

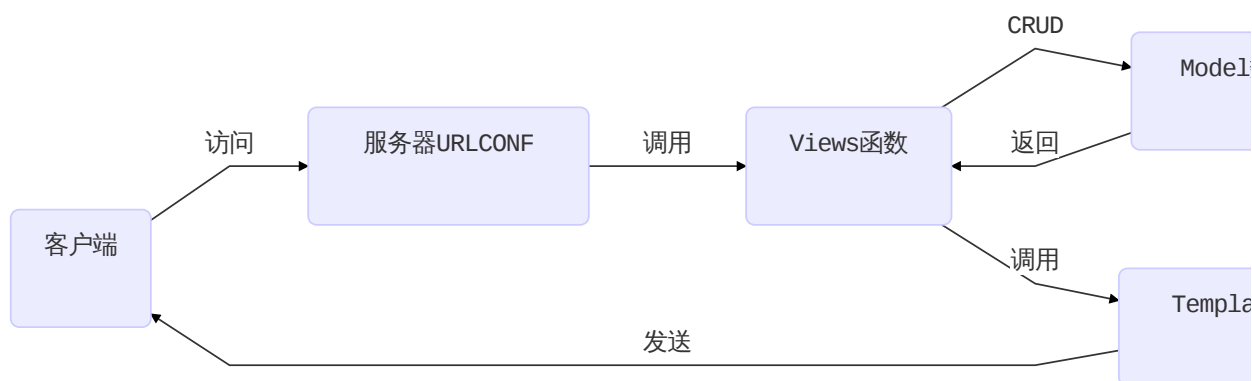
nsd_1908_devweb_day03

常用的web框架

- django
- flask
- tornado

MTV模式

- Model：模型，对应数据库
- Template：模板，对应web页面
- View：视图，对应函数



在虚拟环境中安装django

本地安装

```
(nsd1908) [root@room8pc16 day03]# pip install /var/ftp/pub/zzg_pypkgs/dj_pkgs/*
```

在线安装

```
(nsd1908) [root@room8pc16 day03]# pip install django==
```

```
(nsd1908) [root@room8pc16 day03]# pip install django==1.11.6
```

创建项目

- 一个项目由很多文件构成，一般将这些文件统一放到一个目录中

```
# 创建项目，方法一：
(nsd1908) [root@room8pc16 day03]# django-admin startproject mytest
(nsd1908) [root@room8pc16 day03]# ls
mytest
```

```
# 创建项目，方法二：pycharm
# File -> New Project -> 左窗格点django，右窗格输入项目目录/xxx/mysite
```

```
(nsd1908) [root@room8pc16 mysite]# pwd
/var/ftp/nsd2019/nsd1908/devweb/day03/mysite # 项目的根目录
(nsd1908) [root@room8pc16 mysite]# tree .
```

```
.
├── manage.py          # 项目管理程序
├── mysite              # 项目配置目录
│   ├── __init__.py    # 初始化文件
│   ├── settings.py    # 项目的配置文件
│   ├── urls.py        # urlconf路由文件
│   └── wsgi.py        # 部署项目到服务器的配置文件
└── templates          # 模板目录，存放网页的目录
```

启动开发服务器。django为方便程序员看到代码效果，它内置了一个测试服务器，该服务器只能用于开发环境，不能用于生产环境。

```
(nsd1908) [root@room8pc16 mysite]# python manage.py runserver
# 访问 http://127.0.0.1:8000
```

修改配置

```
# 创建数据库
[root@room8pc16 ~]# mysql -uroot -ptedu.cn
MariaDB [(none)]> CREATE DATABASE dj1908 DEFAULT CHARSET utf8;
```

```
# mysite/settings.py
ALLOWED_HOSTS = ['*'] # 允许任意主机访问本机的任何地址
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'dj1908',
        'USER': 'root',
        'PASSWORD': 'tedu.cn',
        'HOST': '127.0.0.1',
        'PORT': '3306',
    }
}
LANGUAGE_CODE = 'zh-hans'
TIME_ZONE = 'Asia/Shanghai'
USE_TZ = False

# 配置到mysql的连接
import pymysql

pymysql.install_as_MySQLdb()
```

```
# 重新运行开发服务器，运行在0.0.0.0的80端口
(nsd1908) [root@room8pc16 mysite]# python manage.py runserver 0:80

# 生成后台数据库表
(nsd1908) [root@room8pc16 mysite]# python manage.py makemigrations
(nsd1908) [root@room8pc16 mysite]# python manage.py migrate

# 创建管理员用户
(nsd1908) [root@room8pc16 mysite]# python manage.py createsuperuser
Username (leave blank to use 'root'): admin
Email address: admin@tedu.cn
Password: 1234.com

# 访问后台管理页面：http://127.0.0.1/admin
```

创建应用

- 应用就是一个功能模块
- 每个应用对应一个目录
- 一个项目可以包一到多个应用
- 一个应用也可以应用到多个项目

投票应用

规划

- <http://127.0.0.1:8000/polls/>：投票首页，显示所有的投票项
- <http://127.0.0.1:8000/polls/1/>：1号问题投票详情，可以投票
- <http://127.0.0.1:8000/polls/1/result/>：1号问题投票结果页

```
# 创建应用
(nsd1908) [root@room8pc16 mysite]# python manage.py startapp polls
(nsd1908) [root@room8pc16 mysite]# ls
db.sqlite3  manage.py  mysite  polls  templates

# 将应用加入到mysite项目
# mysite/settings.py
INSTALLED_APPS = [
    ...
    'polls',
]
```

