# Cross Platform Mobile Application Development using Xamarin and Azure

Richard Taylor
Technical Lead/Sr. Software Developer
Logical Advantage
@rightincode
rtaylor@rightincode.com
http://www.rightincode.com
http://www.logicaladvantage.com

# Who am I?

Web/Mobile Application Development

Huntersville, NC

Co-Organizer of Modern Devs Charlotte

Organizer of Charlotte Xamarin Developers

@rightincode / http://www.rightincode.com


@LogicalAdv

http://www.logicaladvantage.com

Technical Lead/Sr. Software Developer

Charlotte, NC

# Goals of this Talk

- Introduce Xamarin and Xamarin Forms
- Introduce Azure App Service and Mobile Apps
- Demonstrate using an Azure App Service Mobile App as a backend service for a Xamarin Forms Application

# What is Xamarin?

- Allows developers to deliver native Android, iOS, and Windows applications
  - Creates native user interfaces, provides native API access, and delivers native performance on each target platform
- Allows developers to leverage their existing C# skills to build mobile applications
- Allows developers to build a common codebase that can be shared between each platform target

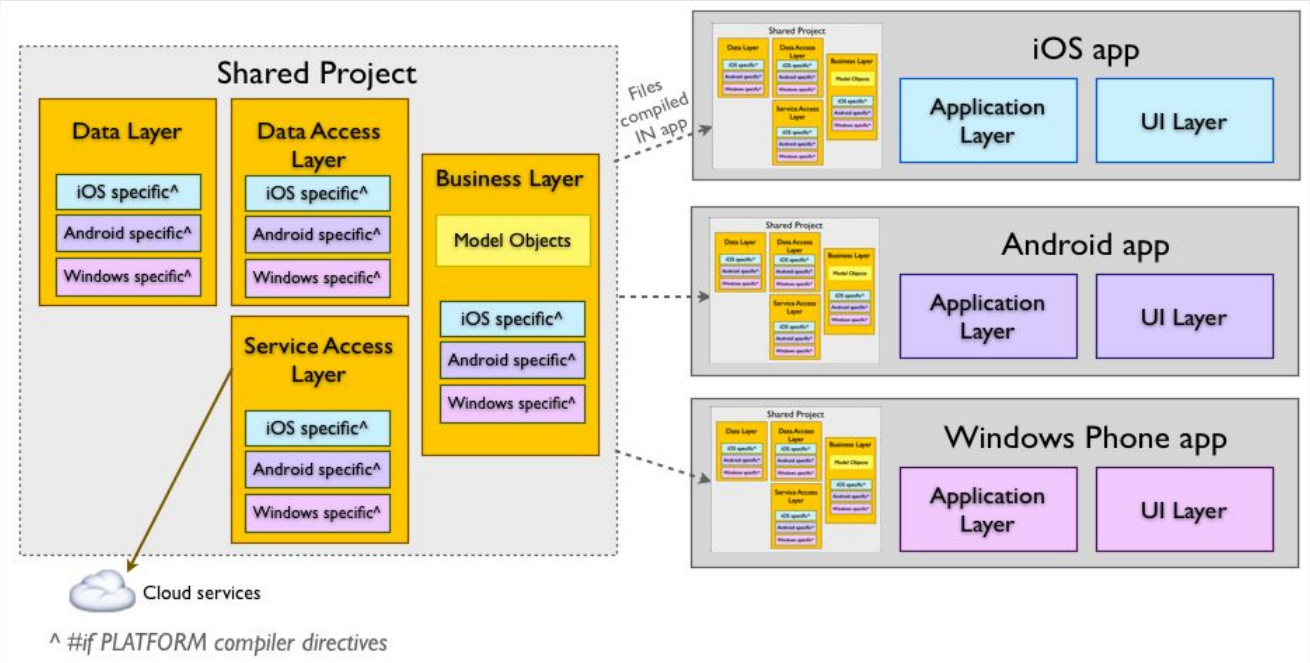
# Xamarin - Sharing Code

- Shared Projects
- Portable Class Libraries
- .NET Standard Libraries

# Xamarin - Sharing Code

- Shared Projects
  - Allows a developer to place code in a common location that can be shared between the platform targets
  - Compiler directives are used to include/exclude platform-specific functionality for each platform target
  - During the build process, the code in the shared project is included in each of the platform target assemblies (there is no output assembly for the shared project)

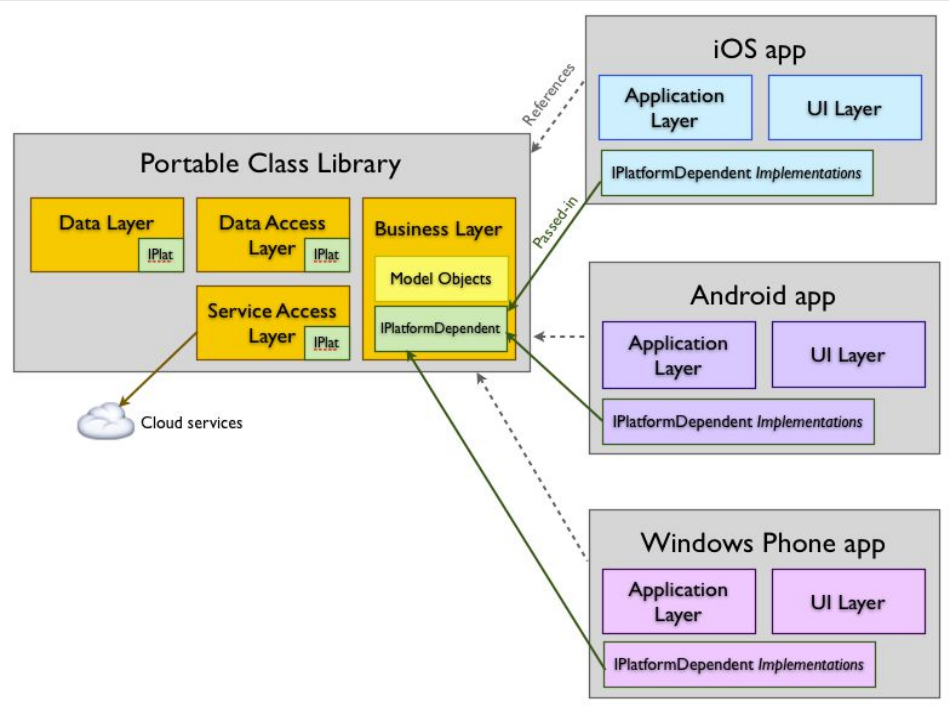
# Xamarin - Sharing Code (contd.)



