.NET Development

Placeholder for .NET topics

- Blazor
  - URL: Passing variables
- MAUI - IOS Emulator / Building without MAC
- MAUI - physical IPhone deployment (without Mac)
- MAUI - Connect IPhone to debug
- Visual studio project version
- Azure
  - Pipeline - building offline packages
- IIS Rewrite URL - basepath

Blazor

Blazor

URL: Passing variables

Add the path

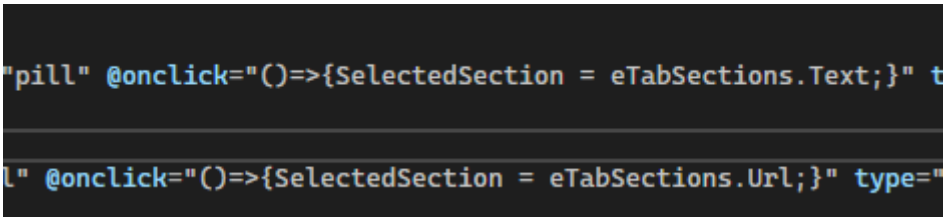
```
@page "/reader/{Id:int}"
```

Bind the parameter

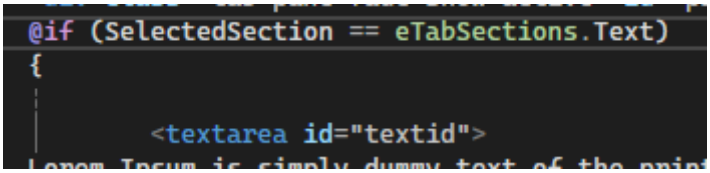
```
[Parameter]
public string Id { get; set; }
```

It should be said that via the easy mix of code between the server and the GUI - this approach to pass variables might not the best way.
But it could be used to implement an external facing URL, for example.

You'd then assign a variable with the **@onclick** - where the @-sign denotes it will be run on the server.



It then allows you to write code that renders different HTML.

