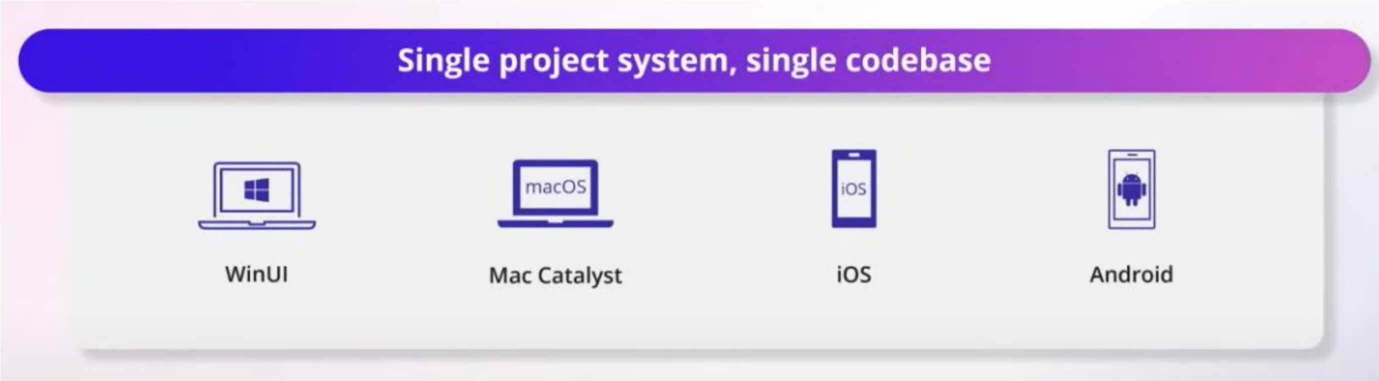
.Net MAUI app using C# Markup

https://learn.microsoft.com/en-us/windows/apps/windows-dotnet-maui/tutorial-csharp-ui-maui-toolkit

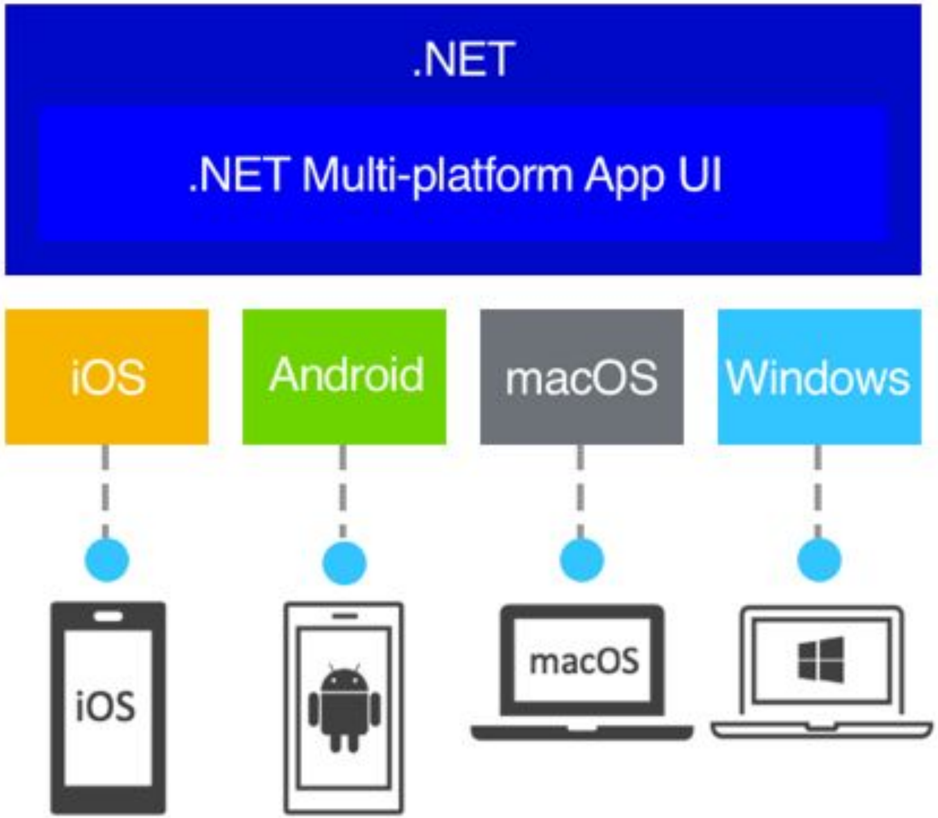
What is MAUI?

- **Cross-platform** framework to develop apps for Windows, Android, iOS, macOS from a single **shared code-base**



https://dotnet.microsoft.com/en-us/apps/maui

High-level view of the architecture



Written on Windows / Mac, compiled to native

- ❑ **Android**: C# -> intermediate language (**IL**). When launch, just-in-time (**JIT**) compiled to a native assembly
- ❑ **iOS**: fully ahead-of-time (**AOT**) compiled to native ARM assembly code
- ❑ **macOS**: use Mac **Catalyst**, built with **UIKit** to the desktop, and augments with additional **AppKit** and platform APIs as required
- ❑ **Windows**: use Windows UI 3 (**WinUI 3**) library to create native apps that target the Windows desktop

What is C# Markup?

- ❑ One of the features of **.NET MAUI Community Toolkit**
  - ❑ Extensions
  - ❑ Behaviors
  - ❑ Animations
  - ❑ Other helpers
- ❑ The ability to create a user interface entirely in C# code

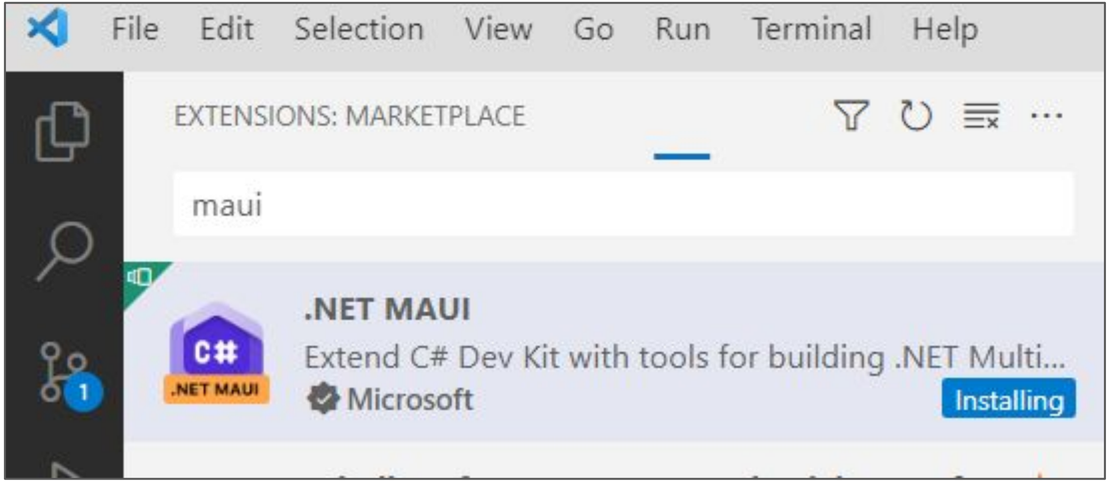
Why should we use C# markup instead of XAML?

- ❑ MVU style of React switch (Elm)
  - ❑ iOS: SwiftUI
  - ❑ Android: Jetpack Compose
- ❑ Quick learning curve for C# developers

C# markup Quick start guide

Setting up the environment

- ❑ **Visual Studio Code**
  - ❑ Install extension: **.Net Maui**
  - ❑ **dotnet workload install maui**



maui

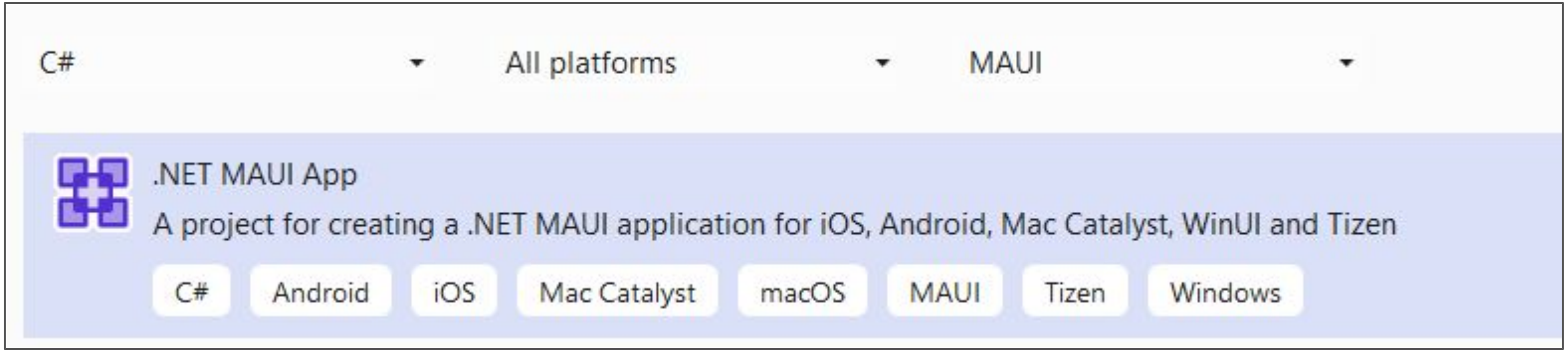
.NET MAUI

Extend C# Dev Kit with tools for building .NET Multi...

