USING MUDBLAZOR

A BLAZOR COMPONENT LIBRARY

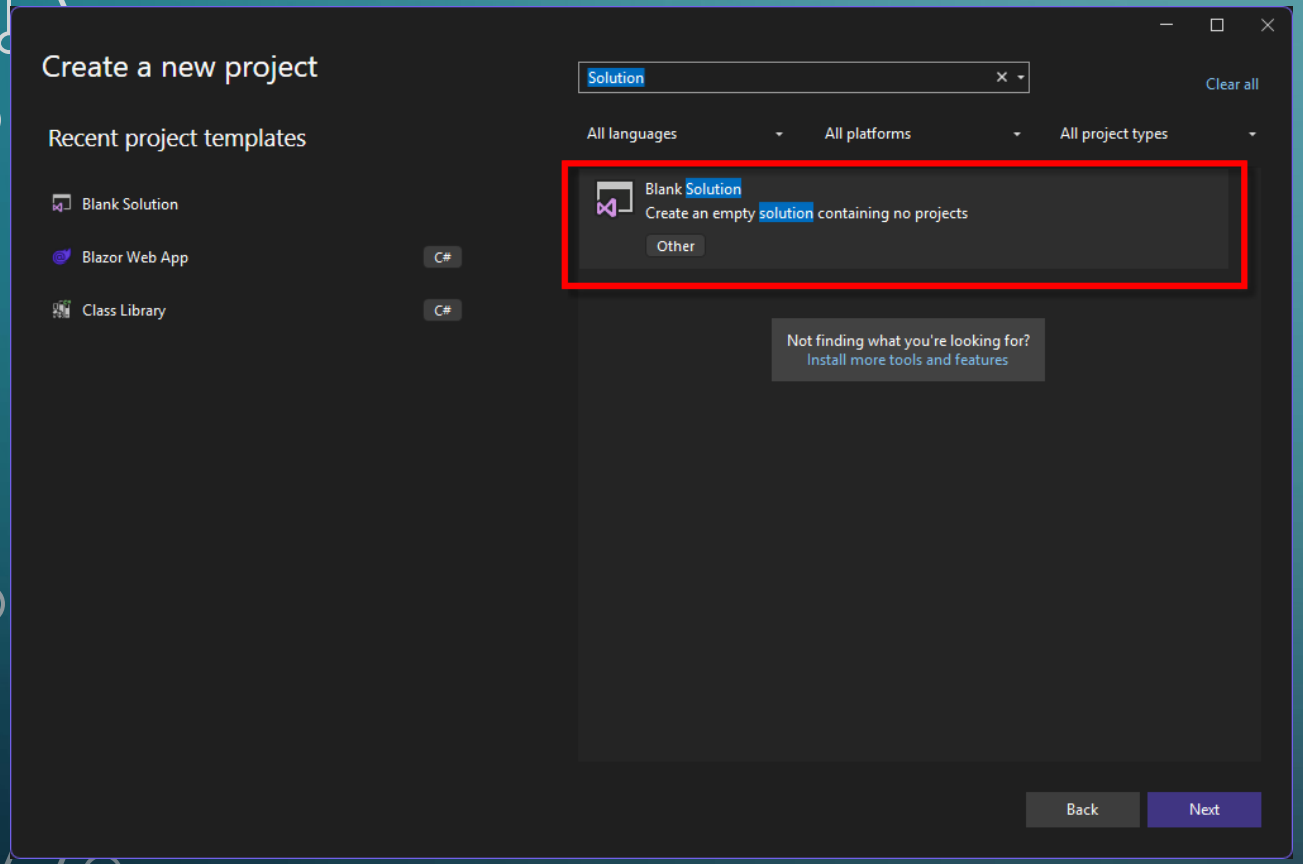
WHAT IS MUDBLAZOR?

- MudBlazor is an Open Source (no cost) Component Library that can be used with Blazor projects. It was created by Jonny Larsson and Meinrad Recheis but has had many contributors since.
- MudBlazor is based on Material Design. (https://m2.material.io/design/introduction)

MUDBLAZOR TEMPLATES

- The quickest method to get MudBlazor up and running is to use their Templates through Visual Studios.
- Since this class uses Microsoft Identity for Authentication we will create the template with authentication.

