Web Framework Implementation With Java

David Alejandro Vasquez Carreño
Ingeniería de sistemas
Escuela Colombiana de Ingeniería Julio Garavito
Bogotá, Colombia
david.vasquez@mail.escuelaing.edu.co

***Abstract*—With java programming language, we have made an easy and basic implementation of a web server that receives GET requests from a client browser that specify the resource he wants to obtain. The application protocol http help us to obtain standard information across the network.**

The current implementation is a simple server that only receives GET requests and return HTML pages with CSS styles, scripts in javascript, limited images and the main favicon that some pages request.

***Index Terms*—Http, html, javascript, Server**

I. Introduction

Web servers are a common component in a company architecture, because they are a need in competition, and for being able to show all their products line up, they need a web page that let clients see what they offer. There are many implementations of web servers right now, some are simple frameworks and other are complex systems with extensible configuration and network management, like Apache.

The problem to handle is the ability to offer simple and tiny solutions to a web server, that could come handy with very simple pages that just make queries to a server with images or files with GET requests, like could be a personal web page, a simple gallery, or a album. For being able to make simple pages, web servers are needed, and we must make clear that this approach is recommended for tiny developments and not for complex web sites.

We have designed a tiny web server that could be an option for tiny development, so by just mapping all the possible requests, and uploading files into your project, you should have a web server ready to show all the content you have prepared.

First we will show some data about the importance of the development of this kind of servers, then we will talk about some concepts of web development, out approach will be shown and finally a simple evaluation about what we did and found.

II. The need of simple web servers

As we said previously, web servers are just needed for companies to show all their products, so it would be correct to say that web servers are a common thing to use if we want out company to be competitive.

All those tools are a very competent tool for handling our web page requests, so at bigger level, those tools are very competitive.

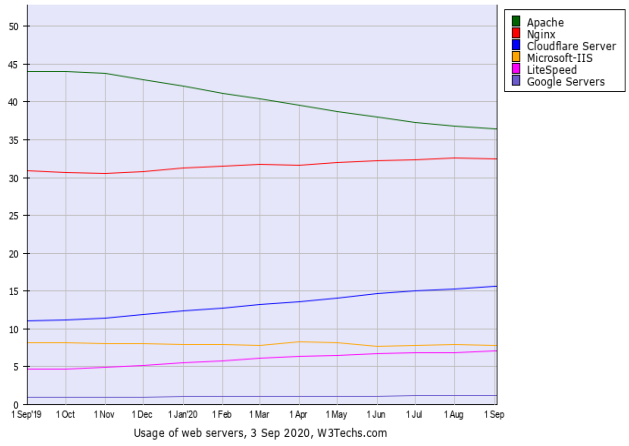

Fig. 1. Web servers statistics

All those web servers are very complex and configurable to create any company web server. The image [1] shows the most used web serves in the world, and with competitive tools like Microsoft internet information services, we can make ourselves a big idea of the competitors in this technology.

Apache is also one of the most used tools in linux based systems like slackware or centOS, with with the years, another linux tool called nginx has been becoming popular over the years due to the great performance [2] it offers in our physical servers.

III. What is a web server?

In a architecture, a web server is a software or hardware, that contains all the logic for the handling of files, likes HTML documents, images, CSS stylesheets, and scripts for a web page. A web server has a constant connection to the

internet, and it is often referenced with a domain name, like *google.com* or *facebook.com*.

A web server works via HTTP requests, that is knows as *hypertext transfer protocol*, that is a application protocol designed for interchanging hypertext data between browsers and clients [3].

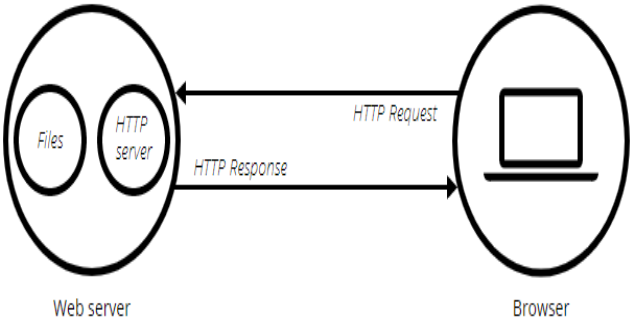

Fig. 2. Web server communication

The image 2 shows a simple communication that occurs when a client attempts to access a document of a web page, like images or scripts. Web servers are often stateless, which means they do not save client data directly into the server, but the client needs to give all the parameters at the moment of the request, and the server will process this information and will return appropriate information like images, forms, styles or scripts [4].

The servers gives the main HTML file for the web page, and the client will process this information and will request all other elements that it may need to complete the web page, like style or scripts for javascript.

IV. Solution

The solution implements is focused just in receiving GET requests from a browser, and it has the simple simple implementation of sending images and scripts, all of those via Data streams in bytes, to avoid incompatibility with most kind of data, because data is transported in the network in bytes.

