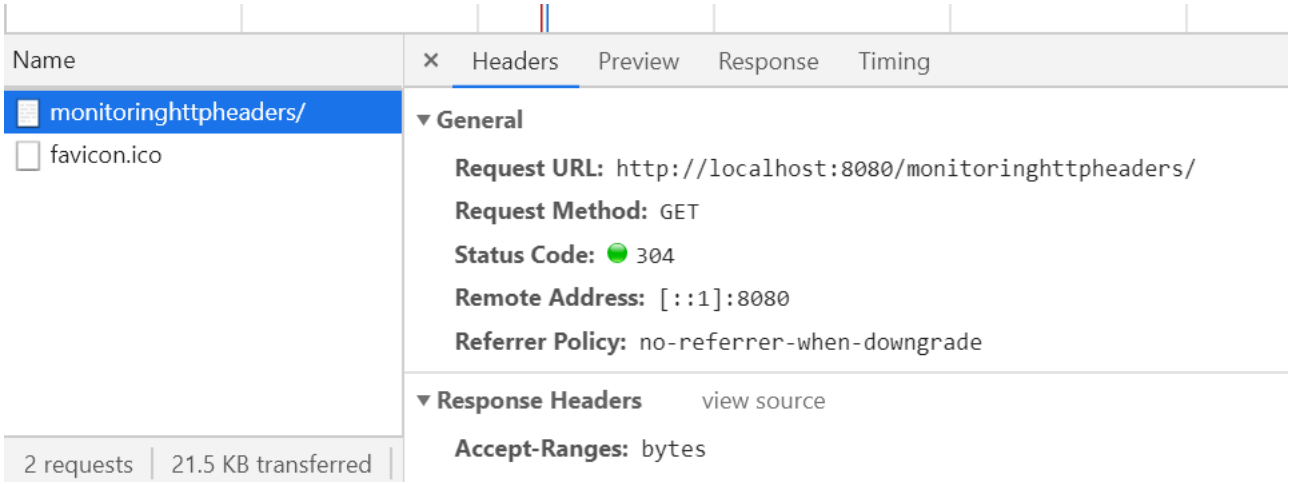
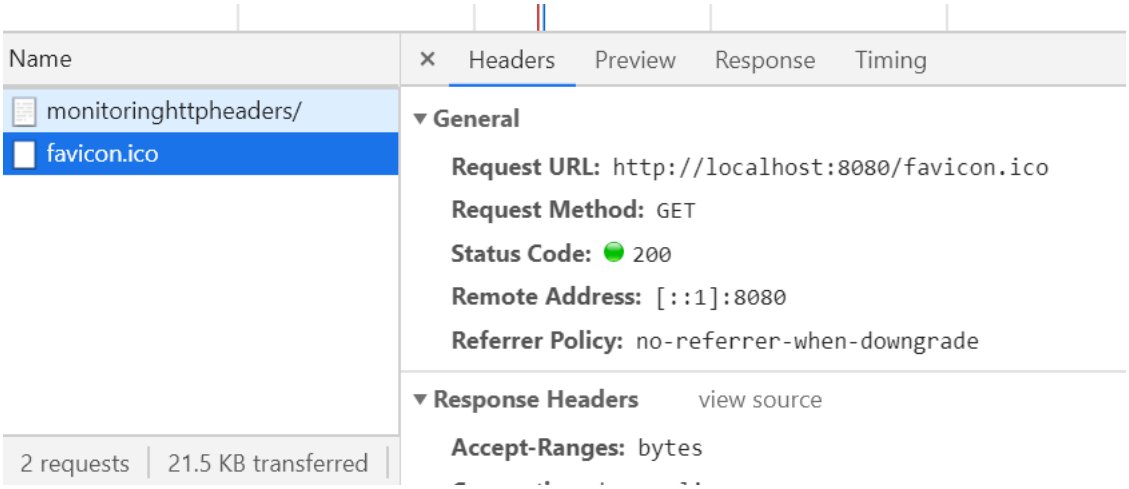
The HTTP Protocol

1) Monitoring HTTP Headers 1

Observe and explain each of the values monitored (use view source to see the plain messages).



The 304 status code is used for caching purposes, because it tells the client that the response hasn't been modified, so the client can continue to use the same cached version of the response.

