



Formation Wordpress

Préparé par: Sarvesh Bhiwa

Table of Contents

Feuille de route.....	4
Introduction.....	5
Installation.....	5
Présentation.....	5
Objectif.....	5
Les méthodes d'installation.....	5
Manuellement.....	5
WP-CLI.....	6
Ressource.....	7
Post type.....	8
Présentation.....	8
Objectif.....	8
Article.....	8
Pages.....	8
Fichiers joints.....	8
Révisions.....	8
Post type custom.....	9
Meta Box.....	9
Récupérer les contenus.....	10
Apparence.....	11
Présentation.....	11
Objectif.....	11
Thème.....	11
Installation.....	12
Sous-thème.....	12
Nouveau thème.....	12
Hiérarchie des modèles.....	13
Widgets.....	13
Où ajouter ces widgets?.....	13
Widgets custom.....	14
Région de widget personnalisé.....	15
Affichage Widgets et les régions Widget.....	16
Extensions.....	17
Présentation.....	17
Objectif.....	17
Guidelines.....	17
Installation.....	18
Extensions (Plugins).....	18
Implementation classique.....	18
Boilerplate – Extensions.....	19
Development custom.....	20
Présentation.....	20
Objectif.....	20
API/HOOK.....	20
Filters.....	20
Actions.....	21
Functions.php.....	21
Ajax.....	21
Wordpress REST API.....	22

Plugins utile.....22

Mini-projet.....23

Ressources.....24

Feuille de route

La formation s'étale sur une durée de 4 semaine. Cela peut durée moins dépend aussi sur votre niveau de connaissance que vous avez déjà sur Wordpress.

Semaine	Sujet	Commentaire
1	Introduction	
1	Installation	
1	Post Type	
2	Apparence	
2	Extensions	
3	Development custom	
3	Wordpress REST API	
4	Mini Projet	
4	Présentation du mini site	

Introduction

Wordpress est le CMS (Content Management System) utilisé sur 35.8 % de tous les site web ainsi représentant en moyen 63.1 % du marché CMS. WordPress est une plate-forme très polyvalente, c'est donc un choix idéal pour les grandes et les petites entreprises, ainsi que pour les particuliers. Les grandes entreprises, les blogs bien connus, les petites entreprises et les particuliers affluent vers WordPress pour créer des sites Web pleinement fonctionnels et esthétiques.

Utilisé auparavant comme une plateforme de blogs. Wordpress se positionne désormais comme une plateforme capable de gérer des grosse fonctionnalités tel que des site e-commerce ou des web apps avec l'arrivé des autres techno supplémentaire sur le marché.

Chez Esokia, la plateforme Wordpress représente une part importante de notre production quotidienne, donc ce document a été préparé dans le but de faciliter la formation de nouvelles recrues chez Esokia sur Wordpress.

Installation

Présentation

Pour pouvoir utiliser le CMS Wordpress il est démise d'installer et configurer correctement le site. Sur la plupart des plateforme hébergements Wordpress est proposé comme de quick install donc sur ce genre de plate-forme vous auriez pas besoin d'aller l'installer manuellement.

Objectif

Au sein d'Esokia, vous aurez besoin de bien comprendre comment installer le site manuellement sur votre poste.

L'objectif de ce chapitre est que vous compreniez bien chaque étape de l'installation afin qu'à la fin de celui-ci vous arriviez à installer un site wordpress rapidement.

Les méthodes d'installation

Manuellement

Méthode commune pour l'installation du site sur votre poste local.

