

SAE 1.05 & SAE 1.06

Documentation Wordpress



Table des matières

| | |
|--|----|
| Installation d'un environnement | 3 |
| Récupération depuis une sauvegarde..... | 10 |
| Gestion des sauvegardes..... | 11 |
| Utilisateurs wordpress et droits/rôles associés..... | 12 |
| Chartre graphique | 13 |
| Fonctionnement de l'administration du site..... | 15 |
| Liste des pages | 16 |
| Liste des articles..... | 16 |
| Plugins | 16 |
| Gestion de projet | 18 |

Installation d'un environnement

Commençons par deroot avec la commande **>su** :

```
su
```

Mettons à jour les informations des dépôts de logiciels :

```
apt-get update
```

Apache 2

Installation d'apache avec la commande suivante afin d'héberger le site web :

```
apt-get install apache2
```

Appuyez sur la touche O puis sur Entrée pour valider.

A présent, vous allez pouvoir tester vos pages html et les déposer dans le répertoire utilisé par Apache :

/var/www/html

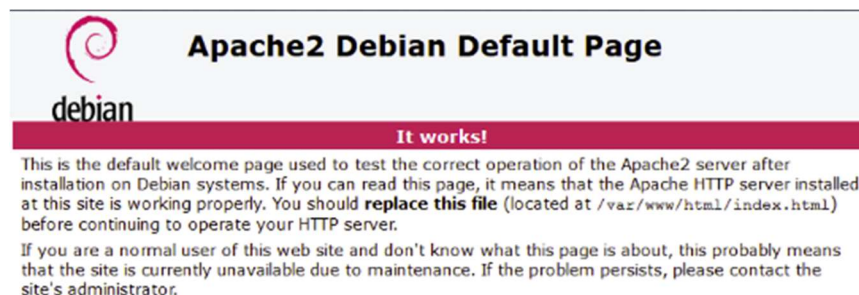
Pour cela, accordez les droits à votre userid :

>sudo chown -R user:user /var/www/html (remplacer user par votre userid)

Ensuite démarrez le serveur web avec la commande :

>sudo service apache2 start

Vous pouvez maintenant accéder à votre serveur web apache2 en tapant dans votre navigateur : **http://localhost**.



Installation de PHP et de MySQL

Sous linux, on installe d'abord php pour apache :

>sudo apt install php

puis MySQL :

>sudo apt install php-mysql

On vérifie que l'on a les droits en écriture sur /var/www/html. (sinon on fait manuellement un **sudo chmod 777 -R /var/www/html**)

Installation de MariaDB

Entrer la commande suivante :

>sudo apt install mariadb-server

Appuyer sur la touche Y puis sur Entrée pour procéder à l'installation.

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  galera-4 libcgi-fast-perl libcgi-pm-perl libclone-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmariadb3
  libmysqlclient21 libnuma1 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl liburing2 libwrap0
  mariadb-client mariadb-client-core mariadb-common mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4
  mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo mariadb-plugin-provider-snappy mariadb-server-core
  mysql-common pv socat
Suggested packages:
  libmldbm-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl
  libio-compress-brotli-perl libbusiness-isbn-perl libregexp-ipv6-perl libwww-perl mailx mariadb-test doc-base
The following NEW packages will be installed:
  galera-4 libcgi-fast-perl libcgi-pm-perl libclone-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libencode-locale-perl libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmariadb3
  libmysqlclient21 libnuma1 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl liburing2 libwrap0
  mariadb-client mariadb-client-core mariadb-common mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4
  mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo mariadb-plugin-provider-snappy mariadb-server
  mariadb-server-core mysql-common pv socat
0 upgraded, 40 newly installed, 0 to remove and 55 not upgraded.
Need to get 19.1 MB of archives.
After this operation, 198 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Rechargement du serveur Apache2

Après avoir installé PHP et le connecteur (php-mysql), on doit recharger la configuration d'Apache2 afin qu'il soit pris en compte. Pour ce faire, lancez la commande suivante :

```
service apache2 reload
```

Installation de WordPress

Tout d'abord, on doit se rendre sur le dossier qu'a créé Apache2 pour recevoir les pages PHP :

```
cd /var/www/html
```

