



What's up with .NET Maui

Mark Allibone
Rey Technology

@mallibone



Hello my name is Mark



Mark Allibone



@mallibone



Rey Technology



<https://mallibone.com>



<https://nullpointers.io>

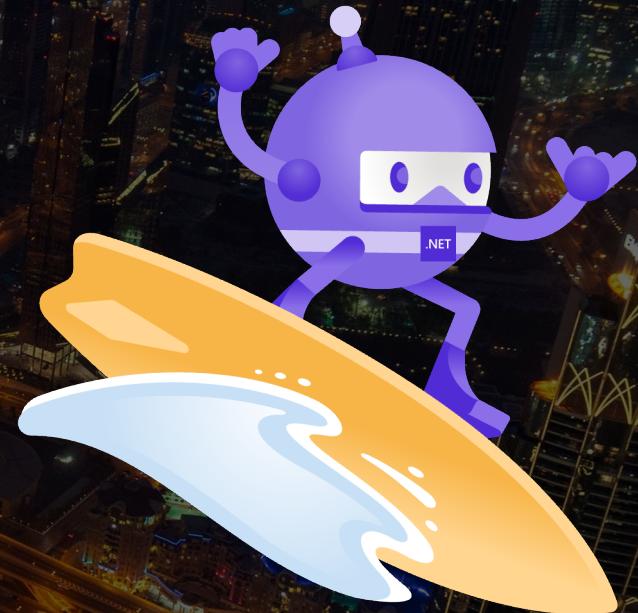




@mallibone

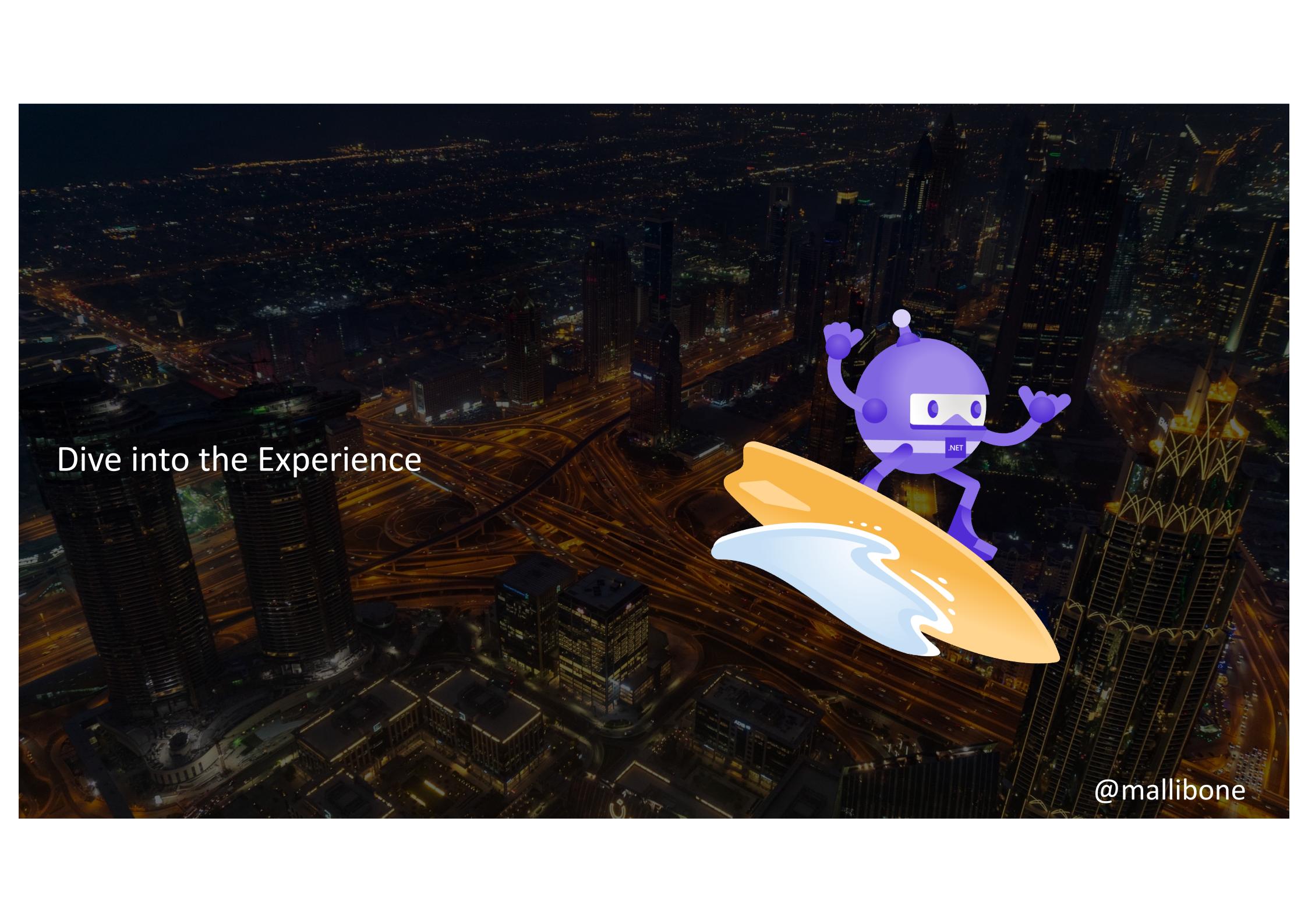


.NET MAUI the evolution of Xamarin Forms

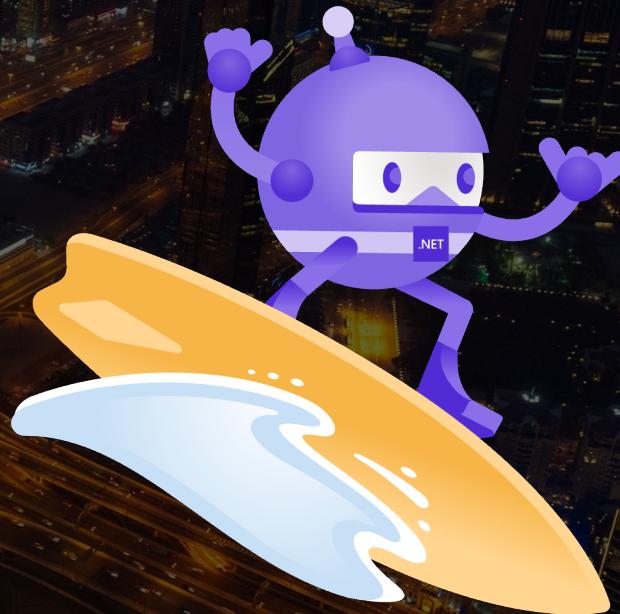


@mallibone

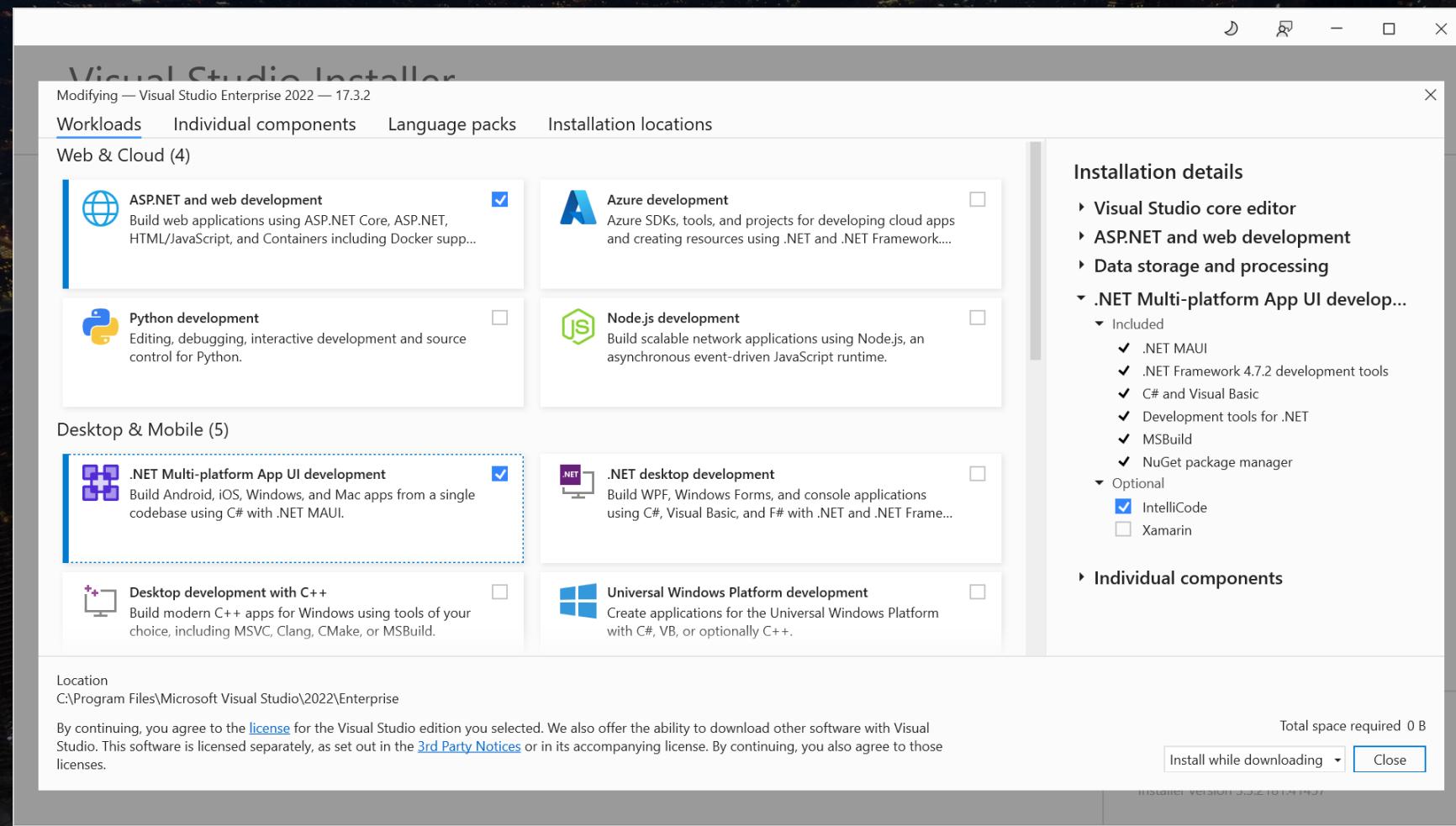




Dive into the Experience



@mallibone



@mallibone

Visual Studio Installer

Modifying — Visual Studio Enterprise 2022 — 17.3.2

Workloads Individual components Language packs Installation locations

Web & Cloud (4)



ASP.NET and web development

Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...



Azure development

Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....



Python development

Editing, debugging, interactive development and source control for Python.



Node.js development

Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.



Desktop & Mobile (5)



.NET Multi-platform App UI development

Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.



.NET desktop development

Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...



Desktop development with C++

Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.



Universal Windows Platform development

Create applications for the Universal Windows Platform with C#, VB, or optionally C++.



Installation details

C:\Program Files\Microsoft Visual Studio\2022\Enterprise

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Total space required

Install while downloading ▾

Installer version 17.3.2.10147497

Visual Studio Installer

Modifying — Visual Studio Enterprise 2022 — 17.3.2

Workloads Individual components Language packs Installation locations

Web & Cloud (4)



ASP.NET and web development

Build web applications using ASP.NET Core, ASP.NET, HTML/JavaScript, and Containers including Docker supp...



Azure development

Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework....



Python development

Editing, debugging, interactive development and source control for Python.



Node.js development

Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.



Desktop & Mobile (5)



.NET Multi-platform App UI development

Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.



.NET desktop development

Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...



Desktop development with C++

Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.



Universal Windows Platform development

Create applications for the Universal Windows Platform with C#, VB, or optionally C++.



Location

C:\Program Files\Microsoft Visual Studio\2022\Enterprise

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Total space require

Install while downloading ▾

Installer version 17.3.2.10147497

Installation details

- ▶ Visual Studio core editor
- ▶ ASP.NET and web development
- ▶ Data storage and processing
- ▶ .NET Multi-platform App UI development