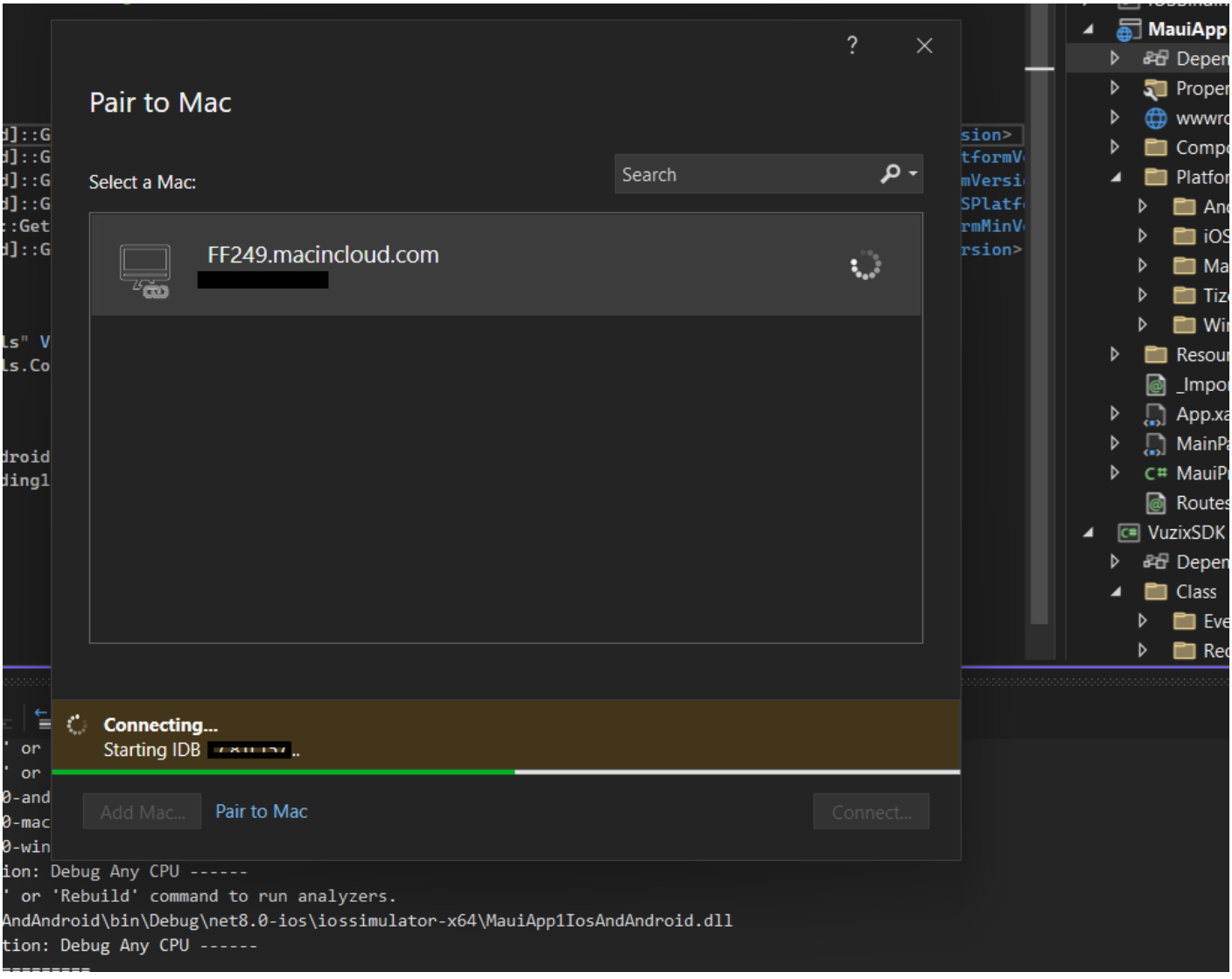


MAUI - IOS Emulator / Building without MAC

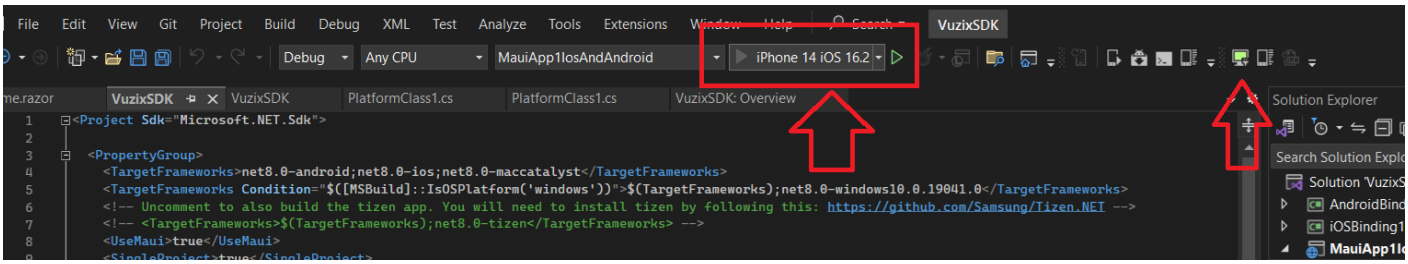
It seems nearly impossible to run an Iphone emulator without a Mac or Iphone device.
You could invest in a Mac / Macbook or reason to get something cheap refurbished for a few builds.

But for my development, http://www.macincloud.com was sufficient and quite fast to continue my work.
Just make sure to get the option to SSH - so you can pair your "Mac in the cloud" to your Visual studio.

Use the credentials that come with the subscription.