Microsoft

Create a new project

- ❑ **Visual Studio**



C# ▾ All platforms ▾ MAUI ▾

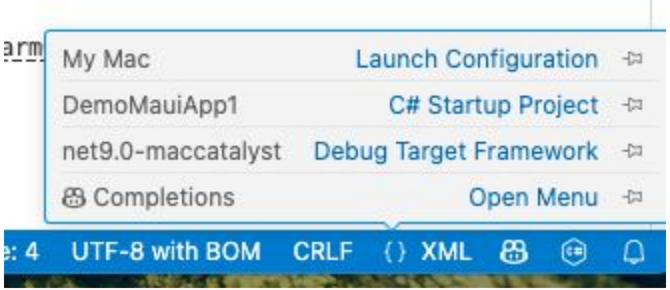
.NET MAUI App

A project for creating a .NET MAUI application for iOS, Android, Mac Catalyst, WinUI and Tizen

C# Android iOS Mac Catalyst macOS MAUI Tizen Windows

- ❑ **Visual Studio Code**
  - ❑ **Ctrl + Shift + P** > New Project > .Net Maui App Mobile Desktop
  - ❑ Choose folder > Choose the name of the project

Select config



My Mac Launch Configuration

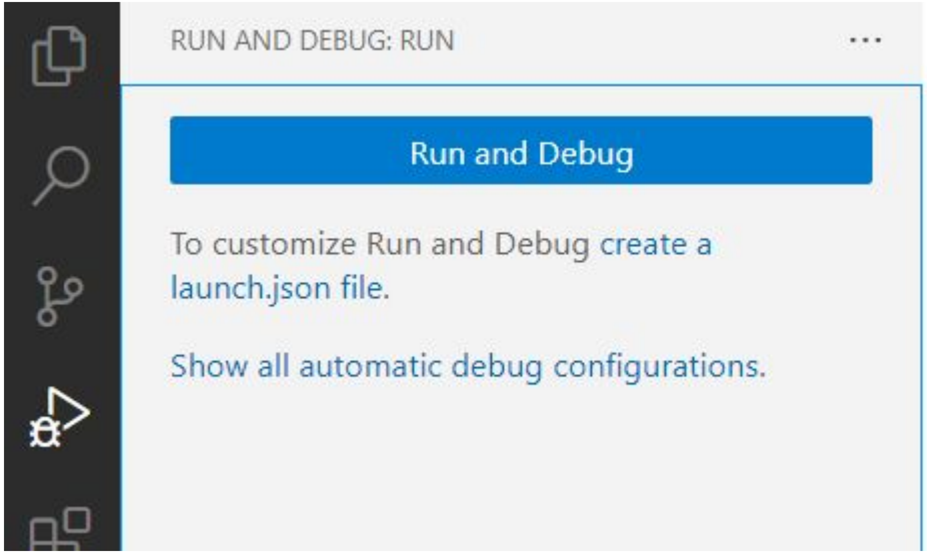
DemoMauiApp1 C# Startup Project

net9.0-maccatalyst Debug Target Framework

Completions Open Menu

VS Code: Create a launch.json file to build and run

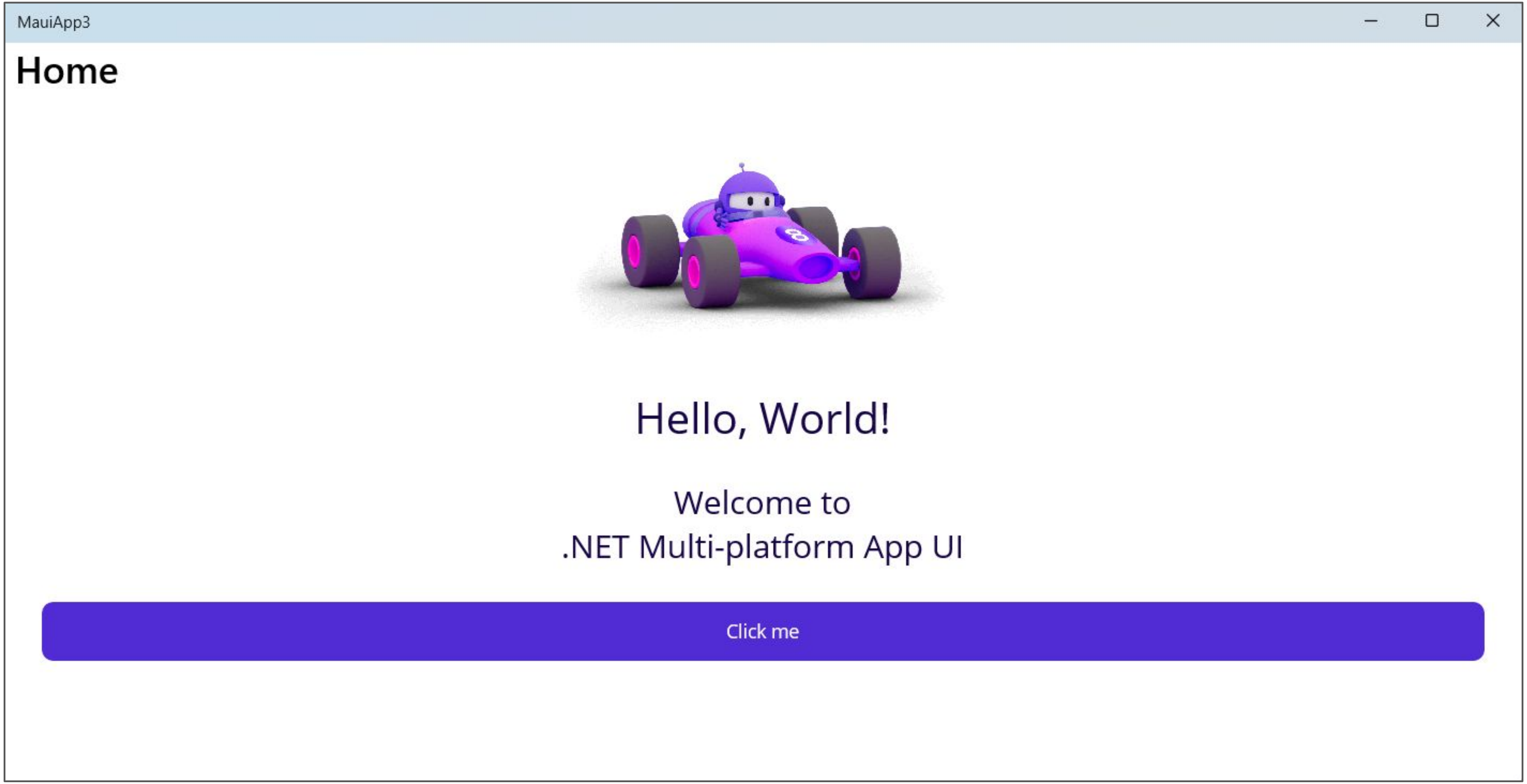
- ❑ Open file **csproj**
- ❑ Click on Create a launch.json file



To customize Run and Debug create a launch.json file.

Show all automatic debug configurations.

Sample result on Windows

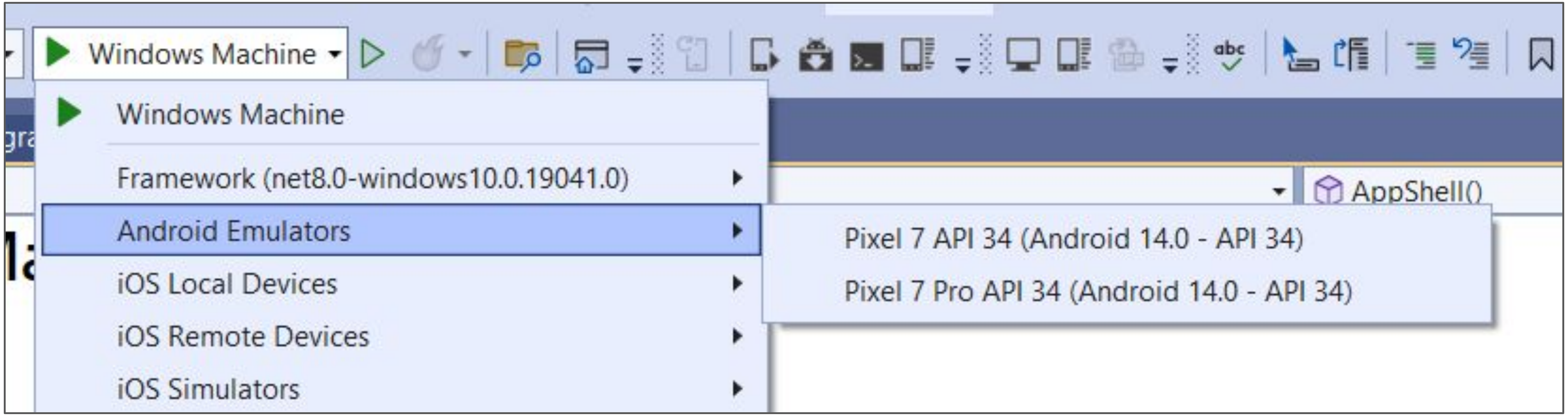


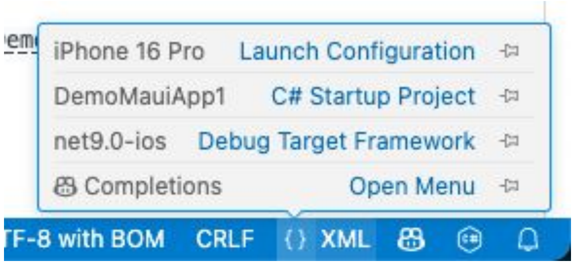
Home

Hello, World!

