

# Xamarin

Roman Jašek

*Software Architect, Riganti s.r.o.*

*Microsoft Most Valuable Professional (MVP)*

[roman.jasek@riganti.cz](mailto:roman.jasek@riganti.cz)

# Tradiční přístup k vývoji aplikací



iOS App

Objective-C  
Swift

Xcode



Android App

Java  
Kotlin

Android Studio

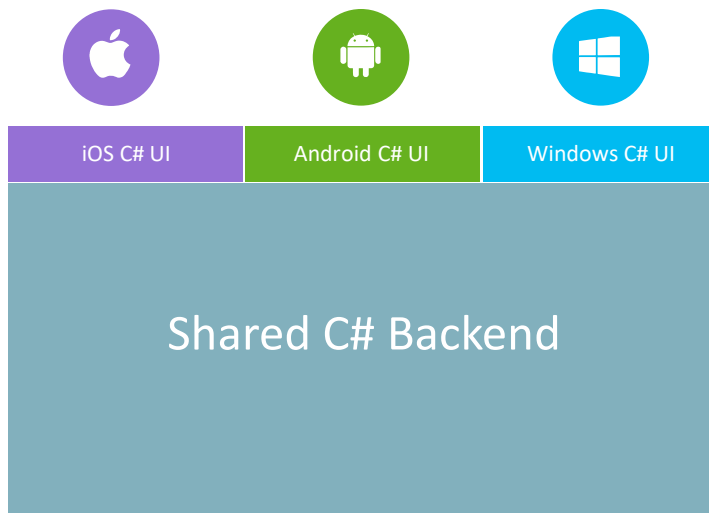


Windows App

C#

Visual Studio

# Xamarin



Xamarin

- Společná logika pro všechny platformy
- UI psáno v C#
- Možnost využít vývoj podobný vývoji pro specifické platformy

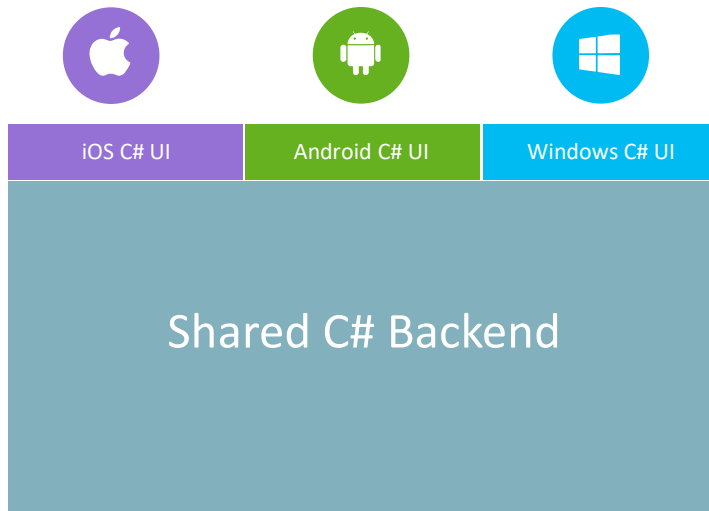
# Xamarin Android

- Celý vývoj je možný ve Visual Studiu
- Podpora pro .axml
- Integrovaný designer
- Kód psaný v C#
- Možnost debugovat na zařízeních

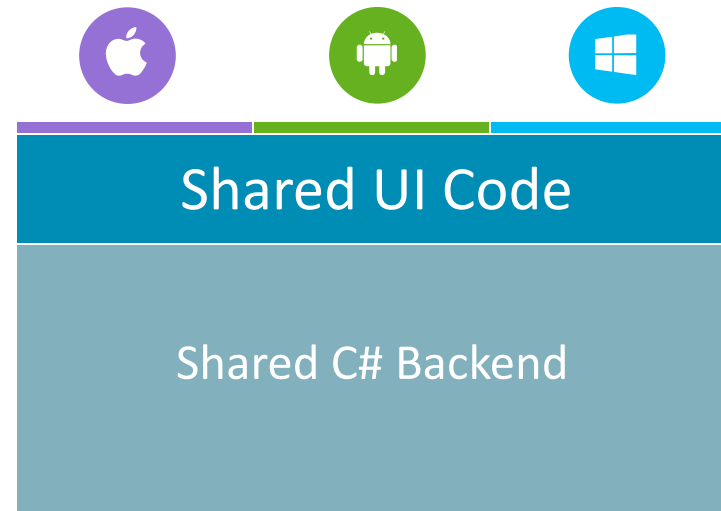
# Xamarin iOS

- Je nutno mít někde Mac
- Vývoj probíhá ve Visual Studiu
- Mac na síti zbuildí a spustí aplikaci
- iOS Simulátor běží na Macu, ale existuje Xamarin iOS Simulator, který ho „streamuje“ do Windows
- Možnost debugovat na zařízení

