ALLAN THRAEN | 1 years ago | PDF |

C# Tips and Tricks Digizuite Optimizely (Episerver)

MAKE SOME ASSET TYPES DOWNLOAD AUTOMATICALLY IN OPTIMIZELY CMS 12 USING THE DIGIZUITE INTEGRATION



Here is just a quick tip for how you can do custom blob http handling when using the Digizuite integration to Optimizely (Episerver) CMS 12.

This is a quick code sample I put together for a recent client request - I figured it might be useful to others using the Digizuite integration for Optimizely CMS so I am sharing it here.

In this case there was a need to make sure that certain types of assets (files) in Digizuite were always downloaded as attachments when the user would click on a link to them in Optimizely CMS.

Naturally, Digizuite supports the standard Optimizely approach of sending assets out as attachments when the url to them is appended with "/download" but in this case it was needed to be the default behavior for any links to them.

In my local sample site and test DAM I had some PDF files so in this case I configured it for PDF files - but in reality it could be any asset type in Digizuite this would work for.

As you can see in the code sample below, I essentially just piggy-back on the existing infrastructure in the Digizuite integration.

First step is straight forward: Define a custom content type in Optimizely CMS for these asset-types, so we can handle them separately from the rest.

In the case of Pdf files, they are by default handled as a DigizuiteFile (the generic catch all content type included in the integration) so I just inherited from that and made a new type called "DownloadableDocument".

This step is fairly well documented here:
https://digizuite.atlassian.net/wiki/spaces/DD/pages/3354657560/DFO+3.1.0+-+2+Modelling+Asset+Content+Types

Next, I inherit the DigizuiteBlobHttpHandler with the single purpose of re-using it's downloading functionality - but just making sure it's used every time.

And finally I created a new Template handler in the shape of a DownloadableFileMediaEndpoint that ties the new content type together with the new custom BlobHttpHandler.

```
1   //Register the PDF's as a specific content type in Optimizely. Ensure Guid is unique.
2   //Connect to Digizuite asset-types using the DigizuiteType attribute.
3   [ContentType(DisplayName = "Downloadable Digizuite File", GUID = "a4ca5288-35ce-4ee5-b094-72f76ca8e07b", Description = "", AvailableInEditMode = false)]
4   [DigizuiteType(AssetTypeIds = new[] { Digizuite.Optimizely.Constants.GlobalConstants.AssetTypes.PDF })]
5   public class DownloadableDocument : DigizuiteFile
6   {
7   }
```

DownloadableDocument.cs hosted with ♥ by GitHub view raw

```
1   //Create a custom HTTP Handler for the blobs of these types - but just inherit from the DigizuiteBlobHttpHandler.
2   //Just make sure you override the "IsDownloadingRequest". The default behavior here would be for it to check if the url ends with "/download".
3   [ServiceConfiguration]
4   public class DownloadableDocumentBlobHttpHandler : Digizuite.Optimizely.BlobHandling.DigizuiteBlobHttpHandler
5   {
6     public DownloadableDocumentBlobHttpHandler(HttpClient httpClient, IOptions<DAMIntegrationConfiguration> options) : base(httpClient, options)
7     {
8     }
9
10    public override bool IsDownloadingRequest(HttpContext httpContext)
11    {
12      return true;
13    }
14  }
```

DownloadableDocumentBlobHttpHandler.cs hosted with ♥ by GitHub view raw

```
1   //Make a custom Media Endpoint that connects the new content type with the new BlobHttpHandler.
2   [TemplateDescriptor(Inherited = false, AvailableWithoutTag = true, TemplateTypeCategory = TemplateTypeCategories.HttpHandler)]
3     public class DownloadableFileMediaEndpoint : Endpoint, IRenderTemplate<DownloadableDocument>, IRenderTemplate
4     {
5       private static ContentActionDescriptor _mediaContentActionDescriptor = new ContentActionDescriptor()
6       {
7         Inherited = false,
8         ModelType = typeof(DownloadableDocument)
9       };
10      private static AuthorizeAttribute _authorizeAttribute = new AuthorizeAttribute("episerver:read");
11      private static AuthorizeAttribute _previewAttribute = new AuthorizeAttribute("episerver:preview");
12
13      public DownloadableFileMediaEndpoint(DownloadableDocumentBlobHttpHandler dblobhandler) : base(new RequestDelegate(dblobhandler.Invoke), new Enc
14        (object) DownloadableFileMediaEndpoint._mediaContentActionDescriptor,
15        (object) DownloadableFileMediaEndpoint._authorizeAttribute,
16        (object) DownloadableFileMediaEndpoint._previewAttribute
17      }), nameof(DownloadableFileMediaEndpoint))
```



DownloadableFileMediaEndpoint.cs hosted with ♥ by GitHub view raw

C# Tips and Tricks Digizuite Optimizely (Episerver)

RECENT POSTS

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

Copyright © 2024