ALLAN THRAEN | 4 years ago | PDF |

Optimizely (Episerver) Addon Development Tips and Tricks

OVERRIDING AND EXTENDING EPISERVER DOJO FUNCTIONALITY

This is a small guide on how you can easily change or extend any of the Episerver dojo modules.

I still find myself being oddly scared whenever I'm asked to dabble in the dark arts, aka Dojo.

And when I do seemingly succeed in my endavours, I never seem to be able to shake the feeling that surely something must be broken somewhere I just haven't spotted it. Yet.

It always seems like Dojo - and Episerver Dojo in particular - is something best learned after spending 5 years at Hogwarts having your world turned upside down.

But - as this code sample will show, it is after all just javascript. So, any code structure is arbitrary and only enforced by developers - and that can work to our advantage.

Task: I was asked to disable the context menu item "Open in image Editor" in the Asset manager, for images implementing a certain interface.

First step is to figure out what episerver dojo code to override/change. Obviously you can download the UI source as a nuget package - but you can also just unzip the Episerver CMS Module zip file, and all the scripts are available in an uncompressed version within it.

I'll admit I had a hard time figuring out where the logic was I needed to change. In the end I had to call a friend (thanks, Deane) who called a friend - and in the end the great Greg Wiechec prevailed (thanks again!) and pointed me to "epi-cms\command\EditImage.js". It had a function called _onModelChange which looked at the currently selected model and which interfaces it uses - which was just what I needed. So, all I needed to do was to extend that method with my logic and I should be set.

But how do I do that correctly in dojo? Surely this I must know, I thought. Had we been in c# land it would be easy - inherit the class, override what you want and register it with dependency injection. But now?

Looking through some of Greg's old posts, he provides a wonderful example as a part of this solution: https://gregwiechec.com/2018/09/view-on-website-in-new-window/.

I'm sharing my version here as I'm sure myself (and maybe some others can use the same approach to overriding logic in other cases).

Basically the steps are as follows:

1. Create an Initializer.js in your ClientResources/Scripts.
2. Register it in your module.config to ensure it's loaded
   <clientModule initializer="alloy/Initializer">
   <moduleDependencies>
   <add dependency="CMS" type="RunAfter" />
   </moduleDependencies>
   </clientModule>
   Notice I use "alloy" as I'm doing it in the site's module.config, on an alloy site where the script path is named "alloy". This should of course change.
3. In the "initialize" function of your js file (which is pretty barebones), simply override the function on the relevant component you wish to change. In my case, I first call the original function, then add my logic - so I'm extending rather than overriding the logic. Code sample below.

```
define([
    "dojo/_base/declare",
    "epi/_Module",
    "epi/shell/TypeDescriptorManager", //This is only needed for my specific case, to check if a mode
    "epi-cms/command/EditImage" //This is the 'class' I want to extend
], function (declare, _Module,TypeDescriptorManager,EditImage) {

        return declare([_Module], {
            initialize: function () {
                this.inherited(arguments);

                var original = EditImage.prototype._onModelChange; //Store original functionality in
               //Implement extended functionality
                EditImage.prototype._onModelChange = function () {
                    original.apply(this, arguments); //First call old functionality
                    var target = this._getSingleSelectionData();
                    if (target) {
                        var isMyInterface =
                            TypeDescriptorManager.isBaseTypeIdentifier(target.typeIdentifier, "my.low

                        if (isMyInterface) {
                            this.set("isAvailable", false);
                            this.set("canExecute", false);
                        }
                    }
                }
            }
        });

});
```

Optimizely (Episerver) Addon Development Tips and Tricks

CodeArt ApS
Teknikerbyen 5, 2830 Virum, Denmark
Email: info@codeart.dk
Phone: +45 26 13 66 96
CVR: 39680688