Voici ci-dessous les étapes à suivre :

Étape 1°: Télécharger

- Téléchargez le dernier release wordpress sur votre poste depuis <https://fr.wordpress.org/download/>

- Décompressez le fichier téléchargé dans un dossier sur votre poste

Étape 2° : La base de données (BDD)

- Sur votre phpmyadmin crée une base de données vierge
- Rassurez-vous que votre utilisateur de base mysql ait tous les droits requis pour la manipulation de la BDD

Étape 3° : Configuration

- Dans le dossier de votre site, trouvez le fichier wp-config-sample.php et faites une copie en le renommant wp-config.php
- Dans le fichier wp-config.php, changez les valeurs de variables suivantes :
 - DB_NAME : le nom de la bdd
 - DB_USER : le nom d'utilisateur
 - DB_PASSWORD : le mot de passe de l'utilisateur
 - DB_HOST : normalement "localhost" ou sinon si vous partagez la bdd d'un collègue mettez son IP

Étape 4° : Installation

- Tapez l'adresse de votre poste sur votre navigateur préféré, par ex., <http://localhost>
- Ensuite remplissez le formulaire qui s'affiche pour terminer l'étape d'installation et 'hop vous avez votre site wordpress up and running.

WP-CLI

Méthode idéale pour l'installation de site sur un serveur via SSH ou votre poste local via la ligne de commande.

Prérequis :

- Il faut que la librairie wp-cli soit installée sur votre poste et si c'est pas déjà installé suivez ce tutoriel <https://make.wordpress.org/cli/handbook/guides/installing/>
- Vous devez déjà avoir le host, nom de la bdd, l'utilisateur et son mot de passe

Pour cette partie vous devez utiliser votre terminal de commande et lancer les commandes suivantes:

Étape 1°: Créer le dossier du site

```
mkdir myfirstsite
```

Étape 2° : Télécharger et configurer wordpress

```
cd myfirstsite
```

```
wp core download
```

wp core config --dbhost=le nom ou ip du host --dbname={le nom de la bdd} --dbuser={le nom de l'utilisateur} --dbpass={le password de l'utilisateur}

Étape 3° : Configurer wp-config.php

chmod 644 wp-config.php

wp core install --url={myfirstsite.com} --title={ "My first site" } --admin_name={le nom admin souhaitez} --admin_password:{le password souhaitez}

Étape 4° : Activer upload des fichiers

cd wp-content

mkdir uploads

chgrp web uploads/

chmod 775 uploads/

Ressource

<https://wordpress.org/support/article/how-to-install-wordpress/>

<https://www.youtube.com/watch?v=H8jihECqIgE>

Post type

Présentation

Un post type est tout simplement un type de publication sur Wordpress utilisé pour afficher une catégorie de contenu stocké dans la table wp_posts de la BDD. Par défaut, plusieurs post types sont disponible à l'installation par e.g : Articles, Pages, Fichiers joints, et Révisions

Objectif

Ce chapitre à pour but de vous rendre familier avec les termes techniques au niveau de contenus de votre site wordpress.

Article

L'article est un type de contenu le plus souvent utiliser par les blogs.

Ce type de publication en temps normal contient un titre, le contenu (wysiwyg) et un image mis en avant, qui peut être aussi assigné à des catégories et des étiquettes permettant de filtrer les contenus en front.

<https://www.youtube.com/watch?v=cuFQUA4q8cE>

Pages

Une page est aussi un type de publication disponible par défaut, qui a pour but d'afficher des contenus très spécifique comparé aux posts.

<https://www.youtube.com/watch?v=1QdQ-QCio9A>

Fichiers joints

Les fichiers joints sont un autre type de publication qui contiennent des informations de tout média téléversé sur votre site wordpress.

<https://www.youtube.com/watch?v=rz3p8kBVOK0>

Révisions

Les révisions sont un type de publication qui sauvegarde une historique des autres publications. Donc, si par exemple, pendant que l'admin est en train de modifier un article il appui sur le bouton

publié par accident, l'ancien état de l'article peut-être restaurée. <https://www.youtube.com/watch?v=onSPi4t1ao0>

Post type custom

Autre que de post type mentionné, les clients ont très souvent besoin de post type custom. Ceci peut-être fait via un plugin ou le code suivant

```
function create_posttype() {
    register_post_type(
        'car',
        array(
            'labels' => array(
                'name' => __( 'Cars' ),
                'singular_name' => __( 'Car' )
            ),
            'public' => true,
            'has_archive' => false,
            'rewrite' => array('slug' => 'car'),
        )
    );
}

//On ajoute un add_action pour lié le hook init au callback create_posttype
add_action( 'init', 'create_posttype' );
```

Ou vous pouvez aussi utiliser l'extension <https://wordpress.org/plugins/custom-post-type-ui/>

Meta Box

Tous le type de publication créé ou qui est disponible sur wordpress pour ajouter des contenu, n'ont que des metabox basique comme Titre, Contenus, Image mis en avant, catégorie ou tags.

