Home / Reference / Classes / WP_Role / WP_Role::add_cap()

Search resources

WP_Role::add_cap(string $cap, bool $grant = true)

Assign role a capability.

Parameters

`$cap` string required

Capability name.

`$grant` bool optional

Whether role has capability privilege.

Default: `true`

More Information

Changing the capabilities of a role is persistent, meaning the added capability will stay in effect until explicitly revoked.

This setting is saved to the database (in table wp_options, field wp_user_roles), so it might be better to run this on theme/plugin activation.

Source

wp-includes/class-wp-role.php

Copy

```
public function add_cap( $cap, $grant = true ) {
	$this->capabilities[ $cap ] = $grant;
	wp_roles()->add_cap( $this->name, $cap, $grant );
}
```

View all references·View on Trac·View on GitHub

Related

Uses

wp_roles()

wp-includes/capabilities.php

WP_Roles::add_cap()

wp-includes/class-wp-roles.php

Changelog

Version	Description
2.0.0	Introduced.

User Contributed Notes

Codex9 years ago ^ 3 ˅

Example

Copy

```
function add_theme_caps() {
    // gets the author role
    $role = get_role( 'author' );

    // This only works, because it accesses the class instance.
    // would allow the author to edit others' posts for current theme only
    $role->add_cap( 'edit_others_posts' );
}
add_action( 'admin_init', 'add_theme_caps');
```

NB: This setting is saved to the database, so it might be better to run this on theme/plugin activation

Expand code Copy

```
function add_theme_caps(){
     global $pagenow;

     if ( 'themes.php' == $pagenow && isset( $_GET['activated'] ) ){ // Test if theme is active
         // Theme is active
         // gets the author role
         $role = get_role( 'author' );

         // This only works, because it accesses the class instance.
         // would allow the author to edit others' posts for current theme only
         $role->add_cap( 'edit_others_posts' );
     } else {
         // Theme is deactivated
```

To add capability to specific user :

Copy

```
$user = new WP_User( $user_id );
$user->add_cap( 'can_edit_posts' );
```

Log in to add feedback

Will Skora4 years ago ^ 2 ˅

Will Skora 4 years ago

Note that add_cap only accept one string; you cannot add multiple using a single add_cap call; you need to make multiple add_cap calls like the following:

```
$your_var -> add_cap( 'edit_others_posts' );
$your_var -> add_cap( 'upload_files' );
```

Log in to add feedback

You must log in before being able to contribute a note or feedback.