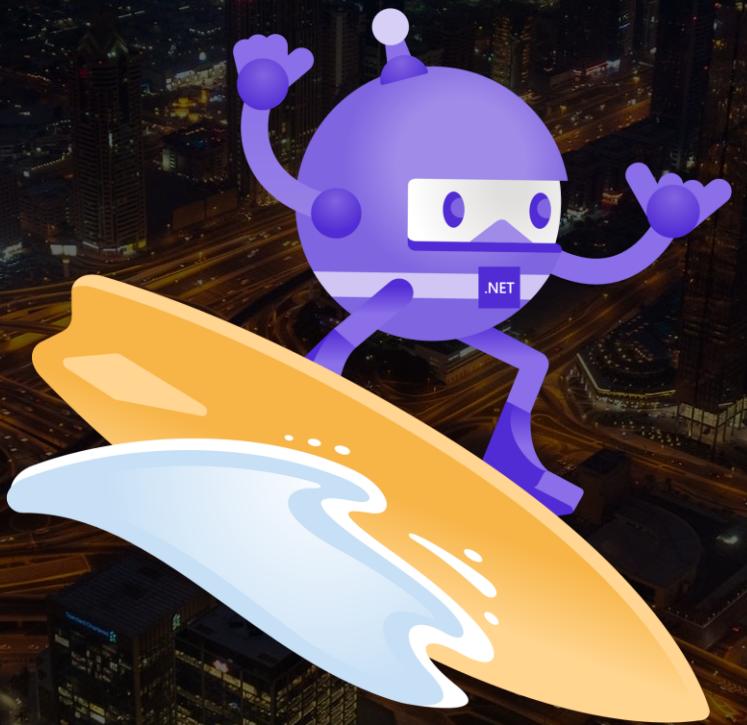


Creating cross-platform applications with .NET Maui

Mark Allibone
Rey Technology

@mallibone



@mallibone



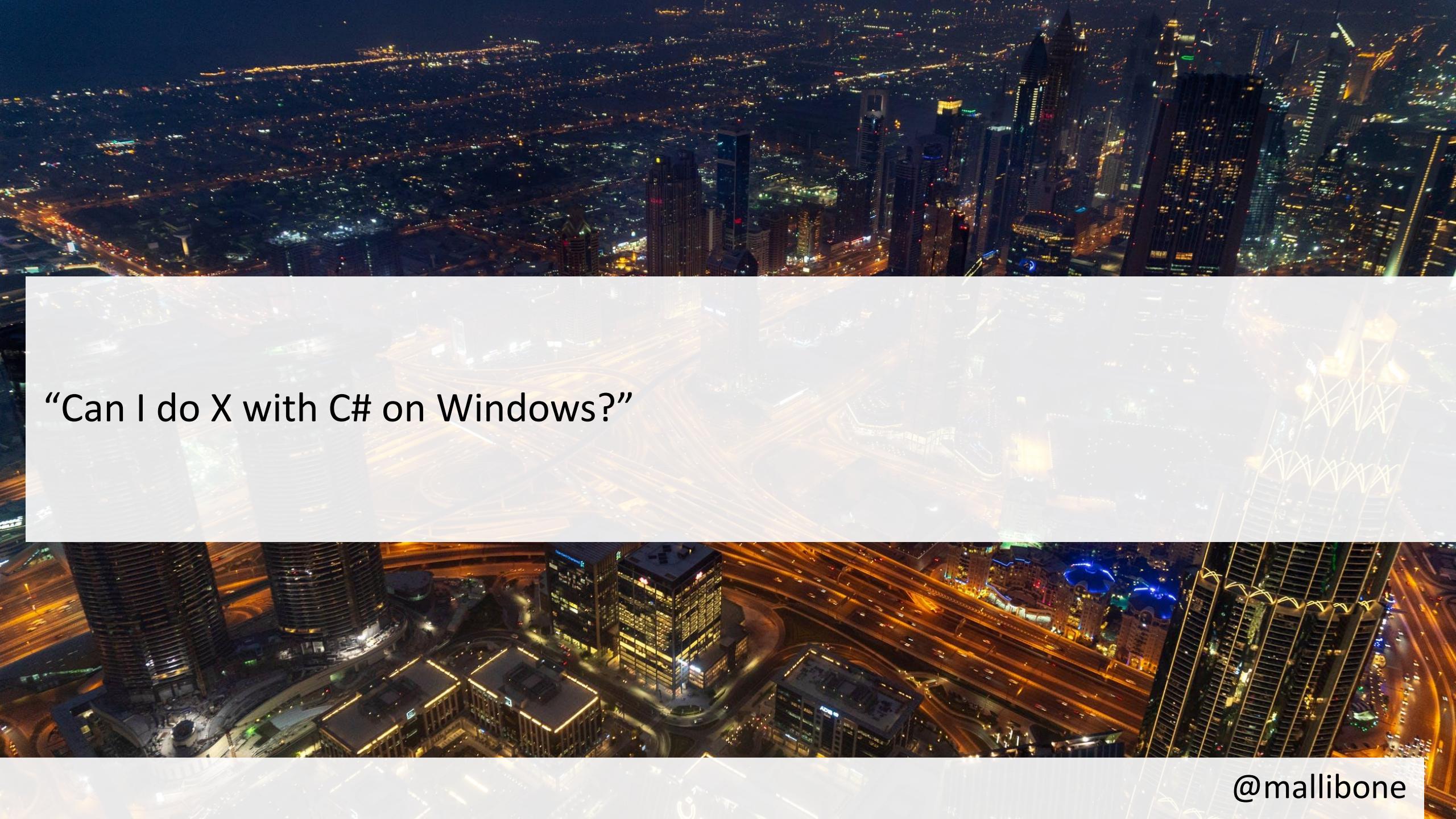
.NET MAUI the evolution of Xamarin Forms



@mallibone

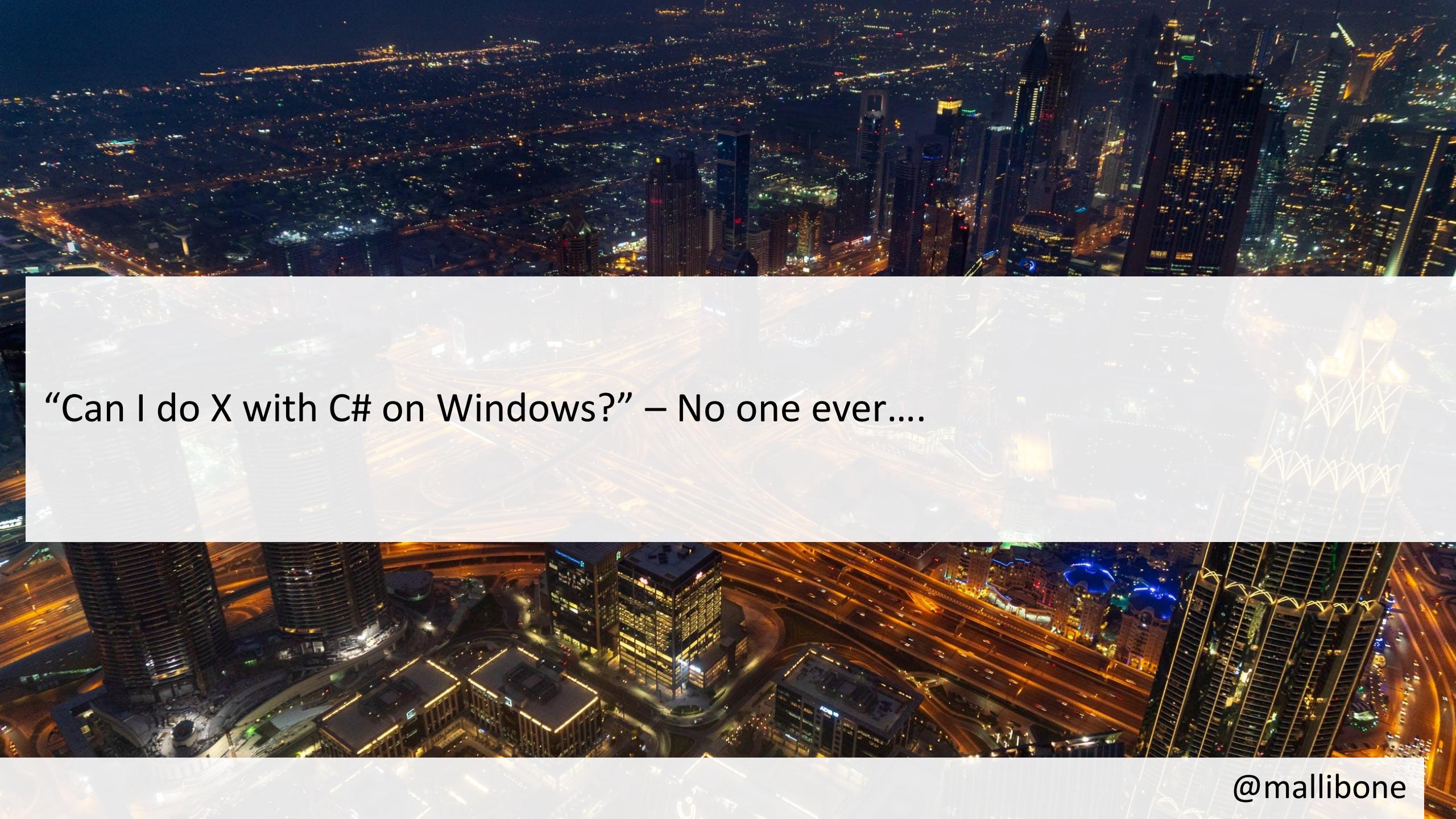


@mallibone

The background image shows a wide-angle aerial view of a city at night, likely Dubai, with numerous skyscrapers and a complex network of illuminated highways. The city lights create a dense pattern of yellow and white against the dark sky.