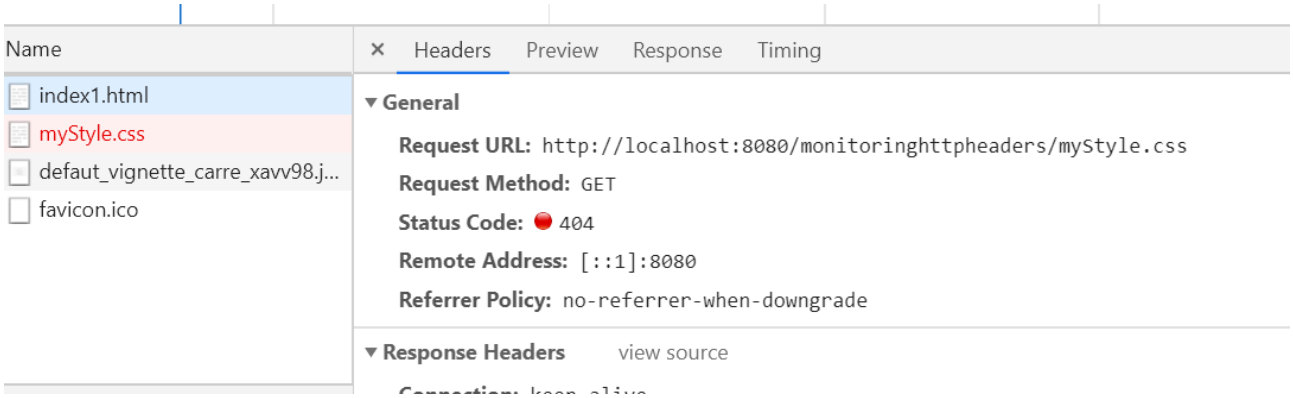


The 200 status code means that the GET request method has succeeded.

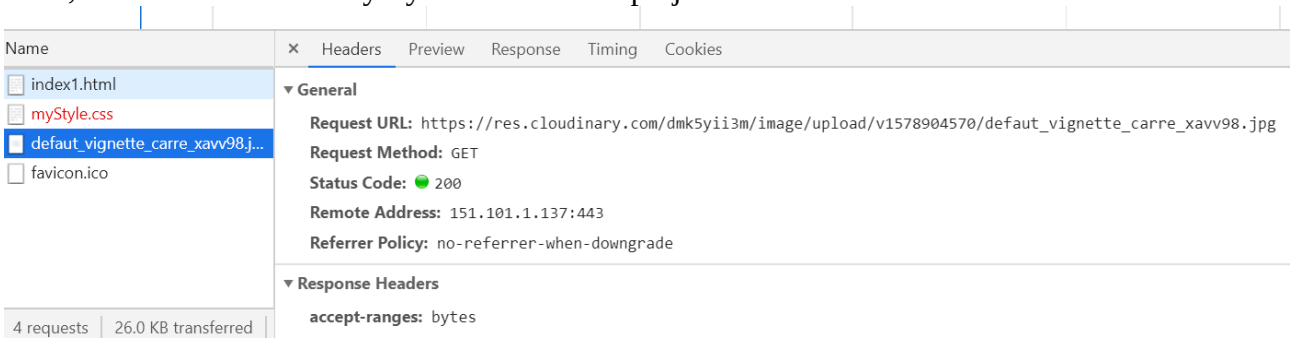
2) Monitoring HTTP Headers 2

Explain the purpose of the connection header.

The connection header controls whether the network connection stays open after the current transaction finishes. If the value sent is keep-alive, the connection isn't closed, allowing for subsequent requests to the same server to be done.



The 404 status code mean that the server can't find requested resource, which it in this scenario can't, since I didn't add a myStyle.css file to the project.