# Xamarin - Sharing Code (contd.)

- Portable Class Library Projects
  - Allows a developer to place code in a common location that can be shared between the platform targets
  - PCL’s are referenced by the platform targets (there is an output assembly)
  - PCL’s cannot contain any platform-specific code
  - PCL’s have a profile that describes which features are supported (typically the broader the profile the smaller the number of available features)

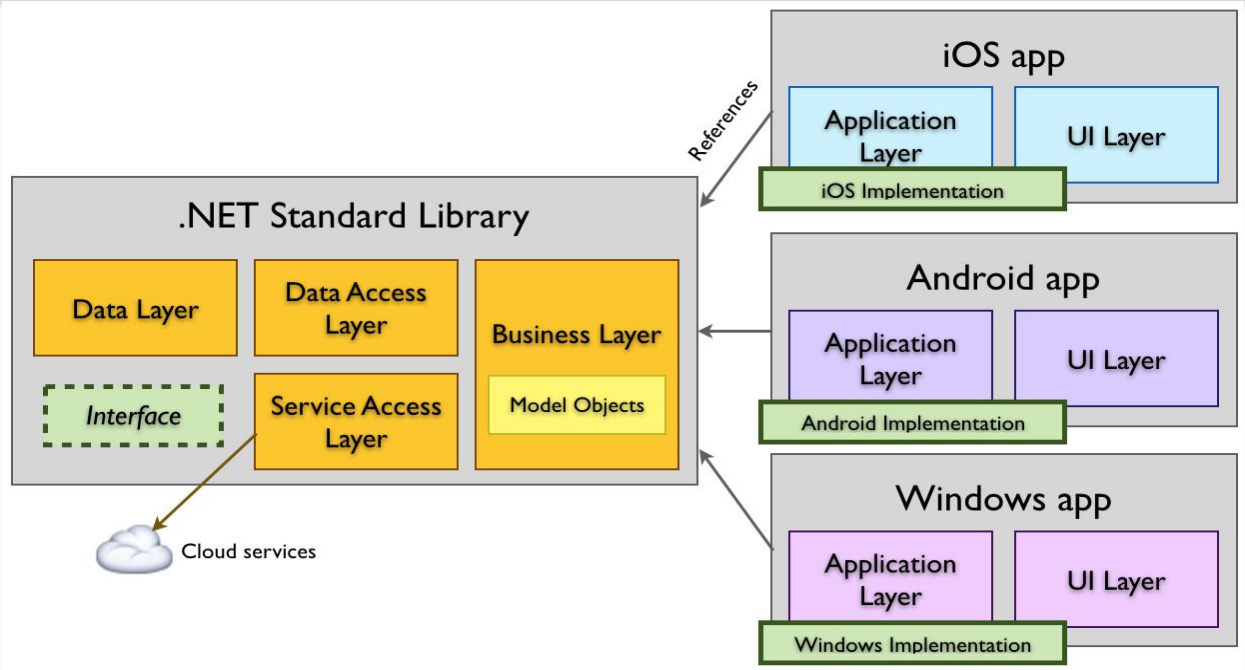
# Xamarin - Sharing Code (contd.)



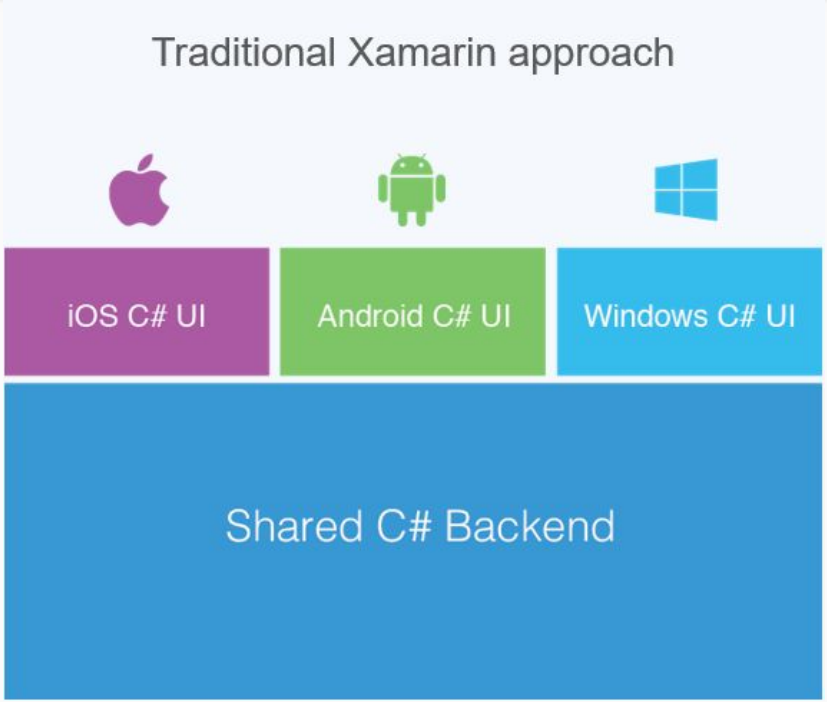
# Xamarin - Sharing Code (contd.)

