

Web Server & IoC Framework for web applications using POJOS (September 2019)

Daniel Mauricio Cifuentes Ramirez

I. INTRODUCTION

A web server is a program whose primary function is to store, process and deliver web pages (most frequently html documents, which can include text, images, style sheets and scripts). This communication between client and server is achieved using the Hypertext Transfer Protocol. This document will describe the architecture, the implementation and associated tests of a simple web server capable of delivering html documents and PNG images. This server will also provide an IoC framework for the construction of web applications using POJOS, as well as a sample web app.

II. SERVER ARCHITECTURE

This project is comprised of four main components. A set of handlers, the annotations, the web apps and the server itself. A more detailed description of each of these components is shown below.

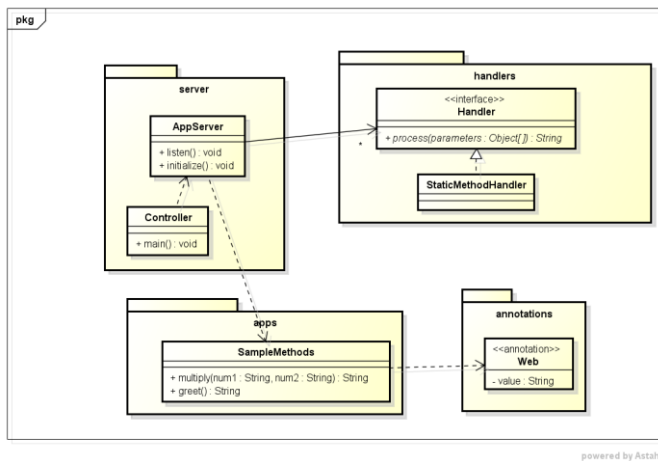


Figure 1. Diagram describing the project's architecture

Server: The server is in charge of receiving and responding to client requests (opening/closing sockets and forming http responses) and mapping every resource to its respective handler.

Handlers: This component stores the handlers for each type of method that the framework supports. Each handler is in charge of invoking the method that it is associated with. This project only has a handler for static methods.

Annotations: The annotations tell the server the name of the resource that a POJO is assigned to so that the server is able to map the handler to the correct address.

Apps: This component stores the apps that use the annotations.

When the server starts up it searches in the apps component for every method containing the @web annotation, it then gets its value and uses it to map a handler to a new unique name for the resource.

III. DEPLOYMENT ARCHITECTURE

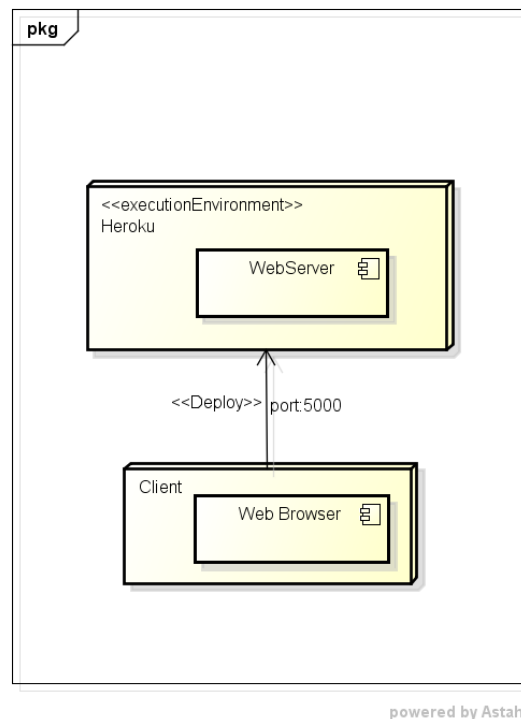


Figure 2. Diagram describing how the project is deployed

The project is deployed in Heroku and can be seen at <https://dcifuentes-arem-p1.herokuapp.com/>

IV. TESTS

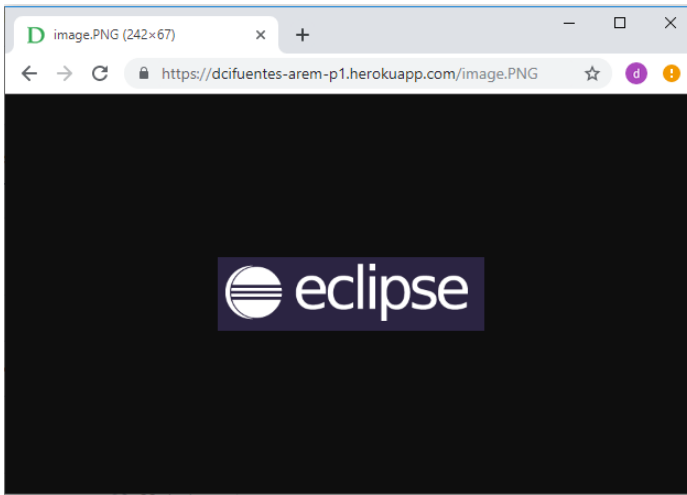


Figure 3. Requesting an image from the server

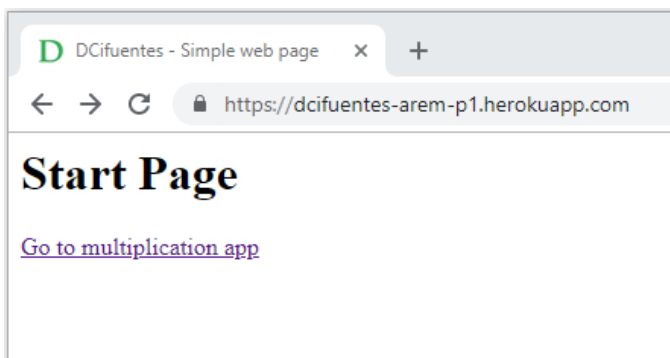


Figure 4. The index.html of the sample web application

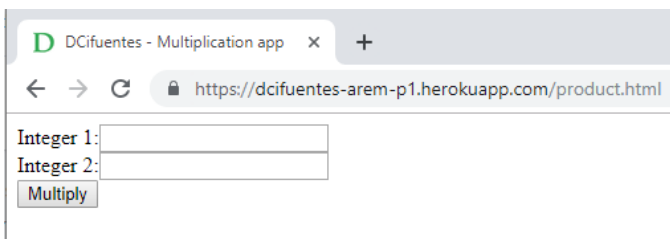


Figure 5. The product.html file of the sample web application

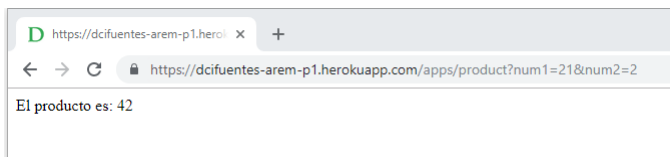


Figure 6. The web application in action