

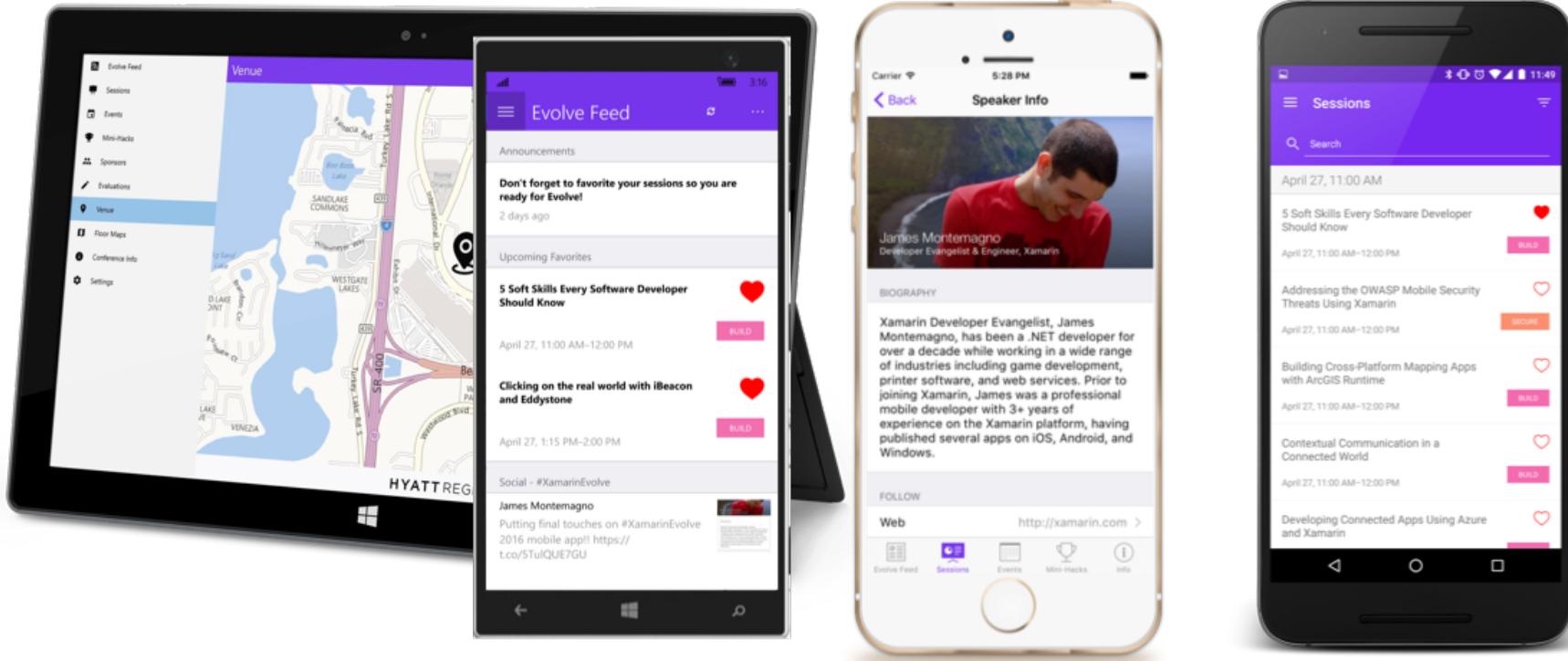


by



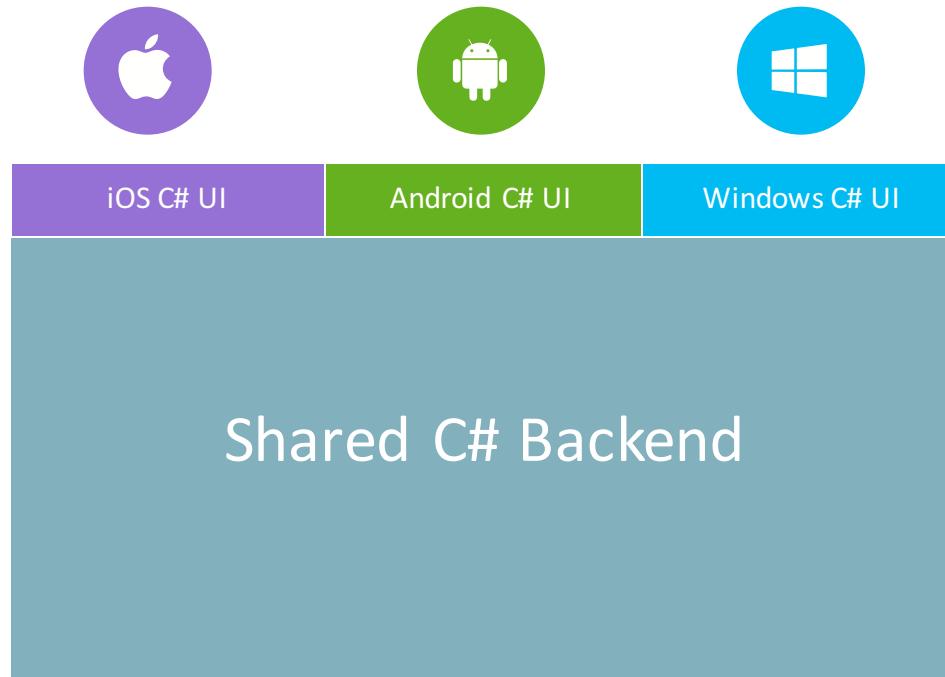
Diego Bonilla
@dbonillanareia

Meet Xamarin.Forms

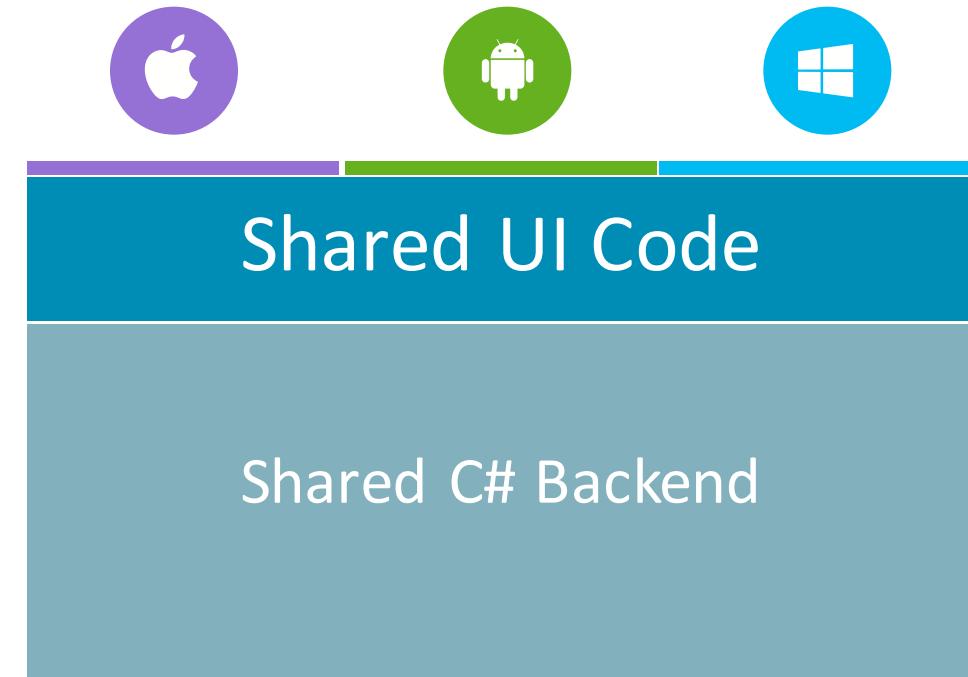


Build native UIs for iOS, Android, and Windows Phone from a single, shared C# codebase.

Xamarin + Xamarin.Forms

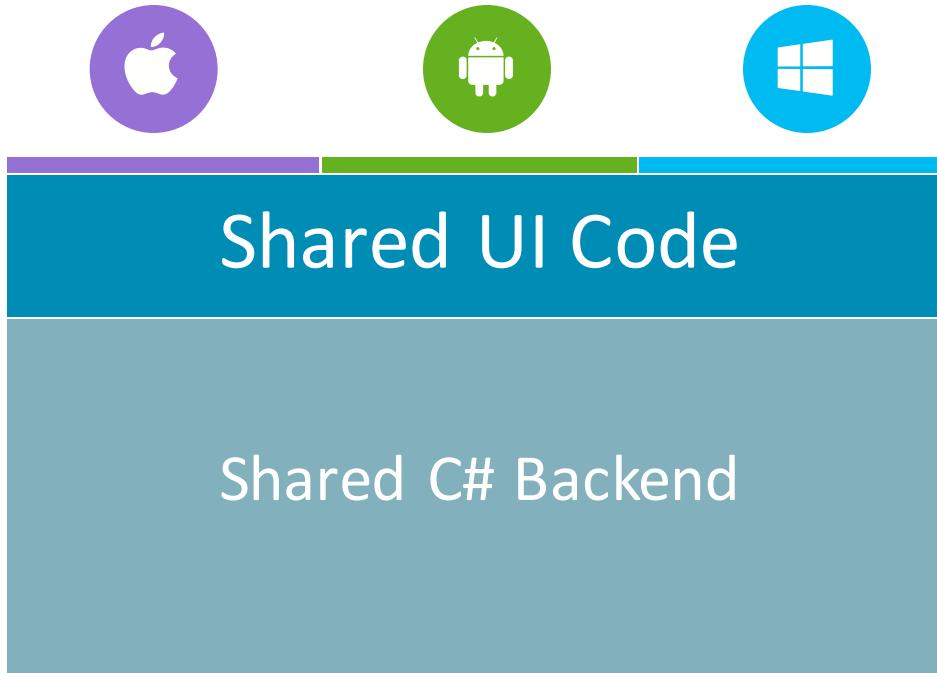


Traditional Xamarin Approach



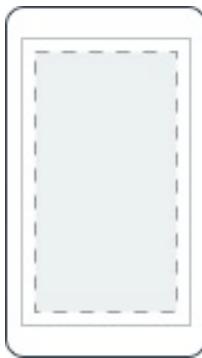
With Xamarin.Forms:
More code-sharing, all native

