

HTTP-protokolla

Web-sovelluksen käyttöliittymä ja palvelimella sijaitseva sovelluslogiikka viestivät keskenään HTTP-protokollan (HyperText Transfer Protocol) välityksellä.

HTTP-viestien rakennetta ja otsakkeita hyödyntämällä voidaan käyttää mm.

- evästeitä,
- autentikointimenetelmiä ja
- istunnon hallintaa.

HTTP-siirtotapahtuma

Web-palvelin kuuntelee web-asiakkaiden asiakaspyyntöjä valitussa TCP-portissa. HTTP-siirtotapahtuma alkaa siitä, kun web-palvelin saa asiakkaalta pyynnön. Web-palvelin käsittelee sen ja palauttaa selaimelle datan. Selain esittää sitten saadun datan.

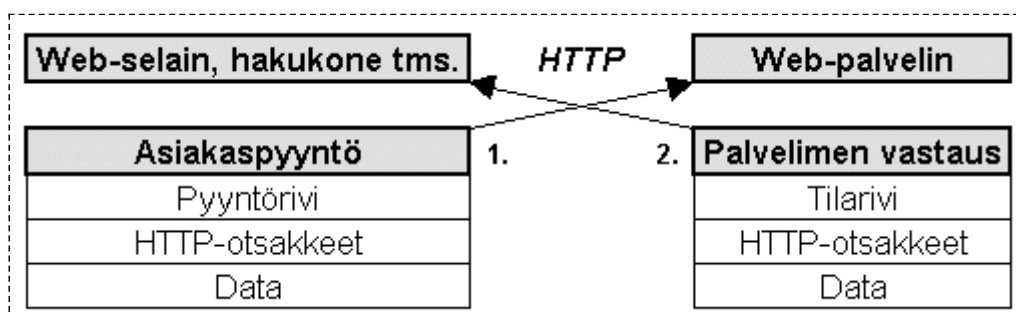
HTTP-siirtotapahtuma koostuu kahdesta HTTP-viestistä (HTTP message), jotka ovat:

1. Asiakaspyyntö (HTTP request), jonka sisältönä on

- Pyyntöriivi (Request-Line)
- HTTP-otsakkeet
- Data

2. Palvelimen vastaus (HTTP response), jonka sisältönä on

- Tilarivi (Status-Line)
- HTTP-otsakkeet
- Data



Asiakaspyyntö

HTTP-asiakaspyyntö etenee seuraavassa muodossa.

[1] Asiakas ottaa yhteyden halutun palvelimen tiettyyn porttiin (yleensä 80)

[2] Asiakas lähettää HTTP-komennon (pyyntörivi), jonka kolmesta välilyönnillä erotetusta kentästä löytyy

- käytettävä metodi
- pyydetty dokumentti
- HTTP:n versio

Esimerkki:

```
GET /manual/ HTTP/1.0
```

[3] Asiakas lähettää mahdolliset HTTP-otsakkeet riveittäin. Kukin rivi koostuu otsakkeen nimestä, kaksoispisteestä, välilyönnistä ja otsakkeen arvosta ja rivinvaihtomerkillä (CRLF). Esimerkiksi Web-selain voi kertoa näin nimensä, hyväksymänsä dokumenttityypit, toivomansa kielen jne.

Esimerkki:

```
ACCEPT_LANGUAGE = en-US,en;q=0.9
```

[4] Asiakas lähettää rivinvaihtomerkin (CRLF ~ tyhjä rivi)

[5] Asiakas **voi** lähettää muuta dataa. Näin tapahtuu esimerkiksi silloin, kun Web-selain lähettää HTML-lomakkeen sisältämän datan POST-metodilla.

[6] Pyyntö päättyy kahteen peräkkäiseen rivinvaihtomerkkiin. Jos tämä tehdään jo otsakkeiden lähettämisen jälkeen kohdassa 4, niin kohdan 5 muuta dataa ei tällöin lähetetä.