# Xamarin + Xamarin.Forms

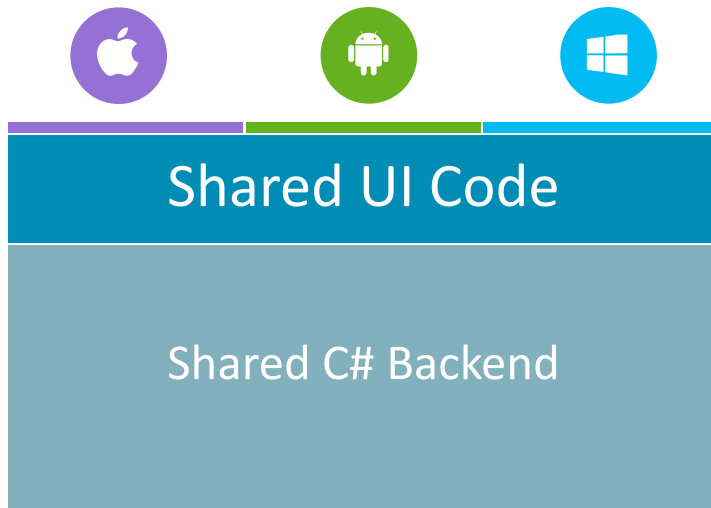


Xamarin



Xamarin.Forms

# Xamarin Forms

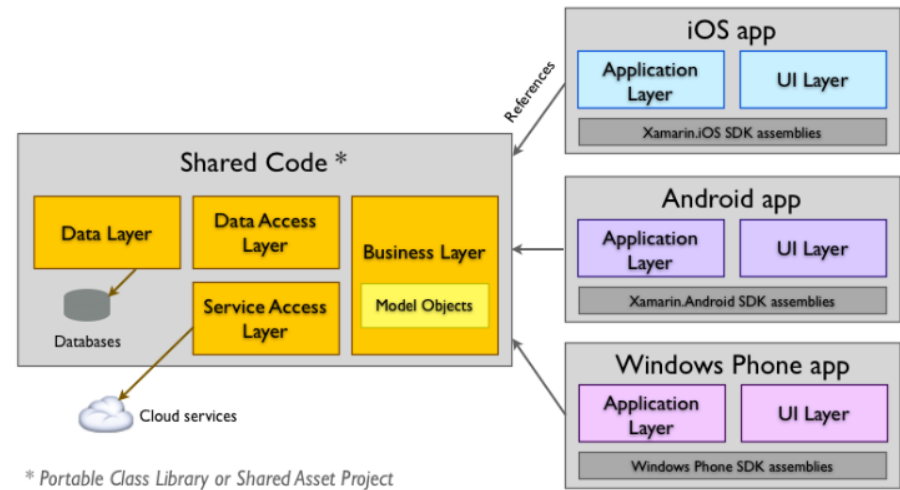


- Společná logika pro všechny platformy
- UI psáno v C#
- Možnost využít vývoj podobný vývoji pro specifické platformy
- **Společné uživatelské rozhraní pro všechny platformy**

# Shared Project vs .Net Standard

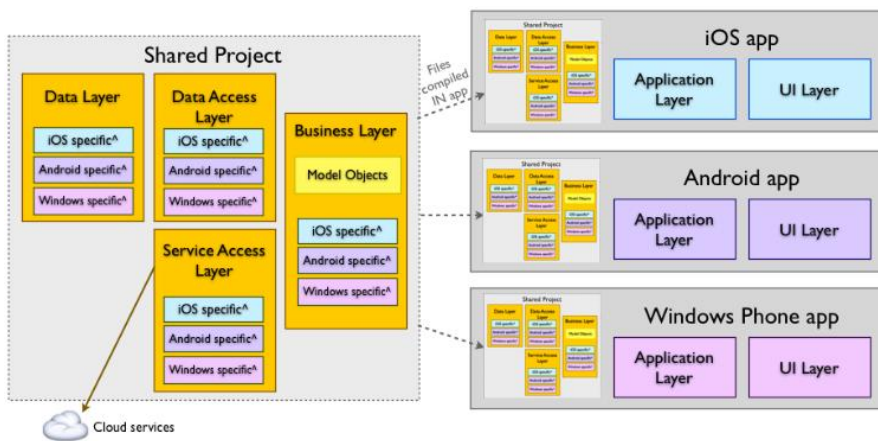
## Shared Project

- Jednodušší (zdánlivě)
- Špagety kód



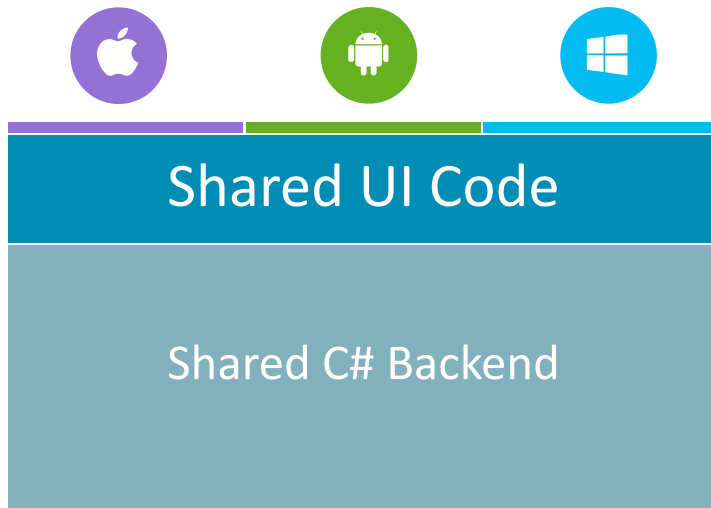
## .Net Standard Class Library

- Vhodné kombinovat s IoC
- Více kódu
- Spousta projektů
- Jednodušší dlouhodobě





# Aktuální stav



- ✓ 40+ stránek, layoutů, komponent (XAML)
- ✓ Data binding
- ✓ Navigace
- ✓ Animace
- ...