What's included



- ✓ 40+ Pages, layouts, and controls
(Build from code behind or XAML)
- ✓ Two-way data binding
- ✓ Navigation
- ✓ Animation API
- ✓ Dependency Service
- ✓ Messaging Center

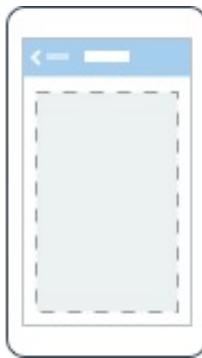
Pages



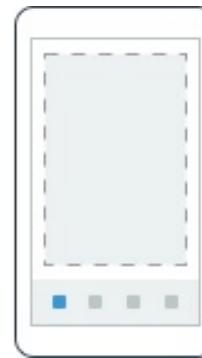
Content



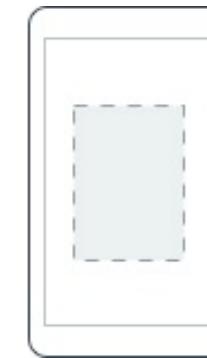
MasterDetail



Navigation

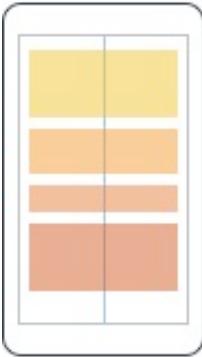


Tabbed

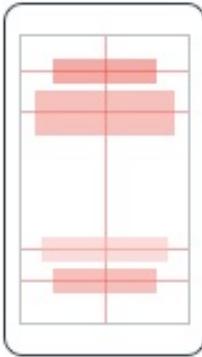


Carousel

Layouts



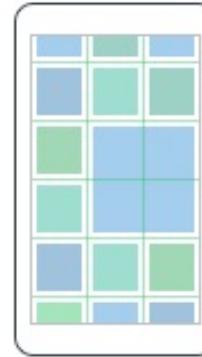
Stack



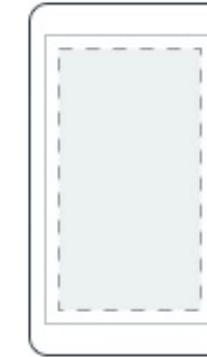
Absolute



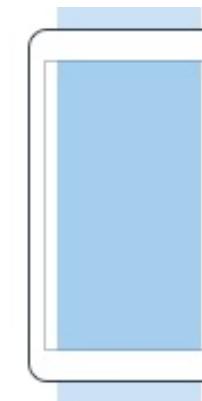
Relative



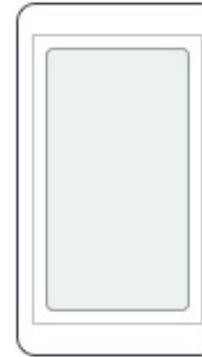
Grid



ContentView



ScrollView



Frame

Controls

ActivityIndicator

BoxView

Button

DatePicker

Editor

Entry

Image

Label

ListView

Map

OpenGLView

Picker

ProgressBar

SearchBar

Slider

Stepper

TableView

TimePicker

WebView

EntryCell

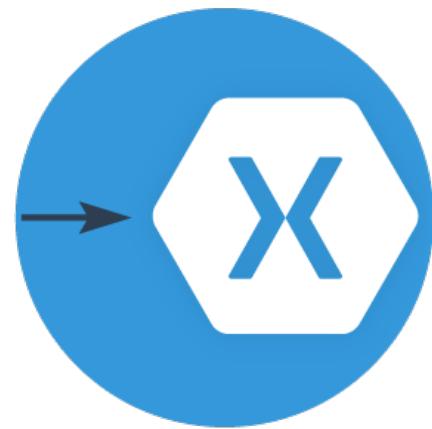
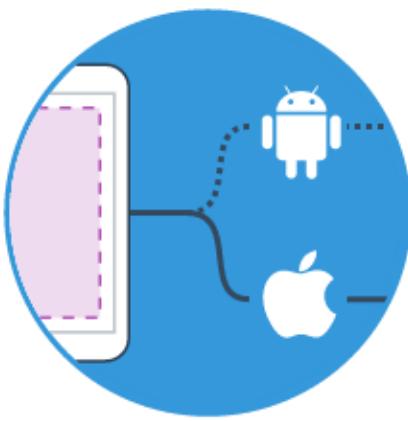
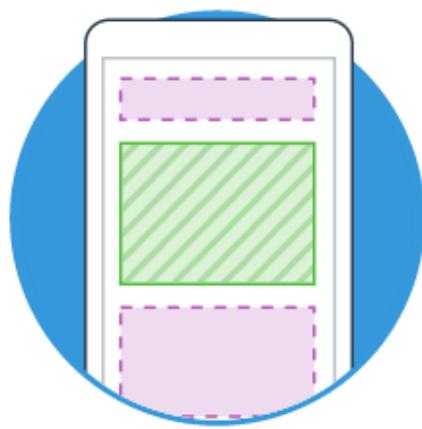
ImageCell

SwitchCell

TextCell

ViewCell

Extensibility



Embed Custom Views Anywhere

Not only are `Xamarin.Forms` pages mixable with custom screens, you can embed custom views built directly against `Xamarin.iOS` and `Xamarin.Android` into `Xamarin.Forms` pages.

Call Platform APIs via Shared Services

Need platform-specific functionality within a `Xamarin.Forms` page, such as querying the accelerometer? We've included services that quickly and easily abstract platform-specific API calls to keep you writing as much shared code as possible.

Easy migration to the Xamarin Platform

When your native platform integration requirements outgrow `Xamarin.Forms`, your shared code is easily migrated to `Xamarin.iOS` and `Xamarin.Android` using the full `Xamarin` platform.

Xamarin.Forms Ecosystem



Control Comparison

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

Binding Comparison

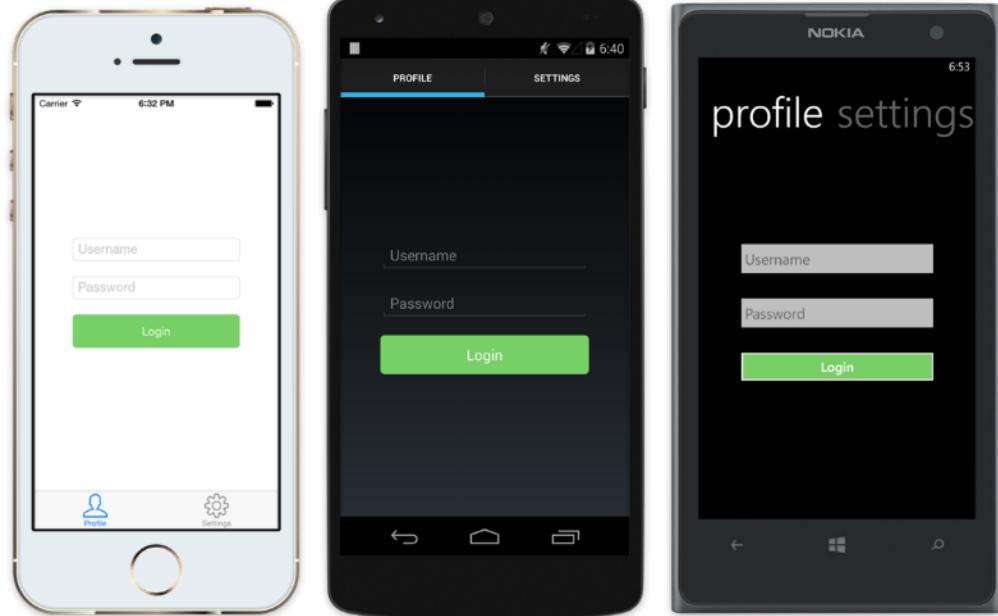
Windows	Xamarin.Forms
DataContext	BindingContext
{Binding Property}	{Binding Property}
ItemsSource	ItemsSource
ItemTemplate	ItemTemplate
DataTemplate	DataTemplate

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

Platform Customization

```
<ContentPage.Padding>
    <OnPlatform x:TypeArguments="Thickness">
        <OnPlatform.iOS>
            0, 20, 0, 0
        </OnPlatform.iOS>
        <OnPlatform.Android>
            0, 0, 0, 0
        </OnPlatform.Android>
        <OnPlatform.WinPhone>
            0, 0, 0, 0
        </OnPlatform.WinPhone>
    </OnPlatform>
</ContentPage.Padding>
```

Native UI from shared code



```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="MyApp.MainPage">
<TabbedPage.Children>
<ContentPage Title="Profile" Icon="Profile.png">
    <StackLayout Spacing="20" Padding="20"
                 VerticalOptions="Center">
        <Entry Placeholder="Username"
              Text="{Binding Username}"/>
        <Entry Placeholder="Password"
              Text="{Binding Password}"
              IsPassword="true"/>
        <Button Text="Login" TextColor="White"
               BackgroundColor="#77D065"
               Command="{Binding LoginCommand}"/>
    </StackLayout>
</ContentPage>
<ContentPage Title="Settings" Icon="Settings.png">
    <!-- Settings -->
</ContentPage>
</TabbedPage.Children>
```

Let's Take a Look

What's new?