Welcome to
.NET Multi-platform App UI

Switch to another platform - Visual Studio


Windows Machine
Windows Machine
Framework (net8.0-windows10.0.19041.0)
Android Emulators
iOS Local Devices
iOS Remote Devices
iOS Simulators
Pixel 7 API 34 (Android 14.0 - API 34)
Pixel 7 Pro API 34 (Android 14.0 - API 34)
AppShell()


iPhone 16 Pro Launch Configuration

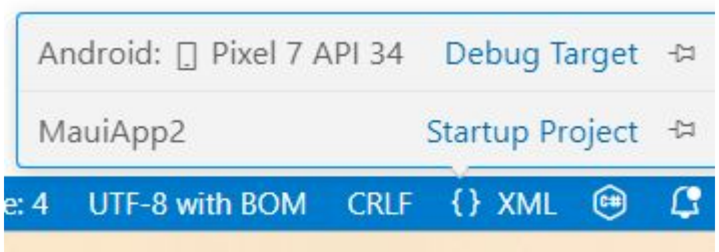
DemoMauiApp1 C# Startup Project

net9.0-ios Debug Target Framework

Completions Open Menu

Switch to another platform - Visual Studio Code

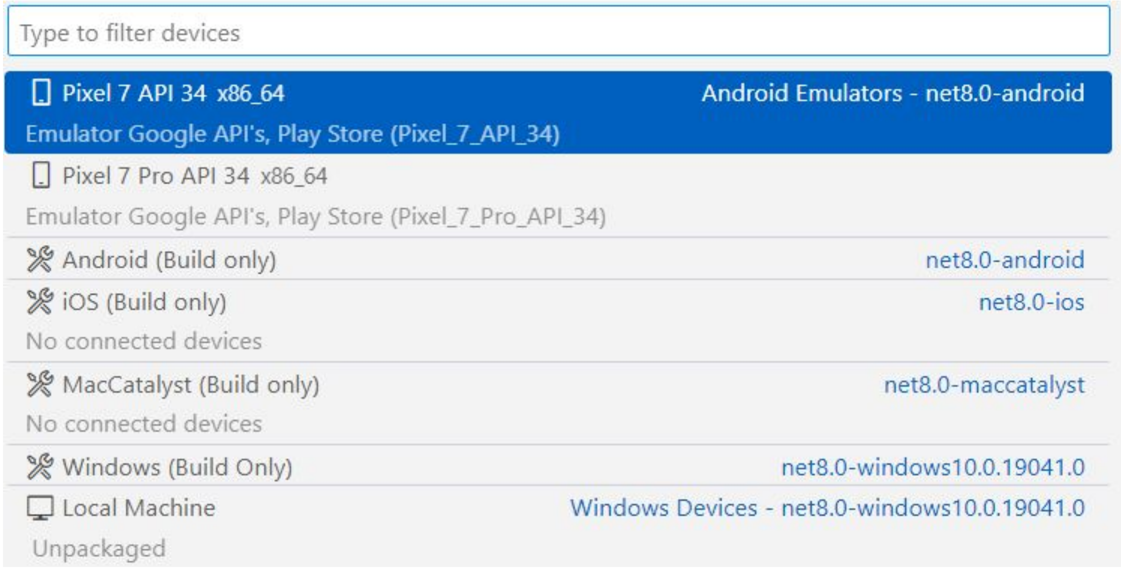
- Click on the {} button on the bottom right corner



Android: ▯ Pixel 7 API 34 Debug Target

MauiApp2 Startup Project

e: 4 UTF-8 with BOM CRLF { } XML



Type to filter devices

▯ Pixel 7 API 34 x86_64 Android Emulators - net8.0-android
Emulator Google API's, Play Store (Pixel_7_API_34)

▯ Pixel 7 Pro API 34 x86_64
Emulator Google API's, Play Store (Pixel_7_Pro_API_34)

Android (Build only) net8.0-android

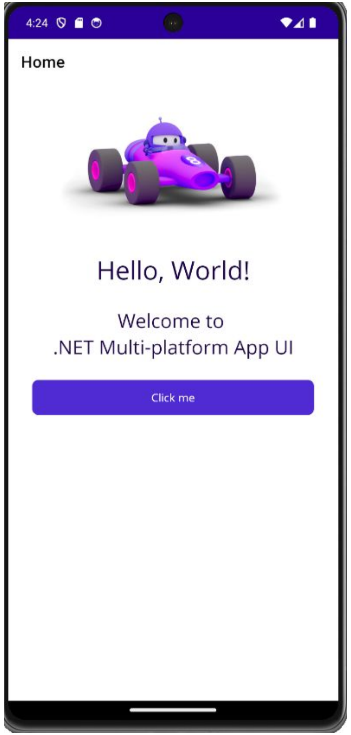
iOS (Build only) net8.0-ios
No connected devices

MacCatalyst (Build only) net8.0-maccatalyst
No connected devices

Windows (Build Only) net8.0-windows10.0.19041.0

Local Machine Windows Devices - net8.0-windows10.0.19041.0
Unpackaged

Sample result on Android

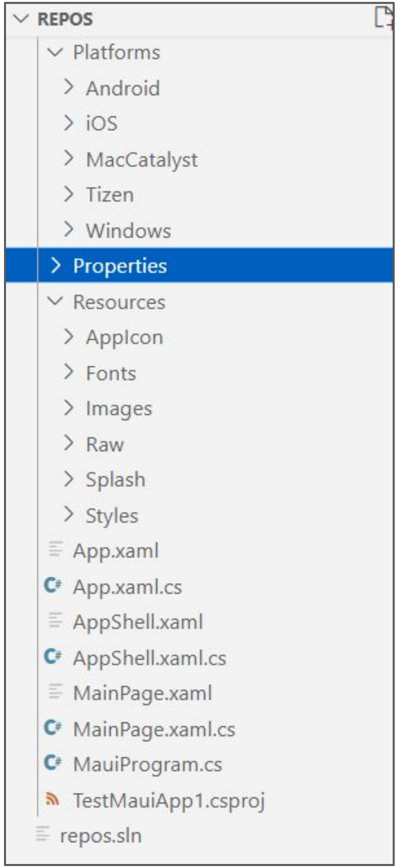


Hello, World!

Welcome to
.NET Multi-platform App UI

- ❏ May need to install **Microsoft JDK 17** (not 21)
- ❏ May need to install Android Platform tool 35
- ❏ Read the errors and instructions carefully

Project structure



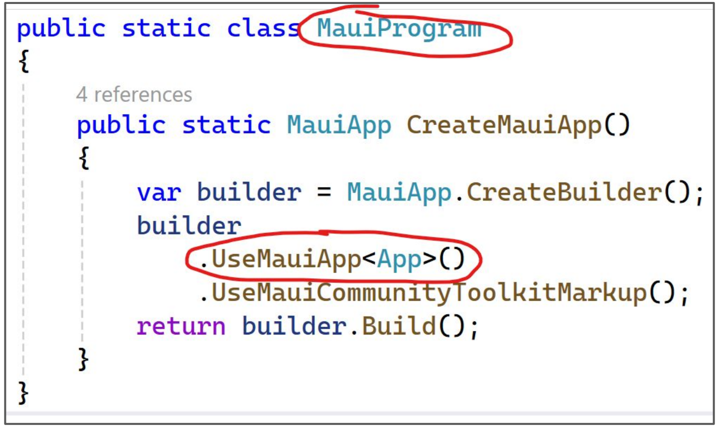
- REPOS
  - Platforms
    - Android
    - iOS
    - MacCatalyst
    - Tizen
    - Windows
  - Properties
  - Resources
    - AppIcon
    - Fonts
    - Images
    - Raw
    - Splash
    - Styles
  - App.xaml
  - App.xaml.cs
  - AppShell.xaml
  - AppShell.xaml.cs
  - MainPage.xaml
  - MainPage.xaml.cs
  - MauiProgram.cs
  - TestMauiApp1.csproj
- repos.sln

Supported platforms

- ❏ Android
- ❏ iOS
- ❏ Mac
- ❏ Windows

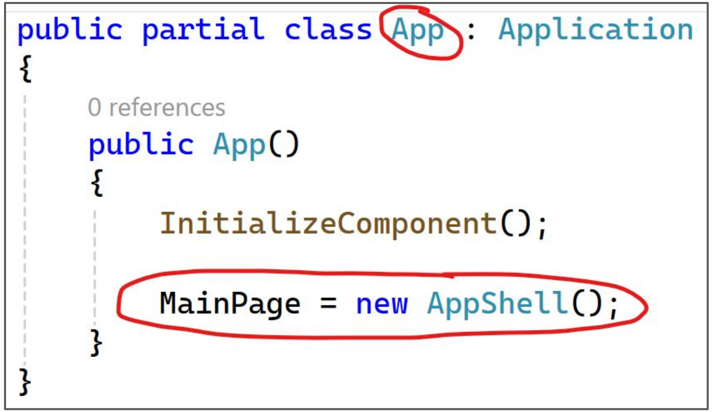
Entry point & flow of execution

MauiProgram.cs



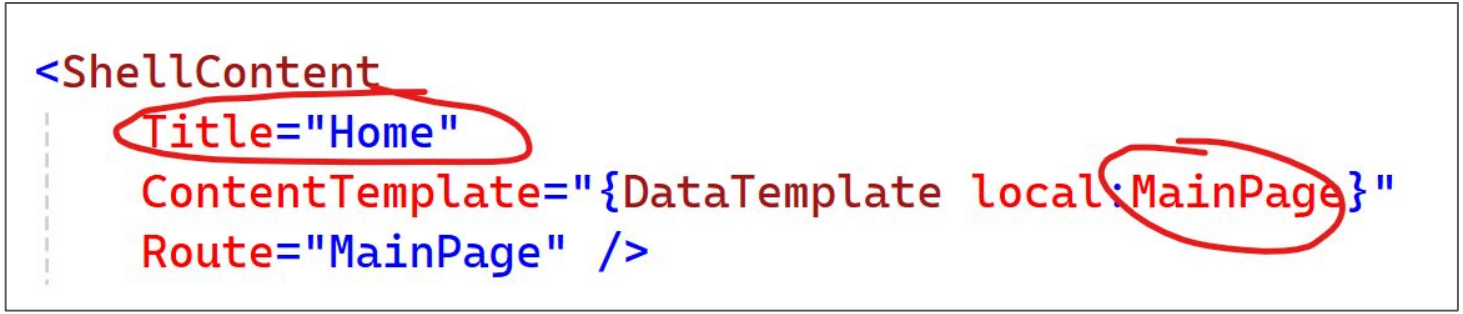
```
public static class MauiProgram
{
    4 references
    public static MauiApp CreateMauiApp()
    {
        var builder = MauiApp.CreateBuilder();
        builder
            .UseMauiApp<App>()
            .UseMauiCommunityToolkitMarkup();
        return builder.Build();
    }
}
```