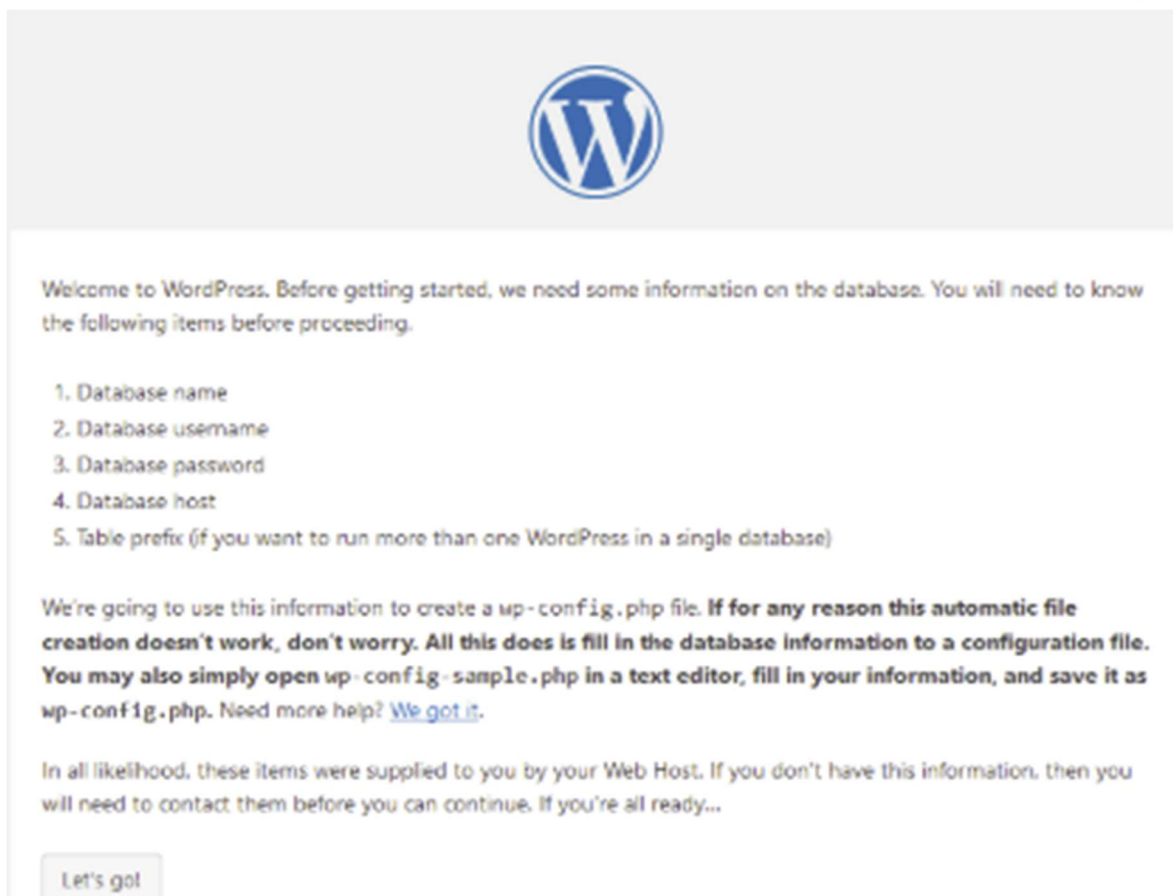
Pour récupérer le code de WordPress, on a besoin de **Git**. Par défaut, celui-ci n'est pas installé. On doit donc faire un :

```
apt-get install git
```

Après l'installation de **Git**, on peut lancer la commande qui va récupérer le code de WordPress :

```
git clone https://github.com/WordPress/WordPress.git
```

Si vous faites la commande **ls**, on peut voir un nouveau dossier WordPress qui a été créé. Vous pouvez y accéder avec un navigateur et en rajoutant dans la barre d'adresse **/WordPress** après **http://localhost**.