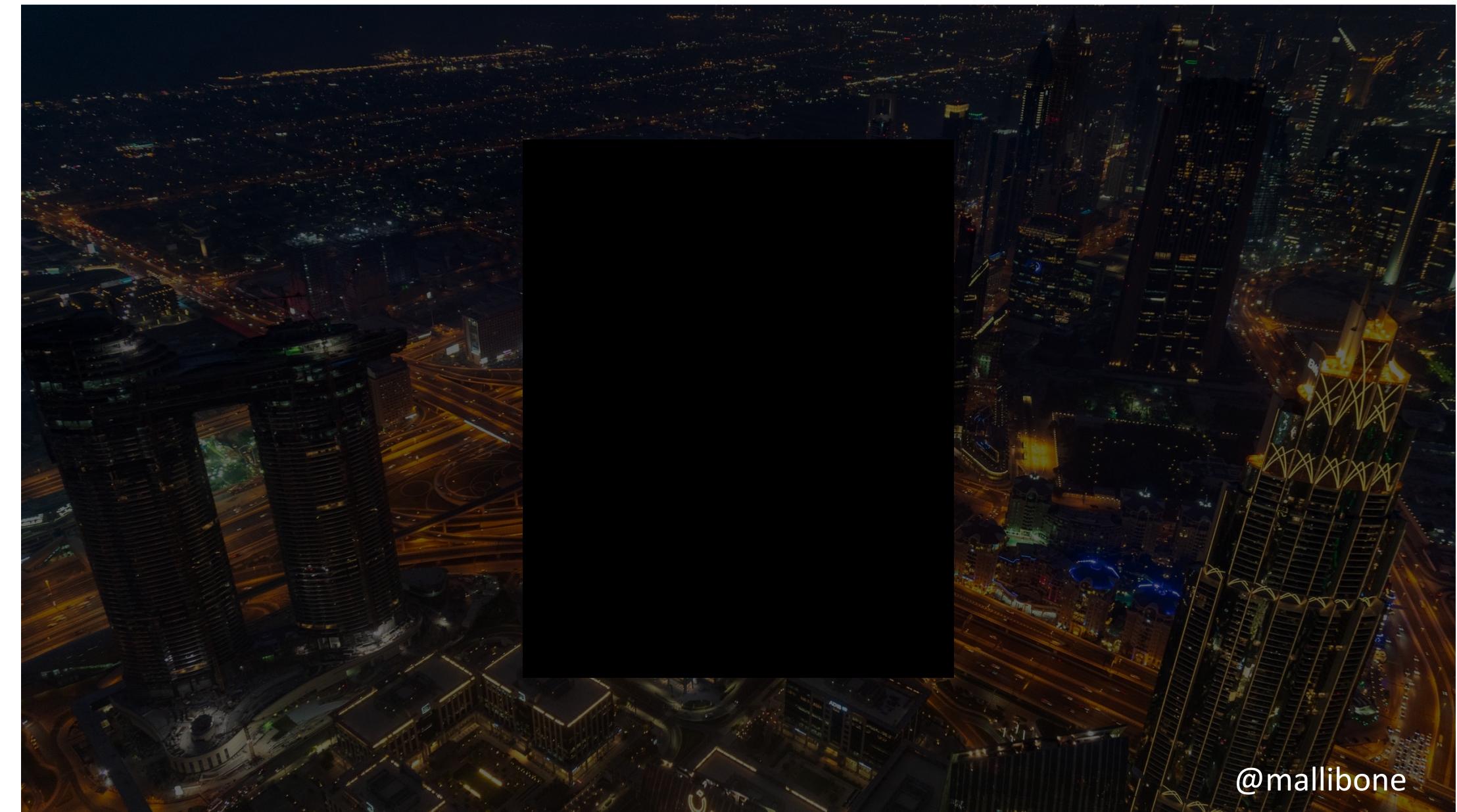
Included

- ✓ .NET MAUI
- ✓ .NET Framework 4.7.2 development tools
- ✓ C# and Visual Basic
- ✓ Development tools for .NET
- ✓ MSBuild
- ✓ NuGet package manager

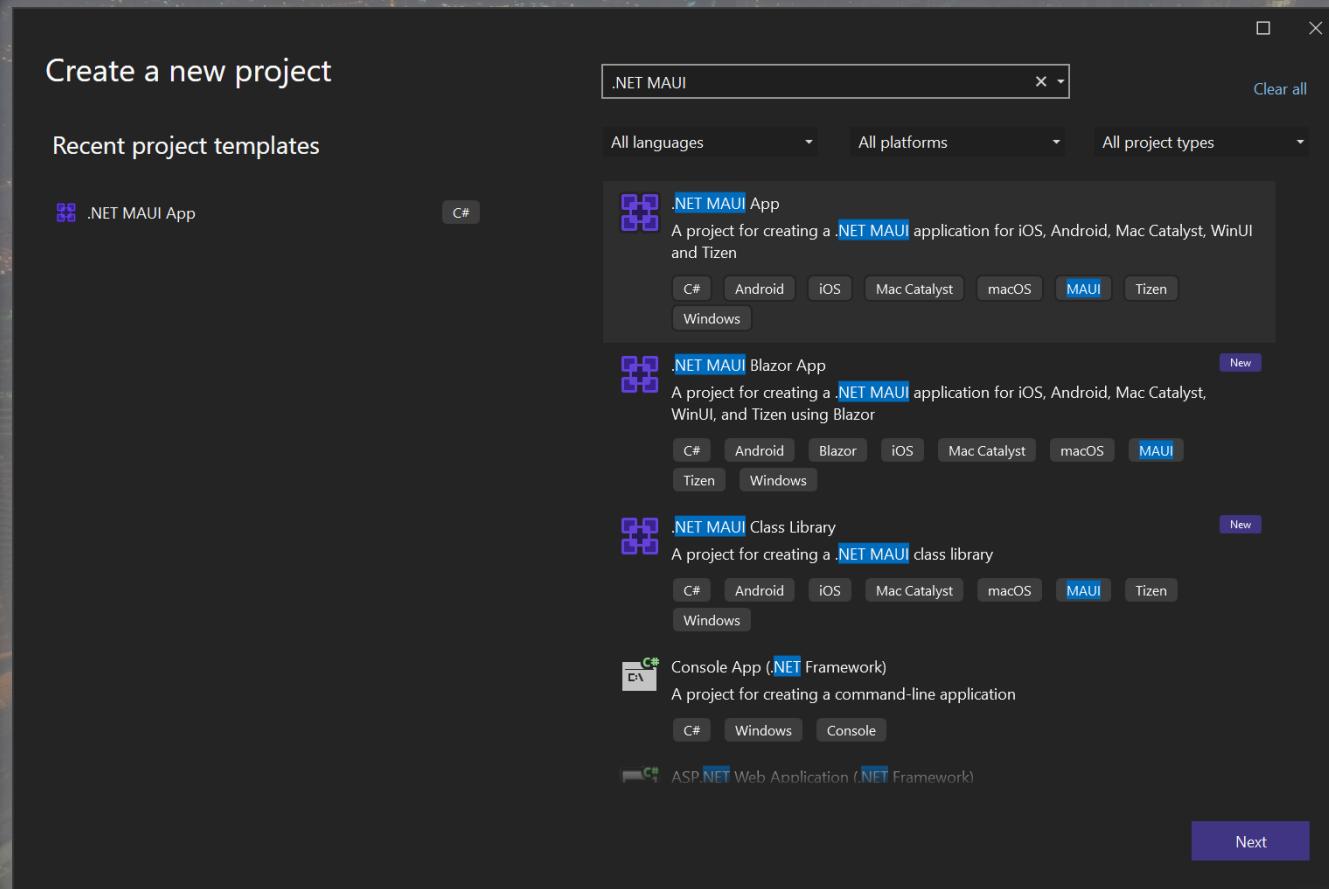
Optional

- IntelliCode
- Xamarin

Individual components



@mallibone



@mallibone

File Edit View Git Project Build Debug XML Test Analyze Tools Extensions Window Help Search **MauiApp1**

Live Share Notifications

Toolbox SQL Server Object Explorer

MauiApp1.csproj

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net6.0-android;net6.0-ios;net6.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks Condition="\"$([MSBuild]::IsOSPlatform('windows'))\">$([TargetFrameworks]);net6.0-windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$([TargetFrameworks]);net6.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>MauiApp1</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>MauiApp1</ApplicationTitle>

    <!-- App Identifier -->
    <ApplicationId>com.companyname.mauiapp1</ApplicationId>
    <ApplicationIdGuid>2128AE37-4BEC-46F8-B35E-B17F617BB30D</ApplicationIdGuid>

    <!-- Versions -->
    <ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>
    <ApplicationVersion>1</ApplicationVersion>

    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'ios'\">14.2</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'maccatalyst'\">14.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'android'\">21.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'windows'\">10.0.17763.0</SupportedOSPlatformVersion>
    <TargetPlatformMinVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'windows'\">10.0.17763.0</TargetPlatformMinVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'tizen'\">6.5</SupportedOSPlatformVersion>
  </PropertyGroup>

  <ItemGroup>
    <!-- App Icon -->
    <MauiIcon Include="Resources\Icon\appicon.svg" ForegroundFile="Resources\Icon\appiconfg.svg" Color="#512BD4" />

    <!-- Splash Screen -->
    <MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />

    <!-- Images -->
    <MauiImage Include="Resources\Images\*" />
    <MauiImage Update="Resources\Images\dotnet_bot.svg" BaseSize="168,208" />

