
Istunnon hallinta

- Istunto = Sellainen toisiinsa yhteenkuuluva kokoelmaa HTTP-pyyntöjä ja -vastauksia, joille voidaan määritellä aloitus- ja lopetustapahtuma.
- PHP:ssä sisäänrakennettuna välineet istunnon hallintaan

Usein istunto alkaa sisäänkirjautumisella, jonka yhteydessä käyttäjä autentikoituu saadakseen valtuutuksen niihin toimenpiteisiin, mitä istunnon aikana on tarkoitus tehdä. Tällainen istunto voi muodostua esimerkiksi asioinnista verkkopankissa: Istunto alkaa sisäänkirjautumisesta ja päättyy uloskirjautumiseen

Istuntokohtainen tunniste evästeenä

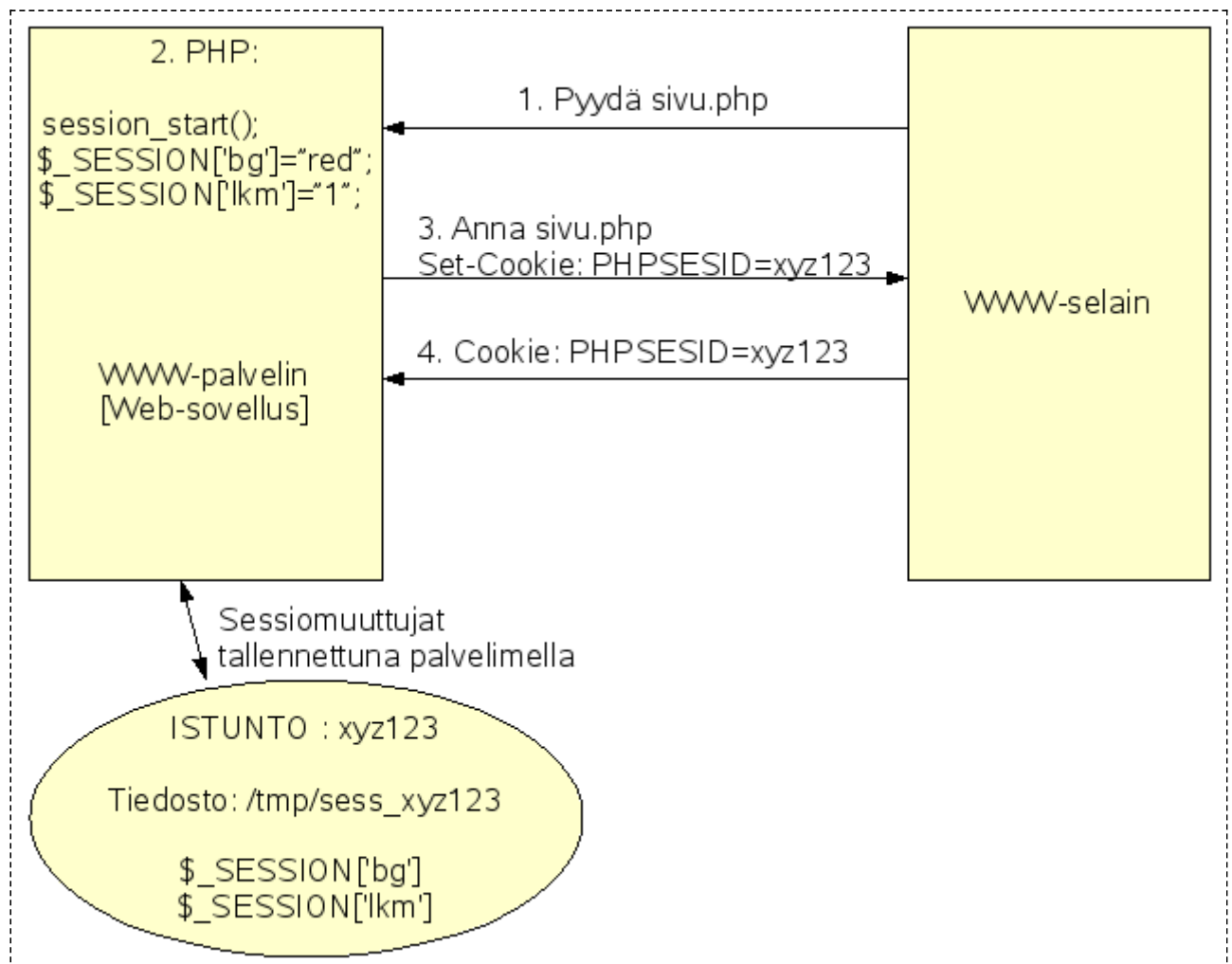
Tyypillisesti istuntoa ylläpidetään seuraavan periaatteen mukaan

- Generoidaan aluksi (istuntokohtainen) tunniste ja kuljetetaan sitä **evästeenä** jokaisen HTTP-siirtotapahtuman yhteydessä.
- Palvelinskripti vertaa jokaisen HTTP-pyyntöä yhteydessä selaimelta saamaansa tunnistetta (eväste) palvelimelle tallennettuun vastineeseen ja liittää siihen mahdollisesti halutussa tietovarastossa (tekstitiedosto, relaatiotietokanta) säilytetyn muun datan.

