



WEB разработка

**PHP
СЕССИИ**

Съдържание

- PHP Sessions
- Стартиране на сесия
- Working with Session Variables
- Destroying Sessions and Unsetting Session Variables
- Working with Registered Users

PHP Sessions

PHP Sessions

Сесиите разполагат с уникален идентификатор - session ID и стартирани от потребителя, могат да бъдат използвани за съхраняване и използване на информация, свързани с този ID.

Всички променливи, асоциирани с текущата сесия са достъпни в кода чрез суперглобалната PHP променлива

`$_SESSION`

PHP Sessions

Асоциираните данни с текущата сесия /session state/ обикновено се запазват във временен файл. Може да се използва и база данни или друг метод, чрез функцията

```
session_set_save_handler()
```

Стартиране на сесия

Стартиране на сесия

- За да работите със сесия трябва изрично да я стартирате или да промените настройките в `php.ini`.
- По подразбиране сесиите не започват автоматично.
- За да започват автоматично, трябва да промените -

от `session.auto_start = 0`

на `session.auto_start = 1`

Стартиране на сесия

При `session.auto_start = 1`

Сесията стартира за всеки документ зареден от уеб сървъра

При `session.auto_start = 0`

в началото на всеки скрипт трябва да поставите

`session_start()`

за да стартира сесията за текущия документ.

Стартиране на сесия

След стартиране на сесията може да достъпите нейното ID чрез

```
session_id()
```

Пример Стартиране или възстановяване на сесия.

```
1: <?php
2: session_start();
3: echo "<p>Your session ID is ".session_id()."</p>";
4:
```

Стартиране на сесия

По подразбиране сесията продължава докато се затвори браузъра.

Продължителността на сесията може да се промени в `php.ini`

Working with Session Variables

Working with Session Variables

- След стартиране на сесията, може да запазите произволен брой променливи в \$_SESSION и да ги достъпвате в страници, в които също е стартирана сесия

Пример Добавяне на две променливи към \$_SESSION \$product1 и \$product2

```
1: <?php
2: session_start();
3: $_SESSION['product1'] = "Sonic Screwdriver";
4: $_SESSION['product2'] = "HAL 2000";
5: echo "The products have been registered.";
```

Working with Session Variables

Магията от предишния код ще даде резултат, когато потребителят се премести в нова страница /изпрати заявка и сървъра зареди нов файл/.

Working with Session Variables

Пример. Достъпване на запазените в сесията променливи

```
1: <?php
2: session_start();
3: ?>
4: <!DOCTYPE html>
5: <html lang="en">
6: <head>
7: <title>Your Products</title>
8: </head>
9: <body>
10: <h1>Your Products</h1>
11: <p>Your chosen products are:</p>
12: <ul>
13: <li><?php echo $_SESSION['product1']; ?></li>
14: <li><?php echo $_SESSION['product2']; ?></li>
15: </ul>
16: </body>
17: </html>
```

Working with Session Variables

Какво се случва в действителност?

Зад кадър PHP записва информацията във временен файл на сървъра.

Може да намерите, къде се съхранява този файл с функцията

```
session_save_path()
```

Задача. Открийте файла, в който са записани данните от сесията.

Working with Session Variables

След като добавите променлива към `$_SESSION`, може да променяте нейната стойност в последствие в кода и да използвате променената стойност.

Ако искате да запазите променената стойност в сесията, трябва да презапишете данните в `$_SESSION`.

```
<?php
$a = 10;
$_SESSION['a'] = $a;
echo $_SESSION['a']; //10
$a = 20;
echo $_SESSION['a']; //10
$_SESSION['a'] = $a;
echo $_SESSION['a']; //20
```


Working with Session Variables

Пример Добавяне на масив към \$_SESSION, с функцията serialize()

```
1: <?php
2: session_start();
3: ?>
4: <!DOCTYPE html>
5: <html lang="en">
6: <head>
7: <title>Storing an array with a session</title>
8: </head>
9: <body>
10: <h1>Product Choice Page</h1>
11: <?php
12: if (isset($_POST['form_products'])) {
13:     if (!empty($_SESSION['products'])) {
14:         $products = array_unique(
15:             array_merge(unserialize($_SESSION['products']),
16:                 $_POST['form_products']));
17:         $_SESSION['products'] = serialize($products);
18:     } else {
19:         $_SESSION['products'] = serialize($_POST['form_products']);
20:     }
21:     echo "<p>Your products have been registered!</p>";
22: }
23: ?>
```

