

Django基础

2020年7月15日 20:49

1. 创建项目

a. django-admin startproject 项目名

i. 如: django-admin startproject project01

查看项目的树形结构

i. tree . /F (查看根目录下的文件树形结构)

ii. tree ./project01 /F (查看项目project01的属性结构)

```
(base) D:\pyCharm\learn>tree ./project01 /F
卷 software 的文件夹 PATH 列表
卷序列号为 CCAB-CDBF
D:\PYCHARM\LEARN\PROJECT01
|
|_ manage.py
|
|_ project01
    |
    |_ asgi.py
    |_ settings.py
    |_ urls.py
    |_ wsgi.py
    |_ __init__.py
```

b. 目录层级说明:

- i. **manage.py**: 一个命令行工具, 可以使我们用多种方式的django项目进行交互。
- ii. **__init__.py**: 一个空文件, 它告诉Python这个目录应该被看做一个package。
- iii. **settings.py**: 项目的配置文件。
- iv. **urls.py**: url管理器, 项目的url声明。
- v. **wsgi.py**: 项目与WSGI兼容的web服务器入口。

2. 基本操作

a. 设计表结构

i. 班级表结构

1) 表名: grade

2) 字段:

- a) 班级名称 (gname)
- b) 成立时间 (gdate)
- c) 女生总数 (ggirlnum)
- d) 男生总数 (gboynum)
- e) 是否删除 (isdelete)

ii. 学生表结构

1) 表名: students

a) 字段:

- i) 学生姓名 (sname)
- ii) 学生性别 (sgender)
- iii) 学生年龄 (sage)
- iv) 学生简介 (scontent)
- v) 所属班级 (sgrade)
- vi) 是否删除 (sdelete)

b. 配置数据库 (django默认使用sqlite数据库)

i. 在settings.py文件中, 通过DATABASES选项进行数据库的配置。

ii. 配置MySQL

1) python3.X安装的是PyMySQL

2) 在__init__.py中写入两行代码

```
import pymysql
pymysql.install_as_MySQLdb()
```

iii. 配置DATABASES:

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'py0715',
        'USER': 'root',
        'PASSWORD': 'root',
        'HOST': 'localhost',
        'PORT': '3306',
    }
}

```

iv. 创建应用（在一个项目中可以创建多个应用，每个应用进行一种业务处理）

- 1) 打开黑屏终端，进入项目目录。
- 2) 执行 `python manage.py startapp myapp01`
- 3) myapp01 目录说明：
 - a) admin.py: 站点配置
 - b) models.py: 模型
 - c) views.py: 视图

v. 激活应用

- 1) 在 settings.py 文件中，将 myapp01 应用加入到 `INSTALLED_APPS` 的选项中。

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp01'
]

```

vi. 定义模型

- 1) 概述：有一个数据表，就对应一个数据表。
- 2) 在 model.py 中定义模型
 - a) 引入 models（模型类要继承 models.Models）
 - i) `from django.db import models`
 - b) 创建模型类
 - c) 说明：不需要定义主键，主键在生成时自动添加，并且值为自动增加。

vii. 生成数据表

- 1) 生成迁移文件
 - a) `python manage.py makemigrations`

```

(base) D:\pyCharm\learn\project01>python manage.py makemigrations
Migrations for 'myapp01':
  myapp01\migrations\0001_initial.py
    - Create model Grades
    - Create model Student

```

- b) 在 migrations 目录下生成迁移文件，此时数据库中没有生成数据表。

2) 执行迁移（生成数据表）

- a) `python manage.py migrate`：相当于执行 sql 语句创建数据表

viii. 测试数据操作

- 1) 进入到 python shell
 - a) 执行 `python manage.py shell`
 - b) 引入包
 - i) `from myapp01.models import Grades, Students`
 - ii) `from django.utils import timezone`
 - iii) `from datetime import *`

```
In [1]: from myapp01.models import Grades, Student

In [2]: from django.utils import timezone

In [3]: from datetime import *
```

c) 查询所有数据

i) 类名.objects.all()

```
In [4]: Grades.objects.all()
Out[4]: <QuerySet []>

In [5]: Student.objects.all()
Out[5]: <QuerySet []>
```

d) 插入数据

i) 创建对象: g02=Gradel()

```
In [14]: g02.gname='追求卓越'

In [15]: g02.gdate=date(year=2020,month=7,day=16)

In [16]: g02.ggirlnum=22
```

ii)

```
In [17]: g02.save()
F:\users\lenovo\Anaconda3\lib\site-packages\django\db\models\fields\__init__.py:1310: RuntimeWarning:
atetime (2020-07-16 00:00:00) while time zone support is active.
RuntimeWarning)
```

e) 查看某个对象 (查找)

i) 对象名.objects.get(pk=索引)

One. 如: Grades.objects.get(pk=2)

f) 修改数据

i) 修改: 模型对象.属性 = 新值

ii) 保存: 模型对象.save()

g) 删除数据 (物理删除)

i) 模型对象.delete()

ii) 如:

```
In [9]: g.delete()
One. Out[9]: (1, {'myapp01.Student': 0, 'myapp01.Grades': 1})
```

h) 关联对象:

i) 获取关联对象的集合

One. 如: 获取一年级的所有学生

```
In [30]: g1=Grades.objects.get(pk=1)
```

First.

```
In [31]: g1.student_set.all()
Out[31]: <QuerySet [<Student: Student object (1)>, <Student: Student object (2)>]>
```

ii) 创建赵四

One. In [35]: s4 = g1.student_set.create(sname=u'赵四',sgender=True,scontend=u"我是三啥子",sage=44)

Two. 不用保存直接可以插入数据库