App.xaml.cs



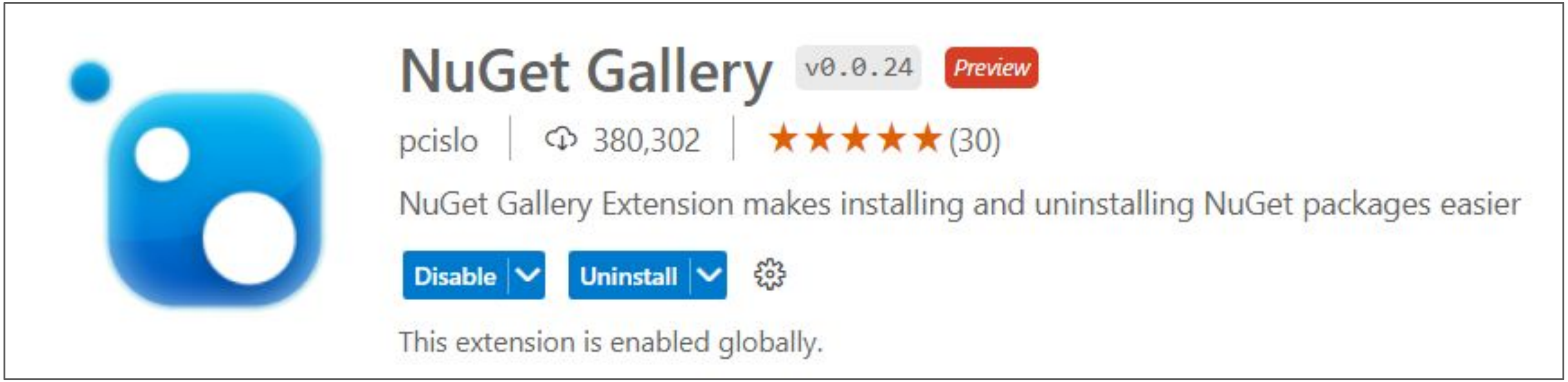
```
public partial class App : Application
{
    0 references
    public App()
    {
        InitializeComponent();

        MainPage = new AppShell();
    }
}
```

AppShell.xaml



```
<ShellContent
    Title="Home"
    ContentTemplate="{DataTemplate local:MainPage}"
    Route="MainPage" />
```

VSCode: Install Nuget Gallery extension



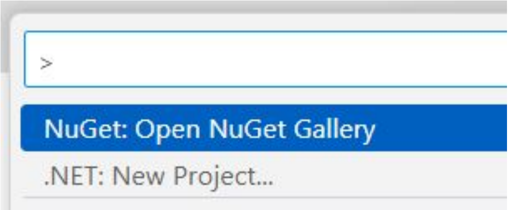
NuGet Gallery v0.0.24 Preview

pcislo | 380,302 | ★★★★★ (30)

NuGet Gallery Extension makes installing and uninstalling NuGet packages easier

This extension is enabled globally.

Ctrl + Shift + P > NuGet: Open NuGet Gallery



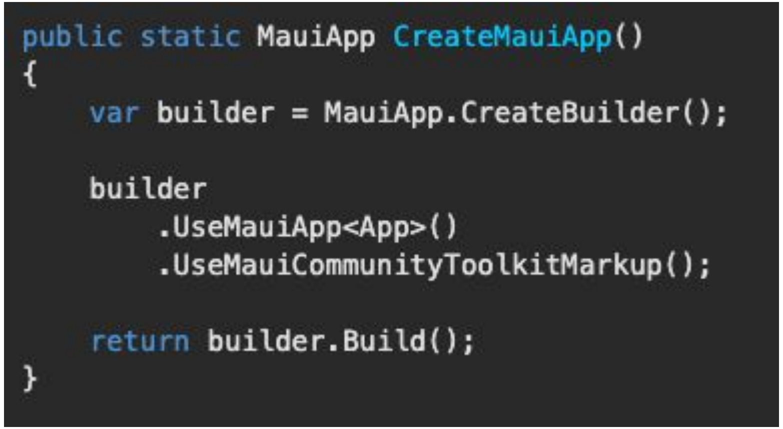
Add C# Markup using Community Toolkit

- **CommunityToolkit.Maui.Markup**

```
dotnet add package CommunityToolkit.Maui.Markup
```

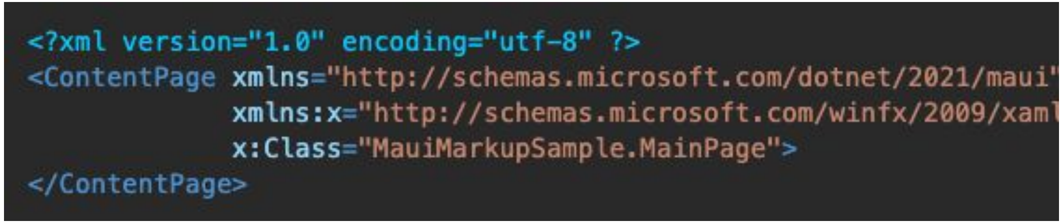
Build a UI with C# Markup

MauiProgram.cs



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

    builder
        .UseMauiApp<App>()
        .UseMauiCommunityToolkitMarkup();

    return builder.Build();
}
```

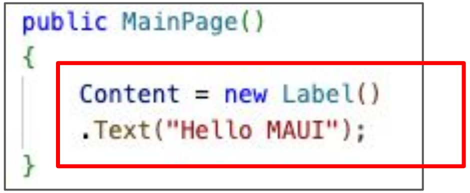
MainPage.xaml: Delete the file content



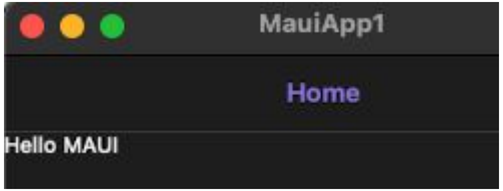
```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml
             x:Class="MauiMarkupSample.MainPage">
</ContentPage>
```

MainPage.xaml.cs

using CommunityToolkit.Maui.Markup;

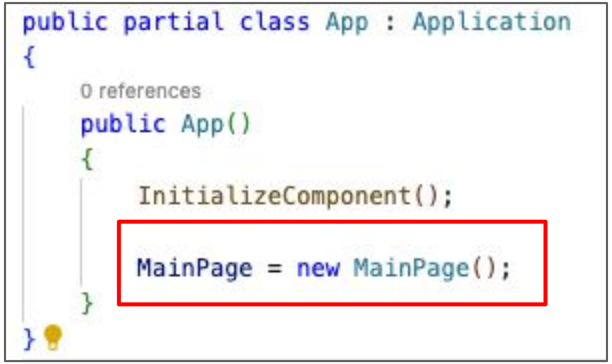


```
public MainPage()
{
    Content = new Label()
    .Text("Hello MAUI");
}
```



Removing the AppShell (for simplicity)

1. Delete two files: (We will talk about Shell later)
   a. **AppShell.xaml**
   b. **AppShell.xaml.cs**
2. Modify **App.xaml.cs** to display our MainPage

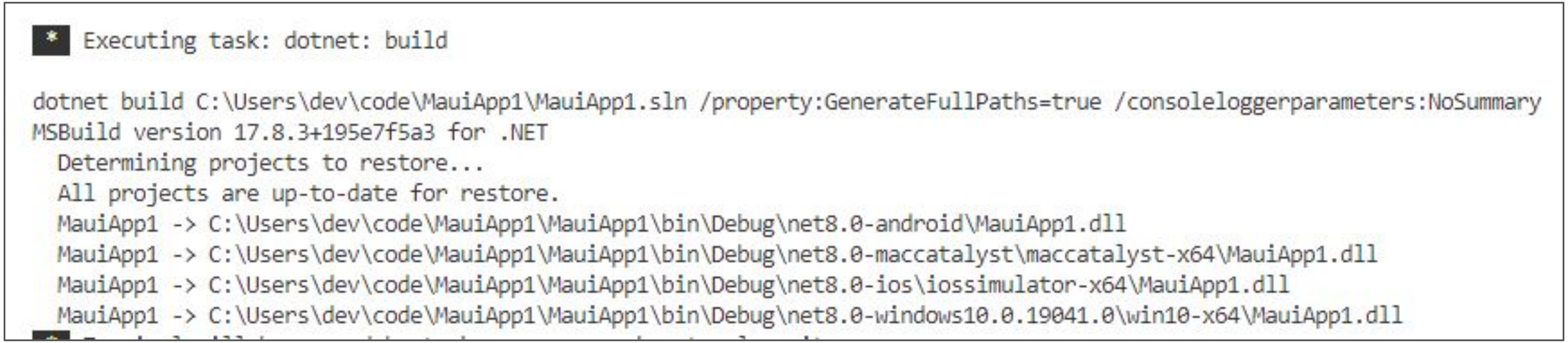


```
public partial class App : Application
{
    0 references
    public App()
    {
        InitializeComponent();