Si on veut ajouter des metaboxes (champs) customisé, voici un bout de code qui vous sera très utile. Pour ajouter le meta box au post type News qu'on vient de créer

```
Function custom_add_box()
{
    add_meta_box(
        'custom_meta_box', //ID unique
        'Custom meta box', //Titre
        'custom_box_html', //Callback pour affichage de contenu
        'news', //Post type
    );
}
```

```
}  
add_action( 'add_meta_boxes', 'custom_add_box');
```

Pour afficher le contenu de meta box Custom meta box qu'on a créé.

```
Function custom_box_html($post){  
?>  
<label for="wporg_field">Description for this field</label>  
<select name="wporg_field" id="wporg_field" class="postbox">  
    <option value="">Select something...</option>  
    <option value="something">Something</option>  
    <option value="else">Else</option>  
</select>  
<?php  
}
```

Pour récupérer la valeur de meta box

```
$value = get_post_meta($post->ID, 'custom_meta_box', true);
```

OU

Vous pouvez utiliser ce plugin pour créer vos meta box custom:

<https://wordpress.org/plugins/advanced-custom-fields/>

Récupérer les contenus

ARTICLES :

get_posts: https://developer.wordpress.org/reference/functions/get_posts/

WP_Query: https://developer.wordpress.org/reference/classes/wp_query/

PAGES :

get_pages: https://developer.wordpress.org/reference/functions/get_pages/

WP_Query: https://developer.wordpress.org/reference/classes/wp_query/#highlighter_654020

CATEGORIES :

https://developer.wordpress.org/reference/functions/get_categories/

ÉTIQUETTES :

https://developer.wordpress.org/reference/functions/get_tags/

Apparence

Présentation

L'affichage de votre site wordpress est entièrement configurable, donc vous pouvez le modifier à votre goût et surtout à la demande du client. C'est là que vient en jeu la partie Apparence du site.

Dans votre back-office, dans la barre menu latérale gauche vous pouvez retrouver le menu. Les trois items les plus populaires de ce menu sont :

- Thèmes
- Widgets
- Menus

Ces items aident à alimenter votre site en termes d'affichage

Objectif

Le but de ce chapitre de votre formation, vous permettra d'apprendre ce qu'est un thème, des widgets et le menu, comment les configurer et utiliser.

Thème

Le thème de votre site wordpress est celui qui change le design donc votre site est affiché en front. Le thème dans un langage plus simple, les thèmes sont des collections de différents fichiers qui fonctionnent ensemble pour créer ce que vous voyez et ainsi définir le comportement de votre site. Sans un thème le visiteur de votre site aura une page blanche car wordpress a besoin d'un thème pour avoir un rendu.

Pour pallier à ce problème Wordpress inclut par défaut 3 thèmes de la communauté gratuitement sur son installation.

La dernière version vient avec ces trois thèmes :

- twenty nineteen
- twenty seventeen
- twenty twenty

En plus de thèmes disponibles gratuitement, il y a aussi des thèmes créés par des professionnels qui peuvent être achetés.

Installation

Étape 1°: Télécharger le thème

- Télécharger votre thème
- Décompresser le fichier dans le dossier wp-content/themes de votre site

Note: *Passer à l'étape 2 si vous utilisez déjà le thème qui vient avec votre installation de wordpress*

Étape 2° : Activation

- Cliquer sur « Apparence » dans votre menu de back-end
- Ensuite sélectionnez « Thèmes »
- Trouvez votre thème et survolez l'illustration du thème pour cliquer sur «Activer »

Sous-thème

Tous les thèmes télécharger sont sujet à de mis-à-jour à un moment donné, il est donc impératif d'avoir un sous-thème avant de manipuler les fichiers d'un thème.

