LABO 1

**OPDRACHT 1:**

Teken hieronder een schematische voorstelling van de communicatie tussen een client en een server:

Web Server

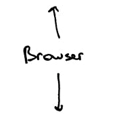
…

Internet

Client

Request

Response



**OPDRACHT 2:**

Duid de verschillende onderdelen van de volgende URL:

<https://www.bol.com/nl/p/hoe‐werkt‐datnou/9200000057347012/?country=BE&suggestionType=browse#product_alternatives>

ONDERDELEN:

https = **protocol**

www = sub domein

bol.com = **host**

bol = domein naam

nl = domein extensie

hoe-werkt-datnou/9200000057347012/ = **path**

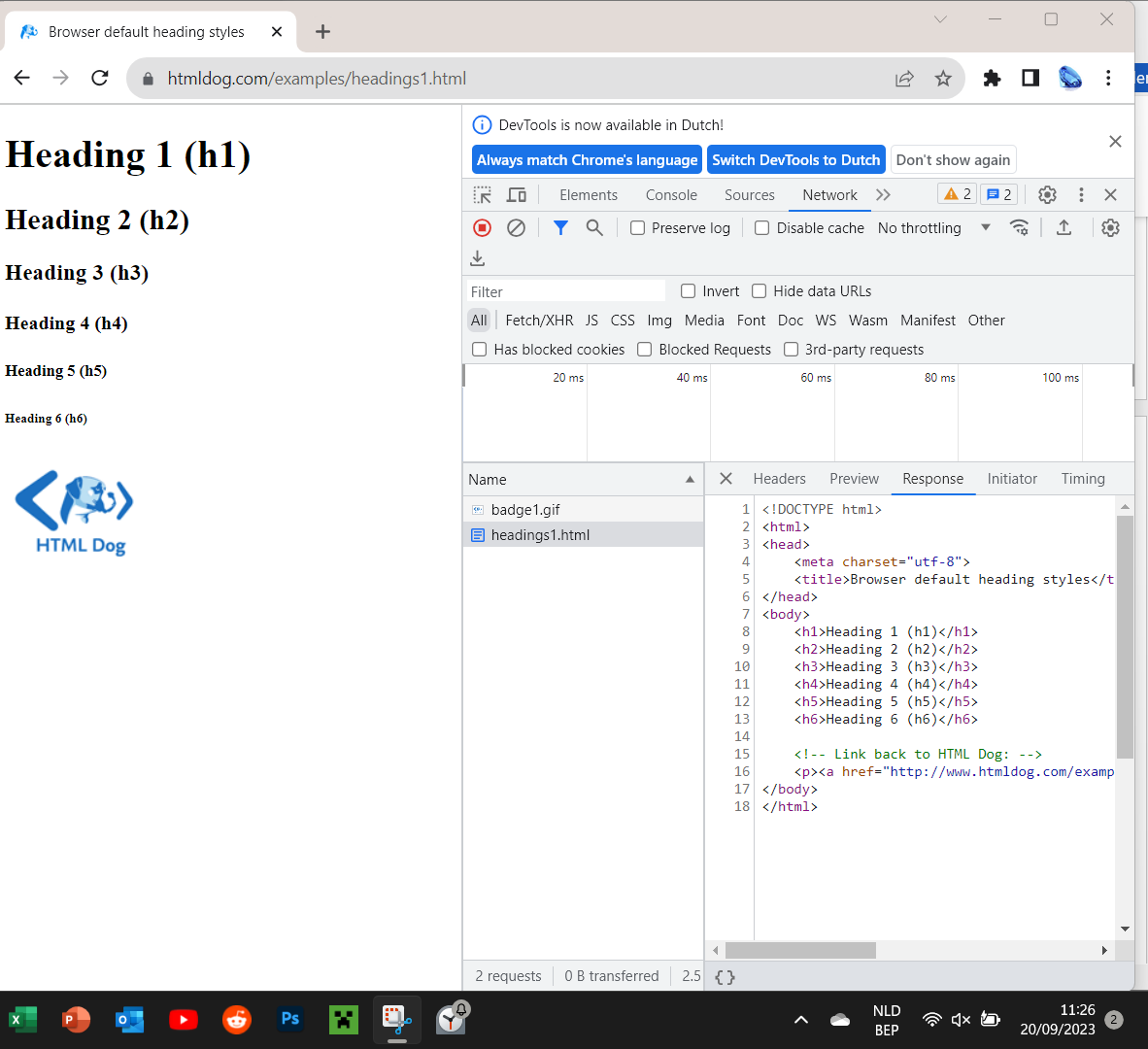
country = **paramater** (= BE)

suggestionType = **parameter** (= browse)