XAML Previewer

Debug | iPhone Simulator > iPhone 6s iOS 9.3 Xamarin Studio Business 7 ⓘ Press '⌘.' to search

App.cs ImageCircleRenderer.cs ImageCircleRenderer.cs CoffeesPage.xaml ⚙

Hide Preview

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ContentPage
3   xmlns="http://xamarin.com/schemas/2014/forms"
4   xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
5   xmlns:local="clr-namespace:CoffeeCups;assembly=CoffeeCups"
6   xmlns:controls="clr-namespace:ImageCircle.Forms.Plugin.Abstractions;assembly=ImageCircle.Forms.Plugin.Abstractions"
7   x:Class="CoffeeCups.CoffeesPage"
8   Title="Cups Of Coffee">
9     <AbsoluteLayout HorizontalOptions="FillAndExpand" VerticalOptions="FillAndExpand">
10       <Grid AbsoluteLayout.LayoutFlags="All"
11         AbsoluteLayout.LayoutBounds="0,0,1,1"
12         RowSpacing="0">
13           <Grid.RowDefinitions>
14             <RowDefinition Height="Auto"/>
15             <RowDefinition Height="*"/>
16             <RowDefinition Height="Auto"/>
17           </Grid.RowDefinitions>
18           <StackLayout Orientation="Horizontal" Padding="16" Spacing="16" BackgroundColor="White">
19             <Label Text="Made at Home?">
20               TextColor="White"
21               VerticalOptions="Center"
22               HorizontalOptions="Start"
23               FontSize="16">
24                 <Label.FontColor>
```

Device: Phone Tablet Platform: Android iOS

Made at Home? ADD COFFEE

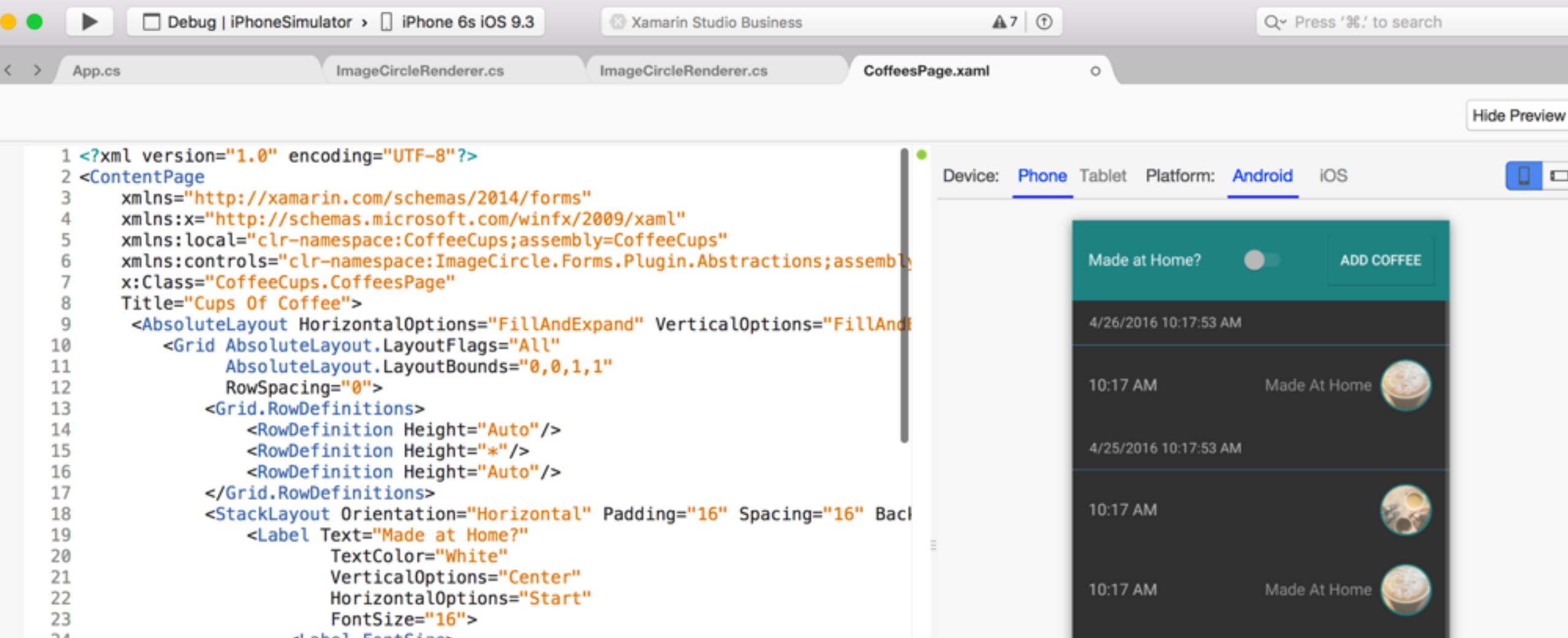
4/26/2016 10:17:53 AM

10:17 AM Made At Home 

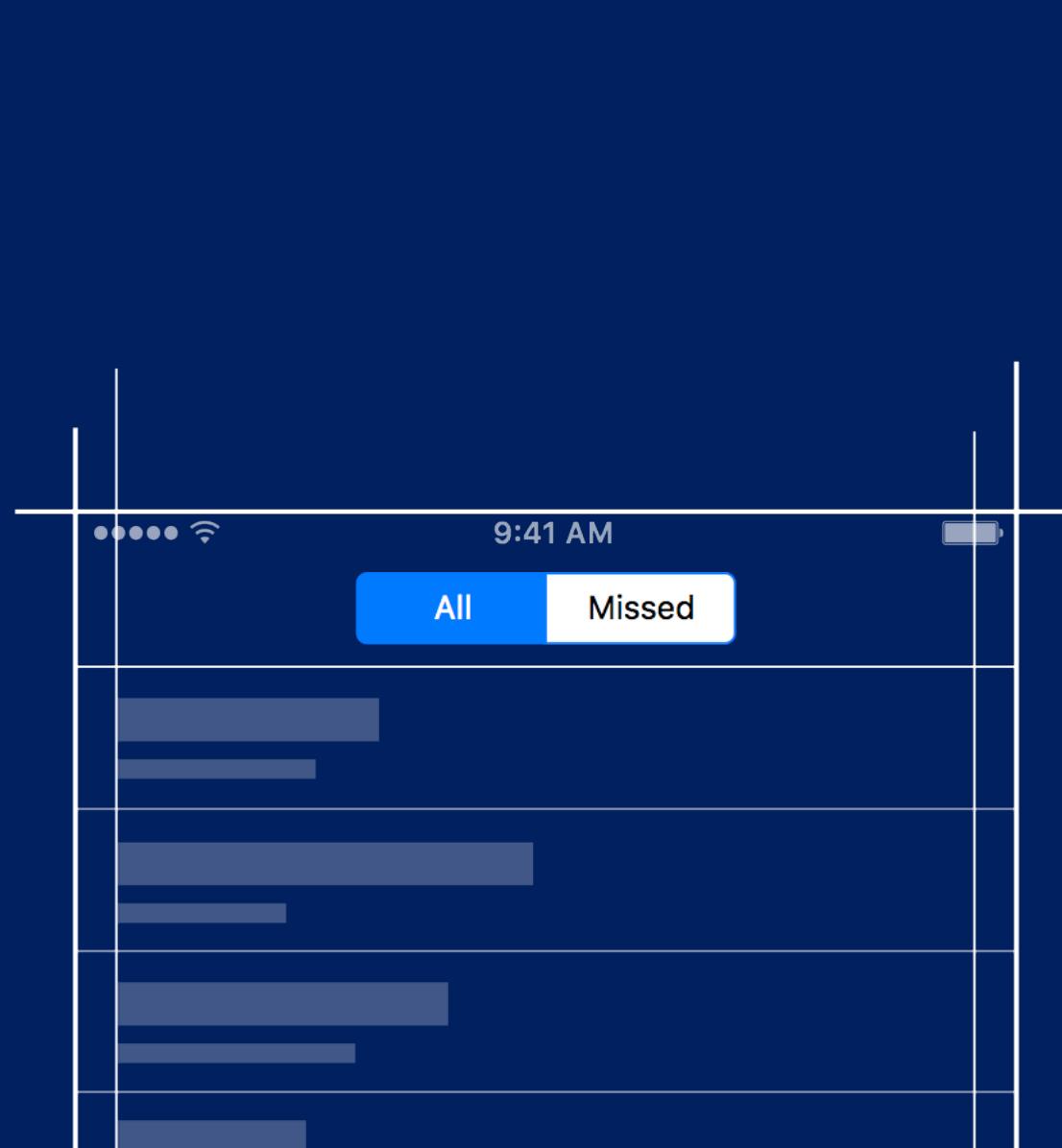
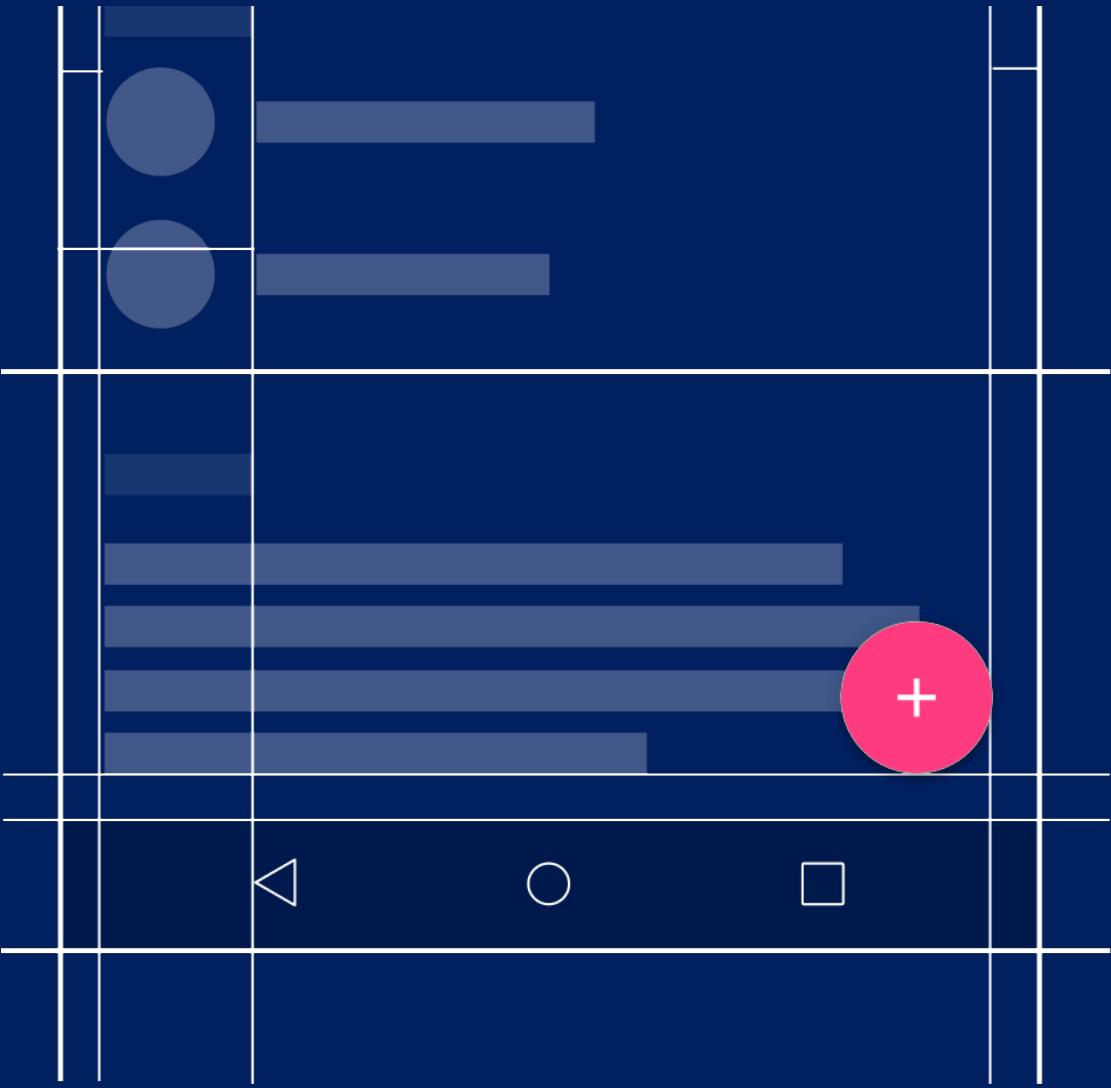
4/25/2016 10:17:53 AM

10:17 AM 

10:17 AM Made At Home 



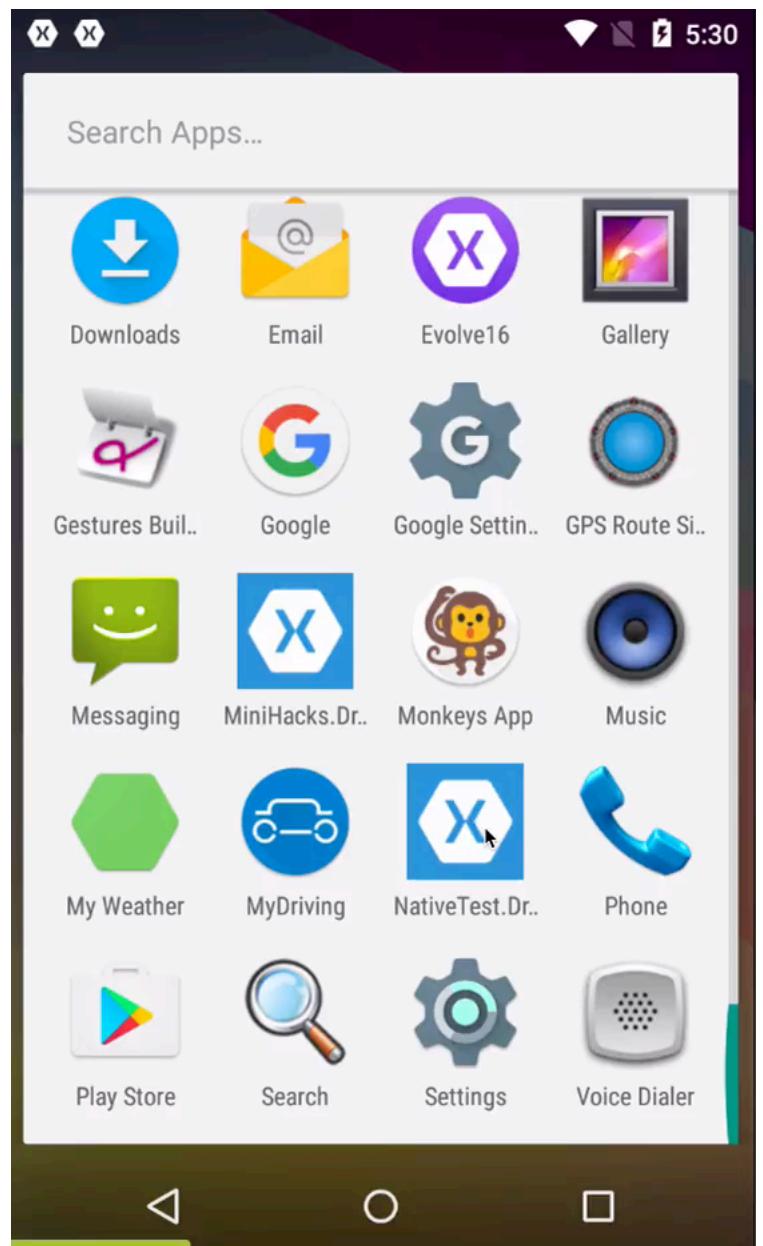
Native Embedding



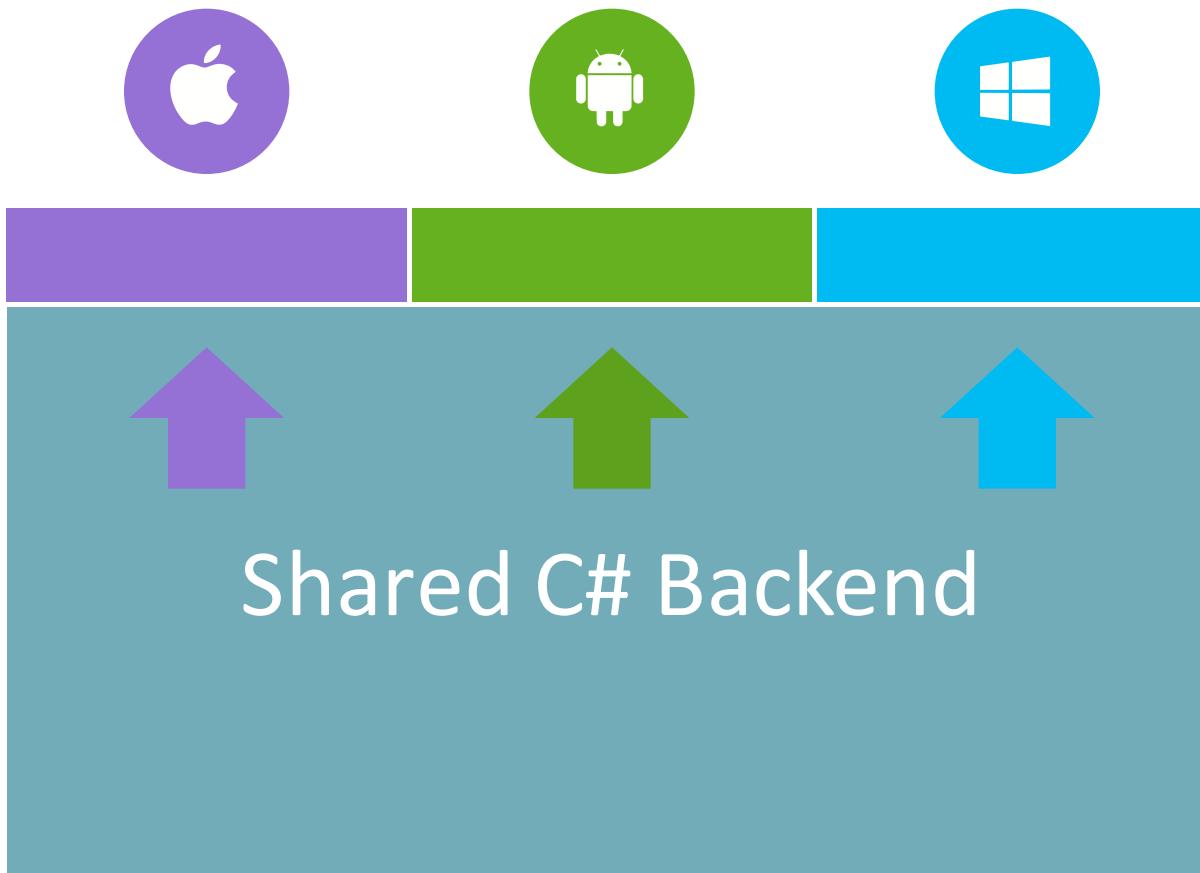
Native Embedding