“Can I do X with C# on Windows?”

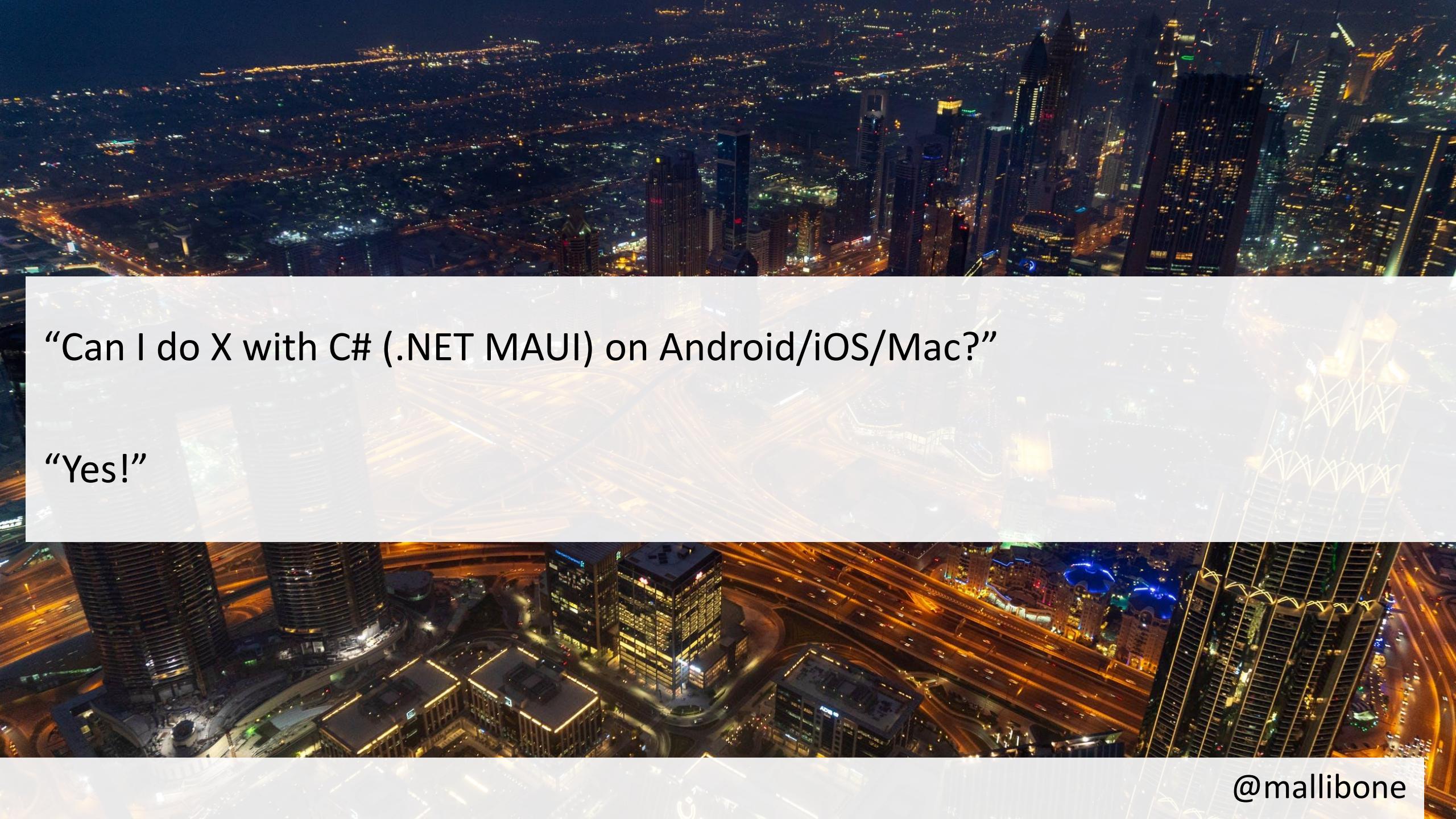
@mallibone



“Can I do X with C# on Windows?” – No one ever....

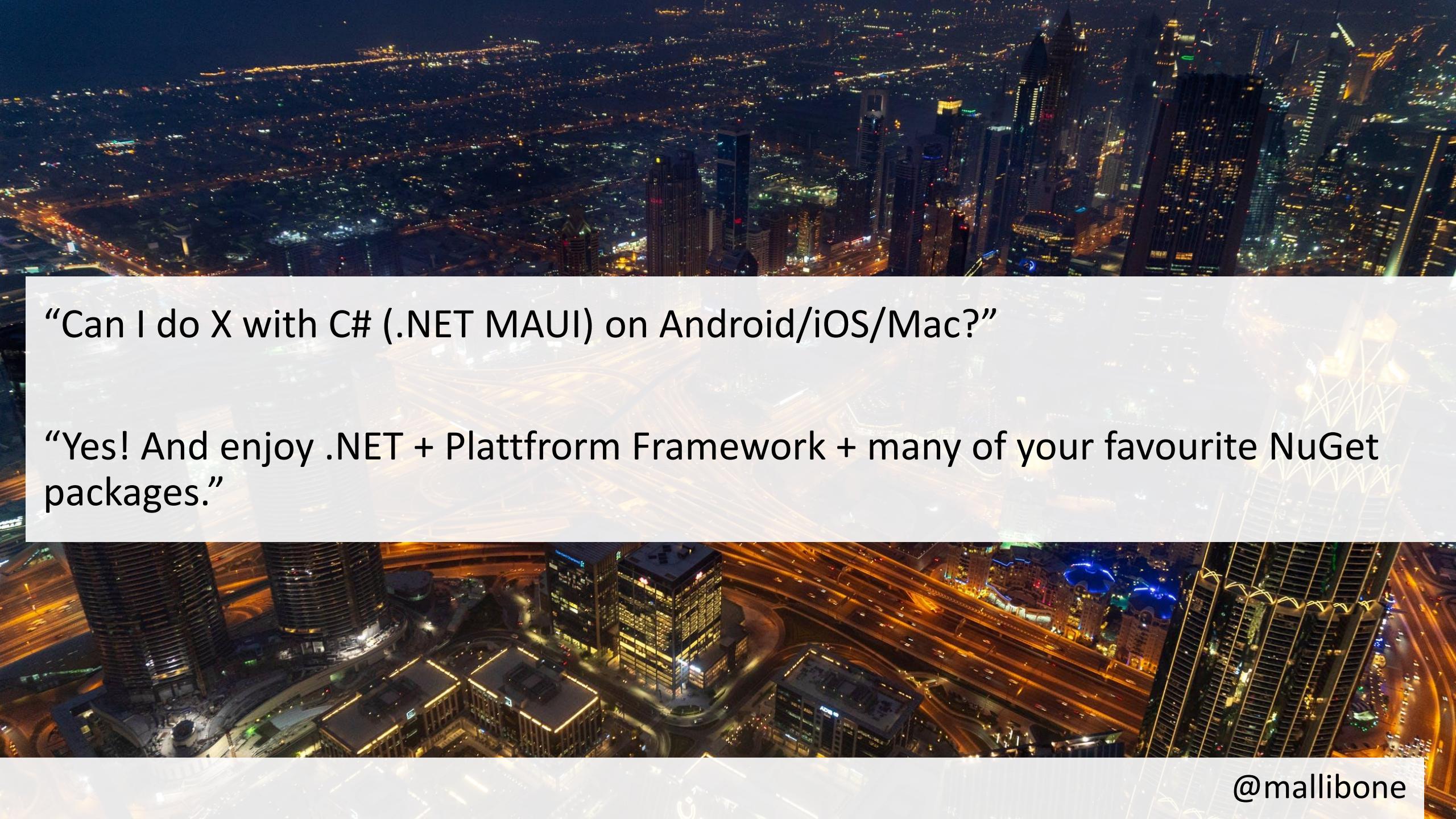


“Can I do X with C# (.NET MAUI) on Android/iOS/Mac?”

The background image shows a panoramic aerial view of a city at night, likely Dubai, with numerous skyscrapers and a dense network of illuminated roads.

“Can I do X with C# (.NET MAUI) on Android/iOS/Mac?”

“Yes!”



