

# Programování v PHP

Katedra softwarového inženýrství  
Fakulta informačních technologií  
České vysoké učení technické v Praze

© Pavel Štěpán, Helena Wallenfelsová, 2013

**Práce se stavem (session)**  
BI-PHP



Web (přesněji HTTP protokol) je BEZSTAVOVÝ. To znamená, že PHP program standardně „zapomíná“ (ruší) proměnné (globální i definované v objektech), jakmile se provádění programu skončí a vytvářená stránka je odeslána browseru. Např. tedy hodnoty proměnných mezi dvěma stisky tlačítka Submit se nepamatují!

Pro uchování proměnných (stavu) poskytuje PHP aparát tzv. Sessions (do češtiny někdy překládaných příšerným výrazem Sezení, jindy trochu lepším Relace). Session zhruba odpovídá jednomu běhu klasické aplikace. A na jednom počítači ta samá aplikace může (obvykle) být spuštěna vícekrát. Podobně jednu web aplikaci může najednou používat několik uživatelů (tj. několik browserů může procházet najednou stránky téže web aplikace).

Technicky se session realizuje tak, že každý návštěvník (browser) dostane přidělen jednoznačný identifikátor (SessionID) - o což požádá voláním funkce `session_start()` (viz příklad). Toto SessionID je posláno zpět browseru (který si jej uloží) spolu s požadovanou stránkou. A naopak, browser odesílá automaticky toto SessionID zpět serveru s každým požadavkem další stránky dané aplikace. Tím tedy server (respektive PHP program) ví, o kterou session (uživatele) jde.

SessionID se může posílat buď pomocí tzv. cookies - tj. v hlavičkách protokolu HTTP - a je ukládáno browserem jako textový soubor na klientském počítači. To je standardní postup. Nebo - např. pokud jsou cookies zakázány - může být SessionID připojováno na konec URL adres v dané stránce (tzv. prepisování URL). To lze nastavit pomocí příslušných položek v konfiguračním souboru `php.ini`.

SessionID se potom používá při ukládání hodnot proměnných na serveru. Programátorsky se to řeší poměrně snadnou syntaxí: chci-li na serveru uložit hodnotu, uložím ji do superglobálního pole `$_SESSION` [`"Moje_promenna"`] = `$i`; (`"Moje_promenna"` je jméno, pod kterým chce programátor ukládat danou hodnotu.) Opětovné načtení provede opět z `$_SESSION`.

PHP provede uložení např. tak, že do nějakého určeného adresáře ukládá textové soubory, obsahující název proměnné, ukládanou hodnotu a SessionID. Právě SessionID umožňuje při opětovném načítání nalézt ty hodnoty, které si právě daná session uložila (název budou mít pro danou proměnnou v příslušné aplikaci pochopitelně všechny session stejny).

Mechanismus ukládání ovšem může být jiný (opět se určuje v `php.ini`). Např. je možné k ukládání session proměnných použít tabulku v databázi.

### První stránka: First.php. Zde se zahájí session a uloží proměnná:

```
<?php session_start(); ?>
<!-- při použití cookies POVINNE na zacatku stranky - PRED odeslaním
jakýchkoli tagu, textu a dalších casti stranky (hlavicky v HTTP
protokolu MUSEJI byt odesilany před obsahem stranky)
-->
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Zahajeni session, ulozeni hodnoty do session</title>
  </head>
  <body>

    <form action="First.php" method="get">
      Jmeno:<br>
      <!-- uživatelem zadana ukladana hodnota -->
      <input type="text" name="txtJmeno"><br><br>
```

```

        <!-- stisk tlačítka uloží do session -->
        <input type="submit" name="btnSubmit" value="Ulož jméno">

        <!-- hyperlink pro přechod na další stránku -->
        <a href="Second.php">Jdi na druhou stránku</a>
    </form>

    <?php
        if(!empty($_GET["btnSubmit"])){                                // odeslano tlačítkem?
            $_SESSION["Name"] = $_GET["txtJmeno"]; // uložení do session
        }
        // var_dump($_SESSION);    // případný výpis informací o proměnné
    ?>
</body>
</html>

```

**Poznámka:** v praxi by se pochopitelně neoddělovalo uložení do session a přechod na další stránku (uživatel by určitě jedno nebo druhé zapomněl udělat). Obojí by se provedlo např. uvedením adresy druhé stránky v atributu action tagu form.

**Druhá stránka: Second.php.** Zde se vůbec session proměnné nepoužívají - ale přesto se k nim budeme moci na třetí stránce vrátit:

```

<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Stránka, nepoužívající session proměnné.</title>
    </head>
    <body>
        <!-- pro přechod na další nebo předchozí stránku -->
        <a href="Third.php">Jdi na třetí stránku</a><br>
        <a href="First.php">Jdi na první stránku</a><br>
    </body>
</html>

```

**Třetí stránka: Third.php.** Zde použijeme proměnnou, uloženou do session na 1. stránce.

**Poznámka:** trochu matoucí session\_start nejenom zahajuje session, ale také je nutné k pokračování session na všech stránkách, kde se se session proměnnými pracuje!

```

<?php session_start(); ?>
<!-- pro práci se session proměnnými opět NUTNO použít session_start -->
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Použití drive uložené proměnné.</title>
    </head>
    <body>
        <a href="Second.php">Jdi na druhou stránku</a><br>
    </body>
</html>

```

```
<?php
    if(isset($_SESSION["Name"])){
        // nacteni ulozene hodnoty (session promenne)
        echo "Zdravime ".$_SESSION["Name"]."<br>"; // nacteni ze session
        // nacteni SessionID (obvykle není třeba znát)
        echo "Session ID je ".$_SESSION["id"]."<br>";
    }
?>
</body>
</html>
```

**Závěrečná poznámka:** Session obvykle uživatel explicitě nekončí (i když je možné dát na stránku nějaké tlačítko **Konec**, po jeho stisku zrušit session promenné (funkcí `unset`) a zobrazit uživateli stránku s poděkováním za použití aplikace. Takové tlačítko ale obvykle uživatel ignoruje a prostě přejde na nějaké jiné stránky někde na webu.

Proto se web aplikace obvykle končí **timeoutem**, který se nastavuje v `php.ini` jako délka života session (max. doba, po kterou uživatel nepřistoupil k nějaké stránce aplikace). Je dost obtížné tuto hodnotu správně nastavit: je-li příliš krátká, uživatel nestačí před koncem session vyplnit a odeslat složité formuláře - jeho session zatím skončí a minulá data se ztratí (bude vás mít hodně rád). Naopak při příliš dlouhé session budete na serveru zbytečně udržovat data uživatelů, kteří se třeba na váš web jenom podívali a hned šli někam jinam. V praxi je třeba vyzkoušet, jak dlouho (pomalejším) uživatelům trvá práce s vaší aplikací.