Création d'une base de données pour WordPress

"Pour ne pas être bloqué lors de l'installation de WordPress, nous allons créer en amont la base de données, ainsi qu'un utilisateur qui aura les droits de lecture/écriture. Pour cela, on doit d'abord se connecter à la base de données en console. Vous devez utiliser la commande suivante :

```
mysql -u root -p
```

Vous devez aussi taper votre mot de passe.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.1.26-MariaDB-0+deb9u1 Debian 9.1

Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Une fois connecté à la base de données, tapez les commandes suivantes
en changeant password par un mot de passe fort :

```
CREATE DATABASE wordpress;  
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER  
ON wordpress.*  
TO wordpress@localhost  
IDENTIFIED BY 'password';  
FLUSH PRIVILEGES;
```

Celles-ci vont nous permettre de créer la base de données et de donner les droits à l'utilisateur fraîchement créé. Le résultat sera le suivant :

```
Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> CREATE DATABASE wordpress;  
Query OK, 1 row affected (0.00 sec)  
  
MariaDB [(none)]> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER  
-> ON wordpress.*  
-> TO wordpress@localhost  
-> IDENTIFIED BY 'password';  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)  
  
MariaDB [(none)]>
```


On peut maintenant sortir de la base de données. Pour cela, tapez la commande :

>exit

Après toute la configuration que nous avons réalisée, nous allons effectuer l'installation grâce à l'interface web de WordPress.

Installation/Configuration de WordPress

Pour cela, vous devez retourner sur votre navigateur et taper l'url pour accéder à votre site : <http://adresseip/>. Vous arriverez sur la même page que précédemment. Ensuite, **cliquez sur le bouton c'est parti**. Vous obtiendrez la page suivante :



Below you should enter your database connection details. If you're not sure about these, contact your host.

| | | |
|---------------|--|--|
| Database Name | <input type="text" value="wordpress"/> | The name of the database you want to use with WordPress. |
| Username | <input type="text" value="wordpress"/> | Your database username. |
| Password | <input type="text" value="password"/> | Your database password. |
| Database Host | <input type="text" value="localhost"/> | You should be able to get this info from your web host, if localhost doesn't work. |
| Table Prefix | <input type="text" value="wp_"/> | If you want to run multiple WordPress installations in a single database, change this. |

Remplissez les différentes informations que nous avons renseignées précédemment, vous pouvez laisser la Table Prefix comme tel.

Copie du fichier de configuration dans WordPress

WordPress n'a pas le droit de créer un nouveau fichier de configuration. Il vous demande donc de créer le fichier pour lui et de coller les éléments nécessaires à son bon fonctionnement.



Sorry, but I can't write the wp-config.php file.

You can create the wp-config.php file manually and paste the following text into it.

```
* visit the Codex.
*
* @link https://codex.wordpress.org/Debugging_in_WordPress
*/
define( 'WP_DEBUG', false );

/* That's all, stop editing! Happy blogging. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', dirname( __FILE__ ) . '/' );
}

/** Sets up WordPress vars and included files. */
require_once( ABSPATH . 'wp-settings.php' );
```

After you've done that, click "Run the installation."

Run the installation

Pour ce faire, retournez sur le terminal et tapez les commandes suivantes :

```
cd /var/www/html/WordPress
ls
nano wp-config.php
```

Vous devez ensuite **coller** le contenu du code que **WordPress** vous a fourni. Dès que c'est fait, enregistrez le fichier et **cliquez sur le bouton Run the installation** dans votre navigateur.

Création d'un compte WordPress

WordPress vient d'initialiser la base de données de votre site. Il vous demande ensuite de créer un compte administrateur qui vous permettra de gérer votre site web.

Vous pouvez remplir les différents éléments de création de compte. Ensuite, vous pouvez **cliquer** sur le bouton **Install WordPress**.

Celui-ci va créer le compte et vous rediriger vers la page de connexion. Cliquez ensuite sur le bouton **Log In** après la connexion, vous arriverez sur la première page de votre magnifique site web.