        MainPage = new MainPage();
    }
}
```

Artifacts

Perform a build only

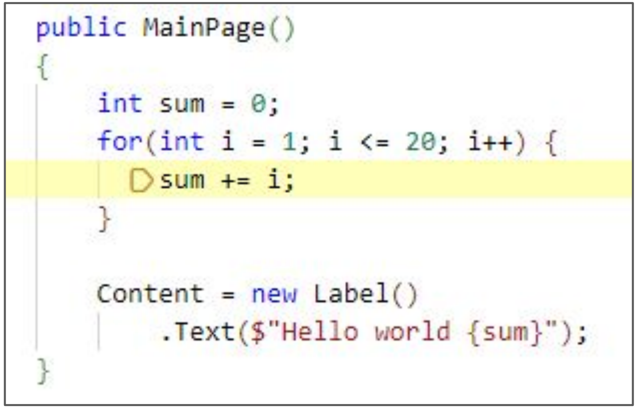
Ctrl + Shift + P > .Net Build



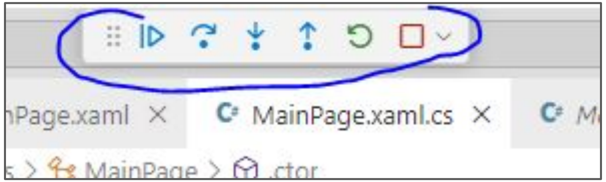
```
* Executing task: dotnet: build

dotnet build C:\Users\dev\code\MauiApp1\MauiApp1.sln /property:GenerateFullPaths=true /consoleloggerparameters:NoSummary
MSBuild version 17.8.3+195e7f5a3 for .NET
  Determining projects to restore...
  All projects are up-to-date for restore.
  MauiApp1 -> C:\Users\dev\code\MauiApp1\MauiApp1\bin\Debug\net8.0-android\MauiApp1.dll
  MauiApp1 -> C:\Users\dev\code\MauiApp1\MauiApp1\bin\Debug\net8.0-maccatalyst\maccatalyst-x64\MauiApp1.dll
  MauiApp1 -> C:\Users\dev\code\MauiApp1\MauiApp1\bin\Debug\net8.0-ios\iossimulator-x64\MauiApp1.dll
  MauiApp1 -> C:\Users\dev\code\MauiApp1\MauiApp1\bin\Debug\net8.0-windows10.0.19041.0\win10-x64\MauiApp1.dll
```

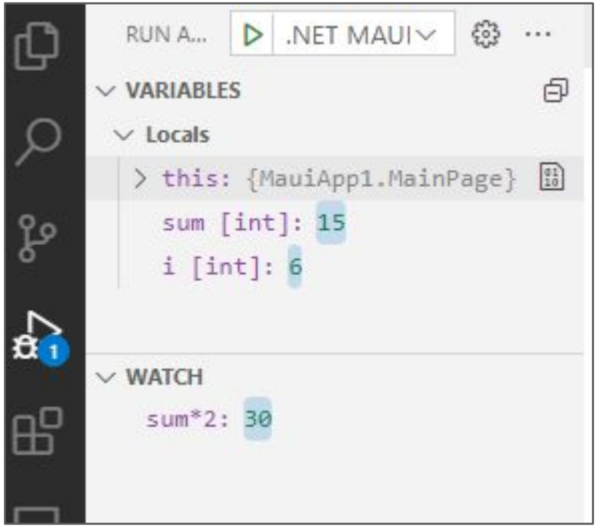
Debug

Modify MainPage.xaml.cs & set breakpoint



```
public MainPage()
{
    int sum = 0;
    for(int i = 1; i <= 20; i++) {
        sum += i;
    }

    Content = new Label()
        .Text($"Hello world {sum}");
}
```





VARIABLES

Locals

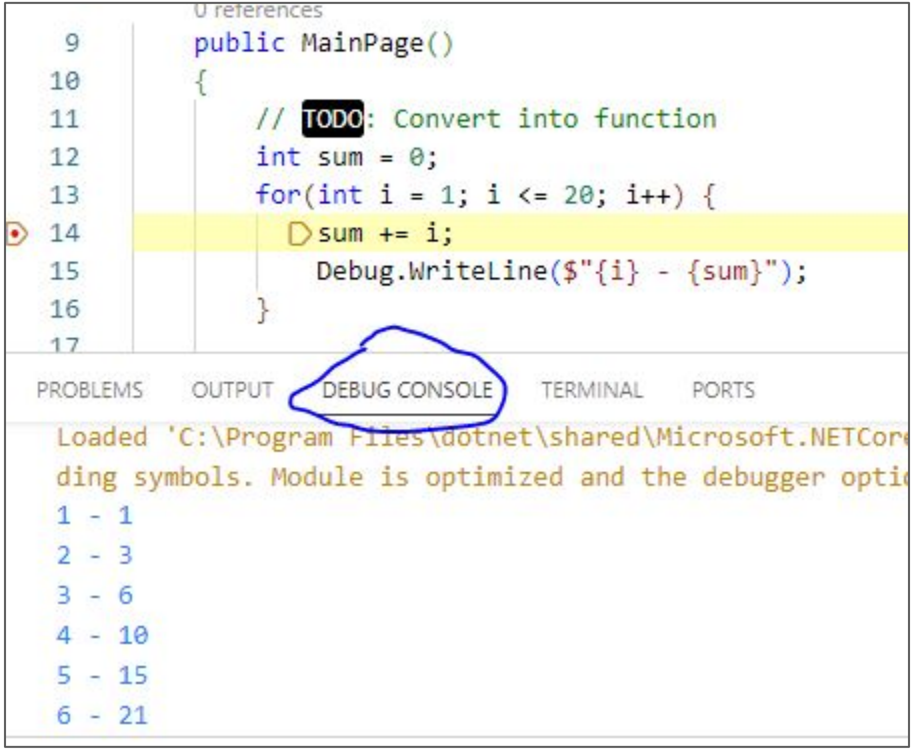
```
> this: {MauiApp1.MainPage}
  sum [int]: 15
  i [int]: 6
```

WATCH

```
sum*2: 30
```

- ❑ Continue: F5
- ❑ Step over: F10
- ❑ Step into: F11
- ❑ Step out: Shift+F11

Adding debug info into console

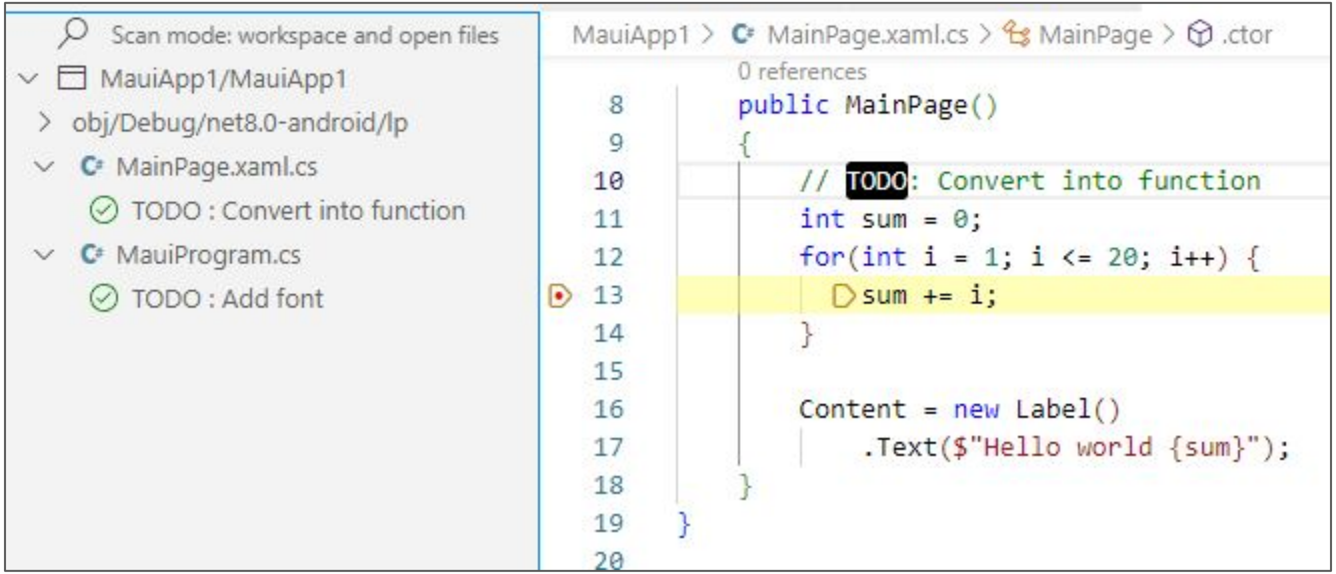


```
public MainPage()
{
    // TODO: Convert into function
    int sum = 0;
    for(int i = 1; i <= 20; i++) {
        sum += i;
        Debug.WriteLine($"{i} - {sum}");
    }
```

```
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore
ding symbols. Module is optimized and the debugger opti
1 - 1
2 - 3
3 - 6
4 - 10
5 - 15
6 - 21
```

TODO

Install TODO Tree Extension



Scan mode: workspace and open files

- MauiApp1/MauiApp1
  - obj/Debug/net8.0-android/lp
  - MainPage.xaml.cs
    - TODO : Convert into function
  - MauiProgram.cs
    - TODO : Add font

MauiApp1 > MainPage.xaml.cs > MainPage > .ctor

```
    0 references
    public MainPage()
    {
        // TODO: Convert into function
        int sum = 0;
        for(int i = 1; i <= 20; i++) {
            sum += i;
        }

        Content = new Label()
            .Text($"Hello world {sum}");
    }
}
```

View code behind extension

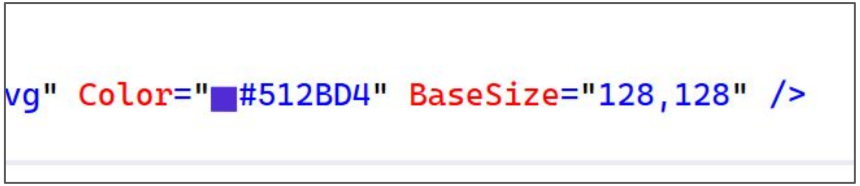
To switch between code & markup file

App icon & Splash screen

https://learn.microsoft.com/en-us/dotnet/maui/user-interface/images/app-icons

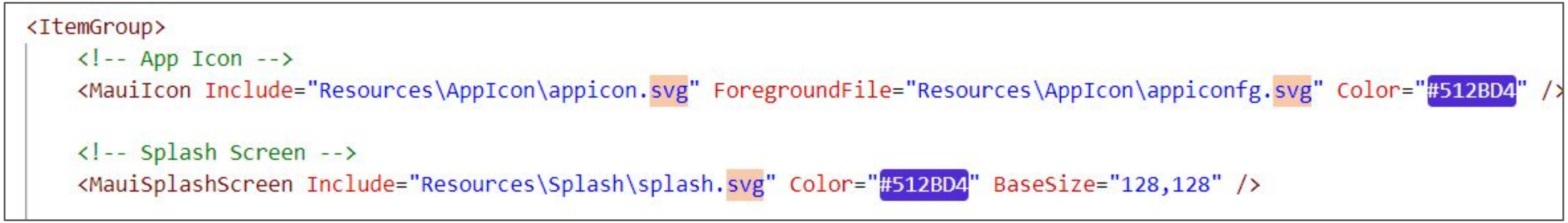
Add color highlight

- **Visual Studio**: Right Click > Edit project file (Install **Inline Color Picker** extension)