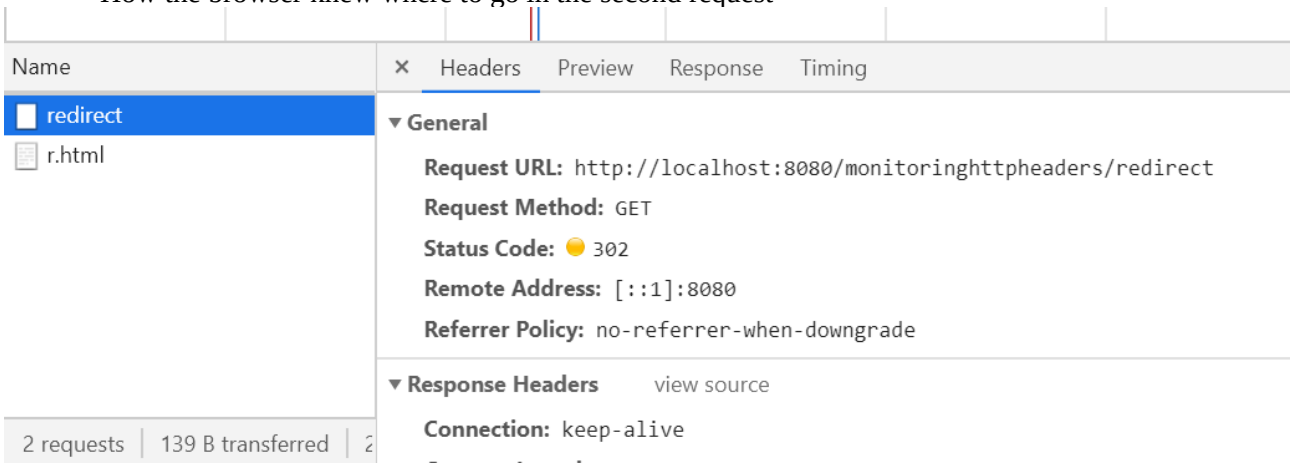


The picture has successfully been found.

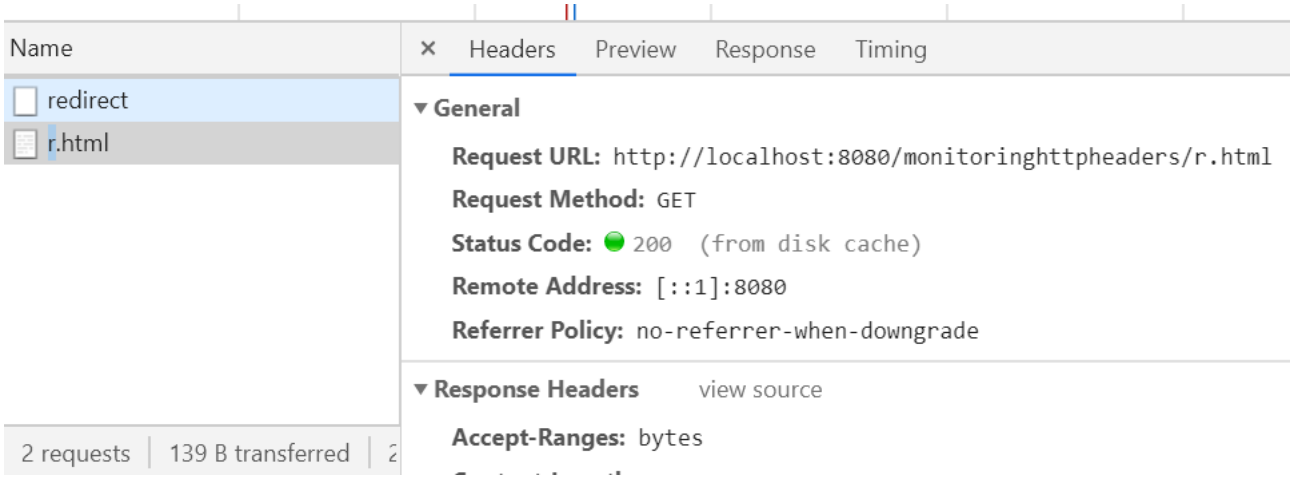
3) Monitoring HTTP Headers 3 (Response-codes 3xx)

Enter the address for the servlet (**http:localhost:8080/redirect)** into the browser and explain:

- The two HTTP-request you see
- How the browser knew where to go in the second request

