

数据库配置和表说明

1.数据库连接配置

数据库使用可以使 mysql,也可是其他关系型数据库，但需要修改加载的中间件。以 mysql 为例，在 webapp 目录下，修改 webapp/settings.py,找到 DATABASES 配置项，修改相应的 name(database name),username,password,host ,port 等配置参数，如需其他数据库，则修改 engine 配置为相应的引擎，相关参数请参考 django 配置官方文档。

2.同步数据库

配置好数据库连接后，在 webapp 目录下运行：

```
python manage.py syncdb
```

同步数据库，连接 mysql 登陆后即可在数据库中看到已建好的表结构
也可通过 smartclass.sql 导入到数据库中

3.表结构说明

model 和表示——对应的，所以这里仅对 model 做注释

#自定义用户类，增加了 utype 和头像

```
class MyUser(AbstractUser):
```

```
    utype = models.CharField(max_length=30)#用户类型，可为 student or teacher
```

```
    avatar = models.ImageField(upload_to='images',max_length=1000)
```

#未使用

```
class Student(models.Model):
```

```
    stud_id = models.IntegerField()
```

```
    name = models.CharField(max_length=30)
```

```
    password = models.CharField(max_length=20)
```

```
    def __unicode__(self):
```

```
        return self.name
```

#未使用

```
class Teacher(models.Model):
    teach_id = models.IntegerField()
    name = models.CharField(max_length=30)
    password=models.CharField(max_length=20)
    def __unicode__(self):
        return self.name
```

#课程类

```
class Course(models.Model):
    course_id = models.IntegerField(default=1401)#课程 id
    name = models.CharField(max_length=30,unique=True)#课程名
    teacher = models.IntegerField()#教师 id
    number_student = models.IntegerField(default=0)#加入课程的学生数目
    number_project = models.IntegerField(default=0)#课程项目数
    number_test = models.IntegerField(default=0)#课程测试数
    status = models.IntegerField(default=-1)#课程状态 0 开始 1 结束
    def __unicode__(self):
        return self.course_name
```

#学生、课程关联表

```
class stud_course_entry(models.Model):
    course = models.IntegerField()#课程 id
    student = models.IntegerField()#学生 id
```

#未使用

```
class Group(models.Model):
    group_id = models.IntegerField()
    group_name = models.CharField(max_length=30)
    group_student = models.IntegerField()
    group_course = models.IntegerField()
```

#签到表，未使用

```
class Sign(models.Model):
    student = models.IntegerField()
    course = models.IntegerField()
    time = models.DateTimeField(auto_now_add=True)
```

#答案表，未使用

```
class Answer(models.Model):
    ans_id = models.IntegerField()
    ans_student_id = models.IntegerField()
    ans_question_id = models.IntegerField()
    ans_course_id = models.IntegerField()
    ans_test_id = models.IntegerField()
    ans_text = models.CharField(max_length=100)
```

#提问表，未使用

```
class Question(models.Model):
    qst_id = models.IntegerField()
    qst_code = models.IntegerField()
    qst_content = models.CharField(max_length=100)
    qst_student = models.IntegerField()
    qst_course = models.IntegerField()
```