#product alternatives = **fragment**

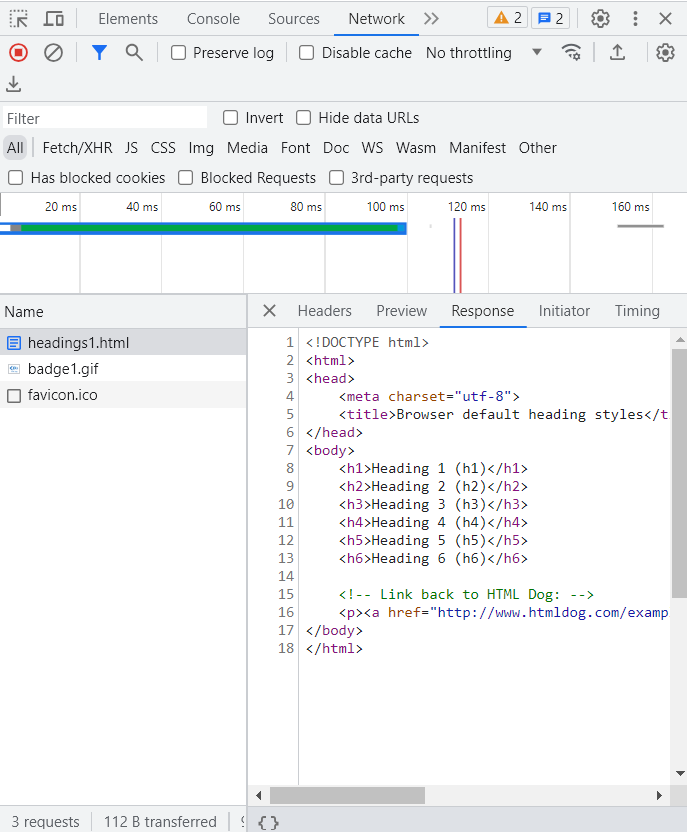
**OPDRACHT 3:**

Open de Chrome developer tools (F12), ga naar het Network tabblad en surf naar:

****

**OPDRACHT 4:**

Welke resources heeft je browser nog meer opgevraagd? Hoe zie je dit?



RESOURCES OPGEVRAAGD:

-badge1.gif

-favicon.ico

HOE KUN JE DIT ZIEN ?

Door te kijken naar Network -> All -> Response en links zien we alle aanvragen

Kun je in het HTML document in het response body terugvinden waarom net die resources werden opgevraagd?

\*Het werd opgevraagd om de pagina titels weer te geven en een link te presenteren die je naar HTML Dog door verwijst

**OPDRACHT 5:**

Welke andere soorten resources worden opgevraagd door het inladen van deze pagina?

Html-, CSS-, xml-, js- files… , cookies, fonts, jpeg, png, img, gif, script, java script

Werden alle requests naar dezelfde server verstuurd?

Nee, sommige naar vives, andere naar de servers van google en nog andere naar die van facebook.

Is dit de server die je bedoelde of is het domein bv. Vives een server ? server: cloudfare ≠ ssfe

Hoe kun je dit uit de timing informatie afleiden?

Ja, hoelang een request moest wachten vooraleer het een response kreeg.

Lijkt het erop dat het aantal gelijktijdige ‘onafgewerkte’ requests beperkt is?

Ja. Gelijktijdige requests werden snel na elkaar beantwoord.

**OPDRACHT 6:**

Waarvoor zouden die 'spontane' requests dienen?

Om spam naar de spam folder te verplaatsen of naar de prullebak.

Om fishing en onbetrouwbare mails uit de inbox te halen.

Om de mailbox te synchroniseren zodat je kunt zien of er nieuwe mails zijn.

**OPDRACHT 7:**

Merk op dat er wel degelijk een response teruggestuurd wordt alhoewel de pagina niet bestaat. Hoe komt dit? Wat betekent de status code 404 in de response header?