```
vg" Color="■#512BD4" BaseSize="128,128" />
```

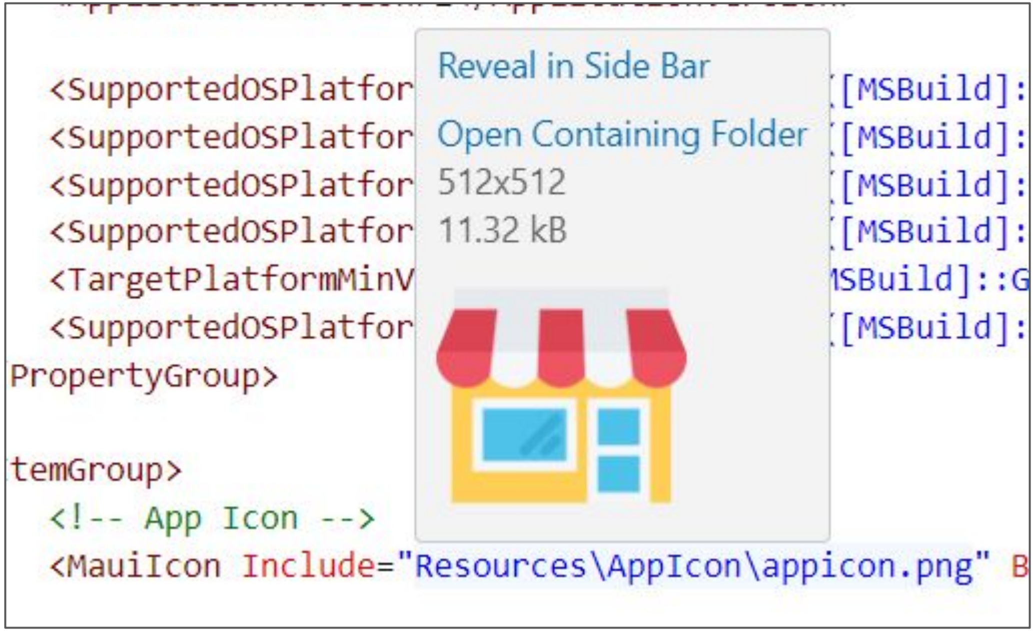
- **Visual Studio Code**: Open .csproj file



```
<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" />
```

(Install **Color highlight** extension)

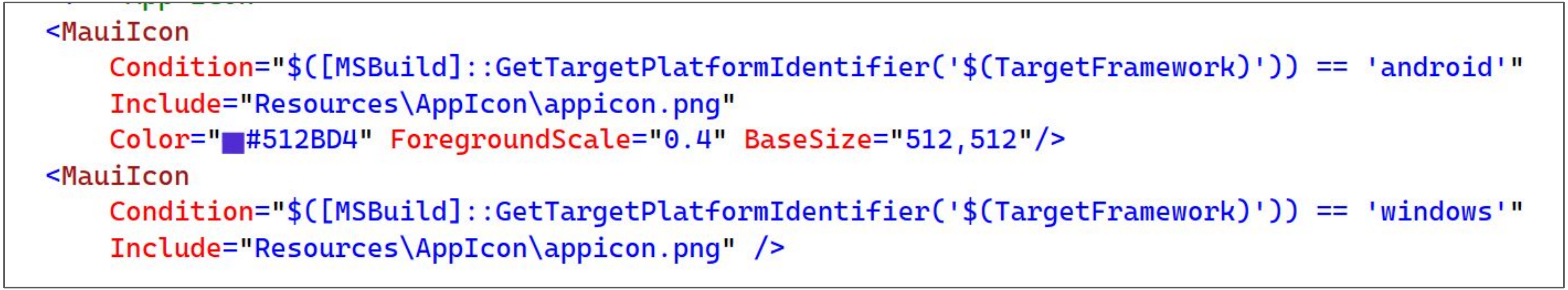
Replace **appicon**.svg with your image (keep the name)



```
  <SupportedOSPlatfor                                   [MSBuild]:
  <SupportedOSPlatfor                                   [MSBuild]:
  <SupportedOSPlatfor                                   [MSBuild]:
  <SupportedOSPlatfor                                   [MSBuild]:
  <TargetPlatformMinV                                   MSBuild]::G
  <SupportedOSPlatfor                                   [MSBuild]:
PropertyGroup>

temGroup>
  <!-- App Icon -->
  <MauiIcon Include="Resources\AppIcon\appicon.png" B
```

(Install extension **Image Preview** for hover effect)

Conditional for Android

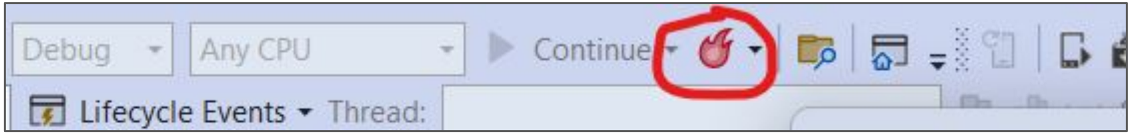


