

.NET Conf

2022

Student Zone



In partnership with:



Reactor

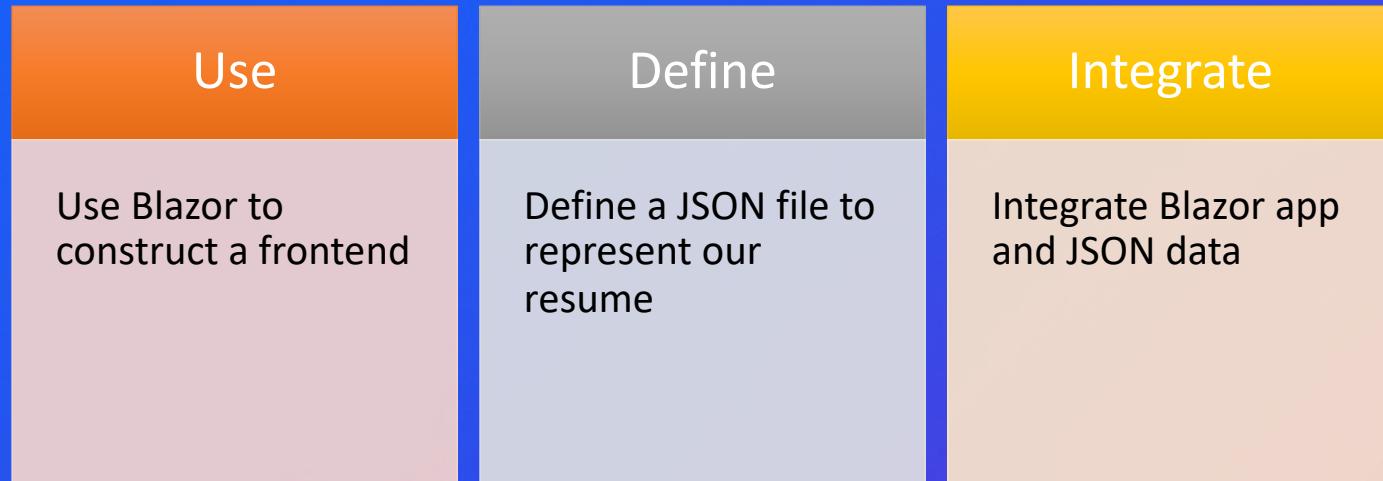


Build a powerful resume app using Blazor

Matt Soucoup
Principal Cloud Advocate - .NET

Our plan

- Build a resume using Blazor



Thinking in Components

A component is something that should do one thing well

Comes in two different flavours

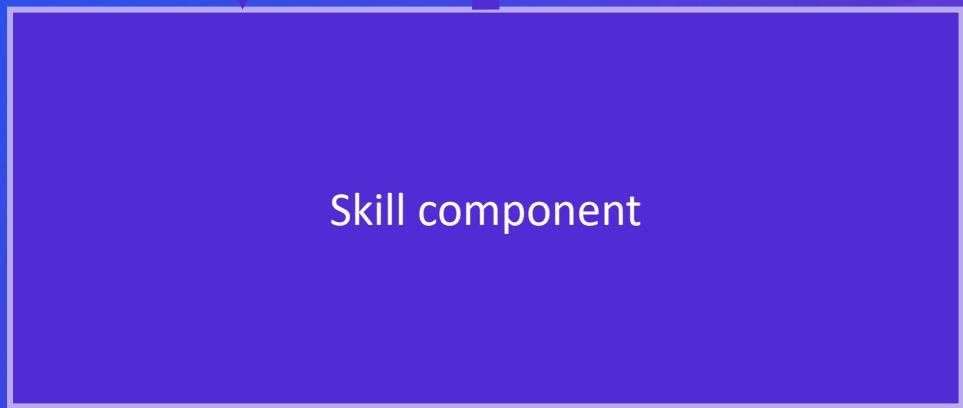
- Presentation components, "just" presents the data
- "Smart" components, manages data and events