Récupération depuis une sauvegarde

Pour effectuer une sauvegarde :

créez un dossier **savegardesitewordpressdatedujour** :

>mkdir savegardesitewordpressdatedujour

Copiez l'ensemble du dossier contenant les fichiers WordPress dans votre dossier de sauvegarde :

>cp -r /chemin/vers/votre/site/wordpress/ savegardesitewordpressdatedujour/

Ou tout simplement avec l'explorateur de fichier.

Puis, créez un dump de votre base de donnée avec la commande :

>mysqldump -u root -p wordpress > wordpress.sql

Placer ensuite le fichier dans votre dossier **savegardesitewordpressdatedujour**.

Compressez le dossier de sauvegarde :

>zip -r savegardesitewordpressdatedujour.zip savegardesitewordpressdatedujour/

Copier ensuite le fichier zippé sur une clé USB ou sur un drive par exemple.

Autre solution

Installez une extension sur votre site WordPress afin d'effectuer les sauvegardes facilement et les télécharger ensuite.

Un exemple d'extension : **All-in-One WP Migration**

Récupération et restauration d'une sauvegarde WordPress

Téléchargez le fichier de sauvegarde compressé (zip) sur votre machine si nécessaire.

Décompressez le fichier :

>unzip sauvegardesitewordpressdatedujour.zip

Vous obtiendrez :

- Un **dossier** contenant les fichiers WordPress.
- Un **fichier** wordpress.sql (dump de la base de données).

Naviguez jusqu'au dossier où WordPress doit être installé (/var/html/wordpress)

Remplacer les fichiers existants :

>rm -rf /chemin/vers/site/wordpress/*

> cp -r sauvegardesitewordpressdatedujour/* /chemin/vers/site/wordpress/

Importer la base de données :

>mysql -u root -p wordpress < chemin/vers/wordpress.sql

Autre solution

Utiliser l'extension **All-in-One WP Migration** par exemple et restaurer la sauvegarde en l'important sur le site.

Gestion des sauvegardes

Pourquoi sauvegarder ?

Les sauvegardes régulières permettent de récupérer rapidement votre site en cas de problème (panne serveur, piratage, erreur humaine).

Comment sauvegarder ?

- **Manuellement** : En copiant les fichiers et en exportant la base de données.
- **Automatiquement** : Avec des plugins comme **All-in-One WP Migration** ou via un script bash.

Où stocker les sauvegardes ?

- Stockage local (disque dur).
- Cloud (Google Drive, Dropbox).
- Hébergement externe.











Quand sauvegarder ?

Il faut essayer de sauvegarder le plus régulièrement possible afin de ne pas être embêté en cas de problème, surtout lorsque beaucoup de modifications ont été effectués.

Comment avons-nous fait les sauvegardes du site :

Les sauvegardes ont été réalisées tout au long de la SAE à l'aide de l'extension All-in-One WP Migration. Nous avons effectué des sauvegardes du site chaque fois qu'il y avait des avancées significatives. Ensuite, ces sauvegardes ont été transférées sur un disque dur externe pour pouvoir être utilisées en cas de besoin.

La liste des sauvegardes est indiquée ci-dessous :

| Name | Date modified | Type | Size |
|---|------------------|---------------------|------------|
|  4-234-161-76-wordpress-20241228-1538... | 28/12/2024 16:41 | WPRESS File | 596 003 KB |
|  4-234-161-76-wordpress-20241230-2023... | 30/12/2024 21:24 | WPRESS File | 603 549 KB |
|  4-234-161-76-wordpress-20250101-1806... | 01/01/2025 19:07 | WPRESS File | 681 897 KB |
|  4-234-161-76-wordpress-20250104-1345... | 04/01/2025 14:47 | WPRESS File | 736 380 KB |
|  4-234-161-76-wordpress-20250107-1901... | 07/01/2025 20:08 | WPRESS File | 769 143 KB |
|  4-234-161-76-wordpress-20250111-1140... | 11/01/2025 11:41 | WPRESS File | 787 239 KB |
|  4-234-161-76-wordpress-20250111-1331... | 11/01/2025 13:32 | WPRESS File | 795 233 KB |
|  4-234-161-76-wordpress-20250111-1523... | 11/01/2025 15:23 | WPRESS File | 799 698 KB |
|  4-234-161-76-wordpress-20250112-1107... | 12/01/2025 11:28 | WPRESS File | 816 080 KB |
|  sauvegardesitewordpress15-11-24.zip | 15/11/2024 11:17 | Compressed (zipp... | 72 320 KB |