    <!-- Custom Fonts -->
    <MauiFont Include="Resources\Fonts\*" />

    <!-- Raw Assets (also remove the "Resources\Raw" prefix) -->
    <MauiAsset Include="Resources\Raw\**" LogicalName="%{RecursiveDir}%(Filename)%{Extension}" />
  </ItemGroup>

```

100 % No issues found Ln: 1 Ch: 1 TABS CRLF

Error List Output

Ready Add to Source Control Select Repository

Solution Explorer

- Solution 'MauiApp1' (1 of 1 project)
 - MauiApp1
 - Dependencies
 - Properties
 - Platforms
 - Android
 - iOS
 - MacCatalyst
 - Tizen
 - Windows
 - Resources
 - App.xaml
 - AppShell.xaml
 - MainPage.xaml
 - C# MauiProgram.cs

Properties

XML Document

Misc Encoding Unicode (UTF-8)

Output

Schemas C:\Program Files\Microsoft Visual S

Stylesheet

Encoding Character encoding of the document.

HOT RELOAD ALL THE THINGS



imgflip.com

@mallibone

File Edit View Git Project Build Debug XML Test Analyze Tools Extensions Window Help Search **MauiApp1**

Live Share Notifications

Toolbox SQL Server Object Explorer

MauiApp1.csproj

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net6.0-android;net6.0-ios;net6.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks Condition="\"$([MSBuild]::IsOSPlatform('windows'))\">$([TargetFrameworks]);net6.0-windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$([TargetFrameworks]);net6.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>MauiApp1</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>MauiApp1</ApplicationTitle>

    <!-- App Identifier -->
    <ApplicationId>com.companyname.mauiapp1</ApplicationId>
    <ApplicationIdGuid>2128AE37-4BEC-46F8-B35E-B17F617BB30D</ApplicationIdGuid>

    <!-- Versions -->
    <ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>
    <ApplicationVersion>1</ApplicationVersion>

    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'ios'\">14.2</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'maccatalyst'\">14.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'android'\">21.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'windows'\">10.0.17763.0</SupportedOSPlatformVersion>
    <TargetPlatformMinVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'windows'\">10.0.17763.0</TargetPlatformMinVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'tizen'\">6.5</SupportedOSPlatformVersion>
  </PropertyGroup>

  <ItemGroup>
    <!-- App Icon -->
    <MauiIcon Include="Resources\Icon\appicon.svg" ForegroundFile="Resources\Icon\appiconfg.svg" Color="#512BD4" />

    <!-- Splash Screen -->
    <MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />

    <!-- Images -->
    <MauiImage Include="Resources\Images\*" />
    <MauiImage Update="Resources\Images\dotnet_bot.svg" BaseSize="168,208" />

    <!-- Custom Fonts -->
    <MauiFont Include="Resources\Fonts\*" />

    <!-- Raw Assets (also remove the "Resources\Raw" prefix) -->
    <MauiAsset Include="Resources\Raw\**" LogicalName="%(RecursiveDir)%(Filename)%(Extension)" />
  </ItemGroup>

```

100 % No issues found Ln: 1 Ch: 1 TABS CRLF

Error List Output

Ready Add to Source Control Select Repository

Solution Explorer

- Solution 'MauiApp1' (1 of 1 project)
 - MauiApp1
 - Dependencies
 - Properties
 - Platforms
 - Android
 - iOS
 - MacCatalyst
 - Tizen
 - Windows
 - Resources
 - App.xaml
 - AppShell.xaml
 - MainPage.xaml
 - C# MauiProgram.cs

Solution Explorer Git Changes

Properties

XML Document

Misc Encoding Unicode (UTF-8)

Output

Schemas C:\Program Files\Microsoft Visual S

Stylesheet

Encoding Character encoding of the document.

The screenshot shows the Visual Studio IDE interface with a dark theme. The top navigation bar includes 'Help', 'Search' (with a magnifying glass icon), and the project name 'MauiApp1'. Below the navigation bar is a toolbar with various icons for file operations like Open, Save, and Print.

The main code editor area displays C# code for a Maui application. The code includes target framework declarations:

```
Frameworks>
<TargetFrameworks>net6.0-windows10.0.19041.0</TargetFrameworks>
izen by following this: https://github.com/Samsung/Tizen.NET -->
cks> -->
```

Below this, there's a 'Guid' section:

```
Guid>
```

At the bottom of the code editor, there are additional target framework declarations:

```
entifier('$(TargetFramework)') == 'ios'">14.2</SupportedOSPlatformVersion>
entifier('$(TargetFramework)') == 'maccatalyst'">14.0</SupportedOSPlatformVersi
entifier('$(TargetFramework)') == 'android'">21.0</SupportedOSPlatformVersion>
entifier('$(TargetFramework)') == 'windows'">10.0.17763.0</SupportedOSPlatformV
```

The right side of the interface features the 'Solution Explorer' window. It shows a tree view of the project structure under the 'MauiApp1' node. A red box highlights the 'Platforms' folder, which contains sub-folders for 'Android', 'iOS', 'MacCatalyst', 'Tizen', and 'Windows'. Other files listed in the Solution Explorer include 'App.xaml', 'AppShell.xaml', ' MainPage.xaml', and 'MauiProgram.cs'.

The screenshot shows the Visual Studio IDE interface with the project 'MauiApp1.csproj' selected in the Solution Explorer. The code editor displays the project's configuration file, specifically the `<PropertyGroup>` section. A red box highlights the following configuration settings:

```
<PropertyGroup>
    <TargetFrameworks>net6.0-android;net6.0-ios;net6.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks Condition="<MSBuild>::IsOSPlatform('windows')</MSBuild>">$(TargetFrameworks);net6.0-windows10.0.19041.0
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/S
    <!-- <TargetFrameworks>$(TargetFrameworks);net6.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>MauiApp1</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>MauiApp1</ApplicationTitle>

    <!-- App Identifier -->
    <ApplicationId>com.companyname.mauiapp1</ApplicationId>
    <ApplicationIdGuid>2128AE37-4BEC-46F8-B35E-B17F617BB30D</ApplicationIdGuid>

    <!-- Versions -->
    <ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>
    <ApplicationVersion>1</ApplicationVersion>

    <SupportedOSPlatformVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'ios'">1
    <SupportedOSPlatformVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'maccata"
```

```
<TargetFrameworks Condition="$(MSBuild) : : IsOSPlatform('windows'))">$(TargetFrameworks);net6.0-windows10.0.19041.0
<!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/S
<!-- <TargetFrameworks>$(TargetFrameworks);net6.0-tizen</TargetFrameworks> -->
<OutputType>Exe</OutputType>
<RootNamespace>MauiApp1</RootNamespace>
<UseMaui>true</UseMaui>
<SingleProject>true</SingleProject>
<ImplicitUsings>enable</ImplicitUsings>

<!-- Display name -->
<ApplicationTitle>MauiApp1</ApplicationTitle>

<!-- App Identifier -->
<ApplicationId>com.companyname.mauiapp1</ApplicationId>
<ApplicationIdGuid>2128AE37-4BEC-46F8-B35E-B17F617BB30D</ApplicationIdGuid>

<!-- Versions -->
<ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>
<ApplicationVersion>1</ApplicationVersion>

<SupportedOSPlatformVersion Condition="$(MSBuild) : : GetTargetPlatformIdentifier('$(TargetFramework)') == 'ios'">1
<SupportedOSPlatformVersion Condition="$(MSBuild) : : GetTargetPlatformIdentifier('$(TargetFramework)') == 'macat</SupportedOSPlatformVersion>
<SupportedOSPlatformVersion Condition="$(MSBuild) : : GetTargetPlatformIdentifier('$(TargetFramework)') == 'android'">1
<SupportedOSPlatformVersion Condition="$(MSBuild) : : GetTargetPlatformIdentifier('$(TargetFramework)') == 'windows'">1
<TargetPlatformMinVersion Condition="$(MSBuild) : : GetTargetPlatformIdentifier('$(TargetFramework)') == 'windows'">10.0.19041.0
<SupportedOSPlatformVersion Condition="$(MSBuild) : : GetTargetPlatformIdentifier('$(TargetFramework)') == 'tizen'">1
</PropertyGroup>

<ItemGroup>
    <!-- App Icon -->
    <MauiIcon Include="Resources\AppIcon\appicon.svg" ForegroundFile="Resources\AppIcon\appiconfg.svg" Color="#512BD4" />

    <!-- Splash Screen -->
    <MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128 128" />

