

Clients & Services

Johan Sebastian Arias - AREP 2020-2

September

1 Introduction

This is a program developed in Java. The main objective of this program is to thoroughly understand the use of the socket concept, and to include it in the manual implementation (from scratch) of a server receiving web requests of all kinds. Similarly, the idea of this project is to integrate requests to endpoints that return queried data from a database.

2 Objectives

1. To create a simple web server with sockets from scratch.
2. To understand the life cycle of a socket.
3. Understand and implement web requests.
4. To understand the cycle of a web application deployed.
5. To understand the basics of Maven, Git, Heroku, CircleCI via command line.
6. To learn and implement continuous integration and realize of its importance in real environments.
7. To understand how a simple web server is built from scratch.
8. To identify the components of a client-server based architecture.

3 Definitions and Context

3.1 What is socket?

A socket is a software object that acts as an end point establishing a bidirectional network communication link between a server-side and a client-side program. In UNIX, a socket can also be referred to as an endpoint for interprocess communication(IPC) within the operating system(OS).

In Java, socket classes represent the communication between client and server programs. Socket classes handle client-side communication, and server socket classes handle server-side communication.



Figure 1: Apache Spark features
[3]

3.2 What is a web server?

A web server stores and delivers the content for a website – such as text, images, video, and application data – to clients that request it. The most common type of client is a web browser program, which requests data from your website when a user clicks on a link or downloads a document on a page displayed in the browser.

A web server communicates with a web browser using the Hypertext Transfer Protocol (HTTP). The content of most web pages is encoded in Hypertext Markup Language (HTML). The content can be static (for example, text and images) or dynamic (for example, a computed price or the list of items a customer has marked for purchase). To deliver dynamic content, most web servers support server-side scripting languages to encode business logic into the communication. Commonly supported languages include Active Server Pages (ASP), Javascript, PHP, Python, and Ruby. [2]

PrintWriter : The Java PrintWriter class (java.io.PrintWriter) enables you to write formatted data to an underlying Writer. For instance, writing int, long and other primitive data formatted as text, rather than as their byte values.

The Java PrintWriter is useful if you are generating reports (or similar) where you have to mix text and numbers. The PrintWriter class has all the

same methods as the `PrintStream` except for the methods to write raw bytes. Being a `Writer` subclass the `PrintWriter` is intended to write text. [1]

BufferedReader:

The `Java.io.BufferedReader` class reads text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines. Following are the important points about `BufferedReader`