For being able to send different files, we take care of let the user store HTML files into the server, so that HTML code does not have to be written in java code, but it is still a limitation.

For being able to test the capacilities of the web server, a noSQL database has been created to take all that info and put it into the web page, so every time the page loads, the web server will call a function mapped in the web framework, and will return a file formatted to contain a table with the data that is into the database.

An interesting implementation, is the use of headers collections for the web server, for being able to tell the client exactly

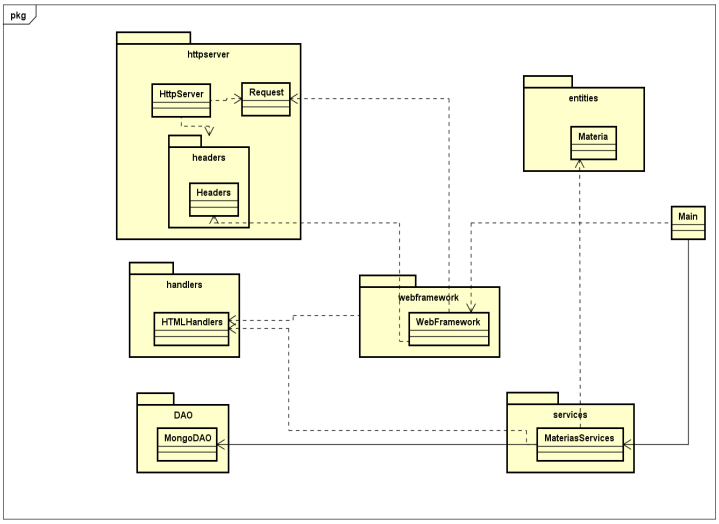

Fig. 3. Design

what the response was, and maintain a clean architecture like REST.

The request class allow us to encapsulate all information in the HTTP request , so that we do not need to be formatting with regular expressions too often, and instead we can have all that data cached into an instance of a Request. The HTMLHanlder make it easier to query HTML files that are directly stored into the web server, and the developer can write this document parallel to java, and he just needs to make a reference to that file into the web framework.


BASE DE DATOS DE MATERIAS DE LA ECI CON MONGODB ;)

Sigla	Creditos	Nombre	Descripcion
CVDS	4	Ciclos de vida del desarrollo de software	Desarrollo de software con maven, integraci&#65533;n y despliegue continuo
AYED	4	Algoritmos y estructuras de datos	Algoritmos avanzados y estructuras de datos como heaps
MATD	4	Matem&#65533;ticas discretas	Demostraciones formales
FDGP	3	Fundamentos de desarrollo y gerencia de proyectos	Fundamentos sobre gerencia seg&#65533;n el PMI
AREP	4	Arquitecturas empresariales	Arquitecturas empresariales

Fig. 4. Simple Web Page

All data in the web page 4 has been loaded from the web server, the images, the styles, and the table with information from the mongoDB database that uses JSON format, readed in java. Java queries the database information from the database, make it object with json format's, read the HTML and insert text representations of those objects into the HTML file, with table tags for make it nice looking.

V. Evaluation

This solution is very recommend if you want to have total control about the way the server behaves, son for tiny

pages it is excellent. The server performs the basic operations for being able to create, test and show web pages in real environments of just GET requests.

A nice addition to this implementation of more requests like POST or PUT, so that the developer is able to create more responsive pages, but even with the actual implementation, you can create different javascript file that make your page responsive, because the web server is able to send and segregate those.

VI. Conclusions

The creation of a simple web server is very easy with languages like java, with so many options, and this is a good exercise for see how a web server works, and if you want to implement your own, now you have an idea of the general organization you should follow.

A general recommendation is to study how a socket works and how it transfer messages in a communication point to point. A strong recommendation for java is the use of spring, that is the union of a web framework and a web server, with the addition of dependency injection into our application at boot. Companies use to have very complex web servers, running directly over the OS, so the software runs as a total different application and the files are accessed by configuration of files, like in linux, or files with service configuration, like in windows.

References

[1] W. Techs, "Historical trends in the usage statistics of web servers." https://w3techs.com/technologies/history_overview/web_server, September 2019. Accessed on 2020-09-03.

[2] Kavya, "Apache vs nginx – which is the best web server for you?." https://serverguy.com/comparison/apache-vs-nginx/#:~:text=At%20serving%20static%20content%2C%20Nginx,up%20to%201%2C000%20simultaneous%20connections., June 2020. Accessed on 2020-09-03.

[3] Mozilla, "Http." https://developer.mozilla.org/es/docs/Web/HTTP, March 2019. Accessed on 2020-09-03.

[4] Mozilla, "What is a web server?." https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server, August 2020. Accessed on 2020-09-03.