“Can I do X with C# (.NET MAUI) on Android/iOS/Mac?”

“Yes! And enjoy .NET + Plattfrom Framework + many of your favourite NuGet packages.”



Dive into the Experience



@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...
- Python development**
Editing, debugging, interactive development and source control for Python.
- Azure development**
Azure SDKs, tools, and projects for developing cloud apps and creating resources using .NET and .NET Framework...
- 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.
- Desktop development with C++**
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.
- .NET desktop development**
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...
- Universal Windows Platform development**
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

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

Total space required 0 B

Install while downloading ▾ Close

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.

INSTALLER VERSION 3.3.2 TO 1.4.1457

@mallbone

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 required

Install while downloading

Cloud

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

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.

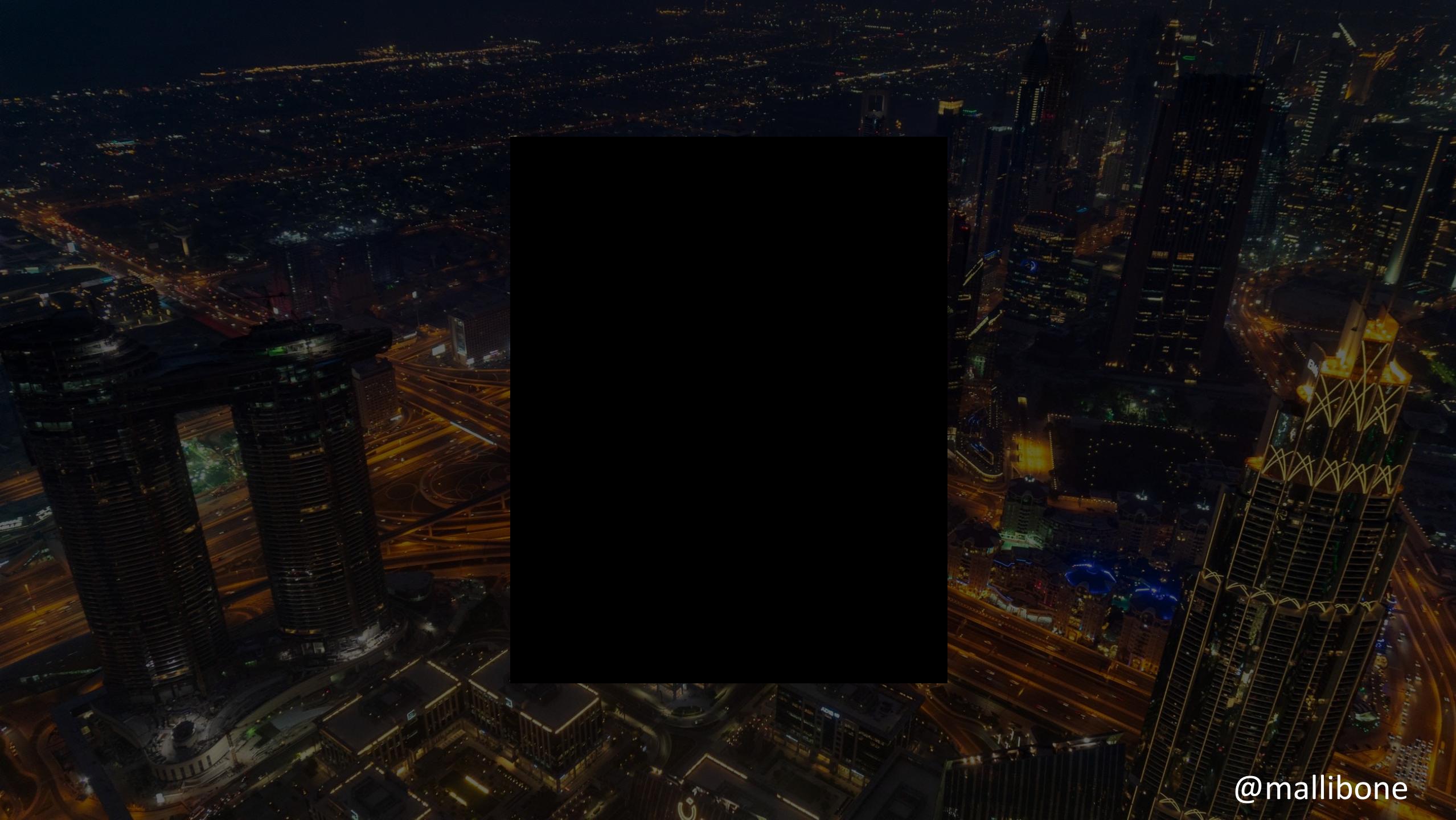
Total space required

Install while downloading

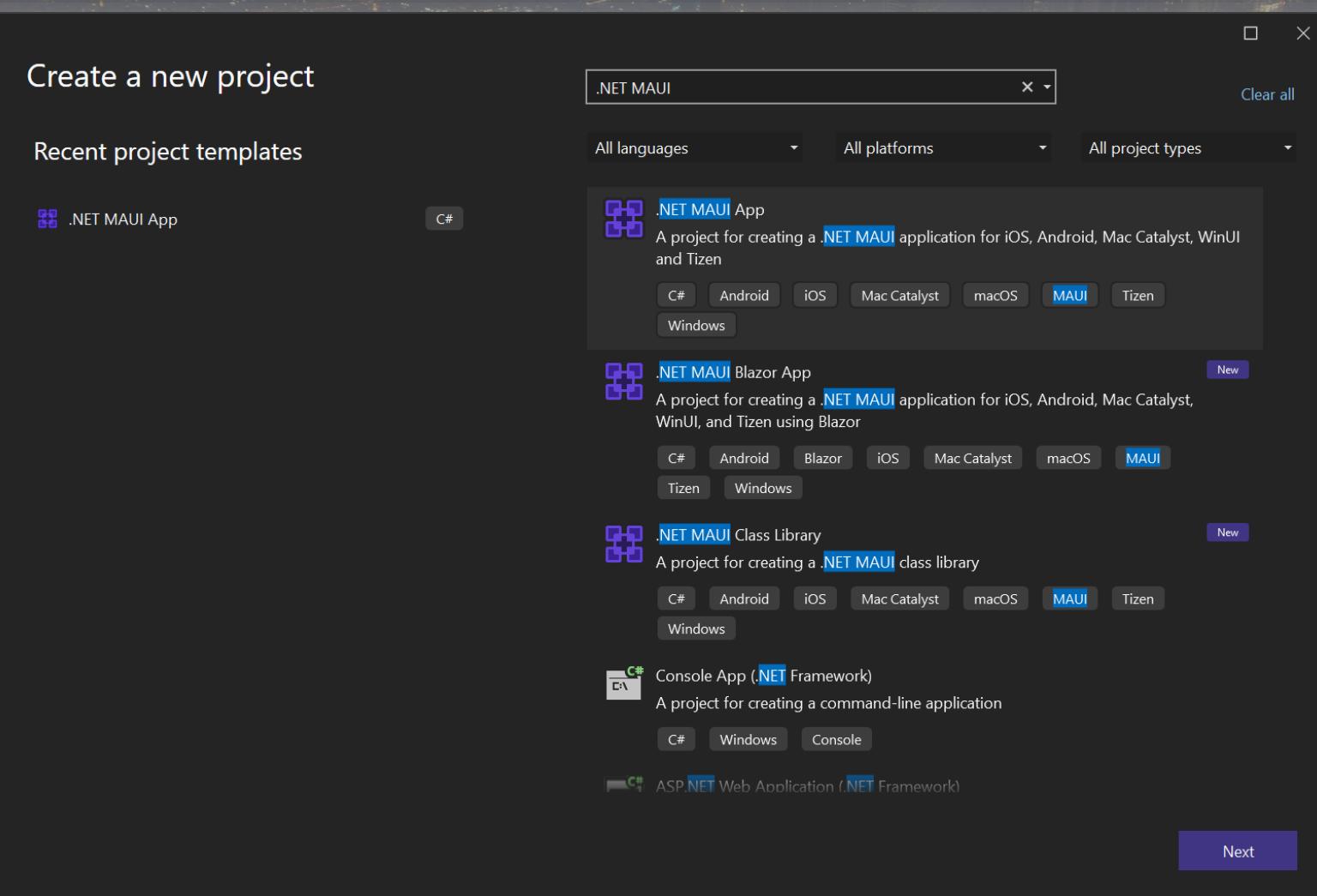
Cloud

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

Any CPU | Debug | Android Emulator | Live Share

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 | Error List | Output | Add to Source Control | Select Repository | Ready

Solution Explorer | Search Solution Explorer (Ctrl+U)

MauiApp1 | Dependencies | Properties | Platforms | Resources | App.xaml | AppShell.xaml | MainPage.xaml | MauiProgram.cs

Properties | XML Document | Misc | Encoding: Unicode (UTF-8) | Output: C:\Program Files\Microsoft Visual 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

