**Sessions et cookies en PHP**

**Table des matièresPlier**

* [III. Manipuler les sessions](https://php.developpez.com/cours/sessions/?page=page_3#LIII)
  + [III-A. Mise en garde](https://php.developpez.com/cours/sessions/?page=page_3#LIII-A)
    - [Les registers\_globals](https://php.developpez.com/cours/sessions/?page=page_3#Lno-III-A-1)
    - [Les fonctions session\_register(), session\_unregister(), session\_is\_registered() et session\_unset()](https://php.developpez.com/cours/sessions/?page=page_3#LIII-A-1)
  + [III-B. Débuter une session](https://php.developpez.com/cours/sessions/?page=page_3#LIII-B)
  + [III-C. Sauvegarder une variable](https://php.developpez.com/cours/sessions/?page=page_3#LIII-C)
  + [III-D. Récupération de données dans une session](https://php.developpez.com/cours/sessions/?page=page_3#LIII-D)
  + [III-E. Savoir si une variable appartient à une session](https://php.developpez.com/cours/sessions/?page=page_3#LIII-E)
  + [III-F. Supprimer une variable d'une session](https://php.developpez.com/cours/sessions/?page=page_3#LIII-F)
  + [III-G. Détruire une session](https://php.developpez.com/cours/sessions/?page=page_3#LIII-G)
  + [III-H. Détruire toutes les variables d'une session](https://php.developpez.com/cours/sessions/?page=page_3#LIII-H)
  + [III-I. Régénérer les identifiants de session](https://php.developpez.com/cours/sessions/?page=page_3#LIII-I)

**III. Manipuler les sessions**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

**III-A. Mise en garde**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

**Les registers\_globals[▲](https://php.developpez.com/cours/sessions/?page=page_3" \o "Haut de page)**

Attendez-vous éventuellement à des surprises de taille si les autoglobales sont activées sur votre serveur (directive **register\_globals** mise à **On** dans votre fichier php.ini). Ce comportement est aujourd'hui formellement déconseillé et déprécié ! Que peut-il donc se passer ? La réponse est simple : en plus des problèmes de sécurité évidents que cette configuration peut poser, vous risquez de modifier des variables de session sans même vous en rendre compte. Illustration avec un exemple :

Sélectionnez

<?php

[session\_start](https://www.php.net/manual/fr/function.session-start.php)();

if ([ini\_get](https://www.php.net/manual/fr/function.ini-get.php" \t "_blank)('register\_globals') == '1') {

[echo](https://www.php.net/manual/fr/function.echo.php) 'Vous devriez mettre <b>register\_globals</b> à <b>Off</b><br/>';

}

$\_SESSION['login'] = 'toto';

$login = 'titi';

// Affiche :

// - 'titi' avec register\_globals à On donc modification, car dans CE cas

// $\_SESSION['login'] et $login représente la même variable

// - 'toto' avec register\_globals à Off

[echo](https://www.php.net/manual/fr/function.echo.php) $\_SESSION['login'];

?>

À noter que le chargement des registers\_globals est fait avant le début du script. Il est donc inutile de tenter de les désactiver avec ini\_set(). Cependant, il est possible de le faire sur certains serveurs grâce au fichier .htaccess, grâce à la ligne ci-dessous :

.htaccess

Sélectionnez

php\_flag register\_globals Off

**Les fonctions session\_register(), session\_unregister(), session\_is\_registered() et session\_unset()**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

Vous pouvez trouver de nombreux tutoriels utilisant les fonctions session\_register(), session\_unregister(), session\_is\_registered().

Il **ne faut plus** utiliser ces fonctions. Elles appartiennent à d'anciennes versions de PHP et sont désormais dépréciées. Elles ne doivent plus être utilisées !

De même, la fonction session\_unset() qui, bien qu'elle ne soit pas officiellement dépréciée, est très fortement déconseillée, car elle suit la même optique que les trois précédentes…

**III-B. Débuter une session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

**Il existe deux façons différentes d'ouvrir une session :**

* automatique sur toutes vos pages si l'option session.auto\_start est égale à 1 ou on. Cette option a pour valeur par défaut 0 ou off.
* Extrait du fichier php.ini :

Sélectionnez

session.auto\_start = 0 ; initialize session on request startup

Extrait du phpinfo :

|  |  |  |
| --- | --- | --- |
| **Directive** | **Local Value** | **Master Value** |
| session.auto\_start | Off | Off |