- .NET Standard Libraries
  - Allows a developer to place code in a common location that can be shared between the platform targets
  - .NET Standard Libraries are referenced by the platform targets (there is an output assembly)
  - .NET Standard Libraries cannot contain any platform-specific code
  - .NET Standard Libraries have a larger surface area (available features) than PCL's
  - .NET Standard Libraries have a uniform API for all .NET Platforms

# Xamarin - Sharing Code (contd.)



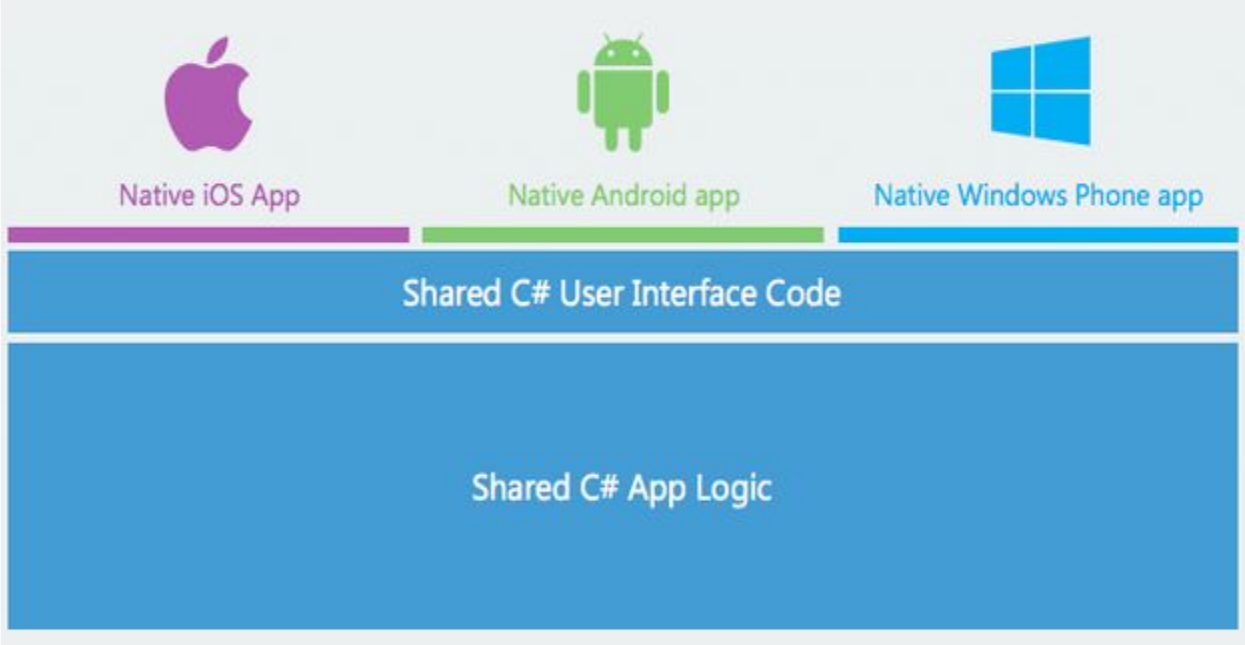
# Xamarin - Sharing Code (contd.)



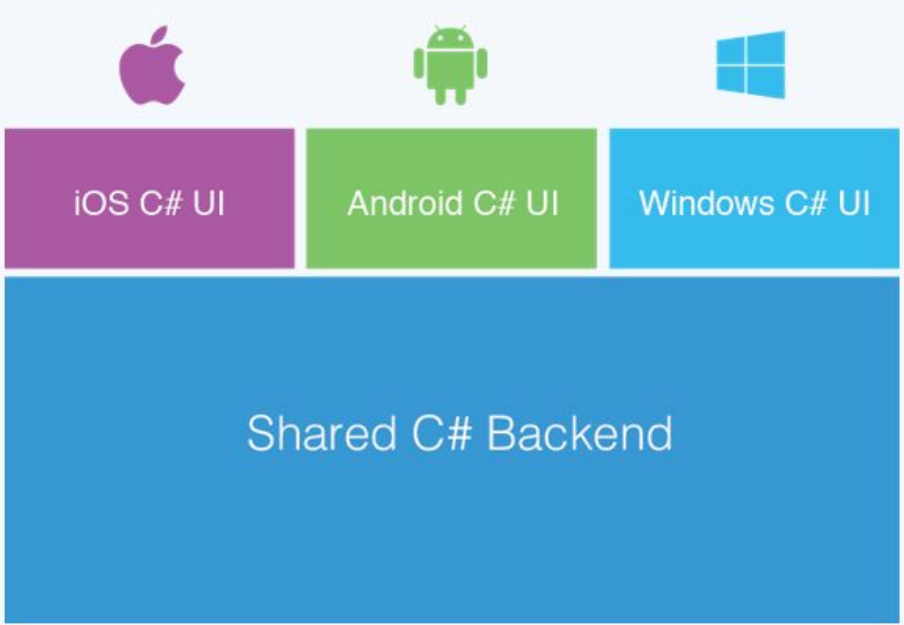
# Xamarin Forms

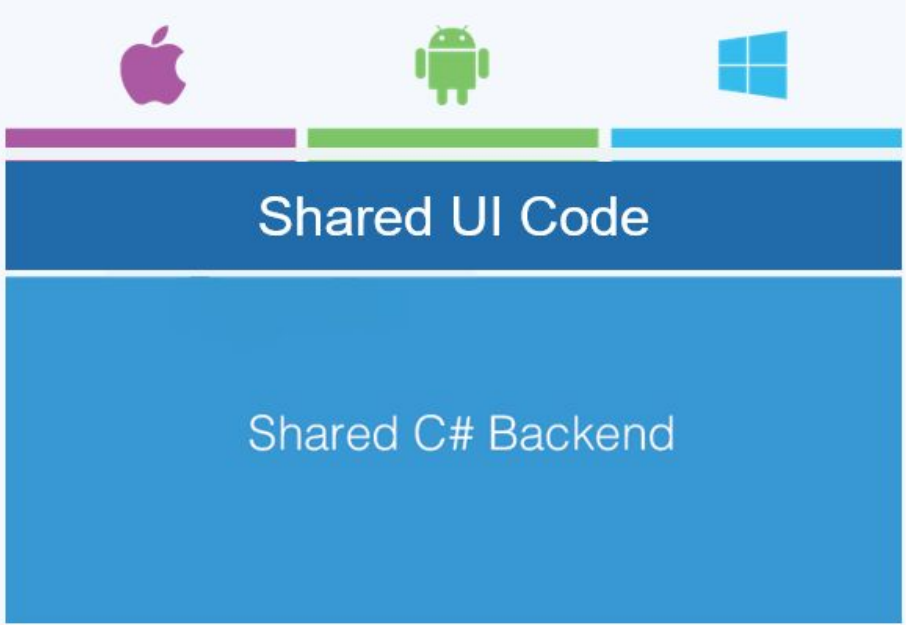
- Allows building of native UI's for iOS, Android, & Windows
- The UI's can be built using C#, XAML, or both
- Screens are represented by pages
- Pages contain various views (controls) that define the UI
- Pages and their views are rendered as native UI elements
- By connecting these views to shared backend code, we have a fully native iOS, Android, and Windows application built with shared C# code.
- Based on the application and technical design, we can achieve over 96% code reuse across platforms