```
<MauiIcon
    Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'android'"
    Include="Resources\AppIcon\appicon.png"
    Color="■#512BD4" ForegroundScale="0.4" BaseSize="512,512"/>
<MauiIcon
    Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'windows'"
    Include="Resources\AppIcon\appicon.png" />
```

Hot reload

Usage

- ❏ Make change in the source code
- ❏ Click on **Hot reload** button to see changes **immediately**



Only on Windows with Visual Studio not VS Code

On by default in pre-release version

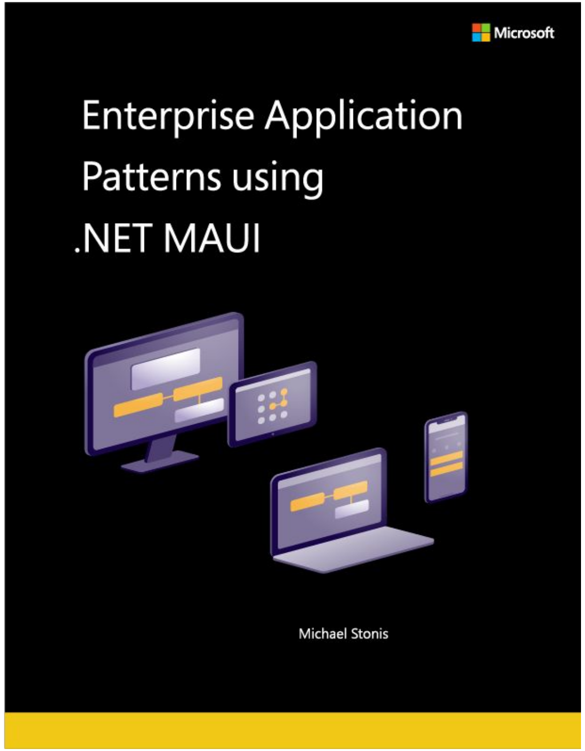
- ❑ Check maui.markup samples
  - ❑ https://github.com/CommunityToolkit/Maui.Markup

- C# Hot Reload with **Android** and **WinUI** should work, but **iOS/Catalyst** aren't yet supported.

Architecture

Enterprise Application Patterns Using .NET MAUI

https://docs.microsoft.com/dotnet/architecture/maui/



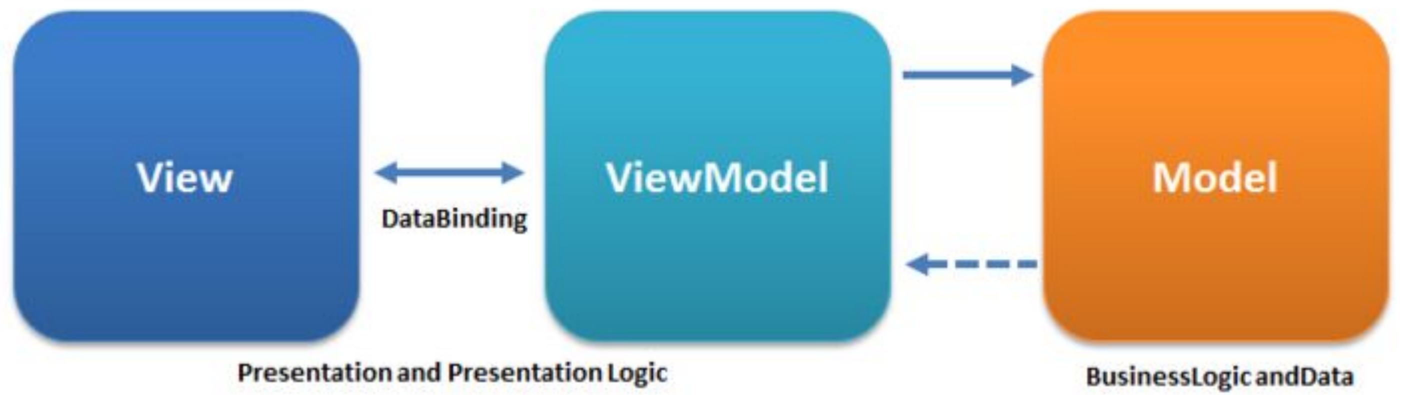
Microsoft

Enterprise Application Patterns using .NET MAUI

Michael Stonis

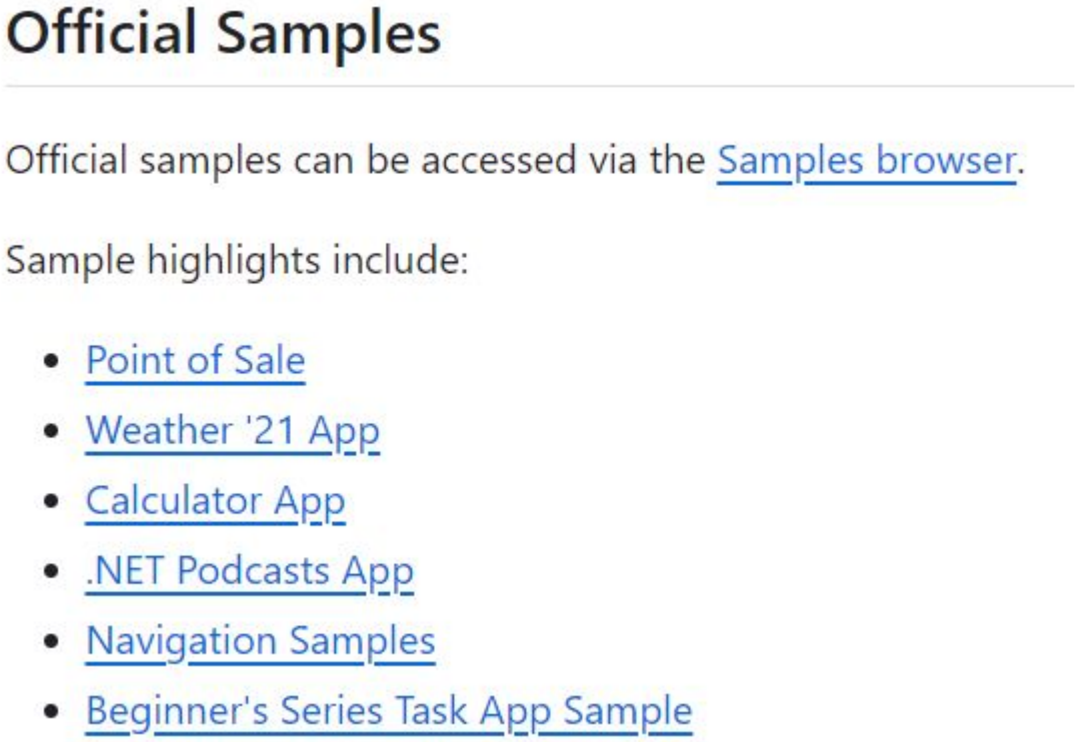
MVVM Tutorial

https://www.tutorialspoint.com/mvvm/index.htm



Learn from sample code

1. https://github.com/dotnet/maui-samples



Official Samples

Official samples can be accessed via the Samples browser.

Sample highlights include:

- Point of Sale
- Weather '21 App
- Calculator App
- .NET Podcasts App
- Navigation Samples
- Beginner's Series Task App Sample

Community Samples

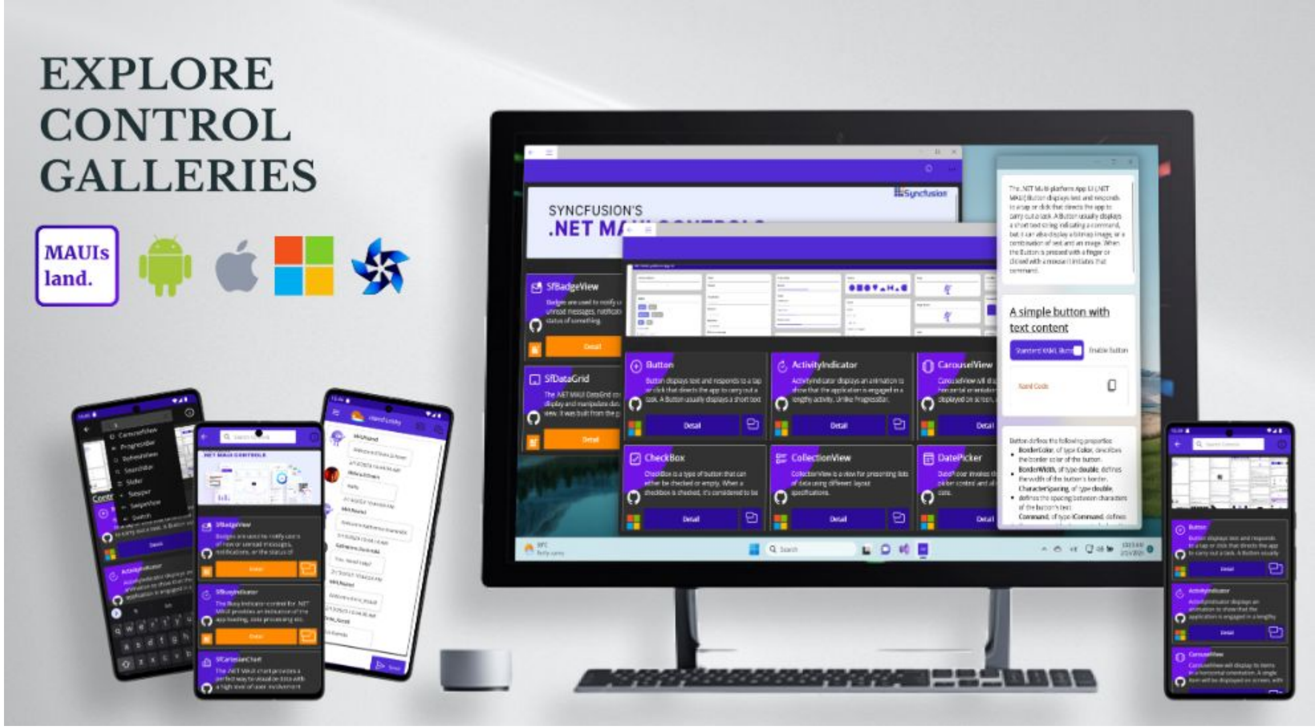
- .NET MAUI Showcase
- Awesome .NET MAUI

Clone this sample repo

1. **$git clone https://github.com/dotnet/maui-samples.git**
2. Open folder **9.0**
3. Investigate sample source codes

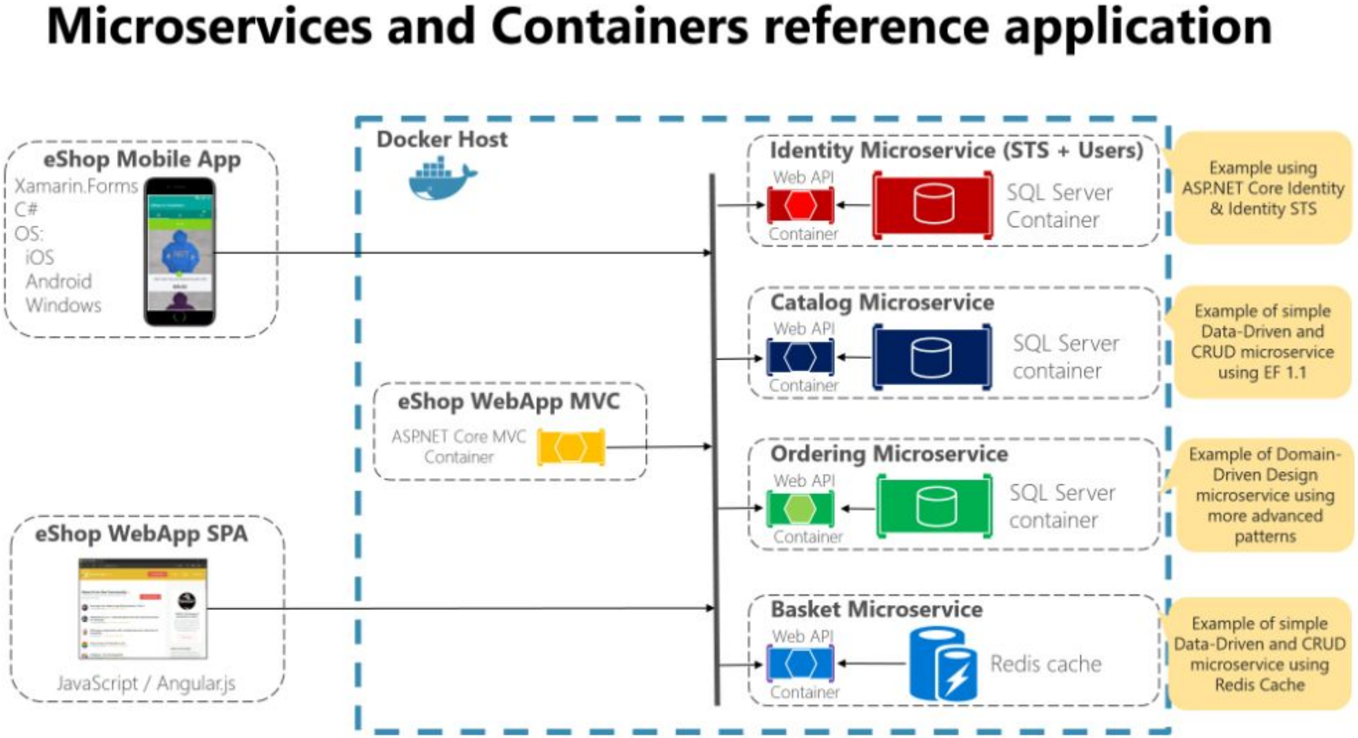
2. Maui showcase - Good looking samples