Any CPU | Debug | Android Emulator | Live Share

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>
```

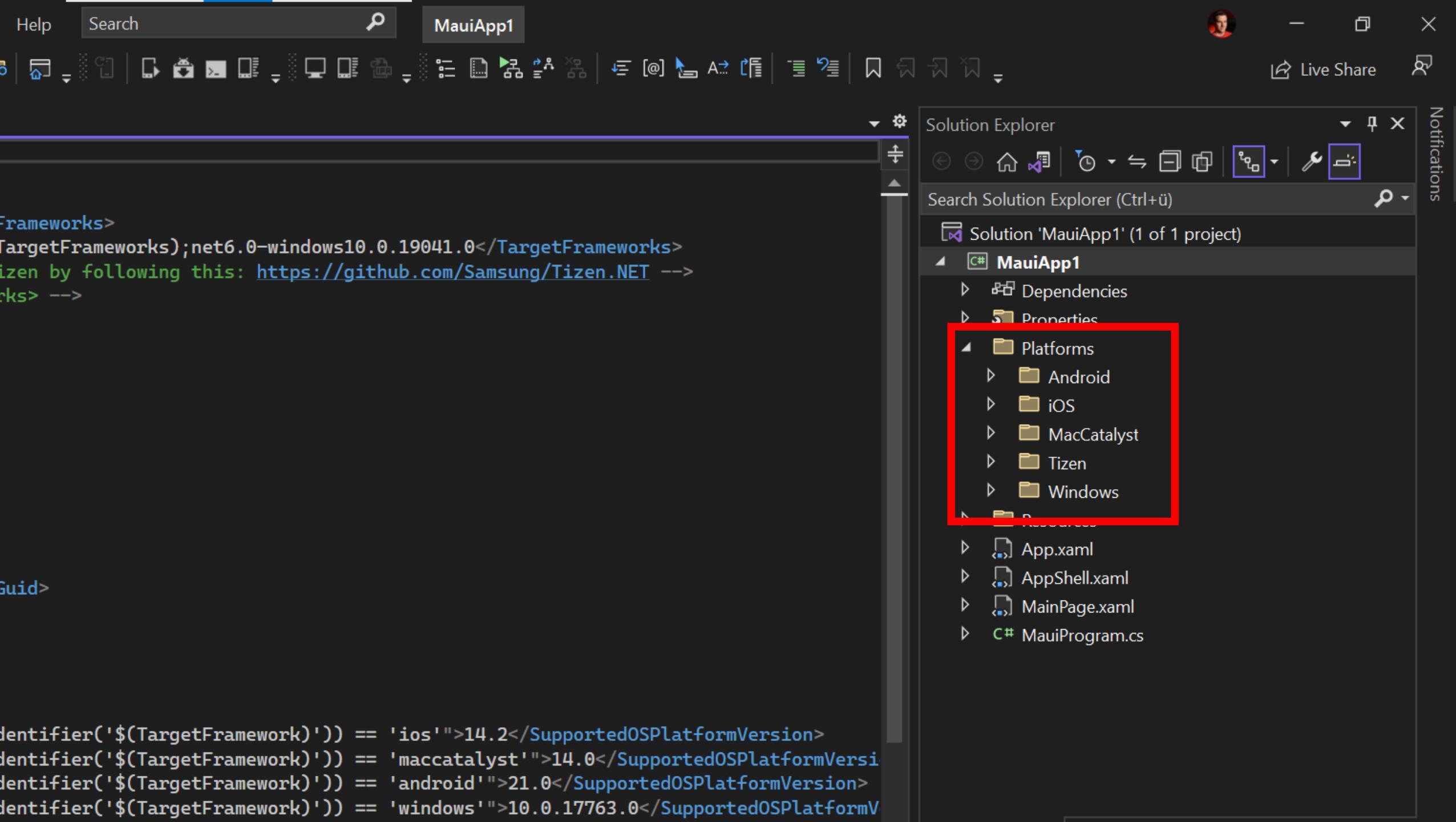
100 % | No issues found | Error List | Output | Add to Source Control | Select Repository | Ready

Solution Explorer | Search Solution Explorer (Ctrl+U)

MauiApp1 | Dependencies | Properties | Platforms | Resources | App.xaml | AppShell.xaml | MainPage.xaml | MauiProgram.cs

Properties | XML Document | Misc | Encoding: Unicode (UTF-8) | Output: C:\Program Files\Microsoft Visual Schemas: "C:\Program Files\Microsoft Visual S Stylesheet:

Encoding: Character encoding of the document.



The screenshot shows the Visual Studio IDE interface with the project file `MauiApp1.csproj` open in the editor. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, XML, Test, Analyze, Tools, Extensions, Window, Help, and a Search bar. Below the menu is a toolbar with various icons for file operations like Open, Save, and Build. The main editor area displays the XML code for the project file. A red box highlights the following section of the code:

```
<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>') == 'maccataly
```

```
<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)') == 'maccatalyst'">1
<SupportedOSPlatformVersion Condition="$(MSBuild)::GetTargetPlatformIdentifier('$(TargetFramework)') == 'android'">1
<SupportedOSPlatformVersion Condition="$(MSBuild)::GetTargetPlatformIdentifier('$(TargetFramework)') == 'windows'">1
<TargetPlatformMinVersion Condition="$(MSBuild)::GetTargetPlatformIdentifier('$(TargetFramework)') == 'windows'">1
<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" />
</ItemGroup>
```

```
targetFrameworks Condition="!${msbuild} || !${tizen}">windows_99>${targetFrameworks}, net6.0 windows10.0.19041.0, ${targetFrameworks}
comment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET-->
targetFrameworks>${TargetFrameworks};net6.0-tizen</TargetFrameworks> -->
true</UseMaui>

```

```
Icon -->
    <Icon Include="Resources\AppIcon\appicon.svg" ForegroundFile="Resources\AppIcon\appiconfg.svg" Color="#512BD4" />
    <Splash Screen -->
        <SplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />
```

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

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

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <TargetFrameworks>net6.0-android;net6.0-ios;net6.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks_Condition="&lt;([MSBuild]::GetOSPlatform('windows'))&gt;$(<TargetFrameworks>);net6.0-windows10.0.19041.0</TargetFrameworks_Condition>>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/S-->
    <!-- <TargetFrameworks>$(<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>

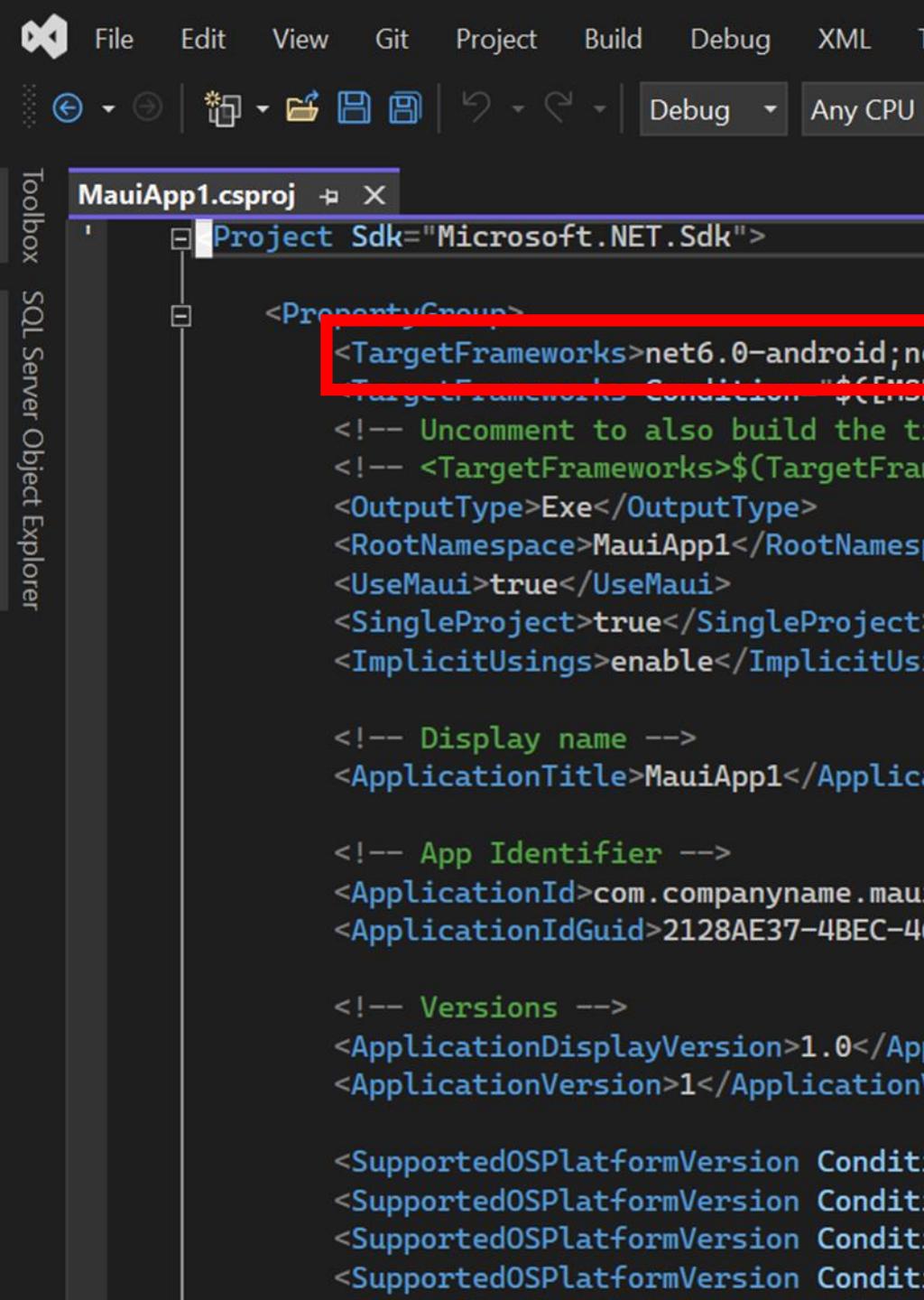
    <SupportedOSPlatformVersion Condition="&lt;([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')&gt;='ios')&gt;1</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition="&lt;([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')&gt;='maccataly-->
    <SupportedOSPlatformVersion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

Single Project

Multi Targeting

Access Platform Specific Features

- Partial Classes
 - Conditional Compilation e.g. #if ANDROID



The image shows two Microsoft Edge Dev browser windows side-by-side, displaying the same code editor interface. Both windows have a dark theme and show the same file, `FloatingPlayer.cs`, from a .NET project named `Podcast.Web`.

Code Snippet:

```
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;
    }
}
```

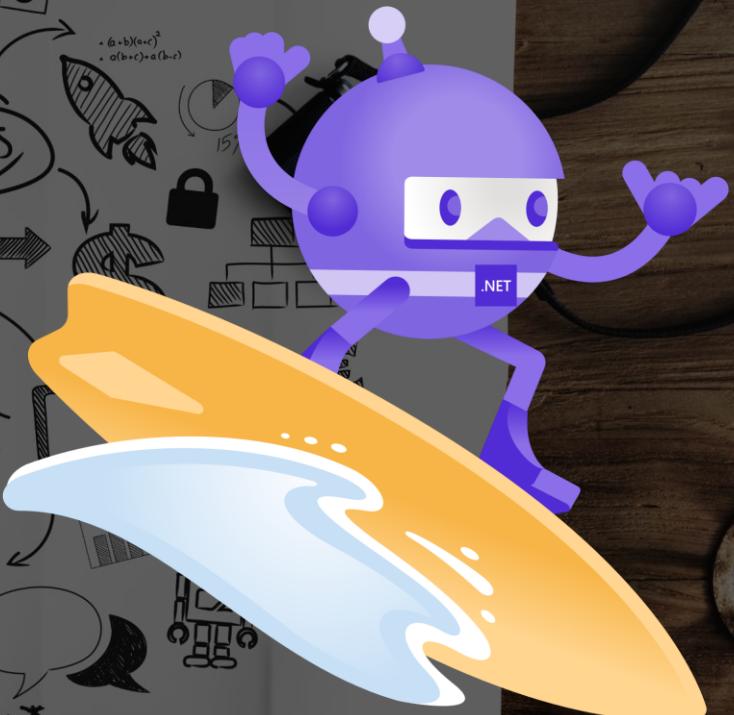
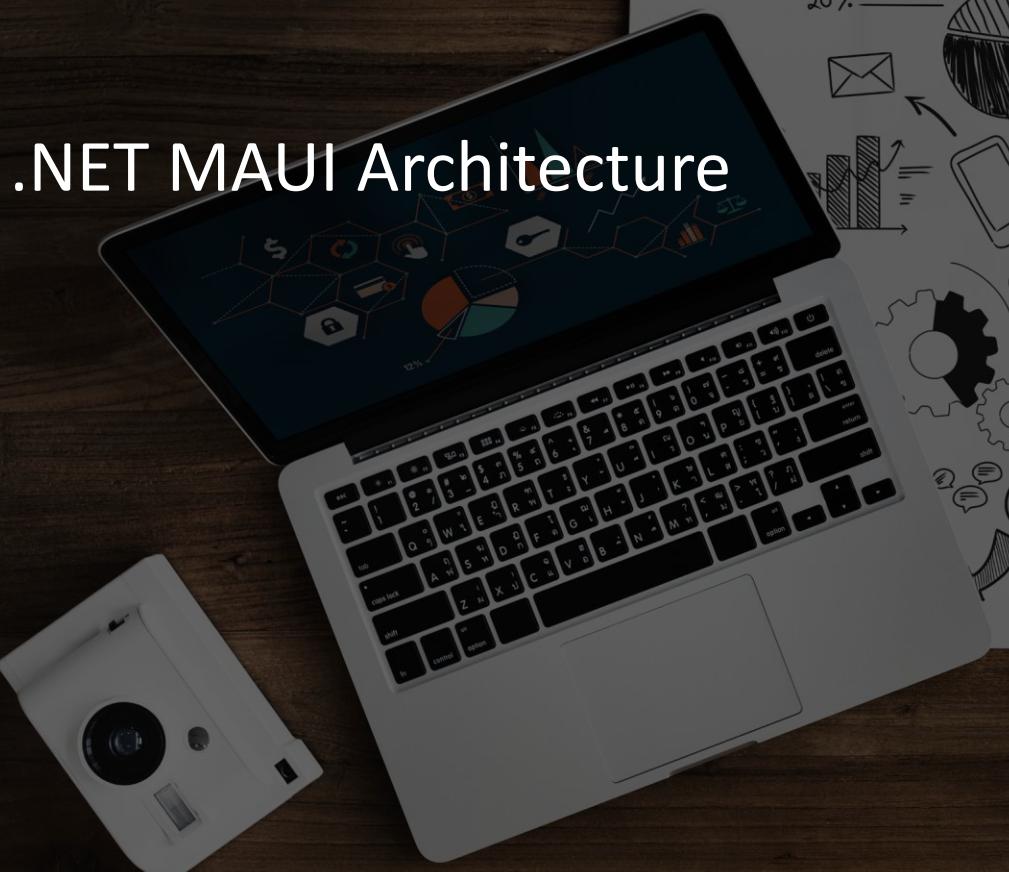
Solution Explorer:

The Solution Explorer shows the project structure:

- Podcast.Web (main)**
 - Podcast.Client**
 - Podcast.Components**
 - Connected Services
 - Dependencies - 2 updates
 - Events**
 - Extensions**
 - ListenTogether**
 - Player**
 - FloatingPlayer.razor**
 - FloatingPlayer.razor.cs
 - Shared**
 - wwwroot**
 - _Imports.razor**
 - AudioInterop.cs**
 - ClipboardInterop.cs**
 - ListenTogetherHubClient.cs**
 - LocalStorageInterop.cs**
 - ThemeInterop.cs**
- Podcast.Pages**
 - Connected Services
 - Dependencies - 1 update
- Data**
 - ListenLaterService.cs**
 - PlayerService.cs**
 - SubscriptionsService.cs**
- Events**
 - CustomEvents.cs**
- Models**
- Pages**
 - CategoriesPage.razor**
 - CategoriesPage.razor.cs

@mallibone

.NET MAUI Architecture



@mallibone

.NET MAUI Architecture

UI Framework

.NET MAUI XAML

Blazor .NET MAUI

Comet

Fabulous

Framework

.NET MAUI

Platform

iOS

Android

Windows

macOS

Tizen

Web?

@mallbone



imgflip.com

**NO PRESENTS
WITH THAT UI FRIENDO**

@mallibone

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

Any CPU | Debug | Android Emulator | Live Share

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 | Error List | Output | Add to Source Control | Select Repository | Ready

Solution Explorer | Search Solution Explorer (Ctrl+U)

MauiApp1 | Dependencies | Properties | Platforms | Resources | App.xaml | AppShell.xaml | MainPage.xaml | MauiProgram.cs

Properties | XML Document | Misc | Encoding: Unicode (UTF-8) | Output: C:\Program Files\Microsoft Visual Schemas: "C:\Program Files\Microsoft Visual S Stylesheet:

Encoding: Character encoding of the document.

Demo Highlights

- .NET MAUI Single Project makes adding Fonts and Images easier
- Architecture Pattern MVVM
- MAUI Graphics for some extra Pizzazz



XAML vs Coded UI

.NET MAUI Architecture

UI Framework

.NET MAUI XAML

Blazor .NET MAUI

Comet

Fabulous

Framework

.NET MAUI

Platform

iOS

Android

Windows

macOS

Tizen

Web?

@mallbone

.NET MAUI Architecture

UI Framework

.NET MAUI XAML

Blazor .NET MAUI

Comet

Fabulous

Framework

.NET MAUI

Platform

iOS

Android

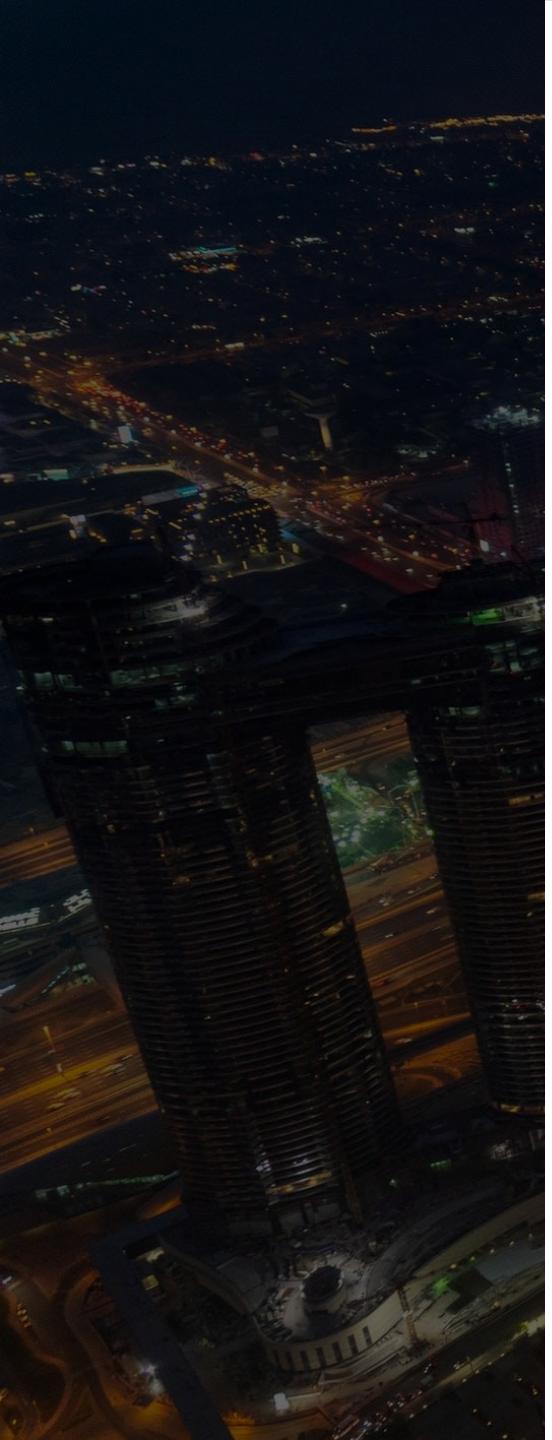
Windows

macOS

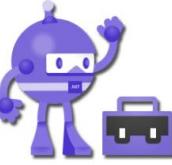
Tizen

Web?

@mallbone



☰ README.md

<https://github.com/CommunityToolkit/Maui>

Azure Pipelines succeeded nuget v1.0.0-rc2 ▾ 7.7k

.NET MAUI Community Toolkit

The .NET MAUI Community Toolkit is a collection of common elements for development with .NET MAUI that developers tend to replicate across multiple apps. It simplifies and demonstrates common developer tasks when building apps with .NET MAUI.

All features are contributed by you, our amazing .NET community, and maintained by a core set of maintainers. And – the best part – the features you add to the .NET MAUI Toolkit may one day be included into the official .NET MAUI library! We leverage the Community Toolkits to debut new features and work closely with the .NET MAUI engineering team to nominate features for promotion.

Getting Started

In order to use the .NET MAUI Community Toolkit you need to call the extension method in your `MauiProgram.cs` file as follows:

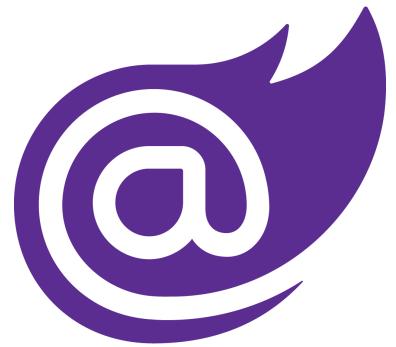
```
using CommunityToolkit.Maui;