C'est un étape recommander pour de raison suivant :

- Si vous modifiez un thème directement et il est mis à jour, vos modifications peuvent être perdues. En utilisant un thème enfant vous assurer que vos modifications sont conservées.
- L'utilisation d'un thème enfant peut accélérer le temps de développement.
- Lorsque vous travaillez sur un thème enfant et que vous oubliez de coder quelque chose, la fonctionnalité du thème parent est toujours disponible comme option de secours.

<https://www.youtube.com/watch?v=cbU12Tw7Q4U>

Nouveau thème

Si le client nous a pas donné un thème à utiliser en va devoir créer votre propre thème à zéro, le tutoriel suivant vous aiderez à créer votre thème.

<https://www.youtube.com/watch?v=8QOZPdyR-Q8>

Hiérarchie des modèles

Les template qui viennent avec un thème sont modulables et réutilisables. Ils sont utilisés pour générer des pages web.

Certains modèles tels que header ou footer sont utilisés partout sur le site, et d'autres sont utilisés sous des conditions spécifiques, exemple une zone de connexion.

https://docs.google.com/document/d/1bL75DhuAQCP72f9SWLR9CahR8d0qyrIieJ0dokbxx8/edit?usp=drive_web&authuser=0

<https://developer.wordpress.org/themes/basics/template-hierarchy/>

<https://www.youtube.com/watch?v=Pb4oQsWdmL0>

Widgets

Les widgets sont des blocs de contenu que vous pouvez ajouter aux barres latérales, pied du site et autres zones de votre site.

Chaque widget peut ajouter une fonctionnalité ou une fonction à votre site, sans avoir à écrire de code.

Les widgets disponibles à ajouter à votre site sont les menus, les listes de publications populaires, les calendriers, les bannières publicitaires, les icônes sociales, etc.

Où ajouter ces widgets?

Tout dépend de votre thème WordPress.

De nombreux thèmes ont des zones de widget de barre latérale et de pied de page. Certains vous permettent également de placer des widgets dans l'en-tête, la page d'accueil ou d'autres zones de votre site.

Pour ajouter un widget suivez les étapes suivantes:

1. Parcourez le menu de votre back office Apparence » Widgets
2. Une fois sur cette page, vous verrez les widgets à gauche et les zones de widget à droite
3. Glisser le widget voulu dans le zone de votre choix pour faire apparaître le widget en front

Widgets custom

Les étapes ci-dessous permettront de développer complètement un widget personnalisé.

Étape 1° : Pour créer un widget, vous avez seulement besoin d'étendre la classe de WP_Widget standard et certaines de ses fonctions.

```
class foo_Widget extends WP_Widget {  
    ...votre implementation  
}
```

Étape 2° : Définit le nom des widgets, etc.

```
function __construct() {  
    parent::__construct(  
        'foo_widget', → l'id de votre widget  
        __( 'Titre de widget', 'text_domain' ), → le nom du widget  
        array( 'description' => __( 'A Foo Widget', 'text_domain' ), ) → Un array d'autre  
        valeur Args  
    );  
}
```

Étape 3° : Afficher les contenus du widget

```
public function widget( $args, $instance ) {  
    echo $args['before_widget']; → des wrappers prédéfini  
    if ( ! empty( $instance['title'] ) ) {  
        echo $args['before_title'] . apply_filters( 'widget_title', $instance['title'] ).  
        $args['after_title'];  
    }  
    echo __( 'Hello, World!', 'text_domain' );  
    echo $args['after_widget']; → des wrappers cloture prédéfini  
}
```