```
var stack = new StackLayout
{
    Padding = 50,
    HorizontalOptions = LayoutOptions.Center,
};

#if __ANDROID__
var fab = new CheckableFab(Forms.Context);
fab.SetImageResource(Droid.Resource.Drawable.ic_fancy_fab_icon);
fab.Click += async (sender, e) =>
{
    await MainPage.DisplayAlert("Native FAB clicked",
        "Whoa!!!!!!", "OK");
};
stack.Children.Add(fab);
#endif
```



Plugins for Xamarin





UI+APIs

Battery
GPS
Lights
Notifications
Settings
Text To Speech

UI + APIs

Battery
GPS
Lights
Notifications
Settings
Text To Speech

UI + APIs

Battery
GPS
Lights
Notifications
Settings
Text To Speech

Platform Specific Code

What if we didn't have to write this code?

What if we could access it from shared code?

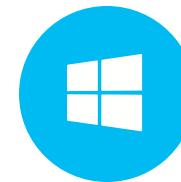
```
Speak("Hello World");
```



AVSpeechSynthesizer

TextToSpeech

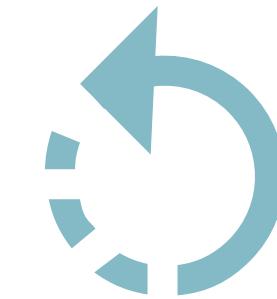
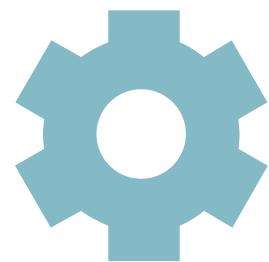
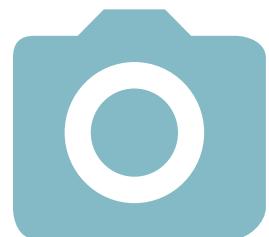
SpeechSynthesizer



Plugins for Xamarin

github.com/xamarin/plugins

Common API



Test



Challenges

- Device Fragmentation
- App Complexity
- Fast Release Cycles
- Short Sessions
- High Mobile User Expectations

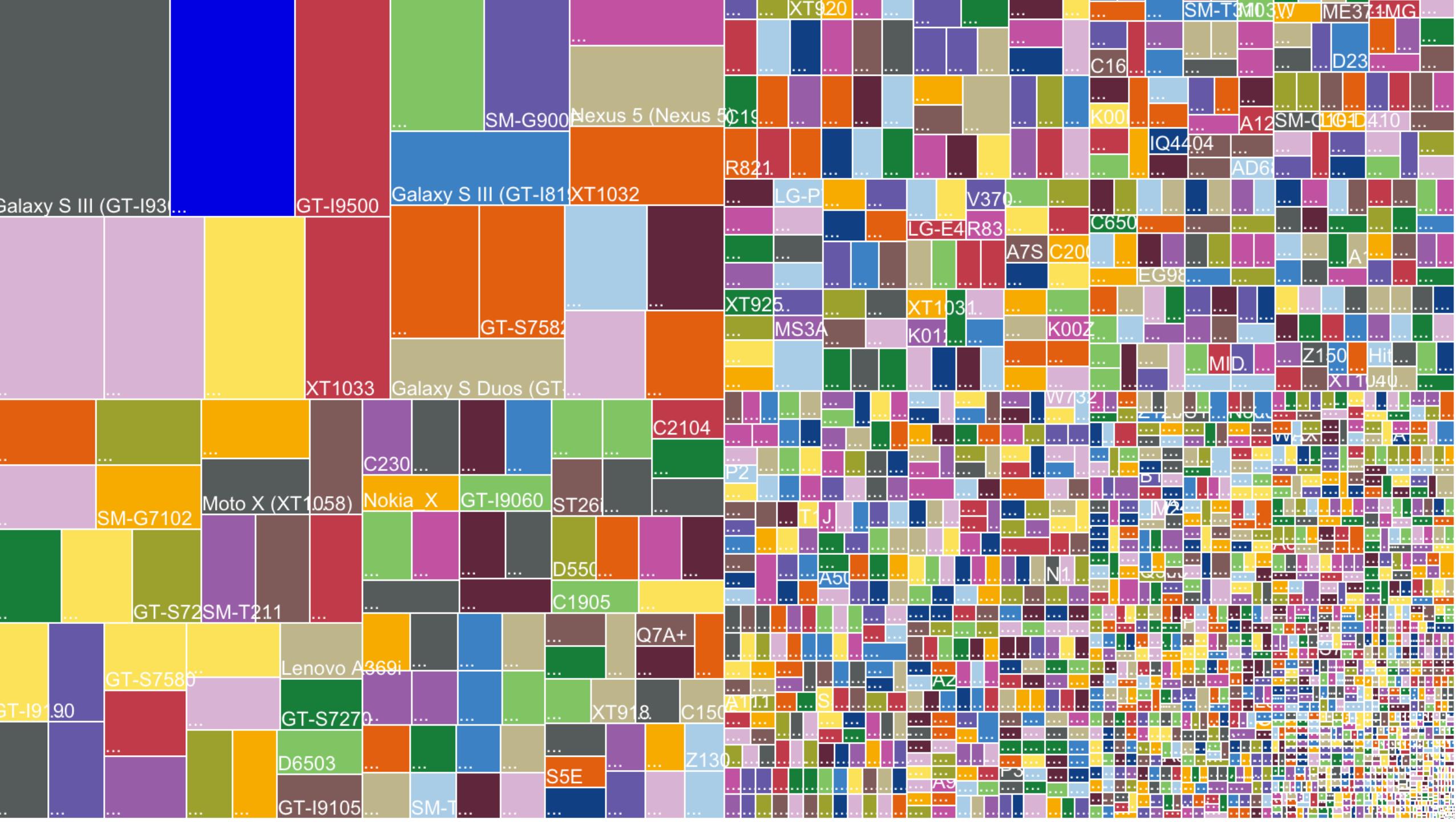
On Mobile, Quality is Hard



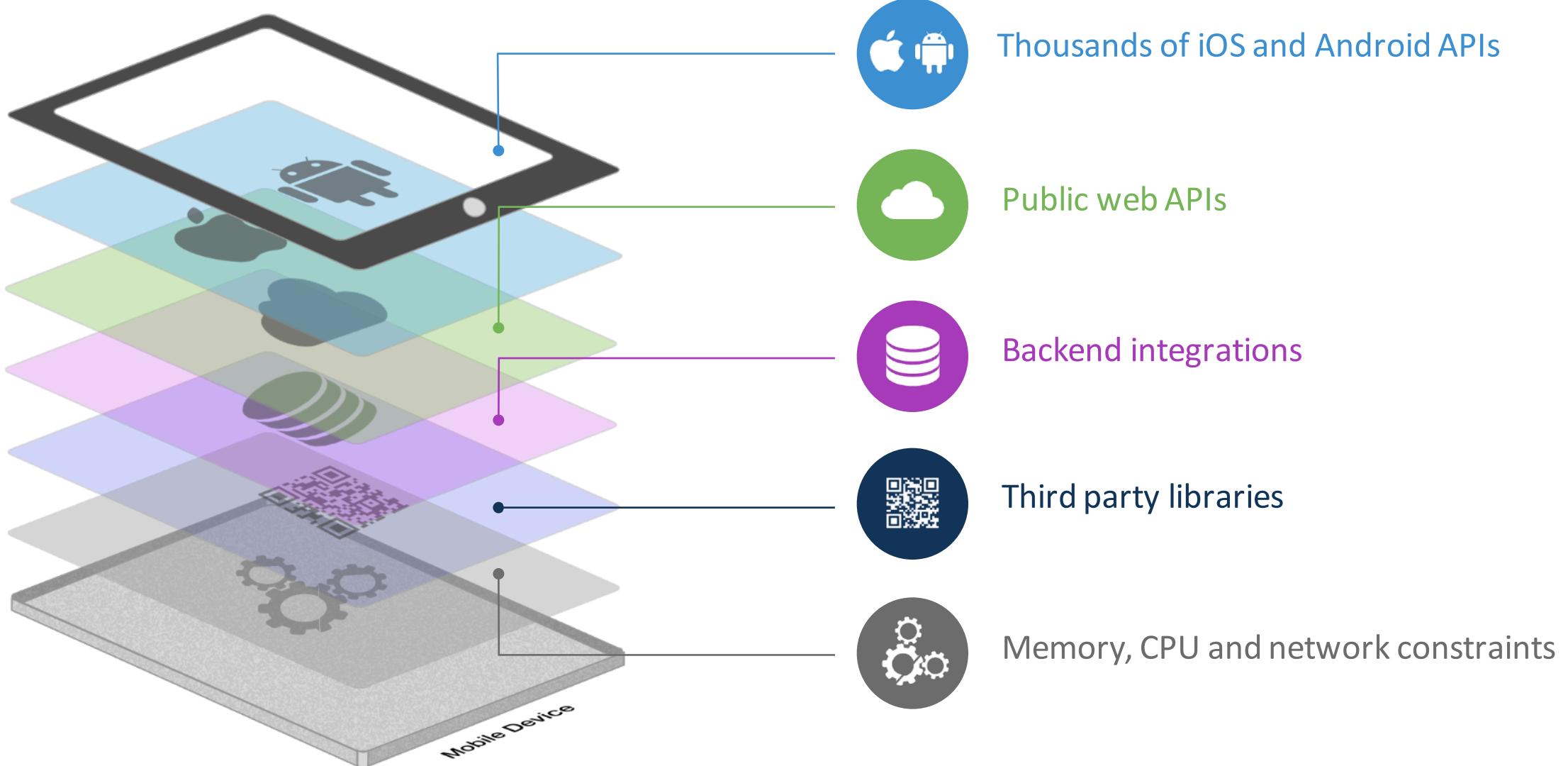
{ 5 OS versions
20 Devices
20 Languages
35 Locales
6 Screen sizes