Ainutkertaisen istuntotunnisteen etuja

- Mahdolliset autentikointitiedot välitetään ainoastaan kerran ja niitä ei tallenneta selaimen evästeisiin
- Tunnisteen joutuminen ulkopuolisille istunnon jälkeen ei ole ongelmallista
- Logout toteutettavissa luotettavasti: Tunniste voidaan poistaa käyttäjän toimesta tai se voi yksinkertaisesti vanhentua
- Pyyntöjen yhteydessä kuljetettava datamäärä on pieni (pelkkä tunniste)

PHP:n sisäänrakennettu istunnon hallintamekanismi



Tarkastellaan aluksi ohjelmaa:

```
1  <?php
2  // sessio-v01.php
3
4  session_start();
5
6  // $_SESSION['lkm'] alustetaan ensimmäisellä kerralla
7  if (!isset($_SESSION['lkm'])) $_SESSION['lkm'] = 0;
8
9  // Lisätään tämä sivulataus
10 $_SESSION['lkm']++;
11
12 // Tulostetaan latauskerrat
13 echo "Olet ladannut sivun <b>".$_SESSION['lkm'].</b> kertaa!";
```

Käytettäessä ohjelmaa huomataan

- lkm-laskurin `$_SESSION['lkm']` arvon kasvaminen latauksien mukaan
- lkm-laskuri ei menetä tietojansa, vaikka välillä vierailtaisiin muilla sivuilla

- jos selain suljetaan ja käynnistetään uudelleen (kaikki ikkunat ja välilehdet), lkm-laskurin lukema alkaa juosta alusta

Ohjelma toiminta on seuraava:

- Ensimmäisellä HTTP-pyynnöllä `session_start()` -funktio luo palvelimen kannalta yksikäsitteisen tunniste. Tämä voi olla esimerkiksi `sess_21418d6f7bbb11ff3c4ffe07ddcdf71e`, joka sekä tallennetaan palvelimelle että asetetaan (oletusarvoisesti) selaimen PHPSESSID-nimisen evästeen arvoksi.
- Kun ohjelmaa kutsutaan toista tai useampaa kertaa siten, että selain lähettää istuntotunniste, `session_start()` -funktio palauttaa tähän istuntoon liittyvän datan käytettäväksi `$_SESSION`-taulukossa. Tässä tapauksessa muuttujan `$_SESSION['lkm']` arvo palautetaan istuntoon tallennetusta säilöstä.
- Evästeen elinaika on oletuksena 0, joten selain poistaa sen sammutuksensa yhteydessä. Kun selain käynnistetään uudelleen ja ohjelmaa kutsutaan, sen on luotava uusi istuntotunniste, johon liittyvä data on oletusarvoissaan.

`student.labranet.jamk.fi`-palvelimella istuntokohtainen data säilytetään `~/php_sessions`-kansiossa säilytettävissä tiedostoissa:

```
1 | $ ls ~/php_sessions/sess_*
2 | /tmp/sess_21418d6f7bbb11ff3c4ffe07ddcdf71e
3 | /tmp/sess_e7338470c812f9f47f15084dae8049bd
```

Jos eri istuntoihin liittyviä laskureita on pyöritetty vaikkapa 6 ja 3 kertaa, tiedostojen sisällöt näyttäisivät seuraavilta:

```
1 | $ more /tmp/sess_21418d6f7bbb11ff3c4ffe07ddcdf71e
2 | lkm|i:6;
3 | $ more /tmp/sess_e7338470c812f9f47f15084dae8049bd
4 | lkm|i:3;
```

Esimerkkejä

Pyyntöjen välisen ajan mittaaminen

Tarkastellaan ohjelmaa, joka esittää käyttäjälle sekunteina sen ajan, joka on kulunut edellisestä sivun latauksesta eli HTTP-pyyntöä. Ohjelma näyttää ensimmäisellä kerralla kuluneeksi ajaksi 0 sekuntia. Helppona harjoituksena voit miettiä kuinka ensimmäiselle lataukselle ei näytettäisi kulunutta aikaa ollenkaan.

```

1  <?php
2  // sessio-v02.php
3
4  session_start();
5
6  // Alustus ensimmäisellä kerralla
7  if (!isset($_SESSION['edellinen_aika'])) {
8      $_SESSION['edellinen_aika'] = time();
9  }
10
11 // Lasketaan kulunut aika. Ekalla kerralla näytetään
12 // kuluneeksi ajaksi 0 sekuntia
13 $aika_nyt = time();
14 $kulunut_aika = $aika_nyt - $_SESSION['edellinen_aika'];
15
16 echo "Edellisestä vierailusta on kulunut ";
17 echo "<b>$kulunut_aika</b> sekuntia";
18
19 // Nykyinen aika seuraavan pyynnön edelliseksi
20 $_SESSION['edellinen_aika'] = $aika_nyt;
21 ?>