404 = pagina niet gevonden terwijl men nog wel met de server kan communiceren/verbinden. Onze browser stuurt een vraag naar de server, waarop de het antwoord van de server is dat de pagina niet kan gevonden worden.

**OPDRACHT 8:**

Wat is het verschil met de vorige opdracht?In tegenstelling tot op. 7, maken we geen verbinding met de server en daardoor krijgt de browser geen antwoord/response terug van de server.

**OPDRACHT 9:**

Zoek onderstaande HTTP status codes op (bv. wikipedia) die in een response kunnen voorkomen en schrijf hun betekenis op. Omcirkel deze die je al eens bent tegengekomen bij het surfen. Schrijf ook een gepaste hoofding boven de kolommen.

\*URI = om iets te identificeren

|  |  |  |  |
| --- | --- | --- | --- |
| SUCCESS | REDIRECTION | CLIENT ERRORS | SERVER ERRORS |
| 200  = OK | 301 = permanente doorverwijzing | 400 = BAD request  (de server kan/wil de request niet verwerken door een client error) | 500 = interne server error |
| 204  = No content  (de server heeft de aanvraag verwerkt en stuurt geen content terug) | 302 = FOUND  (zegt de klant om te browsen naar een andere URL… rederict) | 401 = unauthorized  (authenticatie ontbreekt of werkt niet) | 503 = service unavailable (server kan je request niet aan dr. Overload of maintenance) |
|  | 303 = SEE OTHER  (de request can gevonden worden door een ander URI nl. GET te gebruiken) | 404 = pagina niet gevonden |  |

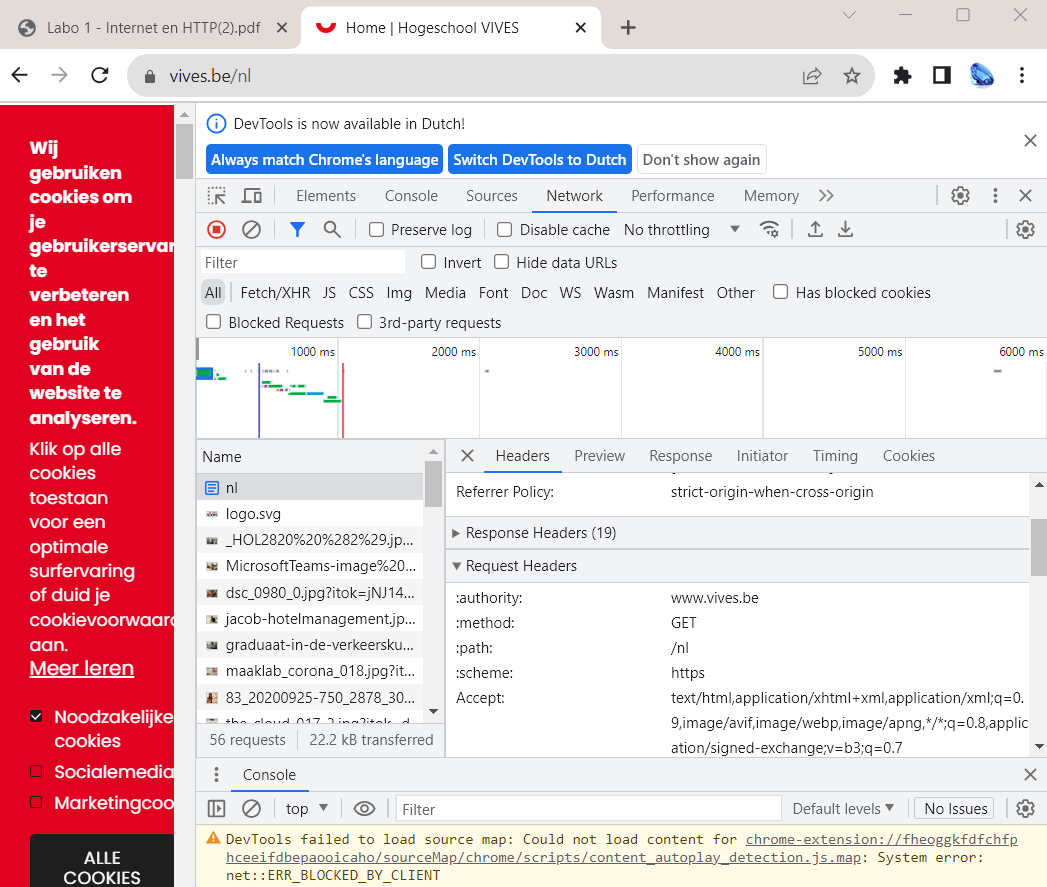
**OPDRACHT 10:**

Zoek op het internet welke HTTP request methods er bestaan en schrijf ze neer. Waarvoor dienen de vaak gebruikte GET en POST methods?

