
(61, 2019, 87, 5100004, 3017000013),  
(62, 2019, 55, 5100009, 3017000013),  
(63, 2019, 57, 5100014, 3017000013),  
(64, 2019, 65, 5100020, 3017000013),  
(65, 2019, 75, 5100001, 3017000014),  
(66, 2019, 57, 5100004, 3017000014),  
(67, 2019, 82, 5100009, 3017000014),  
(68, 2019, 49, 5100014, 3017000014),  
(69, 2019, 53, 5100020, 3017000014),  
(70, 2019, 95, 5100001, 3017000015),  
(71, 2019, 65, 5100004, 3017000015),  
(72, 2019, 53, 5100009, 3017000015),  
(73, 2019, 76, 5100014, 3017000015),  
(74, 2019, 52, 5100020, 3017000015),  
(75, 2019, 83, 5100001, 3017000016),  
(76, 2019, 48, 5100004, 3017000016),  
(77, 2019, 50, 5100009, 3017000016),  
(78, 2019, 87, 5100014, 3017000016),  
(79, 2019, 95, 5100020, 3017000016),  
(80, 2019, 89, 5100001, 3017000017),  
(81, 2019, 49, 5100004, 3017000017),  
(82, 2019, 48, 5100009, 3017000017),  
(83, 2019, 80, 5100014, 3017000017),  
(84, 2019, 68, 5100020, 3017000017),  
(85, 2019, 67, 5100001, 3017000018),  
(86, 2019, 72, 5100004, 3017000018),  
(87, 2019, 84, 5100009, 3017000018),  
(88, 2019, 62, 5100014, 3017000018),  
(89, 2019, 64, 5100020, 3017000018),  
(90, 2019, 63, 5100001, 3017000019),  
(91, 2019, 92, 5100004, 3017000019),  
(92, 2019, 98, 5100009, 3017000019),  
(93, 2019, 99, 5100014, 3017000019),  
(94, 2019, 48, 5100020, 3017000019),  
(95, 2019, 52, 5100001, 3017000020),

```
(96, 2019, 73, 5100004, 3017000020),
(97, 2019, 76, 5100009, 3017000020),
(98, 2019, 45, 5100014, 3017000020),
(99, 2019, 92, 5100020, 3017000020);
```

将app/admin.py文件重命名为adminx.py，在里面注册模块：

```
import xadmin
from .models import *

class StudentAdmin(object):
    list_display = ('sid', 'sname', 'sex', 'inage', 'inyear', 'sclass')

class CourseAdmin(object):
    list_display = ('cid', 'cname', 'teacher_name', 'credit', 'suitable_grade',
                    'cancle_year')

class EnrollAdmin(object):
    list_display = ('student_sid', 'course_cid', 'year', 'score')

xadmin.site.register(Student, StudentAdmin)
xadmin.site.register(Course, CourseAdmin)
xadmin.site.register(Enroll, EnrollAdmin)
```

添加搜索、过滤、排序功能：

```
class StudentAdmin(object):
    search_fields = (
        'sid',
        'sname',
    )
    list_filter = (
        'sid',
        'sname',
        'sex',
        'inage',
        'inyear',
        'sclass',
    )

class CourseAdmin(object):
    search_fields = (
        'cid',
        'cname',
        'teacher_name',
    )
    list_filter = (
        'cid',
        'cname',
    )
```

```

        'teacher_name',
        'credit',
        'suitable_grade',
        'cancle_year',
    )

class EnrollAdmin(object):
    search_fields = (
        'student_sid__sid',
        'student_sid__sname',
        'student_sid__sclass',
        'course_cid__cid',
        'course_cid__cname',
    )
    list_filter = (
        'student_sid',
        'course_cid',
        'year',
        'score',
    )

```

登陆页面定制:

```

from xadmin.views.website import LoginView

class LoginViewAdmin(LoginView):
    title = '学生信息管理系统'

xadmin.site.register(LoginView, LoginViewAdmin)