```

Istunnon elinkaaren hallinta

Älä takerru ohjelman yksityiskohtiin. Sovelluksesta on tarkoitettu havaittavaksi seuraavat asiat:

- käyttäjällä on mahdollisuus sekä sulkea että avata istunto
- istunto suljetaan automaattisesti, jos käyttäjä on lähettämättä pyyntöjä yli ennalta asetetun ajan
- istunnon poistamiseen liittyvät toimet

Ohjelman lähdekoodi:

```

1  <?php
2  // sessio-v03.php
3
4  /***** Pääohjelma *****/
5
6  session_start();
7
8  $aika_nyt = '';

```

```

9  $vanha_aika    = '';
10 $kulunut_aika  = '';
11
12 istunnon_tila($aika_nyt, $vanha_aika, $kulunut_aika);
13 tulosta_istuntodata($aika_nyt, $vanha_aika, $kulunut_aika);
14 nayta_painikkeet();
15
16 /***** Funktiot *****/
17
18 function istunnon_tila(&$aika_nyt,
19                       &$vanha_aika,
20                       &$kulunut_aika) {
21
22     $action = (isset($_REQUEST['action'])) ?
23               $_REQUEST['action'] : '';
24     $aika_nyt = time();
25     if (isset($_SESSION['edellinen_aika'])) {
26         $kulunut_aika = $aika_nyt - $_SESSION['edellinen_aika'];
27     } else {
28         $_SESSION['edellinen_aika'] = 0;
29         $kulunut_aika = 0;
30         $_SESSION['laskuri'] = 0;
31     }
32
33     // Edellinen aika talteen tulostamista varten.
34     $vanha_aika = $_SESSION['edellinen_aika'];
35
36     if (($action == "Sulje istunto")
37         OR ($kulunut_aika > 10)) {
38         poista_istunto_totaalisesti();
39         echo "<h3>Istunto on suljettu</h3>";
40     } else {
41         $_SESSION['edellinen_aika'] = $aika_nyt;
42         $_SESSION['laskuri']++;
43         echo "<h3>Istunto auki. Vastaus nro." .
44             "{$_SESSION['laskuri']}</h3>";
45     }
46 }
47
48 function tulosta_istuntodata($aika_nyt,
49                              $vanha_aika,
50                              $kulunut_aika) {
51

```

```

52 if (!isset($_COOKIE['PHPSESSID'])) {
53     $_COOKIE['PHPSESSID'] = '';
54 }
55
56 $minpit = 14;
57 $maxpit = 36;
58 $tayte = ".";
59
60 // Katso merkkijonojen muotoiluun liittyvät asiat
61 // merkkijonoja käsittelevästä luvusta:
62 $format = "%'$tayte-$minpit.$maxpit" . "s";
63 $format .= "%'$tayte" . "36s
64 ";
65
66 echo "<pre>";
67 printf ($format, "Aika nyt", $aika_nyt);
68 printf ($format, "Vanha aika", $vanha_aika);
69 printf ($format, "Kulunut aika", $kulunut_aika);
70 printf ($format, ".", ".");
71 printf ($format, "session_id()", session_id());
72 printf ($format, '$_PHPSESSID', $_COOKIE['PHPSESSID']);
73 echo "</pre>";
74 }
75
76 function nayta_painikkeet() {
77     ?>
78     <hr>
79     <form action="<?php echo ($_SERVER['PHP_SELF'])?>" method="po
80     <input TYPE="submit" name="action" value="Sulje istunto">
81     <input TYPE="submit" name="action"
82     value="Avaa/Jatka istuntoa">
83     </form>
84     <?php
85     }
86
87 // Poistetaan kaikki istuntoon liittyvä data
88 function poista_istunto_totaalisesti() {
89     // Tyhjätään $_SESSION-taulukko
90     $_SESSION = array();
91     // Poistetaan istuntoon liittyvä eväste
92     if (isset($_COOKIE[session_name()])) {
93         setcookie(session_name(), '', time()-86400, '/');
94     }

```

```
95     // Poistetaan itse istunto
96     session_destroy();
97 }
98 ?>
```

Jätetty tarkoituksella tyhjäksi

© #AriRantala