# Xamarin Forms (contd.)



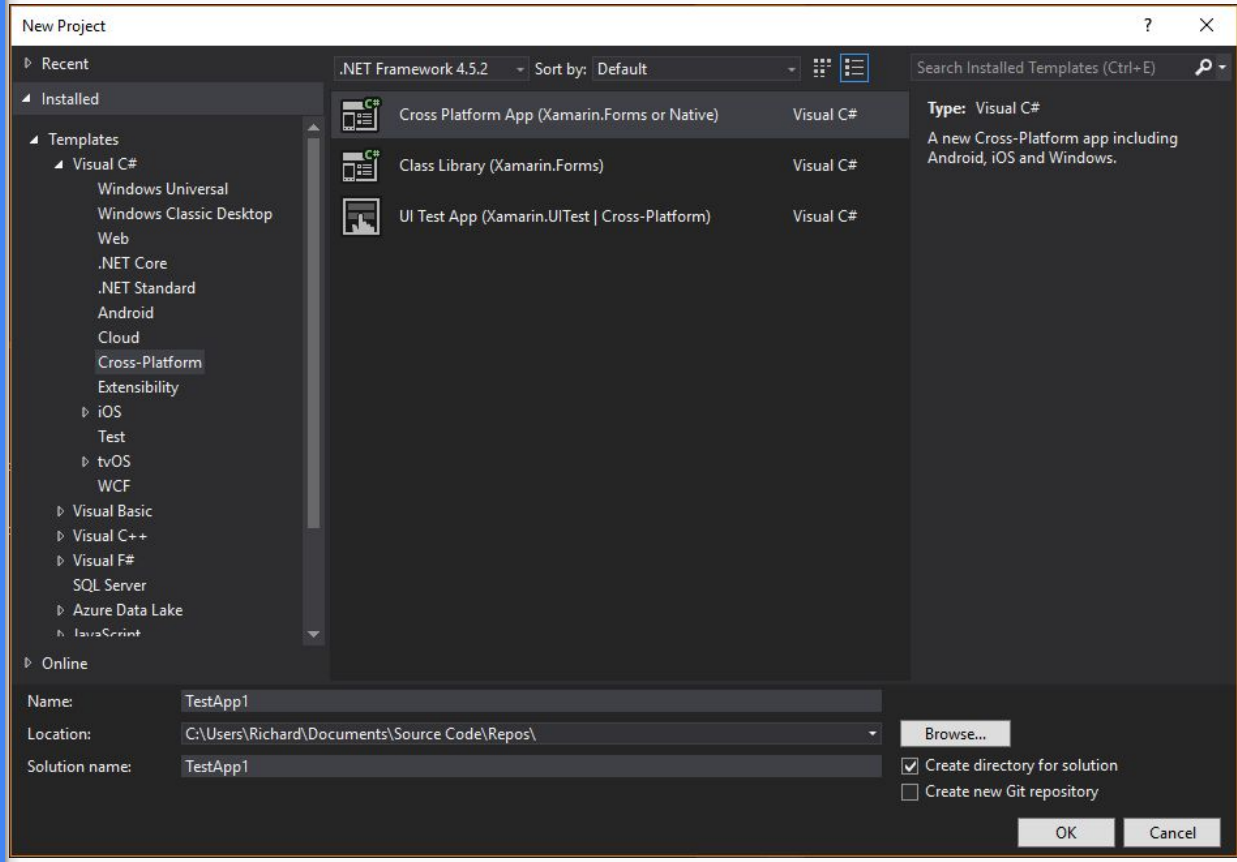
# Xamarin <-> Xamarin Forms



# Xamarin Forms Project

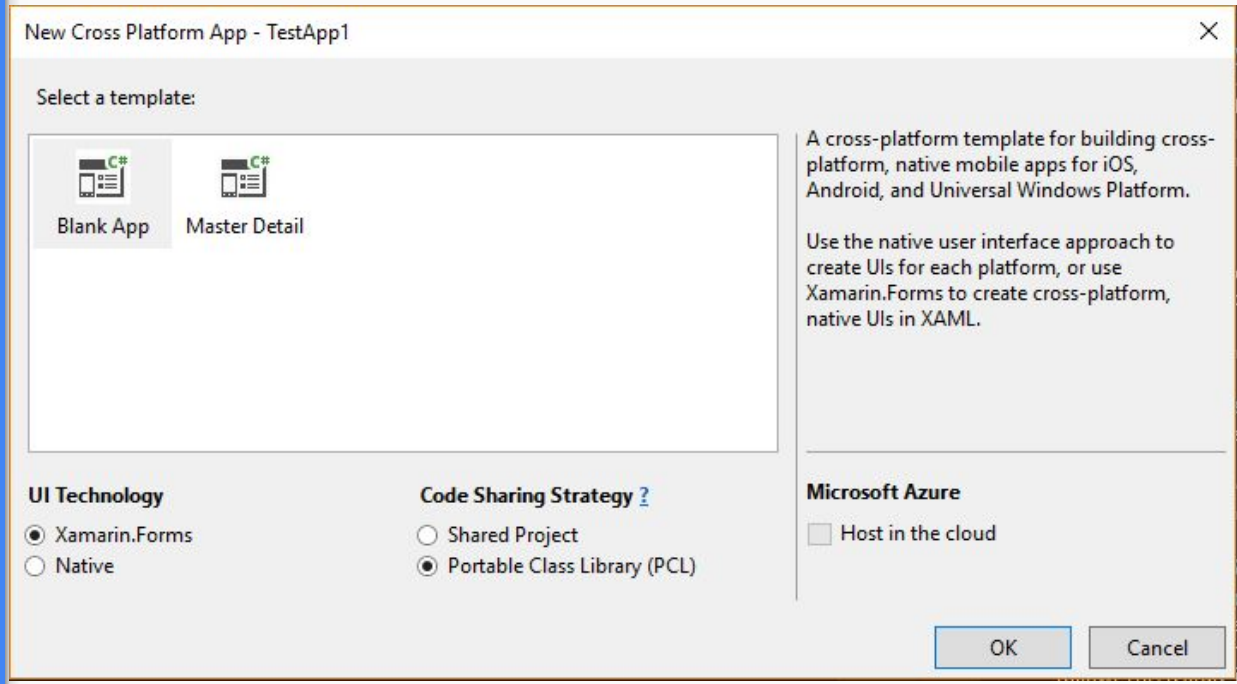
# Visual Studio 2017




Richard Taylor - @rightincode

