

WordPress : ACF, hook sur les champs

Les **hooks** (ou filtres) d'Advanced Custom Fields (ACF) vous permettent de modifier les valeurs des champs ACF avant qu'elles ne soient sauvegardées ou après leur récupération depuis la base de données.

Ces hooks vous offrent une flexibilité supplémentaire pour manipuler les données des champs selon vos besoins spécifiques.

[Documentation officielle](#)

Injecter des valeurs dans une liste déroulante

Pour un champ **select** avec plusieurs choix, vous pouvez utiliser les hooks **acf/load_field** et **acf/update_value** pour manipuler les valeurs des champs avant leur affichage et leur sauvegarde.

Voici comment procéder :

Filtre acf/load_field : ce filtre vous permet de modifier les choix disponibles dans un champ select avant qu'ils ne soient affichés dans l'interface d'administration.

```
add_filter('acf/load_field/name=mon_champ', 'my_acf_load_field');
```

```
function my_acf_load_field($field) {  
    // remplacer les choix existants par vos propres choix  
    $field['choices'] = array(  
        'option1' => 'Option 1',  
        'option2' => 'Option 2',  
        'option3' => 'Option 3',  
        // ajoutez d'autres choix ici  
    );  
  
    // retourner le champ modifié  
    return $field;  
}
```

Remarque : vous pouvez très bien récupérer des valeurs à partir d'une requête personnalisée **WP Query**.

Filtre acf/update_value : ce filtre vous permet de manipuler la valeur du champ select avant qu'elle ne soit sauvegardée dans la base de données lors de la publication ou de la mise à jour d'un article.

```
add_filter('acf/update_value/name=mon_champ', 'my_acf_update_value', 10, 3);
```

```
function my_acf_update_value($value, $post_id, $field) {  
    // faire des modifications sur la valeur avant de la sauvegarder  
    $modified_value = 'Valeur modifiée : ' . $value;
```

```

    // retourner la valeur modifiée pour la sauvegarde
    return $modified_value;
}

```

Filtrer les résultats d'un champ relation

Pour un champ relation, vous pouvez utiliser les hooks **acf/load_field** et **acf/update_value** de la même manière que pour le champ select.

Cependant, avec le champ relation, vous aurez besoin d'utiliser ces hooks de manière légèrement différente, car les valeurs sont stockées sous forme d'IDs de publications.

Voici comment utiliser les hooks pour le champ relation :

Filtre acf/load_field : ce filtre vous permet de modifier les choix disponibles dans le champ relation avant qu'ils ne soient affichés dans l'interface d'administration.

```
add_filter('acf/load_field/name=mon_champ_relation', 'my_acf_load_field');
```

```

function my_acf_load_field($field) {
    // remplacer les choix existants par vos propres choix
    // exemple : obtenez les 5 derniers articles de type 'post'
    $args = array(
        'post_type'      => 'post',
        'posts_per_page' => 5,
    );
    $query = new WP_Query($args);

    if ($query->have_posts()) {
        while ($query->have_posts()) {
            $query->the_post();
            $field['choices'][get_the_ID()] = get_the_title();
        }
    }

    // remettre en place la requête d'origine pour les autres processus ACF
    wp_reset_query();

    // retourner le champ modifié
    return $field;
}

```

Filtre acf/fields/relationship/query/ : vous pouvez également utiliser ce filtre pour personnaliser la requête qui récupère les éléments affichés dans le champ relation.

Ce filtre vous permet de filtrer les résultats avant qu'ils ne soient affichés dans la liste de sélection du champ relation.

```
add_filter('acf/fields/relationship/query/name=mon_champ_relation',
'my_acf_relationship_query', 10, 3);
```

```
function my_acf_relationship_query($args, $field, $post_id) {
```

```
// modifier la requête pour le champ relation

// exemple : récupérer uniquement les articles de type 'post' avec un
certain mot-clé dans le titre
$args['post_type'] = 'post';
$args['meta_query'] = array(
    array(
        'key'      => 'meta_key_name', // remplacez par le nom de la clé de
métadonnée souhaitée
        'value'    => 'valeur_cible', // remplacez par la valeur de la clé de
métadonnée souhaitée
        'compare' => '=',              // opérateur de comparaison (par exemple,
'=', '>', '<', 'LIKE', etc.)
    ),
);

// retourner les arguments modifiés pour la requête
return $args;
}
```

Remarque : un filtre **acf/fields/post_object/query/** existe pareillement pour le champ **Objet de la publication**.

Filtre acf/update_value : ce filtre vous permet de manipuler les valeurs du champ relation avant qu'elles ne soient sauvegardées dans la base de données lors de la publication ou de la mise à jour d'un article.

```
add_filter('acf/update_value/name=mon_champ_relation', 'my_acf_update_value',
10, 3);

function my_acf_update_value($value, $post_id, $field) {
    // faire des modifications sur les valeurs avant de les sauvegarder
    // dns le cas du champ relation, $value est un tableau d'IDs d'articles

    // exemple : ajouter "123" à la liste des relations
    $value[] = 123;

    // retourner les valeurs modifiées pour la sauvegarde
    return $value;
}
```

Dans cet exemple, nous avons ajouté l'ID "123" à la liste des relations avant de sauvegarder les valeurs.

Modifier la valeur affichée

Le hook **acf/fields/relationship/result/** vous permet de manipuler les résultats récupérés pour un champ relation avant qu'ils ne soient affichés dans la liste de sélection du champ.

Ce filtre vous donne la possibilité de personnaliser la façon dont les résultats sont présentés aux utilisateurs dans le champ relation.

```
add_filter('acf/fields/relationship/result/name=mon_champ_relation',
'my_acf_relationship_result', 10, 4);

function my_acf_relationship_result($text, $post, $field, $post_id) {
    // personnaliser le texte affiché pour chaque résultat dans le champ
    relation
    // exemple : ajouter le type de contenu entre parenthèses après le titre de
    chaque résultat
    $text .= ' (' . $post->post_type . ')';

    // retourner le texte modifié
    return $text;
}
```

Les paramètres du filtre `acf/fields/relationship/result/` sont les suivants :

- **\$text** : le texte affiché pour chaque résultat dans la liste de sélection du champ relation. Par défaut, ce sera le titre de l'article ou de l'élément lié.
- **\$post** : l'objet post représentant l'élément lié.
- **\$field** : l'objet champ relation ACF en cours d'utilisation.
- **\$post_id** : l'ID de la publication en cours d'édition.