* démarrage explicite et manuel de la session en utilisant la fonction session\_start(). Elle permet non seulement de créer une session, mais aussi de restaurer une session via l'identifiant de session. Dans la mesure du possible, je vous recommande donc d'utiliser cette fonction pour démarrer les sessions. Elle vous permettra de récupérer la session existante ou d'en créer une si aucune session ne correspond à l'utilisateur.  
  Un appel à session\_start() va faire générer à PHP des en-têtes HTTP pour la réponse : il va spécifier au client qu'il ne veut pas qu'il mette la page en cache (on verra ça plus bas).  
  S'il n'avait pas reçu d'identifiant de session, par cookie ou trans-id, PHP va alors générer aussi un en-tête HTTP set-cookie, pour lui passer l'identifiant qu'il viendra de générer fraichement pour nous (sauf si on désactive le cookie, voir plus bas).  
  C'est pour cela, que comme on l'a expliqué plus haut avec setCookie(), session\_start() doit être appelé avant toute sortie (echo, print ou autre).

° Cette fonction renvoie toujours vrai, inutile de tester son retour.

° Si vous souhaitez nommer la session, vous devez utiliser session\_name() avant session\_start().

° Si vous utilisez des sessions basées sur les cookies, alors il faut utiliser la fonction session\_start() comme une fonction d'en-tête, c'est-à-dire que rien ne doit être affiché avant.

**III-C. Sauvegarder une variable**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

Les variables de sessions sont, tout simplement, stockées dans le tableau super-global $\_SESSION. Vous pouvez y stocker des variables de type chaîne, entier, flottant, caractère, etc. ainsi que les objets et tableaux dont les attributs ou le contenu correspondent aux types définis ci-avant (sous certaines conditions au niveau des objets notamment).

Sélectionnez

<?php

[session\_start](https://www.php.net/manual/fr/function.session-start.php)(); // Création de la session

$\_SESSION['prenom'] = 'Jean-Pierre'; // Sauvegarde dans la session créée de la variable "prenom"

?>

**III-D. Récupération de données dans une session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

Quand on démarre une session avec session\_start(), le tableau super-global $\_SESSION est automatiquement initialisé avec les variables de la session. S'il contenait quelque chose, ce contenu n'est plus disponible après.

Sélectionnez

<?php

[echo](https://www.php.net/manual/fr/function.echo.php) $\_SESSION['prenom'];

?>

**III-E. Savoir si une variable appartient à une session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

Il est utile de savoir si des variables de session sont bel et bien enregistrées dans la session. Si ce n'est pas le cas, dans le cadre d'un script d'identification, nous allons demander à l'utilisateur de s'identifier. Pour cela, nous utilisons sur le tableau $\_SESSION la fonction isset() qui renvoie vrai si la variable passée en argument existe réellement. Par exemple, nous allons reprendre la même variable de session que précédemment : prenom.

Sélectionnez

<?php

if ([isset](https://www.php.net/manual/fr/function.isset.php" \t "_blank)($\_SESSION['prenom'])) {

[echo](https://www.php.net/manual/fr/function.echo.php) 'La variable prenom est déjà enregistrée !';

// On est certain de pouvoir y accéder ici

} else {

[echo](https://www.php.net/manual/fr/function.echo.php) 'La variable prenom n\'est pas enregistrée !';

}

?>

**III-F. Supprimer une variable d'une session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

Pour supprimer une variable, uniquement, d'une session, on utilise la fonction unset(), qui prend en paramètre la variable à détruire. Nous allons ici continuer la série d'exemples par la destruction de la variable de session prenom.

Sélectionnez

<?php

[unset](https://www.php.net/manual/fr/function.unset.php)($\_SESSION['prenom']); // La variable de session "prenom" a été supprimée, on ne peut plus y avoir accès !

?>

**III-G. Détruire une session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

La fonction session\_destroy(), qui ne prend aucun paramètre et qui renvoie vrai en cas de succès et faux en cas d'erreur, est utilisée lorsque vous souhaitez vous déconnecter. Elle s'emploie donc ainsi :

Sélectionnez

<?php

if ([session\_destroy](https://www.php.net/manual/fr/function.session-destroy.php" \t "_blank)()) {

[echo](https://www.php.net/manual/fr/function.echo.php) 'Session détruite !';

} else {

[echo](https://www.php.net/manual/fr/function.echo.php) 'Erreur : impossible de détruire la session !';

}

?>

**III-H. Détruire toutes les variables d'une session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

Il est aussi possible de détruire toutes les variables de session, ce qui permet de conserver votre session : il suffit de réinitialiser le tableau $\_SESSION.