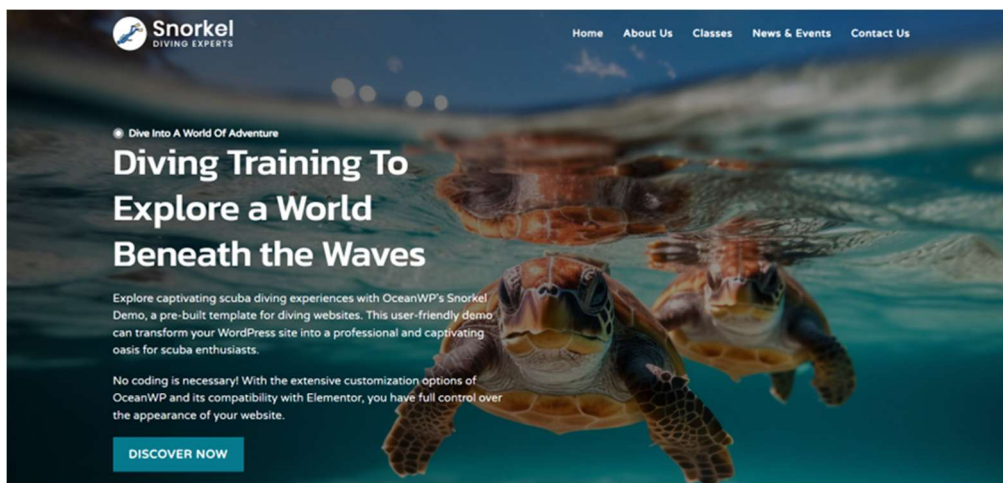
Utilisateurs wordpress et droits/rôles associés

Les utilisateurs WordPress sont au nombre de quatre, tous avec le même rôle d'administrateur. Ce rôle est adapté pour ce projet, car il facilite la mise en place des tâches et ne nécessite pas de validation par un rôle supérieur. Il permet de travailler librement sur WordPress.

Chartre graphique

Template

Le thème choisi pour ce site est *OceanWP*. Voici un aperçu d'une démonstration avec ce template :



Son design minimaliste et épuré nous a fait pencher pour ce plugin parfaitement adapté pour le site d'une ville telle que La Ciotat !

Nous avons également choisi ce template car il a été conçu pour être utilisé avec les plugins *Elementor* et *WooCommerce*, thème que nous avons utilisé pour la conception du site.

Logo

Nous avons retenu, pour le logo le blason de la ville qui représente selon nous le choix le plus pertinent car ces armoiries représentent depuis des siècles la ville de La Ciotat et son histoire.



Couleurs

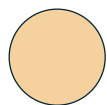
La palette de couleur choisie est la suivante :



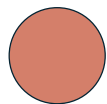
Bleu Méditerranée (#0077B6) – Un bleu profond et vif qui évoque la mer.



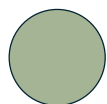
Bleu Ciel (#89CFF0) – Pour rappeler le ciel clair et ensoleillé.



Sable Doré (#F4D19D) – Une couleur sable chaud pour les plages.



Terracotta (#D47F6A) – Pour évoquer les toits en tuiles et les façades provençales.



Vert Olive (#A4B494) – Inspiré des oliviers et de la végétation méditerranéenne.



Blanc Cassé (#F5F5DC) – Pour l'architecture et la lumière éclatante.

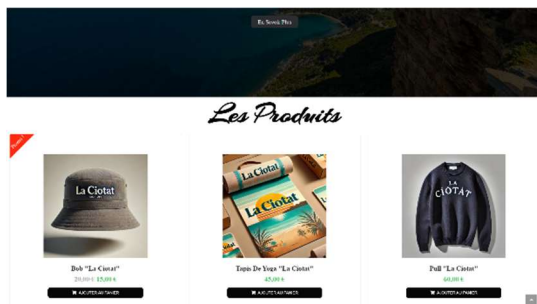
Typographie

Nous avons fait le choix d'utiliser 3 polices d'écritures différentes :

Arizonia : les grands titres du site

Poppins : sous-titres

Times New Roman : les paragraphes



Autre

En ce qui concerne les entreprises mentionnées sur le site, nous avons fait le choix de nous adapter à leur propre charte graphique afin de ne pas dénaturer la palette de couleurs des représentants.

