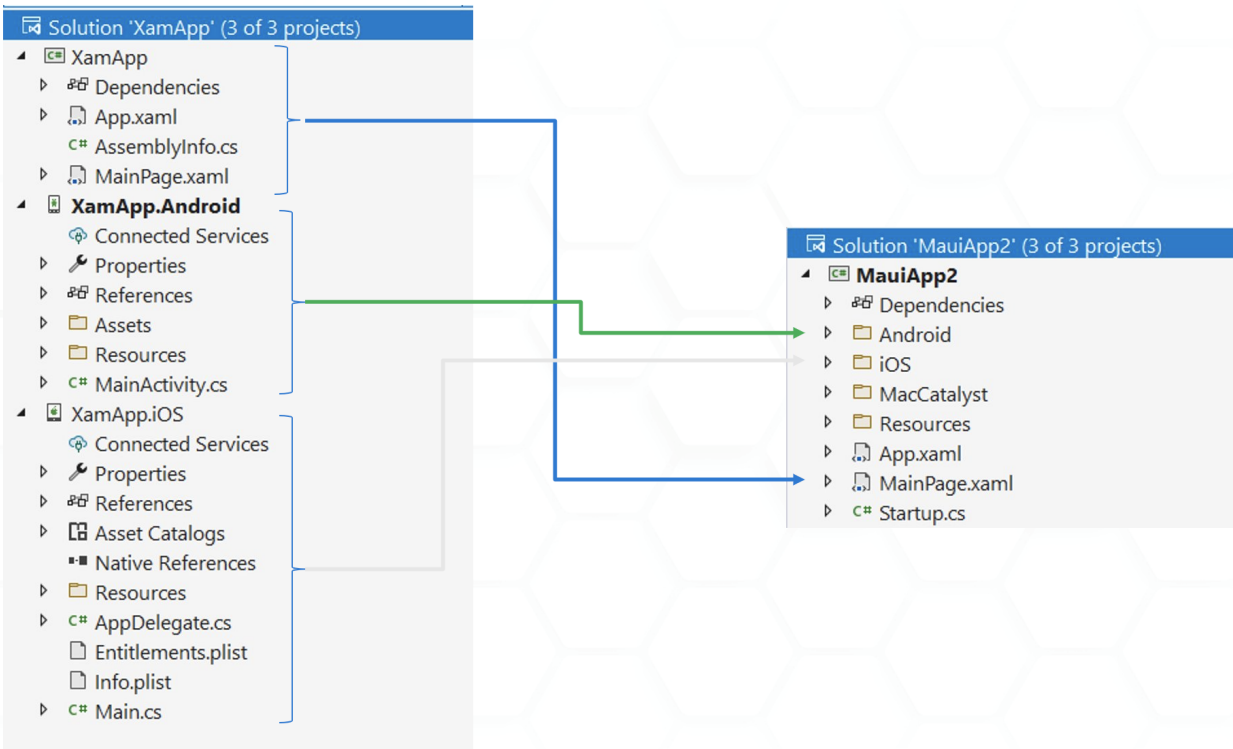


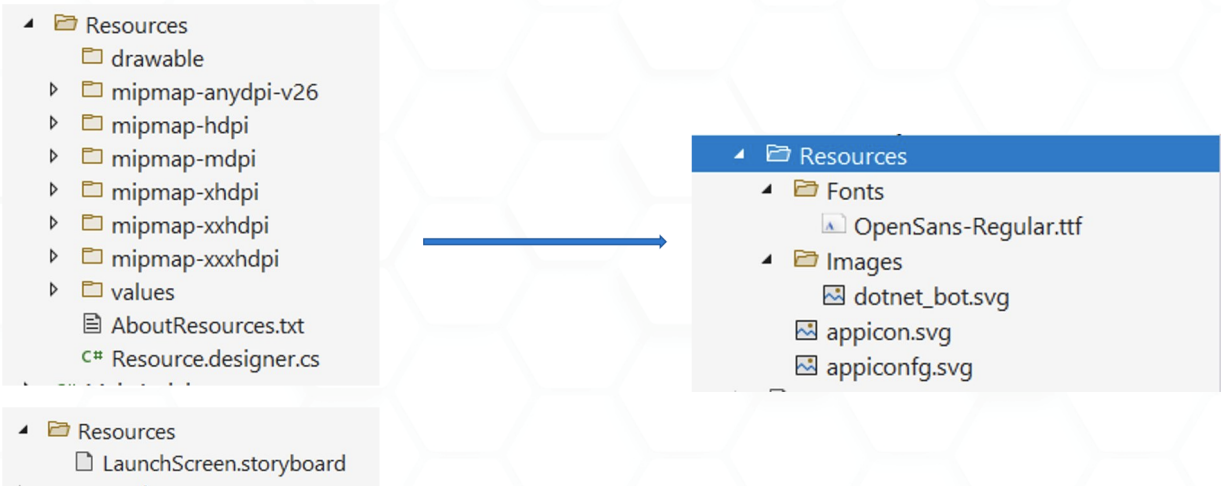
MAUI 进阶

MAUI 与 Xamarin 的对比

1. 通过一个项目进行维护， 更容易管理



2. 资源归一化



3. 精简的项目描述文件

之前在 Xamarin 的描述文件

```

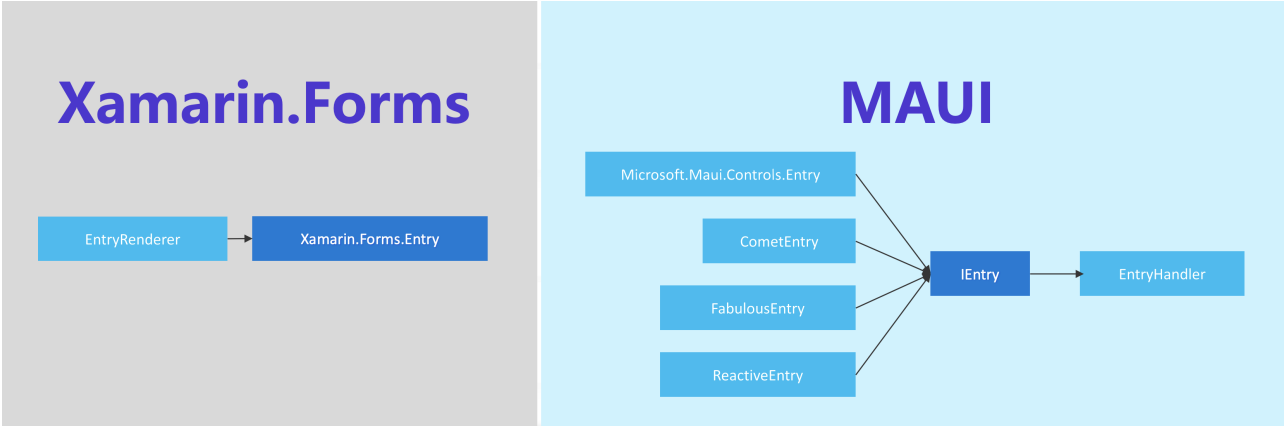
Xamarin.Android.csproj
C:\Users> explorer > source > repos > Xamarin > Xamarin > Xamarin.Android > Xamarin.Android.csproj
1 <?xml version="1.0" encoding="utf-8"?>
2 <Project DefaultTargets="Build" ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3   <PropertyGroup>
4     <Configuration Condition=" '$(Configuration)' == '' ">Debug</Configuration>
5     <Platform Condition=" '$(Platform)' == '' ">AnyCPU</Platform>
6     <ProjectGuid>{0F065D11-E9AB-4158-B258-293943582D77}</ProjectGuid>
7     <ProjectTypeGuids>{FFB8B077-5A72-4CE8-AF49-833B27855DCE};{FA1044C8-201F-11D0-BF68-00C04F796FC}</ProjectTypeGuids>
8     <TemplateGuid>{c955e55-cab5-42a1-8398-01506e8a570f}</TemplateGuid>
9     <OutputType>Library</OutputType>
10    <RootNamespace>Xamarin.Droid</RootNamespace>
11    <AssemblyName>Xamarin.Android</AssemblyName>
12    <Deterministic>true</Deterministic>
13    <AndroidApplication>true</AndroidApplication>
14    <AndroidResgenFile>Resources\Resource.designer.cs</AndroidResgenFile>
15    <AndroidResgenClass>Resource</AndroidResgenClass>
16    <AndroidManifest>Properties\AndroidManifest.xml</AndroidManifest>
17    <MonoAndroidResourcePrefix>Resources</MonoAndroidResourcePrefix>
18    <MonoAndroidAssetsPrefix>Assets</MonoAndroidAssetsPrefix>
19    <AndroidUseLatestPlatformSdk>false</AndroidUseLatestPlatformSdk>
20    <TargetFrameworkVersion>v11.0</TargetFrameworkVersion>
21    <AndroidEnableSGenConcurrent>true</AndroidEnableSGenConcurrent>
22    <AndroidUseAapt2>true</AndroidUseAapt2>
