Blazor Basics 3

What is it, how does it work, when and how to use.

Henk Holterman
h.holterman@4dotnet.nl

Agenda: Day 3

- Summary of Day 2
- Overflow Day 2
  - A simple MD editor with Preview
  - StateManagement
  - JS Interop (GeoLocation)

Testing the TabControl, test outline:

```
[Fact]
public void TabControlShouldActivatePageOnClick()
{
    // Arrange
    using var ctx = new TestContext();
    var subject = ctx.Render(fixture);


    // act
    var heading2 = subject.FindAll("a").Last();
    heading2.Click();


    // Assert
    var pageContent = subject.Find("p");


    pageContent.MarkupMatches(@<p>twee</p>);
}
```

Databinding in Blazor: 2-way binding

- In Home.razor, add this:

```
<Counter Step="10" Count="myCount" CountChanged="c => myCount = c"/>
```

- The Counter should now do what we want.

- This is the basic pattern, used a lot. There is a shorthand notation. Add a second Counter like this:

```
<Counter Step="5" @bind-Count="myCount"  />
```

- This does exactly the same and both Counters bind to the same variable.

Blazor syntax, parameters and directives

```
@onclick="IncrementCount"          Ok
onclick="IncrementCount"           JS error
```

- The thing on the right hand side of = is called a *value*.
- On the left we have:
- HTML Attributes class="btn" onclick="click"
- Blazor Parameters Message="@message" Rows="12"
- Blazor Directives @bind="Value" @onclick="Click"

Build a MD editor with a Preview

- New Project, add an Editor page, add

```
<textarea rows="10" cols="50" @bind="@mdText"
@bind:event="oninput" />
```

- Add a NuGet package for MD, eg Markdig
- Convert to a string and display in the page.
  How to show it in real-time?
  Why does this show HTML?
- Blazor has a MarkupString to display ‘unsafe’ text.

About @bind and @bind-Value

- @bind only works for HTML inputs

```
<textarea @bind="mdText"
    @bind:event="oninput" />
```

- And **@bind-PropertyName** is for Blazor Input components:

```
<InputTextArea @bind-Value="mdText"
@oninput="OnInput" />

void OnInput(ChangeEventArgs e)
     => mdText = (string?) e.Value ?? "";
```

Browser Storage

Soort	Algemeen	Blazor Server	Blazor Wasm
Session	Scope: 1 Browser Tab	Size in kB. Can encrypt.	Limit 5-10 MB.
Local	Scope: All Tabs, persistent	Size in kB. Can encrypt.	Limit 5-10 MB.
IndexedDb	Semi SQL	Not relevant	Limit 10 MB – 2GB
Browser Cache	Linked to URL	Not relevant	Used by Blazor, PWA

- Blazor Server Interactive:
  - transport over SignalR. Not suited for large data.
  - The server can encrypt, hiding the content from the user.

- The limits (quota) are Browser dependent, this are averages.

Exercise

- Open the StateManagement solution.
- Study the StateService class.
- Register it with DI (Singleton, Scoped, Transient ?)
- Inject it in the Counter Page
- Make the Counter resume where it was when switching pages.
- Find the stored value in the Dev Tools.

Limits of razor: creating html tags

- The challenge: can we make a <h3> tag like this?

```
<Header Level="3">
    Im a header tag
</Header>
```

1. Try to make a Header1.razor,
   see if you can do this.
2. Make a Header2.cs,
   - inherit from ComponentBase
   - override **BuildRenderTree**