Palvelimen vastaus

Kun asiakas on lähettänyt HTTP-pyyntöä, palvelin vastaa siihen. Palvelimen vastaus etenee seuraavassa muodossa:

[1] Palvelin vastaa tilarivillä, jonka kolmesta välilyönnillä erotetusta kentästä löytyy

- HTTP:n versio
- tilakoodi
- tilakoodin tekstimuotoinen selitys

Tyypillisesti onnistuneeseen pyyntöön vastataan:

```
HTTP/1.0 200 OK
```

[2] Palvelin (tai palvelinskripti) lähettää mahdolliset HTTP-otsakkeet riveittäin. Kukin rivi koostuu otsakkeen **nimestä**, kaksoispisteestä, välilyönnistä ja otsakkeen **arvosta** ja **rivinvaihtomerkillä** (CRLF). Tyypillisesti **voidaan** kertoa

esimerkiksi vastauksen luontiaika, tietoja palvelimesta ja palautettavan dokumentin sisältötyyppi.

Esimerkki:

Date: Mon, 09 Sep 2019 05:10:28 GMT

Server: Apache/2.4.6 (CentOS)

Content-Type: text/html; charset=UTF-8

[3] Palvelin lähettää rivinvaihtomerkin (CRLF ~ tyhjä rivi)

[4] Pyynnön onnistuessa vastaukseksi lähetetään mahdollisesti haluttu data, joka voi olla esimerkiksi

- staattinen tiedosto: HTML-dokumentti, kuvatiedosto, Excel-asiakirja, zip-tiedosto jne.
- PHP-skriptin generoima HTML-dokumentti, kuvatiedosto jne.

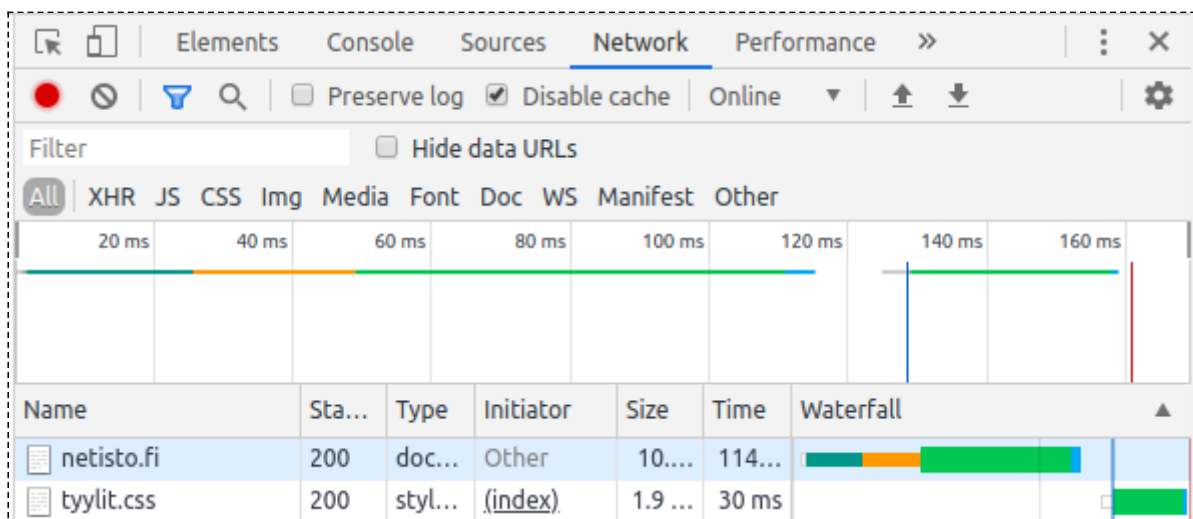
Web-sovellus rakentuu peräkkäisistä HTTP-siirtotapahtumista

HTTP on tilaton protokolla. Peräkkäisten siirtotapahtumien välillä ei ole mahdollista pitää tilatietoa protokollan avulla. Näin ollen

