**HTTP & curl**

(1)



* Request-Type + http Version
* Host des Servers
* Anfragender (User, Server etc.)
* verständliche Rückgabe Typen

(2)

curl -I [http://google.de](http://google.de/) curl -I [https://www.google.de](https://www.google.de/)

Unterschiede:

*Grob*:

Aus Sicht von Rechts (- Fehlt; + Ergänzt):

+ p3p:

+ set-cookie

- Location

*Explizit*:

ResponseCode -> 301 Moved Permanently | 200

Cache-Control: public, max-age=259200 | private

Content-Type: text/html; charset=UTF-8 | text/html; charset=ISO-8859-1

(3)

Die Informationen sind im DevTool unter „Network“ zu finden.

(4)

Der erste Aufruf geht nach "[http://google.de](http://google.de/)". Dieser wird abgefangen und als Value um Response-Header verwendet. Somit wird ein Aufruf auf "<http://www.google.de/>" losgetreten.

(5)

Response Preview  
  
gesamte Response als Raw-Format

(6)  
Die Request Method zeigt an um welche Art Request es sich handerlt.

Die mitgelieferten Query Parameter sind Teil der URL, welche aus key-value Paaren bestehn

(7)

Die gesendeten Daten sind im Request-Tab als "PHP-Format" einsehbar.

(8)

HTTP/2 Server Push bedeutet, dass der Server zusätzliche Ressourcen dem Client schickt, bevor dieser spezifisch nach diesen fragt. Dadurch wird versucht, die Latenz zu verringern.

(9)

"keep-alive" behält die Verbindung am Leben. Dies verhindert ein andauerndes neuverbinden und ermöglicht eine Anreihung an Anfragen.

(10)



**Mozilla/5.0:** generelles Token -> Browser ist Mozilla-Kompatibel

**Windows NT 10.0 ... :** Betriebssystem / Plattform

**AppleWebKit/537.36 ... |... Gecko:** Browser Basis

**Chrome/98 ... | Safari/537 ... | OPR/84.0 :** Browser und "Vorgänger" worauf er aufbaut

(11)

Content Type gibt den Typ der zu runterladenden Dateien wie auch mögliche boundaries (bsp. UTF-8) an.

Beim Aufruf auf "<https://www.fh-aachen.de/>" werden Text/html aufgerufen.

**CSS Diner:**

