

Créer un rôle utilisateur WordPress en PHP

Ajouter un rôle utilisateur : `add_role()`

Pour créer un rôle utilisateur WordPress via PHP, vous pouvez utiliser la fonction **`add_role()`** fournie par WordPress.

Voici un exemple de code pour créer un rôle utilisateur personnalisé :

```
<?php
// inclure le fichier des fonctionnalités de WordPress
require_once(ABSPATH . 'wp-admin/includes/user.php');

// fonction pour créer un rôle utilisateur personnalisé
function creer_role_utilisateur_personnalise() {
    if ( ! get_option( 'custom_roles_version' ) ) {
        // définir les paramètres du rôle
        $role_slug = 'mon_role'; // slug du rôle (doit être unique)
        $role_name = 'Mon rôle utilisateur'; // nom affiché du rôle

        // ajoutez ici les capacités (capabilities) spécifiques pour ce rôle
        // exemple : 'read' => true, 'edit_posts' => true, 'delete_posts' =>
false
        $capabilities = array(
            'edit_posts' => true,
            'upload_files' => true,
            'manage_options' => true
        );

        // vérifier si le rôle n'existe pas déjà
        if ( ! role_exists( $role_slug ) ) {
            // créer le rôle utilisateur
            add_role( $role_slug, $role_name, $capabilities);

            /* pour éviter de systématiquement enregistrer le rôle personnalisé,
            on enregistrer dans la table wp_options une valeur qui sera testée avant
            écriture du rôle */
            update_option( 'custom_roles_version', 1 );
        }
    }
}

// Action pour créer le rôle lors de l'initialisation de WordPress
add_action( 'init', 'creer_role_utilisateur_personnalise' );
```

Voici quelques-unes des capacités couramment utilisées :

- **read** : permet à un utilisateur de lire le contenu du site.
- **edit_posts** : permet à un utilisateur de modifier les articles.

- **delete_posts** : permet à un utilisateur de supprimer les articles.
- **edit_pages** : permet à un utilisateur de modifier les pages.
- **delete_pages** : permet à un utilisateur de supprimer les pages.
- **edit_others_posts** : permet à un utilisateur de modifier les articles des autres utilisateurs.
- **publish_posts** : permet à un utilisateur de publier des articles.
- **upload_files** : permet à un utilisateur de télécharger des fichiers et des médias.
- **edit_themes** : permet à un utilisateur de modifier les fichiers de thème.
- **activate_plugins** : permet à un utilisateur d'activer les plugins.
- **manage_options** : permet à un utilisateur de gérer les options du site.

Ces capacités ne sont qu'un échantillon des possibilités offertes par WordPress.

Il existe de nombreuses autres capacités disponibles pour personnaliser les rôles utilisateurs en fonction de vos besoins spécifiques.

Pour obtenir la liste complète des capacités prédéfinies de WordPress, vous pouvez consulter la documentation officielle de WordPress sur la fonction `get_role()` :

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

Assurez-vous d'inclure ce code dans le fichier **functions.php** de votre thème WordPress actif ou dans un plugin personnalisé.

Lorsque vous exécutez le code, le rôle utilisateur personnalisé sera créé lors de l'initialisation de WordPress.

Remarque : le rôle utilisateur est enregistré en base de données, en cas de retrait du code PHP, ce dernier restera présent. Vous pouvez toutefois utiliser la fonction **remove_role()** pour le supprimer.

Ajouter / Supprimer des capacités à un rôle utilisateur : `add_cap()` et `remove_cap()`

Pour ajouter ou supprimer des capacités à un rôle utilisateur WordPress, vous pouvez utiliser les fonctions **add_cap()** et **remove_cap()**.

Ajouter une capacité à un rôle utilisateur :

```
<?php
```

```
// ID ou slug du rôle utilisateur
$role = 'editor';
```

```
// capacité à ajouter
$capability = 'edit_pages';
```

```
// obtenir l'objet WP_Role pour le rôle spécifié
$role_object = get_role($role);
```

```
// vérifier si le rôle existe et s'il ne possède pas déjà la capacité
if ($role_object && !$role_object->has_cap($capability)) {
```

```

    // ajouter la capacité au rôle
    $role_object->add_cap($capability);
}

```

Supprimer une capacité d'un rôle utilisateur :

```

<?php

// ID ou slug du rôle utilisateur
$role = 'editor';

// capacité à supprimer
$capability = 'edit_pages';

// obtenir l'objet WP_Role pour le rôle spécifié
$role_object = get_role($role);

// vérifier si le rôle existe et s'il possède la capacité
if ($role_object && $role_object->has_cap($capability)) {
    // supprimer la capacité du rôle
    $role_object->remove_cap($capability);
}

```

Dans les exemples ci-dessus, vous devez spécifier l'**ID** ou le **slug** du rôle utilisateur cible dans la variable **\$role** et la capacité à ajouter ou supprimer dans la variable **\$capability**.

Assurez-vous d'exécuter ce code dans le fichier **functions.php** de votre thème WordPress actif ou dans un plugin personnalisé.

Lorsque le code est exécuté, il ajoutera ou supprimera la capacité spécifiée pour le rôle utilisateur cible.

Vérifier le rôle ou les capacités de l'utilisateur actif

Pour vérifier le rôle ou les capacités de l'utilisateur WordPress actif, vous pouvez utiliser la fonction **current_user_can()**.

Vérifier le rôle de l'utilisateur actif

```

<?php

// vérifier si l'utilisateur actif a le rôle d'administrateur
if (current_user_can('administrator')) {
    // l'utilisateur actif est un administrateur
    echo 'Utilisateur actif avec le rôle d'administrateur';
}

```

Vérifier les capacités de l'utilisateur actif

```

<?php

// vérifier si l'utilisateur actif a la capacité de modifier les pages

```

```
if (current_user_can('edit_pages')) {  
    // l'utilisateur actif a la capacité de modifier les pages  
    echo 'Utilisateur actif avec la capacité de modifier les pages';  
}
```

Assurez-vous d'exécuter ce code dans le contexte approprié, tel que le fichier **functions.php** de votre thème WordPress actif ou dans un autre fichier de votre thème dans lequel vous avez besoin de vérifier le rôle ou les capacités de l'utilisateur actif.