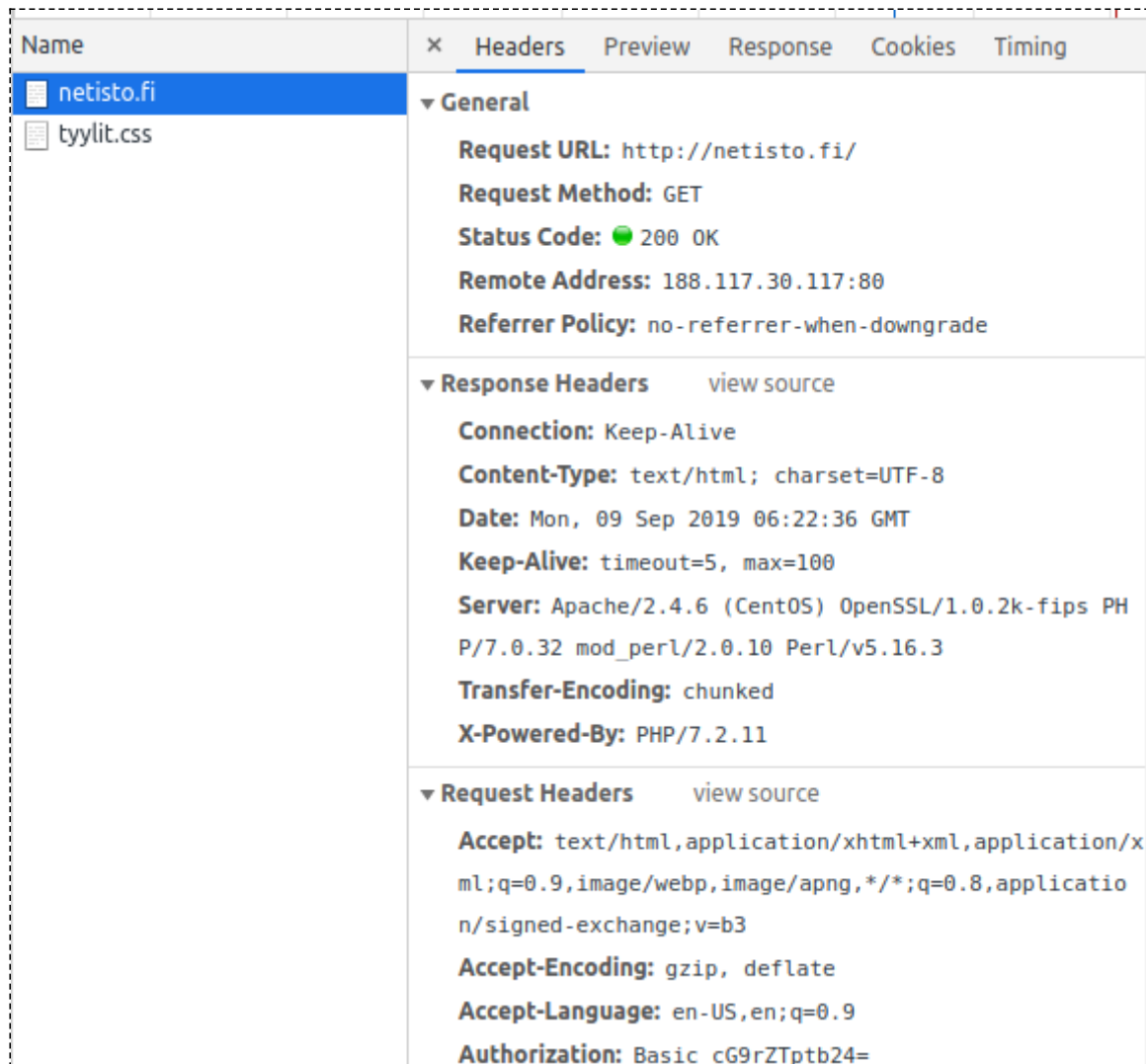
- tapahtumien seuranta ei kuormita HTTP:tä -> kevyt protokolla
- jokainen HTTP-siirtotapahtuma on aloitettava aivan alusta -> Web-sovellusten ohjelmointi on työlästä

HTTP-liikenteen seuraaminen

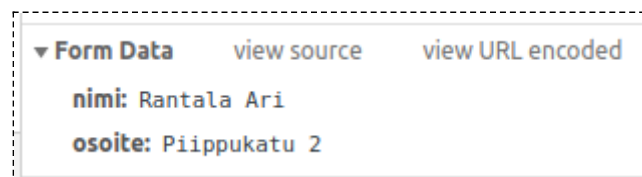
Käynnistä Chrome-selaimessa web-konsoli (F12) ja Network-välilehti. Liikenteen seurannassa Disable cache -valinnan aktivointi on usein paikallaan. Ladattaessa esim. sivu netisto.fi näemme että HTML-sisältö (netisto.fi) ja tyylitiedosto ladattiin erillisillä HTTP-siirtotapahtumilla



Klikkaamalla netisto.fi-sisältöä nähdään mm. HTTP-siirtotapahtumassa välitetyt HTTP-otsakkeet



Jos lähetettynä olisi ollut POST-metodin sisältämä (aiemmin esitetty) syöttölomake.html, niin pyynnön yhteydessä nähtäisiin myös näin lähetetty data:



HTTP-otsakkeita

[Jukka Korpelan Quick reference to HTTP headers](#) dokumentti listaa HTTP/1.1-protokollaan liittyvät HTTP-otsakkeet. Muutama esimerkki:

- **Last-Modified:** Dokumentin viimeinen muokkaushetki.