4 Design of this project

4.1 Architecture Diagrams

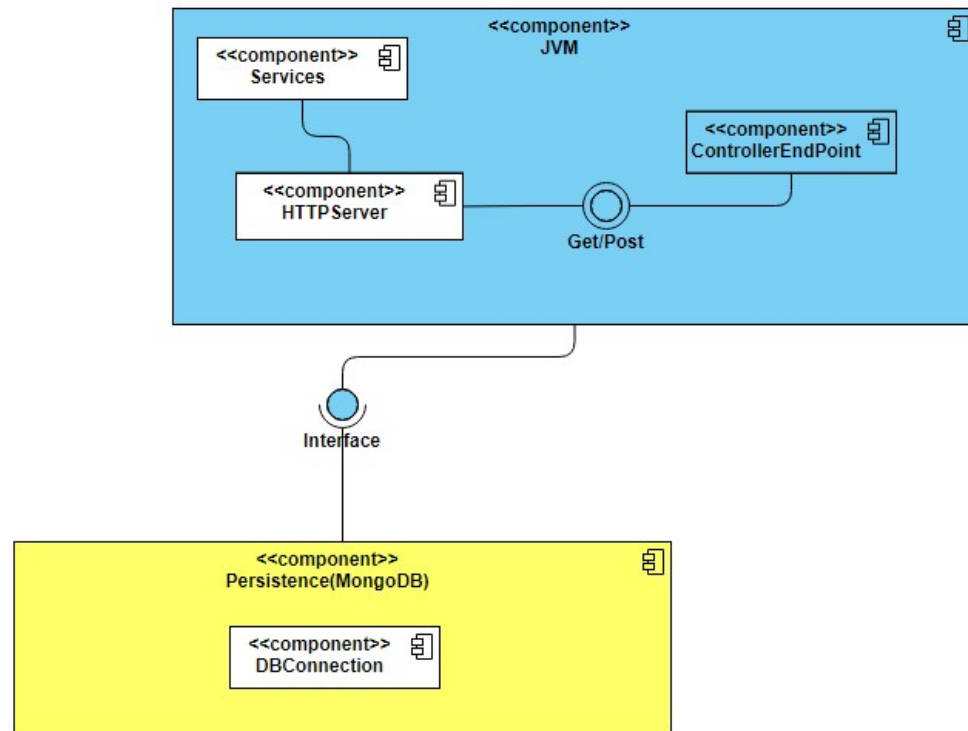


Figure 2: Components diagram of this project

4.2 Class diagram:

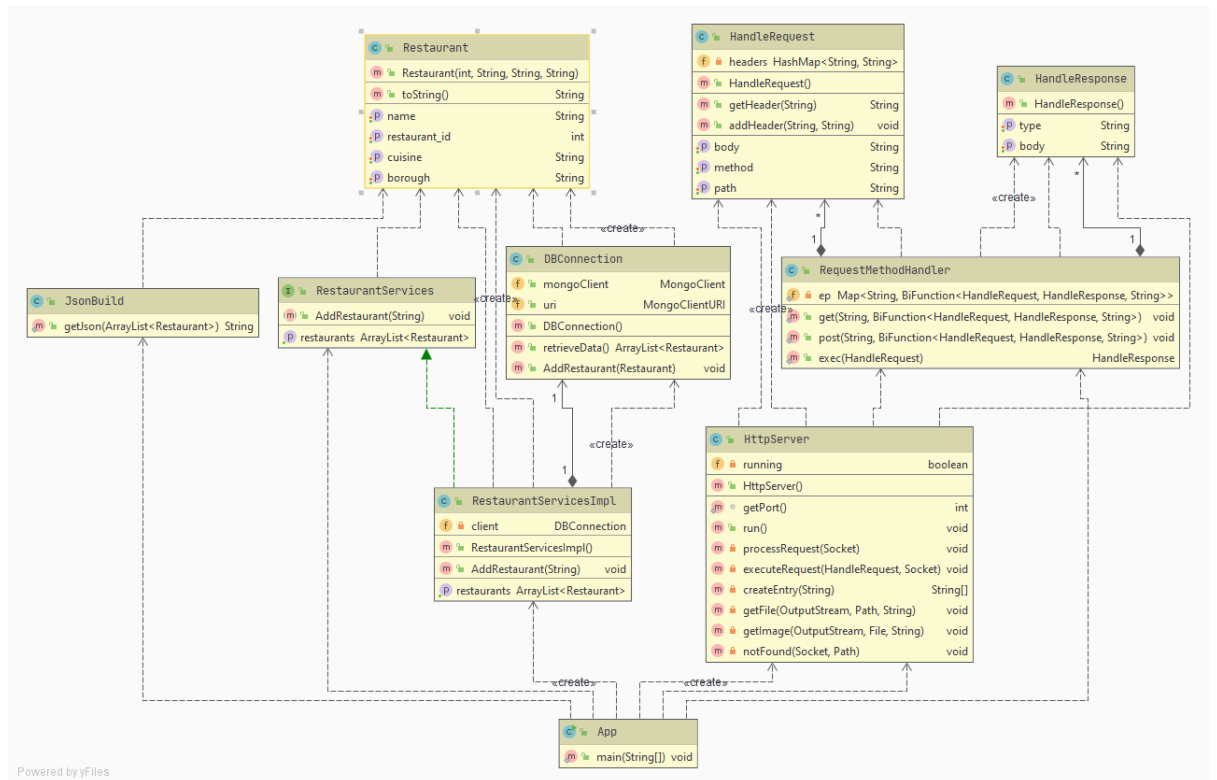


Figure 3: Class Diagram of the design

5 Setting up this project at your local computer

Requisites

- Apache Maven
- Java 8
- git

1. Clone this repository:

```
git clone https://github.com/JohanS11/Lab3Arep.git
```

2. Build the project with maven:

```
cd Lab3Arep && mvn package
```

3. Execute the project with maven:

```
mvn exec:java -Dexec.mainClass="edu.eci.arep.lab3.Challenge2.App"
```

4. Now you should be able to run this project locally at **http://localhost:36000**

6 Conclusion

From this project I have identified the architecture behind the deployment an manual implementation of a web server. I have understand the principles of sockets and web requesting.

References

- [1] Java printwriter. <http://tutorials.jenkov.com/java-io/printwriter.html>. 2020-09-02.
- [2] What is a web server? <https://www.nginx.com/resources/glossary/web-server/>. 2020-09-02.
- [3] What is socket. <https://www.techopedia.com/definition/16208/socket>. Accessed on 2020-09-02.