
Koláčky, sezení

Martin Klíma

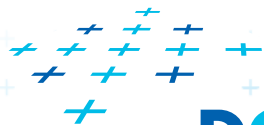


UDRŽOVÁNÍ STAVU APLIKACE



Udržování stavu aplikace

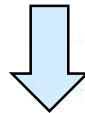
1. Pomocí skrytých polí (viz příklad wizard)
2. Pomocí obohacování odkazů
3. Pomocí cookies
4. Pomocí sessions



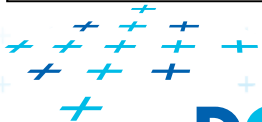
Obohacování odkazů

- Veškeré odkazy z dokumentu budou obohaceny o identifikátor s daty sezení
- Není to moc elegantní
- Může to selhat
- Pokud se děje automaticky, náročné na výkon

```
<a href="dalsi_stranka.php">Další stránka</a>  
<form action="obsluha.php" method="post">  
</form>
```

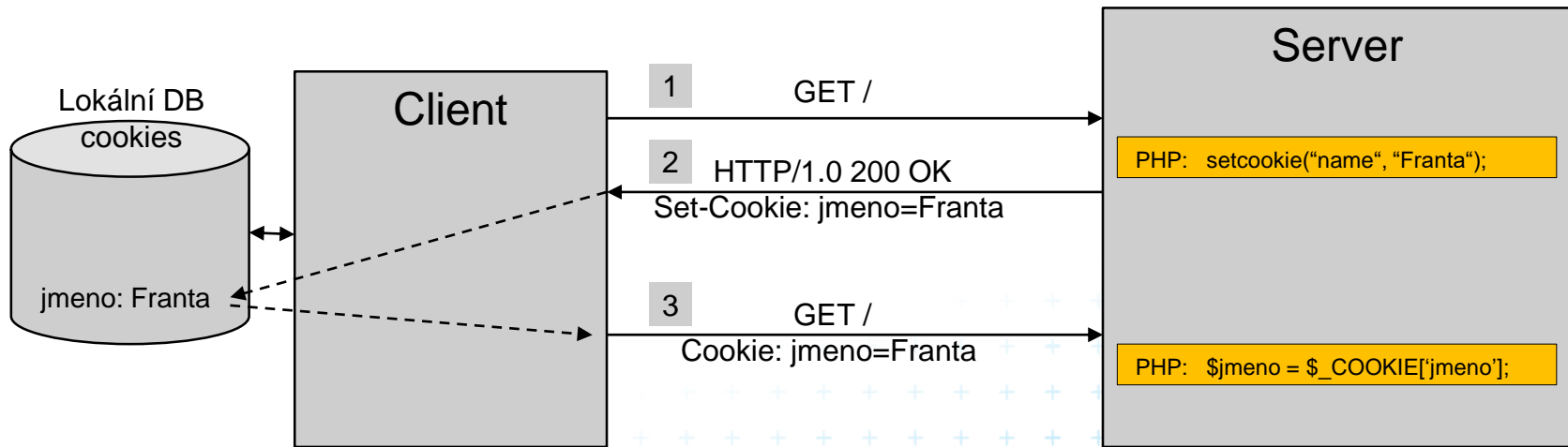


```
<a href="dalsi_stranka.php?session_id=XXX">Další stránka</a>  
<form action="obsluha.php?session_id=XXX" method="post">  
</form>
```



Cookie

- Informace uložena u klienta
- Posílá jí server
- Jen ten, kdo ji poslal, ji může opět přečíst



Pomocí cookies

- Informace uložená serverem na klientovi
- Klient posílá tuto informaci zpět pokud
 - URI spadá do vymezeného rozsahu
 - Doména odpovídá
 - Nevypršela platnost
 - Je zaručena odpovídající bezpečnost
- Využití:
 - Počítadlo přístupů
 - Další statistiky
 - **Udržování stavu aplikace**



Signatura funkce `setcookie`

`setcookie (`

`string $name`

`[, string $value = ""`

`[, int $expire = 0`

`[, string $path = ""`

`[, string $domain = ""`

`[, bool $secure = false`

`[, bool $httponly = false]`

`]]]])`

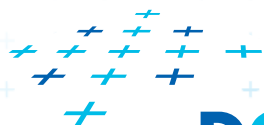
Unix timestamp, pokud 0, pak do konce života prohlížeče

Omezení jen na určenou cestu (adresář), např. /galerie/

Zobecnění pro celou poddoménu, např. cvut.cz

Pokud true, bude poslána jen přes https (bezpečný kanál)

Pokud true, nebude dostupná pro JavaScript, to je bezpečnostní opatření.