Fonctionnement de l'administration du site

Nous avons mis en place une répartition des tâches : chaque personne a une page attitrée à réaliser ainsi qu'une tâche telle que la charte graphique ou les backups. Pour la mise en commun et le partage des informations, nous avons utilisé Notion pour faciliter l'organisation. Notion permet de partager des fichiers de tout type.

Liste des pages

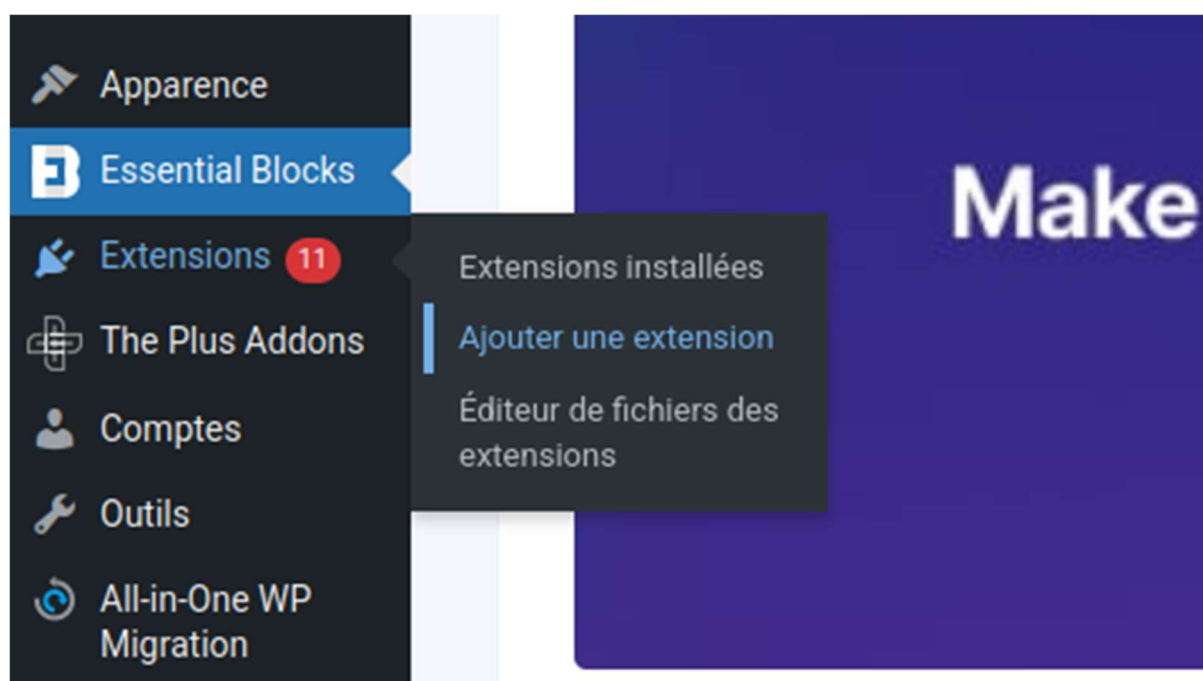
- Page d'accueil / articles : La Ciotat
- Page statistique : La Ciotat en chiffres
- Page des opportunités professionnelles : Travailler à La Ciotat
- Page des espaces vert et de la culture locale : Vivre à La Ciotat
- Page de la boutique : La Boutique
- Page du panier : Le Panier
- Les pages de visualisation de chaque article de la boutique