Il ne faut jamais utiliser unset() directement sur $\_SESSION, cela rendrait impossible l'accès aux variables de la session courante jusqu'à sa fermeture.

Sélectionnez

<?php

$\_SESSION = array(); // $\_SESSION est désormais un tableau vide, toutes les variables de session ont été supprimées

?>

**III-I. Régénérer les identifiants de session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_3)

Il est possible de demander à PHP un nouvel identifiant de session grâce à la fonction session\_regenerate\_id(). Les données ne sont alors pas effacées de la session en cours. L'identifiant de session est simplement modifié, l'ancien devenant invalide. Cela est utile lorsque, par exemple, l'utilisateur change de niveau d'accès, s'il a accès à de nouvelles possibilités ou au contraire n'a plus accès à d'anciennes pages. De cette manière, si quelqu'un avait réussi à trouver l'ancien identifiant de session, il n'aura pas accès aux nouveaux privilèges de l'utilisateur.  
À l'inverse, si les identifiants de sessions sont régénérés trop souvent, cela peut poser des problèmes au niveau du référencement des pages web, surtout dans les cas où ils sont passés via l'URL.

**IV. Aller plus loin**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

**IV-A. Utiliser plusieurs sessions dans la même page**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

Il est impossible d'ouvrir simultanément plusieurs sessions. Cependant, on peut tout à fait ouvrir plusieurs sessions l'une après l'autre. Dans ce cas, il faut fermer la première session sans la détruire, grâce à session\_write\_close(), puis assigner les nouveaux session\_name et session\_id, et enfin ouvrir la nouvelle session avec session\_start().

Sélectionnez

<?php

[session\_name](https://www.php.net/manual/fr/function.session-name.php)('utilisateur');

[session\_start](https://www.php.net/manual/fr/function.session-start.php)(); // Création de la première session

[...] // Utilisation de la première session

[session\_write\_close](https://www.php.net/manual/fr/function.session-write-close.php)(); // Fermeture de la première session, ses données sont sauvegardées.

[session\_name](https://www.php.net/manual/fr/function.session-name.php)('admin'); // Indication du nom de la seconde session

[session\_start](https://www.php.net/manual/fr/function.session-start.php)(); // Ouverture de la seconde session

[...] // Utilisation de la seconde session.

?>

On peut refaire la manipulation autant de fois que l'on veut.

Une fois la session fermée, il est toujours possible d'accéder en lecture (les modifications ne seront pas prises en compte) aux variables de l'ancienne session. $\_SESSION ne sera vidé et prérempli qu'au prochain appel à session\_start().

**IV-B. Configuration liée aux sessions**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

**La transmission de l'identifiant de session**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

Comme vu au début, un identifiant unique identifie chaque session. PHP se charge lui-même de transmettre cet identifiant d'une page à l'autre, mais on peut indiquer explicitement comment le faire à l'aide de quatre directives de configurations :

* [session.use\_cookies](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-cookies) : cette option fait stocker l'identifiant (et seulement l'identifiant) dans un cookie si elle est égale à 1. Sa valeur par défaut est 1. Si l'option est à 0, et qu'un cookie de session est présent dans le navigateur, il sera simplement ignoré ;
* [session.use\_only\_cookies](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-only-cookies) : si cette option est égale à 1, alors PHP ignorera les identifiants transmis via l'URL pour n'utiliser que ceux contenus dans les cookies. Sa valeur par défaut est 0 ;
* [session.use\_trans\_sid](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-trans-sid) : si cette option est égale à 1, alors PHP insèrera automatiquement l'identifiant dans tous les liens non absolus (ne commençant pas par http:// ou ftp:// ou mailto: ou d'autres liens absolus, les liens commençant par un / (relatifs à la racine du site) seront eux pourvus de l'identifiant.) Sa valeur par défaut est 0 ;
* [session.url\_rewriter\_tags](http://fr.php.net/manual/fr/ref.session.php#ini.session.url-rewriter-tags) : indique où et dans quelles balises HTML insérer l'identifiant de session dans le cas où [session.use\_trans\_sid](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-trans-sid) est à 1. Sa valeur par défaut est : a=href,area=href,frame=src,input=src,form=fakeentry,fieldset= Le format est balise1=attribut1,balise2=attribut2,… Le cas des balises form et fieldset est particulier, car cela se traduit en fait par l'apparition d'un champ input (de type hidden) supplémentaire.