# XAML

- Formát pro serializaci hierarchie objektů
- Nejen pro popis uživatelského rozhraní
- Mapování .NET namespaces na XML namespaces
- Možné propojení s code-behind třídou

# Kde všude XAML potkáme?

- Windows Presentation Foundation (WPF)
  - Poprvé v .NET 3.0
- Silverlight
- Windows Phone
- Universal Windows Platform (UWP)
- Windows Workflow Foundation (WF)
  - Neslouží pro popis UI
- Xamarin.Forms

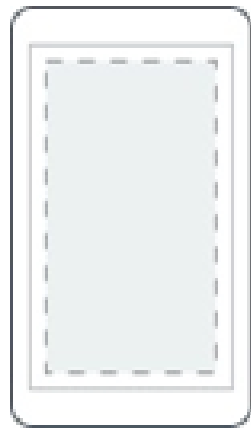
# XAML

- `x:Class` ... třída s code-behindem
- `UserControl` ... z čeho tato třída dědí
- `xmlns:x` ... speciální namespace pro účely XAMLu (povinný)
- `xmlns` ... namespace s vestavěnými komponentami

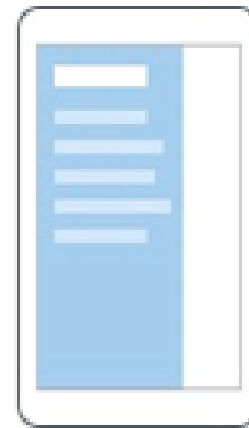
```
<?xml version="1.0" encoding="utf-8" ?>  
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"  
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"  
             x:Class="XamarinCvut.Forms.Views.Page1">  
</ContentPage>
```

# Stránky

- Content Page  
Jednoduchý obsah
- MasterDetail Page  
Seznam + detail položky



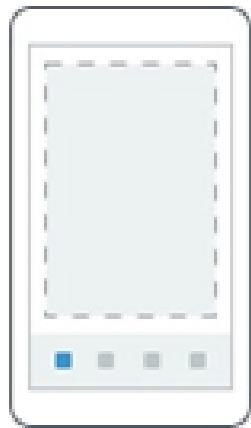
Content



MasterDetail

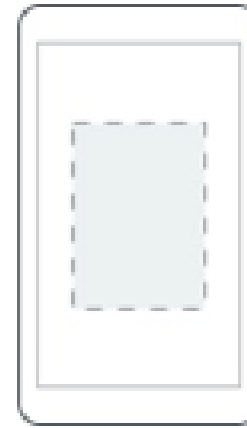
# Stránky

- TabbedPage  
Záložky



Tabbed

- CarouselPage  
Stránky vedle sebe

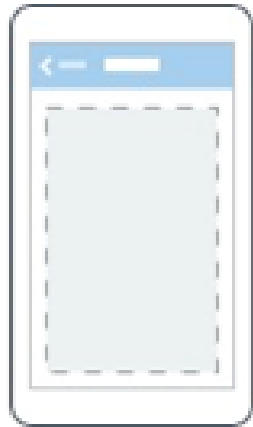


Carousel

# Stránky

- `NavigationPage`

Aby fungovala navigace



Navigation

- `TemplatedPage`

Prázdná stránka, bázová třída pro ostatní

# Stránky

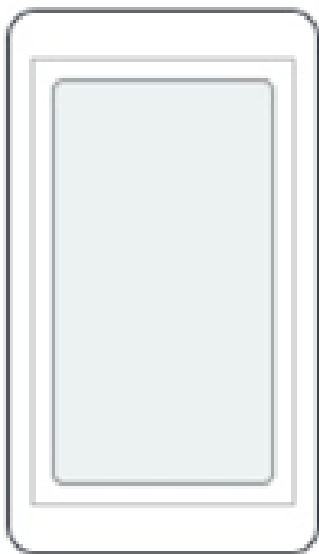
- První, co se zobrazí, je v souboru App.cs
  - Defaultně MainPage



