

DAM Assets AddOn

Deployment

1. In the Visual Studio, add the DAM Asset AddOn package to your solution using the from Package Manager.
2. Update the start up process in the Startup.cs

```
// ..... OTHER SERVICES / MODULES

using DamAssetAddOn.ServiceExtensions;

public class Startup
{
    private readonly IWebHostEnvironment _webHostingEnvironment;

    public Startup(IWebHostEnvironment webHostingEnvironment)
    {
        _webHostingEnvironment = webHostingEnvironment;
    }

    public void ConfigureServices(IServiceCollection services)
    {
        // ..... OTHER SERVICES / MODULES

        // Required by Wangkanai.Detection
        services.AddDetection();

        services.AddDamAssetAddOn();

        // ..... OTHER SERVICES / MODULES
    }

    public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        // ..... OTHER SERVICES / MODULES

        app.UseDamAssetAddOn();
        app.UseEndpoints(endpoints =>
        {
            endpoints.MapContent();
            endpoints.MapControllers();
        });
    }
}
```

3. Create a Controller for the DAM Asset Block. For example `Controllers\DamAssetBlockViewComponent.cs`

```
using Microsoft.AspNetCore.Mvc;
using EPiServer.Web.Mvc;

namespace DamAssetAddOn.Gadget
{
    public class DamAssetBlockViewComponent : BlockComponent<DamAssetBlock>
    {
        protected override IViewComponentResult InvokeComponent(DamAssetBlock currentContent)
        {
            return View("~/Views/DamAssetBlock/index.cshtml", currentContent);
        }
    }
}
```

4. Create a View for the DAM Asset Block. For example Views\DamAssetBlock\index.cshtml

```

@model DamAssetAddOn.Gadget.DamAssetBlock

<div class="dam-asset-block">
<h2 class="title">DAM Assets</h2>

@if (!string.IsNullOrEmpty(Model.damAssetList))
{
    <div class="grid-container">
        @{
            // Parse the JSON string into a dynamic object
            var assets = System.Text.Json.JsonSerializer.Deserialize<System.Text.Json.JsonElement>(Model.damAssetList);
        }

        @foreach (var asset in assets.EnumerateArray())
        {
            <div class="grid-item">
                <div class="image-container">
                    
                </div>
            </div>
        }
    </div>
}
else
{
    <p>No assets available.</p>
}
</div>

<style>
/* Overall Container Styles */
.dam-asset-block {
    font-family: 'Arial', sans-serif;
    margin: 20px auto;
    max-width: 1200px;
}

.dam-asset-block .title {
    text-align: center;
    font-size: 24px;
    font-weight: bold;
    color: #333;
    margin-bottom: 20px;
    text-transform: uppercase;
    letter-spacing: 1px;
}

/* Grid Styles */
.grid-container {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
    gap: 20px;
    padding: 20px;
}

.grid-item {
    background-color: #fff;
    border-radius: 10px;
}

```

```
        overflow: hidden;
        box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
        transition: transform 0.3s ease, box-shadow 0.3s ease;
    }

    .grid-item:hover {
        transform: translateY(-5px);
        box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);
    }

    /* Image Container Styles */
    .image-container {
        position: relative;
        width: 100%;
        padding-top: 100%;
        /* Makes the container square */
        overflow: hidden;
        background-color: #f9f9f9;
    }

    .image-container img {
        position: absolute;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        object-fit: cover;
        /* Ensures the entire image fits without distortion */
    }
</style>
```

Configure DAM Asset Block

1. Start Optimizely CMS
2. Go to **Settings** menu and select the **DAM Asset Block**
3. Update the settings for the block. Adjust the **Description** field as below Note: Each entry must be on a separate line and must follow the **Name = Value** format
 - a. Update the **Panel URL**
 - b. Enter the **Connection Key**
4. Save the Block settings