Extrait du phpinfo :

|  |  |  |
| --- | --- | --- |
| **Directive** | **Local Value** | **Master Value** |
| session.use\_cookies | On | On |
| session.use\_only\_cookies | Off | Off |
| session.use\_trans\_id | On | On |
| session.url\_rewriter\_tags | a=href, area=href, frame=src, input=src, form=fakeentry, fieldset= | a=href, area=href, frame=src, input=src, form=fakeentry, fieldset= |

Voici deux exemples de configuration pour ces options :

1. **Le premier est celui qui présente le plus de chances de faire transiter l'identifiant :**
   * [session.use\_cookies](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-cookies) à 1 ;
   * [session.use\_only\_cookies](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-only-cookies) à 0 ;
   * [session.use\_trans\_sid](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-trans-sid) à 1 ;
   * [session.url\_rewriter\_tags](http://fr.php.net/manual/fr/ref.session.php#ini.session.url-rewriter-tags) à  
     a=href,area=href,frame=src,iframe=src,input=src,form=fakeentry,fieldset=   
     Si vous souhaitez produire une page respectant les standards XHTML, nous vous conseillons plutôt la valeur : a=href,area=href,frame=src,iframe=src,input=src,fieldset= et d'utiliser une balise fieldset dans vos balises form ;
2. **Le second est celui qui est totalement invisible à l'œil d'un visiteur classique mais, si les cookies sont désactivés, il ne fonctionnera pas.**
   * [session.use\_cookies](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-cookies) à 1 ;
   * [session.use\_only\_cookies](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-only-cookies) à 1 ;
   * [session.use\_trans\_sid](http://fr.php.net/manual/fr/ref.session.php#ini.session.use-trans-sid) à 0 ;
   * [session.url\_rewriter\_tags](http://fr.php.net/manual/fr/ref.session.php#ini.session.url-rewriter-tags) à vide.

Il faut savoir que faire passer l'identifiant dans l'URL ([session.use\_trans\_sid](http://fr.php.net/manual/fr/ref.session.php" \l "ini.session.use-trans-sid" \t "_blank) à 1) peut présenter des risques. En effet, quelqu'un copiant un lien pour un ami lui donnerait par là-même son identifiant de session et sa session serait ouverte quand on cliquerait sur le lien.

**La sécurité des sessions**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

**Il existe plusieurs directives de configuration permettant de jouer sur la sécurité des sessions, en voici quelques-unes.**

* **Vérifier la provenance du visiteur**  
  session.referer\_check permet de spécifier une chaîne de caractères qui doit être présente dans le « referer » (adresse de la page de provenance) si celui-ci est indiqué par le navigateur. Une bonne idée est d'indiquer le nom de domaine de votre site (par exemple). Par contre, gardez à l'idée que cela ne permet pas d'être sûr que le visiteur provient d'une des pages de votre site. En effet, le referrer provient du protocole HTTP, lequel est très facile à manipuler !
* **Augmenter la complexité de l'identifiant de session**  
  session.hash\_function permet de choisir entre deux algorithmes de création des identifiants de sessions : MD5 (mettre la valeur 0) ou SHA-1 (mettre la valeur 1), le second permet de générer des identifiants de sessions plus longs.  
  session.hash\_bits\_per\_caracter permet de définir les plages de caractères utilisées pour l'identifiant de session 4 pour les caractères '0' à '9' et 'a' à 'f', 5 pour '0'-'9' et 'a'-'v' et 6 pour '0'-'9', 'a'-z', 'A'-'Z', '-' et ','. Il s'agit uniquement d'une représentation de l'identifiant, en aucun cas ça ne modifie sa complexité intrinsèque. Ces directives de configuration ne sont disponibles que depuis PHP 5.
* **Modifier la façon dont sont formatées les données de sessions**  
  session.serialize\_handler permet de modifier la façon dont sont formatées les données de sessions avant d'être sauvegardées. Par défaut, (valeur php) c'est une sérialization proche de la fonction serialize qui est utilisée. Si votre installation de PHP le supporte, vous pouvez utiliser WDDX (valeur WDDX), qui permet de récupérer un format XML.
* **Modifier la façon dont sont sauvegardées les données de sessions**  
  session.save\_handler permet de modifier la façon dont sont sauvegardées les données de sessions. La configuration actuelle ne permet pas de choisir d'autres « handler » que celui par défaut (dans un fichier non crypté, valeur files). Cependant, il existe une fonction PHP qui permet de définir un nouveau handler, voir un peu plus loin.
* **Modifier les données utilisées pour créer les identifiants de sessions**  
  session.entropy\_file permet de spécifier un fichier dont seront extraites des données permettant de rendre aléatoire l'identifiant de session généré. En général, c'est un « fichier » spécial qui est utilisé, dont le contenu est différent et aléatoire à chaque fois qu'on tente de le lire (/dev/random ou /dev/urandom sous unix par exemple).  
  session.entropy\_length permet de spécifier le nombre de caractères qui seront lus dans ce fichier.
* **Modifier la configuration du cookie stockant les identifiants de sessions**  
  session.cookie\_lifetime, session.cookie\_path et session.cookie\_domain permet de modifier la durée de vie, le chemin et le domaine attribués à un cookie.  
  session.cookie\_secure restreint l'utilisation du cookie aux connexions HTTPS (connexions cryptées).  
  session.cookie\_httponly restreint l'accès au cookie via le protocole HTTP. Autrement dit, les langages de script côté client (Javascript par exemple) n'auront pas accès au cookie. Ce n'est pas supporté par tous les navigateurs, et disponibles uniquement depuis PHP5.
* **Modifier la mise en cache des pages par le navigateur et les proxys**  
  session.cache\_limiter permet d'autoriser ou non la mise en cache des pages Web par le navigateur et les proxies. nocache (valeur par défaut) interdit la mise en page, private l'autorise uniquement pour les navigateurs et public pour tout le monde. Certains navigateurs peuvent rencontrer quelques problèmes avec l'expiration du cache : dans ce cas, il vaut mieux spécifier private\_no\_expire.  
  session.cache\_expire permet de spécifier la durée de vie des pages web mises en cache.

