APPLICATION SERVER ARCHITECTURE WORKSHOP, OBJECT META PROTOCOLS, IOC PATTERN, REFLECTION

Juan Sebastian Nieto Molina
February 2021

Abstract

This document is the solution of a guide that has to objective make a framework like spring-boot, using IOC pattern, reflection and POJO to make a web service, then deploy it on heroku. it contains views where you can see .css or .js files and .png or .jpg photos.
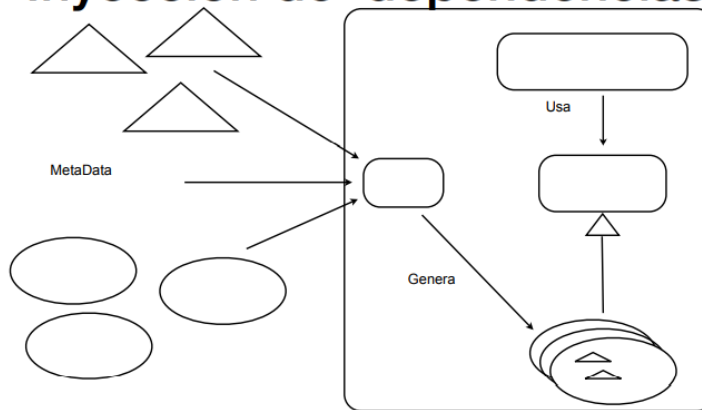
Contents

1   Introduction

This guide has the intention of make a framework like Srping-Boot making use of tags like @RequestMapping, but it only use the http method (get), the idea her is do a deploy in heroku creating this service.for this case, you will need java 11, heroku cli if you want prove it on local host, it contain views in format .css, .js, .png, jpg or .html in the index case. It use reflection to obtain the function that contains tags @RequestMapping and obtain the path and value of this functions to render it on the scream.
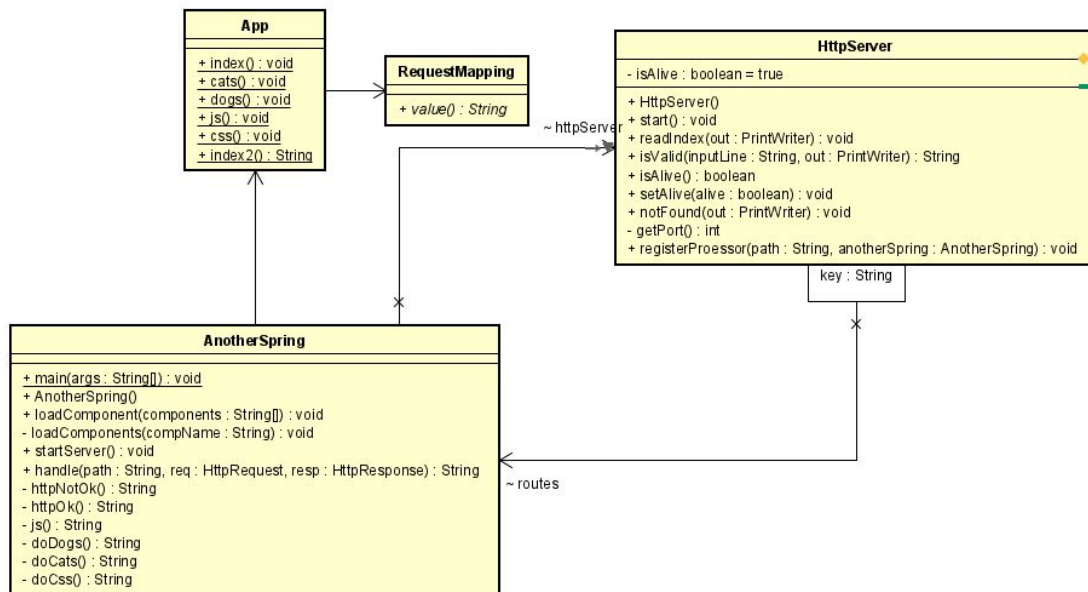
2   Design

This design has 3 classes, the first class App, that contain the controllers with tags @RequestMapping, the second class has the http server with client-server to communicate when the user want a get petition, and finally Another-Spring class, that contain all the logic to recognize when a path belong to us or not and generate a request with 200 or 404 code. The IOC was implement whit this designs.
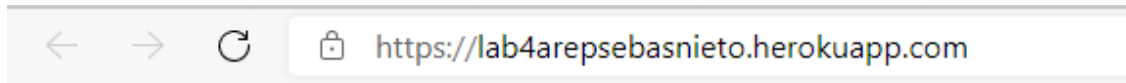
Arquitectura de IoC con Inyección de dependencias

The class diagram has this disigns

App

+ index() : void
+ cats() : void
+ dogs() : void
+ js() : void
+ css() : void
+ index2() : String

RequestMapping

+ value() : String

HttpServer

- isAlive : boolean = true

+ HttpServer()
+ start() : void
+ readIndex(out : PrintWriter) : void
+ isValid(inputLine : String, out : PrintWriter) : String
+ isAlive() : boolean
+ setAlive(alive : boolean) : void
+ notFound(out : PrintWriter) : void
- getPort() : int
+ registerProessor(path : String, anotherSpring : AnotherSpring) : void

~ httpServer

key : String

AnotherSpring

+ main(args : String[]) : void
+ AnotherSpring()
+ loadComponent(components : String[]) : void
- loadComponents(compName : String) : void
+ startServer() : void
+ handle(path : String, req : HttpRequest, resp : HttpResponse) : String
- httpNotOk() : String
- httpOk() : String
- js() : String
- doDogs() : String
- doCats() : String
- doCss() : String

~ routes

You have to complete the path that is deployed in heroku to see the views correctly

https://lab4arepsebasnieto.herokuapp.com

Buen día qué necesitas hoy?

Ver imagen uno, agregar en la ruta /App/cats.png

Ver imagen dos, agregar en la ruta /App/dogs.jpg

Ver css, agregar en la ruta /App/css

Ver js, agregar en la ruta /App/js

2.1 Http server

A http server is piece of software that can interpret the protocol http, in this case is static, because it has a relation one-one, it means that it gonna give you the information that it has stored, with this you can do petitions in the browser and if the server has the information stored it return that, other case it will launch the famous 404, well, if it is well configure.

2.2 POJO

The acronym for Plain Old Java Object, emphasizes the use simple classes that are not dependent on a framework especially. They arise in Java by frameworks every time more complex hiding a real model.

2.3 Ioc

Object-oriented design pattern, in which objects are supplied to a class instead of creating these objects itself, removing the responsibility for instantiating by delegating it to a different component.

3 Conclusion

For this laboratory, create a framework with http get method is hard when you cant read properly the headers that contain the files, even when you return it you have to create some new pieces because heroku need reed it, also you have to control the httpServer, even if it is not parallel or concurrent. Learn hot to create web pages with use of a framework create by us, and how to deploy in heroku. Also we see how to work with tangs and Ioc

4 References

[1] Benavides, P. L. D. (2021). TALLER DE ARQUITECTURAS DE SERVIDORES DE APLICACIONES, META PROTOCOLOS DE OBJETOS, PATRÓN IOC, REFLEXIÓN, Bogotá, Colombia.

[2] Sebastian Nieto (2021). Tomado de: https://github.com/sebastianNietoMolina/AREP-lab4

[3] Benavides, P. L. D. (2021). Mecanismos para generación de frameworks de aplicaciones empresariales Bogotá, Colombia.