23    <AndroidHttpClientHandlerType>Xamarin.Android.Net.AndroidClientHandler</AndroidHttpClientHandlerType>
24    <NuGetPackageReportStamp>
25    </NuGetPackageReportStamp>
26  </PropertyGroup>
27  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Debug|AnyCPU' ">
28    <DebugSymbols>true</DebugSymbols>
29    <DebugType>portable</DebugType>
30    <Optimize>false</Optimize>
31    <OutputPath>bin\Debug</OutputPath>
32    <DefineConstants>DEBUG</DefineConstants>
33    <ErrorReport>prompt</ErrorReport>
34    <WarningLevel>4</WarningLevel>
35    <AndroidLinkMode>None</AndroidLinkMode>
36  </PropertyGroup>
37  <PropertyGroup Condition=" '$(Configuration)|$(Platform)' == 'Release|AnyCPU' ">
38    <DebugSymbols>true</DebugSymbols>
39    <DebugType>portable</DebugType>
40    <Optimize>true</Optimize>
41    <OutputPath>bin\Release</OutputPath>
42    <ErrorReport>prompt</ErrorReport>
43    <WarningLevel>4</WarningLevel>
44    <AndroidManagedSymbols>true</AndroidManagedSymbols>
45    <AndroidUseSharedRuntime>false</AndroidUseSharedRuntime>
46  </PropertyGroup>
47  <ItemGroup>
48    <Reference Include="Mono.Android" />
49    <Reference Include="System" />
50    <Reference Include="System.Core" />
51    <Reference Include="System.Xml.Linq" />
52    <Reference Include="System.Xml" />
53    <Reference Include="System.Numerics" />
54    <Reference Include="System.Numerics.Vectors" />
55  </ItemGroup>
56  <ItemGroup>
57    <PackageReference Include="Xamarin.Forms" Version="5.0.0.2032" />
58    <PackageReference Include="Xamarin.Essentials" Version="1.6.1" />
59  </ItemGroup>
60  <ItemGroup>
61    <Compile Include="MainActivity.cs" />
62    <Compile Include="Resources\Resource.designer.cs" />