Bien qu'il soit possible de crypter les sessions, elles sont stockées en tant que fichiers sur le serveur Web. À ce titre, elles n'ont pas de système d'identification (login + password spécifiques) comme le shell et la base de données. Nous étudierons plus loin une solution permettant de modifier la manière dont sont enregistrées les sessions (et ainsi d'utiliser la base de données).

**La durée de vie des sessions**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

Pour modifier la durée de vie des sessions, en plus de jouer sur la durée de vie du cookie et du cache, on peut également jouer sur le temps au-delà duquel PHP considèrera les données stockées comme obsolètes.

**Pour cela, il existe trois directives de configuration :**

* session.gc\_maxlifetime : il s'agit effectivement de la durée au-delà de laquelle des données de session seront considérées comme périmées ;
* session.gc\_probability : la vérification du temps de vie des données n'est pas systématique, c'est lancé aléatoirement selon une fréquence prédéfinie. Il s'agit de la probabilité de déclenchement de l'opération ;
* session.gc\_divisor : pour obtenir la fréquence de lancement de la procédure de vérification des données, il faut diviser la probabilité par ce paramètre. Par exemple, si gc\_probability vaut 1 et que gc\_divisor vaut 100, alors ce sera lancé 1 fois toutes les 100 ouvertures de session.

**Modifier localement la configuration des sessions**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

Comme pour toutes les directives de configuration de PHP, il est possible de modifier la configuration des sessions sur certains serveurs. On utilise pour cela le fichier .htaccess. Pour modifier la valeur d'une directive de configuration, il suffit d'indiquer php\_value ou php\_flag (dans le cas d'une directive de type on/off), suivi du nom de la directive de configuration et de sa nouvelle valeur (en séparant le tout par des espaces).  
Cependant, gardez à l'esprit que cela dépend entièrement de la configuration du serveur, il est même possible que seule une partie des directives soit modifiable, voire aucune…

**IV-C. Modifier la façon dont sont stockées les données de sessions**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

Comme indiqué plus haut, PHP ne fournit pas de gestionnaire de données de sessions autres que des fichiers non cryptés. Il est cependant possible de définir vous-même les opérations à faire pour ouvrir, fermer, lire, écrire et vérifier la durée de vie d'une session.  
Cette fonctionnalité vous permettra par exemple de crypter et de stocker très facilement vos sessions dans une base de données (par exemple) en écrivant les fonctions de requêtage adéquates. Une fois ceci fait, vous n'aurez alors plus besoin de définir les opérations d'accès à la base de données pour les sessions !  
Vous trouverez un peu plus bas un exemple de fonctions vous permettant de stocker les sessions dans une base de données.

