**.NET Maui | 2 April 2025**

* <https://learn.microsoft.com/en-us/training/paths/build-apps-with-dotnet-maui/>
* <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-9.0>
* <https://dotnet.microsoft.com/en-us/learn/maui>
* What is it?
  + An open-source app platform from Microsoft for building modern, performant, and multi-platform iOS, android, MacOS, and windows apps with C# and .NET from a *single shared codebase*.
* ActivityIndicator (spinning circles, loading)
  + iOS—UIActivityIndicator
  + Android—ProgressBar
* Slider (adjustment bar, like for brightness)
  + iOS—Slider(?)
  + Android—SeekBar

A diagram of a computer

AI-generated content may be incorrect.

A diagram of a device

AI-generated content may be incorrect.

* .NET for iOS—Ahead of Time (AOT)
* .NET for Android—Just in Time (JIT) and Ahead of Time (AOT)

**Our First App**

* Project Structure—everything will be in one project with single shared codebase
  + MonkeyFinder
    - Model—Classes
    - View—UI; MainPage, DetailsPage
    - ViewModel—provides interaction logic
      * Model View, View Model (MVVM)
      * Model View Controllers (MVC)
        + MVVM and MVC are architectures
    - Resources—Fonts,…
    - Platforms—holds android/iOS/windows-specific features
  + A screenshot of a project structure

    AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

* Nuget Packages
  + CommunityToolkit.Mvvm
    - Only if using MVVM architecture pattern
* Add Model folder
  + Add class—Monkey.cs
    - Make it a public class
  + Public properties
    - Name, Location, Details, string Image, int Population, double Latitude/Longitude
  + Close Monkey.cs
* Add Services folder
  + Add MonkeyService class
    - Make it public
  + Finish making the MonkeyService class
    - Close MonkeyService.cs
* Design base ViewModel class
  + Will inherit from observableojbect
    - Base class written for observing the properties of…
  + Add ViewModel folder (will automatically fall under Services folder)
  + Add View Folder (automatically fall under Services folder)
    - Move MainPage.xaml under view, drag and drop
    - A screenshot of a computer program

      AI-generated content may be incorrect.
  + Write base ViewModel
    - Add “BaseViewModel.cs” class in ViewModel Folder
      * Make public
      * Make it inherit from “BaseViewModel : ObservableObject”
      * Make empty constructor
      * Make properties to be inherited from…
    - Will have to make the BaseViewModel class as partial when adding [ObservableProperty] attribute
    - using CommunityToolkit.Mvvm.ComponentModel; should populate
* Create MonkeysViewModel class in ViewModel folder
  + Make public partial
  + Inherit from BaseViewModel
* MauiProgram.cs
  + Register services
* View 🡪 MainPage.xaml
  + Delete the ScrollView
  + Add: xmlns:model=”clr-namespace:MonkeyFinder.Model”
  + Add: xmlns:model=”clr-namespace:MonkeyFinder.ViewModel”
  + Add: x:DataType=”viewmodel:MonkeysViewModel”
  + Add: Title=”{Binding Title}”
  + And close it >
* Add image to resources->images
  + Right click images, add existing file
* MainPage.xaml
  + Lines 9-
* MainPage.xaml.cs
* MonkeysViewModel
  + Add GetClosestMonkey()
* View Folder
  + Add 🡪 new item 🡪 .NET MAUI ContentPage (XAML)
    - DetailsPage.xaml
* MainPage.xaml
  + Add Frame.GestureRecognizers
    - Lines 20-
  + Go to MonkeysViewModel.cs and add new function to go to details page
    - Async Task GoToDetailsAsync
  + AppShell.xaml.cs
    - Routing.RegisterRoute….
  + Back to MainPAge.xaml and finish Frame.GestureRecognizers
* Back to Details Page
  + Line 5: xmlns:viewmodel="clr-namespace:MonkeyFinderv2.ViewModel"
  + Change Title
* Viewmodel folder 🡪 Add class
  + MonkeyDetailsViewModel.cs
    - Add using MonkeyFinder.Model;
* DetailsPage.xaml
  + Add lines 6-7
  + Add scrollview, change everything under vertical stack layout
* DetailsPage.xaml.cs
* MauiProgram.cs
  + Register detailspage and MonkeyDetailsViewModel
    - Transient means each time you need a separate object
* MonkeyDetailsViewModel.cs
  + Add IMap service
  + Add monkeydetailsviewmodel (imap map) constructor
  + Add openmapasync()
* DetailsPage.xaml
  + Add “Show on map” button after image
* Test the app, should be able to get monkeys, get closest monkey, find closest monkey, and get map directions to a monkey