## 1) Monitoring HTTP Headers 1

Vi ser en 200 ok så den er gået igennem og det er en get metode som har Hello World! i body.

Efter rename til index1.html får jeg 404 fejl.

### 2) Monitoring HTTP Headers 2

Meningen med connection header er at du kan navigere rundt i dine filer. Vi tilføjede et css og derved kom der en request efter vores css hvor vi så fik svar og kan se vores css i denn url: <http://localhost:8080/MonitoringHttpHeaders/myStyle.css>

### 3) Monitoring HTTP Headers 3

Vi ser de 2 requests hvor redirect har en status kode på 302 hvilket betyder den er flyttet midlertidigt og i response headeren har redirect også en location der er navnet på vores anden request nemlig r.html.

r.html har så en status kode på 304 hvilket står for ikke modificeret og det vil sige der ikke er behov for at videresende de efterspurgte ressourcer (dette er endestationen).

### 3a) Redirecting to HTTPs instead of HTTP

Vi ser på det ene request at det har status kode 301 hvilket er at vi bliver flyttet permanent og derfra bliver vi flyttet fra http til https.

### 4a) Status Codes

500

### 4b) Status Codes

404

### 4c) Status Codes

302 - Midlertidig flyttet

304 - Ikke modificeret(endestation)

301 - Permanent flyttet

500 - Intern serverfejl

404 - Dødt link

3 - Har noget med redirects at gøre (Redirection)

4 - Har noget med clienten at gøre. Som i opgaven prøvet at tilgå en side der ikke eksistere (Client Error)

5 - Har noget med serveren at gøre. Som i opgaven noget er lavet forkert eller går galt på serveren (Server Errior)

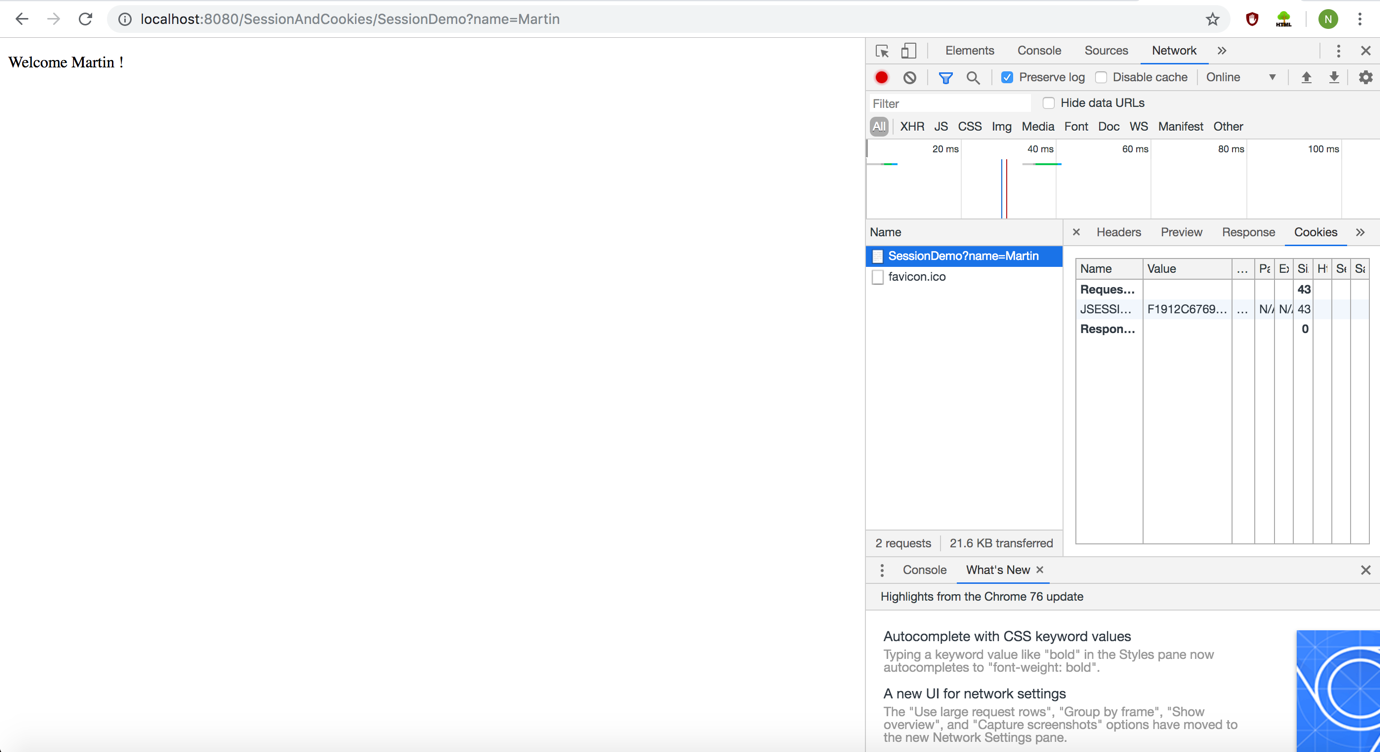
### 6) Get/Post-parameters

Get metoden: Indsætter værdierne i firstname og lastname til de indtastede værdier og så sender det den skjulte værdi med også. Vi ser også at den afslutter med # fordi dette er stedet vi ønsker at sende vores data hen.

