

# Registering Additional Theme Categories

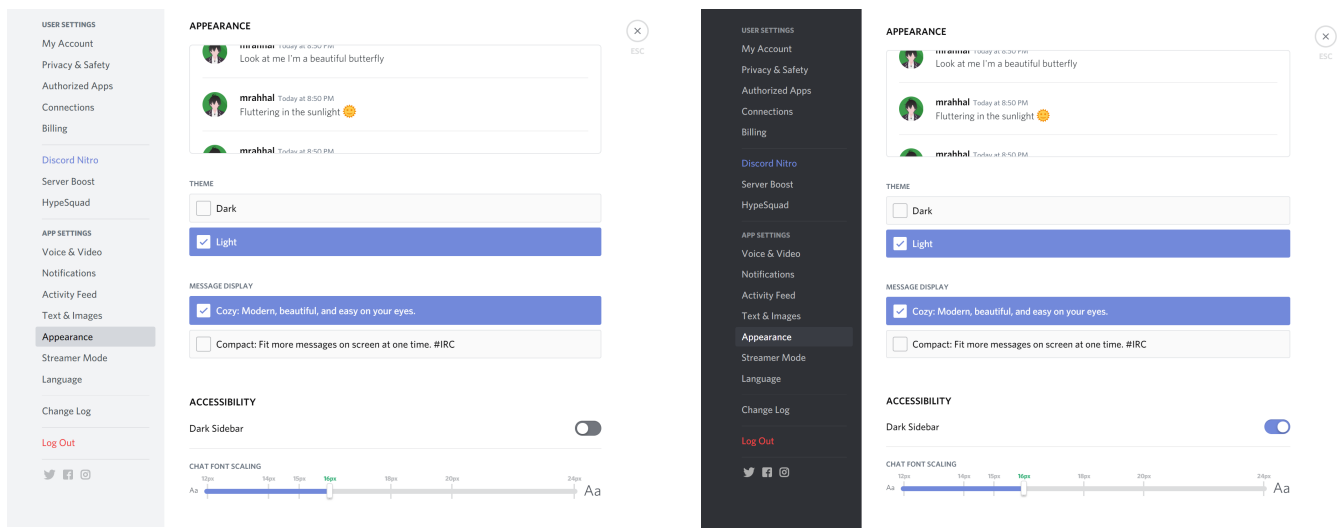
This lesson explains what is meant by registering an additional theme category and how to do it.

## WE'LL COVER THE FOLLOWING ^

- Introduction
- Theme categories
- Example

## Introduction #

Let's look at [discord](#) for a second:



You'll notice that you can change the sidebar's theme and that it becomes unrelated to the main theme of the app. That's really convenient, and this leads to the idea of having multiple **theme categories** as we'll call them.

## Theme categories #

Taking advantage of theme categories will allow you to create as many categories as you want so that you can apply themes at different levels of your app. Maybe you have some kind of a **launcher** page that you want to apply

app. Maybe you have some kind of a *launcher* page that you want to apply themes for. And maybe you also have some kind of *navbar* or *sidebar* to which you also want to scope another category of themes. Most of the functions in `css-theming` takes an optional `category` parameter, so each category scopes its themes so that you can deal with them individually.

It's a very powerful feature of `css-theming` and we'll learn now how to use it.

## Example #

As an example, we'll be creating a theme category for the sidebar in our app.

First, we'll start by registering this new category. We use the `ct-themes-category-register` mixin for that:

```
// Don't forget this import in your variables files!
@import '../..../node_modules/css-theming/src/scss/pure';

// Register a new themes category that we'll use for the sidebar
$ct-sidebar-themes: (
  'default': (
    'brightness': 'light',
  ),
  'default-dark': (
    'brightness': 'dark',
  )
);

@include ct-themes-category-register('sidebar', $ct-sidebar-themes, $applied-at: '.sidebar');
```

**Note:** The default/main theme category of the app is `''` (empty string) in CSS, and `null` in javascript. The `Theme` object contains this `category` property.

As you can see, we just created a new theme category called **sidebar** with 2 themes inside of it. We have to provide that name when calling the mixin, the themes map, and also where you want to apply the theme.

This `.sidebar` is the element that the theme's CSS variables will be set in.

Next, we have to make sure to call `initializeTheming` for this category

Next, we have to make sure to call `initializeTheming` for this category.

```
// Initializes the main default category
initializeTheming();

// Initializes the sidebar category
initializeTheming( /* theme */ null, /* category */ 'sidebar');
```

---

Now you are familiar with how to use `css-theming` in your app. The next section shows some samples to see how `css-theming` works in practice and across many use cases.