

Créer des champs personnalisés

PHP, Wordpress

Web Compétences : 3

Inutile de créer des types de publications personnalisés sans y ajouter des champs personnalisés. Là aussi, Wordpress est limité, mais un plugin très répandu, le fameux ACF, va pouvoir nous sortir de l'impasse et créer des champs personnalisés à l'instar de ma belle image de fond

Publié dimanche 22 septembre 2019, 11h49
Modifié lundi 26 août 2024, 10h06



By Olivier Paudex

Introduction

Les champs personnels sont indissociables des types personnels. Un champ avec une donnée personnelle n'a pas de sens en dehors d'un type personnel et vice-versa. La réflexion autour des champs personnels a commencé bien en amont de ce chapitre, c'est peut-être l'une des premières tâches à laquelle je me suis attelé. En effet, lors des cogitations sur les différentes informations que je voulais afficher sur mon site, la tendance s'est rapidement divisée en deux groupes distincts, d'un côté, le type, et de l'autre, les différentes informations liées au type. Sans le savoir vraiment, j'avais découvert par moi même qu'il faudrait créer des champs personnalisés.

Les champs personnalisés

Une fois les types personnalisés introduit dans WordPress à l'aide du plugin CPT UI, la question se pose tout naturellement pour les champs personnalisés. La première étape à réaliser est très certainement d'en

faire une liste. Je vous donne en vrac ceux de mon type **"Voyages"**.

- Titre
- Description (extrait)
- Image à la une (image de fond)
- Classification (type de voyage)
- Affichage des plus belles images en début d'articles
- Le pays
- Le mois et l'année

Suivant cette idée, les possibilités de WordPress sont vite parcourues.

- Le titre, la description et l'image de fond sont déjà intégrés à WordPress.
- La classification, qui porte le nom savant de taxonomie dans WordPress est aussi intégré, mais uniquement pour le type "**Articles"**. Un type d'article à deux taxonomie, **"la catégorie" et "l'étiquette"**.
- L'affichage des plus belles images est un option qui n'existe pas, ainsi que les données sur le pays, le mois et l'année du voyage. Il fallait donc recourir à un plugin supplémentaire, le quatrième, après Elementor, Elementor Pro et CPT UI. **Le choix s'est porté sur ACF**.



Le plugin ACF

ACF et les champs personnalisés

Créer un champ personnel avec ACF est simple. On commence par créer un groupe de champs, puis les champs les uns après les autres. Pour mon type **"Voyages"**, j'ai complété les cinq données ci-dessous, ainsi que les valeurs min et max pour l'année du voyage. Il y a d'autres champs que je n'ai pas utilisés, libre à vous de les essayer.

Titre du champ	Nom du champ	Type de champ	Instructions	Requis
Pays	travel_country	Texte	Entrez le pays	Oui
Mois	travel_month	Liste déroulante	Sélectionnez le mois du voyage	Oui
Année	travel_year	Nombre	Entrez l'année du voyage	Oui

Les champs tels que demandés dans ACF

La liste déroulante

Pour le champ **"Mois"**, il faut créer une liste déroulante. Manuellement, on peut la créer dans**ACF** en entrant dans la zone **"Choix"**, les douze mois de l'année en respectant la syntaxe suivante **"janvier : Janvier"**. Le premier étant une étiquette, le deuxième étant la valeur.

La paresse étant un vilain défaut, la code PHP ci-dessous (en anglais), remplira automatiquement **une liste déroulante ACF** avec les mois en anglais pour peu que celle-ci porte le nom de**"travel_month"** . Libre à vous de l'utiliser ou pas, mais je trouve ça bien pratique, surtout si on multiplie les listes déroulantes qui comportent les mois de l'année.

```
// Populate ACF selector form field with Travel months
function populate_travels_acf_dropdown_list ($field) {
  // Reset choices
  $field['choices'] = array();
  // Add the placeholder item
  $field['choices'] = array(
    'january'     =>  'January',
```

```
    'february'    =>  'February',
    'march'       =>  'March',
    'april'       =>  'April',
    'may'         =>  'May',
    'june'        =>  'June',
    'july'        =>  'July',
    'august'      =>  'August',
    'september'   =>  'September',
    'october'     =>  'October',
    'november'    =>  'November',
    'december'    =>  'December'
  );

  return $field;
}add_filter ('acf/load_field/name=travel_month', 'populate_travels_acf_dropdown_list',10,1);
```

Affichage des plus belles images

Si vous avez bien suivi mon déroulement, vous aurez remarqué qu'il n'y a pas mention du champ **"Affichage des plus belles images"**. Ce n'est pas un oubli, mais une option qui n'est pas gratuite. En effet, le champ de type galerie n'est fourni qu'avec la version **ACF Pro**.