# Xamarin Forms Project

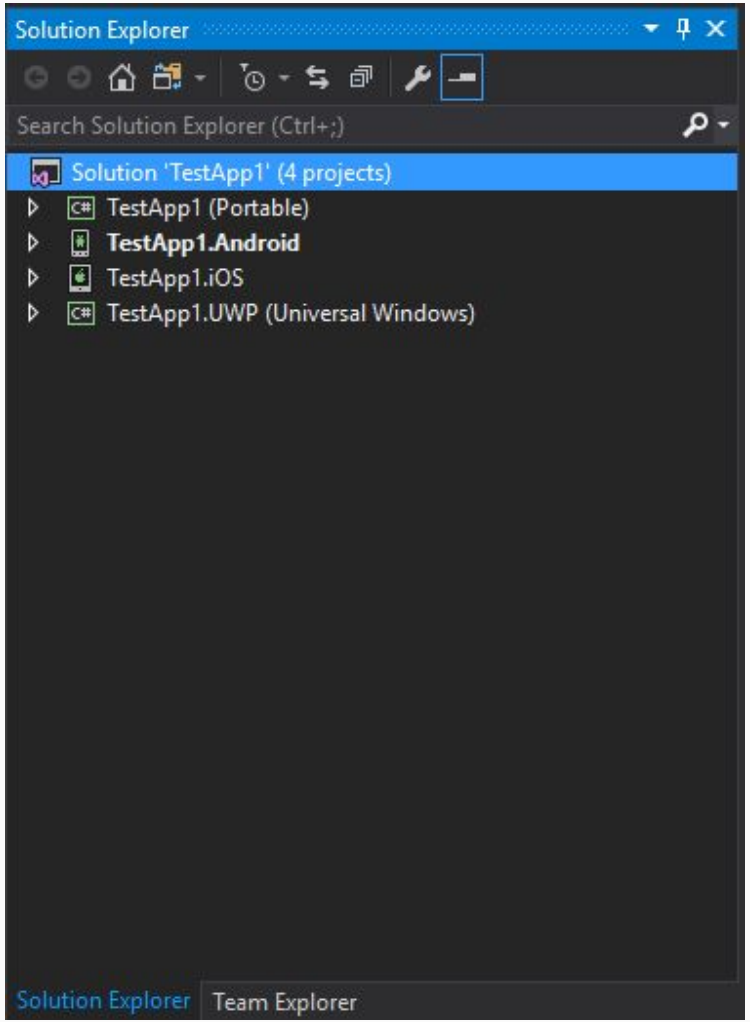
# Visual Studio 2017




Richard Taylor - @rightincode

# Xamarin Forms Project

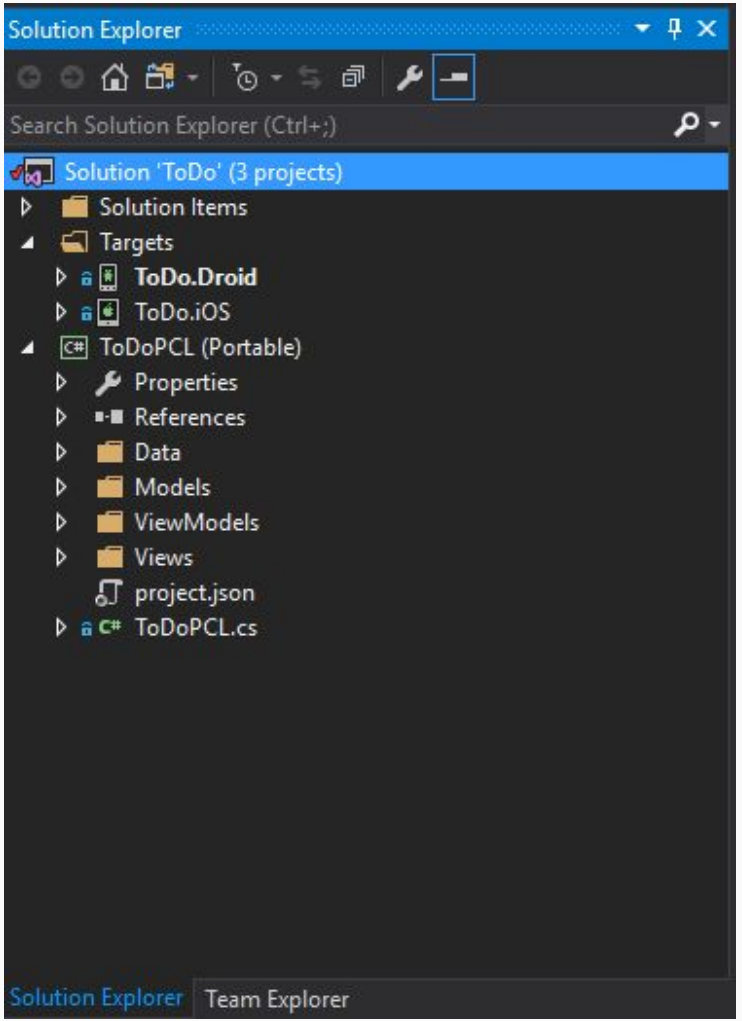
## Visual Studio 2017
(Default Project Structure)