{ 9 OS versions
19K+ Distinct devices
39 Languages
57 Locales
27 Screen sizes
15 Manufacturers
6 Screen configurations



App Complexity





Tap



Scroll



Swipe



Pinch



Multi Finger



Text Entry

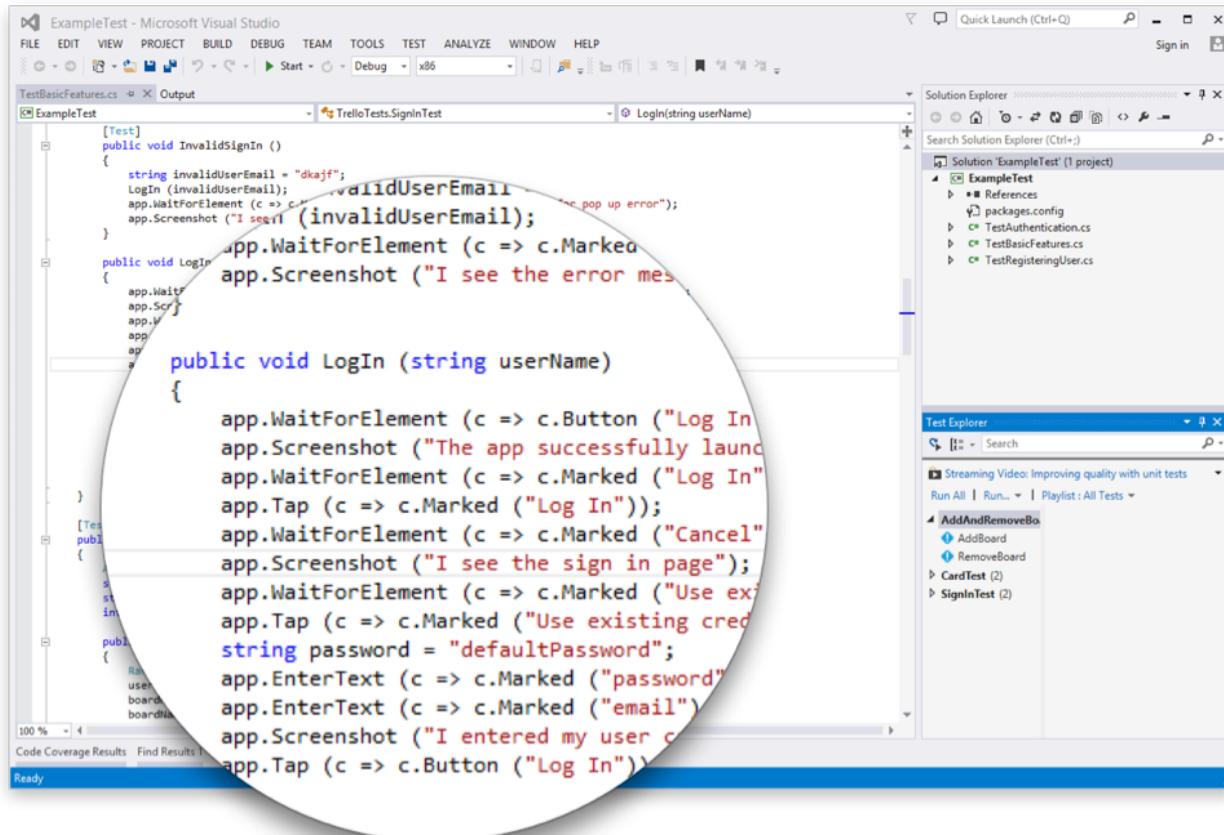


Rotation



GPS

Introducing Xamarin.UITest



Create Automated User Interface tests all in C#

Upload to the Test Cloud or run against a Device or Simulator

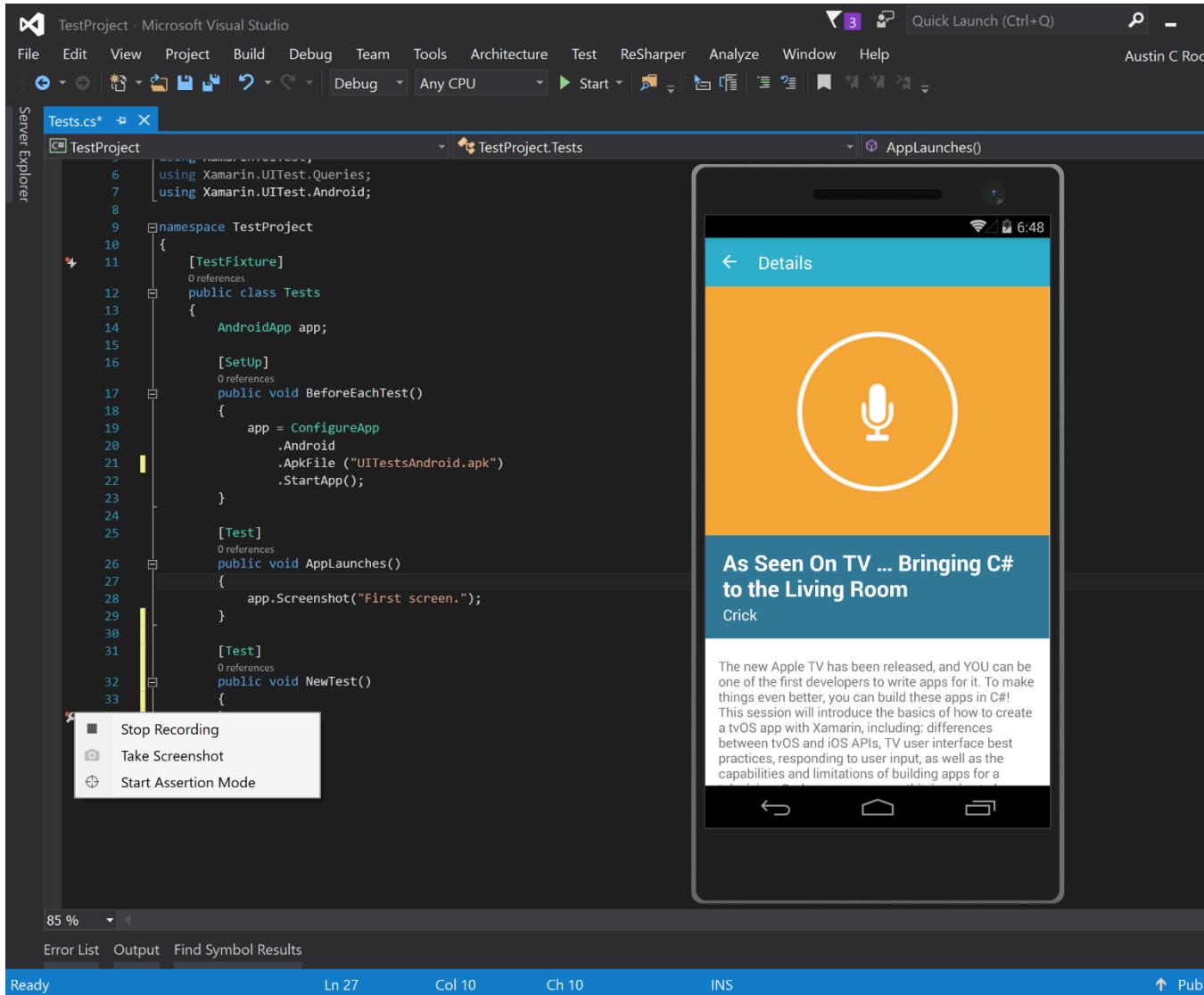
Run directly from Visual Studio or Xamarin Studio

Freely available for testing on a Simulator*

Works on ANY app: Native, Hybrid, or Xamarin

Test Recorder for Visual Studio

- Record UITests from Visual Studio
- Replay or ship to Test Cloud
- Use Touch Screen on Android Emulator



Test Recorder for Mac

- Record UITests for iOS and Android apps
- Replay or ship to Test Cloud

The image shows the Xamarin Test Recorder application running on a Mac. On the left, the recorder's interface displays a sequence of recorded interactions:

- Record button (red)
- Run button (play)
- Stop button (black square)
- Open button (green leaf)
- Device: Nexus 5 (Lollipop) - 5.1
- App: Simple Credit Car...
- Tap on EditText [creditCardNumberText]
- Type text: 123456789012345
- Tap on Button "Validate Credit Card"
- Tap on TextView "Credit card number..."

The central area shows the generated C# code for these interactions:

```
public void NewTest ()  
{  
  
    app.Tap(x => x.Class("EditText").Id("creditCardNumberText"));  
  
    app.EnterText(x => x.Class("EditText").Id("creditCardNumberText"),  
        "123456789012345");  
  
    app.Tap(x => x.Class("Button").Id("validateButton").Text("Validate  
Credit Card"));  
  
    app.Tap(x =>  
        x.Class("TextView").Id("errorMessagesText").Text("Credit card number  
is too short."));  
}
```

To the right, a screenshot of the "Simple Credit Card" app running on an Android device (Nexus 5, Lollipop) shows the following state:

- A green hexagonal icon with the text "Enter Credit Card Number".
- An input field containing the text "123456789012345".
- A large blue button labeled "Validate Credit Card".
- An error message at the bottom stating "Credit card number is too short."



Xamarin Test Cloud

xamarin.com/testcloud

The screenshot shows the Xamarin Test Cloud interface for a test run of the Flipboard app. The top navigation bar indicates the project is 'Flipboard' and the branch is 'master', with a timestamp of 'Sep 23, 2014 7:30:39 PM'. The user 'Vinicius' is logged in.

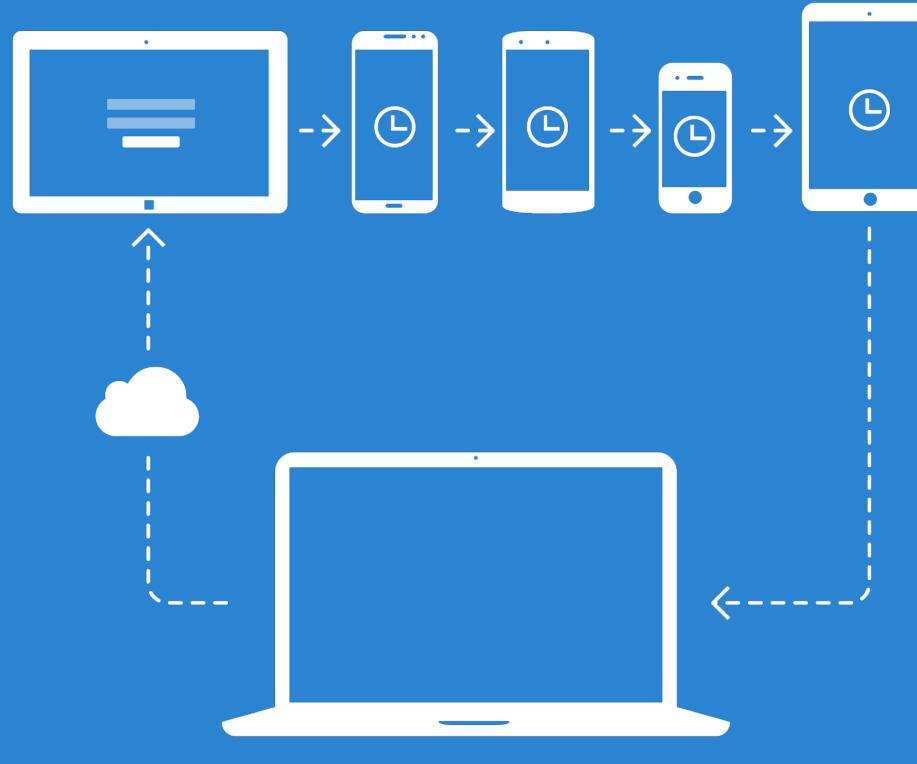
The left sidebar displays a list of test cases for the 'Sign in' scenario:

- User creates an account (5 green checkmarks)
- Given I am on the start screen
 - When I go to the login screen (5 green checkmarks)
 - And I enter valid credentials
 - Then I should be logged in
- User signs in with Facebook (green checkmark)
- User signs in with Google (green checkmark)
- User has incorrect password (green checkmark)
- User has incorrect email (3 red error icons)
- User signs out (green checkmark)

The main area shows 15 smartphone icons representing different device models and Android versions, each displaying the Flipboard welcome screen. The devices listed are:

- LG Nexus 5 (Android 4.4.2)
- Samsung Galaxy S II (Android 4.1.2)
- Samsung Galaxy S III (Android 4.1.2)
- Samsung Galaxy S Duos (Android 4.0.4)
- Samsung Galaxy Core (Android 4.1.2)
- Samsung Galaxy Grand Duos (Android 4.2.2)
- Samsung Galaxy S Duos 2 (Android 4.2.2)
- LG Nexus 4 (Android 4.4.2)
- HTC One (Android 4.4.2)
- Samsung Galaxy Note (Android 4.1.2)
- Sony Xperia Z (Android 4.2.2)
- Samsung Galaxy S5 (Android 4.4.2)
- LG G2 (Android 4.4.2)
- Samsung Galaxy S4 (Android 4.4.2)
- Samsung Galaxy Note 3 (Android 4.4.2)

Approaches to Mobile Testing



Device Remoting

Tests are performed one at the time,
which consumes more time and delays
bugs detection



Automated Testing

Test on thousands of devices simultaneously,
saving lots of time and detecting
bugs more quickly

Build



Visual Studio Team Services

- Code
- Work
- Build
- Test
- Release

The screenshot displays the Visual Studio Team Services interface for the 'MyDriving' project. The main navigation bar includes links for HOME, CODE, WORK, BUILD, TEST, and RELEASE. The current view is the 'Overview' page.

Build Status:

- MyDriving.Xamarin.iOS:** Completed on 5/10/2016. A green bar chart shows a single successful build.
- MyDriving.Xamarin.Android:** Completed on 5/10/2016. A green bar chart shows multiple successful builds.
- MyDriving.Xamarin.UWP:** Completed on 5/10/2016. A green bar chart shows two successful builds.

Xamarin Test Cloud: Shows test results from March 21, 2016, at 1:05:53 AM. The app version is 1.0.0.36 (36). The results indicate 0/13 tests failed and 0/22 devices failed. Key metrics include Total Device Time (620 min), Run Time (46 min), Devices (22), OS Versions (12), Peak Memory (495.05 MB), and App Size (25.66 MB). A grid of 22 smartphone icons represents the test devices.

HockeyApp Integration: Three separate sections provide links to HockeyApp for iOS, Android, and UWP. Each section includes an 'Overview' icon, a 'Download' icon, and a 'Crash repo' icon.

Visual Studio Team Services - Work

Team Services / MyDriving James Montemagno | 🔍 ⚙️ 🌐 ?

HOME CODE WORK BUILD TEST RELEASE Search work items

Backlogs Queries

You can now add more backlog levels to this team. Access this setting through the [settings dialog](#).

Features

Backlog Items

Backlog Board Forecast Off Mapping Off Parents Hide In progress items Show Filter

New + Create query Column options Add

Order	Work Item Type	Title	State	Effort	Value Area	Iteration Path	Tags
1	Bug	Fix NullReferenceException in MyTrips.Services.OBDDataProcessor+...	New		Business	MyDriving	
2	Product Backlog...	Use Release Management to deploy against multiple HockeyApp co...	New		Business	MyDriving	
3	Product Backlog...	Update Xamarin Test Cloud for Android builds with correct devices ...	New		Business	MyDriving	
4	Bug	Fix IllegalArgumentException in com.squareup.picasso.Picasso.load...	New		Business	MyDriving	
5	Bug	Fix crash in MyTrips.Droid.Services.GeolocationHelper.get_LocationS...	New		Business	MyDriving	
6	Bug	Fix ArgumentException in ObdLibAndroid.ObdWrapper+<Disconnect>...	New		Business	MyDriving	
7	Bug	Fix IllegalArgumentException in android.view.WindowManagerGlobal...	New		Business	MyDriving	
8	Bug	Fix IllegalArgumentException in com.squareup.picasso.Picasso.load...	New		Business	MyDriving	
9	Bug	Fix SQLException in Microsoft.WindowsAzure.MobileServices.SQLite...	New		Business	MyDriving	
10	Bug	Fix NotImplementedException in Microsoft.Azure.Devices.Client.Dev...	New		Business	MyDriving	
11	Bug	Fix NullReferenceException in MyTrips.Services.OBDDataProcessor+...	New		Business	MyDriving	
12	Bug	Fix NullReferenceException in ObdLibAndroid.ObdWrapper+<Init>...	New		Business	MyDriving	
13	Bug	Fix IllegalArgumentException in md511b398f68db4d0ef3494f1d176...	New		Business	MyDriving	
14	Bug	Fix ActivityNotFoundException in android.app.Instrumentation.check...	New		Business	MyDriving	
15	Bug	Fix NullPointerException in com.google.android.gms.common.inter...	New		Business	MyDriving	
16	Bug	Fix crash in Windows.Devices.Bluetooth.Rfcomm.RfcommDeviceServ...	New		Business	MyDriving	
17	Bug	Fix crash in Acr.UserDialogs.UserDialogsImpl.<>c__DisplayClass2_0...	New		Business	MyDriving	
18	Bug	Fix crash in System.Runtime.ExceptionServices.ExceptionDispatchInfo...	New		Business	MyDriving	
19	Bug	Fix crash in Windows.UI.Xaml.Controls.ContentDialog.ShowAsync	New		Business	MyDriving	