# Layouts – jedna komponenta

- **ContentPresenter**

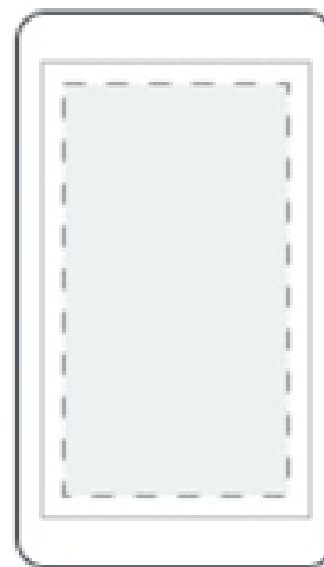
Místo pro obsah - šablony



Frame

- **ContentView**

Jeden obsah, bázová třída

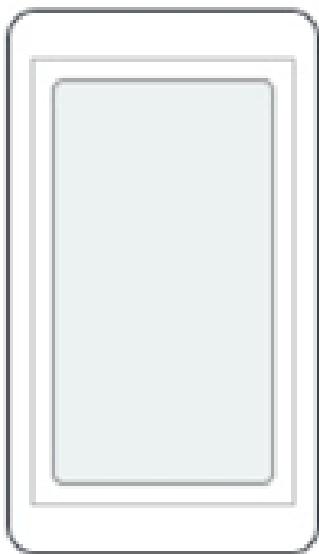


ContentView

# Layouts – jedna komponenta

- Frame

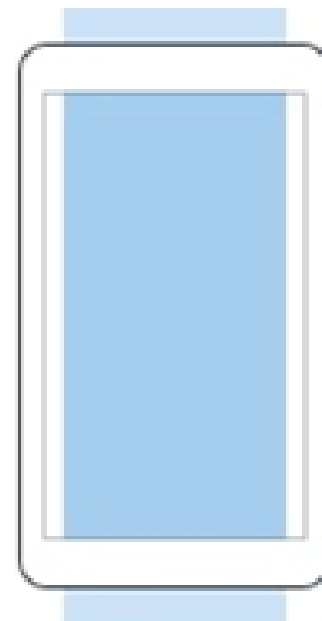
Možnost přidat rámeček



Frame

- ScrollView

Pokud se nevejde, scrolluje se

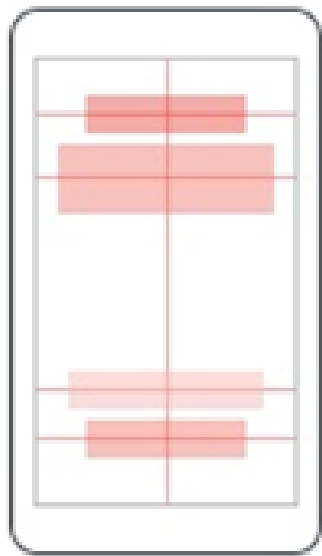


ScrollView

# Layouts – více komponent

- `AbsoluteLayout`

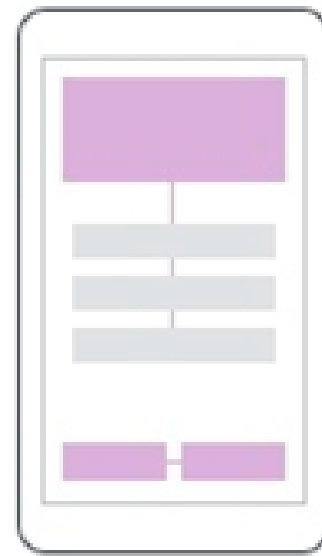
Absolutní pozicování komponent



Absolute

- `RelativeLayout`

Rozmístění pomocí constraintů



Relative

# Layouts – více komponent

- Grid

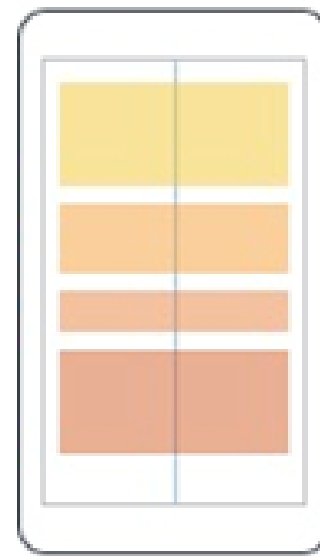
Tabulkový layout



Grid

- StackLayout

Pod sebe nebo vedle sebe



Stack

# Layouts – více komponent

- FlexLayout

Obdoba webového Flexboxu



FlexLayout

# Grid

- Skoro stejné jako ve WPF
- RowDefinitions, ColumnDefinitions
  - Width / Height = 150 | 3\* | Auto
- Grid.Row, Grid.Column, Grid.RowSpan...

# StackLayout

- Obdoba StackPanelu ve WPF
- Řadí komponenty pod sebe nebo vedle sebe
- Orientation = Horizontal | Vertical
- Spacing – rozestup jednotlivých komponent
- HorizontalOptions a VerticalOptions
  - Start, Center, End a Fill
  - ...AndExpand – komponenta může dostat víc místa, než nutně potřebuje, pokud je to místo k dispozici

# Ovládací prvky

ActivityIndicator

BoxView

Button

DatePicker

Editor

Entry

Image

Label

ListView

Map

OpenGLView

Picker

ProgressBar

SearchBar

Slider

Stepper

Switch

TableView

TimePicker

WebView

EntryCell

SwitchCell

TextCell

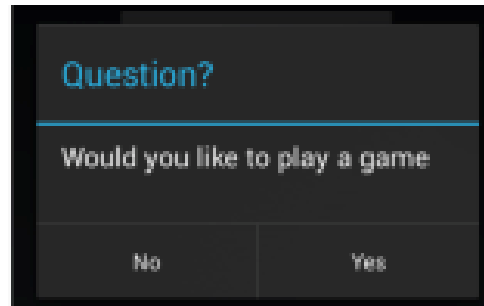
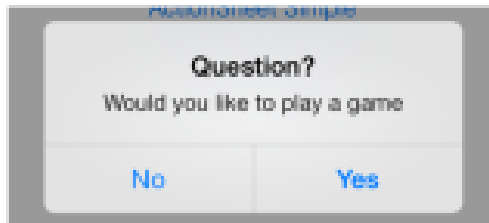
ImageCell