Étape 4° : Le formulaire back-office du widget

```
public function form( $instance ) {  
    $title = ! empty( $instance['title'] ) ? $instance['title'] : __( 'Titre', 'text_domain' );  
    ?>  
    <p>  
    <label for="<?php echo $this->get_field_id( 'title' ); ?>"><?php _e( 'Title:' ); ?></label>  
    <input class="widefat" id="<?php echo $this->get_field_id( 'title' ); ?>" name="<?php  
    echo $this->get_field_name( 'title' ); ?>" type="text" value="<?php echo  
    esc_attr( $title ); ?>">  
    </p> <?php
```

```
}
```

Étape 5° : Sauvegarder le valeur renseignée dans le formulaire

```
public function update( $new_instance, $old_instance ) {  
    $instance = array();  
    $instance['title'] = ( ! empty( $new_instance['title'] ) ) ?  
strip_tags( $new_instance['title'] ) : '';  
    return $instance;  
}
```

Étape 6°: Enregistrer le widget dans `widgets_init`. Le code est a mettre en dehors du class.

```
function register_foo_widget() {  
    register_widget( 'Foo_Widget' );  
}  
add_action( 'widgets_init', 'register_foo_widget' );
```

Le widget personnalisé que vous créez peut être ajouté dans des régions prédéfinies par wordpress mais si vous avez à mettre le widget dans une autre région qui n'est pas définie, il vous faut créer des régions personnalisées.

Région de widget personnalisé

La région personnalisé peut être utilisée pour de widget prédéfini ou personnalisé. Pour créer la région il vous faut d'abord votre sous-thème et si vous l'avez pas suivez les instruction de section 1.3.

Sous le répertoire du sous-thème vous devrez initialiser votre fonction dans le fichier `function.php`.

```
function myfoo_widgets_init() {  
    register_sidebar( array(  
        'name'          => 'foo_sidebar_1',  
        'before_widget' => '<div>', → ouverture d'un wrapper pour le widget  
        'after_widget'  => '</div>',  
        'before_title'  => '<h2 class="rounded">', → ouverture de tag pour le titre  
        'after_title'   => '</h2>',  
    ));  
}  
add_action( 'widgets_init', 'myfoo_widgets_init' );
```

Affichage Widgets et les régions Widget

Pour afficher les widgets dans un template ou une page avec le filtre PHP activé:

```
<?php if ( dynamic_sidebar('foo_sidebar_1') ) : else : endif; ?>
```

Pour vérifier si la région est actif avant la rendu en utilisant la fonction ***is_active_sidebar***:

is_active_sidebar est une fonction de base de wordpress, qui a comme role de verifié si un widget particulier exist ou pas.

```
<?php if ( is_active_sidebar( 'foo-widget' ) ) : ?>
    <div id="custom-sidebar" class="custom-sidebar widget-area" role="complementary">
        <?php dynamic_sidebar( 'foo_sidebar_1' ); ?>
    </div><!-- #custom-sidebar -->
<?php endif; ?>
```


Extensions

Présentation

Les extensions comme dit le nom sont utilisé pour étendre les fonctionnalités déjà offert par le core wordpress ou pour créer des nouvelles fonctionnalités.

À Esokia, sur la majorité de projet on est appelé à répondre à des demandes très spécifique de la part des clients donc l'utilisation des extensions à part de ce qu'offre wordpress est probable.

Objectif

L'objectif de ce chapitre est pour élaborée les guidelines pour choisir le bon extension et aussi de vous introduire aux extensions, l'installation et l'utilisation.

Guidelines

Afin de sélectionner l'extension appropriées, il faut que vous suivez quelques guidelines très simple:

1. Avant plan :

Noté tous les fonctionnalités dont vous aurez besoin par ordre d'importance. Car il ce peut que pour une fonctionnalité vous allez avoir à installer plus d'un extension.

2. Chercher l'extension :

Rechercher l'extension ici <https://wordpress.org/plugins/> , vous allez peut-être pas avoir le nom exacte mais utiliser un mot clé, par exemple si vous cherchez un extension permettant de gérer le SEO, il suffit de taper « seo » comme mot de recherche.

