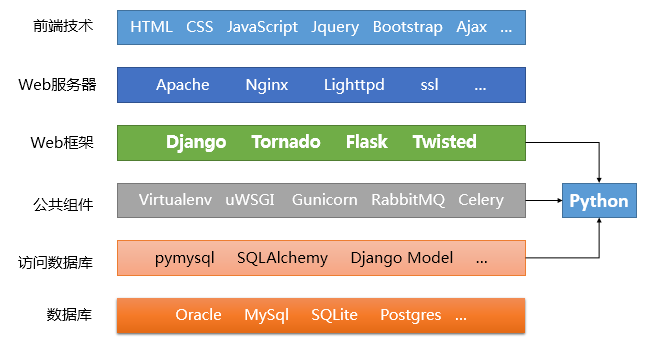
# Web开发

**基于Python的Web开发技术栈：**



什么是Web开发？

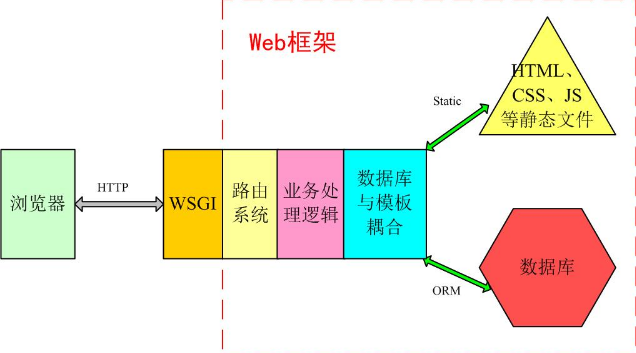
Web开发指的是开发基于Browser/Server架构，通过前后端的配合，将后台服务器的数据在浏览器上展现给前台用户的应用。

Web框架

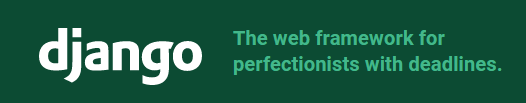
Web框架致力于解决一些共同的问题，为Web应用提供通用的架构，让用户专注于网站应用业务逻辑的开发，而无须处理网络应用底层的协议、线程、进程等方面的问题，从而大大提高开发者的效率和Web应用程序的质量。

1. 开始编写第一个Web应用，经过大量地摸索和踩坑，完成了工作。
2. 开始编写新的Web应用
3. 从第一步中总结经验（找出其中通用的代码），并运用在第二步中
4. 重构代码使得能在第二个应用中使用第一个程序中的通用代码
5. 重复2-4步若干次
6. 发明了一个Web框架。

最初的Web开发框架就是这么来的！



## Django



Django是一个由Python编写的具有完整架站能力的开源Web框架。使用Django，只要很少的代码，开发人员就可以轻松地完成一个正式网站所需要的大部分内容，并进一步开发出全功能的Web服务。

Django本身基于MVC架构，即Model（模型）+View（视图）+ Controller（控制器）设计模式，因此天然具有MVC的出色基因：开发快捷、部署方便、可重用性高、维护成本低等优点。

Django诞生于2003年，2006年加入了BSD许可证，成为开源的Web框架。Django这一词语是根据比利时的爵士音乐家Django Reinhardt命名的，含有希望Django能够优雅地演奏（开发）各种乐曲（Web应用）的美好含义

## MVC和MTV设计模式：

在目前基于Python语言的几十个Web开发框架中，几乎所有的全栈框架都强制或引导开发者使用MVC设计模式。

**MVC设计模式：**

最早由Trygve Teenskaug在1978年提出，上世纪80年代是程序语言Smalltalk的一种内部架构。后来MVC被其他领域借鉴，成为了软件工程中的一种通用架构模式。MVC把Web框架分为三个基础部分：

**模型(Model)**：用于封装与应用程序的业务逻辑相关的数据及对数据的处理方法，是Web应用程序中用于处理应用程序的数据逻辑的部分，Model只提供功能性的接口，通过这些接口可以获取Model的所有功能。白话说，这个模块就是业务逻辑和数据库的交互层，定义了数据表。