ViewCell



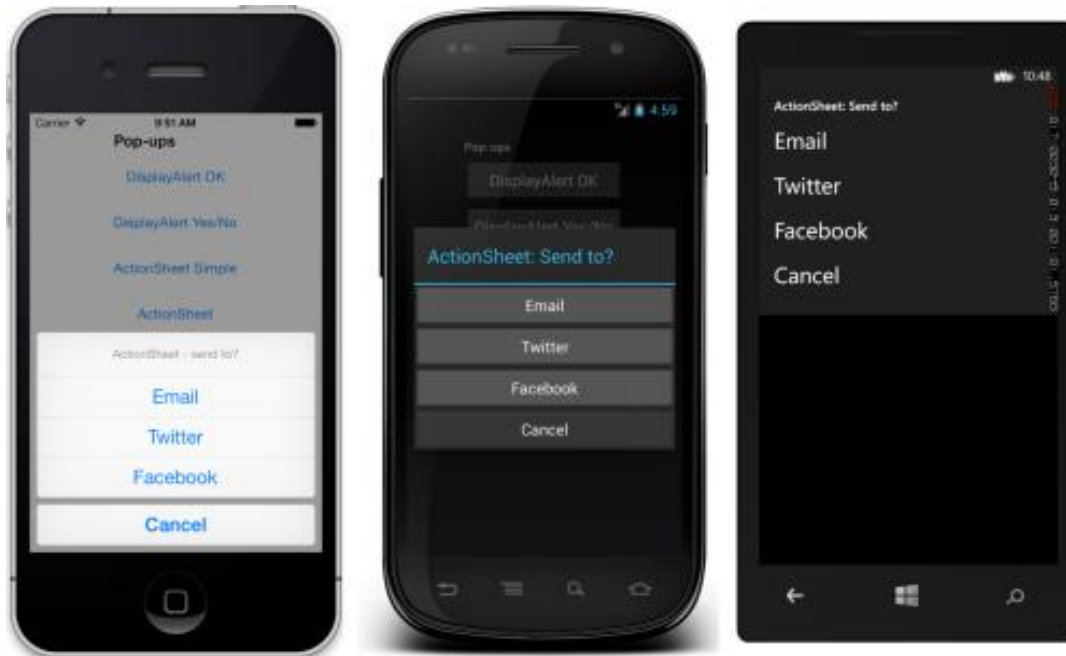
# Popups - alert

```
DisplayAlert("Question?", "Would you like to play a game",  
"Yes", "No")
```



# Popus – action sheet

- `DisplayActionSheet("ActionSheet: Send To?", "Cancel", null, "Email", "Twitter", "Facebook")`



# Komponenty třetích stran



# Rozdíly – názvy

WPF	Xamarin Forms
StackPanel	StackLayout
TextBox	Entry
ListBox	ListView
CheckBox	Switch
ProgressBar	ActivityIndicator
Grid	Grid
Label	Label
Button	Button
Image	Image
Date/TimePicker	Date/TimePicker

# Rozdíly – Data Binding

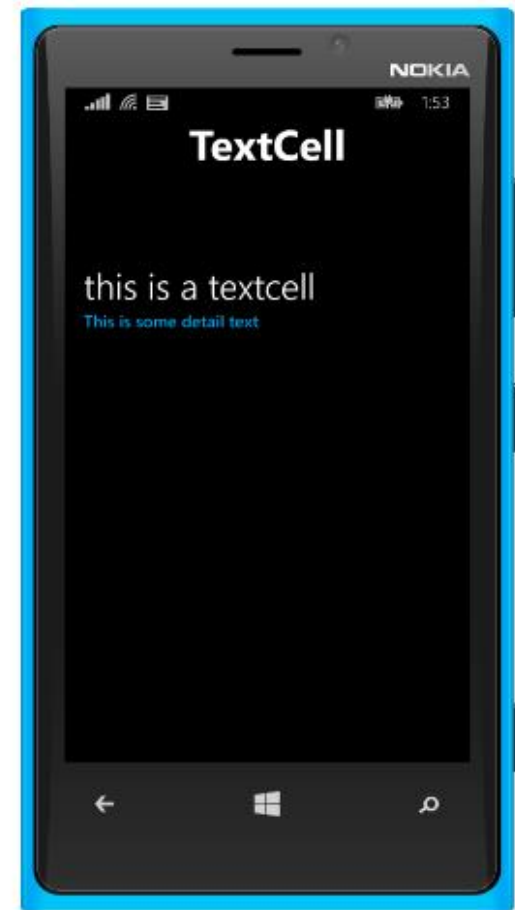
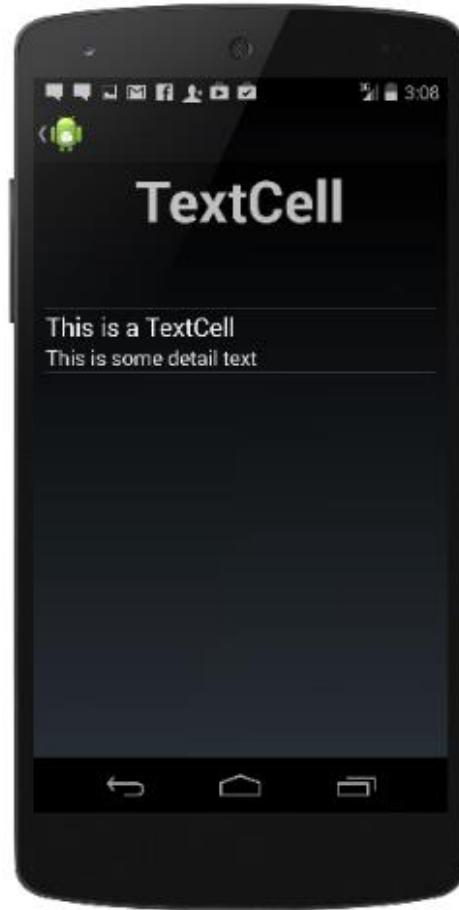
Windows	Xamarin.Forms
DataContext	BindingContext
{Binding Property}	{Binding Property}
ItemsSource	ItemsSource
ItemTemplate	ItemTemplate
DataTemplate	DataTemplate

```
<Label Text="{Binding Color.R,  
        Converter={StaticResource intConverter},  
        ConverterParameter=255,  
        StringFormat='R={0:X2}'}" />
```

