ALLAN THRAEN  |  🕐 6 years ago  |  📄 PDF  |  🗨

Tips and Tricks    Addon Development    Optimizely (Episerver)

# CUSTOM VIEWS FOR AN INTERFACE

**Ever played around with adding custom views in Episerver CMS? It's a really powerful way to extend the UI. But why does it work when you register your view for a model class, but not for an interface implemented by that model? I had a look and found out.**

Dont get your hopes up - this is just a short blog post to remind myself of a fun little detail that would otherwise cause me to probably waste an hour again some day looking for exactly the same issue.

It's a neat and pretty common thing to add custom views for your content. I've done it in a dozen integrations of various kinds, and it's described in many (!) blog posts:

- https://world.episerver.com/blogs/Duong-Nguyen/Dates/2013/12/Custom-views-and-plugin-areas-in-EPiServer-75/
- https://world.episerver.com/blogs/Linus-Ekstrom/Dates/2014/4/Adding-custom-views-to-your-content/
- http://jondjones.com/learn-episerver-cms/episerver-developers-guide/episerver-customizing-episervers-ui/displaying-a-custom-on-page-editing-view

Today, I was once again using the views, but I needed to have them work for an interface rather than the specific model class. Something like this:

```
[ContentType(GUID = "EE3BD195-7CB0-4756-AB5F-E5E223CD9820")]
[MediaDescriptor(ExtensionString = "pdf,docx")]
public class GenericMedia : MediaData, IMyContentInterface
{
    /// <summary>
    /// Gets or sets the description.
    /// </summary>
    public virtual String Description { get; set; }
}
```

But when I changed it to be an interface it stopped showing up. That seemed a bit odd, so instantly I fired up dotPeek and took a look at how the Episerver views are registrered. And they are registrered towards IContentData and IContent - so why won't it work for me? Here's my view:

```
[ServiceConfiguration(typeof(EPiServer.Shell.ViewConfiguration))]
public class MyView : ViewConfiguration<IMyContentInterface>
{
    public MyView()
                    {
        Key = "myView";
        Name = "My View";
        Description = "A view with my stuff";
        ControllerType = "epi-cms/widget/IFrameController";
        ViewType = VirtualPathUtility.ToAbsolute("~/mypathtomycontroller/myview");
        IconClass = "myicon";

    }
}
```

Turns out to be stupid simple: Views needs to match up their types with the proper UIDescriptors. If you don't have a UI Descriptor, then get one - even if you don't plan to use it.

This fixed the issue:

```
[UIDescriptorRegistration]
public class MyUIDescriptor : UIDescriptor<IMyContentInterface>
{
    public MyUIDescriptor()
                    {
        //Potentially set the default view or disable some other views
    }
}
```

Tips and Tricks    Addon Development    Optimizely (Episerver)