**视图(View)**：负责数据的显示和呈现，是对用户的直接输出。

**控制器(Controller)**：负责从用户端收集用户的输入，可以看成提供View的反向功能。

这三个部分互相独立，但又相互联系，使得改进和升级界面及用户交互流程，在Web开发过程任务分配时，不需要重写业务逻辑及数据访问代码。

**MTV设计模式：**

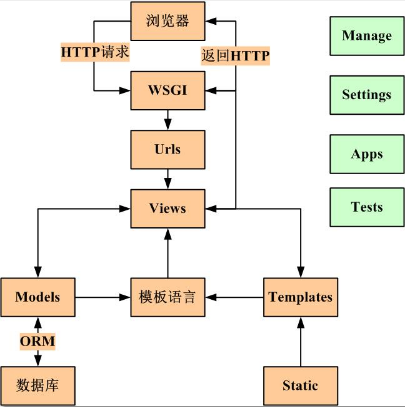
MTV和MVC本质上是一样的。

Django对传统的MVC设计模式进行了修改，将视图分成View模块和Template模块两部分，将动态的逻辑处理与静态的页面展示分离开。而Model采用了ORM技术，将关系型数据库表抽象成面向对象的Python类，将数据库的表操作转换成Python的类操作，避免了编写复杂的SQL语句。

**模型(Model)**：和MVC中的定义一样

**模板(Template)**：将模型数据与HTML页面结合起来的引擎

**视图(View)**：负责实际的业务逻辑实现



#WSGI全称Python Web Server Gateway Interface，指定了web服务器和Python web应用或web框架之间的标准接口，以提高web应用在一系列web服务器间的移植性。

# 编写一个Django 应用

1. Install django software

|  |
| --- |
| pip 3 install django  mkdir django  django-admin startproject mysite  C:\Users\Administrator\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\4F691310.tmp |

最外层的 **mysite/** 根目录只是你项目的容器， 根目录名称对 Django 没有影响，你可以将它重命名为任何你喜欢的名称。

**manage.py**: 一个让你用各种方式管理 Django 项目的命令行工具。

里面一层的 **mysite/** 目录包含你的项目，它是一个纯 Python 包。它的名字就是当你引用它内部任何东西时需要用到的 Python 包名。 (比如 **mysite.urls**).

**mysite/\_\_init\_\_.py**：一个空文件，告诉 Python 这个目录应该被认为是一个 Python 包。

**mysite/settings.py**：Django 项目的配置文件。

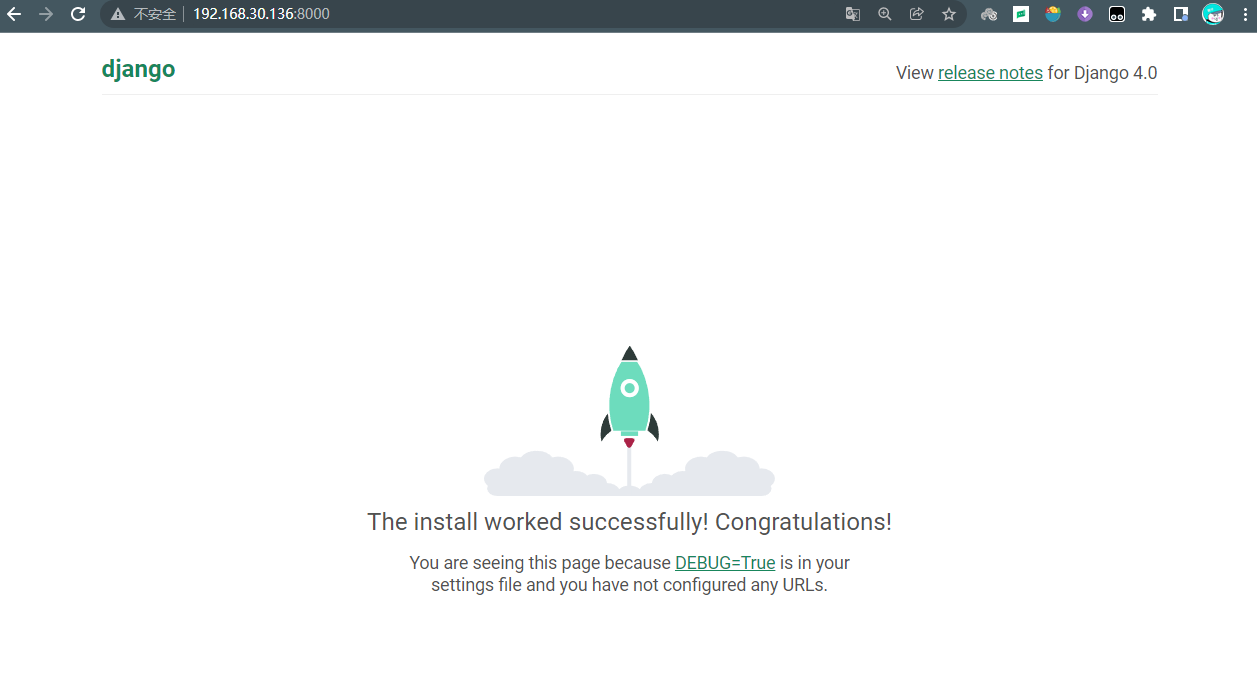
**mysite/urls.py**：Django 项目的 URL 声明，就像你网站的“目录”。