Dans ce cas, il vous faut alors définir ces opérations au début de chaque script (avant le session\_start) !

La fonction permettant de faire ceci se nomme session\_set\_save\_handler() et prend en argument les noms des fonctions permettant les opérations citées ci-dessus (dans le même ordre). L'exemple ci-dessous ne fait que redéfinir les opérations par défaut de PHP, pas très utile, certes, mais ça vous permet de voir comment ça fonctionne…

Sélectionnez

<?php

function ouvrir\_session($chemin\_de\_stockage, $nom\_de\_session)

{

$\_ENV['chemin\_de\_stockage\_des\_sessions'] = $chemin\_de\_stockage;

return true;

}

function fermer\_session()

{

return true; // Rien à faire

}

function lire\_session($identifiant\_de\_session)

{

return [strval](https://www.php.net/manual/fr/function.strval.php)([file\_get\_contents](https://www.php.net/manual/fr/function.file-get-contents.php)($\_ENV['chemin\_de\_stockage\_des\_sessions'].'/sess\_'.$identifiant\_de\_session));

}

function ecrire\_session($identifiant\_de\_session, $donnees\_de\_session)

{

if ($fp = [fopen](https://www.php.net/manual/fr/function.fopen.php)($\_ENV['chemin\_de\_stockage\_des\_sessions'].'/sess\_'.$identifiant\_de\_session, "w")) {

$return = [fwrite](https://www.php.net/manual/fr/function.fwrite.php)($fp, $donnees\_de\_session);

[fclose](https://www.php.net/manual/fr/function.fclose.php)($fp);

return $return;

} else {

return false;

}

}

function detruire\_session($identifiant\_de\_session)

{

return [unlink](https://www.php.net/manual/fr/function.unlink.php)($\_ENV['chemin\_de\_stockage\_des\_sessions'].'/sess\_'.$identifiant\_de\_session);

}

function verifier\_validite\_session($temps\_de\_validite)

{

$files = [glob](https://www.php.net/manual/fr/function.glob.php)($\_ENV['chemin\_de\_stockage\_des\_sessions'].'/sess\_\*');

if(![empty](https://www.php.net/manual/fr/function.empty.php)($files))

{

foreach($files as $file)

{

// Il peut s'agir d'un répertoire ou d'un fichier, mais seuls les fichiers nous intéressent

if([is\_file](https://www.php.net/manual/fr/function.is-file.php" \t "_blank)($file) and [filemtime](https://www.php.net/manual/fr/function.filemtime.php)($file) + $sessionTime < [time](https://www.php.net/manual/fr/function.time.php)())

{

[unlink](https://www.php.net/manual/fr/function.unlink.php)($file);

}

}

}

/\*foreach ($\_ENV['chemin\_de\_stockage\_des\_sessions'].'/sess\_\*') as $fichier\_de\_session) {

if (filemtime($fichier\_de\_session) + $temps\_de\_validite < time()) {

unlink($fichier\_de\_session);

}

}\*/

return true;

}

[session\_set\_save\_handler](https://www.php.net/manual/fr/function.session-set-save-handler.php)(

'ouvrir\_session',

'fermer\_session',

'lire\_session',

'ecrire\_session',

'detruire\_session',

'verifier\_validite\_session'

);

[session\_start](https://www.php.net/manual/fr/function.session-start.php)();

// Maintenant on peut se servir des sessions sans modifications par rapport à avant...

// C'est complètement transparent pour l'utilisateur.

?>

Nous allons maintenant voir un exemple de fonctions permettant de stocker vos sessions dans une base de données. Nous utiliserons MySQL comme [SGBD](https://dico.developpez.com/html/786-Generalites-SGBD-Systeme-de-Gestion-de-Bases-de-Donnees.php) et MySQLi comme extension PHP pour communiquer avec MySQL. Des commentaires indiquent comment remplacer MySQLi par MySQL.

Bien sûr, il ne s'agit ici que d'un exemple ! Ces fonctions ne sont pas parfaitement sécurisées ni optimisées !

Il ne faut surtout pas, par exemple, les utiliser ainsi dans un environnement de production.