```
$ git clone https://github.com/jsuarezruiz/dotnet-maui-showcase.git
```



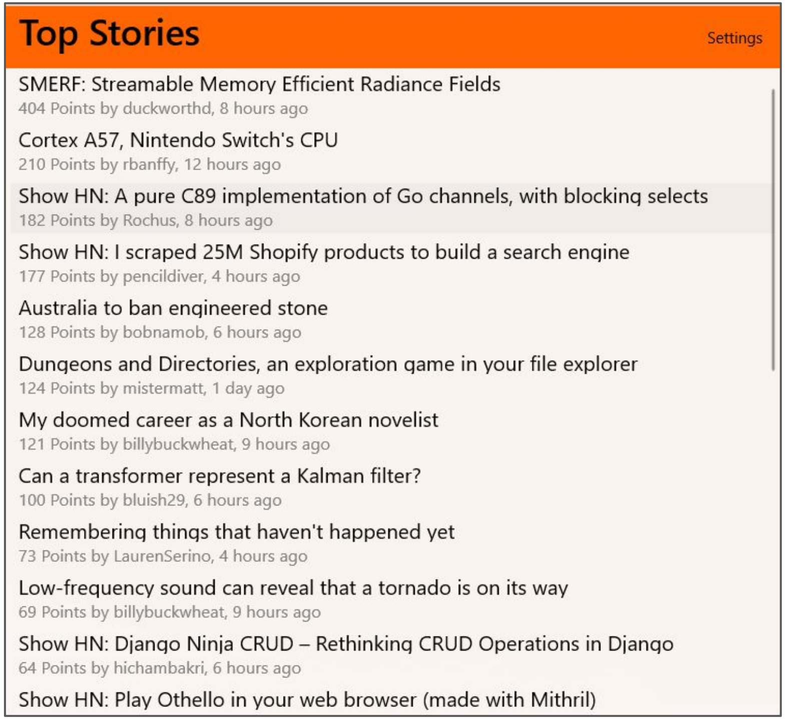
3. eShop mobile client

$ git clone https://github.com/dotnet-architecture/eshop-mobile-client.git


Microservices and Containers reference application
Docker Host
eShop Mobile App
Xamarin.Forms
C#
OS:
iOS
Android
Windows
eShop WebApp MVC
ASP.NET Core MVC
Container
eShop WebApp SPA
JavaScript / Angular.js
Identity Microservice (STS + Users)
Web API
Container
SQL Server
Container
Example using ASP.NET Core Identity & Identity STS
Catalog Microservice
Web API
Container
SQL Server
container
Example of simple Data-Driven and CRUD microservice using EF 1.1
Ordering Microservice
Web API
Container
SQL Server
container
Example of Domain-Driven Design microservice using more advanced patterns
Basket Microservice
Web API
Container
Redis cache
Example of simple Data-Driven and CRUD microservice using Redis Cache

4. Markup community toolkit samples

$ git clone https://github.com/CommunityToolkit/Maui.Markup.git



Top Stories

SMERF: Streamable Memory Efficient Radiance Fields
404 Points by duckworthd, 8 hours ago

Cortex A57, Nintendo Switch's CPU
210 Points by rbanffy, 12 hours ago

Show HN: A pure C89 implementation of Go channels, with blocking selects
182 Points by Rochus, 8 hours ago

Show HN: I scraped 25M Shopify products to build a search engine
177 Points by pencildiver, 4 hours ago

Australia to ban engineered stone
128 Points by bobnamob, 6 hours ago

Dungeons and Directories, an exploration game in your file explorer
124 Points by mistermatt, 1 day ago

My doomed career as a North Korean novelist
121 Points by billybuckwheat, 9 hours ago

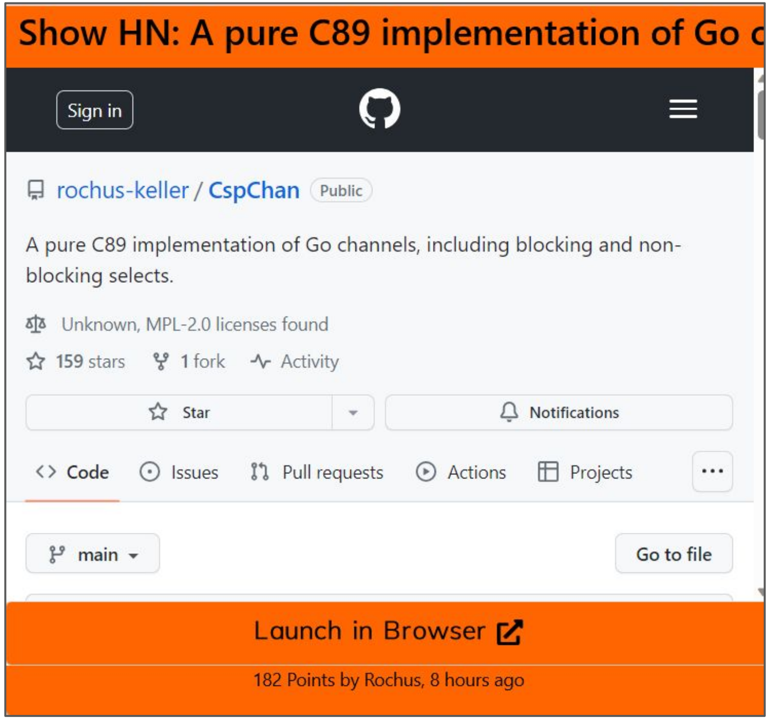
Can a transformer represent a Kalman filter?
100 Points by bluish29, 6 hours ago

Remembering things that haven't happened yet
73 Points by LaurenSerino, 4 hours ago

Low-frequency sound can reveal that a tornado is on its way
69 Points by billybuckwheat, 9 hours ago

Show HN: Django Ninja CRUD – Rethinking CRUD Operations in Django
64 Points by hichambakri, 6 hours ago

Show HN: Play Othello in your web browser (made with Mithril)



Show HN: A pure C89 implementation of Go c

rochus-keller / CspChan Public

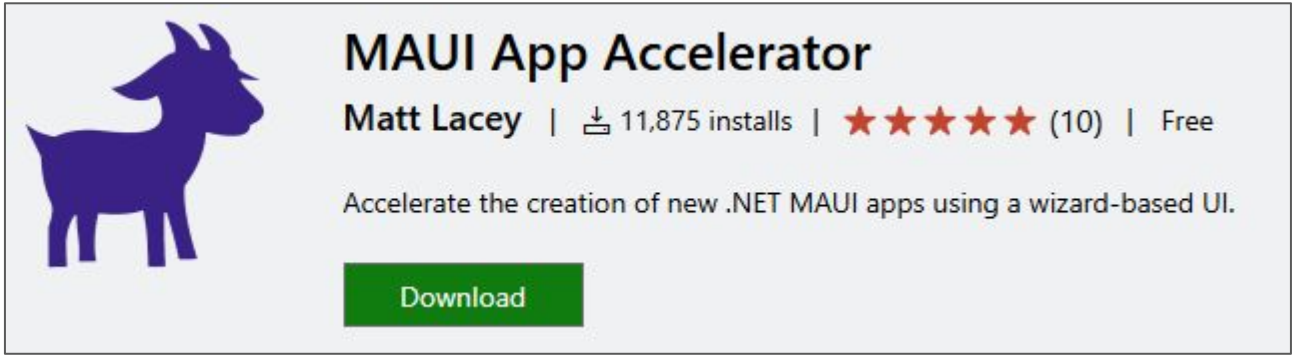
A pure C89 implementation of Go channels, including blocking and non-blocking selects.

Unknown, MPL-2.0 licenses found

159 stars 1 fork Activity

182 Points by Rochus, 8 hours ago

5. Template for project - MAUI App Accelerator

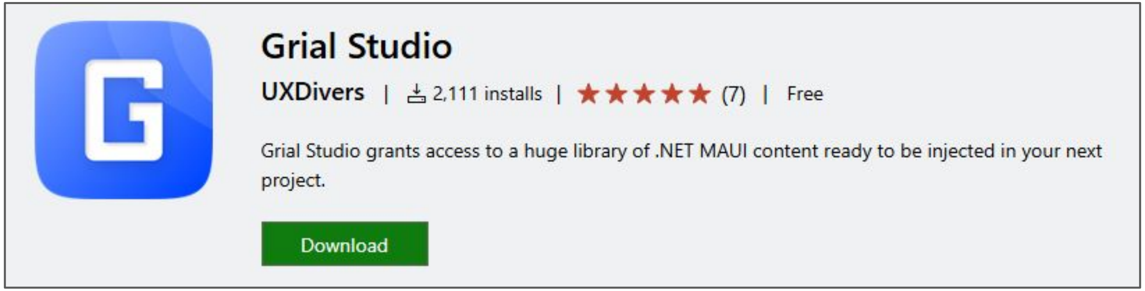


MAUI App Accelerator

Matt Lacey | 11,875 installs | ★★★★★ (10) | Free

Accelerate the creation of new .NET MAUI apps using a wizard-based UI.

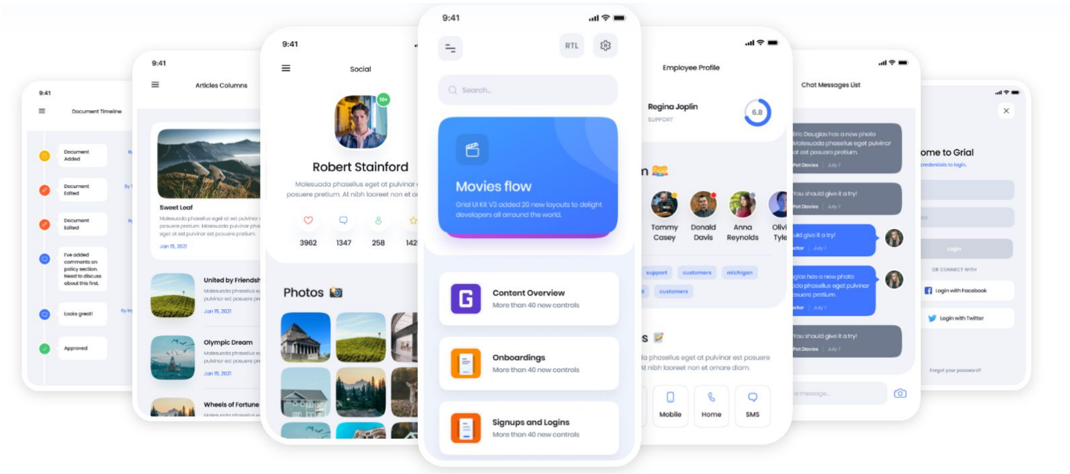
6. Inspired by commercial sample app - Grial Studio



Grial Studio

UXDivers | ⤓ 2,111 installs | ★★★★★ (7) | Free

Grial Studio grants access to a huge library of .NET MAUI content ready to be injected in your next project.



Getting started

https://learn.microsoft.com/en-us/windows/apps/windows-dotnet-maui/tutorial-csharp-ui-maui-toolkit