63    <Compile Include="Properties\AssemblyInfo.cs" />
64  </ItemGroup>
65  <ItemGroup>
66    <None Include="Resources\AboutResources.txt" />
67    <None Include="Assets\AboutAssets.txt" />
68    <None Include="Properties\AndroidManifest.xml" />
69  </ItemGroup>
70  <ItemGroup>
71    <AndroidResource Include="Resources\values\styles.xml" />
72    <AndroidResource Include="Resources\values\colors.xml" />
73    <AndroidResource Include="Resources\mipmap-anydpi-v26\icon.xml" />
74    <AndroidResource Include="Resources\mipmap-anydpi-v26\icon_round.xml" />
75    <AndroidResource Include="Resources\mipmap-hdpi\icon.png" />
76    <AndroidResource Include="Resources\mipmap-hdpi\launcher_foreground.png" />
77    <AndroidResource Include="Resources\mipmap-mdpi\icon.png" />
78    <AndroidResource Include="Resources\mipmap-mdpi\launcher_foreground.png" />
79    <AndroidResource Include="Resources\mipmap-xhdpi\icon.png" />
80    <AndroidResource Include="Resources\mipmap-xhdpi\launcher_foreground.png" />
81    <AndroidResource Include="Resources\mipmap-xxhdpi\icon.png" />
82    <AndroidResource Include="Resources\mipmap-xxhdpi\launcher_foreground.png" />
83    <AndroidResource Include="Resources\mipmap-xxxhdpi\icon.png" />
84    <AndroidResource Include="Resources\mipmap-xxxhdpi\launcher_foreground.png" />
85  </ItemGroup>
86  <ItemGroup>
87    <Folder Include="Resources\drawable\" />
88  </ItemGroup>
89  <ItemGroup>
90    <ProjectReference Include="..\Xamarin\Xamarin.csproj">
91      <Project>{026E2A6C-67F1-44FE-8DA3-1C1C043B002}</Project>
92      <Name>Xamarin</Name>
93    </ProjectReference>
94  </ItemGroup>
95  <Import Project="$(MSBuildExtensionsPath)\Xamarin\Android\Xamarin.Android.CSharp.targets" />
96 </Project>

