Blazor Hybrid – Multiple App Design

The reason for this prototype is to allow the Android, iOS apps to be hosted in the apps stores. Thus customers can download these apps for free. Then once the app is downloaded, the user can add configuration information to connect to a Power-Flex Runtime API. The Power-Flex Runtime API can have the clients download the apps the user is allowed to run and provide application user interfaces that are allowed to be modified without having to worry about side loading applications. Side loading an application is an administrative nightmare.

A screenshot of a computer

Description automatically generated

.NET Projects

* BlazorShared
  + Is the shell that exposes interfaces to Native apps
  + Handles App Settings / Configurations to connect to Power-Flex apps
  + Downloads Assemblies from Power-Flex
* MauiBlazorHybrid
  + This is the native app entry point for Android, iOS / MacCatalyst
* WebBlazorHybridWasm
  + This is the web app entry point
* WpfBlazorHybrid
  + This is the Windows app entry point
* PluginSourceAPI
  + This is the simulated Power-Flex Runtime API that can be used to retrieve application assemblies
* AppDT
  + This is the plugin based app for Doc-Trak functionality. None of the other projects know anything about this project. Thus when I add routing and plugin entry points to this project and the source shows up in the other projects, it has proven the concept.



Each app has an entry point that directly talks to the BlazorShared (Shell). The shell then can navigate to the App’s page. In this case AppDT component1 page.

The app starts and currently runs the entry point. In the future the shell will start with the UI, but for the demo this works. This shows an Android emulator (on the left), a web page (on the top right), and a window’s WPF app (on the bottom right).

A screenshot of a computer

Description automatically generated

I now navigate to the Shell’s Counter page.

A screenshot of a computer

Description automatically generated

Last I clicked the Navigate button and the AppDT component1 page is displayed. This is a route that isn’t referenced directly from any of these projects until the assembly is loaded at runtime.

A screenshot of a computer

Description automatically generated