Titre du champ	Nom du champ	Type de champ	Instructions	Requis
Images en galerie	travel_gallery_images	Galerie	Téléversez les images que vous souhaitez voir dans la galerie en haut de la publication	Non

Le champ galerie n'est disponible que dans la version Pro

> **ACF Pro** a d'autres options bien utiles, comme les champs qui se répètent, un éditeur de blocs personnalisés pour Gutenberg, etc...Un outil également bien agréable est la possibilité d'importer et d'exporter ses champs.

Lier les champs personnalisés avec le type correspondant

La prochaine étape est de lier ces nouveaux champs avec le type **"Voyages"**. Ceci se fait tout simplement en modifiant une règle située juste en dessous des champs. Sélectionnez "Type de publication", puis "est égal à" et enfin le type **"Voyage".**

La classification (Taxonomie)

Si le type **Articles** possède deux classifications par défaut, la catégorie et l'étiquette, les types personnalisés n'en possèdent aucun. Il faut les créer. Cette option ne se trouve pas dans le plugin **ACF**, mais dans le **CPT UI**. Donc rendez-vous à nouveau sur le panneau de configuration du **CPT UI**, mais cette fois-ci, dans la section taxonomie.

L'ajout d'une taxonomie **"Types de voyages"** se fait similairement à un nouveau type personnalisé.

- Entrez un slug pour votre taxonomie
- Entrez le libellé pour le pluriel
- Entrez le libellé pour le singulier
- Cochez le type auquel rattaché la taxonomie

Maintenant vous pouvez ajouter des classifications dans le type **"Voyages"**. Pour l'exemple, j'y ai ajouté **"Loisirs", "Culturel" et "Affaire"**.

L'interface de publications

Une fois les nouvelles classifications et les champs personnalisés ajoutés, une nouvelle publication peut être créée. Les nouveaux champs s'affichent en dessous de la zone de saisie de la publication.

Comme WordPress intègre l'extrait, l'image mis en avant et la classification, ceux-ci s'affichent dans la colonne à droite. Ce n'est pas très joli et on va corriger ceci.

- Retournez dans **CPT UI** et décochez les options.
- Retournez dans **ACF**, et ajoutez les champs image, extrait et classification, comme ceci.

Titre du champ	Nom du champ	Type de champ	Instructions	Requis
Extrait	travel_description	Zone de texte	Entrez une courte description de votre voyage	Non
Classification	travel_taxonomy	Taxonomie / case à cocher	Sélectionnez un type de voyage	Non
Image	travel_image	Image	Sélectionnez une image	Oui

Les champs ACF pour remplacer ceux de WordPress

Cette fois-ci, tous les champs personnalisés, y compris l'image mis en avant, la classification et la description (extrait) sont placés en dessous de la zone de saisie de la publication.**Seul la taxonomie apparaît à double. Mais rassurez-vous, les deux zones se synchronisent automatiquement**

La prochaine étape logique est d'afficher la page d'archives avec des publications contenant des champs personnalisés. Les surprises n'étaient pas terminées. Elementor, avec son widget **"Archives de publications"**, ne permet pas d'afficher les champs personnalisés. Si vous avez suivi ce blog jusqu'ici, peut-être vous souvenez vous qu'un wireframe avait été dessiné, représentant un habillage complètement modifié. Cette idée de design était évoqué au début de ce projet, alors qu'Elementor Pro n'avait pas encore été mentionné, ni adopté comme constructeur de pages. La consultation d'autres sites, pour trouver de l'inspiration, confirma qu'il était possible de modifier **l'habillage de l'archive**. Et la solution s'appelle **Elementor Custom Skin**, sujet du prochain chapitre.