**Mini\_Web 项目总结**

1. 项目背景概述

项目时间：2018.7

参与人员：Python09全体

项目要求：模拟设计简单的web服务器，设计后端frame，通过HTTP协议、WSGI协议实现客户端、服务器、后端框架之间的动态数据请求和响应。

1. 过程概要

历经简单的静态HTTP服务器搭建，到简单的HTTP服务器返回动态请求，再到将后端框架和服务器进行深度解耦，对于动态请求完全由框架来进行数据处理，包括后端框架通过数据库进行相关操作，最后将数据结果返回给服务器，由服务器完成拼装后返回客户端，完成请求和响应的全过程。

1. 业务逻辑

3.1浏览器作为客户端，通过HTTP协议向服务器发送请求

3.2服务器解析请求，判断请求的是静态资源还是动态资源；如果是静态资源则从硬盘存储中取出数据，返回给浏览器；如果是动态资源则需要通过WSGI协议和框架进行交互，框架通过链接数据库等操作，处理好数据后，将响应头和响应体发送给服务器，服务器通过拼装完成然后返回给客户端

3.3客户端收到服务器响应的数据，通过自己的渲染正确显示。

1. 框架结构
2. 服务器改成面向对象
   1. 通过类的封装，将创建套接字和其它属性设置放入\_\_init\_\_方法中
   2. 简化mian函数，只让main函数执行主体功能。
3. 添加支持动态请求

2.1 约定以.py结尾的请求是动态请求，使用字符串的endwith方法来判断；

2.2 如果是动态请求，则模拟简单的数据处理，作为请求体返回。

1. web框架处理动态

3.1 将动态请求的处理部分代码放到一个新的py文件中，通过导包来调用准框架中的函数进行数据处理。

1. web框架服务器 深度解耦 支持WSGI

4.1 按照WSGI协议的要求，严格设置对应的函数，以及设置函数的参数来配置WSGI的API。

4.2 服务器端的两个函数分别是：environ：一个包含所有HTTP请求信息的dict对象；start\_response：一个发送HTTP响应的函数(引用)。

4.3 分为两步，传递header和body，这里也是解耦的一种体现(信息+数据分开)，header信息部分可能是服务器独有的信息，需要加载给header并传递给frame，再由框架加载其它信息制成最终的header，最后return到数据。

1. web框架 添加模板

5.1 将不同类型的文件放入不同的文件夹，也是解耦的一种思想；

5.2 需要注意的是路径问题：所有的路径都是以当前运行的文件的路径为准。

1. 添加配置文件 写shell脚本

6.1 进一步解耦，为服务器预留接口，可以让用户去根据框架和网络信息来配置。

1. 带参数的装饰器—实现路由功能
   1. if -> 字典
   2. 使用装饰器带参数，提高代码的重用性，在添加功能的时候，不用挨个去配置。
2. 支持伪静态 html

8.1 统一请求的后缀。

1. mini\_web框架添加mysql支持，

9.1 让框架链接数据库，真正实现大量数据的查询功能。

9.2 设置模块中的显示功能，让从数据库查询的数据结合前端模板，格式化的显示；

1. 路由支持正则

10.1让业务逻辑多个请求调用同一个函数

1. 增删改
   1. 实现数据的增删改
2. url编码和日志

调用urllib.parse.quote和urllib.parse.unquote方法来处理url编码问题.