```

现在的描述文件

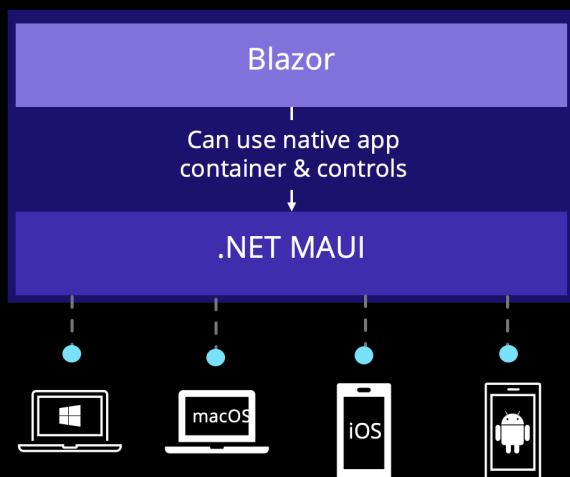
```
<?xml version="1.0" encoding="utf-8"?>
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <TargetFrameworks>net6.0-android;net6.0-ios;net6.0-maccatalyst</TargetFrameworks>
    <TargetFrameworks Condition="$([MSBuild]::IsOSPlatform('windows')) and '$(MSBuildRuntimeType)' == 'Full'">$(TargetFrameworks);net6.0-windows10.0.19041</TargetFrameworks>
    <OutputType>Exe</OutputType>
    <RootNamespace>listviewapp</RootNamespace>
    <UseMaui>true</UseMaui>
    <SingleProject>true</SingleProject>
    <ImplicitUsings>disable</ImplicitUsings>
    <EnablePreviewMsixTooling>true</EnablePreviewMsixTooling>
    <!-- Display name -->
    <ApplicationTitle>listviewapp</ApplicationTitle>
    <!-- App Identifier -->
    <ApplicationId>com.companyname.listviewapp</ApplicationId>
    <!-- Versions -->
    <ApplicationVersion>1</ApplicationVersion>
    <!-- Required for C# Hot Reload -->
    <UseInterpreter Condition=" '$(Configuration)' == 'Debug' " True</UseInterpreter>
    <SupportedOSPlatformVersion Condition=" '$(TargetFramework)' == 'net6.0-ios' " 14.2</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition=" '$(TargetFramework)' == 'net6.0-maccatalyst' " 14.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition=" '$(TargetFramework)' == 'net6.0-android' " 21.0</SupportedOSPlatformVersion>
    <SupportedOSPlatformVersion Condition=" '$(TargetFramework.Contains('-windows')) " 10.0.17763.0</SupportedOSPlatformVersion>
    <TargetPlatformMinVersion Condition=" '$(TargetFramework.Contains('-windows')) " 10.0.17763.0</TargetPlatformMinVersion>
  </PropertyGroup>
  <ItemGroup>
    <!-- App Icon -->
    <MauiIcon Include="Resources\appicon.svg" ForegroundFile="Resources\appiconfg.svg" Color="#512BD4" />
    <!-- Splash Screen -->
    <MauiSplashScreen Include="Resources\appiconfg.svg" Color="#512BD4" />
    <!-- Images -->
    <MauiImage Include="Resources\Images\*" />
    <!-- Custom Fonts -->
    <MauiFont Include="Resources\Fonts\*" />
  </ItemGroup>
  <ItemGroup Condition=" '$(TargetFramework.Contains('-windows')) " >
    <!-- Required - WinUI does not yet have buildTransitive for everything -->
    <PackageReference Include="Microsoft.WindowsAppSDK" Version="1.0.0" />
    <PackageReference Include="Microsoft.Graphics.Win2D" Version="1.0.0.30" />
  </ItemGroup>
  <PropertyGroup Condition=" '$(TargetFramework.Contains('-windows')) " >
    <OutputType>WinExe</OutputType>
    <RuntimeIdentifier>win10-x64</RuntimeIdentifier>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Clancey.Comet" Version="0.3.450-beta" />
    <PackageReference Include="Reloadify3000" Version="1.0.8" />
  </ItemGroup>
</Project>
```