```

```
<ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>
<ApplicationVersion>1</ApplicationVersion>

<SupportedOSPlatformVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'ios'">1
<SupportedOSPlatformVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'macos'">
<SupportedOSPlatformVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'android'">
<SupportedOSPlatformVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'windows'">
<TargetPlatformMinVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'windows'">
<SupportedOSPlatformVersion Condition="<MSBuild>::GetTargetPlatformIdentifier('$(TargetFramework)') == 'tizen'">
</PropertyGroup>

<ItemGroup>
    <!-- App Icon -->
    <MauiIcon Include="Resources\AppIcon\appicon.svg" ForegroundFile="Resources\AppIcon\appiconfg.svg" Color="#512BD4" />

    <!-- Splash Screen -->
    <MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />

    <!-- Images -->
    <MauiImage Include="Resources\Images\*" />
    <MauiImage Update="Resources\Images\dotnet_bot.svg" BaseSize="168,208" />

    <!-- Custom Fonts -->
    <MauiFont Include="Resources\Fonts\*" />

    <!-- Raw Assets (also remove the "Resources\Raw" prefix) -->
    <MauiAsset Include="Resources\Raw\**" LogicalName="%(RecursiveDir)%(Filename)%(Extension)" />
</ItemGroup>
```

100 %

No issues found

Error List Output

Ready

The screenshot shows the Visual Studio IDE interface with the project file `MauiApp1.csproj` open. The code editor displays the XML configuration for the project. A red box highlights the `<TargetFrameworks>` element, which specifies supported platforms: `net6.0-android; net6.0-ios; net6.0-maccatalyst`. The rest of the configuration includes settings for `OutputType`, `RootNamespace`, `UseMaui`, `SingleProject`, `ImplicitUsings`, `ApplicationTitle`, `ApplicationId`, `ApplicationIdGuid`, `ApplicationDisplayVersion`, `ApplicationVersion`, and platform-specific supported versions.

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net6.0-android;net6.0-ios;net6.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks Condition="&lt;[MSBuild]::GetOSPlatform('windows')&gt;&lt;TargetFrameworks&gt;$(TargetFrameworks);net6.0-windows10.0.19041.0</TargetFrameworks>">&lt;!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/S-->
    <!-- <TargetFrameworks>$(TargetFrameworks);net6.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>MauiApp1</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>MauiApp1</ApplicationTitle>

    <!-- App Identifier -->
    <ApplicationId>com.companyname.mauiapp1</ApplicationId>
    <ApplicationIdGuid>2128AE37-4BEC-46F8-B35E-B17F617BB30D</ApplicationIdGuid>

    <!-- Versions -->
    <ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>
    <ApplicationVersion>1</ApplicationVersion>

    <SupportedOSPlatformVersion Condition="&lt;[MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework'))&gt;= 'ios'">1
    <SupportedOSPlatformVersion Condition="&lt;[MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework'))&gt;= 'maccatalyst'">1
    <SupportedOSPlatformVersion Condition="&lt;[MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework'))&gt;= 'android'">1
    <SupportedOSPlatformVersion Condition="&lt;[MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework'))&gt;= 'windows'">1
  </PropertyGroup>

```

```
private Task SetCurrentTime(DateTime time) {
    return _currentTimeSource.OnChangeCurrentTime(new ChangeEventArgs(time));
}

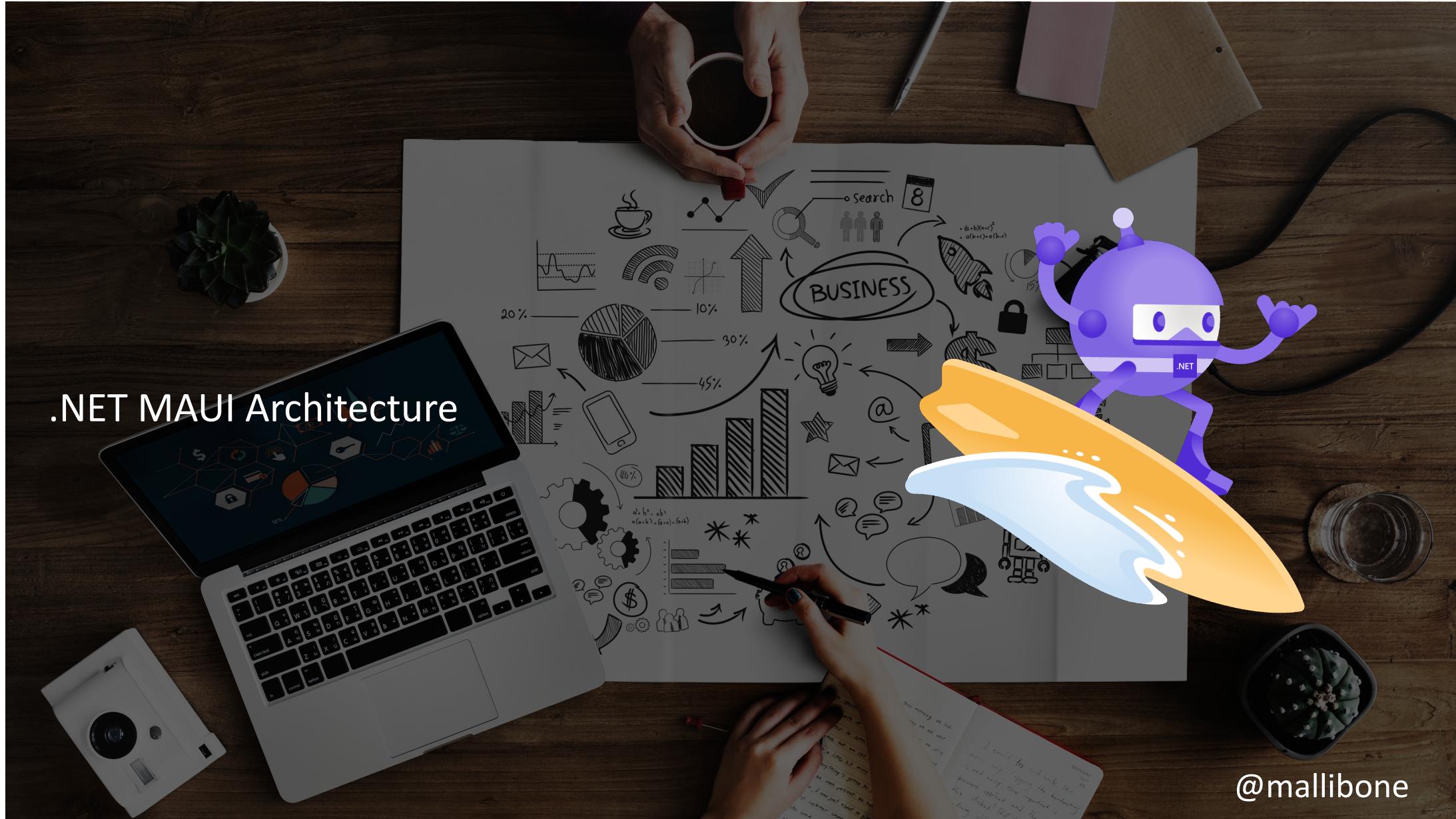
private async Task ChangeVolume(ChangeEventArgs e) {
    if (e.Value != null && int.TryParse(e.Value.ToString(), out var volume)) {
        Volume = volume;
        await OnVolumeChange.InvokeAsync(volume);
    }
}