public static class MauiProgram
{
    public static MauiApp CreateMauiApp()
    {
        var builder = MauiApp.CreateBuilder();

        // Initialise the toolkit
        builder.UseMauiApp<App>().UseMauiCommunityToolkit();
```

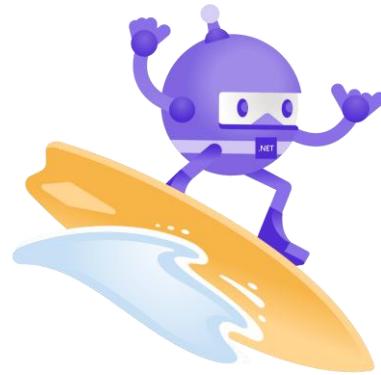
@mallbone



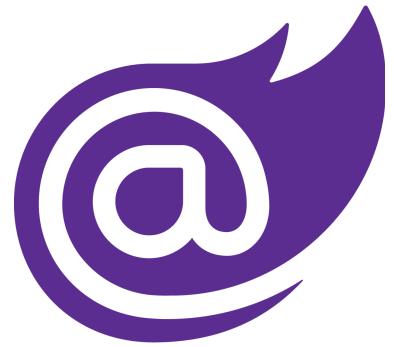


Blazor

+

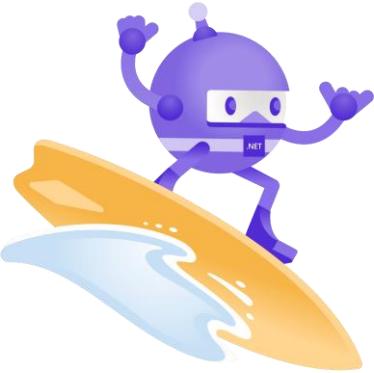


.NET MAUI



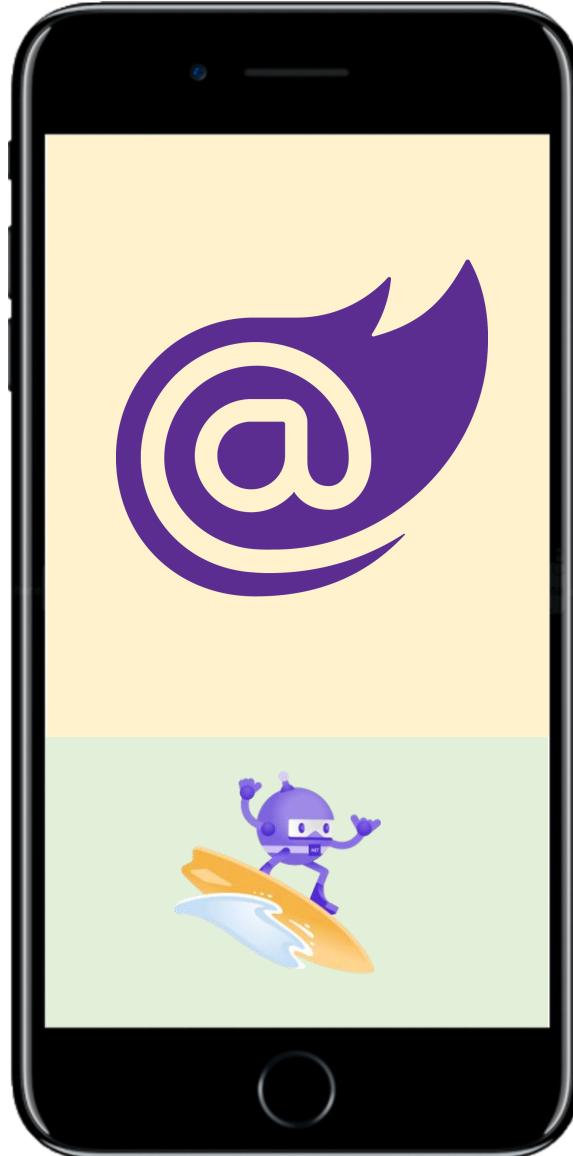
Blazor

+



.NET MAUI

=



.NET MAUI Blazor

Solution Explorer

Search Solution Explorer (Ctrl+U)

MauiApp1

MauiApp1

Dependencies

Properties

Platforms

- Android
- iOS
- MacCatalyst
- Tizen
- Windows

Resources

App.xaml

AppShell.xaml

MainPage.xaml

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.

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

MauiApp1

Any CPU

Debug

Android Emulator

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')</MSBuild>"><TargetFrameworks>$(TargetFrameworks);net6.0-windows10.0.19041.0</TargetFrameworks>
    <!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->
    <!-- <TargetFrameworks>$(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\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

Ln: 1 Ch: 1 TABS CRLF

Error List Output

Ready

Add to Source Control

Select Repository

Demo Review

BlazorWebView for creating Hybrid .NET MAUI Apps

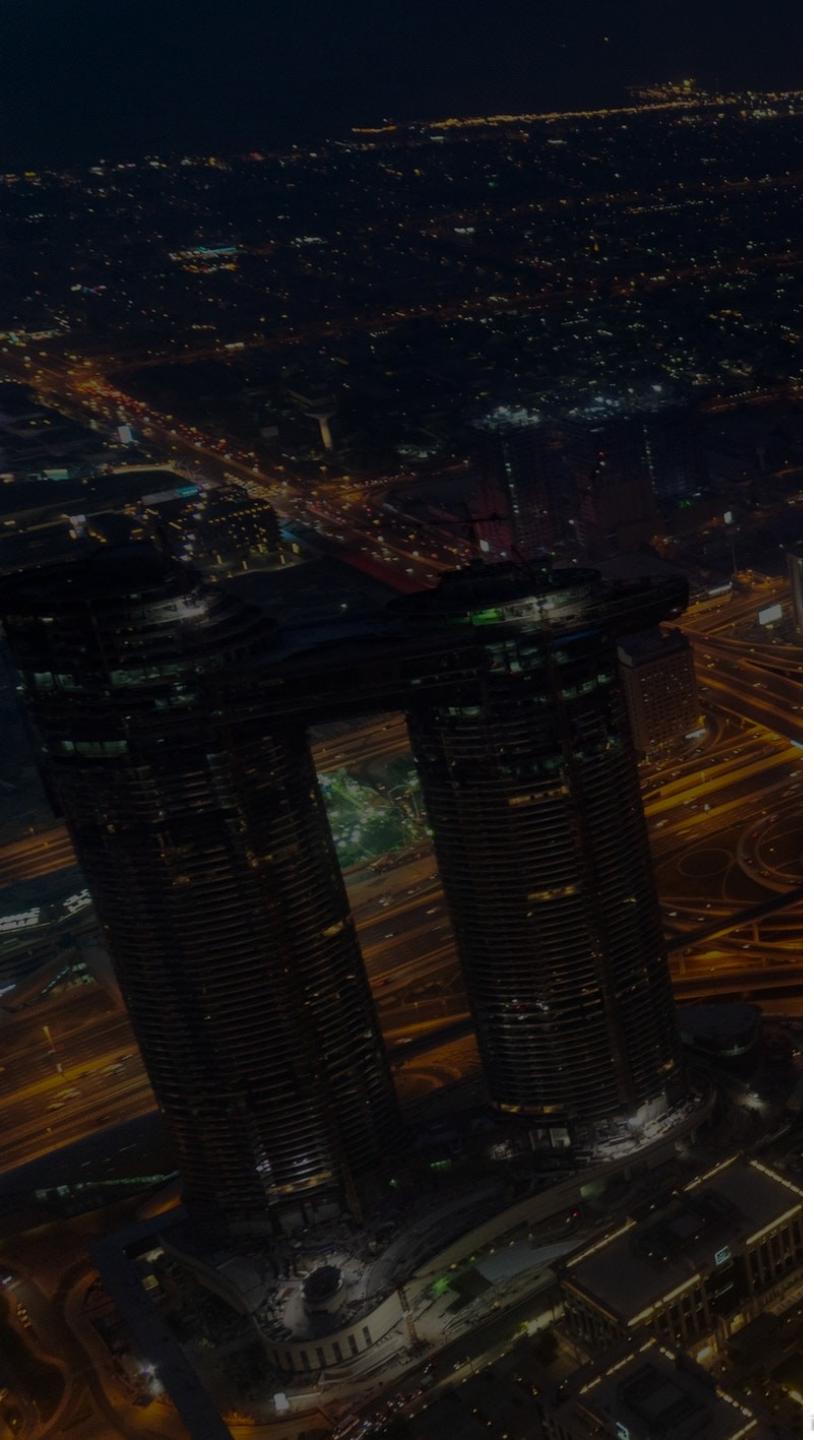
Embed and host Blazor websites in your app

Share state between your apps



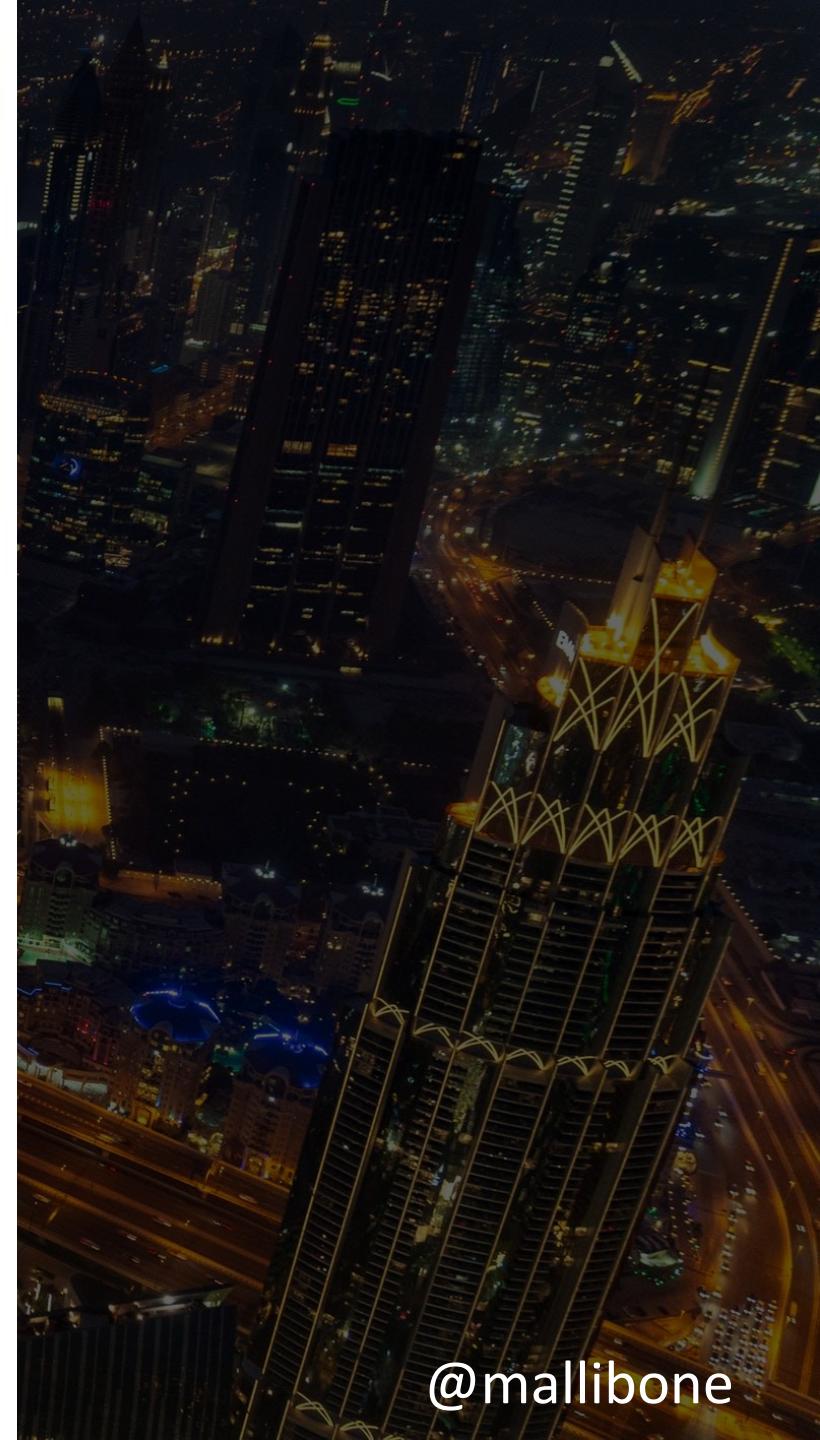
HEY CUPCAKE

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

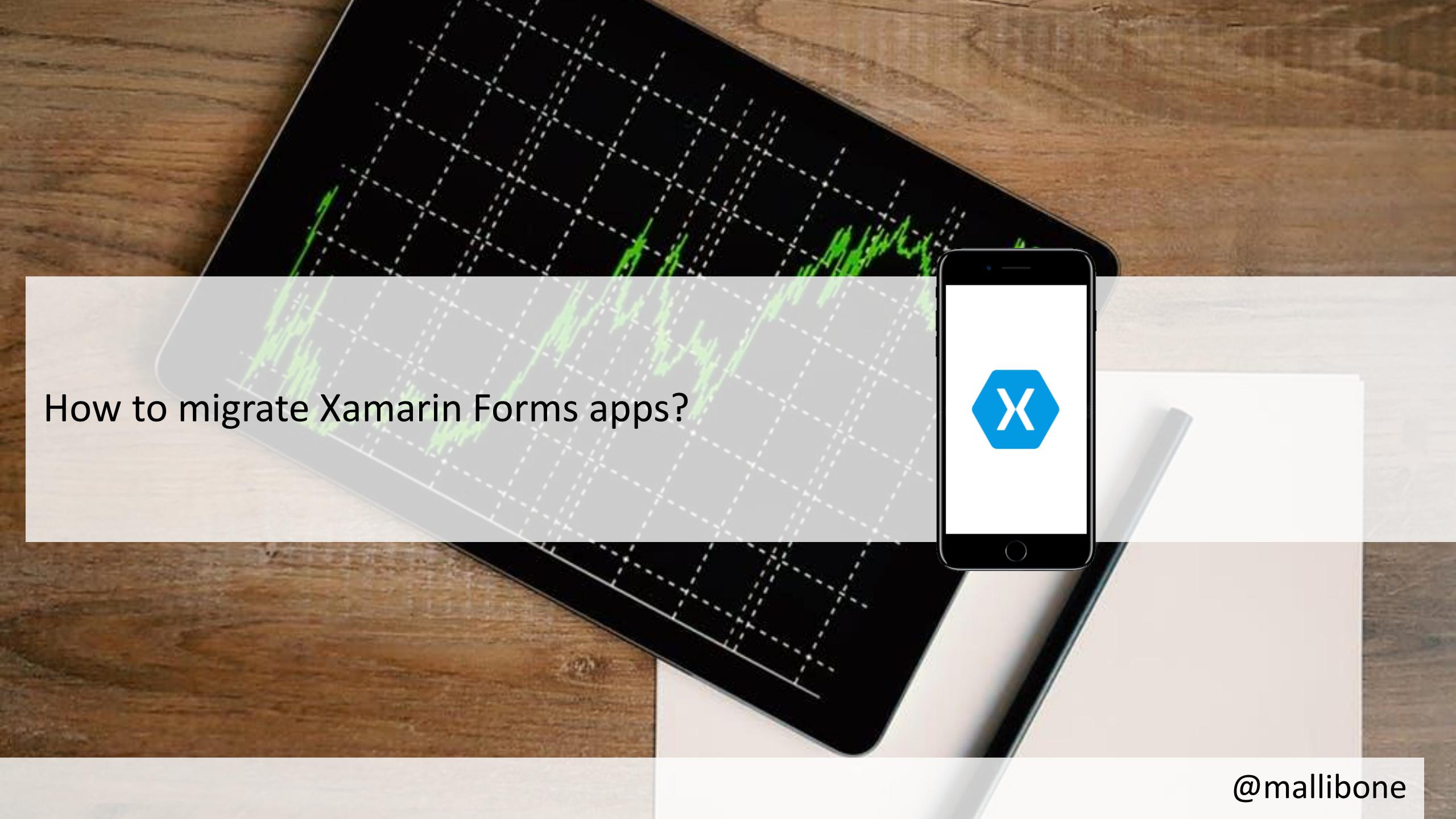


imgflip.com

JAKE-CLARK.TUMBLR



@mallibone



How to migrate Xamarin Forms apps?