Working with Session Variables

Пример Добавяне на масив към \$_SESSION, с функцията serialize()

```
24:         <form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
25:             <p><label for="form_products">Select some products:</label><br>
26:             <select id="form_products" name="form_products[]" multiple="multiple" size="3">
27:                 <option value="Sonic Screwdriver">Sonic Screwdriver</option>
28:                 <option value="Hal 2000">Hal 2000</option>
29:                 <option value="Tardis">Tardis</option>
30:                 <option value="ORAC">ORAC</option>
31:                 <option value="Transporter bracelet">Transporter bracelet</option>
32:             </select></p>
33:             <button type="submit" name="submit" value="choose">Submit Form</button>
34:         </form>
35:         <p><a href="session1.php">go to content page</a></p>
36: </body>
37: </html>
```

Working with Session Variables

Достъпване на сериализирани променливи в сесията

```
1: <?php
2: session_start();
3: ?>
4: <!DOCTYPE html>
5: <html lang="en">
6:     <head>
7:         <title>Accessing Session Variables</title>
8:     </head>
9:     <body>
10:         <h1>Content Page</h1>
11:         <?php
12:         if (isset($_SESSION['products'])) {
13:             echo "<strong>Your cart:</strong><ol>";
14:             foreach (unserialize($_SESSION['products']) as $p) {
15:                 echo "<li>".$p."</li>";
16:             }
17:             echo "</ol>";
18:         }
19:         ?>
20:         <p><a href="arraysession.php">return to product choice page</a></p>
21:     </body>
22: </html>
```

Destroying Sessions and Unsetting Session Variables

Destroying Sessions and Unsetting Session Variables

За да прекратите сесия може да използвате функцията

`session_destroy()`

Тя ще изтрие всички променливи, запазени на сървъра.

В скрипта се използва в следната последователност

```
session_start();  
    //стартираме сесия  
session_destroy();  
    //унищожаваме сесията
```

След унищожаване на сесията, преминаването към страница, стартираща сесия - ще предизвика стартиране на нова сесия. Всички данни от предходната сесия вече не са достъпни, започва запаметяването на нови данни - тези на новостартираната сесия.

Destroying Sessions and Unsetting Session Variables

Извикването на `session_destroy()` не унищожава веднага всички, регистрирани в сесията променливи. Те все още са достъпни в скрипт /файла/, в който е извикана `session_destroy()`, докато не презаредите страницата!

Пример

```
session_start();  
$_SESSION['test'] = 5;  
session_destroy();  
echo $_SESSION['test']; // 5
```

Destroying Sessions and Unsetting Session Variables

За да унищожите в даден момент, регистрираните в сесията променливи трябва да използвате функцията

```
unset()
```

```
session_start();  
$_SESSION['test'] = 5;  
session_destroy();  
unset($_SESSION['test']);  
echo $_SESSION['test']; // prints nothing (or a notice about an undefined index)
```


Working with Registered Users

Working with Registered Users

Да предположим, че сте изградили онлайн платформа, блог, портал или друго приложение, в което потребителите могат да се присъединяват като сергистрират, например с потребителско име и парола. След регистриране, всяко при всяко влизане на потребителя може да запазите потребителското му име /и друга информация/ в сесията. Данните, които решите да запазите в сесията на потребителя, трябва да бъдат, тези, които смятате, че ще използвате доста често и било неефективно да се вадят от базата данни. Например - ако сте разработили портал, в който всеки потребител има определени права за достъп - администратор, регистриран потребител, анонимен гост и т. н. В отделните модули на приложението, ще проверявате правата на потребителя, за да сте сигурни, че той има право да достъпва съответния модул. Или правата на достъп на потребителя, са пример за стойност, която бихте запазили в сесията на съответния потребител, така че скрипта проверяващ дали може да визуализира даден модул по заявка на потребителя да проверява само стойността в сесията, без да е необходимо да се свързва, селектира и търси в базата данни.

Summary

Summary

- С помощта на сесиите, имаме възможност да съхраним състояние, при използване на HTTP заявки /които са stateless - не могат да запамятят състояние - стойности на променливи/.
- Един от вариантите за запис на данните от сесията е да се записват във временен файл, който се съхранява на сървъра.
- Стартиране/Възстановяване/Продължаване сесия с функцията `session_start()`
- При стартиране на сесия, запазваме стойности на променливи в масива `$_SESSION`.
- Данните записани в `$_SESSION` се унищожават с `unset()` или при унищожаването на самата сесия със `session_destroy()`. Във втория случай променливите са достъпни до края на скрипта или докато се презареди страницата.