Tweet

CBCMS2016 - Building your Own Admin Modules - Part 3 - Extending ContentBox 3

Gavin Pickin

Aug 09, 2016

In previous posts, we learned how to create a module inside of ContentBox, and then we added a handler and view so we could view the module behind the security of the login, but having an admin module, without menu items, is fairly silly (usually). In this blog post, we'll learn how you can add Submenu items to existing menus, like the Modules Main Menu.

If you remember back to part 1, we discussed some of the module life cycle events, specifically onLoad() and onUnLoad(). This is where all of the menu code is written. By adding code to both of these functions, the menu items are only added when they are needed.

onLoad - Adding a submenu item

First, we need the Admin Menu Service, so we can do our work.

```
// Let's add ourselves to the main menu in the Modules section
var menuService = controller.getWireBox().getInstance(
"AdminMenuService@cb" );
Now we have the menu service, we need to add the submenu item
menuService.addSubMenu(
    topMenu=menuService.MODULES,
    name="mySecretsSubMenu",
    label="Find my Secrets",
    href="#menuService.buildModuleLink('mysecrets','home')#"
);
```

Lets go through the params

- topMenu - this is the name of the topMenu. MenuService stores the main menus for you, in this case, we're adding a submenu to the Modules menu. This is a pretty standard behavior, so unless you're building a suite of modules, this would suffice.
- name - this is the reference name of the submenu. This is how you programmatically reference this subMenu item. You'll need this for the unload method, to remove the menu when we are unloading the module
- label - this is the actual text to be displayed in the menu itself. Find my Secrets in this case.
- href - this is the link that you'll go to when the submenu is clicked. Note we are using the menuService to build the module link, passing in the module, and the action. You could pass in home.index, but ColdBox's convention is to show the index action if none is specified.

Now, if you reload the module, rescan activate and deactivate is the best way, you'll add a menu item, and it should work.... **BUT WAIT.**

If you add it, and don't add the code to remove it, you could get multiple menu items. I have done this one time, I had to comment out the add, and deactivate and reactivate over and over until we had no menus again... so lets add the code for the unload really quick.

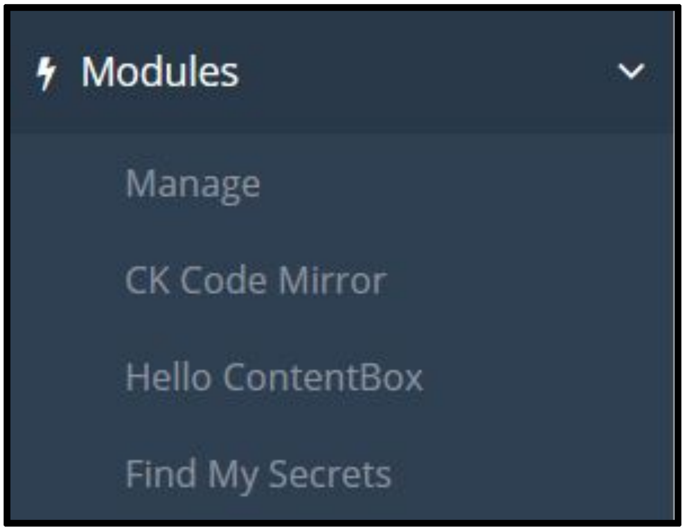
Removing the Submenu

Again, we need to get an instance of the AdminMenuService so we can work with the Admin Menus.

```
var menuService = controller.getWireBox().getInstance(
"AdminMenuService@cb" );
Once we have the Admin Menu Service, we can remove the SubMenu with
just 2 params.
// Remove Menu Contribution
menuService.removeSubMenu(
    topMenu=menuService.MODULES,
    name="mySecretsSubMenu"
);
```

- topMenu - the top Menu item from which you want to remove the SubMenu from.
- name - the programmatic reference to the SubMenu that you added

So your new Module Config should look like this
https://gist.github.com/gpickin/174d0279c80b3a8ddaa6ac352a1dbab4



Now when you deactivate and reactivate the mysecrets module, the menu now exists. Click on the link, and your module will appear
Deactivate again, and its gone.

This is a simple step, but writing it out, has filled this blog post up... so lets call it a day, and in the next post, we'll learn how to make your own Top level menu, for your suite of ContentBox modules.