Liste des articles

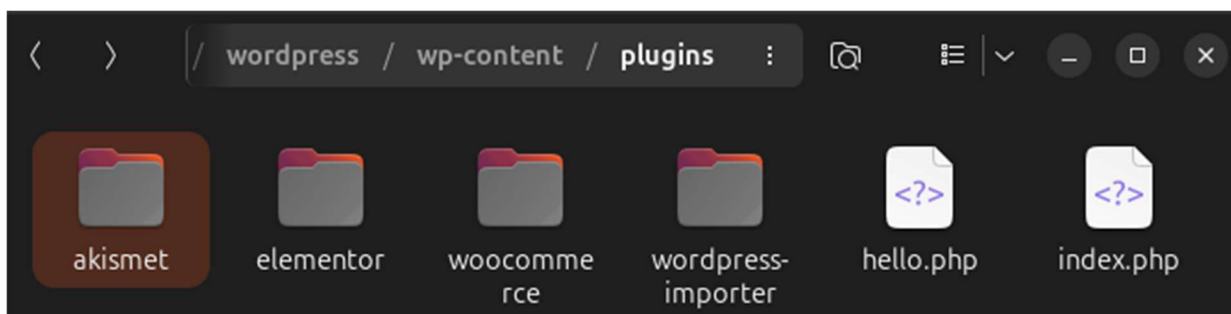
- Article 1 : Les Vœux du maire
- Article 2 : L'exposition de photos
- Article 3 : Le festival de Jazz

Plugins

Pour installer un plugin deux méthodes sont disponibles. La première est la plus simple mais ne fonctionne pas pour toutes les configurations de WordPress. Elle consiste simplement à cliquer sur l'onglet extension, ajouter une extension. Ensuite, recherchez le nom de l'extension voulue et cliquer sur installer. Vérifiez bien que le plugin est actif.



La seconde méthode consiste à rechercher le plugin sur internet et à le télécharger depuis le site du développeur. Vous obtiendrez un fichier.zip que vous dézippez et que vous devrez placer dans le dossier plugins situé dans wordpress/wp-content. Maintenant, retournez dans wordpress et activez le plugin.



Plusieurs plugins peuvent être ajoutés :

- Elementor
- WooLentor
- ShopLentor
- Duplicate Page
- HT Slider
- Ocean extra

GitLab

Un projet GitLab a été créé pour centraliser et partager les fichiers. Il permet aux membres du groupe de récupérer les fichiers et de les partager avec les autres membres.

Différentes commandes sont disponibles pour utiliser GitLab efficacement :

- **git clone** (Cette commande permet de copier un dépôt Git distant sur votre machine locale.)
- **git pull** (Cette commande met à jour votre dépôt local avec les modifications les plus récentes du dépôt distant.)
- **git status** (Cette commande affiche l'état actuel de votre dépôt.)
- **git add nom.txt** (Cette commande ajoute un fichier spécifique (par exemple, nom.txt) à la zone de staging (index), préparant le fichier pour un commit.)

- **git commit -m "ajout dun commentaire"** (Cette commande enregistre les changements dans le dépôt local avec un message descriptif (passé via l'option -m) expliquant les modifications.)
- **git push git pull** (Cette commande envoie les commits locaux (enregistrés avec git commit) vers le dépôt distant, synchronisant ainsi les changements.)

Gestion de projet

Utilisation de GitLab et d'un groupe de discussion sur Notion pour coordonner l'affectation des tâches. Organisation de créneaux horaires de réunion, en présentiel ou à distance, pour faciliter l'entraide entre les membres du groupe.

Interface du projet La Ciotat sur GitLab :

The screenshot shows the GitLab web interface for a project named 'LaCiotat'. On the left is a sidebar with various navigation icons and labels. The main area at the top has a search bar and a notification for 'Auto DevOps'. Below this, the project name 'LaCiotat' is displayed with its repository path 'laciostat / +'. A commit message 'Supprimer README.md' by 'celian brocard' is shown. A table lists the commit history with columns for file name, commit message, and time. On the right, project statistics and a list of actions to improve the project (like adding a README or setting up CI/CD) are visible.

| Name | Last commit | Last update |
|---------------|-------------------------------|-------------|
| Sauvegardes | Add new directory | 1 month ago |
| wordpress | ajout wordpress | 1 month ago |
| wordpress.sql | mise a jour des liens de page | 1 month ago |