**mysite/asgi.py**：作为你的项目的运行在 ASGI 兼容的 Web 服务器上的入口。

**mysite/wsgi.py**：作为你的项目的运行在 WSGI 兼容的Web服务器上的入口。

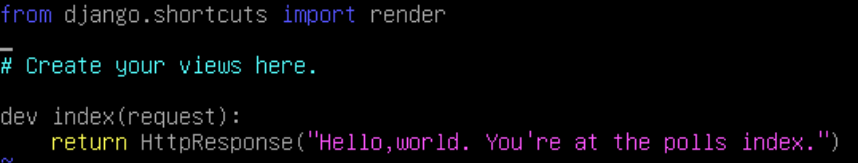
1. 运行测试

|  |
| --- |
| #/django/mysite/mysite/settings.py    Python3 manage.py runserver |

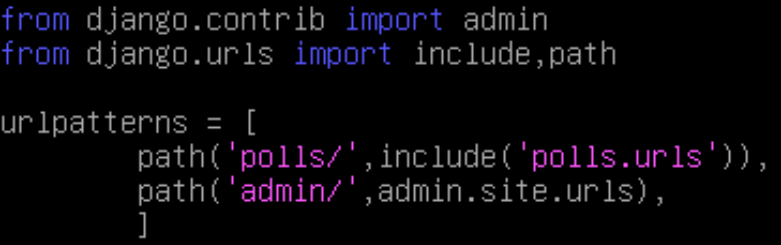


编写视图

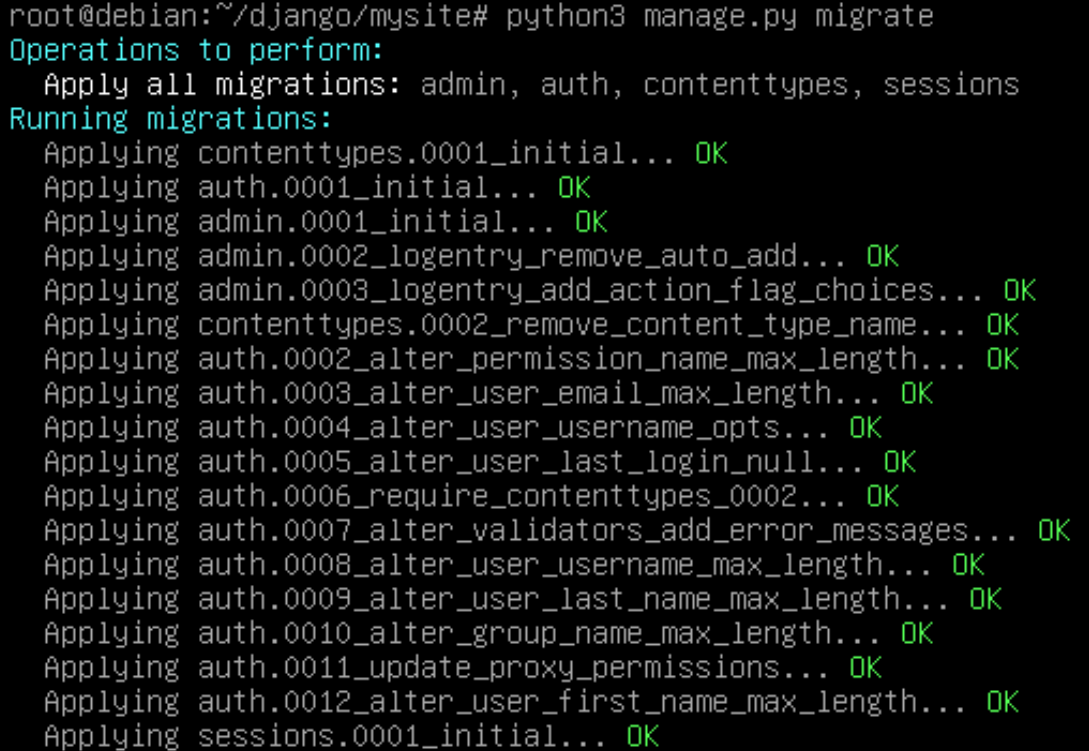
#/django/mysite/polls/views.py



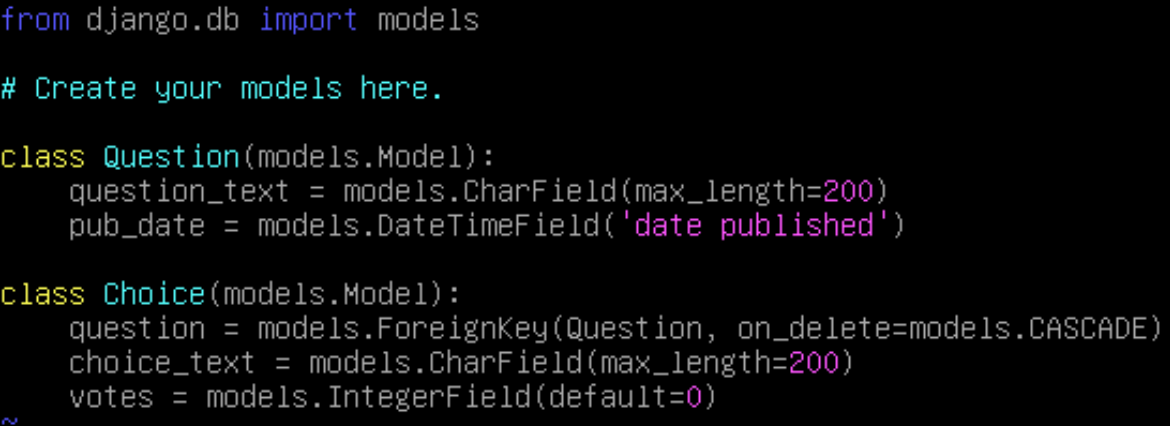
#创建文件/django/mysite/polls/urls.py



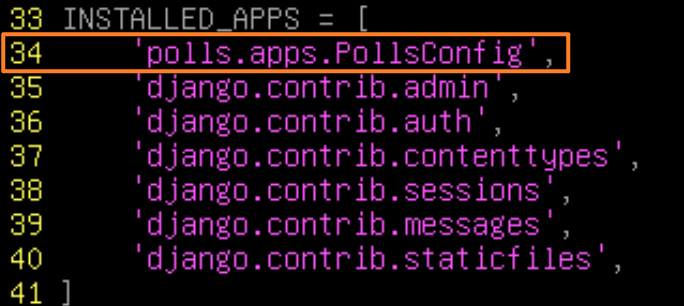