Once it is paired, you then can debug on an Iphone Emulator

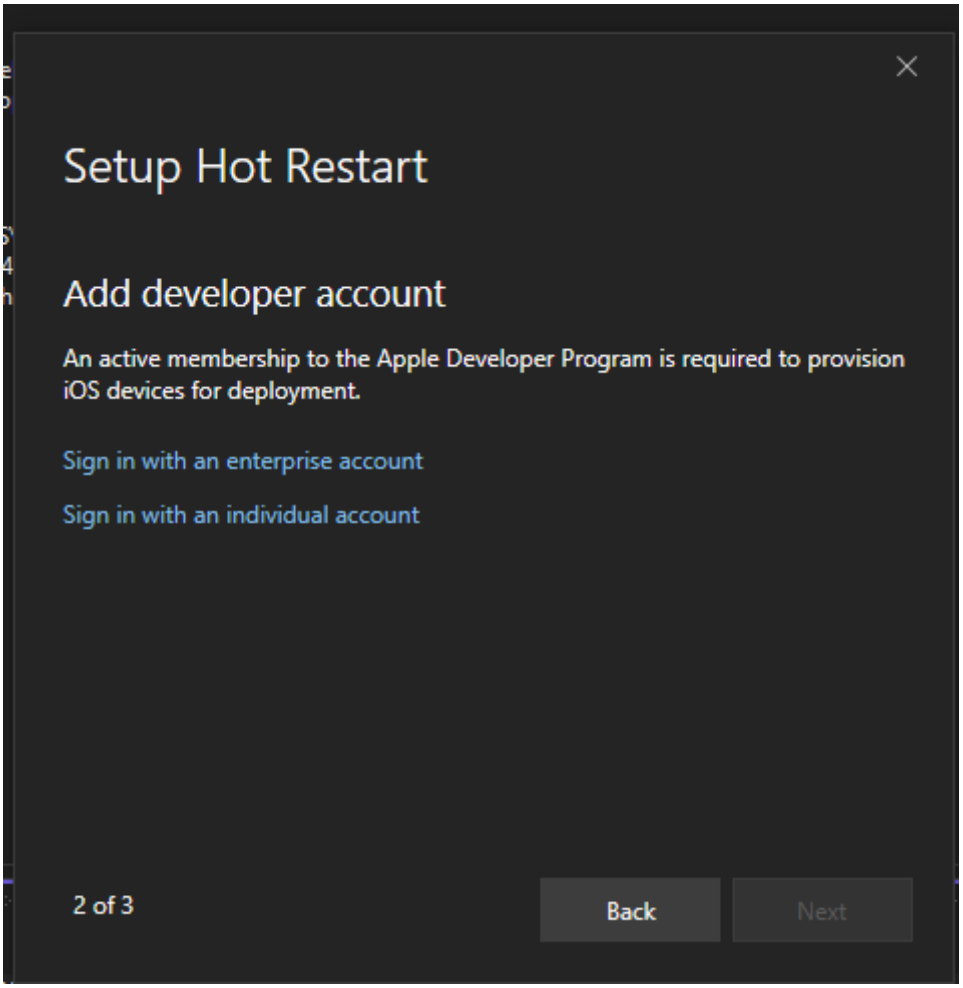


MAUI - physical IPhone deployment (without Mac)

In some cases the MacInCloud isn't sufficient: to test BLE devices you do need a physical device, for example.

In that case, you can get a cheap Iphone SE2020 or simular - unfortunately you cannot link that phone over RDP (your macincloud) to its USB to deploy via Visual Studio.

1) You'll have to install Itunes and get a developer account
2) You'll have to activate your developer account (which is 99€ annually)