TEMPLATE STEPS (1)



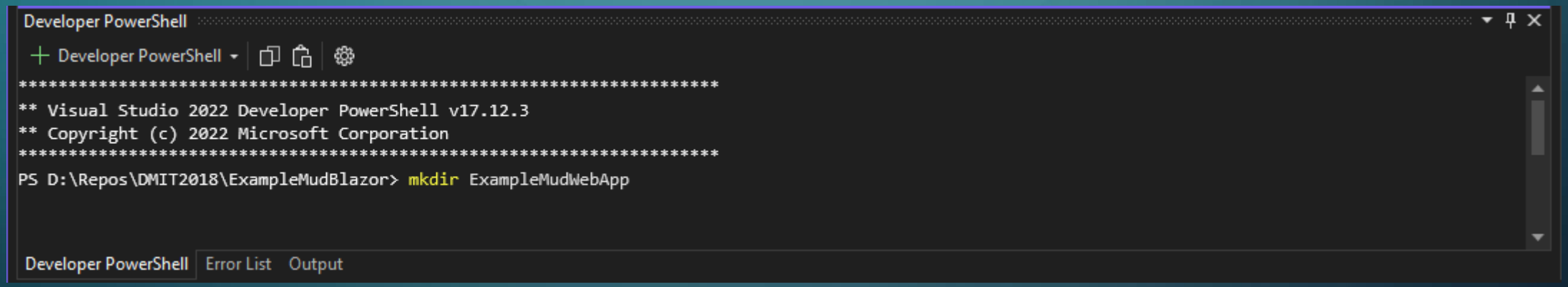
1. Create a blank solution in Visual studios and open the terminal.

TEMPLATE STEPS (2)

2. Open the Terminal (View > Terminal) and create a folder in that solution with the following command:

mkdir [FolderName] (Example: mkdir ExampleMudWebApp)

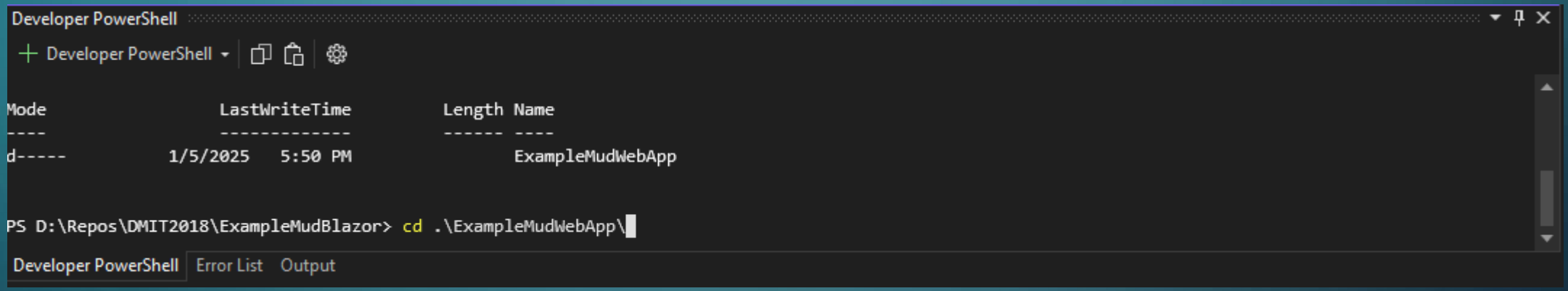
Note: Name the Folder the same name as you want to Blazor Web App project to be named.



TEMPLATE STEPS (3)

3. Navigate to the new folder using the following command:

cd [FolderName] (you can press the tab key to auto-complete)



TEMPLATE STEPS (4)

4. Type or copy in the following command:

dotnet new mudblazor --interactivity Server --auth Individual --all-interactive --use-local-db

