

Sessions PHP

Dans ce chapitre, vous apprendrez à stocker temporairement certaines données sur le serveur à l'aide d'une session PHP.

Qu'est-ce qu'une session

Bien que vous puissiez stocker des données à l'aide de cookies, cela présente des problèmes de sécurité. Étant donné que les cookies sont stockés sur l'ordinateur de l'utilisateur, il est possible pour un attaquant de modifier facilement le contenu d'un cookie pour insérer des données potentiellement nuisibles dans votre application qui pourraient casser votre application.

De plus, chaque fois que le navigateur demande une URL au serveur, toutes les données de cookie d'un site Web sont automatiquement envoyées au serveur dans la demande. Cela signifie que si vous avez stocké 5 cookies sur le système de l'utilisateur, chacun ayant une taille de 4 Ko, le navigateur doit télécharger 20 Ko de données chaque fois que l'utilisateur consulte une page, ce qui peut affecter les performances de votre site.

Vous pouvez résoudre ces deux problèmes en utilisant la session PHP. Une session PHP stocke les données sur le serveur plutôt que sur l'ordinateur de l'utilisateur. Dans un environnement basé sur une session, chaque utilisateur est identifié par un numéro unique appelé identifiant de session ou SID. Cet identifiant de session unique est utilisé pour relier chaque utilisateur à ses propres informations sur le serveur, telles que des e-mails, des publications, etc.

Astuce : Les identifiants de session sont générés de manière aléatoire par le moteur PHP, ce qui est presque impossible à deviner. De plus, comme les données de session sont stockées sur le serveur, il n'est pas nécessaire de les envoyer à chaque requête du navigateur.

Démarrer une session PHP

Avant de pouvoir stocker des informations dans des variables de session, vous devez d'abord démarrer la session. Pour commencer une nouvelle session, appelez simplement la fonction PHP **session_start()**. Il créera une nouvelle session et générera un ID de session unique pour l'utilisateur.

Le code PHP dans l'exemple ci-dessous démarre simplement une nouvelle session.

```
<?php
// Démarrer une nouvelle session
session_start();
?>
```

La fonction **session_start()** vérifie d'abord si une session existe déjà en recherchant la présence d'un ID de session. S'il en trouve une, c'est-à-dire si la session est déjà démarrée, il configure les variables de session et si ce n'est pas le cas, il démarre une nouvelle session en créant un nouvel identifiant de session.

Remarque : Vous devez appeler la fonction `session_start()` au début de la page, c'est-à-dire avant toute sortie générée par votre script dans le navigateur, un peu comme vous le faites lors de la configuration des cookies avec la fonction `setcookie()`.

Stockage et accès aux données de session

Vous pouvez stocker toutes vos données de session sous forme de paires clé-valeur dans le tableau superglobal **\$_SESSION[]**. Les données stockées sont accessibles pendant la durée de vie d'une session. Considérez le script suivant, qui crée une nouvelle session et enregistre deux variables de session.

```
<?php
// Démarrer une session
session_start();

// Stockage de données
$_SESSION["firstname"] = "Peter";
$_SESSION["lastname"] = "Parker";
?>
```

Pour accéder aux données de session que nous avons définies dans notre exemple précédent à partir de n'importe quelle autre page Web du même domaine, recréez simplement la session en appelant **session_start()**, puis transmettez la clé correspondante au tableau associatif **\$_SESSION**.

```
<?php
// Démarrer session
session_start();

// Accéder aux données de sessions
echo 'Salut, ' . $_SESSION["firstname"] . ' ' . $_SESSION["lastname"];
?>
```

Remarque : Pour accéder aux données de la session dans la même page, il n'est pas nécessaire de recréer la session puisqu'elle a déjà été lancée en haut de la page.

Détruire une session

Si vous souhaitez supprimer certaines données de session, désactivez simplement la clé correspondante du tableau associatif `$_SESSION`, comme illustré dans l'exemple suivant :

```
<?php
// Démarrer une session
session_start();

// Supprimer les données de sessions
if(isset($_SESSION["lastname"])){
    unset($_SESSION["lastname"]);
}
?>
```

Cependant, pour détruire complètement une session, appelez simplement la fonction **session_destroy()**. Cette fonction n'a pas besoin d'argument et un seul appel détruit toutes les données de la session.

```
<?php
// Démarrer session
session_start();

// Détruire session
session_destroy();
?>
```

Remarque : Avant de détruire une session avec la fonction **session_destroy()**, vous devez d'abord recréer l'environnement de la session s'il n'y est pas déjà en utilisant la fonction **session_start()**, afin qu'il y ait quelque chose à détruire.

Chaque session PHP a une valeur de délai d'attente - une durée, mesurée en secondes - qui détermine combien de temps une session doit rester active en l'absence de toute activité de l'utilisateur. Vous pouvez ajuster cette durée de temporisation en modifiant la valeur de la variable **session.gc_maxlifetime** dans le fichier de configuration PHP (**php.ini**).