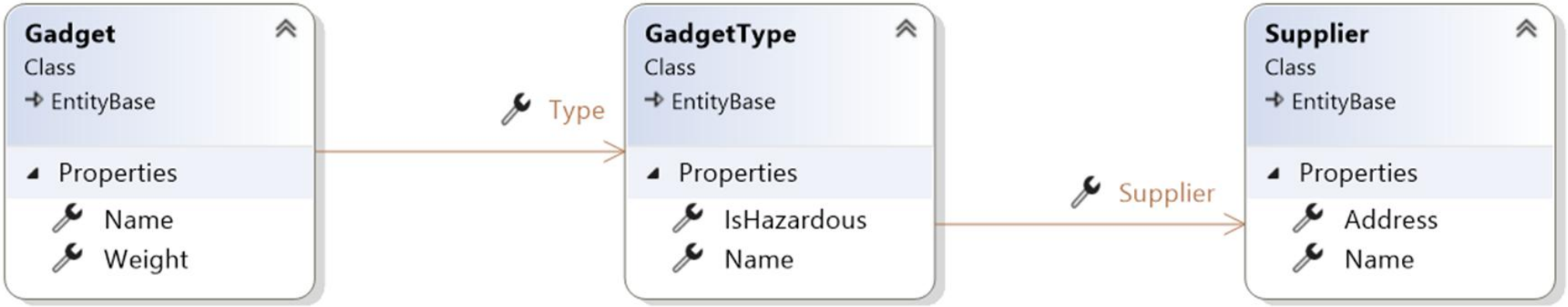
Why? This is an entirely different way to do Blazor, sometimes you will need it.

Short intro JS-Interop: GeoLocation

- This is prepared code, we look at it together.
- Look how the (JS) Module is loaded, async
- Have a quick look at the JS, and how the control flows from C# -> JS -> C#

A basis CRUD app

- In the repo there is a Gadgets solution,
- It contains just a Domain class library



Adding Blazor

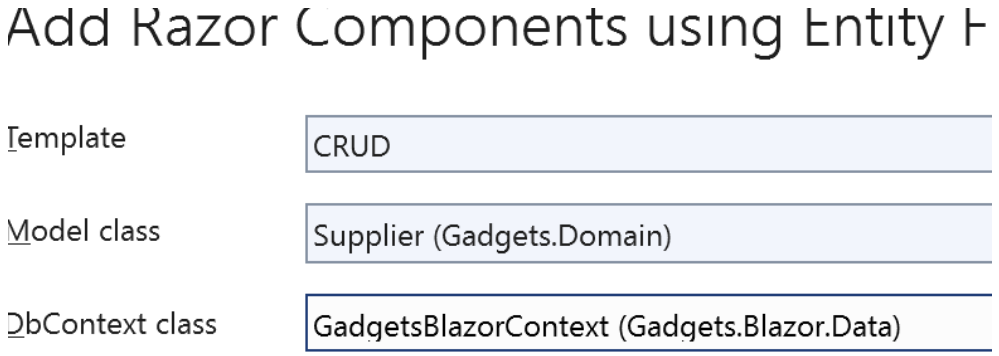
- Add a Blazor (Server, per page, no auth) project to the Solution.
- Name it **Gadgets.Blazor**
- Add the project reference.

- Note: in the following steps, VS will generate a lot of code, add packages and classes.
- To follow this, add the Solution to a (local) git repo and commit between steps.

Scaffold the CRUD

- On the Pages folder, right-click to Add a “Scaffolded Item”
- Pick “Razor Components using EF”
- Select the Supplier class
- Use [+] to add a new DbContext
- Chose SQL or SQLite

- Click [Add], study the code



Scaffolding and the Db

- Run the App, add **/suppliers** in the address bar.
- Try to create a supplier.
- Look at the error p[age, follow the instruction to create the Db

- Run again
- Create a few Suppliers

Scaffolding the GadgetType

- Repeat the process to add the GadgetType pages.
- Create one.
- The SaveChanges will fail. How do we solve this?

a) Make Supplier Nullable, add and apply the Migration.
b) And/Or add markup and code to select a Supplier.

Discussion.

- Is this useful?
- Alternatives?

EditForms: prevent moving away

- Use a (simple) EditForm

```
// prevent moving away
EditForm? editForm;          // use an @ref=
IDisposable? registration;  // implement IDisposable

 override protected void OnAfterRender(bool firstRender)
 {
   if (firstRender)
     { registration = Navigator.RegisterLocationChangingHandler(AreWeClean); }
 }
 async ValueTask AreWeClean(LocationChangingContext locationChangingContext)
 {
   if (editForm?.EditContext?.IsModified() is true)
   {
       locationChangingContext.PreventNavigation();
       // show a message (with async timeout)
       await InvokeAsync(StateHasChanged);
   }
 }
```