# NMUG 10/20/17 Xamarin.Android + MVVMCross

<https://developer.xamarin.com/guides/android/>

<https://www.mvvmcross.com/documentation/>

#### MVVM

Model–view–viewmodel (MVVM) is a software architectural pattern. MVVM facilitates a separation of development of the graphical user interface – be it via a markup language or GUI code – from development of the business logic or back-end logic (the data model).

<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>

#### Inversion of Control | Dependency Injection

In software engineering, inversion of control (IoC) is a design principle in which custom-written portions of a computer program receive the flow of control from a generic framework. - <https://en.wikipedia.org/wiki/Inversion_of_control>

The Dependency Injection pattern is a specialization of the Inversion of Control pattern. The Dependency Injection pattern uses a builder object to initialize objects and provide the required dependencies to the object. - <https://msdn.microsoft.com/en-us/library/ff921087.aspx>

## Visual Studio 2017 Project

#### *Description*

Create an Android application that allows Android versions Kitkat and higher to view DoJ press releases. Application should simulate logging a user in, presenting user with a master/details view listing press release titles and individual press release contents.

https://www.justice.gov/api/v1/press\_releases.json?pagesize=2

1. Create new Xamarin.Android Solution
   1. Android project structure

* Activity
* Fragment
* Resources
  + Layouts

1. Add netstandard2.0 project
2. Add package target fallback
3. Add MVVMCross nugets
   1. MVVMCross (all projects)
   2. MVMCross AppCompat (Android project)
4. Add view model stubs
5. Update activities and fragments
6. Add models
   1. Generate models from JSON result @ <https://quicktype.io/?l=cs&r=json2csharp>
   2. Demo extracting class to new file because it’s the bomb
7. Add service stubs
   1. Add NewtonSoft json dependency
8. Implement HttpClientService
   1. Add MVVMCross.Plugins.Messenger nugget to notify httprequest start/finish (busy indicator)
   2. Inject IMvxMessenger in constructor
   3. Add message models
9. Implement PressReleaseService
   1. Inject IHttpClient in constructor
10. Register service interfaces and implementations with the IoC container
    1. Add shared project App and register dependencies
    2. Add droid project Setup
       1. Add MVVMCross.Support.Design dependency for text input layout
       2. Override protected methods
11. Demo Xamaridea to create layouts
    1. Add texteditinputlayouts on login page
    2. Show drawer layout on main page
    3. Add recycler view on master page
       1. Add template
    4. Add webview on details page
    5. Add toolbar
12. Populate view models
13. Update activities/fragments
14. Update view models to prepare for bindings
15. Update fragment bindings