Cookies

```
<?php
```

```
setcookie("mojecookie1","hodnota1");  
setcookie("mojecookie2","hodnota2");
```

```
?>
```

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
<title>Nastav Cookie</title>
```

```
</head>
```

```
<body>
```

```
<p>Tento skript nastaví cookie a přečte existující cookie.</p>
```

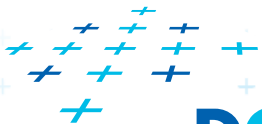
```
<?php
```

```
foreach ($_COOKIE AS $key=>$value) {  
    echo "$key = $value <br>";  
}
```

```
?>
```

```
</body>
```

```
</html>
```



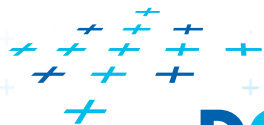
Výhody a omezení cookies

Výhody

- Obchází bezestavovost http protokolu
- Je všude implementováno

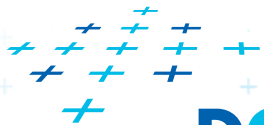
Nevýhody, omezení

- Zatěžuje komunikaci
 - Klient posílá při každém dotazu, i když to server nepotřebuje
- Omezení na délku (4KB)
 - Doporučuje se držet ji velmi malou
- Bezpečnostní riziko?
 - Klient posílá jen to, co dostal. Ale někdy to může být považováno za důvěrnou informaci
- Data nejsou šifrovaná



Vylepšení / nastavení cookies

SESSIONS

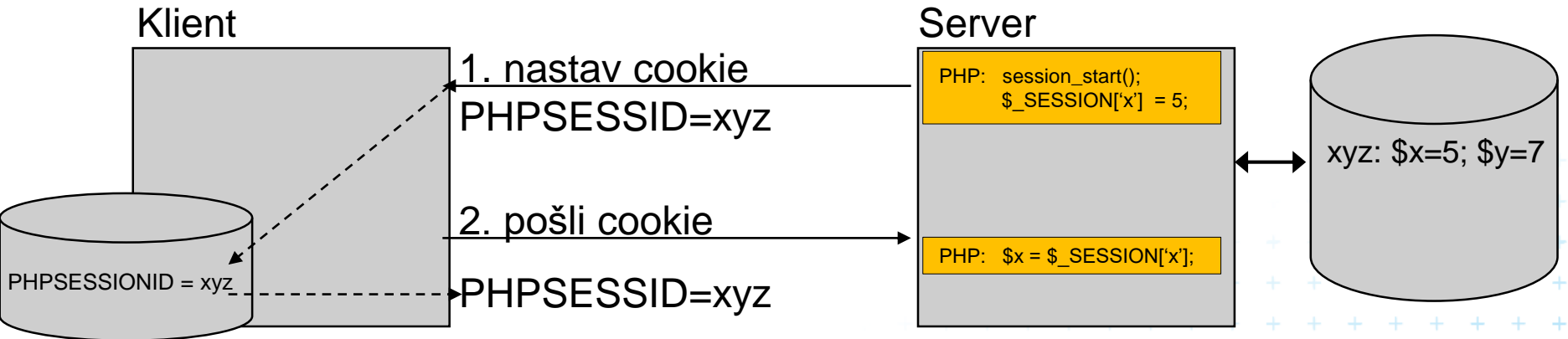


DCGI



Sessions

- Zjednodušení práce
- Podpora stavu
- Kombinace cookies a lokální databáze



Příklad SESSIONS

sessions.php

```
<?
session_start();

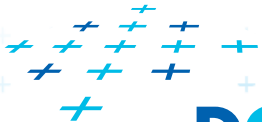
$_SESSION['x'] = 5;
$_SESSION['y'] = 7;

?>
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
<title>Sessions</title>
</head>
<body>
<p>Tento skript vypise seznam session promennych.</p>
```

```
<?
foreach ($_SESSION AS $key=>$value) {
    echo "$key = $value <br>";
}

?>
</body>
</html>
```

Pozor na XSS



DCGI



DB na serveru

- Ve výchozím stavu podle php.ini

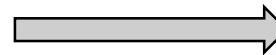
[Session]

```
session.save_handler=files  
session.save_path="C:\xampp\tmp"
```

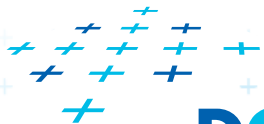
- Adresář C:\xampp\tmp

```
..  
sess_vkfa0df1ombhf63su9qkokfpk7
```

Serializované pole
\$_SESSION



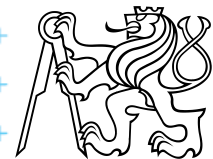
```
x|i:5;y|i:7;
```



Session lze ukládat i do DB

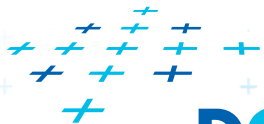
- Lze vnutit vlastní `session_handler`
- `session_set_save_handler(`
 `array($handler, 'open'),`
 `array($handler, 'close'),`
 `array($handler, 'read'),`
 `array($handler, 'write'),`
 `array($handler, 'destroy'),`
 `array($handler, 'gc')`
);

Kde `$handler` je instance třídy `SessionHandlerInterface`



Dotazy?

DĚKUJI ZA POZORNOST



DCGI