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



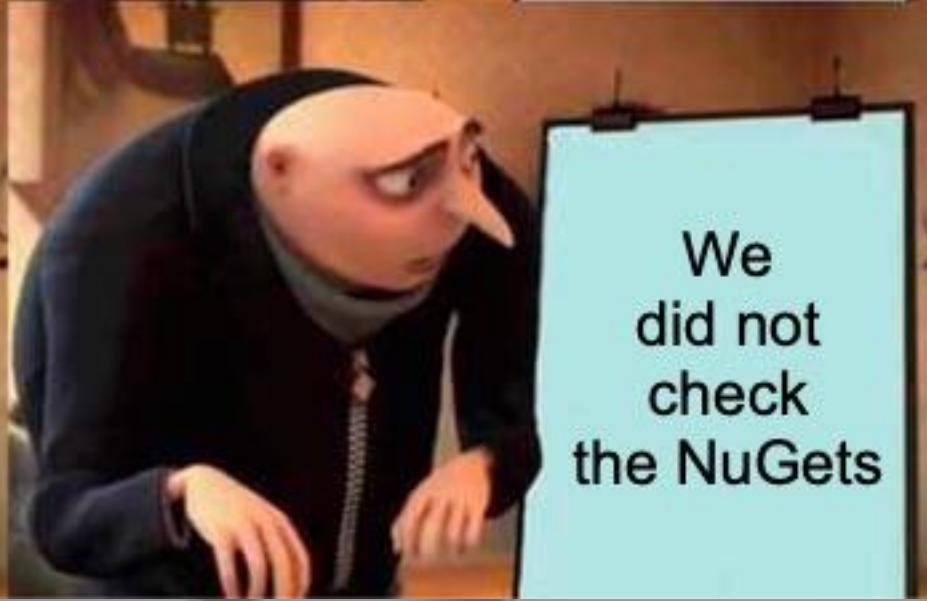
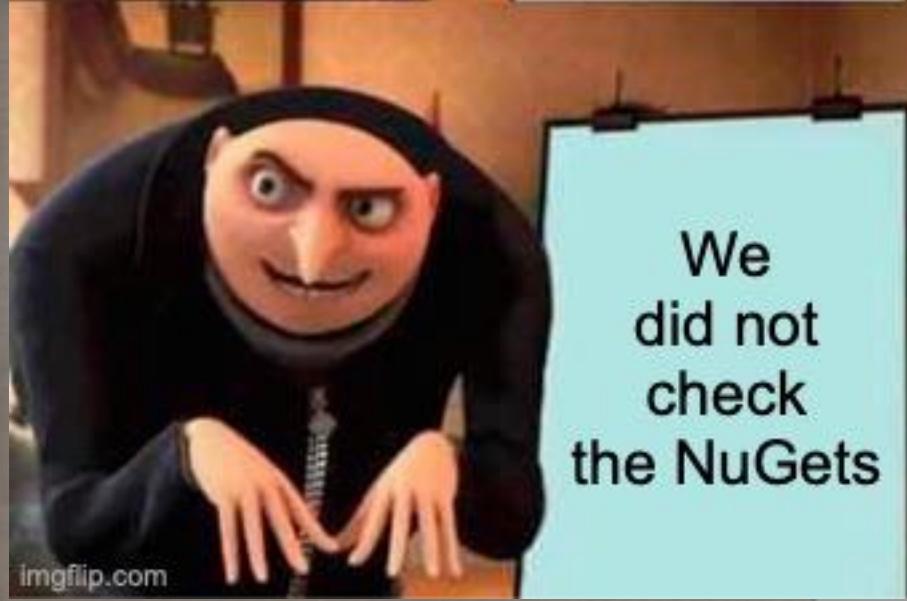
What about the Xamarin Essentials?



The background image is a scenic tropical sunset. The sky is a gradient from deep blue at the top to a bright orange and yellow near the horizon. Several tall palm trees stand in the mid-ground, silhouetted against the light. In the foreground, there's a lush green lawn and some tropical foliage, including a large yellow flower. A circular overlay on the right side contains the text.

Essentials

It's all part of .NET MAUI



Announcing .NET MAUI for .NET 7 General Availability



David Ortinau

November 8th, 2022 | 24 | 9

Six short months ago we introduced you to .NET Multi-platform App UI (MAUI) and today we are excited to announce the general availability of .NET MAUI in our next major release, [.NET 7](#). Our primary work in .NET MAUI during this shortened timeframe has been on addressing your top feedback reports, improving the performance of [CollectionView](#), and introducing desktop features as we have expanded your reach beyond mobile to desktop. This release accompanies the release of Visual Studio 17.4, and the first .NET MAUI features have graduated from preview to stable release on Mac.

.NET MAUI for .NET 7 is supported through May of 2024. This includes a 6-month overlap with support for .NET MAUI for .NET 6 through May of 2023. .NET MAUI support includes the .NET MAUI framework as well as the .NET SDKs for Android, CarPlay, iOS, macOS, Mac Catalyst, and tvOS.

