

HTTP & curl

(1)

```
> GET / HTTP/1.1      1
> Host: google.de     2
> User-Agent: curl/7.60.0 3
> Accept: */*         4
>                     5
```

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

(2)

[illegible]

curl -I <http://google.de>

```
curl -I 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>". 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 handelt.

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

(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)

```
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.109 Safari/537.36 OPR/84.0.4316.52
```

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