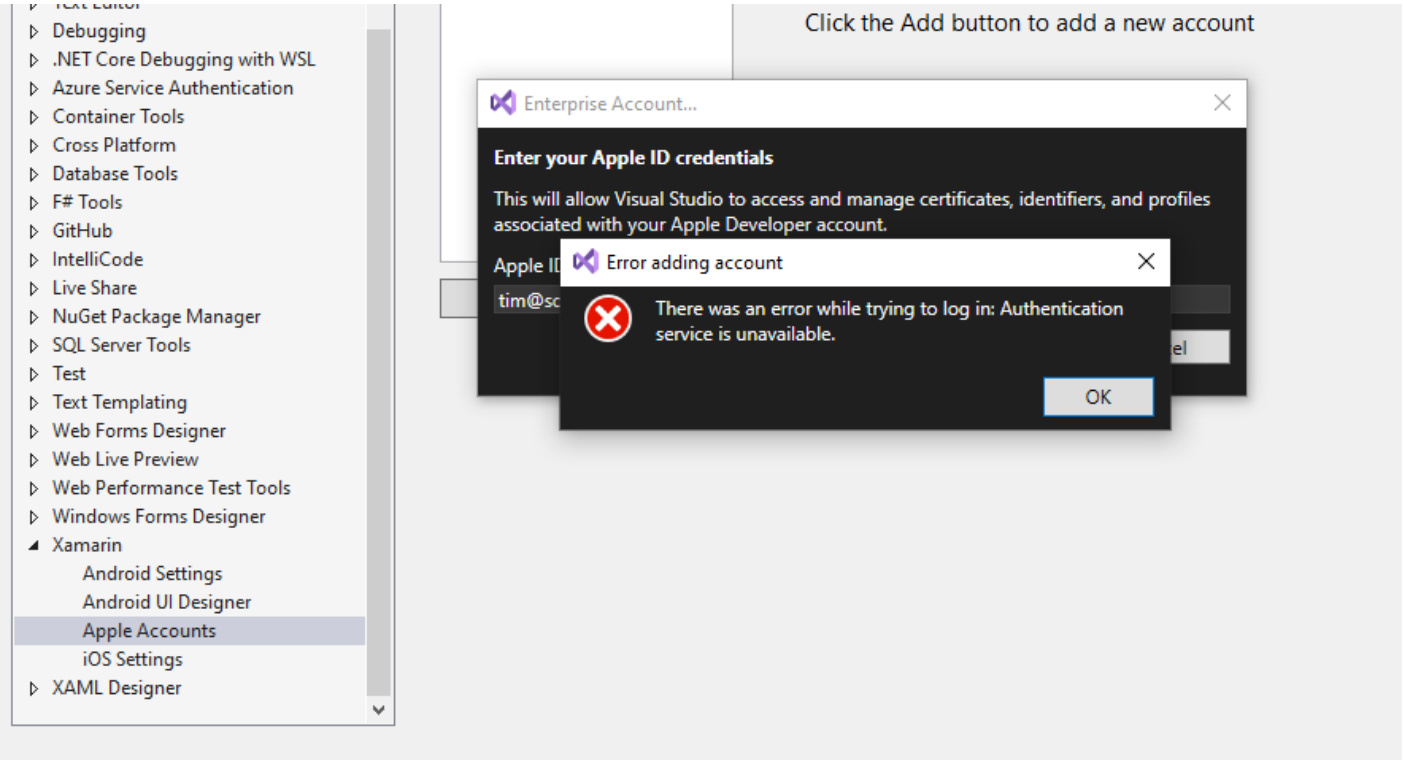
MAUI - Connect IPhone to debug

According to Microsoft, an Itunes installation on Windows is enough to target the phone.

However, after some days of debugging (and setting up a virtual machine to generate keys instead) - it turns out that it's a functionality that's broken. Either by Apple or Microsoft.

