

Otsakkeet PHP:llä - header()

HTTP-otsakkeita voi lähettää PHP:n `header()` -funktiolla. Tyypillisesti halutaan tehdä vaikkapa uudelleenohjaus Location-otsakkeella:

```
1 <?php
2 header("Location: http://www.google.com/");
3 ?>
```

Omissa sovelluksissa on usein tarve uudelleenohjata oman sovelluksen toiseen sijaan. HTTP/1.1 vaatii käytettäväksi absoluuttisen URIn Location-otsakkeen arvona

```
1 <?php
2 header("Location: http://" . $_SERVER['HTTP_HOST']
3       . dirname($_SERVER['PHP_SELF']) .
4       . "nayta-http-data.php");
5 ?>
```

HTTP-otsakkeet tulisi pyrkiä lähettämään ohjelmassasi protokollan mukaisesti **ennen** muuta verkkosivun sisältöä. Jos näin ei toimita, palvelinohjelmisto yleensä yrittää korjata tilanteen omalla tavallaan. Esim. seuraava ohjelma ei toimi kaikissa palvelinkonfiguraatioissa:

```
1 <title>HTTP-otsake -testi</title>
2 <h1>Tämä ei aina toimi</h1>
3 <p>
4 Joissakin palvelimissa tällainen sivu aiheuttaa
5 virheilmoituksen "headers already sent"
6 </p>
7 <?php
8 header("Location: http://www.google.com/");
9 ?>
```

Ohjelman

```
1  <?php
2  header("Content-Type: text/html");
3  echo "<h1>HÖTÖMÖLÖ-otsikko</h1>";?>
```

tulisi muotoilla H1-otsikko selaimeen tavanomaisesti, mutta ohjelman

```
1  <?php
2  header("Content-Type: text/plain");
3  echo "<h1>HÖTÖMÖLÖ-otsikko</h1>";
4  ?>
```

tulisi tulostaa selainikkunaan merkit `<h1>HÖTÖMÖLÖ-otsikko</h1>` ilman HTML-muotoilua, koska ohjeena on näyttää se pelkkänä tekstinä

Download dialog -esimerkki

Selaimen saa kehottamaan lähettämänsä datan tallentamista **Content-Disposition**-otsakkeen avulla:

```
1  <?php
2  // Lähetetään PDF-dokumentti
3  header('Content-type: application/pdf');
4
5  // Käyttäjä näkee tiedoston nimellä downloaded.pdf
6  header('Content-Disposition: attachment; filename="downloaded.pdf"');
7
8  // Alkuperäinen tiedosto voidaan generoida "lennosta" tai lukea
9  readfile('print.pdf');
10 ?>
```

Web-kansion tarjoaminen ZIP-pakettina

```
1  <?php
2  // dir-zipper-downloader.php
3
```

```

4  $hakemisto = dirname($_SERVER['SCRIPT_FILENAME']);
5
6  system("zip -r $hakemisto/paku.zip *");
7  header('Content-type: application/zip');
8  header('Content-Disposition: attachment; filename="paku.zip"');
9
10 readfile('paku.zip');
11 unlink("$hakemisto/paku.zip");
12
13 ?>

```

HTTP-pyyntöt php:llä - fsockopen()

HTTP-pyyntöjä otsakkeineen voi lähettää myös PHP:n `fsockopen()` -funktiolla. Oheisessa esimerkissä muodostetaan ja lähetetään http-pyyntö kaikkine tarvittavine osineen. HTTP-vastaus luetaan ja tulostetaan selaimeen kaikkineen. Sivulle tulostuvat tällöin myös palvelimen lähettämät HTTP-otsakkeet, koska niitä ei suodateta pois

```

1  <?php
2  $kohde_kone = "netisto.fi";
3  $kohde_portti = 80;
4  $kohde_polku = "/index.php";
5  $pyyntometodi = "GET";
6
7  $pyyntorivi = "$pyyntometodi $kohde_polku HTTP/1.1";
8  $otsakkeet = "Host: $kohde_kone";
9  $otsakkeet .= "Connection: Close";
10
11 $fp = fsockopen($kohde_kone, $kohde_portti, $errno, $errstr,
12 if (!$fp) {
13     echo "$errstr ($errno)<br />";
14 } else {
15     //Lähetetään HTTP-pyyntö ja otsakkeet
16     fputs($fp, $pyyntorivi);
17     fputs($fp, $otsakkeet);
18
19     // Luetaan ja tulostetaan HTTP-vastaus
20     while (!feof($fp)) {
21         echo fgets($fp, 256);

```

```
22     }  
23     fclose($fp);  
24 }  
25 ?>
```

web-soketit

Web-sokettien avulla voi luoda pysyvän yhteyden selaimen ja palvelimen välille. Näin palvelimen ja selaimen välinen viestintä suoraviivaistuu tiettyjä sovellustarpeita silmälläpitäen. Esim. monipelattavat verkkopelit vaativat nopeaa pelitilanteen jatkuvaa päivitystä. Myös erilaiset chat-sovellukset vaativat viiveetöntä tilan päivitystä.

Tällä opintojaksolla ei toistaiseksi käsitellä web-soketteja, mutta niiden hyödyntäminen vaikkapa harjoitustyössä on täysin luvallista.

- <https://en.wikipedia.org/wiki/WebSocket>