Lab 4

andresvilla990223

February 2020

1 Introduction

This laboratory intended us to create a java http server that is able to respond get petitions with .html, .js, .png or.jpg depending on the resources asked.

2 Design

2.1 HTTPServer

This challenge was accomplished creating a java server that receives a http petition. It analyzes the petitions, if it is under the scope of a custom annotation, it calls the method of that annotation and answers the petition. If it isn't, it looks for the resources, then send that info as one of the mentioned type of files.

3 Structure

3.1 Java http server

The java app first instantiates the http server, and configures the annotation responses, by looking on the methods of each class, and checking if it has a custom annotation.

```
for (File folder : folders) {
    if (folder.exists()) {
        for (String clase : folder.list()) {
            System.out.println(clase);
            if (clase.endsWith(".class")) {
                Class<?> c=null;
                try{
                    c = Class.forName("edu.escuelaing.arep.annotations."+clase.s
                    Method[] methods = c.getMethods();
                    for (Method m : methods) {
                        if (m.isAnnotationPresent(Web.class)) {
```

If it does, the values of the annotation and the method associated are stored on a Map.

```
webAnnoted.put("/ann/" + m.getAnnotation(Web.class).value(), new AnnnotationHandler(m));
                                    System.out.println("webannoted");
                                }
```

Then the server opens a port and starts to listen. When the servers receives a client socket, it creates a thread to handle it

```
    while (conectado) {
            try{
                clientSocket = serverSocket.accept();
                System.out.println("Conectado");

                Thread t1 = new Thread(new worker(clientSocket,webAnnoted));
                t1.start();
            }
            catch(Exception e){System.out.println("error "+e);
                serverSocket.close();
        }
    }
```

When the petition contains is directed towards an specific URL (/ann/), the server looks up on the map for the method to answer this petition.

```
 else if(header[1].contains("/ann")){
                    String petition = header[1];
                    String param = "";
                    String response ="";
                    if (petition.contains("?")) {
                        param = petition.substring(petition.indexOf("?") + 1, petition.l
                        petition=petition.substring(0, petition.indexOf("?"));
                    }
                    System.out.println("petition "+petition);
                    if(webAnnoted.containsKey(petition)){
                        if(param.equals("")){
                            response = webAnnoted.get(petition).handle();
                        }
                        else{
                            response = webAnnoted.get(petition).handle(param);
                        }

                        respondRaw(out, dataOut, response, "text/html", "200 OK");
                    }

                }
```

If it is another get petition, it then tries to search for the file, if it is found, the server will give the client that file.

```
if (s[0].equals("ok")) {
    rFile = new File(ROOT+ s[1] + header[1]);
    if (rFile.exists()) {
        respond(out, dataOut, rFile, s[2], "200",ROOT+ s[1] + header[1],outS);
    }
```

If the file does not exists, the server will return the "notFound.html".

```
else {
    rFile = new File(ROOT, FILE_NOT_FOUND);
    respond(out, dataOut, rFile, "text/html", "404",ROOT+ FILE_NOT_FOUND,outS);
}
```

If the requested file is from a non supported media type, it will return the "notSupportedMedia.html" file.

```
 else {
    rFile = new File(ROOT, UNSUPPORTED_MEDIA_TYPE);
    respond(out, dataOut, rFile, "text/html", "415",ROOT+ UNSUPPORTED_MEDIA_TYPE,outS);
```

Finally if it is not a get petition, the server will return the method$_{n}ot_{s}upported.html$.

```
 else {

    File f = new File(ROOT, METHOD_NOT_ALLOWED);
    respond(out, dataOut, f, "text/html", "405",ROOT+METHOD_NOT_ALLOWED,outS);

}
```

4 Conclusions

This laboratory showed us the possibilities for managing servers or any kind of app with annotations, allowing for a cleaner and more modular code. As with custom annotations it isnt need as much code as one can use the same code with different variables without having to call explicitly that function.

5 Bibliography

https://www.baeldung.com/java-custom-annotation https://mkyong.com/java/java-custom-annotations-example/ https://www.javatpoint.com/java-annotation