

# The 12 Tips of (ContentBox) Christmas - Day 6 - Themed Modules (<https://www.ortussolutions.com/blog/the-12-tips-of-contentbox-christmas-day-6-themed-modules>)



Gavin Pickin

Dec 20, 2016

As we continue with our 12 Tips of (ContentBox) Christmas, we at Ortus Solutions are giving you a gift of developer productivity, we will share a few tips and tricks that will keep giving all year around. Brad Wood is also doing 12 Tips of (CommandBox) Christmas. (<https://www.ortussolutions.com/blog/category/12-tips-of-commandbox-christmas>)

Modules are one of the key components of everything ContentBox. One frequently asked question is: how do we theme our custom module output? Theming your custom module, so it uses the same theme/layout as the rest of your ContentBox site is vital if you are going to be extending your site with your own modules. For consistency and flexibility, you do not want your module to be tightly coupled to the current theme, making it harder to change themes. We have a helper to help you with that.

## Step 1 : CBHelper Class

To use a theme, requires a lot of different pieces behind the scenes. Usually in the process of displaying a ContentBox blog post, or page, the UI Module that handles the process, prepares the UI Request, setting up the settings, themes etc. This has been made available to

you through the ContentBox CB Helper class. The first step is to get an injection or request the ContentBox helper object via its injection DSL: `CBHelper@cb`. You most likely will do this in your custom module handlers:

```
component{
    property name="cbHelper" inject="CBHelper@cb";
}
```

## Step 2 : prepareUIRequest()

The helper sports the `prepareUIRequest()` method which you will use to tell ColdBox to leverage the ContentBox module theme instead of a traditional ColdBox layout. Here is the method signature:

```
/**
 * Prepare a ContentBox UI request. This sets up settings, theme, etc. This method
 is usually called
 * automatically for you on the UI module. However, you can use it a-la-carte if you
 are building
 * ajax or module extensions
 * @layout An optional layout to set for you in the request, taken from the chosen
 theme
 */
CBHelper function prepareUIRequest( string layout );
```

This method accepts one argument called `layout` which by default is the pages layout from the chosen theme. If the ContentBox selected theme has more than 1 rendering layout, then you can even chose it here.

```
function listData( event, rc, prc ){
  // get some data
  prc.data = myservice.getData();

  // Tell ColdBox to use the ContentBox theme
  prepareUIRequest();

  // set view to render
  event.setView( "genera/data" );
}
```

The string for the name of the layout allows you to pick and choose the layout inside the theme. Pages is a common one, but you could have specifically designed layouts for different parts of your modules, or prefer the blog layout for example. Here is a link to the documentation on PrepareUIRequest: <http://apidocs.ortussolutions.com/contentbox/3.0.0-rc/contentbox/models/system/CBHelper.html#prepareUIRequest> ( <http://apidocs.ortussolutions.com/contentbox/3.0.0-rc/contentbox/models/system/CBHelper.html#prepareUIRequest%28> )

## Step 3 : Build Cool Stuff

We tried hard to make this as simple as possible, so you can drop a module in and have it using the theme in seconds. Try it out and go build cool stuff!

## Adding Meta to your Modules

One common issue with using the theme for your module is the way Meta Tags are generated. Having your own module, its hard to set your Title, Keywords and Description.

In ContentBox 3.1, coming out very soon, you'll be able to do this inside your Module Handler Events

```
function index(event,rc,prc){
  cbhelper.setMetaTitle( 'Gavins cool module' );
  cbhelper.setMetaKeywords( 'Gavins cool keywords' );
  cbhelper.setMetaDescription( 'Gavins cool description' );
  cbhelper.prepareUIRequest( "pages" );
  event.setView( "home/index" );
}
```

[www.ortussolutions.com](http://www.ortussolutions.com) (<http://www.ortussolutions.com>)

© Copyright **Ortus Solutions, Corp.**