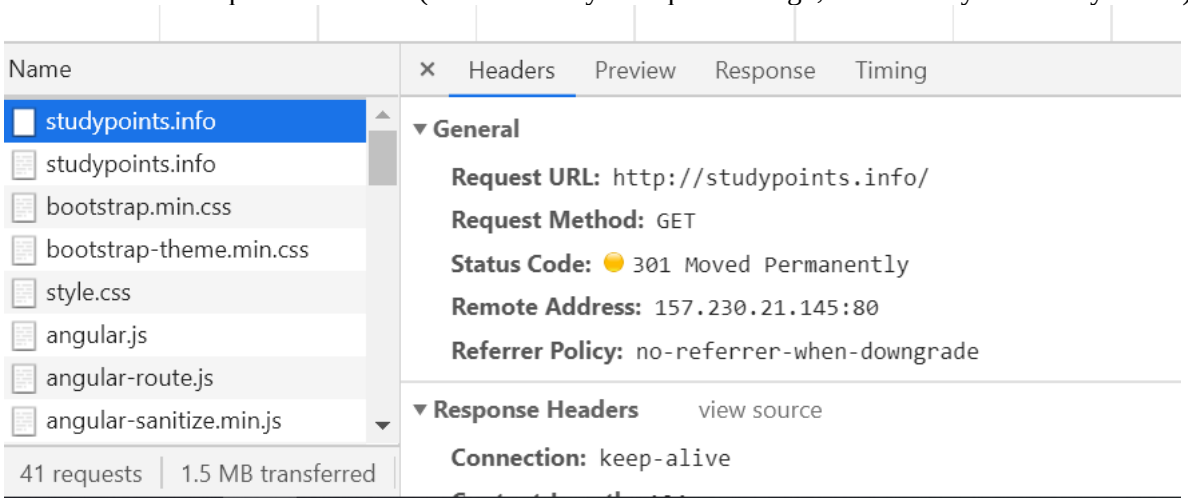


The 302 status code means that the URI of requested resource has been changed temporarily in this case from the servlet to r.html, which it finds and redirects the client to.

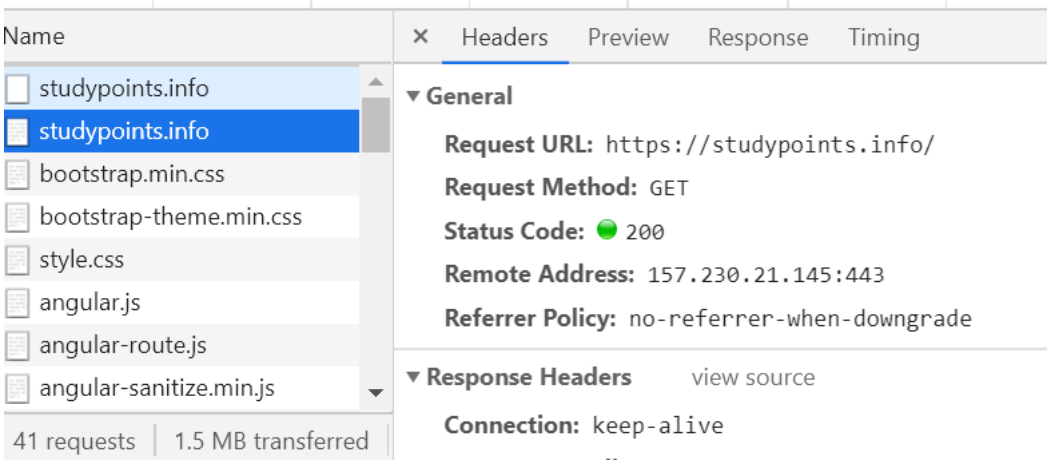


3a) Redirecting to HTTPs instead of HTTP

Explain the first two request monitored (notice where you requested to go, and where you actually ended).

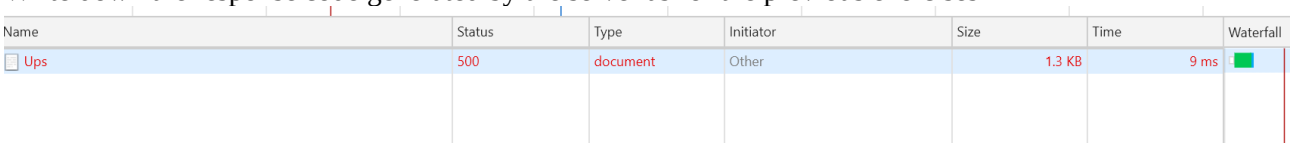


The 301 status code means that the URL of the requested resource has been changed permanently. The new URL is given in the response and the client is redirected to that site, which in this case was https://studypoints.info/#/view1.



