


 ALLAN THRAEN |  5 years ago |  PDF | 

Addon Development Tips and Tricks CMS Optimizely (EpiServer)

ROUTING IN EPISERVER ADDON MODULES



When your build your own addons, modules and extensions for Episerver CMS, you often want to include controllers - and naturally you want to call these controller - but which url should you use? I always forget this, so here is a little reminder.

Case: You have a great idea for an addon for Episerver and you've started coding it. Part of it involves a controller - or some other resource that you want to call through a URL. Perhaps in a menu provider or for a GUI plugin. But what is the right approach?

Solution

As a simple example, I took a vanilla Alloy site, added a new folder under Modules/_protected called "RouteTest" and I also added a new project (class library) to my solution called "RouteTest", and by adding the Nuget package of EPiServer.CMS.AspNet, I got the pretty much all the dependencies I needed in that project. Then I made the simplest possible controller:

```
public class RouteTestController : Controller
{
    public ActionResult Index()
    {
        return Content("Hello World");
    }
}
```

To test it out, I added a reference to the library from my site. But now the big question is - how can I call my controller?

Obviously, it's worth trying out the most basic default route {controller}/{action}/{id}. And guess what - "/RouteTest" works! But sadly you can't really count on that route being available - even though it is in our Alloy test. Also, it's pretty ugly to call protected module controllers like that (not to mention that they aren't behind our usual security which is defined on a path level in web.config).

Instead we need to add a <module.config> file to the root of our module folder - and reference that in web.config in the episerver.shell\protectedModules folder.

This will auto-register a route to EPiServer.Shell.Web.Routing.ModuleRouteCollection for ~/EPiServer/RouteTest which is a great start!

Order #	Name	Type	Url
1	No Name	System.Web.Routing.Route	EPiServer/EPiServer.Cms.TinyMce/2.7.1/ClientResources/tiny_mce/langs/culture.js
2	No Name	EPiServer.Shell.Web.Routing.ModuleRouteCollection	~/EPiServer/RouteTest
3	No Name	EPiServer.Shell.Web.Routing.ModuleRouteCollection	~/EPiServer/EPiServerDeveloperTools
4	No Name	EPiServer.Shell.Web.Routing.ModuleRouteCollection	~/EPiServer/EPiServer.Cms.TinyMce/2.7.1
5	No Name	EPiServer.Shell.Web.Routing.ModuleRouteCollection	~/EPiServer/EPiServer.Cms.TinyMce/2.7.1/ClientResources/tiny_mce/langs/culture.js

The last piece of the puzzle is to also ensure that your module.config file references the assembly that contains the controller - otherwise it wouldn't be able to add this controller to this path:

```
<?xml version="1.0" encoding="utf-8" ?>
<module productName="RouteTest" loadFromBin="false">
  <assemblies>
    <add assembly="RouteTest" />
  </assemblies>
</module>
```

In the module.config there's obviously many other things that can be configured - and when speaking of routes you could add an entire route section with custom mapping of controllers, their prefixes and their defaults. But for my basic use-case that is not needed. Read more here: world.episerver.com/documentation/developer-guides/CMS/configuration/Configuring-moduleconfig/

Dynamically getting the URL

Finally, should you want to dynamically get the path to any resource (controller, view, static file) located in a protected module, you can always use:

```
EPiServer.Shell.Paths.ToResource("RouteTest","RouteTest")
```

Where the first parameter is the module name, and the second the relative path to the resource.

Addon Development Tips and Tricks CMS Optimizely (EpiServer)

RECENT POSTS