private void OnInputCurrentTime(ChangeEventArgs e) {
    if (e.Value != null && int.TryParse(e.Value.ToString(), out var time)) {
        seekingTime = time;
    }
}

private async Task OnChangeCurrentTime(ChangeEventArgs e) {
    if (e.Value != null && int.TryParse(e.Value.ToString(), out var time)) {
        await SetCurrentTime(time);
    }
    seekingTime = null;
}
```

@mallibone

.NET MAUI Architecture



@mallibone

.NET MAUI Architecture

UI Framework

.NET MAUI XAML

Blazor .NET MAUI

MauiReactor

Fabulous

Framework

.NET MAUI

Platform

iOS

Android

Windows

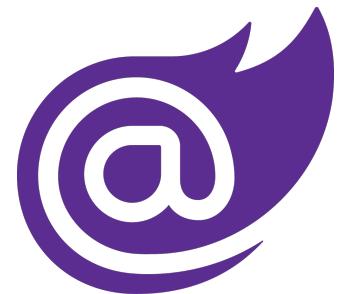
macOS

Tizen

Web?

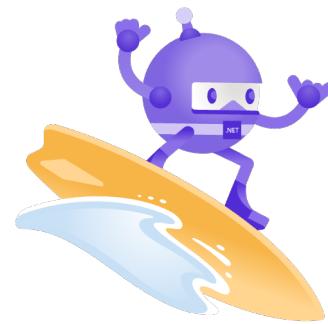
@mallibone





Blazor

+

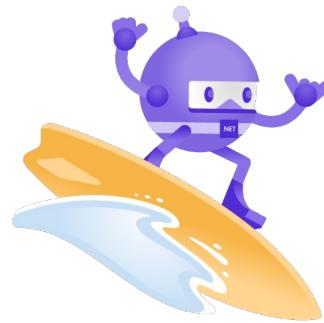


.NET MAUI



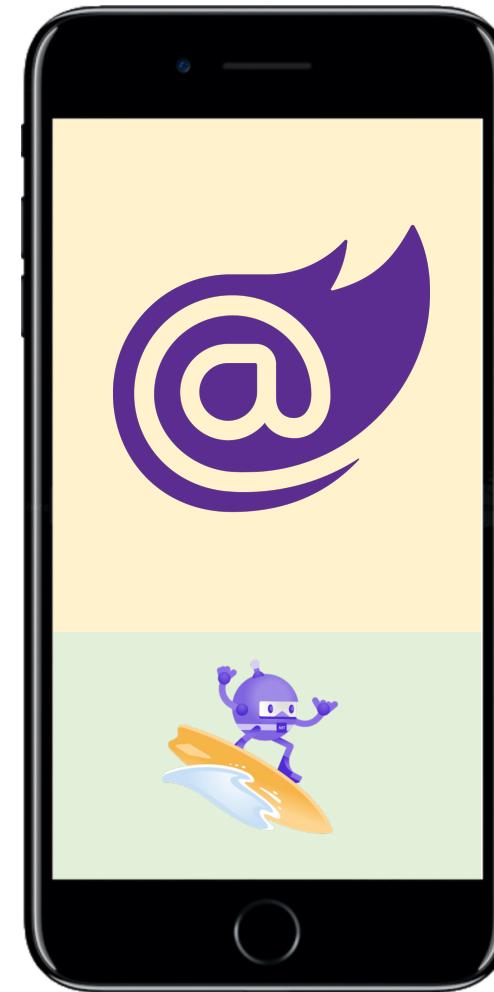
Blazor

+



.NET MAUI

=



.NET MAUI Blazor

File Edit View Git Project Build Debug XML Test Analyze Tools Extensions Window Help Search **MauiApp1**

Live Share Notifications

Toolbox SQL Server Object Explorer

MauiApp1.csproj

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net6.0-android;net6.0-ios;net6.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks Condition="\"$([MSBuild]::IsOSPlatform('windows'))\">$([TargetFrameworks]);net6.0-windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$([TargetFrameworks]);net6.0-tizen</TargetFrameworks> -->
    <OutputType>Exe</OutputType>
    <RootNamespace>MauiApp1</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>enable</ImplicitUsings>

    <!-- Display name -->
    <ApplicationTitle>MauiApp1</ApplicationTitle>

    <!-- App Identifier -->
    <ApplicationId>com.companyname.mauiapp1</ApplicationId>
    <ApplicationIdGuid>2128AE37-4BEC-46F8-B35E-B17F617BB30D</ApplicationIdGuid>

    <!-- Versions -->
    <ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>
    <ApplicationVersion>1</ApplicationVersion>

    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'ios'\">14.2</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'maccatalyst'\">14.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'android'\">21.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'windows'\">10.0.17763.0</SupportedOSPlatformVersion>
    <TargetPlatformMinVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'windows'\">10.0.17763.0</TargetPlatformMinVersion>
    <SupportedOSPlatformVersion Condition="\"$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')\" == 'tizen'\">6.5</SupportedOSPlatformVersion>
  </PropertyGroup>

  <ItemGroup>
    <!-- App Icon -->
    <MauiIcon Include="Resources\Icon\appicon.svg" ForegroundFile="Resources\Icon\appiconfg.svg" Color="#512BD4" />

    <!-- Splash Screen -->
    <MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />

    <!-- Images -->
    <MauiImage Include="Resources\Images\*" />
    <MauiImage Update="Resources\Images\dotnet_bot.svg" BaseSize="168,208" />

    <!-- Custom Fonts -->
    <MauiFont Include="Resources\Fonts\*" />

    <!-- Raw Assets (also remove the "Resources\Raw" prefix) -->
    <MauiAsset Include="Resources\Raw\**" LogicalName="%{RecursiveDir}%(Filename)%{Extension}" />
  </ItemGroup>

```

100 % No issues found Ln: 1 Ch: 1 TABS CRLF

Error List Output

Ready Add to Source Control Select Repository

Solution Explorer

- Solution 'MauiApp1' (1 of 1 project)
 - MauiApp1
 - Dependencies
 - Properties
 - Platforms
 - Android
 - iOS
 - MacCatalyst
 - Tizen
 - Windows
 - Resources
 - App.xaml
 - AppShell.xaml
 - MainPage.xaml
 - C# MauiProgram.cs

Solution Explorer Git Changes

Properties XML Document

Misc Encoding Unicode (UTF-8)

Output Schemas "C:\Program Files\Microsoft Visual S Stylesheet

Encoding Character encoding of the document.

