Pliki Cookies w PHP

Pliki cookie to małe fragmenty tekstu wysyłane do przeglądarki przez odwiedzaną witrynę. Pomagają one witrynie zapamiętać informacje o wizycie użytkownika, co może zarówno ułatwić ponowne odwiedzenie witryny, jak i uczynić ją bardziej użyteczną dla użytkownika.

Zalety plików cookie

1. Przyjazność dla użytkownika

Pliki cookie są niezwykle przyjazne dla użytkownika. Klient może wybrać, co chce zrobić z plikami cookie. Wszystkie przeglądarki mają ustawienia umożliwiające wyczyszczenie historii, w tym plików cookie.

2. Dostępność

Pliki cookie można również ustawić tak, aby były dostępne przez dłuższy czas. Gdy pliki cookie zostaną zapisane na dysku twardym użytkownika, będą dostępne tak długo, jak długo użytkownik nie usunie ich ręcznie.

3. Wygoda

Oprócz stron internetowych, pliki cookie mogą również zapamiętywać informacje związane z formularzami. Tak więc za każdym razem, gdy użytkownik odwiedza witrynę, formularz adresu zostanie wypełniony automatycznie. Pliki cookie nie zapamiętują jednak informacji poufnych, takich jak dane karty kredytowej.

4. Marketing

Większość firm, w szczególności witryny e-commerce, wykorzystuje pliki cookie do kierowania produktów do swoich klientów. Informacje takie jak wyszukiwane hasła, słowa kluczowe i lokalizacje geograficzne są gromadzone na potrzeby kampanii marketingowych.

5. Konfiguracje

Pliki cookie można również skonfigurować zgodnie z wymaganiami. Na przykład, mogą one wygasać po zamknięciu przez użytkownika karty przeglądarki lub istnieć tylko przez określony czas.

6. Wymagania serwera

Wszystkie dane związane z plikami cookie są przechowywane na dysku twardym bez użycia zasobów serwera. Żadne dodatkowe obciążenie nie jest dodawane do serwera. W związku z tym nakładane jest na nie mniejsze obciążenie, co sprawia, że pliki cookie są łatwiejsze do wdrożenia.

Wady plików cookie

1. Wpływ na przeglądarkę

Pliki cookie nie są ograniczone na podstawie korzystania z Internetu. Za każdym razem, gdy użytkownik surfuje po sieci, gromadzi się coraz więcej plików cookie. Dopóki użytkownik ich nie usunie, pliki te będą stanowić część miejsca na dysku twardym. To ostatecznie spowalnia lub opóźnia działanie przeglądarki.

2. Zagrożenia dla bezpieczeństwa

Ponieważ pliki cookie są przechowywane na dysku twardym jako pliki tekstowe, stanowią one poważne zagrożenie dla bezpieczeństwa. Każdy intruz może łatwo otworzyć te pliki i wyświetlić informacje. Ponadto nie wszystkie witryny, które zbierają informacje z plików cookie, są legalne. Niektóre z nich mogą być złośliwe i wykorzystywać pliki cookie do celów hakerskich.

3. Ograniczenia rozmiaru

Pliki cookie mają również ograniczenia rozmiaru. Nie mogą one przechowywać dużej ilości informacji. Większość plików cookie może przechowywać informacje tylko do 4kb. Przeglądarki również nakładają ograniczenia, jeśli chodzi o liczbę plików cookie.

4. Obawy dotyczące prywatności

Oprócz bezpieczeństwa, prywatność jest kolejną obawą użytkowników związaną z plikami cookie. Za każdym razem, gdy użytkownik przegląda Internet, witryny obsługujące pliki cookie będą rejestrować wszystkie działania online.

5. Ręczne wyłączanie

Przeglądarki mają również opcję wyłączenia plików cookie. Użytkownicy, którzy są bardzo świadomi bezpieczeństwa, mogą je po prostu wyłączyć.

6. Kodowanie informacji

Zarówno szyfrowanie, jak i odszyfrowywanie plików cookie jest trudnym procesem, ponieważ wymaga dodatkowego kodowania.

Tworzenie cookie

setcookie(name, value, expire, path, domain, secure, httponly);

Przykład kodu:

```php
<?php
$cookie_nazwa = "uzytkownik";
$cookie_wartosc = "Jan Kowalski";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 sekund = 1 dzien
// funkcja setcookie()musi pojawić się PRZED znacznikiem <html>.
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_nazwa])) {
  echo "Cookie nazwany '" . $cookie_nazwa . "' nie jest ustawiony!";
} else {
  echo "Cookie '" . $cookie_nazwa . "' jest ustawiony!<br>";
  echo "Wartosc to: " . $_COOKIE[$cookie_nazwa];
}
?>

</body>
</html>
```

Usuwanie pliku Cookie

Aby usunąć plik cookie, wystarczy użyć funkcji setcookie() z datą wygaśnięcia w przeszłości:

Przykład kodu:

```php
<?php
//ustaw datę wygaśnięcia na jedną godzinę temu
setcookie("uzytkownik", "", time() - 3600);
?>
<html>
<body>

<?php
echo "Cookie 'uzytkownik' zostalo usuniete.";
?>

</body>
</html>
```

Sprawdzanie czy Cookie są włączone

Poniższy przykład to skrypt, który sprawdza, czy pliki cookie są włączone. Najpierw spróbuj utworzyć testowe ciasteczko z funkcją setcookie(), a następnie policz zmienną tablicową $_COOKIE:

Przykład kodu:

```php
<?php
setcookie("test_cookie", "test", time() + 3600, '/');
?>
<html>
<body>

<?php
if(count($_COOKIE) > 0) {
  echo "Cookies sa wlaczone.";
} else {
  echo "Cookies sa wylaczone.";
}
?>

</body>
</html>
```

Co to jest sesja w PHP?

Podczas pracy z aplikacją otwierasz ją, wprowadzasz zmiany i zamykasz.
Jest bardzo podobny do sesji. Komputer wie, kim jesteś. Wie, kiedy uruchamiasz aplikację i kiedy ją zamykasz. Ale jest jeden problem z Internetem: serwer WWW nie wie, kim jesteś ani co robisz, ponieważ adres HTTP jest nieprawidłowy. Zmienne sesyjne rozwiązują ten problem, przechowując informacje o użytkowniku do wykorzystania na wielu stronach. Domyślnie zmienne sesyjne utrzymują się, dopóki użytkownik nie zamknie przeglądarki. Więc; Zmienne sesyjne przechowują informacje o pojedynczym użytkowniku i są dostępne dla wszystkich stron w ramach jednej aplikacji.

Zalety :

1. Pomaga utrzymywać stany użytkownika i dane w całej aplikacji.

2. Może być łatwo zaimplementowany i możemy przechowywać dowolny rodzaj obiektu.
3. przechowuje dane każdego klienta oddzielnie.
4. Sesja jest bezpieczna i przejrzysta dla użytkownika.

Wady:

1. Narzut wydajności w przypadku dużej liczby użytkowników, ze względu na dane sesji przechowywane w pamięci serwera.
2. Narzut związany z serializacją i deserializacją danych sesji, ponieważ w przypadku trybu sesji StateServer i SQLServer musimy serializować obiekt przed zapisaniem.

Przykład rozpoczęcia sesji w PHP

Sesja jest uruchamiana z **session_start()** funkcją.

```php
<?php
// Zaczynamy sesje
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// ustawiamy wartosci dla sesji
$_SESSION["wzrost"] = "176";
$_SESSION["ulubiony_napoj"] = "lipton";
echo "SWartosci sesji sa ustawione.";
?>

</body>
</html>
```

Uzyskaj wartości zmiennych sesji PHP

Zauważ, że zmienne sesyjne nie są przekazywane indywidualnie do każdej nowej strony, zamiast tego są pobierane z sesji, którą otwieramy na początku każdej strony **session_start()** również, że wszystkie wartości zmiennych sesyjnych są przechowywane w globalnej zmiennej **$_SESSION**:

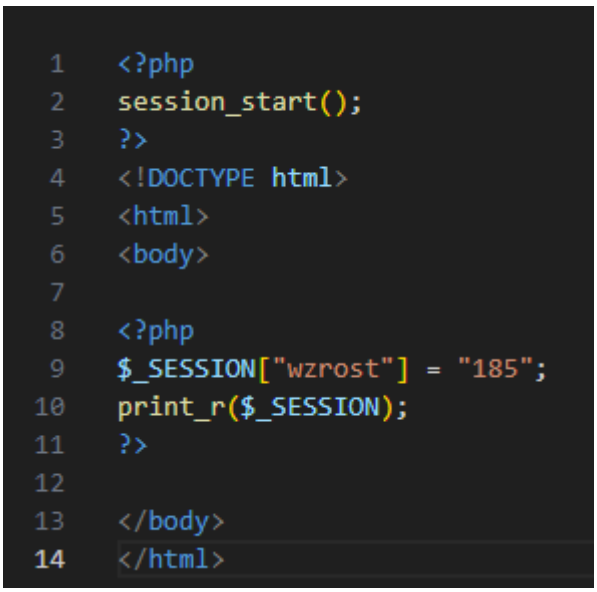
Przykład kodu:

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
echo "Wzrost wynosi " . $_SESSION["wzrost"] . ".<br>";
echo "Ulubionym napojem jest " . $_SESSION["ulubiony_napoj"] . ".";
?>

</body>
</html>
```

Modyfikacja zmiennej w sesji

Aby zmienić zmienną w sesji wystarczy, że zostanie ona nadpisana.

Przykład kodu:



```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
$_SESSION["wzrost"] = "185";
print_r($_SESSION);
?>

</body>
</html>
```

Usunięcie sesji

Żeby usunąć sesje musimy, użyć:

session_unset() i session_destroy()

Przykład kodu:

```php
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Usuwamy zmienne
session_unset();

// usuwamy sesje
session_destroy();
?>

</body>
</html>
```