4a) Status Codes (5xx)

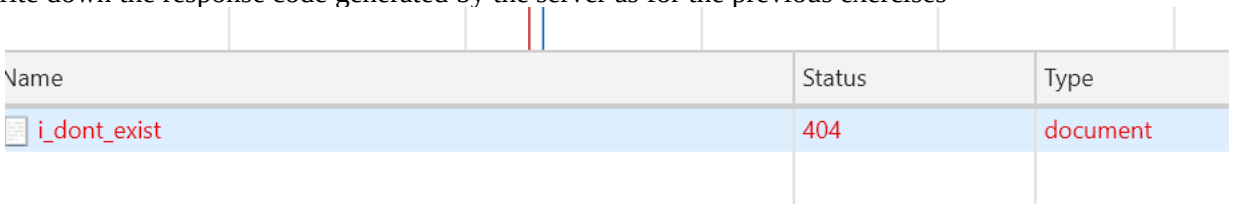
Write down the response code generated by the server as for the previous exercises



The 500 status code means that an internal server error occurred and the server has encountered a situation it doesn't know how to handle.

4b) Status Codes (4xx)

Write down the response code generated by the server as for the previous exercises



Another 404 not found status code (see Monitoring HTTP Headers 2).

4c) Status Codes - Ranges

Your document, containing the Status Codes for all the exercises done so far, should now contain codes like 2xx, 3xx, 4xx and 5xx.
Explain (write down your answer so you won't forget) the meaning of the first digit in the 3-digit Status Codes you have seen so far.

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

- Informational responses (100–199) (This status code range hasn't been seen in this exercise)
- Successful responses (200–299),
- Redirects (300–399),
- Client errors (400–499),
- and Server errors (500–599).

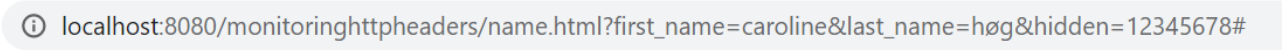
5) Get HTTP Request Headers on the Server

The solution to this exercise can be seen in code in the folder "monitoringhttpheaders" in "/src/main/java/servlets/RequestHeaderServlet.java".

6) Get/Post-parameters

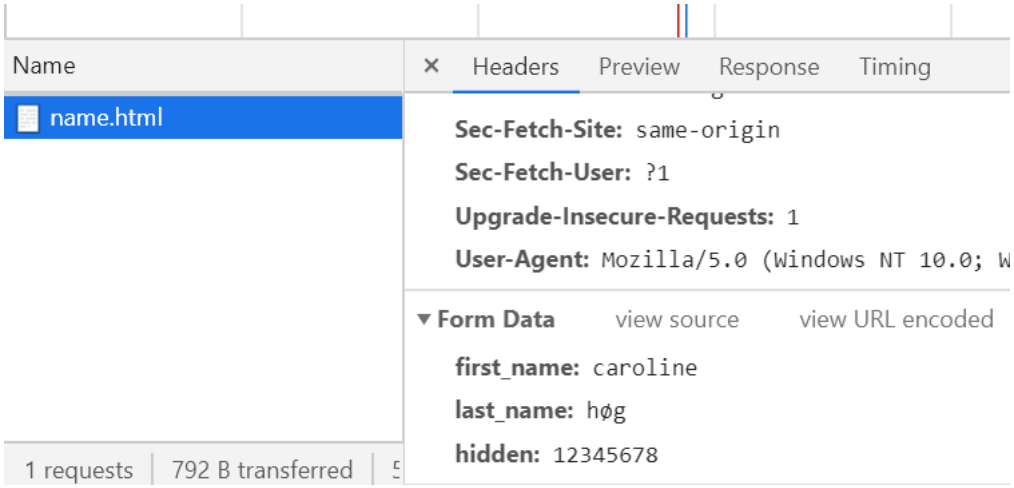
Set the forms method-attribute to the value "GET" (actually the default value) and test the form. Observe what happens in your browser's address field.

GET is used to request data from a specified resource. The URL contains name/value pairs, which is how the GET method stores its parameters.



Change the forms method-attribute to the value “POST” and test the form. Observe the change in your browsers address field. Figure out (using Chrome Developer Tools), how parameters are passed in, for a POST request.

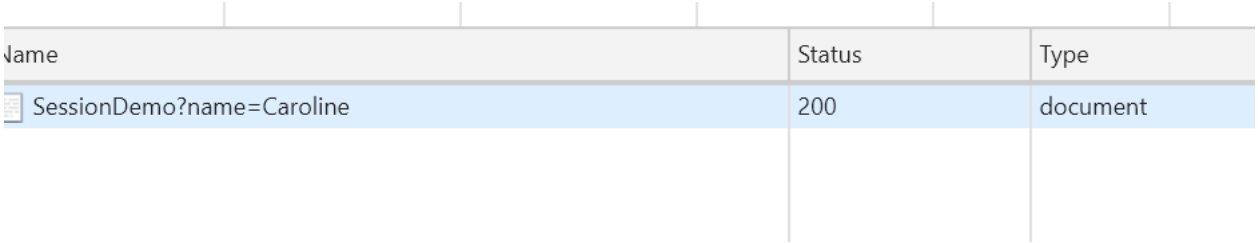
When a POST request is being used, the data sent to the server is stored in the request body of the HTTP request.



