Web Server a Small Journey

Part 1

Every client and webserver communicate in terms of HTTP protocols. In order to do that they have to make a perfect http handshake first. In order to achive that they first need to establish a TCP/IP connection between a client and server. In order to do that they need to establish TCP connection using **Sockets**.

- A client can be Browser/Telent.
- A server can written in any language and also can be hostable in any operating system.

What is a Web Server ?

1. A Web server is a server that responds to HTTP requests from the Clients. Sending back the response/data the requested.
2. A web server is a piece of software which listens on a network port for incoming HTTP requests and responds to them.

Note:

Sockets –> TCP –> HTTP(GET/POST) –> Client

Part 2

We have diffrent Web frameworks and diffrent Web servers. Back in the days a Web framework is created for a Specific Web sever. But now we have lots of Web frameworks(flask, django..etc) how are they working seamlessly there have to be a trade off. Like changing codebases for both Web server and Web framewrok.

- These things working seamlessly because **WSGI(Python Web Server Gateway Interface)**.

Web server vs Web framework

1. A **web server** is a physical machine running an Operating System that allows it to serve files via a protocol called HTTP or Hyper Text Transfer Protocol.
2. A **web framework** is a collection of code written in a specific language which provides the basis for building websites in that language. It generally takes care of all the boilerplate code needed and sets up a structure for writing your code.

Why we need Web server and Web framework

1. A **web server** is a physical machine which is capable of delivering content over internet / intranet.
2. A **web framework** is a piece of software / library which helps faster development of web applications.

WSGI

The power of WSGI: It allows you to mix and match your Web servers and Web frameworks. WSGI provides a minimal interface between Python Web servers and Python Web Frameworks.

It's very simple and it's easy to implement on both the server and the framework side.

Why we need HTTP Headers?

The purpose of the headers is to transmit additional information about the HTTP request/response.

How WSGI server serve requests aimed at WSGI application

- First, the server starts and loads an 'application' callable provided by your Web framework/application
- Then, the server reads a request
- Then, the server parses it
- Then, it builds an 'environ' dictionary using the request data
- Then, it calls the 'application' callable with the 'environ' dictionary and a 'start_response' callable as parameters and gets back a response body.
- Then, the server constructs an HTTP response using the data returned by the call to the 'application' object and the status and response headers set by the 'start_response' callable.
- And finally, the server transmits the HTTP response back to the client