#测试表

```
class Test(models.Model):
    id = models.AutoField(primary_key=True)#测试 id
    tid = models.IntegerField(default=1)#未使用
    name = models.CharField(max_length=30)#测试名
    question_number= models.IntegerField(default=1)#测试问题数目
    course = models.IntegerField(default=-1)#测试所在课程
```

```
status = models.IntegerField(default=-1)#测试状态 0 未添加答案 1 未添加  
题目
```

```
#2 未开始 3 未结束 4 已结束
```

```
files = models.FileField(upload_to='static/files/test/')#题目文件
```

```
file_url = models.CharField(max_length=30)#题目文件 url
```

```
number_student = models.IntegerField(default=0)
```

```
def __unicode__(self):
```

```
    return self.name
```

```
#课程项目表
```

```
class Project(models.Model):
```

```
    pid = models.IntegerField(default=0)#未使用的 id
```

```
    name = models.CharField(max_length=30)#项目名
```

```
    course = models.IntegerField()#所在课程 id
```

```
    group_number = models.IntegerField(default=0)#所分组数
```

```
    is_grouped = models.BooleanField(default=False)#是否分组 0 未分组 1 已  
分组
```

```
    status = models.IntegerField(default=-1)#状态 0 未添加题目 1 未开始分组  
2 分组已开始 3 分组借宿
```

```
    files = models.FileField(upload_to='static/files/project')#项目上传文件
```

```
    file_url = models.CharField(max_length=30)#文件 url
```

```
    number_student = models.IntegerField(default=0)#已提交组队信息的学生  
名单
```

```
    def __unicode__(self):
```

```
        return self.name
```

```
#项目的所有题目表
```

```
class Project_Question(models.Model):
```

```
    qid = models.IntegerField(default=-1)#未使用的题目 id
```

```
    pid = models.IntegerField(default=-1)#项目 id
```

```
    name=models.CharField(max_length=30)#题目名称
```

```

max_group=models.IntegerField(default=-1)#最大组数
number_per_group=models.IntegerField(default=-1)#每组人数
files = models.FileField(upload_to = 'static/files/project')#文件
file_url = models.CharField(max_length=30)
first_number = models.IntegerField(default=0)#已该题目为第一志愿
的人数
second_number = models.IntegerField(default=0)#第二志愿人数
third_number = models.IntegerField(default=0)#第三志愿人数
def __unicode__(self):
    return self.name

```

#测试答案表

```

class Test_Answer(models.Model):
    tid = models.IntegerField(default = -1)#测试 id
    aid = models.IntegerField(default = -1)#答案编号
    answer = models.CharField(max_length=20)#答案
    accuracy =
models.DecimalField(max_digits=5,decimal_places=2,default = 0)#正确率

```

#学生测试表，记录学生提交的测试信息

```

class Student_Test(models.Model):
    tid = models.IntegerField(default=-1)#测试 id
    sid = models.IntegerField(default=-1)#学生 id
    accuracy = models.DecimalField(max_digits=5,
decimal_places=2,default=0)#正确率

```

#学生测试答案，记录学生提交的答案

```

class Student_Test_Answer(models.Model):
    tid = models.IntegerField(default=-1)#测试 id
    sid = models.IntegerField(default=-1)#学生 id
    aid = models.IntegerField(default=-1)#答案的题目编号

```

```
answer = models.CharField(max_length=10)#答案
outcome = models.IntegerField(default=0)#结果 0 错误 1 正确
```

#分组信息表，记录学生提交的分组选题信息

```
class Groupinfo_Question(models.Model):
    pid = models.IntegerField(default=0)#项目 id
    sid = models.IntegerField(default=0)#学生 id
    qid = models.IntegerField(default=0)#题目 id
    qname=models.CharField(max_length=30)#题目名称
    priority=models.IntegerField(default=-1)#优先级 1 一志愿 2 二志愿 3
    三志愿
    priority_bak = models.IntegerField(default=-1)#优先级备份
```

#分组信息，记录学生所选的搭档

```
class Groupinfo_Partner(models.Model):
    pid = models.IntegerField(default=0)#项目 id
    sid = models.IntegerField(default=0)#学生 id
    partner = models.IntegerField(default=0)#所选的同伴 id
```

#分组结果记录表

```
class Group_Outcome(models.Model):
    pid = models.IntegerField(default=0)#项目 id
    gid = models.IntegerField(default=0)#组编号
    qid = models.IntegerField(default=0)#问题 id
    qname = models.CharField(max_length=30)#问题名称
    sid = models.IntegerField(default=0)#学生 id
```

#分组结果，分组与所选题目表

```
class Group_Question_Outcome(models.Model):
    pid = models.IntegerField(default=0)#项目 id
```

```
gid = models.IntegerField(default=0)#组编号
qid = models.IntegerField(default=0)#题目 id
qname = models.CharField(max_length=30)#题目名称
```

#分组结果，学生与所在组关联表

```
class Group_Student_Outcome(models.Model):
    sid = models.IntegerField(default=0)#学生 id
    gid = models.IntegerField(default=0)#组编号
    pid = models.IntegerField(default=0)#项目 id
    sname = models.CharField(max_length=30)#学生名称
```