Razor class library and reusable blazor components

Blazor is an excellent framework and allows us to create reusable Blazor components. We created the following CustomInputSelect component in our previous video. This component allows us to bind an integer property to an input select element.

using Microsoft.AspNetCore.Components.Forms;

namespace Pragim.Components

{

public class CustomInputSelect<TValue> : InputSelect<TValue>

{

protected override bool TryParseValueFromString(string value, out TValue result,

out string validationErrorMessage)

{

if (typeof(TValue) == typeof(int))

{

if (int.TryParse(value, out var resultInt))

{

result = (TValue)(object)resultInt;

validationErrorMessage = null;

return true;

}

else

{

result = default;

validationErrorMessage =

$"The selected value {value} is not a valid number.";

return false;

}

}

else

{

return base.TryParseValueFromString(value, out result,

out validationErrorMessage);

}

}

}

}

Copy

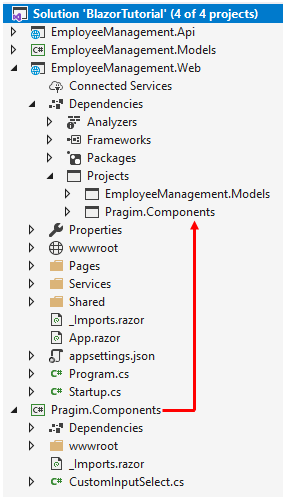
Component reusability and project type

The project type determines the reusability of a blazor component. If the blazor component is created in a web project, we may not be able to reuse it in other projects.

To be able to reuse a blazor component in multiple projects, create it in a **razor class library project**. This project can then be referenced by any project and use the Blazor component. We can even package and ship them as a NuGet package.

Reusing blazor components

To be able to use blazor components in your project, add a project reference to the razor class library project. We can then use the Blazor component as we normally use.



Do not forget to include the required namespace. If you plan to use the blazor component in multiple pages, you may include the namespace in \_Imports.razor file. This allows us to use the components in the namespace in all the pages across our project. There's no need to include the using declaration in each individual page.