Création de la table pour stocker les données

Sélectionnez

USE my\_database;

CREATE TABLE `sessions` (

`identifiant` VARCHAR( 64 ) NOT NULL ,

`nom` VARCHAR( 64 ) NOT NULL ,

`dernier\_acces` TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ,

`donnees` TINYTEXT NOT NULL ,

PRIMARY KEY ( `identifiant` , `nom` )

) ENGINE = MYISAM ;

Les fonctions d'accès en elles-mêmes

Sélectionnez

<?php

/\*\*

\* SGBD est une constante contenant l'adresse du serveur MySQL. [typiquement 'localhost']

\* LOGIN\_SGBD est une constante contenant votre login pour la connexion au serveur MySQL.

\* MOT\_DE\_PASSE\_SGBD est une constante contenant votre mot de passe pour la connexion au serveur MySQL.

\* SESSIONS\_BDD est une constante contenant le nom de la base de données où sont stockées les données. [ici, 'my\_database']

\* SESSION\_TABLE est une constante contenant le nom de la table où sont stockées les données. [ici, 'sessions']

\*/

function ouvrir\_session($chemin\_de\_stockage, $nom\_de\_session)

{

$\_ENV['nom\_de\_session'] = $nom\_de\_session;

return true;

}

function fermer\_session()

{

return true; // Rien à faire

}

function lire\_session($identifiant\_de\_session)

{

$mysql = new mysqli(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD, SESSIONS\_BDD);

//$mysql = mysql\_connect(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD); mysql\_select\_db(SESSIONS\_BDD, $mysql);

if(!$mysql)

{

return '';

}

$resultat = $mysql->[query](https://www.php.net/manual/fr/function.query.php)('SELECT donnees FROM '.SESSIONS\_TABLE

.' WHERE identifiant = \''.$identifiant\_de\_session.'\' AND nom =\''.$\_ENV['nom\_de\_session'].'\'');

//$resultat = mysql\_query('SELECT donnees FROM '.SESSIONS\_TABLE

// .' WHERE identifiant = \''.$identifiant\_de\_session.'\' AND nom =\''.$\_ENV['nom\_de\_session'].'\'', $mysql);

$donnes = '';

if($resultat->num\_rows == 1) //if(mysql\_num\_rows($resultat) == 1)

$donnees = $result->fetch\_array(); //$donnees = mysql\_fetch\_array($resultat);

$resultat->[free](https://www.php.net/manual/fr/function.free.php)(); //mysql\_free\_result($resultat);

$mysql->[close](https://www.php.net/manual/fr/function.close.php)(); //mysql\_close($mysql);

return [strval](https://www.php.net/manual/fr/function.strval.php)($donnes['donnees']);

}

function ecrire\_session($identifiant\_de\_session, $donnees\_de\_session)

{

$mysql = new mysqli(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD, SESSIONS\_BDD);

//$mysql = mysql\_connect(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD); mysql\_select\_db(SESSIONS\_BDD, $mysql);

if(!$mysql)

{

return false;

}

$result = true;

$mysql->[query](https://www.php.net/manual/fr/function.query.php)('UPDATE '.SESSIONS\_TABLE.' SET donnees = \''.$donnees\_de\_session.'\''

.' WHERE identifiant = \''.$identifiant\_de\_session.'\' AND nom =\''.$\_ENV['nom\_de\_session'].'\'') or $result = false;

//mysql\_query('UPDATE '.SESSIONS\_TABLE.' SET donnees = \''.$donnees\_de\_session.'\''

// .' WHERE identifiant = \''.$identifiant\_de\_session.'\' AND nom =\''.$\_ENV['nom\_de\_session'].'\'', $mysql) or $result = false;

$mysql->[close](https://www.php.net/manual/fr/function.close.php)(); //mysql\_close($mysql);

return $result;

}

function detruire\_session($identifiant\_de\_session)

{

$mysql = new mysqli(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD, SESSIONS\_BDD);

//$mysql = mysql\_connect(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD); mysql\_select\_db(SESSIONS\_BDD, $mysql);

if(!$mysql)

{

return false;

}

$result = true;

$mysql->[query](https://www.php.net/manual/fr/function.query.php)('DELETE FROM '.SESSIONS\_TABLE.' WHERE identifiant = \''.$identifiant\_de\_session.'\''

.' AND nom =\''.$\_ENV['nom\_de\_session'].'\'') or $result = false;

//mysql\_query('DELETE FROM '.SESSIONS\_TABLE.' WHERE identifiant = \''.$identifiant\_de\_session.'\''

// .' AND nom =\''.$\_ENV['nom\_de\_session'].'\'', $mysql) or $result = false;