```

定制网页标题和页脚:

```

from xadmin.views import CommAdminView

class GlobalSetting(CommAdminView):
    site_title = '学生信息管理系统'
    site_footer = 'Copyright 刘坤鑫 天津大学 2019'
    #menu_style = 'accordion'

xadmin.site.register(CommAdminView, GlobalSetting)

```

修改每页最多显示条数:

```
list_per_page=20
```

主题多样化:

```

from xadmin import views

class BaseSetting(object):
    """
    主题样式多样化
    """
    enable_themes=True
    use_bootswatch=True

xadmin.site.register(views.BaseAdminView,BaseSetting)

```

原生xadmin并不支持对数据进行统计操作。我的解决方法是利用内建的action操作对数据进行统计，最后返回一个json结果。

首先：

在app下新建xadmin\_action.py文件：

```

import json
from django.http import HttpResponse
from xadmin.plugins.actions import BaseActionView

class GetAvgScoreAction(BaseActionView):
    action_name = "get_avg_score"
    description = '显示成绩分布'
    model_perm = 'change'
    def do_action(self, queryset):
        res={
            '统计人数': len(queryset),
            '平均值': 0,
            '不及格': 0,
            '60-69': 0,
            '70-79': 0,
            '80-89': 0,
            '90-99': 0,
            '满分': 0,
        }
        for obj in queryset:
            res['平均值'] += obj.score
            if obj.score < 60:
                res['不及格'] += 1
            elif obj.score < 70:
                res['60-69'] += 1
            elif obj.score < 80:
                res['70-79'] += 1
            elif obj.score < 90:
                res['80-89'] += 1
            elif obj.score < 100:
                res['90-99'] += 1
            else:

```

```
        res['满分'] += 1
    res['平均值']/=len(queryset)
    return HttpResponse(json.dumps(res), content_type='application/json')
```

在admin.py对应类里面添加:

```
class EnrollAdmin(object):
    actions = [
        GetAvgScoreAction,
    ]
```

运行服务器:

```
python manage.py runserver
```

github完整源码

<https://github.com/qq734628996/student-management-system>

相关截图

登陆界面:



pc端界面:



学生信息列表 | 学生信息管理系统

127.0.0.1:8000/app/student/

学生信息管理系统

增加 主题 欢迎, root 注销

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

学生信息

学生信息

搜索 学生信息

添加学生信息

21 学生信息

1 2 显示全部

导出 显示列 列表 编辑

学号	姓名	性别	入学年龄	入学年份	班级	
3017218061	刘坤鑫	男	19	2017	软件工程1班	
3017000020	沐剑屏	女	12	2017	软件工程2班	
3017000019	雷莉兰	女	22	2017	软件工程1班	
3017000018	王浩瑞	女	35	2017	软件工程3班	
3017000017	温青青	女	16	2017	软件工程2班	
3017000016	任盈盈	女	25	2017	软件工程1班	
3017000015	李慕愁	女	16	2017	软件工程3班	
3017000014	小龙女	女	15	2017	软件工程2班	
3017000013	殷素素	女	23	2017	软件工程1班	
3017000012	杨不悔	女	13	2017	软件工程3班	
3017000011	周芷若	女	32	2017	软件工程2班	
3017000010	楚留香	男	18	2017	软件工程1班	
3017000009	陆小凤	男	23	2017	软件工程3班	
3017000008	李寻欢	男	25	2017	软件工程2班	
3017000007	周伯通	男	24	2017	软件工程1班	
3017000006	王重阳	男	16	2017	软件工程3班	
3017000005	洪七公	男	32	2017	软件工程2班	
3017000004	段智兴	男	16	2017	软件工程1班	
3017000003	欧阳锋	男	14	2017	软件工程3班	

选中了 20 个中的 0 个

pad端界面:






增加一个学生:

增加 学生信息

学号*	3017218061
姓名*	刘坤鑫
性别*	<div> -----▲</div> <div>男</div> <div>女</div>
入学年龄*	
入学年份*	2017
班级*	软件工程1班

 保存

保存并增加另一个

保存并继续编辑

增加一个课程：

增加 课程信息

请修正下面的错误。

课程编号*	<div>fdsaf</div> <div>输入整数。</div>
课程名称*	<div>篮球</div>
授课老师*	<div>蔡老湿</div>
学分*	<div>4</div>
课程适合年级*	<div>2</div>
取消年份	<div></div>

保存

保存并增加另一个

保存并继续编辑

选中复选框然后删除，可以全选

<input checked="" type="checkbox"/>	3017000004	段智兴
<input checked="" type="checkbox"/>	3017000003	欧阳锋
<input type="checkbox"/>	3017000002	黄药师

✕ 删除所选的 学生信息

🔧 选中了 20 个中的 2 个 ▶

确认删除：

你确定吗？

请确认要删除选中的 学生信息 吗？以下所有对象和余它们相关的条目将都会被删除：

学生信息: [3017000004](#)

选课信息: [15](#)

选课信息: [16](#)

选课信息: [17](#)

选课信息: [18](#)

选课信息: [19](#)

学生信息: [3017000003](#)

选课信息: [10](#)

选课信息: [11](#)

选课信息: [12](#)

选课信息: [13](#)

选课信息: [14](#)

是的，我确定

取消

查看某个学生的选课信息：

班级	
软件工程1班	
软件	<div>  选课信息                      </div>

查看结果：

5 选课信息

导出 显示列

<input type="checkbox"/>	课程编号	选课年份	成绩
<input type="checkbox"/>	5100020	2019	92
<input type="checkbox"/>	5100014	2019	45
<input type="checkbox"/>	5100009	2019	76
<input type="checkbox"/>	5100004	2019	73
<input type="checkbox"/>	5100001	2019	52

选中了 5 个中的 0 个

5 选课信息

在选课栏里查看学生详细信息：

3017000001详情

学号	3017000001
姓名	张三丰
性别	男
入学年龄	31
入学年份	2017
班级	软件工程1班

Close Edit

<input type="checkbox"/>	3017000019	5100004	2019	92
<input type="checkbox"/>	3017000019	5100001	2019	63
<input type="checkbox"/>	3017000018	5100020	2019	64

在选课栏里查看课程详细信息：

课程编号	5100014
课程名称	棋
授课老师	蔡老师
学分	4
课程适合年级	2
取消年份	2021

Close

Edit

<input type="checkbox"/>	3017000019		5100004		2019	92
<input type="checkbox"/>	3017000019		5100001		2019	63
<input type="checkbox"/>	3017000018		5100020		2019	64

按学号搜索：

[首页](#)  
信息管理  
[○ 学生信息](#)  
[○ 课程信息](#)  
[○ 选课信息](#)  
管理  
[⚙ 日志](#)  
认证和授权  
[👥 组](#)  
[👤 用户](#)  
[🔒 权限](#)

首页 / 学生信息

学生信息 x 3017000001 [+ 增加 学生信息](#)

1 学生信息

<input type="checkbox"/>	学号	姓名	性别	入学年龄	入学年份	班级	
<input type="checkbox"/>	3017000001	张三丰	男	31	2017	软件工程1班	

🔍 选中了 1 个中的 0 个 ▲

1 学生信息

按姓名搜索，这里使用了模糊搜索



学生信息管理系统

Q 搜索

+ 增加 主题 欢迎, root 注销

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

首页 / 学生信息

学生信息 书签 过滤器 x 李 Q + 增加 学生信息

2 学生信息 导出 显示列 视图 编辑

学号	姓名	性别	入学年龄	入学年份	班级	
3017000015	李莫愁	女	16	2017	软件工程3班	
3017000008	李寻欢	男	25	2017	软件工程2班	

选中了 2 个中的 0 个

2 学生信息

也可以用过滤器进行搜索，功能更强悍：

过滤器

搜索 学生信息

搜索 学号

学号

姓名

性别

入学年龄

入学年份

班级

=

!