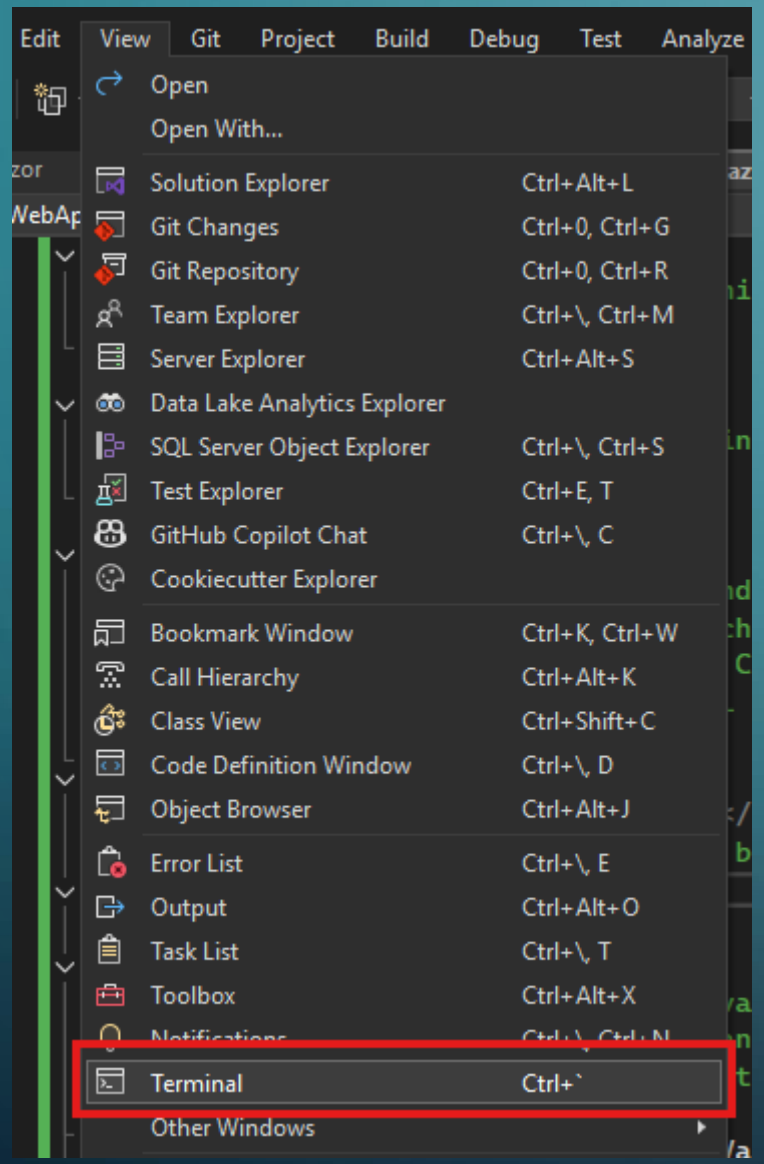
Notes:

- --interactivity can be Server, Auto, or WebAssembly. We are strictly using Server, this indicates the render mode to use
- --auth with Individual indicates we want to include Individual Authorization Pages
- --all-interactive make the project global interactivity (adds the rendermode to the App.razor component so it does not need to be defined on each page)
- --use-local-db make the project use SQLServer and not SQLLite (note: If you see SQLLite added as a Nuget Package, you missed this)

```
PS D:\Repos\DMIT2018\ExampleMudTemplate> cd .\ExampleMudTemplateApp\
PS D:\Repos\DMIT2018\ExampleMudTemplate\ExampleMudTemplateApp> dotnet new mudblazor --interactivity Server --auth Individual --all-interactive --use-local-db
```