1. GET. 🡪 data van de server terugvinden (HTML file, img/vid., JSON doc., CSS file, JavaScript file, XML file)
2. PUT.
3. POST. 🡪 stuurt data naar de server om die te verwerken (input online forums, XML/JSON files, txt data van query parameters.)
4. DELETE.
5. PATCH.
6. HEAD.
7. OPTIONS.
8. TRACE.
9. CONNECT.

Waar in een request staat aangegeven om welke request method het gaat, en hoe vind je dit terug in de Chrome developer tools (zie uitleg bij Opdracht 3)?

In de request header staat om welke request method het gaat.



Als je een url in de adresbalk van je browser typt en op enter drukt, wat voor request method gebruikt de browser dan om die resource op te vragen bij de server?

De request method: GET

Als je in een webpagina op een gewone hyperlink klikt, welke request method wordt er dan gebruikt?

De request method: POST

Of stel dat de browser een accelerator plugin bevat die proactief gelinkte pagina's inlaadt zodat de gebruiker niet hoeft te wachten bij het klikken op een link. Wat zou er dan gebeuren met de productgegevens?

De product gegevens zouden verloren gaan.

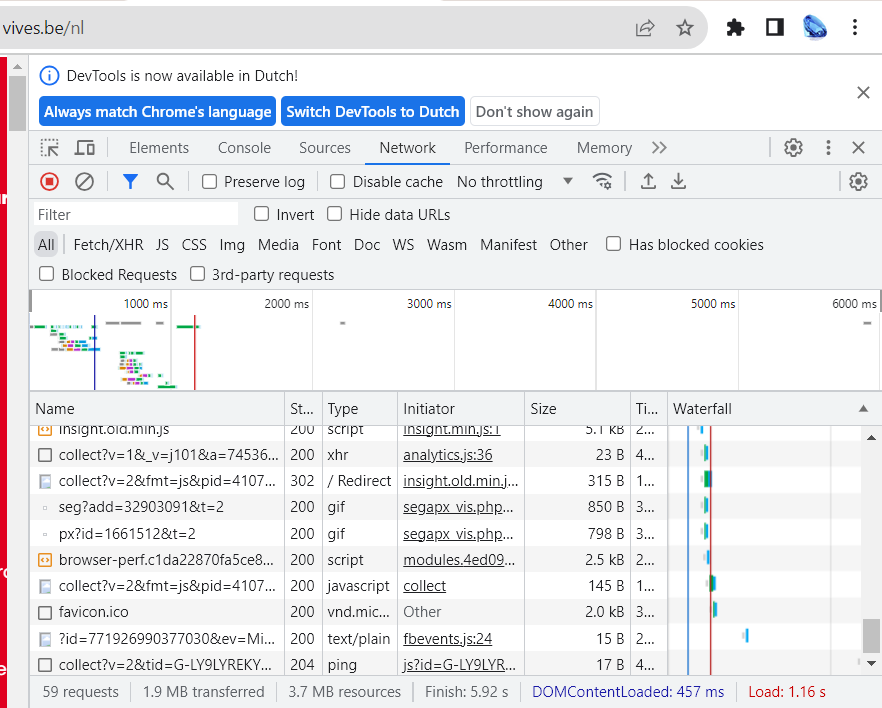
**OPDRACHT 11:**

Bekijk de vele requests die het inladen van die ene pagina heeft veroorzaakt. Hoeveel requests waren er in totaal?

60 requests.