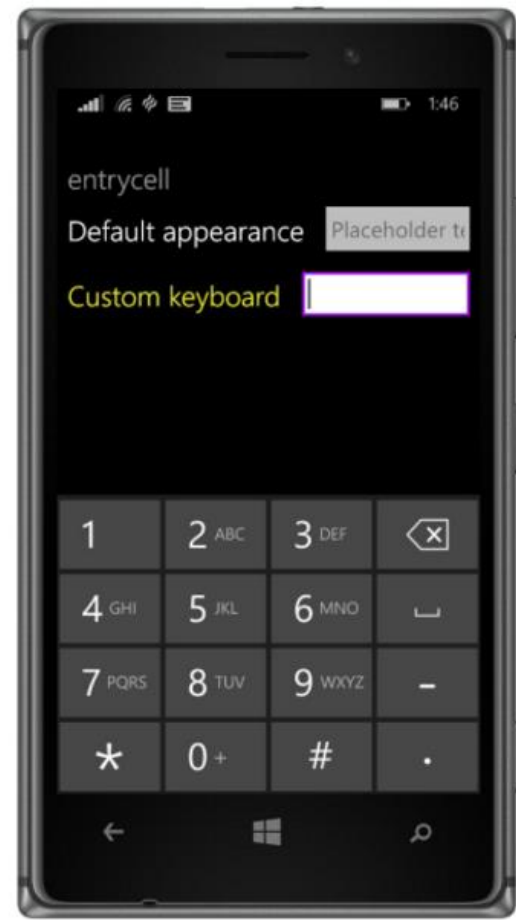
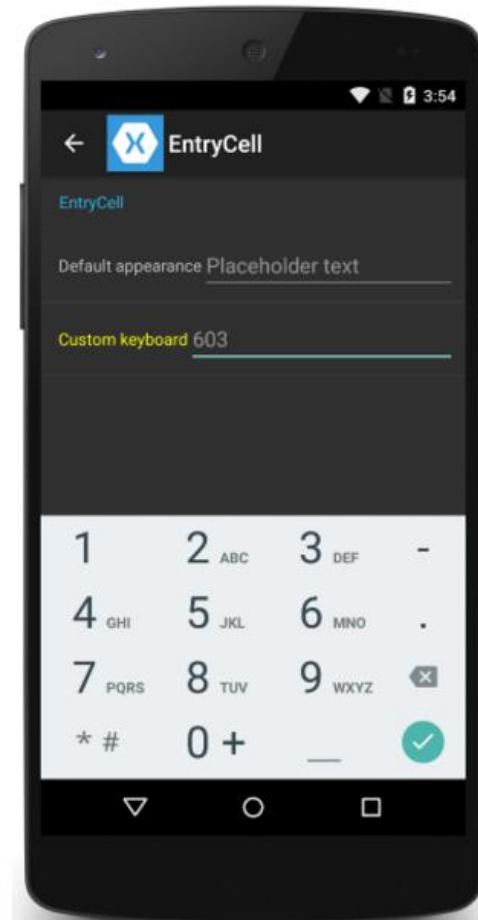
# Grid, StackLayout

- Prakticky stejné použití jako ve WPF
- Můžeme používat Grid v Android/iOS?  
Awesome!