Developer PowerShell | Error List | Output

INSTALLING TEMPLATES – ONLY NEEDS TO BE RAN ONCE

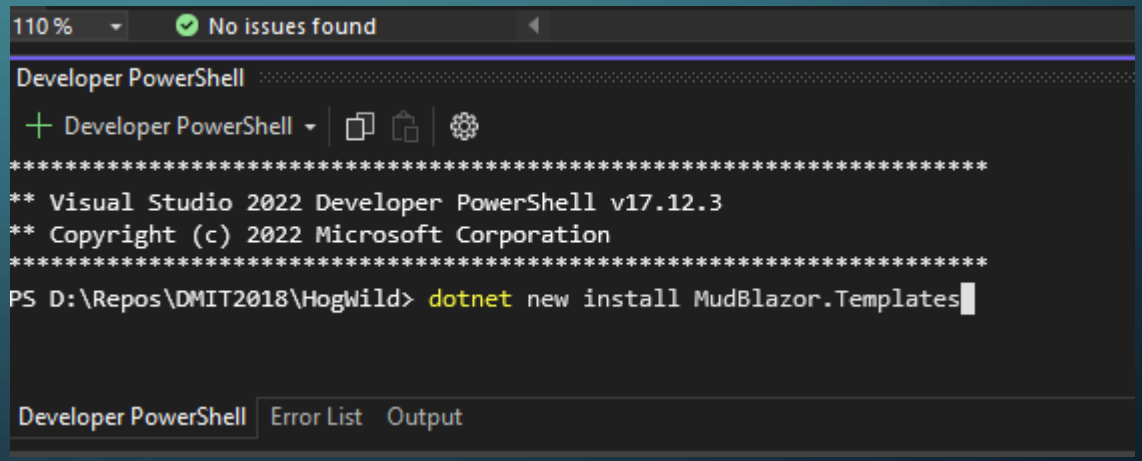


Install Templates to Visual Studio

a. Open the Terminal by selecting View > Terminal from the menu.

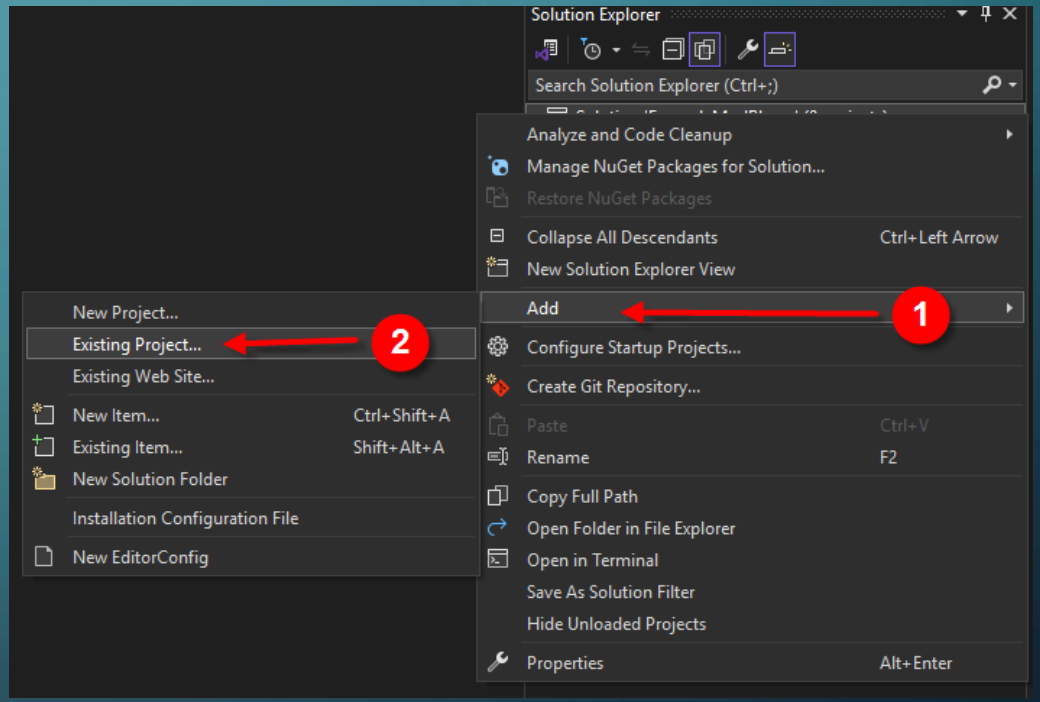
b. Type 'dotnet new install MudBlazor.Templates' in the window that opens and hit enter.

c. Check that a Success message is displayed.



TEMPLATE STEPS (5)

- 5. Add your new project to the solution by right-clicking on the solution and selecting **Add > Existing Project…**



END RESULTS

In the end you will have a Solution with one Blazor Web App project.

Explore the MudBlazor MainLayout and NavMenu and Included Sample Pages to see some uses of MudBlazor Components.

Reference MudBlazor's documentation to see available MudBlazor Components and examples.

https://mudblazor.com/docs/overview