



Les shortcodes sont les partenaires idéals pour intégrer des contenus provenant de la base de données WordPress, d'une base externe, de fichiers de données ou d'une API.

Cet article vous présente un petit exemple de shortcode qui génère une liste de personnes à partir d'une simple table client.

## Avant de commencer

1. Vous devez connaître et maîtriser la gestion de bases de données ainsi que l'extension PHP PDO <http://eilgin.github.io/php-the-right-way/#databases>
2. Créer la table client sur un serveur BD atteignable par votre installation WordPress : [script de création de la table client](#)
3. Créer un dossier `gest-cli` dans le dossier `wp-content/plugins`

## Création du plugin « gest-cli »

### Structure du dossier `gest-cli`

```
gest-cli
├── client.php    //Fonctions métiers "gestion table client"
├── config.php   //Fichier de configuration
├── db.php       //Connexion à la base de données
└── gest-cli.php //Fichier principal du plugin
```

### **config.php**

Fichier de configuration du plugin.



```
<?php
//Base de données
define('DB_HOST', 'localhost'); //Adresse du serveur de BD
define('DB_NAME', 'clients'); //Nom de la BD
define('DB_CHARSET', 'utf8'); //Encodage de la connexion BD
define('DB_PORT', '3306'); //3306 port par défaut MySQL
define('DB_USER', 'votre-login');
define('DB_PASS', 'votre-mot-de-passe-secret');

//Date et heure
date_default_timezone_set('Europe/Zurich');
setlocale(LC_TIME, 'fr_FR', 'fra');
```

## db.php

Tente de créer une connexion à la base de données avec PDO et stocke l'objet retourné dans la variable globale `$db`.

Travailler avec une variable globale évite de devoir se connecter et se déconnecter plusieurs fois durant le même script et limite le nombre de requêtes.

```
<?php
//Variable globale qui contiendra l'objet PDO
$GLOBALS['db'];
//Tentative de connexion à la base de données
try {
    //Chaine de connexion, serveur, base, encodage, port, user, pw
    $db = new PDO('mysql:host='.DB_HOST.';dbname='.DB_NAME.';
        . 'charset='.DB_CHARSET.';port='.DB_PORT,
        DB_USER, DB_PASS);
    //Active la gestion des erreurs
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```



```
} catch(Exception $e) {  
    //Cacher les messages d'erreur en production.  
    //Envoyer email ou fichier de log à la place  
    echo $e->getMessage() . "<br>";  
    echo $e->getFile() . "<br>";  
    echo $e->getLine() . "<br>";  
}
```

## client.php

Fichier contenant l'ensemble des fonctions métiers. En programmation POO, ce fichier définirait un objet métier Client, avec ses propriétés et méthodes.

Il est recommandé de préfixer le nom des tables avec le nom de votre base de données dans la clause FROM de vos requêtes SQL `FROM clients.client, clients.localite`

```
<?php  
/**  
 * Fonction qui retourne la "liste" des clients  
 * sous forme de tableau associatif  
 * en fonction des paramètres passés lors de l'appel  
 *  
 * Cette fonction permet de :  
 * - rechercher un client (chaîne trouvée dans nom ou prénom)  
 * - définir le tri  
 * - définir le nombre de clients à retourner  
 *  
 * @param string $s      chaîne de recherche dans le nom ou prénom  
 * @param string $tri    tri nom et prénom ASC par défaut  
 * @param int $limit    nombre de clients à afficher, 30 par défaut
```



```
* @return array|bool   tableau associatif de clients ou tableau vide
*/
function getClient($s='', $tri= 'client_nom ASC,
    client_prenom ASC', $limit = 30
){
    global $db;

    //Préparation et envoi de la requête de sélection
    try {
        $sql = "SELECT client_nom, client_prenom, client_genre "
            . "FROM clients.client "
            . "WHERE (client_nom LIKE :q OR client_prenom LIKE :q) "
            . "ORDER BY $tri "
            . "LIMIT :limit";

        //Préparation de la requête
        $req = $db->prepare($sql);

        //Passage des paramètres à la requête
        $req->bindValue('q', '%'.$s.'%');
        $req->bindParam('limit', $limit, PDO::PARAM_INT);

        //Exécution de la requête
        $req->execute();

        //Retourne les résultats sous forme de tableau associatif
        return $req->fetchAll();

    }
    catch(Exception $e)
    {
        //A améliorer
        //echo $e->getMessage()."<br>";
        //echo $e->getFile()."<br>";
        //echo $e->getLine()."<br>";
        return [];
    }
}
```



```
}  
}
```

## gest-cli.php

Finalement le fichier principal du plugin qui va :

1. Définir le nom du plugin : `/*Plugin Name: Gestion des clients*/`
  - Très important, cela définira que ce fichier sera chargé en premier par WordPress.
2. Inclure tous les fichiers PHP nécessaires
  - config.php
  - db.php
  - client.php

Toujours inclure vos fichiers avec le chemin complet en récupérant le chemin du dossier `__DIR__`

3. Déclarer la fonction de rappel du shortcode `[clients]`
  - Ce shortcode aura un attribut `s` qui permettra la recherche d'un client par son nom et prénom `[clients s="jean"]`
4. Enregistrer du shortcode `[clients]`

```
<?php  
/*  
Plugin Name: Gestion des clients  
*/
```



```
require_once __DIR__ . '/config.php';
require_once __DIR__ . '/db.php';
require_once __DIR__ . '/client.php';

/* shortcode qui affiche la liste des clients
 * Possède un attribut "s" qui recherche l'existence
 * d'une chaîne dans le nom ou le prénom
 *
 * [clients s="jean"]
 */
function gest_cli_clients_shortcode($atts) {
    //Récupération de l'attribut "search
    //Recherche dans nom ou prénom
    $s = isset($atts['s']) ? $atts['s'] : '';

    //Récupère le tableau des clients
    $clients = getClient($s);

    //Si pas de clients on retourne un message d'info
    if(empty($clients)){
        return "<p>Aucun client trouvé !</p>";
    }

    //Construction de la sortie HTML, <ul>
    $html = '<ul id="clients">';
    //Parcours et ajoute un <li> par client avec nom et prénom
    foreach ($clients as $client) {
        $html .= '<li>' . $client['client_prenom'] . ' '
            . $client['client_nom'] . '</li>';
    }
    //Fermeture de la liste </ul>
    $html .= '</ul>';

    //Retourne le code HTML du shortcode
    return $html;
}
```



```
//Enregistre les shortcodes du plugin
function gest_cli_register_shortcode() {
    add_shortcode( 'clients', 'gest_cli_clients_shortcode' );
}
add_action( 'init', 'gest_cli_register_shortcode' );
```

N'oubliez pas de vous protéger contre les injections en validant, nettoyant et échappant les données que vous recevez.

## Tests

Il ne reste plus qu'à tester votre nouveau plugin !

Pour ce faire :

1. Activez votre plugin dans l'administration WordPress (Menu Extensions)
2. Tester votre shortcode dans un article ou une page avec ces différentes variantes et comparer vos résultats :

- `[clients]`

```
<ul id="clients">
  <li>Laure Dinateur</li>
  <li>Jean Némarre</li>
  <li>Roger Plusdbièresdansfrigo</li>
  <li>Yves Remord</li>
  <li>Sandra Samegratte</li>
```



```
</ul>
```

- `[clients s="an"]`

```
<ul id="clients">
  <li>Jean Némarre</li>
  <li>Roger Plusdbièresdansfrigo</li>
  <li>Sandra Samegratte</li>
</ul>
```

- `[clients s="l'épave"]`

```
<p>Aucun client trouvé !</p>
```