4. 重新绘制的 UI



MAUI 融入到更多的开源和商业项目支持当中

Hybrid apps with Blazor & .NET MAUI



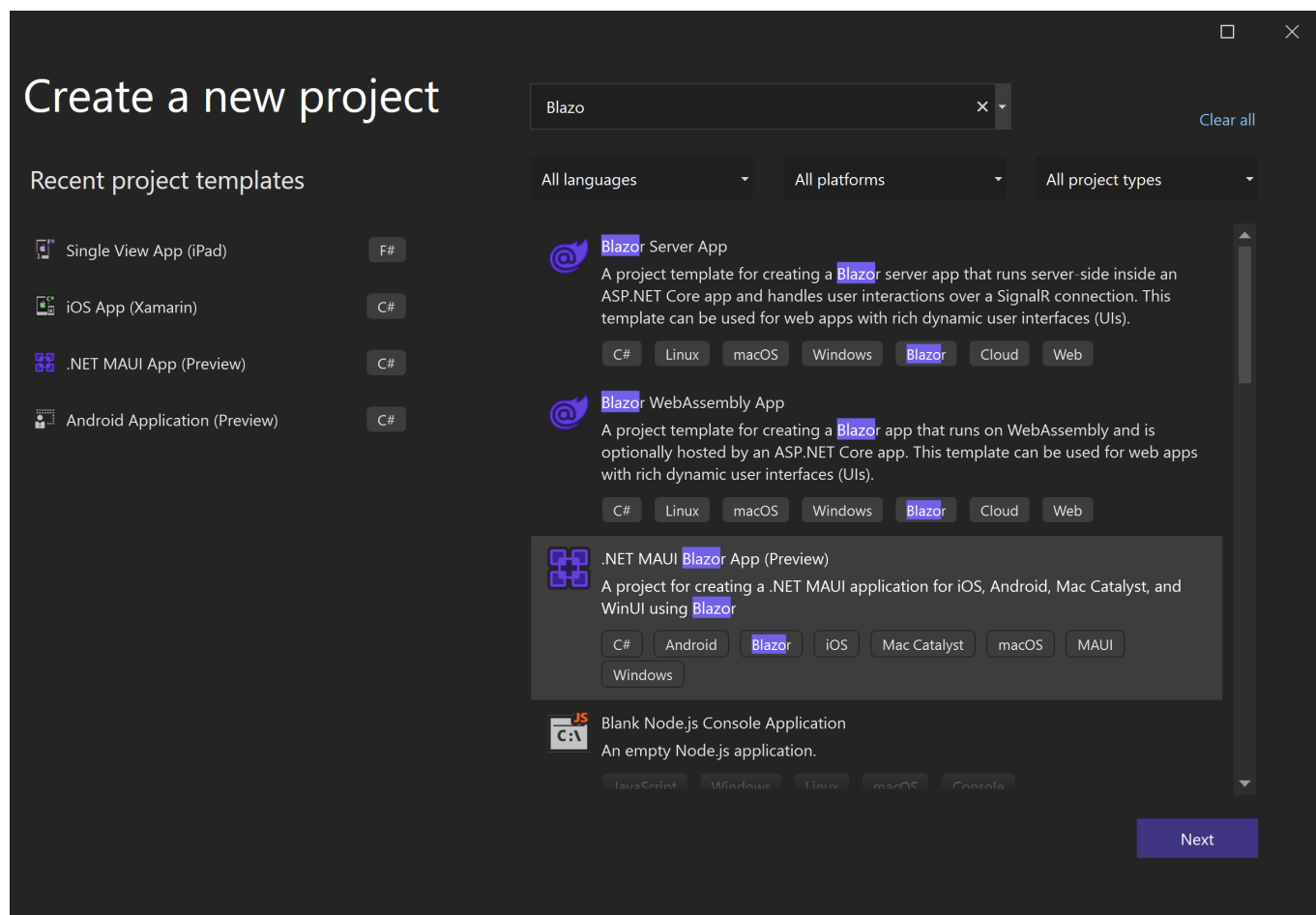
跨原生和 Web 重用 UI 组件

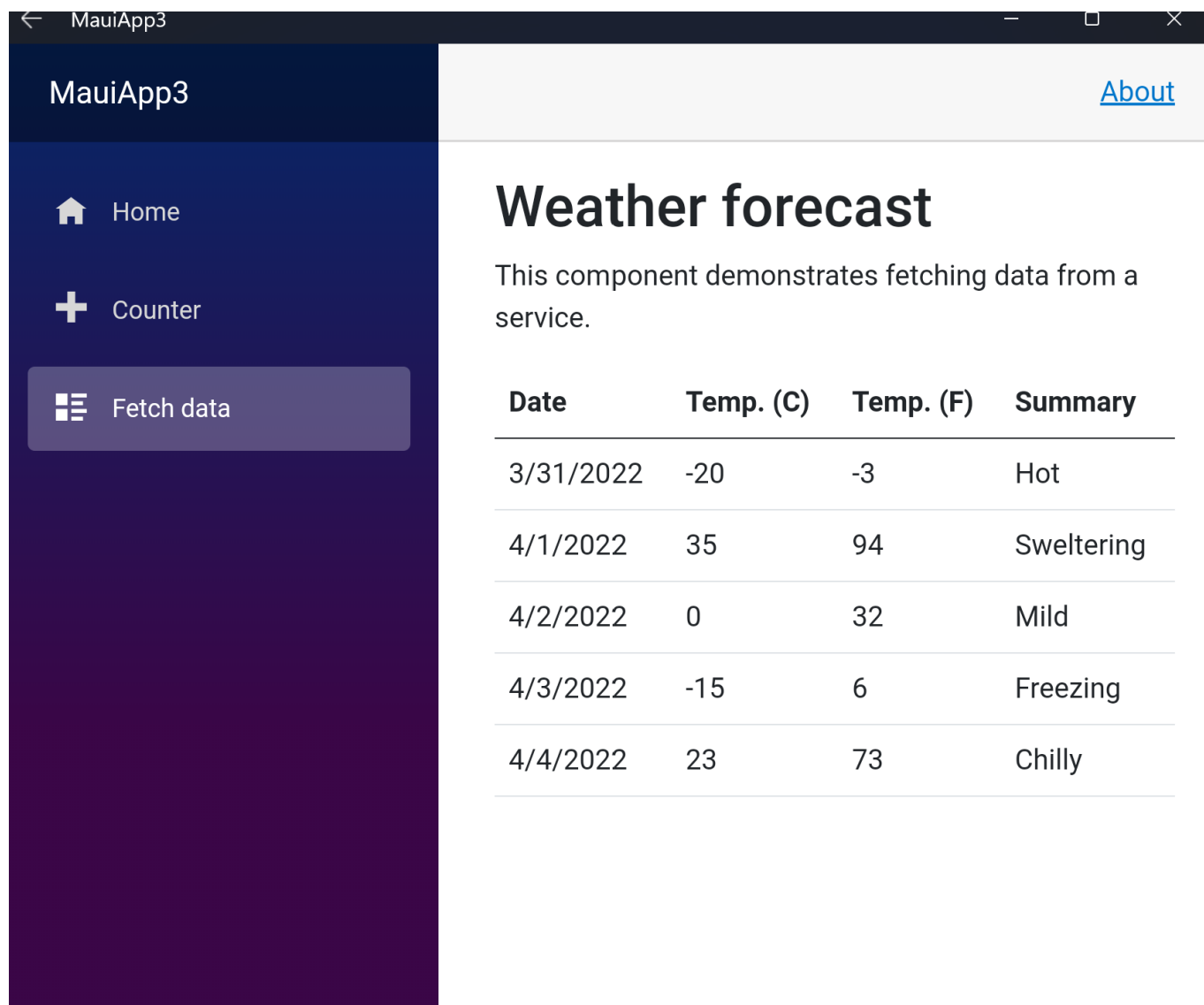
建立在 .NET 多平台应用程序 UI 之上

原生应用容器和嵌入式控件

这是通过 Blazor 完成混合场景的开发，提高了用户体验

打开 Visual Studio 2022 可以选择 MAUI Blazor 应用模版，快速创建应用，你可以作为一名页面前端开发人员来完成具备原生性能的跨平台应用操作





基于 MAUI 的变体也有很多，以 MAUI 做底层也有了不同的衍生跨平台产品 Uno Platform，Avalonia 就是其中之一。

了解 Uno Platform 你也可以访问该网址，获取更多信息 <https://platform.uno/>