Architectural principles

Data flows down
Events flow up

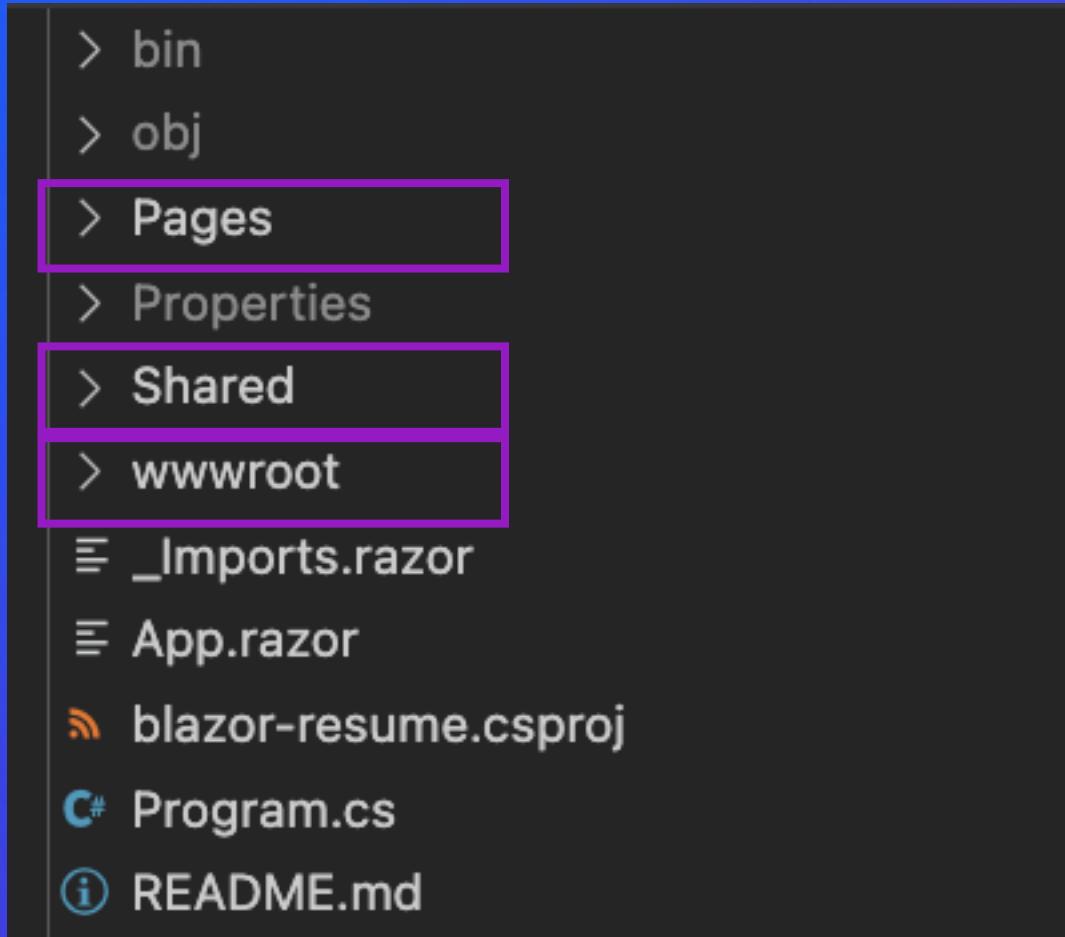
Render skill (data)

Skill added (event)



Create a blazor app

```
dotnet new blazorwasm -o  
ResumeApp
```



A component in Blazor

- Navigation with @page
- Render data with @
- Respond to events with @event="EventHandler"

```
@page "/counter"

<PageTitle>Counter</PageTitle>

<h1>Counter</h1>

<p role="status">Current count: @currentCount</p>

<button class="btn btn-primary" @onclick="IncrementCount">Click me</button>

@code {
    private int currentCount = 0;

    private void IncrementCount()
    {
        currentCount++;
    }
}
```

Components

Architecting our app

Navigation

About

Experiences

Educations

Skills

Resume - navigation

Shared

>MainLayout.razor

MainLayout.razor.css

≡ NavMenu.razor

```
<div class="nav-item px-3">
<NavLink class="nav-link" href="resume">
<span class="oi oi-plus" aria-hidden="true"></span> Resume
</NavLink>
</div>
```

Resume – resume component

- Setup navigation @page, /resume
- Define “name” and render it in template area

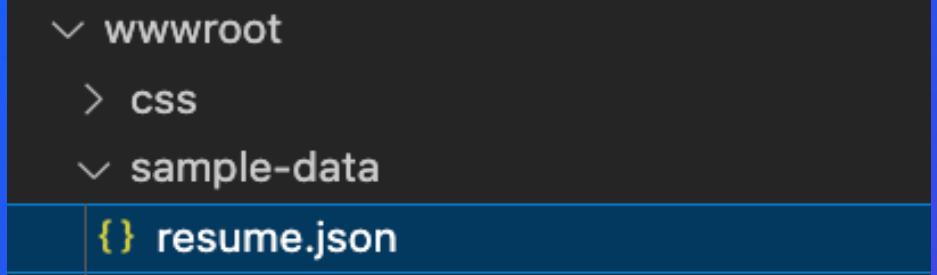
```
@page "/resume"
```

```
<PageTitle>@name</PageTitle>
```

```
<div class="about">
<h2>About</h2>
</div>
```

```
@code {
    string name = "matt";
}
```

