*Arkadiusz Paryz (COS)*

HTTP – Day 1

Exercise 1:

In Chrome, with the use of the developer tools(F12) I see the source code for the index.html file and the status of the file. All of the DOM elements are listed as well.

If I change the name of the index.html to index1.html, when I try to run the project I run project, I see a 404 error massage, because by default, Apache’s Tomcat, when provided with an URL, looks for the “index” HTML file. If I just add /index1.hml to the URL http://localhost:8080/Day1/, I can still access the webpage.

Exercise 2:

If I add an image and reference to myStyle.css (without having the actual css file) the results are as it goes:



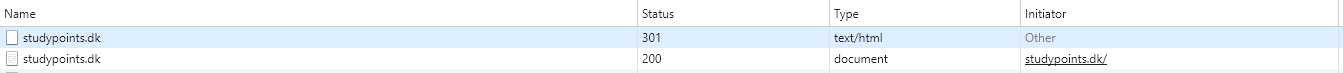
index.html – Status 200, Type Document // Meaning the connection was a success

lunar\_2x.jpg – Status 304, Type JPEG // Nothing changed – the image came with (was initiated by) index.html

myStyle.css – Status 404, Type stylesheet // The resource file wasn’t available, thus it was never loaded

If you eventually load the myStyle.css you will get the status error 304 since nothing is changed.

Exercise 3:

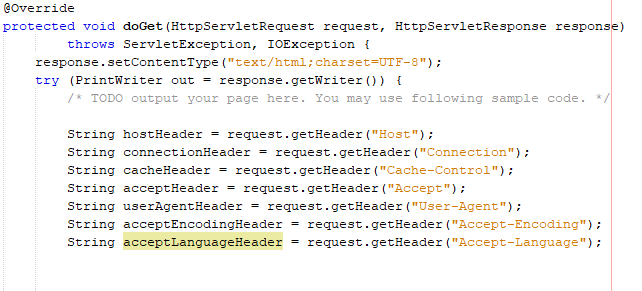


I am redirected from http://studypoints.dk/ to https://studypoint.dk. For security reasons.

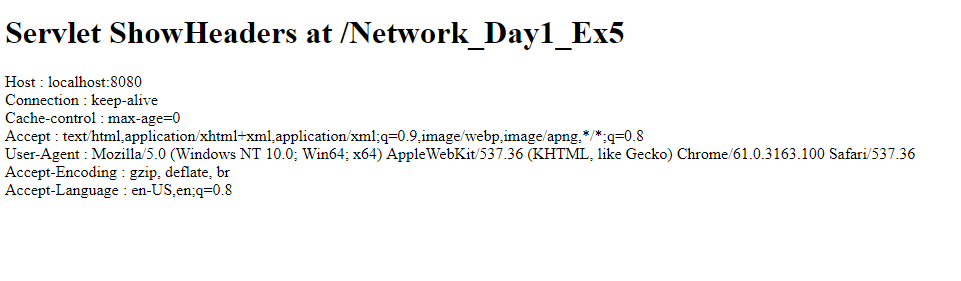
The HTTPS stands for Hypertext Transfer Protocol *Secure.*  It is still HTTP, but it uses two more protocols - SSL (Secure Sockets Layer) and TLS (Transport Layer Security).

Exercise 4:

By creating a ShowHeaders.java servlet, and writing the fallowing code to the doGet()

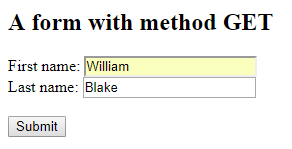


The requested result, using a servlet outputs this to the webpage

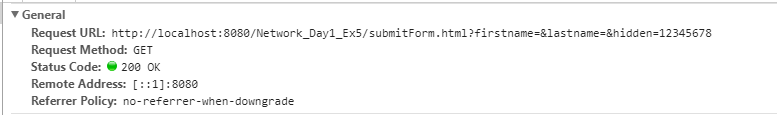
`

Exercise 5:

When I try to “GET” with the hidden input field

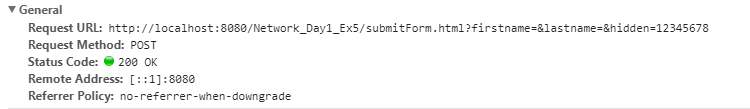


I get



This happens, because the form submits all the values that were in the form. Even though the input “hidden” is not visible it has a predefined value of 12345678, which is going to be submitted anyway

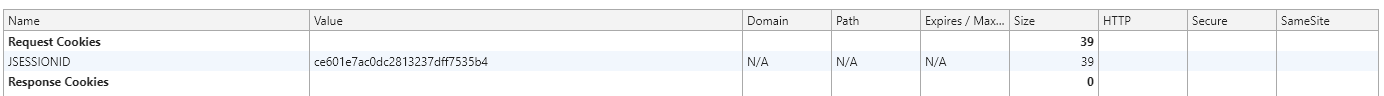
When I try to the same thing, but with a method=”POST” I get almost the same result



The Request Method: POST is the only difference. From then, the server can decide what to do with the information, based on the requested method. GET and POST are simply conventions, that we used. You can still use GET to “post” data, and POST to “get” data, but it’s counterintuitive and quite confusing to do so.

Exercise 6:

When I run the Exercise 6 servlet, and I input a name, I get a “cookie”



The cookie is being sent with every request to the server. This is used, in order to have a “continuous session, a state, while still communicating via an otherwise “stateless” protocol (http).

Every time a request is made, the server will request.getParameter(“name”) to see if there is an active session, between it and the client. If not, then request.getSession().setAttribute("name", name); will create a new session object (the coockie). Now, the client will have a way to identify themselves, when access the server.