



# DotVVM

## Model-View-ViewModel for .NET web apps

Speaker Name

Contact Info

# Web Apps in .NET in 2020

- Plenty of technologies and approaches available today
- Razor Pages or ASP.NET Core MVC
  - Great for mostly static websites and pages
- REST API + Angular or React
  - No limits
  - It's hard to learn both .NET and JavaScript stacks

# Web Apps in .NET in 2020

## • Blazor

- No need to learn JavaScript – stay with C# and HTML
- There is a cost for Blazor
  - Server-Side Blazor needs low-latency network
  - Client-Side Blazor needs to download a few MBs of data to the client

## • DotVVM

- No need to learn JavaScript – stay with C# and HTML
- Different approach than Blazor
  - Less than 100kB of libraries
  - No SignalR connection needed

# What is DotVVM?

- Open-source framework for ASP.NET and ASP.NET Core
  - <https://github.com/riganti/dotvvm>
- Model-View-ViewModel approach
- Stay with C# and HTML
  - Knowing JavaScript helps, but is not necessary 😊
- Member of the .NET Foundation
  - [www.dotnetfoundation.org](http://www.dotnetfoundation.org)

# Who is DotVVM for?

- Anyone who likes C# and .NET
- Anyone who likes MVVM
  - WPF and Xamarin developers
- Anyone who's using ASP.NET Web Forms
  - Learning and switching to DotVVM is easy!
  - <https://www.dotvvm.com/webforms>

# How it works

- ViewModel = C# class
- View = HTML file (with some syntax extensions)

```
public class IncrementViewModel
{
    public int Value { get; set; }

    public void Increment()
    {
        Value++;
    }
}
```

```
<p>
    Current value:
    {{value: Value}}
</p>

<p>
    <dot:Button
        Text="Increment"
        Click="{command: Increment()}" />
</p>
```

# Model-View-ViewModel

## • View

- HTML syntax
- Extended with components and data-binding expressions

## • ViewModel

- C# class
- Represents the state of the UI
- Handles commands coming from the user

## • Model

- Provides data and business logic to the ViewModel
- Commonly seen as "Services" and "Data-Transfer Objects"



# Demo

Basic demo of DotVVM



# How to start

- Install the extension for Visual Studio
  - <https://www.dotvvm.com/install>
- File > New Project > DotVVM in .NET Core
- Learn basic concepts at DotVVM Academy
  - <https://academy.dotvvm.com>

# Features of DotVVM

- Plenty of Built-in Components
  - TextBox, CheckBox, RadioButton, ComboBox...
  - GridView, Repeater, FileUpload...
- Routing
- Validation
- SPA (Single Page Apps)
- Server-Side Rendering

# Supported Frameworks

- ASP.NET Core 2.0 and newer
- .NET Framework 4.5.1
  - Through OWIN
- Extension for **Visual Studio 2019** and **Visual Studio 2017**
  - Project templates (File > New Project > DotVVM)
  - Syntax highlighting
  - IntelliSense

# How much?

## • Free

- Open-source DotVVM framework (with ~30 built-in controls)
  - <https://github.com/riganti/dotvvm>
- Visual Studio extension
  - <https://www.dotvvm.com/install>

## • Commercial extensions

- Bootstrap for DotVVM (helpers for Bootstrap 3 and 4)
- DotVVM Business Pack (more advanced components)
- DotVVM Pro for Visual Studio (VS extension with more features)

# Resources

- GitHub
  - <https://github.com/riganti/dotvvm>
- Documentation
  - <https://dotvvm.com/docs>
- Official website
  - <https://www.dotvvm.com>
- Gitter chat (ask us anything!)
  - <https://gitter.im/riganti/dotvvm>



# DotVVM

## Q&A

Speaker Name

Contact Info