Demo Review

BlazorWebView for creating Hybrid .NET MAUI Apps

Embed and host Blazor websites in your app

Share state between your apps

@mallibone

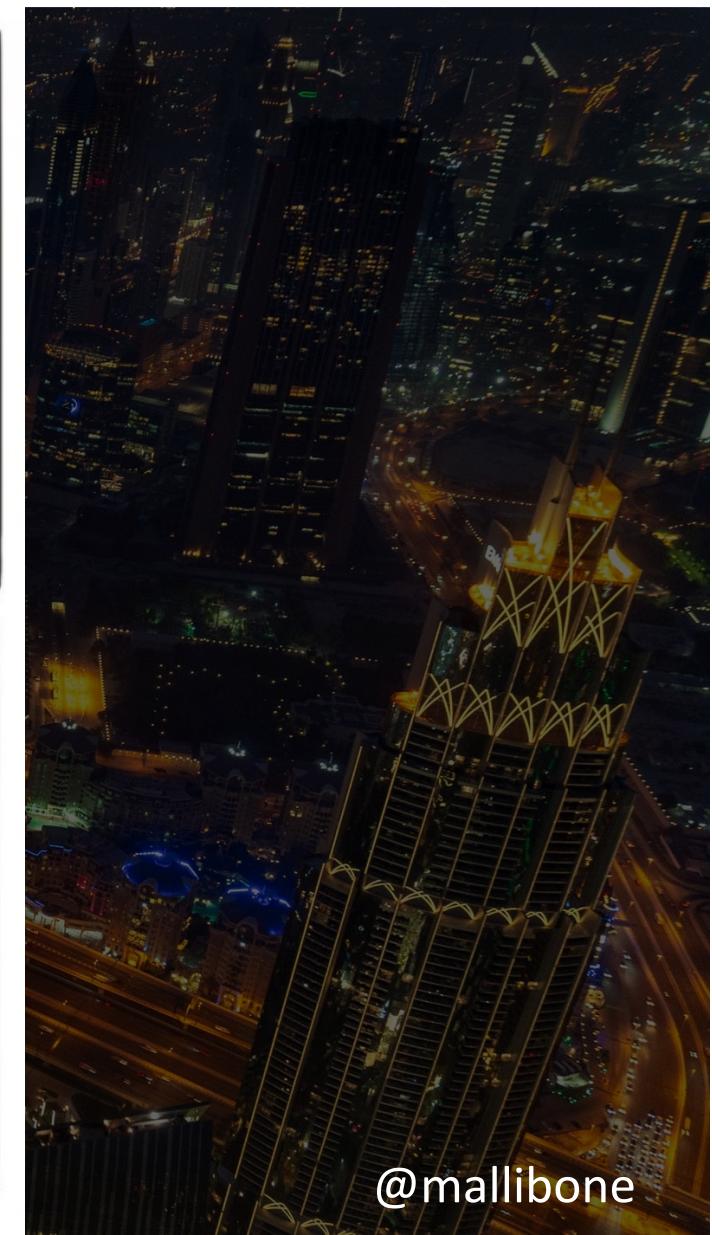
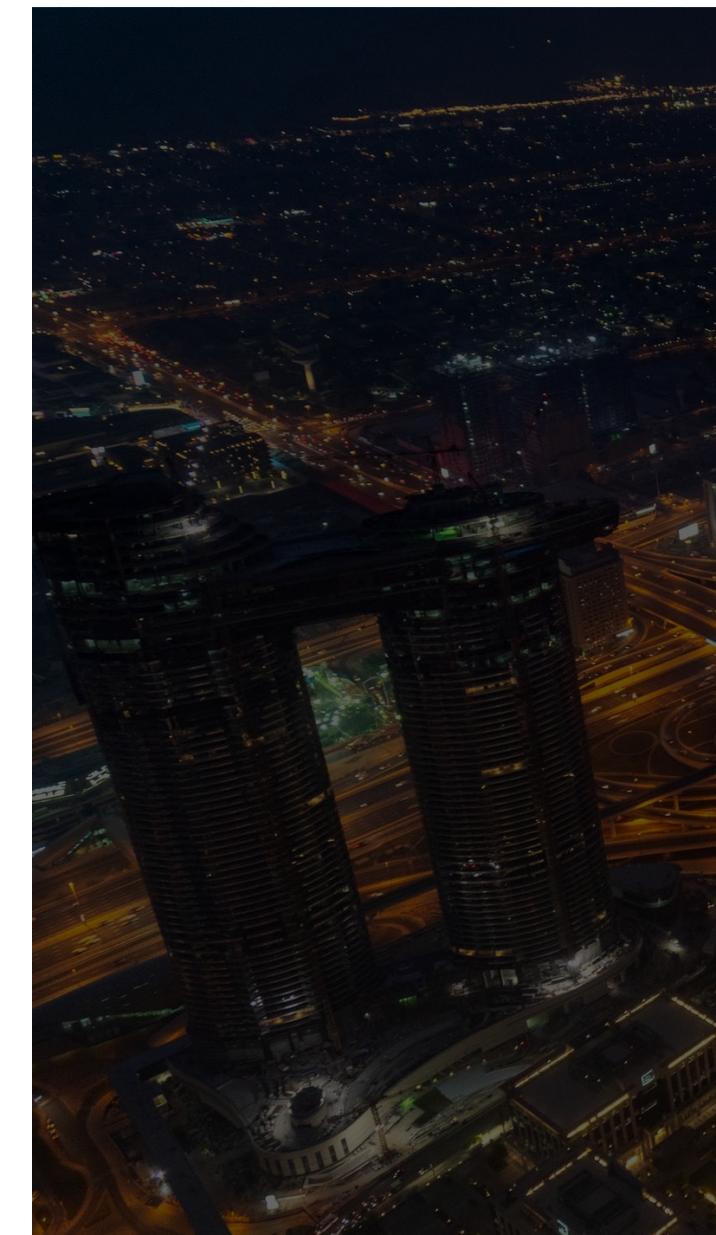