数据库配置

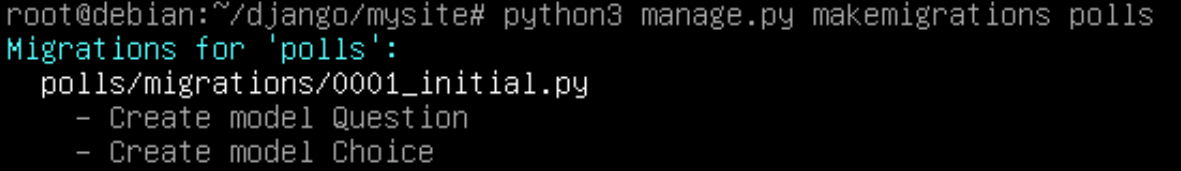


#/django/mysite/polls/Models.py

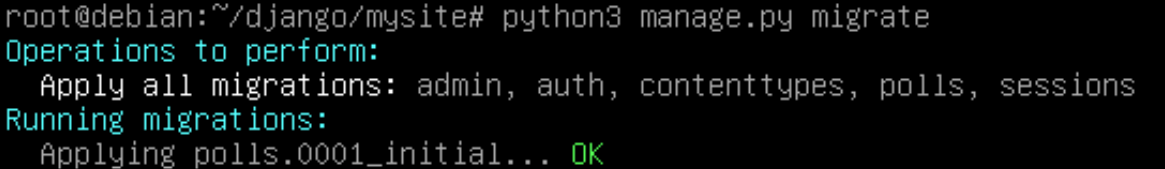


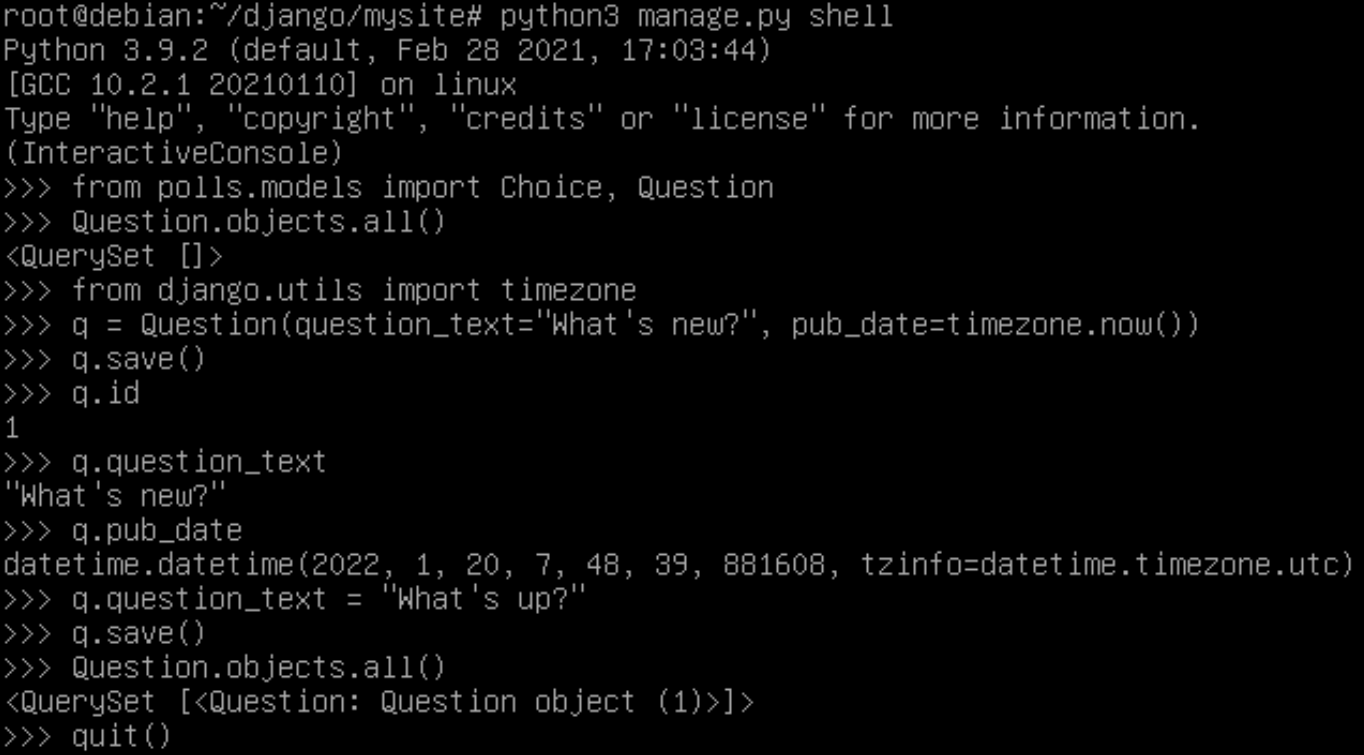
#/django/mysite/mysite/settings.py