Selain voi kysyä esim otsakkeella If-Modified-Since: Tue, 27 Feb 2007 07:36:27 GMT : Anna minulle dokumentti vain siinä tapauksessa, ettei sitä ole muutettu edellisen pyynnön jälkeen (haetaan esim. selaimen välimuistista)

- **Etag:** Entity Tag. Sivun sisällön tunniste. Jos dokumentin sisältö muuttuu, niin tunnisteikin muuttuu.

Selain voi tämänkin avulla kysyä esim. otsakkeella If-None-Match: "712c3db-32-50ca94c0" : Anna minulle dokumentti vain siinä tapauksessa, että sen sisältö on muuttunut.

- **Accept-Ranges:** Palvelin sallii tavuina määritellyt pyynnöt joissa halutaan vain osa dokumentista.
- **Content-Length:** Lähetettävän datan koko.
- **Content-Type:** Palautettavan sisällön mediatyyppi ja halutessa merkitököoodaus (UTF-8). Selaimen tulisi tunnistaa datan sisältö tämän perusteella eikä tiedostopäätteen perusteella.

HTTP-siirtotapahtumia komentoriviltä

HTTP-siirtotapahtumia voi demonstoida vaikkapa Linuxin komentoriviltä telnet-ohjelman avulla

Esimerkki 1 - GET-metodi

- Kirjoitetaan esim. Linuxin komentoriviltä komento telnet netisto.fi 80 (rivi 1), jolla avataan yhteys käytettävän palvelimen porttiin 80.
- Yhteyden syntymisen merkiksi telnet-ohjelma tulostaa kolme riviä ja palvelin jää odottamaan, mitä asiaa asiakkaalla on.
- Tässä tapauksessa pyydetään Web-palvelimen juurihakemistosta dokumenttia hello.html (kirjoittamalla rivi 5)
- HTTP/1.1-protokollassa on pakko kirjoittaa Host-otsake (rivi 6)
- Asiakaspyyntö kuitataan toisellakin rivinvaihdolla (rivi 7)
- Riveillä 8-15 on palvelimen lähettämät HTTP-otsakkeet
- Riveillä 17-19 on palvelimen lähettämä data eli varsinainen verkkosivun HTML-muotoinen sisältö

```
1  $ telnet netisto.fi 80
2  Trying 188.117.30.117...
3  Connected to netisto.fi.
4  Escape character is '^]'.
5  GET /hello.html HTTP/1.1
6  Host: netisto.fi
7
8  HTTP/1.1 200 OK
9  Date: Mon, 09 Sep 2019 06:48:06 GMT
10 Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.0.32
11 Last-Modified: Mon, 09 Sep 2019 06:47:47 GMT
12 ETag: "5f-592192a6f1421"
13 Accept-Ranges: bytes
```

```
14 Content-Length: 95
15 Content-Type: text/html
16
17 <title>hello.html</title>
18 <h1>Hello World!</h1>
19 <p>Olen tavallinen HTML-sivu hello.html</p>
20
21 Connection closed by foreign host.
22 $
```

Esimerkki 2 POST-metodi

POST-metodin tapauksessa HTTP-pyyntö voisi näyttää seuraavalta. Rivit 1 ja 5-11 ovat käyttäjän kirjoittamia. Varsinainen lomakkeen data lähetetään yhden rivinvaihdon (rivi 10) jälkeen rivillä 11. Esimerkkipyynnön tapauksessa HTML-lomakkeessa olisi ollut nimi- ja osoite-nimiset kentät, jotka käyttäjä olisi täyttänyt rivillä näkyvillä arvoilla.

```
1 $ telnet 127.0.0.1 80
2 Trying 127.0.0.1...
3 Connected to 127.0.0.1.
4 Escape character is '^]'.
5 POST /nayta_http_data.php HTTP/1.1
6 Host: localhost
7 Content-type: application/x-www-form-urlencoded
8 Content-length: 26
9 Connection: close
10
11 nimi=Ari&osoite=Piippukatu
12 ...
```