

APLIKACJE INTERNETOWE PHP



TEMAT 5-12: Sesje.

Autor dokumentu: Wojciech Galiński

Środa, 23 listopada 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY: <http://www.php.net/manual/pl>, <http://webmaster.helion.pl/index.php/kurs-php>, <http://pl.wikipedia.org/>,
<http://pl.wikibooks.org/wiki/PHP>, <http://phpkurs.pl/>, <http://kursphp.com/>.

Zagadnienia obowiązkowe

1. **Ważna cecha protokołu HTTP** – „jest protokołem bezstanowym. Oznacza to, że serwer WWW rozpatruje każde żądanie niezależnie od innych, nie szukając żadnych powiązań w stylu wysyłania ich przez tego samego internautę.”
2. **Sesje w PHP** – to powiązanie żądań HTTP z użytkownikiem za pomocą identyfikatora „**session_id**”. Podczas trwania sesji, do kolejnych żądań dołączany jest identyfikator sesji (przechowywany jest w cookie – ciasteczku sesyjnym). Każda sesja ma inny identyfikator, co umożliwia rozróżnienie użytkowników.

UWAGA: aby sesje były dostępne, musi być włączona obsługa ciasteczek (cookie).

3. **Kończenie sesji** – to zadanie realizuje funkcja „**session_destroy**”, lecz nie usuwa ani tablicy „**\$_SESSION**” ani ciasteczka sesyjnego (cookie).
4. **Prosty przykład sesji w języku PHP** – oto on:

```
<?php
    session_start(); // wywołanie tej funkcji MUSI być umieszczone na samym
                    // początku dokumentu - przed wysłaniem nagłówków HTTP, a
                    // więc najlepiej przed jakimkolwiek znacznikiem języka HTML
    if (isset($_GET['koniec_sesji']))
    { $_SESSION = array();    session_destroy();
      if (isset($_COOKIE[session_name()]))
        setcookie(session_name(), '', time()-42000, '/'); }
?> <!DOCTYPE html><html><head><meta charset="utf-8" /></head><body> <?php
// Dodanie/Inkrementacja zmiennej o zasięgu całej sesji
if (isset($_SESSION['licznik'])) ++$_SESSION['licznik'];
else $_SESSION['licznik'] = 1;
var_dump($_SESSION); // wyświetlenie wszystkich zmiennych o zasięgu sesji
echo '<br />ID sesji: '. session_id(). '<br />';
?> <a href="<?php echo $_SERVER['PHP_SELF']; ?>">Następny krok</a>
<a href="<?php echo $_SERVER['PHP_SELF'].'?koniec_sesji'; ?>">Koniec sesji</a>

</body></html>
```

5. **Zabezpieczanie sesji przed włamaniami** – sesja w PHP są zagrożone 2 podstawowymi metodami włamań: „Session Fixation” (sprawdzamy, czy ktoś nie przejął identyfikatora sesji) oraz „Session Hijacking” (badanie, czy żądania sesji przychodzą z komputera o takim samym IP). Aby zabezpieczyć się przed nimi wystarczy każdy skrypt rozpoczynać od poniższego kodu (oczywiście kod ten można rozbudować):

```
session_start();
if (!isset($_SESSION['init']) || !isset($_SESSION['ip'])
    || $_SESSION['ip'] != $_SERVER['REMOTE_ADDR'])
{ // zabezpieczenie przed włamaniem typu „Session Fixation”
    session_regenerate_id();
    // zabezpieczenie przed włamaniem typu „Session Hijacking”
    $_SESSION['ip'] = $_SERVER['REMOTE_ADDR'];    $_SESSION['init'] = true;
}
```

6. Prosta autoryzacja użytkowników – w tym celu można wykorzystać:

- ➔ formularz logowania;
- ➔ mechanizm „PostBack” (wysyłanie danych formularza do dokumentu wysyłającego);
- ➔ mechanizm sesji języka PHP.

PRZYKŁAD:

```
<?php session_start();
if (!isset($_SESSION['inicjalizacja']) || !isset($_SESSION['ip'])
    || $_SESSION['ip'] != $_SERVER['REMOTE_ADDR'])
{ // zabezpieczenie przed włamaniem typu „Session Fixation”
    session_regenerate_id();    $_SESSION['inicjalizacja'] = true;
    // zabezpieczenie przed włamaniem typu „Session Hijacking”
    $_SESSION['ip'] = $_SERVER['REMOTE_ADDR'];
}
logowanie();
function logowanie()
{ $u = 'u'; $h = crypt('h');
  if (isset($_GET['wyloguj'])) wylogowanie();
  elseif (isset($_POST['u']) && isset($_POST['h']) && $_POST['u']==$u)
  { if (crypt($_POST['h'], $h)==$h)
    { $_SESSION['u'] = $_POST['u'];    $_SESSION['potega'] = 1; }
  }
}
function wylogowanie()
{ $_SESSION = array();    session_destroy();
  if (isset($_COOKIE[session_name()]))
    setcookie(session_name(), '', time()-42000, '/');
}
function tresc_przed_zalogowaniem()
{
  ?>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
<fieldset><legend>Logowanie</legend>
  <p>Uzytkownik:<br /><input type="text" name="u" /></p>
  <p>Hasło:<br /><input type="password" name="h" /></p>
  <p style="text-align:center;"><input type="submit" value="Zaloguj" /></p>
</fieldset> </form> <?php
}
function tresc_po_zalogowaniu()
{
  ?>
<p><a href="<?php echo $_SERVER['PHP_SELF']; ?>">Następny krok</a>
<a href="<?php echo $_SERVER['PHP_SELF'].'?wyloguj'; ?>">Wyloguj</a></p>
<p>Witaj, <b><?php echo $_SESSION['u'].</b>. Potega: '.
($_SESSION['potega']*2).'. ';></p> <?php
}
?> <!DOCTYPE html><html><head><meta charset="utf-8" /></head><body>
<?php
  if (isset($_SESSION['u'])) tresc_po_zalogowaniu();
  else tresc_przed_zalogowaniem();
?> </body></html>
```

Zadania

1. Zapoznaj się z przedstawionymi przykładami. Wykonaj je i przeanalizuj ich działanie.
2. Zmodyfikuj powyższy skrypt, aby obsługiwał logowanie dowolnej liczby użytkowników (dane logowania możemy trzymać np. w tablicy asocjacyjnej w postaci: 'użytkownik'=>'hasło').
3. Utwórz skrypt, który – po zalogowaniu – wyświetla imię, nazwisko, nazwę klasy i zdjęcie ucznia oraz informuje ucznia o jego ocenach końcowych z dowolnej liczby przedmiotów.