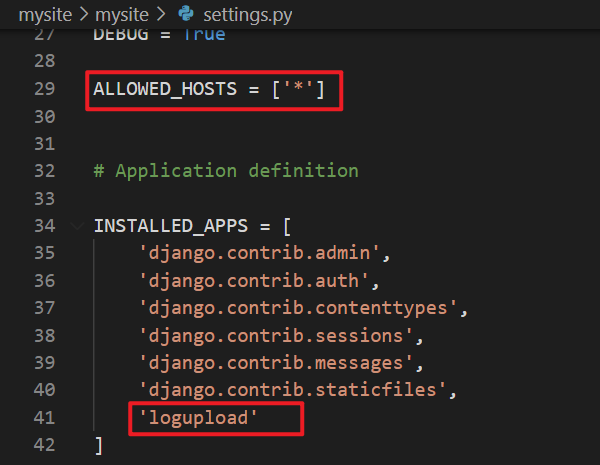
# Logupload 项目

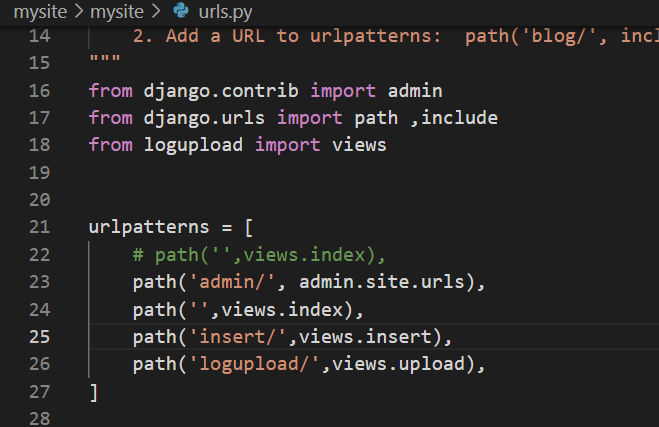
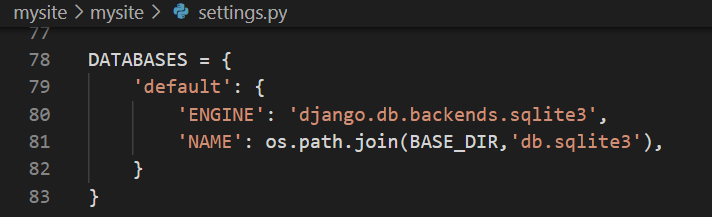
A. 启动项目

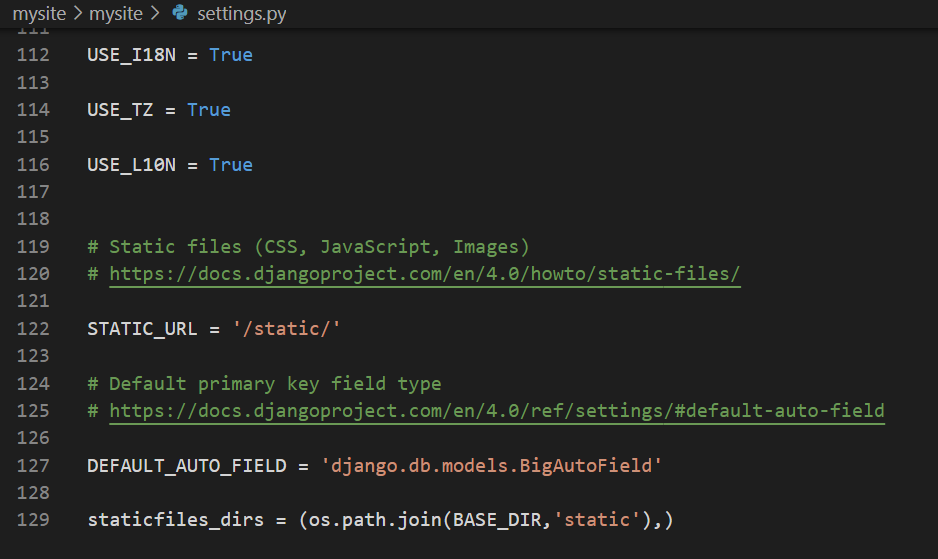
root@PythonDev:/Python-Dev/django# django-admin startproject mysite