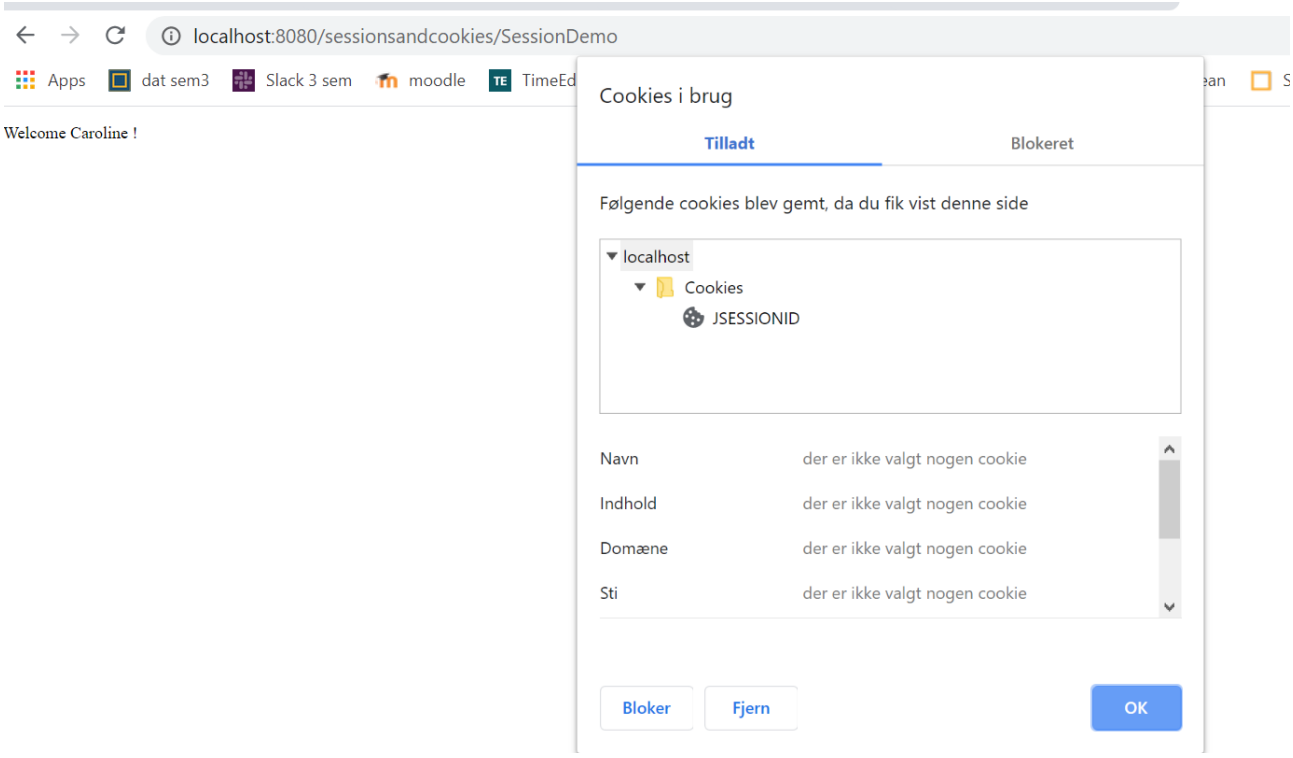
Session and Cookies

7) Sessions (Session Cookies)

d. Enter your name and press submit, copy the URL in the browser into your clipboard, close the tab (but not the browser) and load the page again in a new tab using the URL in the clipboard.
e. While doing the things in step d, you should monitor the content of your local cookies and the HTTP requests being sent, using the development tools in Chrome.
f. **Most import part of this exercise:**
Explain (on paper) using both words and images how the Server can maintain state between subsequent calls even when the state is not transmitted from the client to server.