**OPDRACHT 12:**

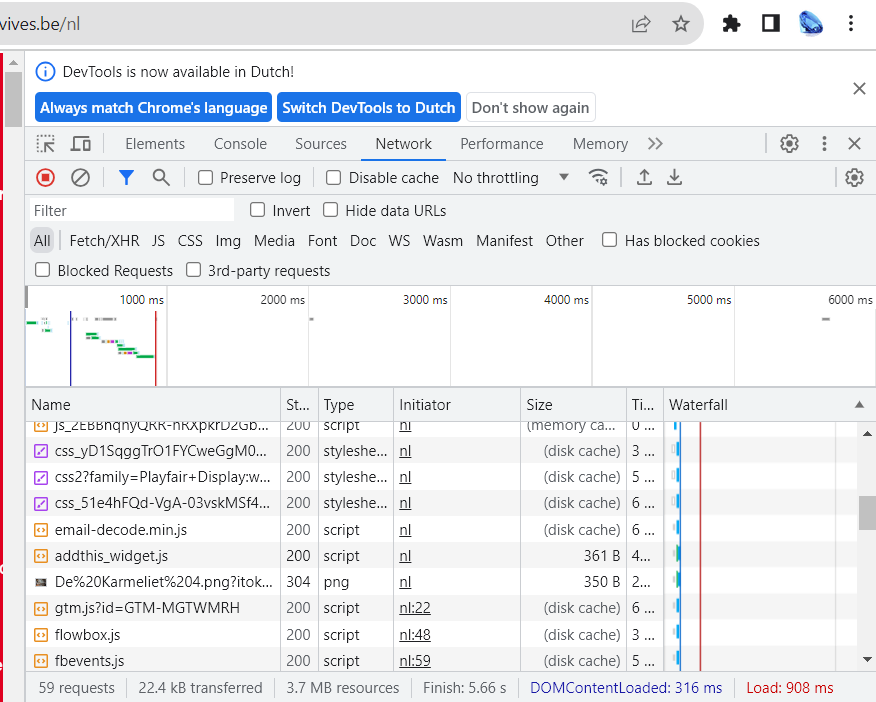
Hoeveel kilobytes of megabytes aan data werd er verstuurd om alle nodige resources in te laden?

1.9 MB transferred 

Hoe lang duurde het vooraleer alle resources van de pagina waren ingeladen?

1.16 s

Klik nu gewoon op de refresh knop. Kijk nogmaals hoeveel data er werd verstuurd. Waarom is dit zoveel minder?



Data verstuurd: 22.4 kB

Waarom zoveel minder ? We gebruiken de opgeslagen gegevens in onze cache en zo moeten we die data niet meer aan de server vragen.

Laadde de pagina sneller?

De pagina laadde in in 908 ms tegenover de 1.16 s van vorige keer.

Waar kun je zien welke documenten daadwerkelijk verstuurd werden en welke niet?

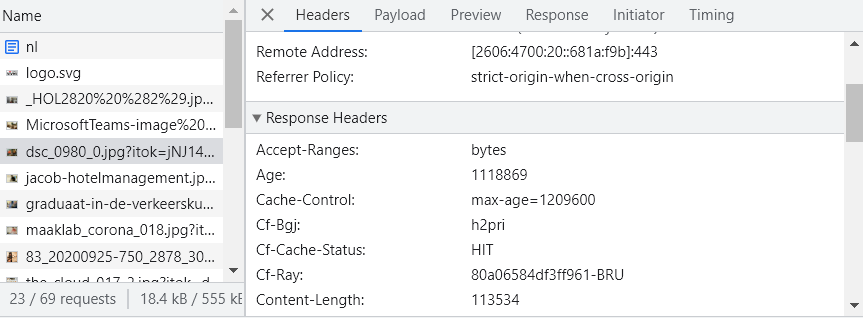
Bij Netwerk onder tab Size kun je zien wat er in cache zat en welke documenten daadwerkelijk verstuurd werden.

Waar vindt de browser dan de inhoud van de documenten die niet bij de server werden opgevraagd?

In de memory cache of disk cache.

Hoe weet de browser welke documenten best opgevraagd (moeten) worden en welke niet? M.a.w. hoe lang mag de browser een bepaalde resource als 'vers' te beschouwen?

Bij Cache-Control: max-age= 1209600



**OPDRACHT 13: IN PROGRESS**

1. 6 (nieuwsblad) Advertising (2), Site Analytics (1), Essential (2), Consent Management (1), CDN (3)
2. 7 (cnn) Advertising (2), Site Analytics (2), Essential (1), Hosting (1), unidentified (4)
3. 8 (vives) Advertising (3), Site Analytics (4), Essential (1)
4. 4 (vrtnws) Advertising (1), Site Analytics (2), Essential (1)