What does our JSON data look like?



```
{  
  "about" : ".NET developer with a proven track  
  record and with a growth mindset. Additional  
  skills in HTML and JavaScript and CSS",  
  "experiences" : [  
    {"companyname": "Microsoft",  
     "title" : "Software developer",  
     "tenure" : "2020-2022",  
     "description" : "Software developer. As part  
       of a team, built 2 services that served 20 000  
       users and improved the service's profit by 10%"  
    },  
    { ...  
    },  
    { ...  
    }],  
  "educations": [ ...  
  ],  
  "skills" : [{ ...  
  }]
```

How do we model that in C#?

```
using System.Collections.Generic;
namespace Types;

public class ResumeModel {
    public string About { get; set; }

    public List<ExperienceModel> Experiences { get; set; }
    public List<EducationModel> Educations { get; set; }
    public List<SkillModel> Skills { get; set; }
}

public class ExperienceModel
{
    public string CompanyName { get; set; }
    public string Tenure { get; set; }

    public string Description { get; set; }
    public string Title { get; set; }
}
```

```
public class EducationModel
{
    public string Name { get; set; }
    public string DegreeYear { get; set; }
    public string Description { get; set; }
}

public class SkillModel
{
    public string ? Name { get; set; }
    public int ? EstimatedLevel { get; set; }
    public string ? Description { get; set; }
}
```

Resume – resume component, fetch data

- Inject HTTP capability
- Define OnInitializedAsync()
- Request data from JSON file

```
@page "/resume"  
@using Types  
@inject HttpClient Http
```

```
<PageTitle>@name</PageTitle>
```

```
<div class="about">  
<h2>About</h2>  
@resume.About  
</div>
```

```
@code {  
    string name = "matt";  
    public ResumeModel resume { get; set; }
```

```
protected override async Task OnInitializedAsync()  
{
```

```
    resume = await Http.GetFromJsonAsync<ResumeModel>("sample-  
    data/resume.json");  
}
```

Let's render the fetched data

```
<div class="experiences">
<h2>Experiences</h2>
@foreach (var experienceModel in resume.Experiences)
{
    <Experience data=@experienceModel></Experience>
}
</div>
```

```
<div class="educations">
<h2>Education</h2>
@foreach (var educationModel in resume.Educations)
{
    <Education data=@educationModel></Education>
}
</div>
```

```
<div class="skills">
<h2>Skills</h2>
@foreach (var skillModel in resume.Skills)
{
    <Skill data=@skillModel></Skill>
}
</div>
```

Resume – Educations, Skills, Experiences

```
@using Types
```

```
<div class="experience">
<h3>@data?.Title</h3>
<h4>@data?.CompanyName (@data?.Tenure)</h4>
<div>@data?.Description</div>
</div>
```

```
@code {
```

```
    [Parameter]
```

```
    public ExperienceModel? data { get; set; }
}
```

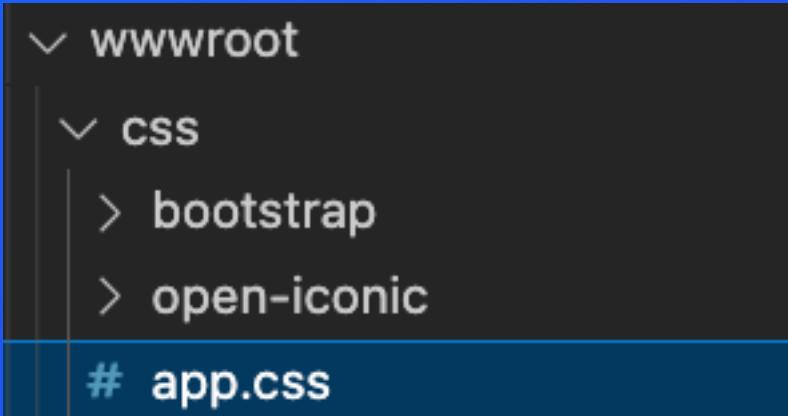
Resume – styling, close to the component

Styling should be close to the component

Component.razor

Component.razor.css

Some exception where you add CSS app wide in



☰ Experience.razor
♯ Experience.razor.css

```
.experience {  
    margin: 10px 0;  
    padding: 10px;  
    background-color: black;  
    border-radius: 5px;  
}
```



DEMO

Student Resources

<http://aka.ms/learnstudent>

Cloud Skills Challenge

<https://aka.ms/dotnetstudentcsc>

GitHub Repo

<https://github.com/microsoft/dotnetconf-studentzone>