1) Monitoring HTTP Headers 1

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

A screenshot of a cell phone

Description automatically generated

Request url: der hvor mappen er.

Request method: Det er en GET Method der bliver brugt.

Status Code: 304 – not modified. Dette er grundet at jeg startede over og den stadig har cachen fra tidligere projekt. Tidligere var status kode 200 – OK.

Remote Address: den port der bliver brugt.

2) Monitoring HTTP Headers 2

A screenshot of a cell phone

Description automatically generated

Request url: der hvor mappen er.

Request method: Det er en GET Method der bliver brugt.

Status Code: 200- OK, og den fortæller os også at den har gemt den i cachen.

Remote Address: den port der bliver brugt.

A screenshot of a cell phone

Description automatically generated

Request url: der hvor mappen er.

Request method: Det er en GET Method der bliver brugt.

Status Code: 404 – Not Found. Da style.css ikke er lavet I projektet.

Remote Address: den port der bliver brugt.

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

A screenshot of a cell phone

Description automatically generated

Jeg får kun et request. Det andet request et bare tomcat iconet.

Request headersne har samme forklaring som tidligere.

Kan ikke svare det andet spørgsmål uden at kunne se det andet request.

3a) Redirecting to HTTPs instead of HTTP

A screenshot of a cell phone

Description automatically generated

A screenshot of a social media post

Description automatically generated

Status kode 301 – moved permanently. Den nye URL burde være givet i response, men kunne ikke loades. Men ved at søge <http://studypoints.info/> bliver vi omdirigeret til <https://studypoints.info/> som kan ses forneden.

A screenshot of a cell phone

Description automatically generated

Her kan man se request URL er blevet til https i stedet for http, og vi får en status kode 200.

4a) Status Codes (5xx)

A screenshot of a cell phone

Description automatically generated

Status kode: 500 – internal server error. (Serveren er stødt i et problem den ikke ved hvordan den skal løse, grundet tilføjelsen af in tresult = 100/0;.

4b) Status Codes (4xx)

A screenshot of a cell phone

Description automatically generated

Status kode 404 – not found. Forklarer lidt sig selv.

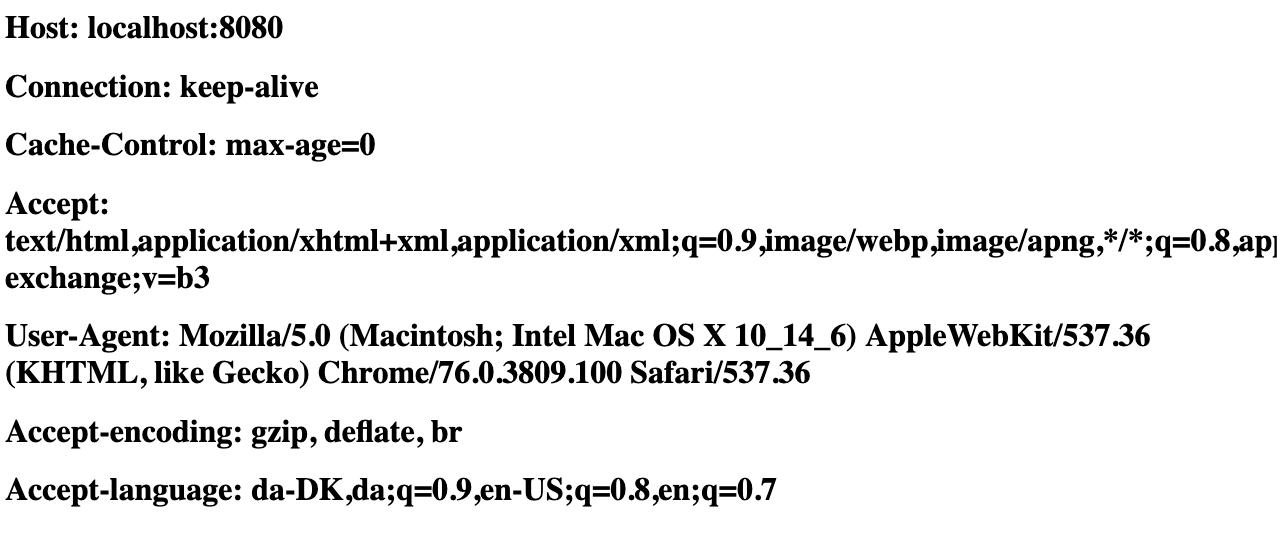
4c) Status Codes - Ranges

1. Informational responses (100–199),
2. Successful responses (200–299),
3. Redirects (300–399),
4. Client errors (400–499),
5. Server errors (500–599).

5) Get HTTP Request Headers on the Server

A close up of text on a black background

Description automatically generated

Denne kode vil give de headers som beskrevet I opgaven, og outputted følgende. Der ikke blevet sat nogle kræfter i præsentation på nogen måde. (som nok kan ses)

Billedet fra oven taget fra <http://localhost:8080/BasicHTTP/info>

6) Get/Post-parameters

GET: 

POST:



Men kan stadig ses i netværk under ”Form Data” som vist forneden.

A screenshot of a cell phone

Description automatically generated

Når man kigger nederest når man kører formen i GET så hedder den i stedet ”Query String Parameters” som set forneden. Her har den samme informationer.

A screenshot of a cell phone

Description automatically generated

Ud fra dette kan vi forstå at GET tager inputtet som parametre og ligger det op i URLen. Mens POST tager inputtet og gemmer det, uden at vise det oppe i URL strengen.

7)      Sessions (Session Cookies)

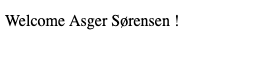
Runner servletten og får følgende frem.



A screenshot of a cell phone

Description automatically generated

Efter at have trykket submit ser URLen således ud: 



Åbner man en ny fane (eller lukker fanen), og taster samme URL ind får man samme velkomstside.

Altså ved at taste: <http://localhost:8080/Sessions/SessionDemo> uden parametre.

Går man ind i developer tools kan man nu se cookies som en fane.

A screenshot of a social media post

Description automatically generated

Man kan ikke komme tilbage til den tidligere side nu som vi så før, grundet denne cookie.

Spørgsmål f: Grunden til at serveren kan holde på navnet som vi har tastet, uden at klienten skal tage et nyt input og sende tilbage til serveren, er fordi at vores ”cookie” er gemt på klienten. Vores browser holder på vores cookie i en cache i udefinerbar tid. Hvis man åbner en anden browser og taster samme adresse, vil man kunne se den første side man så igen.

8)     Persistent Cookies

Mange af billederne ville være ens, så de behøves ikke vises igen.

Det som skifter nu fra den tidligere opgave er, at cookiesne er persisted ind i selve browseren hukkomelse. Hvor tidligere ville den blive slettet hvis vi lukkede for browseren. Vi har sat cookien til at varer i et år, det vil sige går vi ind på den her servlet om et halvt år og starter den op, vil der stadig dukke det samme op.

Det der også har ændret sig, er der ikke længere er kryptiske værdier under cookiefanen som før. Nu ser det således ud.

A screenshot of a social media post

Description automatically generated

Meget mere overskueligt og fylder mindre end tidligere.