授权：将应用的url交给应用处理

```
# 以polls开头的url交给polls应用处理
# mysite/urls.py
from django.conf.urls import url, include
from django.contrib import admin
```

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    # 从http://x.x.x.x/之后开始匹配
    url(r'^polls/', include('polls.urls')),
]

# 创建polls/urls.py
from django.conf.urls import url

urlpatterns = []
```

以上内容为创建项目、创建应用、授权的通用内容。创建任何项目、应用都以此方法进行。

编写应用的各个页面

编写首页

```
# 为首页指定url
# polls/urls.py
from django.conf.urls import url
# from polls import views # 也可以写为以下形式
from . import views

urlpatterns = [
    # 从http://x.x.x.x/polls/后面开始匹配
    # 访问首页，将会调用views.index函数
    # 为http://x.x.x.x/polls/这个url起名为index
    url(r'^$', views.index, name='index'),
]

# 编写视图函数
# polls/views.py
from django.shortcuts import render

# Create your views here.
# 每个函数至少需要有一个参数，用于接收客户端发来的请求
def index(request):
    # render用于查找模板文件，并返回给用户
    return render(request, 'index.html')

# 编写模板网页文件
# templates/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>投票首页</title>
</head>
<body>
<h1>投票首页</h1>
</body>
```

```
</html>
```

制作投票详情页

```
# polls/urls.py
from django.conf.urls import url
# from polls import views # 也可以写为以下形式
from . import views

urlpatterns = [
    # 从http://x.x.x.x/polls/后面开始匹配
    # 访问首页，将会调用views.index函数
    # 为http://x.x.x.x/polls/这个url起名为index
    url(r'^$', views.index, name='index'),
    # 注意正则中后面的/，务必写上
    # \d+匹配数字，用()括起来，它就会自动作为参数传给detail函数
    url(r'^(\d+)/$', views.detail, name='detail'),
]

# polls/views.py
... ..
def detail(request, question_id):
    # 字典的内容，将会被转义成key=val的形式，发送给detail.html
    return render(request, 'detail.html', {'question_id': question_id})

# templates/detail.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>投票详情</title>
</head>
<body>
<h1>{{ question_id }}号问题投票详情</h1>
</body>
</html>
```

制作投票结果页

```
# polls/urls.py
... ..
    url(r'^(\d+)/result/$', views.result, name='result'),
... ..

# polls/views.py
... ..
def result(request, question_id):
    return render(request, 'result.html', {'question_id': question_id})

# templates/result.html
<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>投票结果</title>
</head>
<body>
<h1>{{ question_id }}号问题投票结果</h1>
</body>
</html>
```

ORM

- 对象关系映射
- 将数据库中的表和class映射
- 将表中的记录与class的实例映射
- 表中的每个字段与类变量映射
- 数据库中用到的数据类型，django已定义好对应的class

投票应用涉及到的字段

- 问题、发布时间、选项、选项票数
- 问题表：问题、发布时间
- 选项表：选项、票数、问题ID

```
# polls/models.py
from django.db import models

# Create your models here.
class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField()

# 生成表
(nsd1908) [root@room8pc16 mysite]# python manage.py makemigrations
(nsd1908) [root@room8pc16 mysite]# python manage.py migrate
MariaDB [dj1908]> show tables;
# 表名结构：应用名_class名 全部小写
... ..
| polls_question          |
... ..
MariaDB [dj1908]> desc polls_question;
# class中没有明确声明主键，django自动创建名为id的主键；类变量成为表的字段

# polls/models.py
... ..
class Choice(models.Model):
    choice_text = models.CharField(max_length=50)
    votes = models.IntegerField(default=0)
    q = models.ForeignKey(Question)

# 生成表
```

```
(nsd1908) [root@room8pc16 mysite]# python manage.py makemigrations
(nsd1908) [root@room8pc16 mysite]# python manage.py migrate
MariaDB [dj1908]> show tables;
... ..
| polls_choice |
MariaDB [dj1908]> desc polls_choice;
# 注意, 在Choice类中的q是外键, 数据库表中自动创建一个名为q_id的字段

# 修改q为question, 数据库中表的外键名将成为question_id
class Choice(models.Model):
    choice_text = models.CharField(max_length=50)
    votes = models.IntegerField(default=0)
    question = models.ForeignKey(Question)

(nsd1908) [root@room8pc16 mysite]# python manage.py makemigrations
Did you rename choice.q to choice.question (a ForeignKey)? [y/N] y
(nsd1908) [root@room8pc16 mysite]# python manage.py migrate
```

将模型注册到管理后台

```
# polls/admin.py
from django.contrib import admin
# from polls.models import Question, Choice # 也可以写为
from .models import Question, Choice # 从当前目录的models模块中导入

# Register your models here.
admin.site.register(Question)
admin.site.register(Choice)

# 访问管理后台 http://x.x.x.x/admin。创建的问题, 显示的是Question object。如果创建选项, 选项显示为
Choice object。可以使用以下办法更正:
from django.db import models

# Create your models here.
class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField()

    def __str__(self):
        return "问题: %s" % self.question_text

class Choice(models.Model):
    choice_text = models.CharField(max_length=50)
    votes = models.IntegerField(default=0)
    question = models.ForeignKey(Question)

    def __str__(self):
        return "%s=> %s" % (self.question, self.choice_text)

# 再次访问管理后台
```