20	Bug	Fix (Exception in Windows.Networking.Sockets.StreamSocket.Cancelling...	New		Business	MyDriving	
21	Bug	Fix (Exception in Windows.UI.Popups.MessageDialog.ShowAsync)	New		Business	MyDriving	

Visual Studio Team Services - Build

Team Services / MyDriving

James Montemagno | Search work items

HOME CODE WORK BUILD TEST RELEASE

Definitions / MyDriving.Xamarin.Android-Feature | Builds

Build Options Repository Variables Triggers General Retention History

Save Queue build... Undo

Add build step...

Replace tokens for BingMaps Replace Tokens

NuGet restore src/MobileApps/MyDriving.XS.sln NuGet Installer

Update Version Name Version Assemblies

Update Version Code Version Assemblies

Download keystore Command Line

Activate Xamarin license Xamarin License

Build and Sign Android Project Xamarin.Android

Deactivate Xamarin license Xamarin License

Build tests MSBuild

Test in Xamarin Test Cloud Xamarin Test Cloud

Copy Files to: \$(build.artifactstagingdirectory) Copy Files

Publish Artifact: drop Publish Build Artifacts

Source Path: src/MobileApps/MyDriving/MyDriving.Utils/

Target File Pattern: Logger.cs

Advanced

Control Options

Enabled:

Continue on error:

Always run:

More Information

Explorer

Search all definitions...

My favorites

Team favorites

- MyDriving.Xamarin.Android (Completed 5 hours ago)
- MyDriving.Xamarin.iOS (Completed 5 hours ago)
- MyDriving.Xamarin.UWP (Completed 5 hours ago)

All build definitions

- All build definitions
- MyDriving.Services
- MyDriving.Xamarin.Android
- MyDriving.Xamarin.Android-Feature (selected)
- MyDriving.Xamarin.Android-Regression
- MyDriving.Xamarin.iOS
- MyDriving.Xamarin.iOS Evolve
- MyDriving.Xamarin.iOS-Feature
- MyDriving.Xamarin.iOS-Regression
- MyDriving.Xamarin.UWP
- MyDriving.Xamarin.UWP-Feature
- MyDriving.Xamarin.UWP-Regression

XAML definitions

- All XAML definitions

Visual Studio Team Services - Test

Team Services / MyDriving James Montemagno

HOME CODE WORK BUILD TEST RELEASE

Test plan Runs Machines* **Xamarin Test Cloud** Load test

Xamarin Test Cloud API Key Save API Key Help with Xamarin Test Cloud

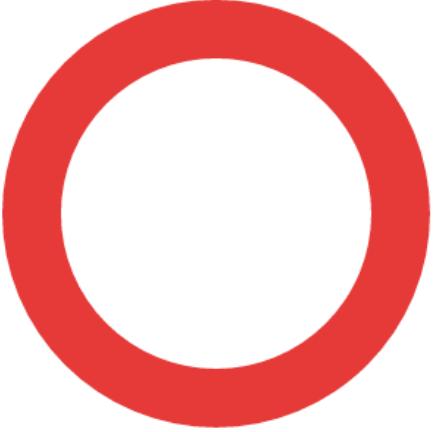
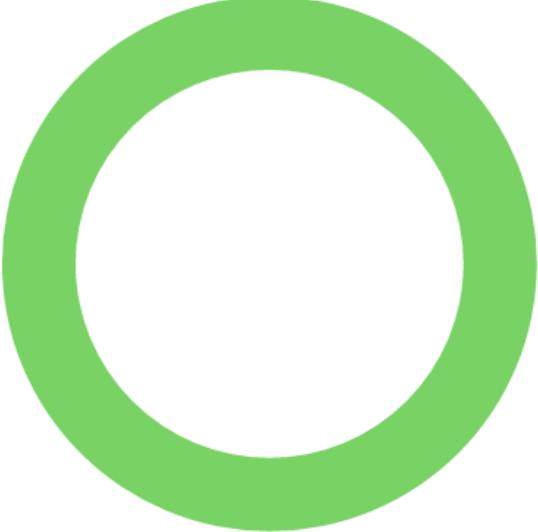
Recent Test Cloud test runs

MyDriving 2016-05-12T18:01:22.591Z

MyDriving MyDriving

Tests View Xamarin Test Cloud for detailed reports Devices View Xamarin Test Cloud for detailed reports

0/12 tests failed 0/19 devices



Current Trip Tests(iOS)

Record Trip Test

Login Tests(iOS)

Skip Authentication Test

12 / 12 Apple iPhone 6s - ios 9.0.2
12 / 12 Apple iPhone 6s Plus - ios 9.1
12 / 12 Apple iPhone 6s Plus - ios 9.1
12 / 12 Apple iPhone 6s - ios 9.0.2

Visual Studio Team Services - Release

Team Services / MyDriving

James Montemagno | ?

HOME CODE WORK BUILD TEST RELEASE

Search work items

Explorer

Go here for help getting started with Release Management.

Definition: MyDriving.Xamarin.Android Releases

Environments Artifacts Configuration Triggers General History

Release Definitions

- All release definitions
- MyDriving.Services
- MyDriving.Xamarin.Android**
- MyDriving.Xamarin.iOS
- MyDriving.Xamarin.UWP

Add environment Add tasks

Alpha Testers 1 / 1 tasks enabled 0
Beta Testers 0 / 1 tasks enabled 0
Push to Store 0 tasks 0

Deploy MyTrips to HockeyApp

HockeyApp Connection Manage
App ID \$(HockeyAppId)
Binary File Path
Symbols File Path
Native Library File Path
Release Notes (File)
Release Notes
Publish?
Mandatory?
Notify Users?

Building IoT or Mobile solutions are fun and exciting. This year for Build, we wanted to show the amazing scenarios that can come together when these two are combined. So, we went and developed a sample

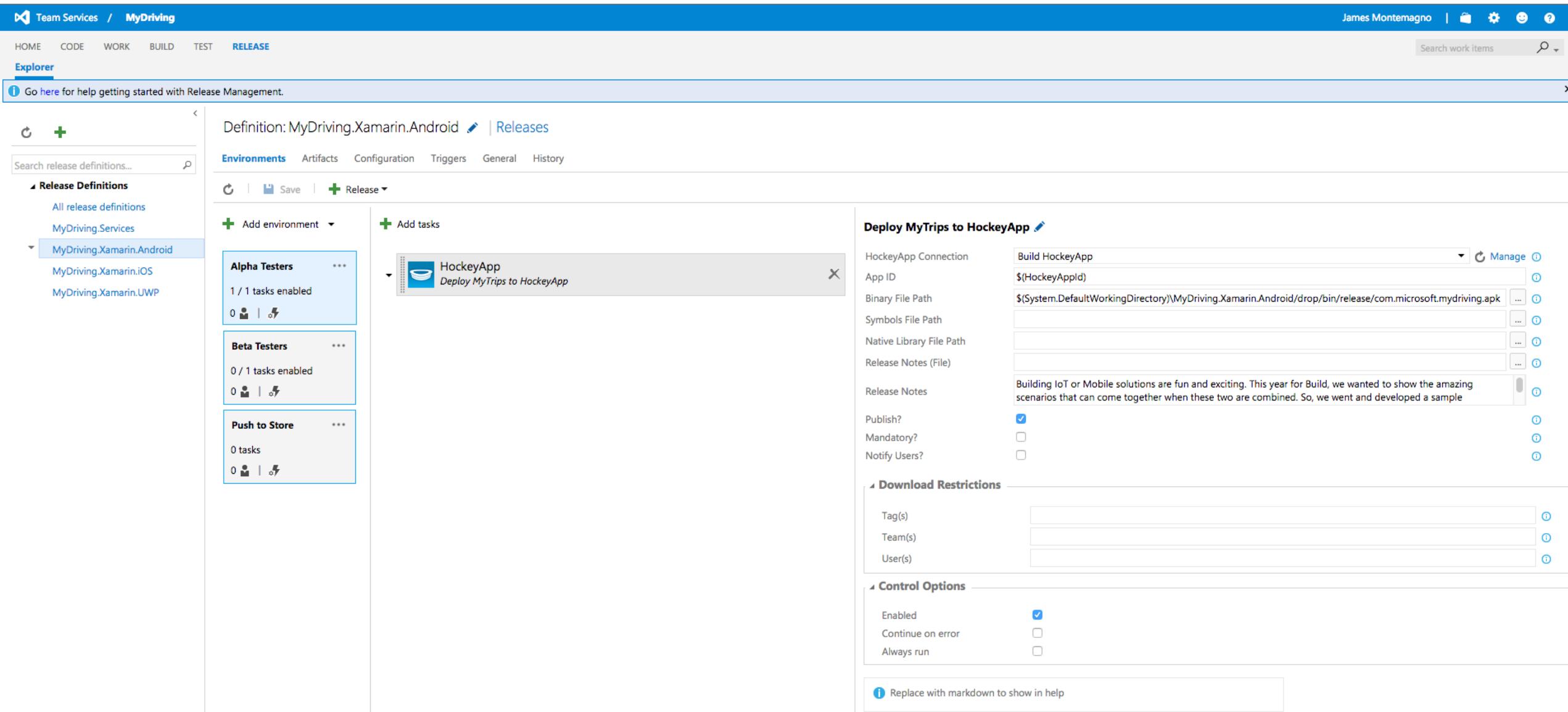
Download Restrictions

Tag(s)
Team(s)
User(s)

Control Options

Enabled
Continue on error
Always run

Replace with markdown to show in help



Distribute & Monitor



Distribute



Monitor

HockeyApp

Improve your apps with
Real-time monitoring.

Native & Managed Crash
Reports from C# API



HockeyApp

MyDriving iOS | Beta ▾ Version 1.0 ▾

Crash Group Overview Crash Logs 134 Bug Tracker

Add Annotation Status: open ▾

- [DelegatePage ThrowIndexOutOfBoundsException(System.Object se...
System.IndexOutOfRangeException: Index was outside the bounds of the array.
DelegatePage.cs, line 105

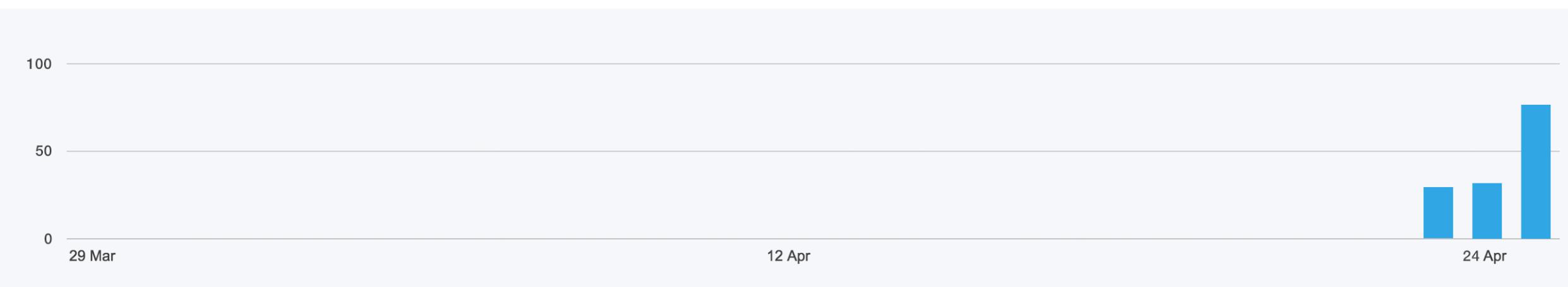
First Occurred 22 Apr 2016, 20:20 Last Occurred 24 Apr 2016, 20:40 Count 134 Jailbroken 0%

Crashes per Day
89 Last 24 Hours **134** Last 7 Days **134** Last 30 Days

Impacted Users per Day
89 Last 24 Hours **119** Last 7 Days **119** Last 30 Days

Top Affected OS's
9.3.1 Top Affected OS

Top Affected Devices
iPhone 6s Top Affected Device



Visual Studio Mobile Center



Test



Build



Distribute



Monitor

mobile.azure.com

Xamarin – Your Complete Mobile Solution



Develop



Test



Build



Distribute



Monitor

Xamarin Live Player

The screenshot shows the Visual Studio for Mac interface with the following details:

- Solution Explorer:** Shows a project named "simpleapp" with files "AppDelegate.cs" and "Info.plist".
- Code Editor:** Displays the "AppDelegate.cs" file with the following code:

```
1 using Foundation;
2 using UIKit;
```
- Output Window:** Shows the message "FinishedLaunching(UIApplication application, NSDictionary launchOptions)".
- Preview Area:** A large window titled "Xamarin Live Player" displays the text "Preview your app, live on devices with **Xamarin Live Player**". It includes an illustration of a computer monitor and a smartphone connected by a cable.
- QR Code:** A large QR code is displayed on the left side of the preview area.
- Bottom Left:** A text field contains the number "123456" and a "Connect" button.
- Bottom Right:** A status bar at the bottom shows the line "var btn = new UIButton();".
- Top Bar:** Shows "Debug | iPhoneSimulator > Pair Device..." and "Visual Studio for Mac".
- Right Side:** A large smartphone screen titled "Cody's iPhone" shows the "Xamarin Live" app with the message "Waiting for Connection...". It includes a "Pair to Visual Studio" button.
- Top Status Bar:** Shows "Record", "Cody's iPhone", "AT&T Wi-Fi", "1:49 PM", and battery level.

Thank You! Questions?

Diego Bonilla

diego.bonilla@nareia.com.uy

@dbonillanareia