3. Comparer les extensions :

Quand vous recherchez des extensions vous allez peut-être tomber sur une listes plugins qui auront les même fonctionnalités que vous avez besoin. Dans de tel cas, suivez ces étapes pour établir l'extension approprié :

- Critique de l'extension :

Vérifier pour chaque extension quels sont les critique fait par les utilisateurs

- L'évaluation :

Vérifier l'évaluation faites par des utilisateurs de l'extension

- Support :

Il faut aussi vérifier si l'extension reçoit toujours le support des développeurs qui l'ont crée et aussi combien de bug il-y-a eu après le dernière release

- Compatibilité :
Est-ce que l'extension est compatible au dernier mis-à-jour de wordpress
4. Tester l'extension :
Finalement avant de finaliser l'utilisation de l'extension sélectionné, il faut tester le plugin au niveau fonctionnalité et aussi niveau performance

Installation

Étape 1° : Télécharger l'extension

- Télécharger l'extension depuis <https://wordpress.org/plugins/>
- Décompressez le fichier dans le dossier wp-content de votre site

Étape 2° : Activation

- Cliquez sur Extensions / Plugins
- Chercher le plugin et cliquez sur « Activer »

A ce niveau vous devriez déjà être capable de créer un site wordpress par vous même !

Extensions (Plugins)

Comme mentionné ci-dessus qu'il-y-a environ 54,000 extensions mises à disposition, mais toutefois dans beaucoup de cas on est appelé à créer une extension custom pour des besoins très spécifiques.

Implementation classique

La création d'un plugin personnalisé se fait en tenant compte des différentes étapes requises.

Étape 1° : Créer dossier

Dans wp-content/plugin il faut créer le dossier my_plugin, puis à l'intérieur créer les fichiers my_plugin.php et readme.txt.

Étape 2° : Info header

Dans le fichier my_plugin.php il est impérative des info concernant l'extension.

/*

Plugin Name : My Plugin

Plugin URI: <https://www.this-is-my-plugin.com>

Description: Custom plugin to implement custom functionalities

Version: 0.1

Author: Myself

Author URI: <https://www.myself.com/>

**/*

Étape 3° : Aspect de sécurité

Afin que personne n'est peut pas accéder au fichier my_plugin.php directement en utilise les codes suivant juste après le headers du fichier pour neutraliser tous les accès direct.

`defined('ABSPATH') or die('Plugin file cannot be accessed directly.');`

Étape 4° : L'implémentation

Ci-dessous est la structure d'implémentation du plugin

`if (! class_exists('MyPlugin')) { //On verifie si la classe exist déjà`

`class MyPlugin`

`{`

`public function __construct()`

`{`

`/* */`

`}`

`}`

`//Initialisé`

`new MyPlugin;`

`}`

Boilerplate – Extensions

À la base sur Wordpress on peut coder en free-style, pas d'obligation dans la façon qu'on peut coder (coding standard) mais toutefois chez Esokia, on veille toujours à ce que nos développeurs respectent certains standard.

Pour cette raison, nous utilisons Boilerplate Plugin comme un backbone très solide, simple et facile à comprendre pour générer des extensions custom. Cela permet aussi à nos développeurs

Comment générer un plugin boilerplate?

1. Sur le lien <https://wppb.me/>
2. Remplissez le formulaire:
 - a) Plugin name: My Plugin
 - b) Plugin slug: my-plugin
 - c) Plugin Uri: <https://www.clientwebsite.com>
 - d) Author name: Client name
 - e) Author Email: client@clientwebsite.com
 - f) Author Uri: <https://www.clientwebsite.com>
3. Cliquez sur "Build Plugin"

Development custom

Présentation

Dans les chapitres précédent on a vu comment est-ce qu'on peut créer un site wordpress avec tous les ressources disponible en ligne a être télécharger et utiliser (plug & play). Cependant on se retrouve souvent dans ce circonstance où les thèmes, widgets et extensions disponible ne réponds pas vraiment au besoin du client donc c'est là que nos développeur rentre en jeu.