You, the .NET customers and ecosystem contributors have also been hard at work building new applications with .NET MAUI, and modernizing old SDKs and libraries to be .NET compatible. Join us for the State of .NET MAUI presentation today on the [.NET Conf 2022 livestream](#) where we will highlight and celebrate all this progress.

.NET 7

- Maps
- Mobile Rendering Performance
- Desktop Enhancements

Announcing .NET MAUI for .NET 7 General Availability



David Ortinau

November 8th, 2022 | 24 | 9

Six short months ago we introduced you to .NET Multi-platform App UI (MAUI). We're excited to announce the general availability of .NET MAUI in our next major release. The primary work in .NET MAUI during this shortened timeframe has been focused on addressing your feedback reports, improving the performance of [CollectionView](#), and expanding the reach of .NET MAUI beyond mobile to desktop. This release includes support for Visual Studio 17.4, and the first .NET MAUI features have graduated to General Availability.

.NET MAUI for .NET 7 is supported through May of 2024. This includes support for .NET MAUI for .NET 6 through May of 2023. .NET MAUI is now available as a separate framework, along with the .NET MAUI framework as well as the .NET SDKs for Android, iOS, macOS, Catalyst, and tvOS.

You, the .NET customers and ecosystem contributors have also been instrumental in helping us get here. We've been working on modernizing applications with .NET MAUI, and modernizing old SDKs and libraries. We're excited to share the State of .NET MAUI presentation today on the [.NET Conf 2023](#) stage, where we can highlight and celebrate all this progress.

Takeaways



Takeaways

- .NET MAUI is an evolution of Xamarin Forms
- 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>