$mysql->[close](https://www.php.net/manual/fr/function.close.php)(); //mysql\_close($mysql);

return $result;

}

function verifier\_validite\_session($temps\_de\_validite)

{

$mysql = new mysqli(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD, SESSIONS\_BDD);

//$mysql = mysql\_connect(SGBD, LOGIN\_SGBD, MOT\_DE\_PASSE\_SGBD); mysql\_select\_db(SESSIONS\_BDD, $mysql);

if(!$mysql)

{

return false;

}

$result = true;

$mysql->[query](https://www.php.net/manual/fr/function.query.php)('DELETE FROM '.SESSIONS\_TABLE.' WHERE ADDDATE(dernier\_acces, INTERVAL '.$temps\_de\_validite.' SECOND) < NOW()')

or $result = false;

//mysql\_query('DELETE FROM '.SESSIONS\_TABLE.' WHERE ADDDATE(dernier\_acces, INTERVAL '.$temps\_de\_validite.' SECOND) < NOW()', $mysql)

or $result = false;

$mysql->[close](https://www.php.net/manual/fr/function.close.php)(); //mysql\_close($mysql);

return $result;

}

[session\_set\_save\_handler](https://www.php.net/manual/fr/function.session-set-save-handler.php)(

'ouvrir\_session',

'fermer\_session',

'lire\_session',

'ecrire\_session',

'detruire\_session',

'verifier\_validite\_session'

);

[session\_start](https://www.php.net/manual/fr/function.session-start.php)();

// Maintenant on peut se servir des sessions sans modifications par rapport à avant...

// C'est complètement transparent pour l'utilisateur.

?>

**IV-D. À savoir**[**▲**](https://php.developpez.com/cours/sessions/?page=page_4)

* L'hébergeur Free est une exception puisque, pour pouvoir utiliser les sessions, il vous faudra créer à la racine de votre FTP un répertoire nommé **sessions**.
* La fonction session\_id() utilisée sans paramètre renvoie l'identificateur actuel de la session et la fonction session\_name() utilisée sans paramètre renvoie le nom courant de la session. Ainsi, pour passer dans un lien l'identificateur de session, ce qui permet de récupérer la session de l'utilisateur même s'il change de répertoire, il suffit de procéder ainsi :

Sélectionnez

<?php

[echo](https://www.php.net/manual/fr/function.echo.php) 'http://adresse\_de\_votre\_site?' . [session\_name](https://www.php.net/manual/fr/function.session-name.php)() . '=' . [session\_id](https://www.php.net/manual/fr/function.session-id.php)();

?>

Copyright © 2006 Developpez.com Developpez LLC. Tous droits réservés Developpez LLC. Aucune reproduction, même partielle, ne peut être faite de ce site ni de l'ensemble de son contenu : textes, documents et images sans l'autorisation expresse de Developpez LLC. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

**V. Conclusion**[**▲**](https://php.developpez.com/cours/sessions/?page=page_5)

**V-A. Épilogue**[**▲**](https://php.developpez.com/cours/sessions/?page=page_5)

Nous avons vu ce qu'est une session, comment l'utiliser et la configurer. Cependant, nous souhaitons attirer votre attention sur le fait que, malgré leur localisation côté serveur, vous n'êtes pas à l'abri de l'usurpation d'identité par vol de l'identifiant de session. Il existe des méthodes pour s'en prémunir, voir les liens ci-dessous, mais cela dépasse le cadre de cet article. En effet, l'aspect sécurité a principalement été abordé au travers de la configuration globale des sessions.

**Liens Developpez :**

* [Utiliser une base de données pour sécuriser vos sessions](https://a-pellegrini.developpez.com/tutoriels/php/session-db/), par Adrien Pellegrini
* [Comment débuter avec MySQL et PHP](https://eusebius.developpez.com/phpmysql-base/), par Eusebius

**Liens externes :**

* [Le manuel PHP sur les sessions](http://fr.php.net/manual/fr/ref.session.php)

**V-B. Remerciements**[**▲**](https://php.developpez.com/cours/sessions/?page=page_5)

Nous adressons nos remerciements à [Yogui](https://www.developpez.net/forums/u30724/yogui/) pour ses recommandations techniques avisées.

Copyright © 2006 Developpez.com Developpez LLC. Tous droits réservés Developpez LLC. Aucune reproduction, même partielle, ne peut être faite de ce site ni de l'ensemble de son contenu : textes, documents et images sans l'autorisation expresse de Developpez LLC. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.