输入数字

X

>

=

输入数字

X

<

=

输入数字

X

应用

苗若兰

女

22

2017

搜索1班所有人的成绩：

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

首页 / 选课信息

选课信息

书签

过滤器

x1班

+

增加选课信息

35 选课信息

1

2

显示全部

导出

显示列

	学生学号	课程编号	选课年份	成绩
	3017000001	5100001	2019	82
	3017000019	5100020	2019	48
	3017000019	5100014	2019	99
	3017000019	5100009	2019	98
	3017000019	5100004	2019	92
	3017000019	5100001	2019	63
	3017000016	5100020	2019	95
	3017000016	5100014	2019	87
	3017000016	5100009	2019	50
	3017000016	5100004	2019	48
	3017000016	5100001	2019	83
	3017000013	5100020	2019	65
	3017000013	5100014	2019	57
	3017000013	5100009	2019	55

按学号或课程编号查看成绩：

过滤器

搜索 选课信息

学生学号

课程编号

选课年份

成绩

全部

3017000001

3017000002

3017000003

3017000004

3017000005

3017000006

3017000007

过滤器可以多重过滤，例如这里查看1班中所有男生信息：

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

首页 / 学生信息 / 3017218061

修改 3017218061

学号*	3017218061
姓名*	刘坤鑫
性别*	男
入学年龄*	19
入学年份*	2017
班级*	软件工程1班

保存

保存并增加另一个

保存并继续编辑

删除

修改学生信息：

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

首页 / 学生信息

学生信息 书签 过滤器 2 搜索 学生信息 增加 学生信息

5 学生信息 导出 显示列 删除

	学号	姓名	性别	入学年龄	入学年份	班级	
<input type="checkbox"/>	3017218061	刘坤鑫	男	19	2017	软件工程1班	
<input type="checkbox"/>	3017000010	楚留青	男	18	2017	软件工程1班	
<input type="checkbox"/>	3017000007	周伯通	男	24	2017	软件工程1班	
<input type="checkbox"/>	3017000004	段智兴	男	16	2017	软件工程1班	
<input type="checkbox"/>	3017000001	张三丰	男	31	2017	软件工程1班	