At time of writing (November 2024) - it hasn't been resolved.

https://developercommunity.visualstudio.com/t/Cannot-add-Apple-Account-VS-2002-Enter/10773193



Visual studio project version

Define a post-build to create a version file

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net8.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
    <GeneratePackageOnBuild>False</GeneratePackageOnBuild>
    <Version>1.2.3</Version>
  </PropertyGroup>
  <Target Name="GetVersion" AfterTargets="PostBuildEvent">
    <GetAssemblyIdentity AssemblyFiles="$(TargetPath)">
      <Output TaskParameter="Assemblies" ItemName="AssemblyInfo" />
    </GetAssemblyIdentity>
    <PropertyGroup>
      <VersionInfo>%(AssemblyInfo.Version)</VersionInfo>
    </PropertyGroup>
    <!--And use it after like any other variable:-->
    <Message Text="VersionInfo = $(VersionInfo)" Importance="high" />
  </Target>
  <Target Name="PostBuild" AfterTargets="PostBuildEvent">
    <!--
    <Exec Command="call $(SolutionDir)BuildScripts\PostBuild.bat $(TargetPath) $(VersionInfo)" />
    -->
    <Exec Command="echo $(VersionInfo)&gt;$(TargetDir)\version.txt" />
  </Target>
</Project>
```

Azure

Azure

Pipeline - building offline packages

I had trouble linking packages in the pipeline which blocked building solutions.
The shortest path I could find, was to update configuration to look into my defined folder.
And get a zipfile from an online resource, and extract it into this folder.

Update the configuration

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <config>
    <add key="globalPackagesFolder" value=".\Packages" />
    <add key="repositoryPath" value=".\Packages" />
  </config>
  <packageSources>
    [....]
```

Define your pipeline task

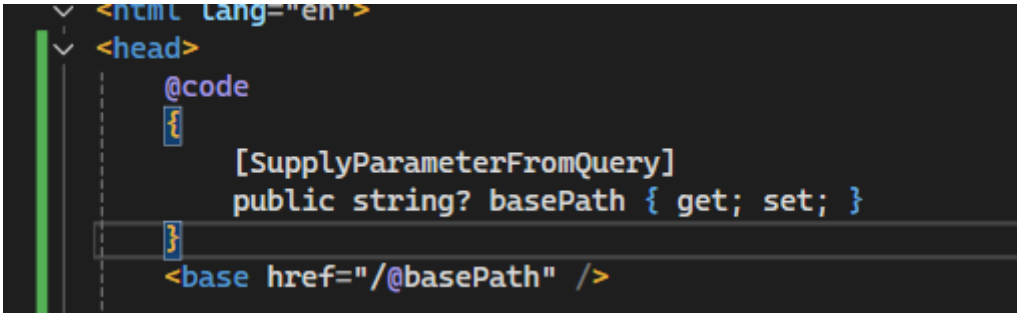
```
- task: PowerShell@2
  inputs:
    targetType: 'inline'
    script: |
      Invoke-WebRequest 'https://dl.dropboxusercontent.com/scl/fi/XXXXXXX8vl&dl=0' -OutFile
'$(Build.ArtifactStagingDirectory)/NuGetPackages.zip'
- task: ExtractFiles@1
  inputs:
    archiveFilePatterns: '$(Build.ArtifactStagingDirectory)/NuGetPackages.zip'
    destinationFolder: '.\Packages\NuGetPackages'
```

```
cleanDestinationFolder: true
overwriteExistingFiles: false
```

IIS Rewrite URL - basepath

When you write a rewrite URL, and there is a subfolder, you will end up that the target application might not know how to manage the relative paths.

1) Write in the App.razor some code to receive the BasePath



2) In the rewrite rule append the basepath

Name:

Laklijn test HTTP

Match URL

Requested URL:

Matches the Pattern

Using:

Regular Expressions

Pattern:

(.*)laklijn/(.*)

Test pattern...

Ignore case

Conditions

Server Variables

Action

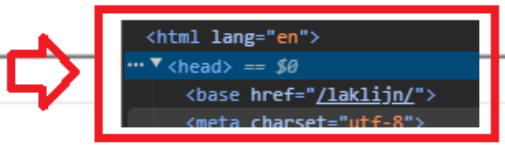
Action type:

Rewrite

Action Properties

Rewrite URL:

http://localhost:35/{R:2}?basePath=laklijn/



Append query string

Log rewritten URL

Stop processing of subsequent rules