django+mysql的web项目

工具准备：

python，pycharm，django，mysql，git全部安装并能正常使用

git使用：

上传本地代码：

1. 在包含git文件的目录里右键git bash here
2. git add .
3. git commit -m “描述”
4. git push

同步远程代码到本地：

1. 在包含git文件的目录里右键git bash here
2. git pull

搭建环境：

1. 创建一个文件夹，cmdcd到文件夹创建虚拟环境：python -m venv wlenv（名字）
2. 激活虚拟环境：wlenv\Scripts\activate
3. 安装django：pip install django
4. 创建django项目：django-admin startporject worklog（名字）
5. 同步数据库：python manage.py migrate
6. 测试django环境：python manage.py runserver（可选ip+端口号）
7. 切库，在mysql创建一个数据库eg：worklog\_db设置字符集为utf-8
8. 切库，安装mysqlclient：pip install mysqlclient
9. 切库，配置settings文件：在DATABASES里将engine数据库改为mysql，填上数据库名字，用户名和密码。
10. 切库，同步数据库：python manage.py migrate
11. 切库，查看数据库中是否创建了表格，runserver服务查看是否切库成功
12. 创建项目app：python manage.py startapp worklogweb（django前端）
13. 配置settings文件，把新创建的APP加入INSTALLED\_APP中
14. 修改语言，settings文件里最后LANGUAGE\_CODE:’zh-Hans’
15. 修改时区，settings文件最后TIME\_ZONE:中国范围内有Asia/Shanghai, Asia/Harbin, Asia/Chongqing, Asia/Urumqi, Asia/Kashgar

web类项目概述

web标准框架：MVC（model，view，controller）

model 模型，控制数据库表结构

view 视图，负责具体前端输出的内容，eg颜色，文字，表单

controller 控制器，具体负责实现的功能，eg增删改查，加减乘除

Router路由

用户访问地址的地址，通过router映射到controller的功能

Template渲染模板

用户将要看的页面，html

django框架：MTV（model，template，view）对应MVC，只是名字不同



用户发出访问请求—>django router（URL）将请求匹配path路径到views里的具体函数—>view根据被匹配的函数执行代码—>可能执行model调取数据库内的数据—>将所有数据渲染成模板发送到用户的浏览器展示

项目文件

urls.py

HTTP请求进入django时有限调用该文件，文件里包含所有路由，第一个参数是访问的地址，第二个参数是要调用的views.py里的映射文件。

settings.py

项目配置文件

wsgi.py

正式启用django项目时配置该网关文件

views.py

文件包含调用函数，让程序调用template里的html页面渲染并发送

setting.py

公有配置

BASE\_DIR=os.path(...)项目绝对路径，注册使用别的文件夹或文件时会用到

debug

启动模式，True为调试模式

1. 检测代码改动后立刻重启服务
2. 报错页面

False为正式模式

ALLOWED\_HOSTS=[ ]

请求头HOST头，只接受请求头包含其中的请求，有过滤作用

局域网内其他主机访问django项目服务

1. 启动方式为

python manage.py runserver 0.0.0.0:8000 （四个0代表服务器所有网卡IP都能被访问）

1. 添加服务器内网IP至ALLOWED\_HOSTS=[内网IP]

INSTALLED\_APPS

配置django应用

MIDDLEWARE

中间件，高级应用

ROOT\_URLCONF

配置主路由文件

自定义配置

1. 大写
2. 个性化，不要与公有配置重名

URL

定义：统一资源定位符

作用：用来表示互联网上某个资源的地址

一般语法格式为：protocol://hostname[:port]/path[?query][#fragment] [ ]内可省略，共6大部分。

1. protocol（协议）指使用的什么协议
2. http通过HTTP访问该资源
3. https通过安全的HTTPS访问该资源
4. file资源是本地计算机上的文件。格式file:///
5. bostname（主机名）是指存放资源的服务器的域名系统（DNS）主机名、域名或IP
6. port（端口号）

1）整数、可选（范围为1024—65535，用户可用的端口号）

1. path（路由地址）

由0或多个“/”符号隔开的字符串，一般用来表示主机上的一个目录或文件地址。路由地址决定了服务器端如何处理这个请求

1. query（查询）以问号开头，问号后都是查询字符串

可选，用于给动态网页传递参数，可有多个参数，用“&”符号隔开，每个参数名和值用“=”隔开

1. fragment（信息片段，锚点）以#号开始，类似书签，前端技术，输入带锚点的地址浏览器将直接定位到锚点所在地方

处理URL请求

eg：浏览器地址栏输入http://127.0.0.1:8000/page/2003/

1. django从settings.py文件中根据ROOT\_URLCONF找到主路由，默认情况下该文件在项目同名目录下的urls.py
2. django加载主路由文件中的urlpatterns变量：一个包含很多path的数组
3. 依次匹配urlpatterns中的path，匹配到第一个合适的path后中断后续匹配
4. 匹配成功—调用对应视图函数处理请求，返回响应
5. 匹配失败—返回404响应

视图函数

用于接收浏览器请求（HttpResquest对象）并通过HttpResquest对象返回响应的函数。此函数可以接收浏览器请求并根据业务逻辑返回响应的内容给浏览器

语法

def XXX\_view（request[,其他参数...]）:

return HttpResponse对象

路由配置

path（）函数

导入：from django.urls import path

语法：path（route, views,name=None）

参数：

1. route：字符串类型，匹配请求路径
2. views：指定路径所对应的试图处理函数的名称
3. name：为地址起别名，在模板中地址反向解析时使用

path转换器

语法：<转换器类型：自定义名>

作用：若转换器类型匹配到对应类型的数据，则将数据按照关键字传参的方式传递给试图函数

eg：path(‘page’/<int:page>’, views.xxx)

转换器类型：

1. str匹配除了/之外的非空字符串
2. int匹配0或任何正整数，返回一个int
3. slug匹配任意由ASCII字母或数字以及连字符和下划线组成的短标签
4. path匹配非空字段，包括路径分隔符

views.py里的函数要指定自定义的转换器的名字，相当于传参进来

eg：def pagex(request, page):

re\_path（）函数（精准匹配）

在url的匹配过程中可以施工正则表达式

语法：re\_path(reg, view, name=xxx) 正则表达式为命名分组模式(?P<name>pattern);匹配提取参数后用关键字传参的方式传递给视图函数