In the following picture it can be seen that I loaded the page in a new tab without the “name=Caroline” parameter; however, it remembers the name. This is because of the session cookie, which is stored in temporary memory. This means that the session cookies are removed when the session is over (browser is closed), but if it’s the same session, the server can maintain its state between subsequent calls.

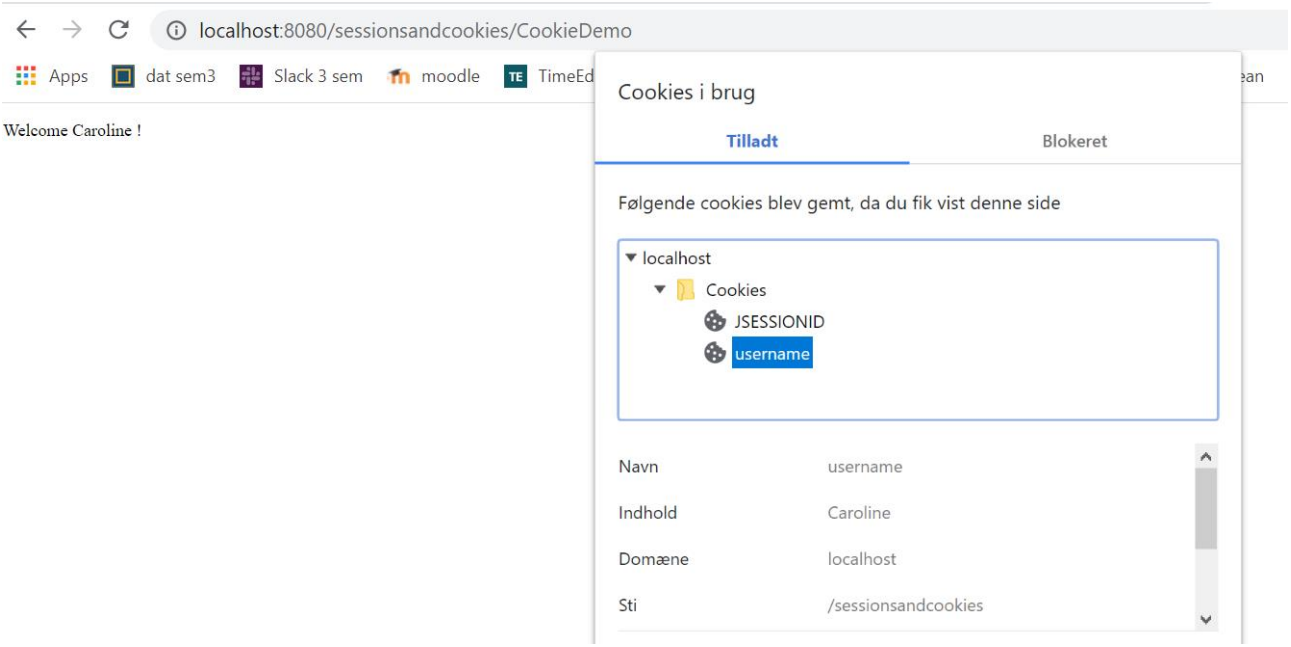


8) Persistent Cookies

d. Enter your name and press submit, copy the URL in the browser into your clipboard, close the tab (but not the browser) and load the page again in a new tab using the URL in the clipboard.
e. Now close your browser (you could even close your laptop, but don't ;-) , open it again and load the page again using the URL in the clipboard
f. While doing the things in step e, you should monitor the content of your local cookies and the HTTP requests being sent, using the development tools in Chrome.
g. **The most import part of this exercise:**
Explain (on paper) how Cookies can be used to maintain "state" on the client between subsequent calls to a server, even when a browser has been closed down.

Name	Status	Type
CookieDemo?name=Caroline	200	document
favicon.ico	200	x-icon

In the following picture, I once again loaded the page in a new tab and didn't give my name as a parameter, but it still remembered. This is due to the persistence cookie, which is stored as a text file on the hard drive of a computer, therefore the cookie isn't removed when the browser is closed. Instead they can have a set expiration date.


localhost:8080/sessionsandcookies/CookieDemo
Apps
dat sem3
Slack 3 sem
moodle
Welcome Caroline !
Cookies i brug
Tilladt
Blokeret
Følgende cookies blev gemt, da du fik vist denne side
localhost
Cookies
JSESSIONID
username
Navn
username
Indhold
Caroline
Domæne
localhost
Sti
/sessionsandcookies