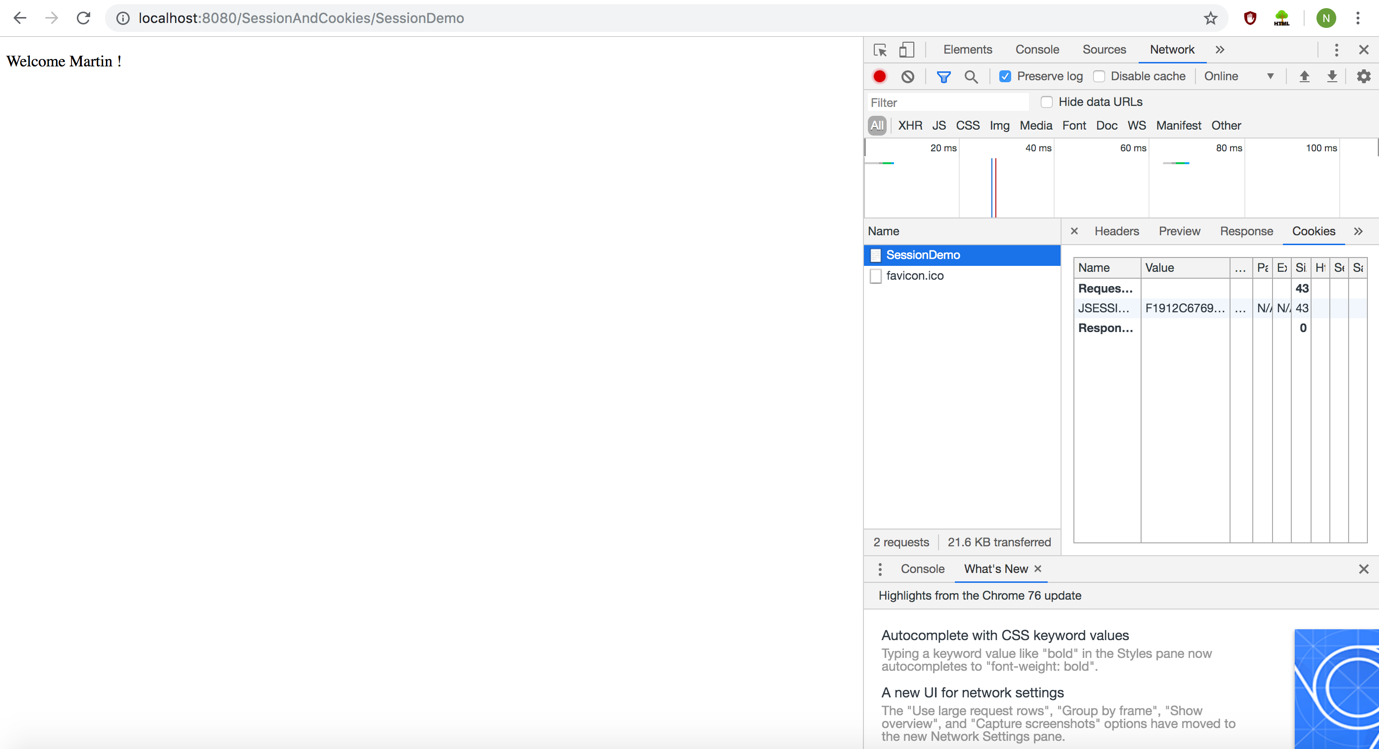
Post metoden: Igen indsætter metoden værdierne i firstname og lastname samt gemmer vores skjulte værdi. Forskellen her ligger i at get metoden sender det med igennem url’en hvor post metoden sender det igennem html body. Derved er post metoden meget mere sikker til sensitiv data sendinger.

Session and Cookies

Browseren gemmer det indtastede navn i en session og derfor kan du ikke skrive et nyt navn ind fordi der allerede er en igangværende session. Hvis du i stedet lukker browseren får du mulighed for at indtaste navnet på ny og derved oprettes en ny session.



Her kan vi se den cookie der bliver oprettet første gang vi indtaster navnet.



Her kan vi se at cookien stadig er den samme selv efter browseren har været lukket se url.

Persistent Cookies

Cookie bliver gemt local og forsvinder først på en bestemt dag(typisk 2 år efter oprettelsen) og bliver kun gemt i den browser du har brugt til at gemme cookien fx chrome browser gemmer en cookie men safari browser skal gemme sin egen cookie.