选中了 5 个中的 0 个

先搜索自己想要查询的成绩，全选后查看成绩分布：

☒

3017000010

i

5100009

☒

i

5100004

删除所选的 选课信息

显示成绩分布

选中了 35 个

清除选中

35 选课信息

1

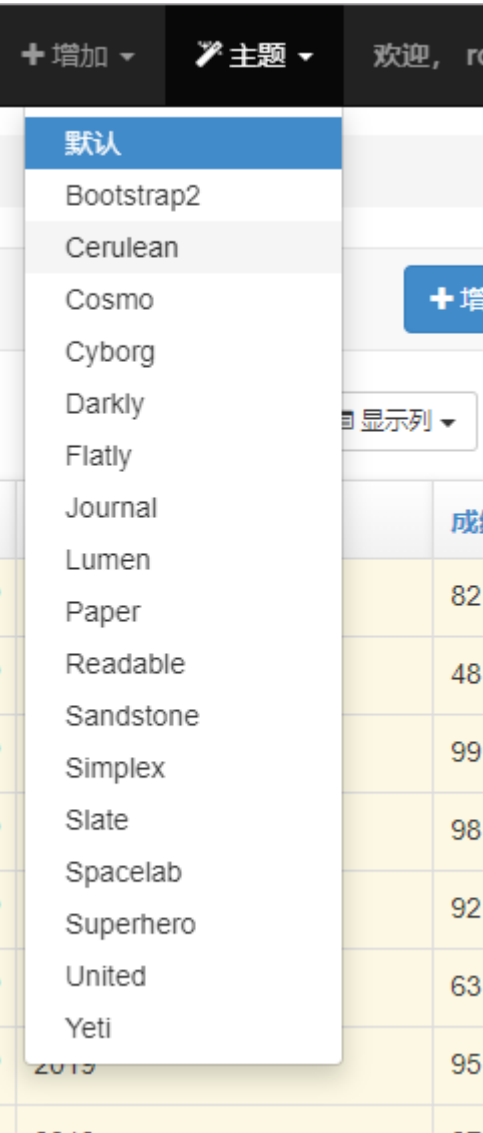
2

显示全部

查看结果：

```
← → ↺ ⓘ 127.0.0.1:8000/app/enroll/?_q_=1班
{
  统计人数: 35,
  平均值: 73.6,
  不及格: 12,
  60-69: 3,
  70-79: 0,
  80-89: 10,
  90-99: 10,
  满分: 0
}
```

换肤：



增加书签或其他功能：



管理日志：

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

首页 / 日志

日志

书签

过滤器

搜索 日志

+ 增加 日志

15 日志

导出

显示列

	记录时间	用户	操作IP	日志	
<input type="checkbox"/>	2019年5月8日 09:17	root	127.0.0.1	批量删除 1 个 课程信息	
<input type="checkbox"/>	2019年5月8日 09:17	root	127.0.0.1	添加 "5435"。	管理对象
<input type="checkbox"/>	2019年5月7日 20:42	root	127.0.0.1	修改 "3017218061" - 修改 sclass	管理对象
<input type="checkbox"/>	2019年5月7日 19:06	root	127.0.0.1	添加 "3017218061"。	管理对象
<input type="checkbox"/>	2019年5月7日 18:13	root	127.0.0.1	批量删除 3 个 选课信息	
<input type="checkbox"/>	2019年5月7日 18:09	root	127.0.0.1	添加 "206"。	管理对象

管理用户：

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

首页 / 用户

用户

书签

过滤器

搜索 用户

+ 增加 用户

1 用户

导出

显示列

	用户名	电子邮件地址	名字	姓氏	职员状态	
<input type="checkbox"/>	root	1@1.com			<span>✓</span>	

选中了 1 个中的 0 个

1 用户

管理权限：

首页

信息管理

学生信息

课程信息

选课信息

管理

日志

认证和授权

组

用户

权限

首页 / 权限

权限 书签

+ 增加 权限

52 权限 1 2 显示全部

导出 显示列 视图 刷新

权限	
添加:日志	
修改:日志	
删除:日志	
查看:日志	
添加:课程信息	
修改:课程信息	
删除:课程信息	

页脚logo

指定显示列：

导出 显示列 视图 刷新

恢复显示列

✓ 学号

✓ 姓名

✓ 性别

✓ 入学年龄

✓ 入学年份

✓ 班级

导出数据：

导出 显示列 视图 刷新

导出 Excel 2007

导出 Excel

导出 CSV

导出 XML

导出 JSON

功能过多不在此一一展示，更多功能可以参考xadmin官网或者我再github上的源码。

## 相关参考

xadmin快速入门:

```
https://www.cnblogs.com/v88v/p/8858853.html  
https://www.cnblogs.com/v88v/p/8886764.html  
https://www.cnblogs.com/v88v/p/8909845.html
```

django的models操作:

```
https://www.cnblogs.com/maociping/p/5255834.html  
https://docs.djangoproject.com/en/1.11/topics/db/models/  
https://www.cnblogs.com/kimyeee/p/7241754.html  
https://www.jianshu.com/p/b23dc2a3508d?utm_campaign
```

xadmin常用样式功能:

```
https://blog.csdn.net/weixin_42042680/article/details/84799260
```