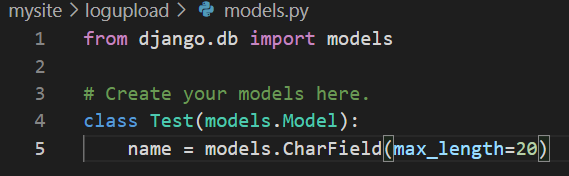
root@PythonDev:/Python-Dev/django/mysite# python3 manage.py runserver

root@PythonDev:/Python-Dev/django/mysite# python3 manage.py startapp logupload







root@PythonDev:/Python-Dev/django/mysite# python3 manage.py makemigrations logupload

root@PythonDev:/Python-Dev/django/mysite# python3 manage.py migrate

查看db.sqlite3数据库文件

apt install sqlite3

sqlite3 db.sqlite3 //进入数据库

sqlite> .tables //查看表

sqlite> select \* from logupload\_test; //查看表中数据

sqlite>.quit //退出数据库

#mysite>logupload>views.py

|  |
| --- |
| from django.shortcuts import render  from django.http import HttpResponse  from random import sample  from logupload.models import Test  import os  # Create your views here.  def index(request):      # return HttpResponse('This is the test page')      Net\_list = ['user1','user2','user3','user4','user5','user6','user7','user8','user9','user0']      random\_select = sample(Net\_list,1)[0]      return render(request,'index.html',{'Name':random\_select})  def upload(request):      if request.method == 'GET':          return render(request,'upload.html')      elif request.method == 'POST':          title = request.POST.get('title')          username = request.POST.get('username')          role = request.POST.get('role')          date = request.POST.get('date')          date = '.'.join(date.split('-'))          upload = request.FILES.get('upload')          if upload == None:              content = '<h1> file has not selected</h1>'              return HttpResponse(content)          filename = '{}({})-{}-{}.docx'.format(title,role,username,date)          filedir = os.path.join(os.getcwd(),'file\_storage')          target\_filename = os.path.join(filedir,filename)          with open(target\_filename,'wb') as f:              for chunk in upload.chunks():                  f.write(chunk)          file\_data = os.listdir(filedir)          content = "<h1>This is the upload filelist </h1>"          content = "<hr>"          content = "<ul>"          for x in file\_data:              content += '<li>{}</li>'.format(x)          content += "</ul>"          return HttpResponse(content)  def insert(request):      test1 = Test(name='Django Models')      test1.save()      return HttpResponse('<p>date add sucess.</p>') |

# Mysite>logupload>templates>upload.html

|  |
| --- |
| {% load static %}  <!DOCTYPE html>  <html>  <head>      <meta charset="UTF-8">      <link rel="stylesheet" href="{%static 'style.css'%}"/>      <title>upload</title>  </head>  <body>      <h1>日志上传</h1>      <h3><div id="time" align='center'></div></h3>      <script>          const $time = document.getElementById('time')          setInterval(() => {              $time.innerText = `Current time:  ${new Date().toLocaleString().replace('T',' ').substr(0,19)}`}, 1000)      </script>      <div class="log-submit">          <form method="post" enctype="multipart/form-data">              {% csrf\_token %}              <label for="log-title"><span class="form-title">标题: </span>                  <select id="log-title" name='title'>                      <option value="第46届世界技能大赛网络系统管理项目精英班训练日志">第46届世界技能大赛网络系统管理项目精英班训练日志</option>                      <option value="第46届世界技能大赛网络系统管理项目中国集训队集训日志">第46届世界技能大赛网络系统管理项目中国集训队集训日志</option>                  </select>              </label>              <br>              <label><span class="form-title">姓名: </span><input type="text" name="username" placeholder="你的名字..."></label>              <br />              <label><span class="form-title">角色: </span></label>              <label><input type="radio" name="role" value="教练">教练</label>              <label><input type="radio" name="role" value="选手">选手</label>              <br />              <label><span class="form-title">日期: <input type="date" name="date"></label>              <br />              <label><span class="form-title">上传: </span><input type="file" name="upload"></label>              <br />              <input type="submit" value="提交">              <input type="reset" value="重置">          </form>      </div>  </body>  </html> |

数据库