HEY CUPCAKE

**WE WILL START A MAJOR
MOBILE PROJECT TOMORROW**

imgflip.com

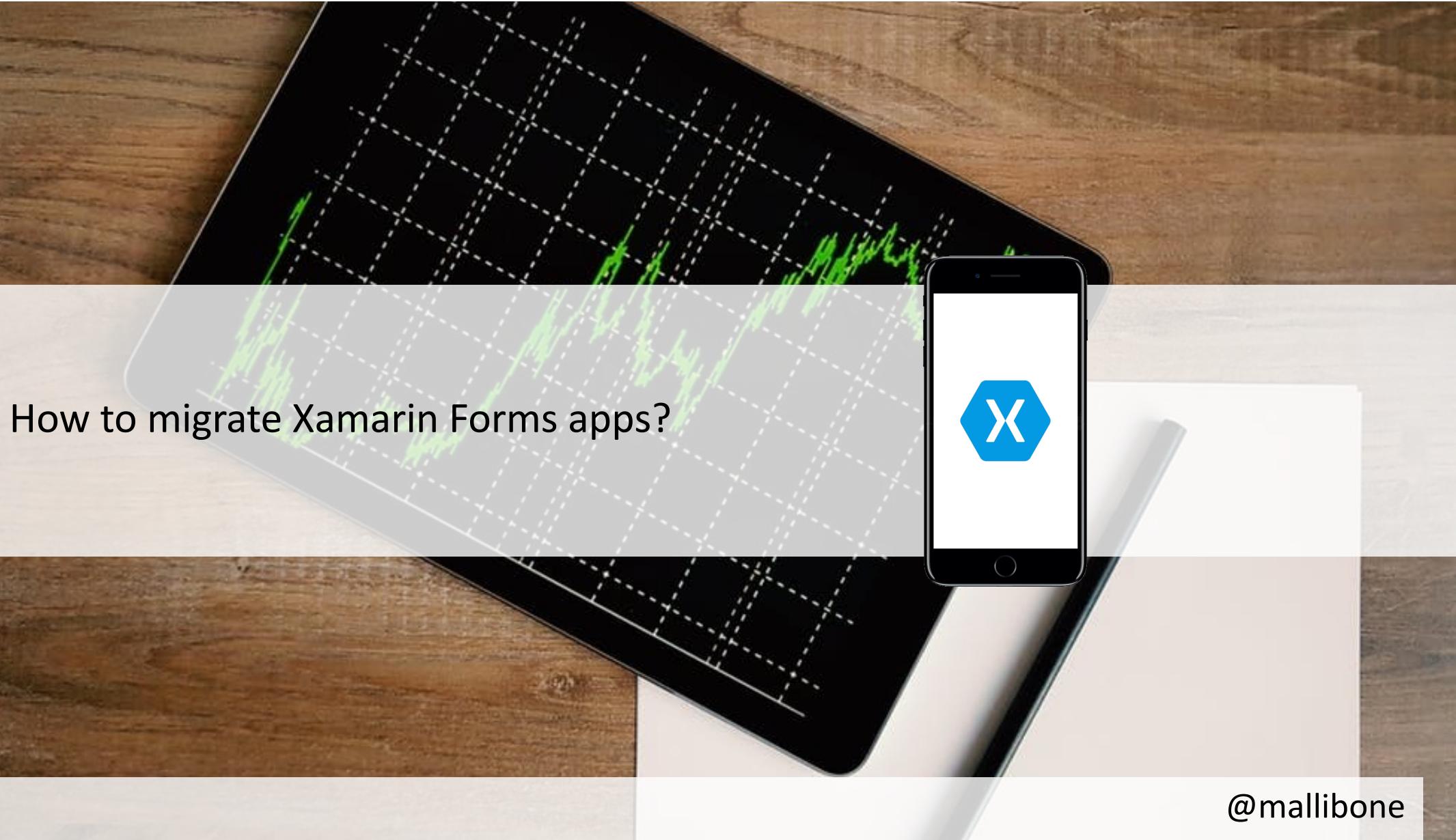
@mallibone



@mallibone

imgflip.com

JAKE-CLARK.TUMBLR



How to migrate Xamarin Forms apps?

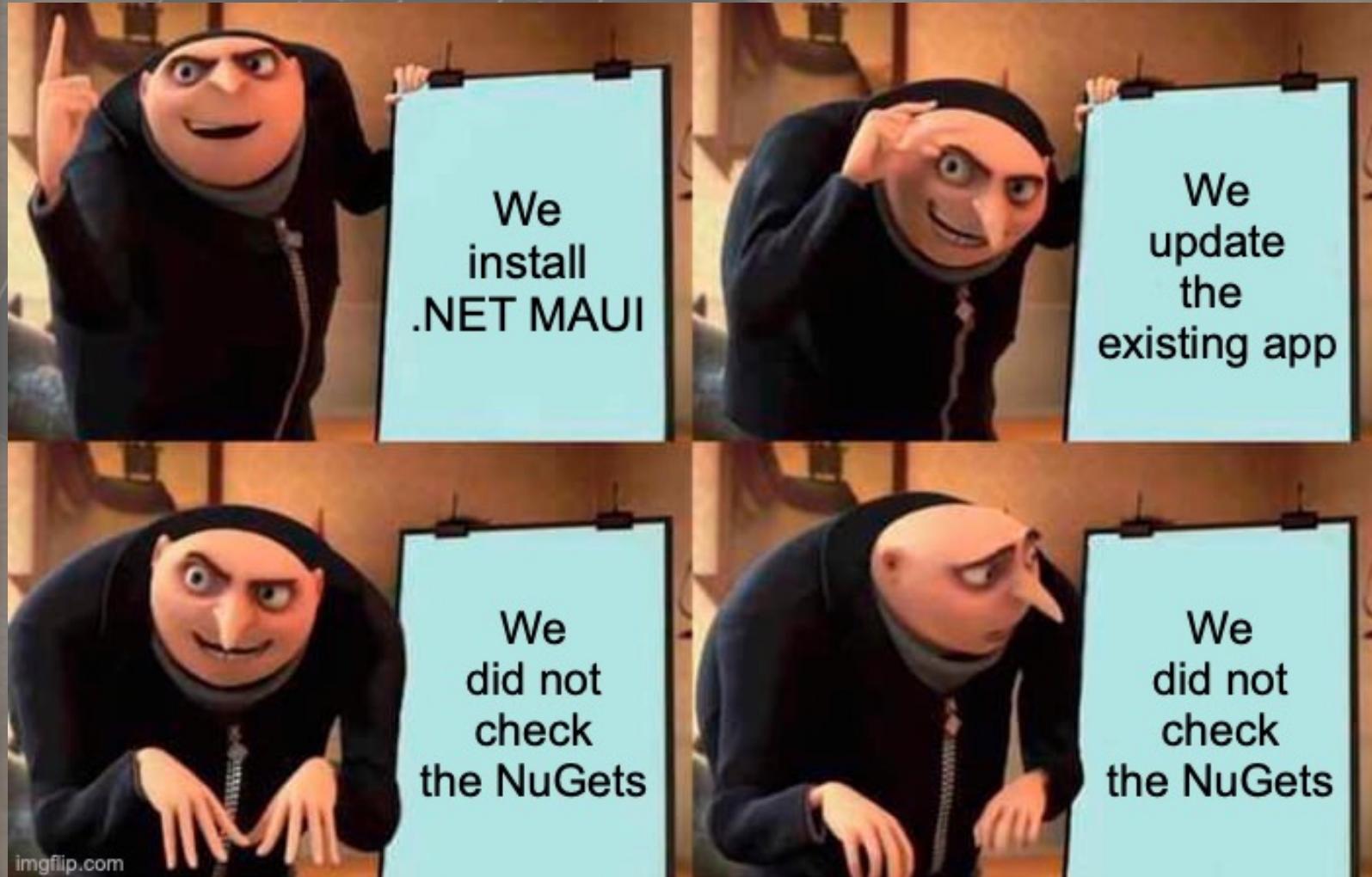
@mallibone

Xamarin Forms to .NET MAUI

Migration Steps

- Remove Xamarin Forms NuGet package
- Add .NET MAUI workload
 - Enable MAUI in your .csproj:
`<UseMaui>true</UseMaui>`
- Update Namespaces:
`using Xamarin.Forms ➔ using Microsoft.Maui`
- Use the dotnet upgrade assistant (Preview)
 - iOS and Android Only

@mallibone



@mallibone

.NET MAUI Roadmap

- Monthly Updates .NET
- Migration Path Xamarin Forms
- UI Rendering Performance
- Dev Experience ❤️

Roadmap

David Ortinau edited this page on Nov 22, 2022 · 39 revisions

.NET MAUI 7 Servicing updates

.NET MAUI will be serviced on the [.NET 7 monthly cadence](#). You can check releases in the [.NET MAUI SDK Ongoing](#) project.

.NET 8 (November 2022 - November 2023)

.NET MAUI stories for .NET 8 include:

- Improve upgrade path from Xamarin to .NET
 - SDK quality
 - .NET Upgrade Assistant
 - Guides and documentation
- Speed up UI rendering
- Reduce the developer inner-loop time
 - Build time improvements
 - Hot reload improvements

Takeaways



Takeaways

- .NET Multi-platform Appl UI
- Improved Architecture with new Handlers, closer to .NET, Single Project
- Dev productivity boost with Hot Reload



Thank you for your time!



Mark Allibone



@mallibone



Rey Technology



<https://mallibone.com>



<https://nullpointers.io>



<https://github.com/dotnet/maui>



<https://bit.ly/3vGrM9C>



Microsoft®
Most Valuable
Professional