# TextCell



# EntryCell

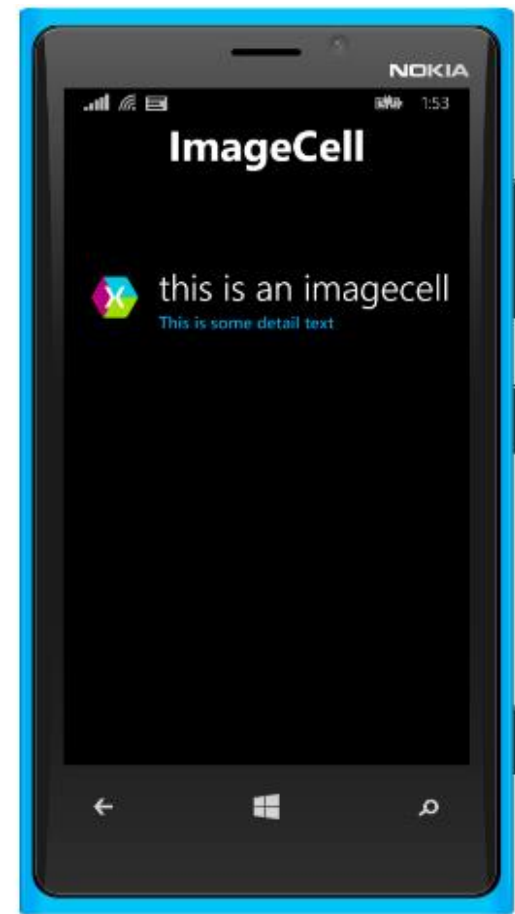
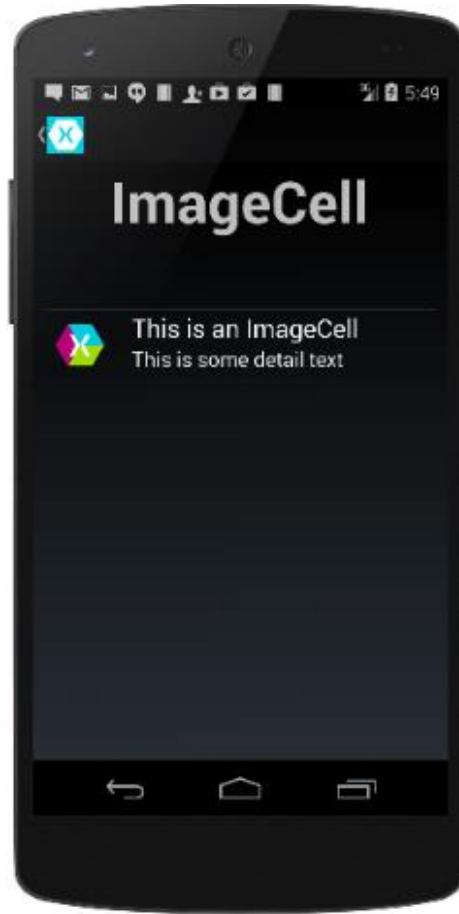




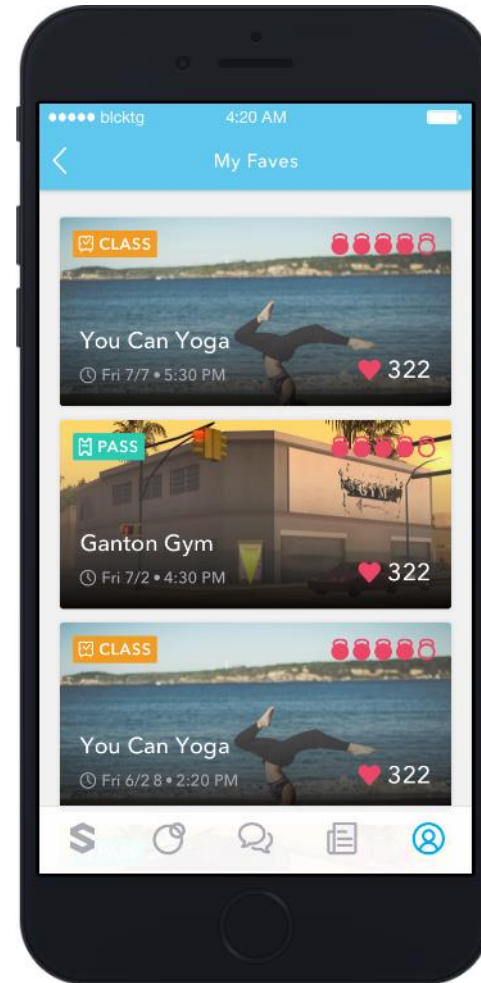
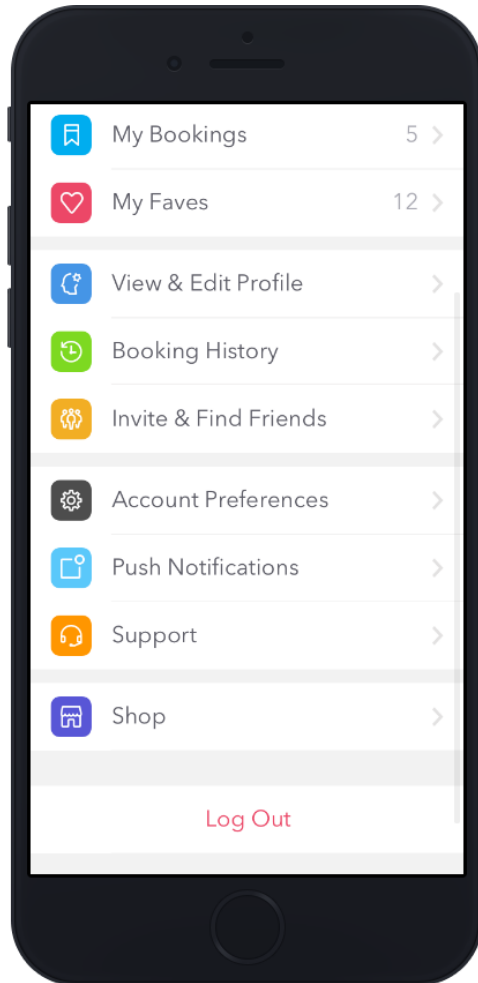
# SwitchCell