Richard Taylor - @rightincode

# Xamarin Forms Project

## Visual Studio 2017
(Recommended Project Structure)



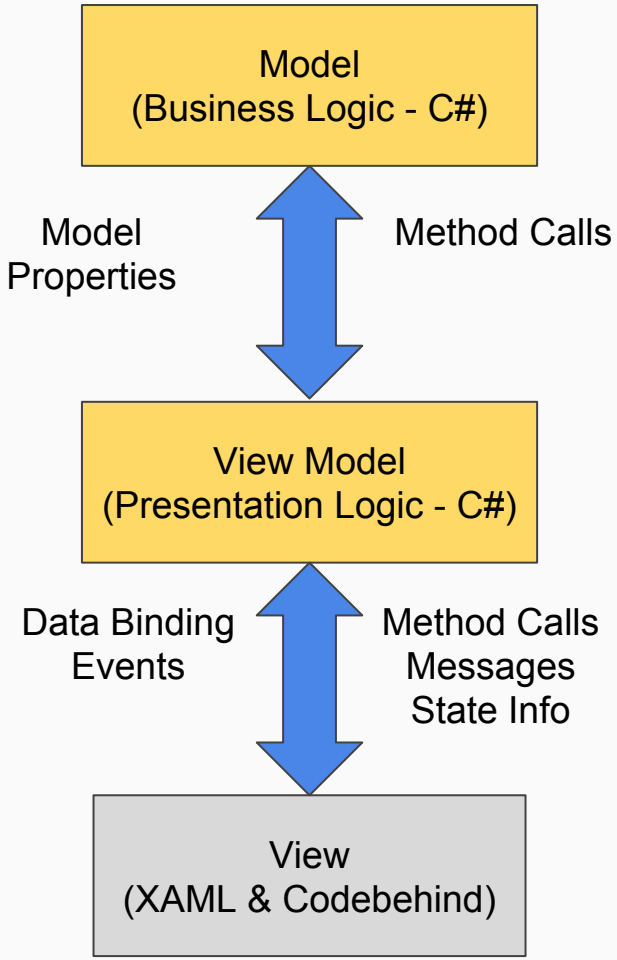

Richard Taylor - @rightincode

# Xamarin Forms Project

## Visual Studio 2017

### MVVM Design Pattern

- Model
- View Model
- View

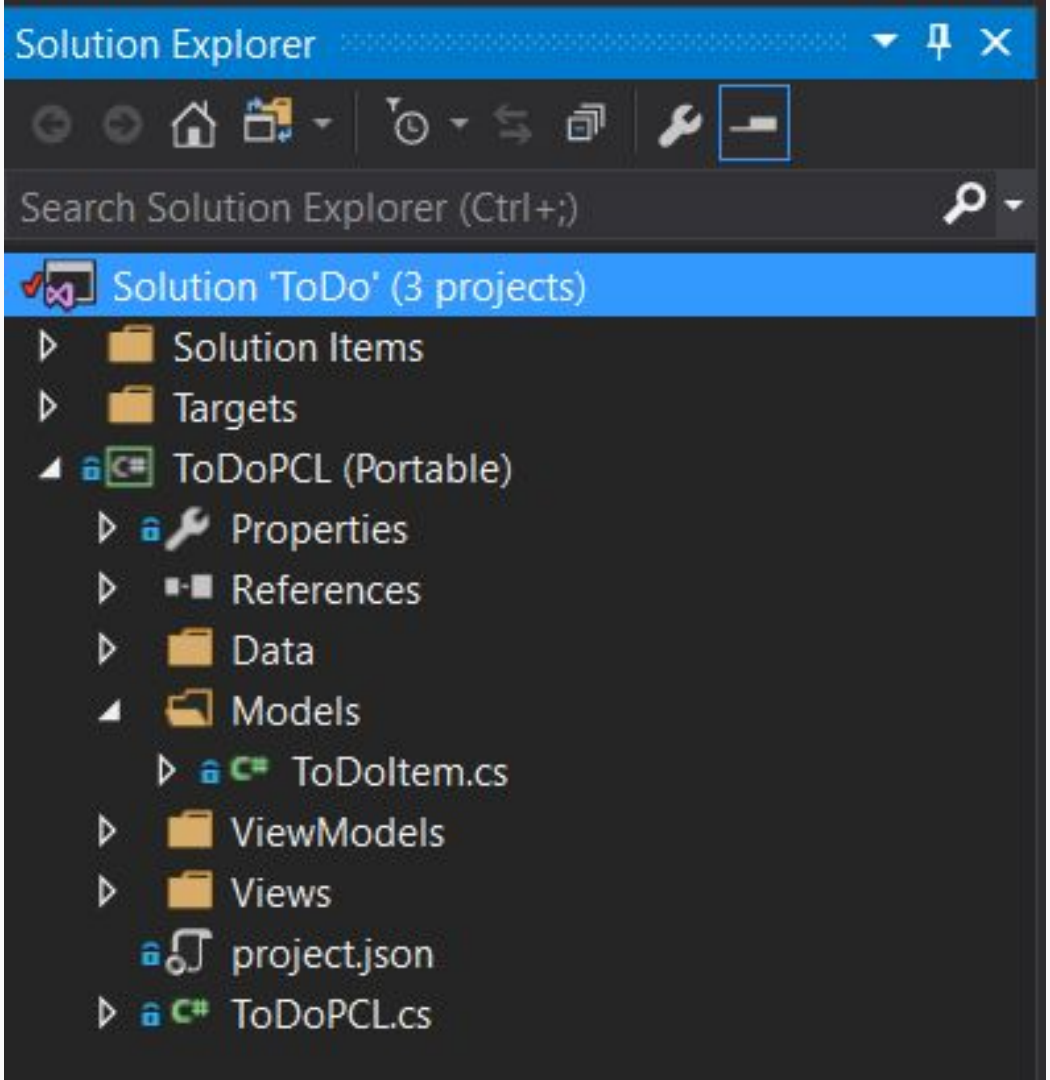



Richard Taylor - @rightincode

# Xamarin Forms Project

## Visual Studio 2017
### Portable Class Library Project

Models Folder

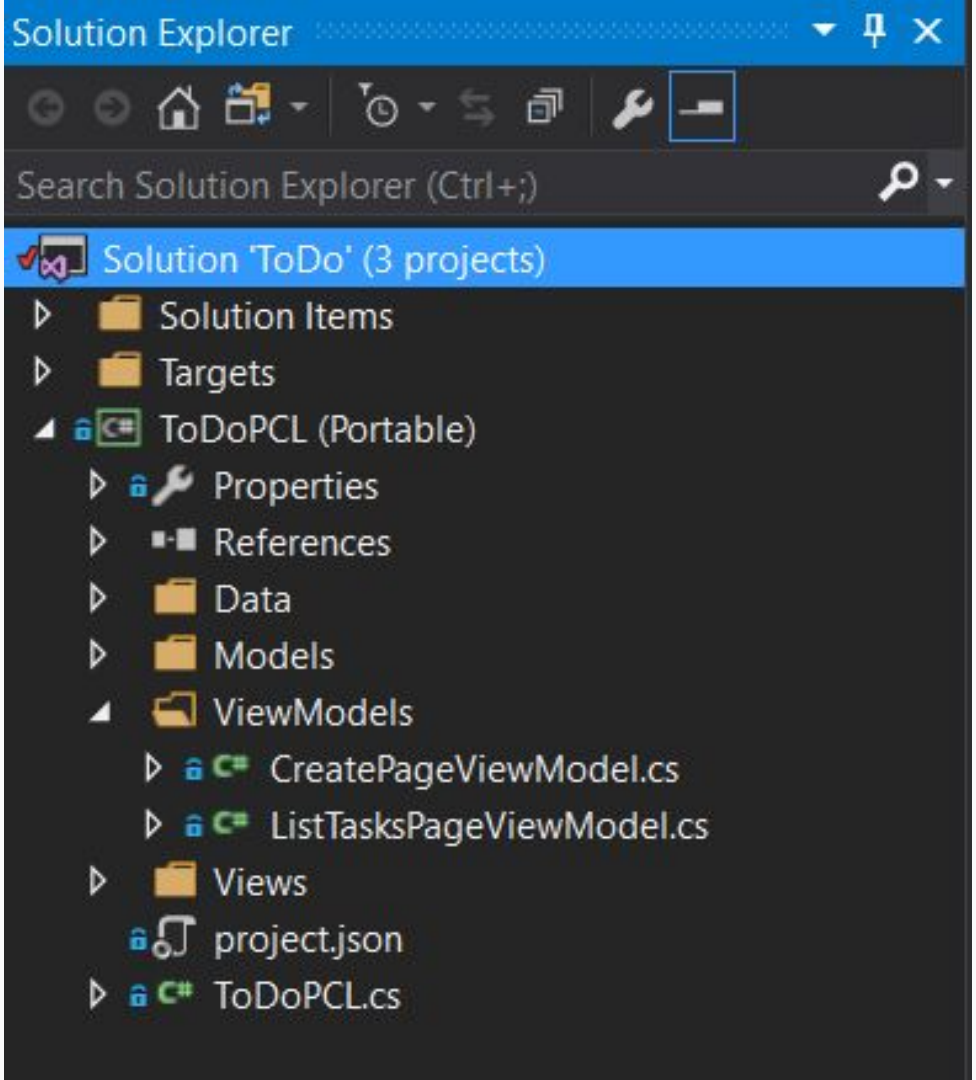



Richard Taylor - @rightincode

# Xamarin Forms Project

## Visual Studio 2017
### Portable Class Library Project

View Models Folder

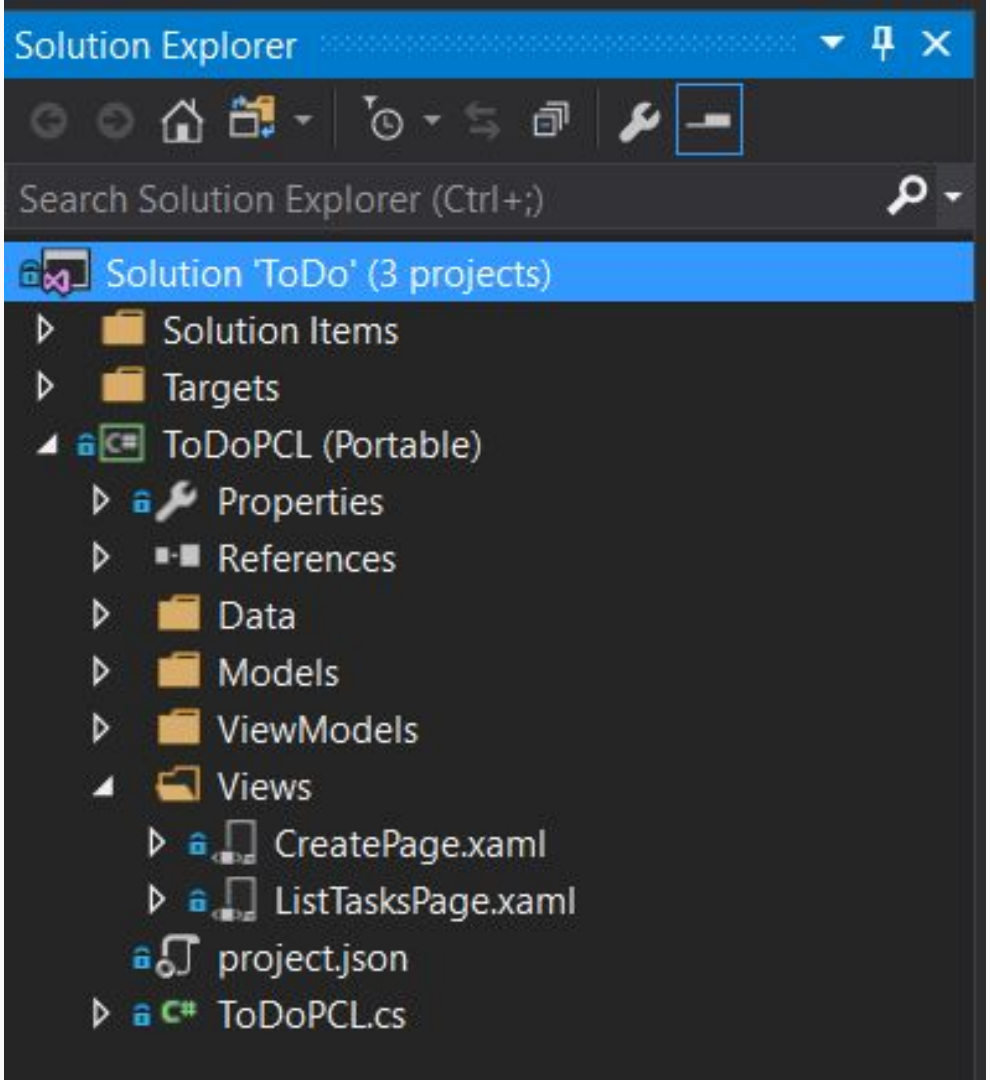



Richard Taylor - @rightincode

# Xamarin Forms Project

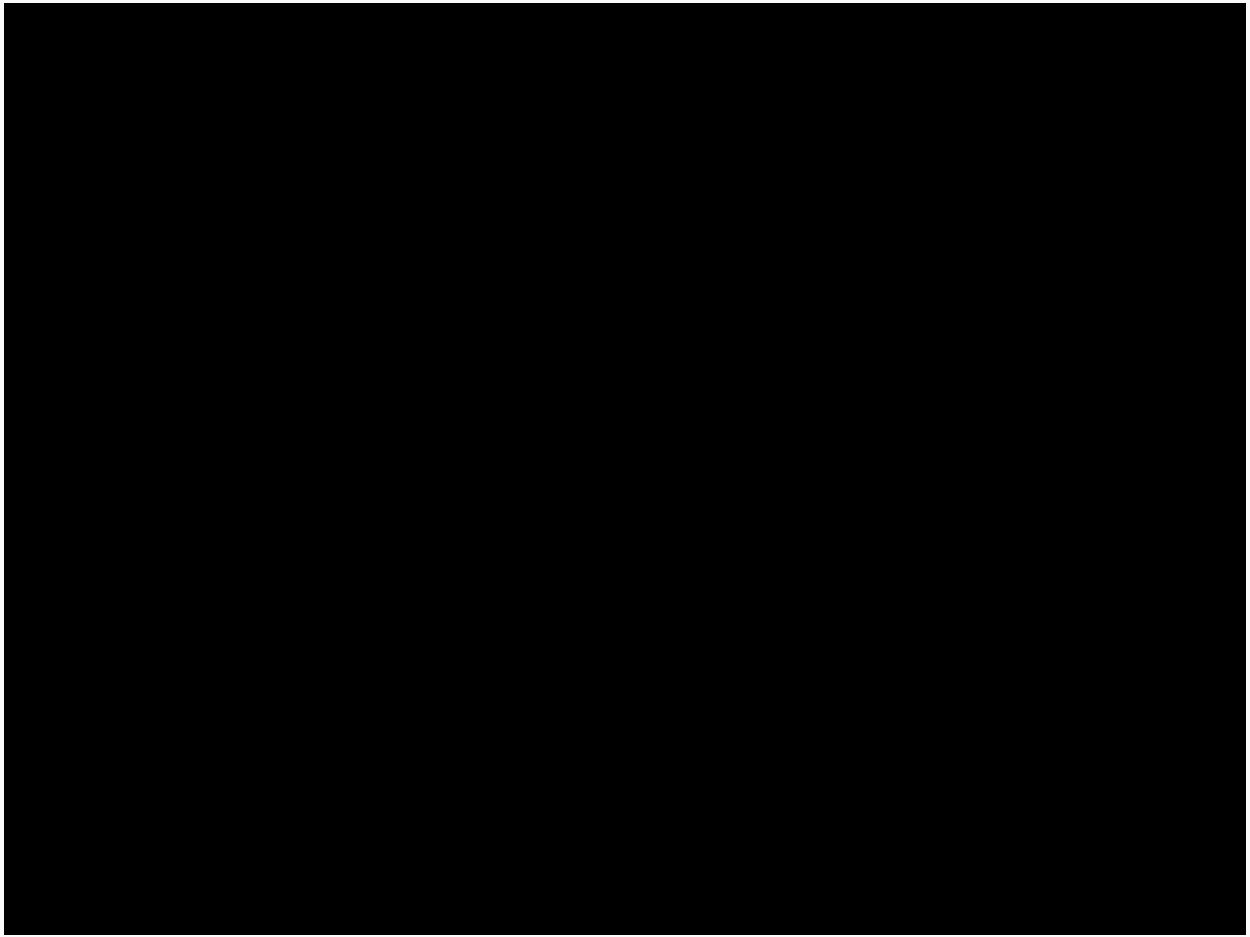
## Visual Studio 2017
### Portable Class Library Project

Views Folder