了解 Avalonia UI 你也可以访问该网址，获取更多信息 <https://github.com/AvaloniaUI/Avalonia>

Comet 来了

基于 MAUI 声明式 UI - Comet来了

Comet 是一种编写跨平台 UI 解决方案, 通过声明式的 UI 来完成 UI 构建。

了解该项目请访问：<https://github.com/dotnet/comet>

什么是命令式UI

UI的更新是由程序员使用代码主动刷新，UI与数据并无必然的映射关系，这种我们称之为命令式UI

什么是声明式UI

UI的更新并非由程序员使用代码来主动刷新，而是由后面隐藏机制来负责维护UI的刷新，UI与数据有映射关系，这种我们就称之为声明式UI

前端开发，原生移动应用开发 iOS/Android ， 桌面端开发都统一属于命令式 UI。

这种命令式UI的模式，是存在一些问题的，表现在：

1. UI 维护成本高， 量大易出错
2. 刷新频繁，性能影响大
3. UI 与数据同步不一致性

声明式 UI 与命令式 UI 相反

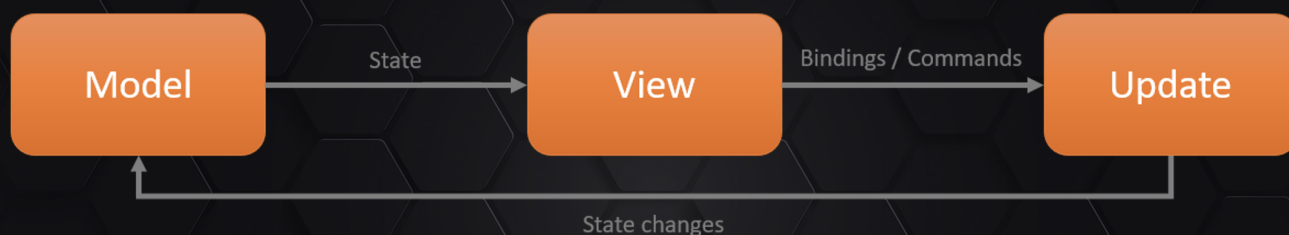
1. UI 一次定义省心，出错机会少
2. 关注数据刷新，降低对 UI 编码量
3. UI 与数据同步一致

现在声明式 UI 用得最好的是 Flutter 以及 ReactNative 还有 SwiftUI , 现在声明式 UI 也来到了 MAUI。

```
public class MyPage : View {
    public MyPage() {
        Body = body;
    }
    View body () => new Text("Hello World");
}
```

新的设计模式

什么是MVU



MVU(Model View Update) MVU 是一种功能响应式编程、架构设计模式。在 MVU 中 UI 是不可变的。所以每次你想更新一个属性时，你都必须重建你的 UI。不变性是函数式编程的特性。C# 9.0 后支持函数式编程范式。UI 和业务登录使用相同的语言（例如 C#）编写。数据流是单向的。它非常适合应用程序的热重载。您不需要学习 XAML 来开发 UI，而是用一种语言开发完整的应用程序。MVU 的 MAUI 实现