Objectif

Ce chapitre sera spécifiquement consacré a vous expliquez comment crée un thème, plugin, widget et fonctionnalités custom.

API/HOOK

Pour étendre les fonctionnalités du core wordpress, on aura besoin de hook/api wordpress qui est divisé en deux section Filter et Action.

Filters

Le hook Filter est offert par Wordpress pour permettre aux extensions (Plugins) de modifier plusieurs types de valeur interne lors de l'exécution.

Liste des filter commun existant : https://codex.wordpress.org/Plugin_API/Filter_Reference

Add_filter

add_filter est utilisé pour lier un callback à un filter, par e.g:

```
function add_class_body($classes){
    if ($post->post_name == "babyboo"){
        $classes[] = "class-babyboo";
    }
    return $classes;
}
add_filter( 'body_class', 'add_class_body');
```

Dans cet exemple on a ajouté un filter qui ajoute une classe au balise body si l'article en cours a le slug 'babyboo'.

Support:

<https://developer.wordpress.org/plugins/hooks/filters/>

Actions

Les actions fournissent un moyen d'exécuter une fonction à un moment précis de l'exécution de WordPress Core, des plugins et des thèmes.

Liste des Actions comment existant : https://codex.wordpress.org/Plugin_API/Action_Reference

Add_action

`add_action` est utilisé pour lier un function callback à une action, par exemple :

```
function update_custom_meta($post_ID, $post, $update){  
    update_post_meta($post->ID, 'pays', 'Maurice');  
}  
add_action( 'save_post', 'update_custom_meta', 10, 2)
```

Dans cet exemple, quand l'administrateur du site sauvegarde un article, une valeur meta avec le nom pays est aussi sauvegardée.

Support

https://developer.wordpress.org/reference/functions/add_action/

<https://developer.wordpress.org/plugins/hooks/actions/>

Functions.php

Pour modifier les comportements par défaut de Wordpress le fichier `functions.php` peut être utiliser. Ce fichier se trouve dans la racine de votre thème.

Ce fichier se comporte comme un extension utilisée pour ajouter des fonctionnalités au site en utilisant la langue PHP.

Les deux exemples de **`add_action`** et **`add_filter`** et leur callback peuvent-être implémentés dans ce fichier.

Ajax

Donc, si vous souhaitez qu'il se déclenche sur le front-end pour les visiteurs et les utilisateurs connectés, vous pouvez le faire comme l'explique ce document :

https://codex.wordpress.org/AJAX_in_Plugins

Wordpress REST API

L'API WordPress REST fournit une interface permettant aux applications d'interagir avec votre site WordPress en envoyant et en recevant des données sous forme d'objets JSON (JavaScript Object Notation).

Il est le fondement de l'éditeur de blocs WordPress et peut également permettre à votre thème, plugin ou application personnalisée de présenter de nouvelles interfaces puissantes pour gérer et publier le contenu de votre site.

<https://developer.wordpress.org/rest-api/reference/>

Plugins utile

Parmi tous ces plugins disponibles, chez Esokia on utilise certains d'entre eux qui nous aident au quotidien, voici la liste de plugin et leur urls.

Custom Post Type UI : Pour créer de post type custom

<https://wordpress.org/plugins/custom-post-type-ui/>

Advanced custom fields : Pour créer de meta box custom

<https://wordpress.org/plugins/advanced-custom-fields/>

Polylang : Pour la gestion de multilangue

<https://wordpress.org/plugins/polylang/>

Yoast SEO : Pour gérer la partie SEO

<https://wordpress.org/plugins/wordpress-seo/>

Contact form 7 : Pour générer des formulaire de contact et gérer l'envoi de mail

<https://wordpress.org/plugins/contact-form-7/>

WP Mail SMTP : Pour gérer les envois de mail

<https://wordpress.org/plugins/wp-mail-smtp/>

Elementor : Extension largement utiliser chez esokia pour construire des pages

<https://wordpress.org/plugins/elementor/>

Mini-projet

Après avoir terminer chaque étape des formations vous devez développer un site sous wordpress avec les specifications suivants:

Thème: Learning Management System

Description: Pour ce projet vous devez créer trois interface suivants:

1. Un interface pour le gens non-connecté au site, qui auront droit visuel sur:
 - i. Une page d'accueil (Le menu, un slider des images campus et étudiant, un bloc de trois ou plus des cours en offre, un bloc avec une formulaire de contact)
 - ii. Une page avec tous les cours offert
Au clic sur une des cours, une page dédiée est afficher, avec tous les details du cour et un bouton "Je participe". Quand l'utilisateur clique dessus son nom est attaché au cour sur le dashboard du formateur
 - iii. Une page à propos avec des texte et des images
 - iv. Une page login avec un toggle formateur et étudiant
2. Un interface pour les formateurs
Les utilisateurs ayant le droit de formateur auront accès aux pages mentionné en haut et aussi cette section qui lui permettra de:
 - i. crée des cours
 - a) Titre
 - b) Catalogue
 - c) Durée
 - d) Lieux
 - e) Les supports: Visible au étudiant connecté en front
 - ii. Voir ses cours et le nombre des étudiants associé à son cours, au clique sur le nombre des étudiants un popup contenant les nom de chaque etudiant est affiché chargé en ajax.
 - iii. son profil: pour changer ses details et son mot de passe
3. Un interface pour les étudiants
Les utilisateurs ayant le droit d'étudiant auront accès aux pages mentionné en haut et aussi cette section qui lui permettra de:
 - i. Le cours ils se sont abonné à et les supports disponible à télécharger.
 - ii. son profil: pour changer ses details et son mot de passe

Ressources

<https://codex.wordpress.org/>
<https://wordpress.org/support/article/how-to-install-wordpress/>
<https://www.youtube.com/watch?v=H8jihECqIgE&list=PLC5E59DD6D84D34DC&index=2>
<https://www.youtube.com/watch?v=cuFQUA4q8cE&list=PLC5E59DD6D84D34DC&index=8>
<https://www.youtube.com/watch?v=1QdQ-QCio9A&list=PLC5E59DD6D84D34DC&index=10>
<https://www.youtube.com/watch?v=rz3p8kBVOK0>
<https://www.youtube.com/watch?v=onSPi4t1ao0>
<https://www.youtube.com/watch?v=iMmRQgF3XkM&list=PLC5E59DD6D84D34DC&index=16>
<https://www.youtube.com/watch?v=cbU12Tw7Q4U&list=PLLnPHn493BHGACfv4rC29kJamYMTw34D9&index=1>
<https://www.youtube.com/watch?v=8QOZPdyR-Q8>
<https://developer.wordpress.org/themes/basics/template-hierarchy/>
https://codex.wordpress.org/Plugin_API/Filter_Reference
https://developer.wordpress.org/reference/functions/add_filter/
<https://developer.wordpress.org/plugins/hooks/filters/>
https://codex.wordpress.org/Plugin_API/Action_Reference
https://developer.wordpress.org/reference/functions/add_action/
<https://www.youtube.com/watch?v=ftPB4qEKlvo&list=PLC5E59DD6D84D34DC>
<https://www.youtube.com/watch?v=cbU12Tw7Q4U&list=PLLnPHn493BHGACfv4rC29kJamYMTw34D9>
<https://www.youtube.com/watch?v=8OBfr46Y0cQ&list=PLpcSpRrAaOaqMA4RdhSnnNcaqOVpX7qi5>
<https://www.youtube.com/watch?v=Mq52jXX6hbI&list=PLT9miexWCpPUQkQwL-COHmo0Jd0qxLjTn>
https://www.youtube.com/watch?v=chwoN-vgmTo&list=PLT9miexWCpPUwGIFk_yIH2cdkDgJDvRBn
<https://developer.wordpress.org/plugins/plugin-basics/best-practices/>
<https://shellcreeper.com/how-to-secure-wordpress-ajax-using-nonce/>
<https://www.wpbeginner.com/>
<https://developer.wordpress.org>