# ImageCell



# ViewCell



# Cells & Lists

- TextCell, ImageCell, SwitchCell, EntryCell...
- ViewCell (dovnitř lze dát cokoliv)
  - ViewCell.ContextActions ... kontextové menu
- ListView
  - ItemsSource="{Binding: ...}"
  - IsPullToRefreshEnabled = true | false
    - zavolá RefreshCommand
  - HasUnevenRows = true | false
    - různé výšky řádků

# Platformně specifické

- Starý zápis je již deprecated
- Nový zápis povoluje jednotnou definici pro víc platforem

```
<Button.BorderWidth>  
  <OnPlatform x:TypeArguments="system:Double">  
    <On Platform="Android" Value="5"/>  
    <On Platform="iOS, UWP" Value="10"/>  
  </OnPlatform>  
</Button.BorderWidth>
```

# Platformně specifické

- Custom Renderer

- Když něco nejde v XAML

- Ve Forms projektu vyrobit třídu

```
public class BoxViewWithRadius : BoxView
```

- V Android/iOS projektu vyrobit Renderer

```
[assembly: ExportRenderer(typeof(BoxViewWithRadius),  
typeof(BoxViewWithRadiusRenderer))]
```

```
public class BoxViewWithRadiusRenderer : BoxRenderer
```

- V XAMLu použití třídy z Forms projektu

- Implementace pro každou platformu zvlášť

# Platformně specifické

- Přístup k souborům
- Práce s push notifikacemi
- Zjišťování stavu baterie
- ...
- Řešení pomocí interface ve sdíleném projektu a implementace zvlášť pro jednotlivé platformy
- Vestavěný mechanismus – DependencyService
- Možnost použití vlastního IoC kontejneru

# Platformně specifické

- Podpora pro iOS a Android
  - 100% podpora pro API při releasu
- Zápis velmi podobný Java/Swift, stejné názvy tříd atd.
- Z existujícího kódu na jinou platformu je možné jednoduše přepsat do Xamarinu
- **Xamarin.Essentials** – multiplatformní přístup k těmto API



# Data binding

- Mezi code-behind a XAML
- Mezi ViewModelem a View
- `Text="{Binding Path=Operand1}"`
- Mezi komponentami ve View
- `Text="{Binding Source={x:Reference Name=DisplayAlertButton}, Path=Text}"`

# Směr bindingu

- Vlastnost **Mode**
  - **OneTime**
    - jen jednou na začátku
  - **OneWay**
    - jedním směrem – ze zdroje do cíle
  - **TwoWay**
    - obousměrně – změna v cíli změní i zdroj
  - **OneWayToSource**
    - jedním směrem – od cíle k zdroji
- Zdroj – vlastnost, na kterou bindujeme
- Cíl – komponenta, která má {Binding}

# Binding - Notifikace o změnách

- OneWay a TwoWay
  - reaguje na změny zdroje
  - zdroj o nich musí dát nějak vědět
  - objekt implementuje INotifyPropertyChanged
    - `public event PropertyChangedEventHandler PropertyChanged`
- Je vhodné přidat metodu `OnPropertyChanged` s parametrem `s` a atributem `[CallerMemberName]`
- Knihovna Fody:
  - Nuget balíček Fody
  - Nuget balíček `PropertyChanged.Fody`

# INotifyCollectionChanged

- `ObservableCollection<T>`
  - implementuje toto rozhraní
- Vlastní kolekce
  - podporu přidáme
- Existující kolekce
  - např. napsat kolem ní "wrapper"

# Data Binding Intellisense

`xmlns:d=http://schemas.microsoft.com/expression/blend/2008`

`xmlns:mc=http://schemas.openxmlformats.org/markup-compatibility/2006`

`mc:Ignorable="d"`

`d:DataContext="{d:DesignInstance test:MainPageViewModel}"`

# Intellisense

- Extensions:
  - Resharper

# MVVM Frameworky

- MVVM Cross
- Simple MVVM
- MVVM Light
- Catel
- ReactiveUI

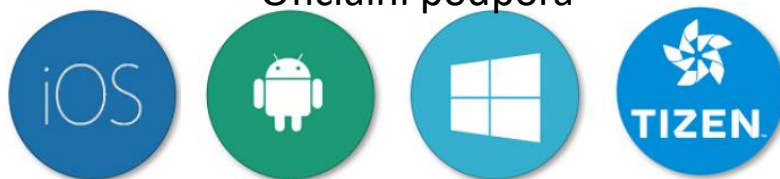
# Tip - načtení dat z webové služby

- Využití frameworku Swagger
- Vytvoření Web API v .Net
- Vygenerování klientského kódu
- Přiložení kódu do projektu jako dll



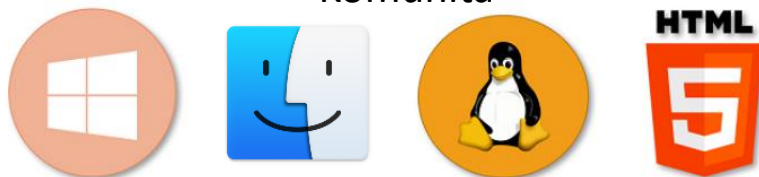
# Platformy - updated

Oficiální podpora\*



UWP

Komunita



WPF

ASP.NET Core

\*Tizen je podporován Samsungem

# Výhody Xamarin Forms

- XAML
- .Net
- Dobrá dokumentace
- Sdílení kódu – 85%+