Richard Taylor - @rightincode

# Demo - Xamarin Forms Application



# Azure App Service

- Platform-as-a-service offering of Microsoft Azure
- It can be used to:
  - Create web and mobile apps for any platform or device
  - Integrate with SaaS solutions
  - Connect to on-premises applications
  - Automate business processes
- Apps created run on fully managed virtual machines in Azure

# Azure App Service - Why Use?

- Support multiple languages and frameworks (ASP.NET, Node, Java, PHP, and Python)
- Supports DevOps optimization (continuous integration and deployment)
- Scale globally with high availability
- Selection of connectors that can be used with popular SaaS platforms and on-premises data
- Security and compliance
- Selection of application templates in the Azure Marketplace
- Visual Studio integration

# Azure App Service - App Types

- **Web Apps** - Used to host websites and web applications
- **Mobile Apps** - Used to host mobile app back ends
- **Api Apps** - Used to host RESTful API’s
- **Logic Apps** - Used to automate business processes and integrating systems and data without writing code

# Azure App Service - Mobile Apps

# Demo - Azure App Service Mobile App

# Resources

- Repo: (code and slides - branch: cltazurebootcamp)
  - https://github.com/rightincode/Xamarin-Forms-ToDo
- Xamarin:
  - https://developer.xamarin.com/guides/
- Xamarin Forms:
  - https://developer.xamarin.com/guides/xamarin-forms/
- Azure App Service:
  - https://docs.microsoft.com/en-us/azure/app-service/
- Mobile Apps:
  - https://docs.microsoft.com/en-us/azure/app-service-mobile/

# Questions?

# Thanks!