**Technical Specification**

**Project: .NET 8 Core MVC Frontend Renderer Custom Extension  
Date: July 2025**

**Objective**

Set up and extend a **Sitefinity .NET Core Renderer frontend** connected to a **Sitefinity CMS 15.3 backend** by using dotnet templates to create custom widgets, and layouts.

**1. Prerequisites**

| **Tool** | **Version** | **Notes** |
| --- | --- | --- |
| .NET SDK | .NET 8.0 | [Download](https://dotnet.microsoft.com/download/dotnet/8.0) |
| Sitefinity CMS Backend | v15.3.8522 | Already installed, running on IIS |
| Visual Studio | 2022 or later | With .NET Core and ASP.NET workloads |
| Node.js (Optional) | 18+ | For asset building (if needed) |

**2. Install Sitefinity Template CLI**

Sitefinity’s frontend development uses CLI templates for scaffolding components.

**dotnet new install Progress.Sitefinity.AspNetCore.Templates::15.2.8432.1**

Additional Notes: Replace version as needed to match your project setup.

**3. Create Custom Templates**

Use these commands in PowerShell **from the root of your .NET Core Renderer project not the .sln root folder**. Normally you will to run:

**cd <project-name>**

**3.1 Widget**

**dotnet new sitefinity-widget --widget-name "MyCustomWidget"**

This creates a new widget under:

Views/Shared/Components/MyCustomWidget/Default.cshtml

ViewsComponents/ MyCustomWidget / MyCustomWidget ViewComponent.cs

**3.2 Layout**

**dotnet new sitefinity-layout --layout-name "TestLayout"**

Layout goes into: Views/Shared/TestLayout.cshtml

**4. Build and Run the Frontend**

**4.1 Restore and Build the Frontend Project**

**dotnet build**

If you're using Visual Studio:

* Right-click project → Set as startup project → F5 or Ctrl+F5

**5. Test Custom Widget in Page Editor**

1. Navigate to **Design → Pages Template → Create a Template** in Sitefinity CMS.
2. Fill in the Name, code-name and choose thumbnail that are suitable.

A screenshot of a computer

AI-generated content may be incorrect.

1. Choose any based templates from .NET Core Templates.

Additional notes: New layout will be added as base templates.

A screenshot of a computer

AI-generated content may be incorrect.

1. Click on **Add section here…** then on **Content** as you can see the new CustomWidget will appear on top.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Extra Notes**

* Always **clean and rebuild** after scaffolding a new widget.
* Use meaningful names for widgets/layouts.
* Note that all the custom widgets will only appear on the new editor.
* Don’t change base templates once draft/publish the page template.
* Once page template is published it can directly access from the **Pages (Editor).**
* For more customize extension and the best practices can refer link **Github** in **References**.

**7. References**

| **Resource** | **Link** |
| --- | --- |
| Sitefinity Docs – Custom ASP.NET Core Widget | [🔗 Docs](https://www.progress.com/documentation/sitefinity-cms/create-a-custom-asp.net-core-widget) |
| Sitefinity ASP.NET Core Renderer Extension Github | [🔗 Docs](https://github.com/Sitefinity/sitefinity-aspnetcore-mvc-samples) |