

Cookies & Sesija

Pirmiausia, norėdami suprasti, kodėl mums reikalinga sesija, pereikime prie ištakų – prie HTTP protokolo.

HTTP Protocol

HTTP protokolas - "HyperText Transfer Protocol" - hiperteksto perdavimo protokolas – tai yra tekstinis protokolas ir jį suprasti neturėtų būti sunku.

Iš pradžių buvo manoma, kad šiuo protokolu bus perduodamas tik HTML, o dabar ko tik nesiunčiama...

Užklausa:

```
GET / HTTP/1.1
Host: example.com
Accept: text/html

<empty string>
```

Atsakymas:

```
HTTP/1.1 200 OK
Content-Length: 1983
Content-Type: text/html; charset=utf-8

<html>
<head>...</head>
<body>...</body>
</html>
```

Tai labai supaprastinti pavyzdžiai, bet čia galite pamatyti, ką sudaro HTTP užklausa ir atsakymas:

1. **pradžios eilutė** - užklausoje yra prašomo puslapio metodas ir kelias, atsakymui - protokolo versija ir atsakymo kodas
2. **headers** (antraštės) – turi key-value formatą, atskirtą dvitaškiu, kiekviena nauja antraštė

rašoma naujoje eilutėje

3. **body** (turinys) – HTML arba duomenys nuo antraščių atskirti dviem eilučių lūžiais (`\n\n` arba `\r\n\r\n`). Tačiau jų ir gali nebūti.

Protokolo esmė yra "išsiuntė užklausa - gaukite atsakymą", ir viskas, serveris ir klientas niekaip nesusiję.

Tačiau toks scenarijus anaip tol ne vienintelis galimas, mes galime turėti autorizaciją, serveris turi kažkaip suprasti, kad ši užklausa atėjo iš konkretaus vartotojo, t.y. klientas ir serveris turi bendrauti per tam tikrą seansą. Ir taip buvo sugalvotas sekantis mechanizmas:

1. Autorizuodamas vartotoją, serveris sugeneruoja ir įsimena unikalų raktą – seanso identifikatorių ir perduoda jį į naršyklę
2. Naršyklė išsaugo šį raktą ir siunčia jį su kiekviena sekančia užklausa

Šiam mechanizmui įgyvendinti buvo sukurti cookie (slapukai) – paprasti tekstiniai failai jūsų kompiuteryje, po vieną failą kiekvienam domenui (nors kai kurios naršyklės saugojimui naudoja SQLite duomenų bazę). Reikia turėti omenį kad naršyklė nustato apribojimą įrašų skaičiumi ir saugomų duomenų dydžiu (dauguma naršyklių yra 4096 baitai, žr. [RFC 2109](#) nuo 1997 m.)

PHP ir sesija

PHP buvo sukurtas taip, kad atitiktų HTTP protokolą – t.y. jo pagrindinė užduotis yra atsakyti į HTTP užklausa ir „mirti“ atlaisvinant atmintį ir išteklius. Todėl PHP seanso mechanizmas veikia ne automatiškai, o rankiniu režimu ir reikia žinoti ką ir kokia tvarka naudoti.

[PHP is meant to die](#)

Pradžiai reikia „startuoti“ sesiją – tam naudojame `session_start()` funkciją.

```
<?php
session_start();
```

Paleidus naršyklėje šį srktiptą hederiose pamatysime:

Set-Cookie: PHPSESSID=dap83arr6r3b56e0q7t5i0qf91; path=/

Tai pamačiusi naršyklė išsaugos slapuką pavadinimu `PHPSESSID`

PHPSESSID — numatytasis pagal nutylėjimą sesijos pavadinimas, kuris nustatomas `php.ini` faile naudojant direktyvą `session.name`. Jei reikia pavadinimą galima pakeisti naudojant funkciją `session_name()`.

Dabar naršyklė su kiekviena nauja užklausa siunčia šią reikšmę į serverį.

Sekantis žingsnis - išsaugoti sesijoje kažkokią reikšmę. Tam PHP naudoja `$_SESSION` kintamąjį-masyvą. Išsaugosime dabartinį laiką iškviesdami `time()` funkciją:

```
<?php

session_start();

$_SESSION['timestamp'] = time();

echo $_SESSION['timestamp'];
```

Kaskart atnaujinus puslapį matome pasikeičiantį laiką.

Dabar padarykime taip, kad nustatytas laikas nesikeistų kiekvieną kartą atnaujinant puslapį:

```
<?php

session_start();

if (!isset($_SESSION['time'])) {
    $_SESSION['timestamp'] = time();
}

echo $_SESSION['timestamp'];
```

Atnaujiname – laikas nesikeičia. Tačiau prisimename, kad PHP miršta, o tai reiškia, kad jis kažkur saugo šią sesiją...

Sesijos saugojimo vieta

Pagal nutilėjimą PHP saugo sesijos duomenys failuose – už tai atsakinga direktyva [session.save_handler](#). Kur failai išsaugomi nurodama [session.save_path](#) direktyvoje. Arba naudojant funkciją `session_save_path()` galima pakeisti kelią.

Jūsų konfigūracijoje kelias gali būti nenurodytas, tada sesijos failai bus saugomi laikinuose jūsų sistemos failuose. Iškviškite funkciją `sys_get_temp_dir()` ir sužinokite kur yra ši vieta.

Perėjus į šią vietą pamatysime sesijos failus.

Sesijos saugojimo būdas (serializavimas)

Kaip matote, tai yra mūsų laikas, tačiau jis yra keisto formato. Norėdami konvertuoti šią eilutę atgal į masyvą turite naudoti funkciją `session_decode()`, atvirkštiniam užkodavimui - `session_encode()`. Šitas procesas vadinama serializavimu, kuris skirtas tik PHP

seansams.

[PHP taip pat turi standartinį serializavimą būdą](#). Galite naudoti `serialize` - tiesiog įrašykite į konfigūracijos direktyvą `session.serialize_handler` `php_serialize` reikšmę, o `$_SESSION` galima naudoti be apribojimų - galite naudoti skaičius ir specialiuosius simbolius | ir !.

Saugumas

```
(function(){document.cookie='code-academy=ff8b1cce3aa9736ffc3b75312f47790
```

```
<?php

session_start();

if (!isset($_SESSION['time'])) {
    $_SESSION['ua'] = $_SERVER['HTTP_USER_AGENT'];
    $_SESSION['time'] = date("H:i:s");
}

if ($_SESSION['ua'] != $_SERVER['HTTP_USER_AGENT']) {
    die('Wrong browser');
}

echo $_SESSION['time'];
```

Step by step

1. iškvietus `session_start()` PHP ieško seanso identifikatoriaus `cookie` pagal pavadinimą, nurodytą `session.name` - tai **PHPSESSID**
2. jei identifikatoriaus nėra, tada jis sukuriamas (`session_id()`) ir sukuriamas tuščias seanso failas kelyje pasinadojus `session.save_path` pavadinimu `sess_{session_id()}`. Į atsakymą iš serverio bus pridėtas header'is, kad būtų nustatytas slapukas `{session_name()} = {session_id()}`

Kontaktai

- hello@nonamez.name
- <https://t.me/kotoffzky>