# 挑战作业1 创建课程作者数据表

## 挑战介绍

本次挑战作业要在上一部分的基础上创建一个 Author 映射类，并使用该映射类创建一张数据表。

#### 环境准备

先做一些准备工作，首先切换到 project 目录下创建虚拟环境：

venv

进入虚拟环境，安装所需工具包：

django

mysqlclient

基于上一部分的项目代码：code04.zip，并在IDE中打开。

启动 MySQL 服务，创建 myweb 数据库：

#### 挑战作业要求

Author 类从属于 learn 应用，所以数据表的表名应是 learn\_author 。

数据表包含如下字段：

* id 自增主键，int 类型
* name 课程作者，字符串类型，最大长度为 64 ，不可为空值
* gender 性别，布尔值类型，男性为 True，女性为 False ，可以为空值

创建好数据表之后，向表中添加三条数据：

| **ID** | **作者名字** | **性别** |
| --- | --- | --- |
| 1 | 卡卡西 | 男 |
| 2 | 娜美 | 女 |
| 3 | 宋神宗 |  |

注意第三条数据的 “性别” 字段为空。添加数据的方式不限，推荐利用映射类编写 Python 代码实现。

完成全部操作后，可以在数据库中获得如下查询结果：

SELECT \* FROM learn\_author;

+----+-----------+--------+

| id | name | gender |

+----+-----------+--------+

| 1 | 卡卡西 | 1 |

| 2 | 娜美 | 0 |

| 3 | 宋神宗 | NULL |

+----+-----------+--------+

3 rows in set (0.01 sec)

#### 涉及知识点

* Django 映射类的创建
* 数据迁移操作
* 利用映射类创建数据

## 提示信息

1、id 主键无需自定义，数据迁移时自动创建；

2、使用映射类生成数据表分两步，生成迁移文件和执行迁移；

3、字段非空约束的参数为 null=True 。

## 参考代码

向 learn/models.py 文件中增加如下代码：

class Author(models.Model):

'''课程作者映射类，包括作者名字、性别等属性

'''

name = models.CharField(max\_length=64)

gender = models.BooleanField(null=True)

def \_\_str\_\_(self):

return self.name

终端执行如下命令创建映射类的迁移文件并执行迁移：

cd myweb

python manage.py makemigrations

python manage.py migrate

执行 python manage.py shell 打开 Python Shell ：

from learn.models import Author

Author.objects.get\_or\_create(name='卡卡西', gender=1)

Out[3]: (<Author: 卡卡西>, True)

Author.objects.get\_or\_create(name='娜美', gender=0)

(<Author: 娜美>, True)

Author.objects.get\_or\_create(name='宋神宗')

(<Author: 宋神宗>, True)

# 挑战作业2 创建求和视图函数

## 挑战介绍

这次挑战的作业是在上一部分的基础上在 learn 应用中创建一个求和的视图函数，首先做一些准备工作。

打开环境，在终端获取项目代码、解压并在PyCharm中打开：

code07.zip

创建虚拟环境，安装所需工具包（如果已完成，可忽略）：

virtualenv venv

venv/Scripts/activate

pip install django

pip install mysqlclient

启动 MySQL 服务，创建数据库：

net start mysql

mysql -uroot -p

CREATE SCHEMA myweb CHARSET = UTF8

切换到项目主目录，执行数据迁移（其实就是创建数据表）：

cd myweb

python manage.py migrate

查看数据库中的数据表：

mysql -uroot -p

SHOW TABLES IN myweb

数据表中的数据与挑战无关，可以不必补充。

#### 挑战要求

在 learn 应用中增加一个函数计算 URL 中的值并显示到屏幕上。函数名为 add\_a\_b ，参数为 a 和 b ，返回两个数的和即可。

例如浏览器上的 URL 为 http://127.0.0.1:8080/learn/add/2/3 返回到屏幕上的数字就是 5 。

完成后截图如下：



#### 知识点

* 视图函数的编写
* URL 映射
* 路由转换器

## 提示信息

* 只需要修改和补充 learn 应用中的 Python 文件
* 注意视图函数的第一个参数须为 request
* 路由转换器要使用 < > 括号

## 参考代码

在 learn/views.py 文件中补充如下代码：

def add\_a\_b(request, a, b):

return HttpResponse(a+b)

将 learn/urls.py 文件修改如下：

from django.urls import path

from .views import index, courses, course, add\_a\_b

urlpatterns = [

path('', index),

path('courses', courses),

path('course/<int:id>', course),

path('add/<int:a>/<int:b>', add\_a\_b),

]

# 挑战作业3 查询作者的全部课程信息

## 挑战介绍

本次挑战作业要求实现一个页面，该页面能够查询课程信息并展示。

如有当前项目，以下步骤可以略过。

前面部分实现后的项目代码：code10.zip

创建虚拟环境，安装所需模块：

virtualenv venv

venv/Scripts/activate

pip install django

pip install mysqlclient

切换到 myweb 项目主目录，启动 MySQL 服务，创建数据库：

cd myweb

net start mysql

mysql -uroot -p

CREATE SCHEMA myweb CHARSET = UTF8

数据迁移（创建数据表）：

python manage.py migrate

进入 MySQL 客户端，添加数据：

mysql -uroot myweb

INSERT INTO `learn\_author` VALUES (1,'卡卡西',1),(2,'娜美',0),(3,'宋神宗',NULL);

INSERT INTO `learn\_course` VALUES

(1,'Linux 基础教程','2011-01-12',123,1),

(2,'Git 与 Github 入门实践','2018-12-22',678,1),

(3,'Python 异步编程','2019-04-03',59,2),

(4,'Julia 简明教程','2019-10-12',80,2),

(5,'Django 入门与实践','1983-01-05',728,2),

(6,'C++ 实现简易 Docker 容器','1974-12-02',158,3),

(7,'MySQL 基础课程','2001-02-22',333,3);

#### 挑战要求

1、在 learn/views.py 文件中增加 search 视图函数，用于处理 http://127.0.0.1:8080/learn/search 路由。

2、search 视图函数使用 render 方法返回响应数据，需要调用 learn/templates/learn/search.html 模板文件。

3、实现页面后，输入作者名字点击提交，在同一页面显示相关课程的名字、出版时间和学生人数。

4、适当考虑页面的美观。

#### 实现结果



#### 知识点

* 模板变量
* 模板标签

## 提示信息

1、在项目已有框架的基础上添加代码即可。

2、前端代码可以参考 login.html 和 courses.html 文件。

3、记得在 learn/urls.py 中补充路由与视图函数的映射。

## 参考答案

参考步骤：

1. 修改 learn/views.py

2. 新建 learn/templates/learn/search.html

3. 修改 learn/urls.py

可以参考代码：code35.zip