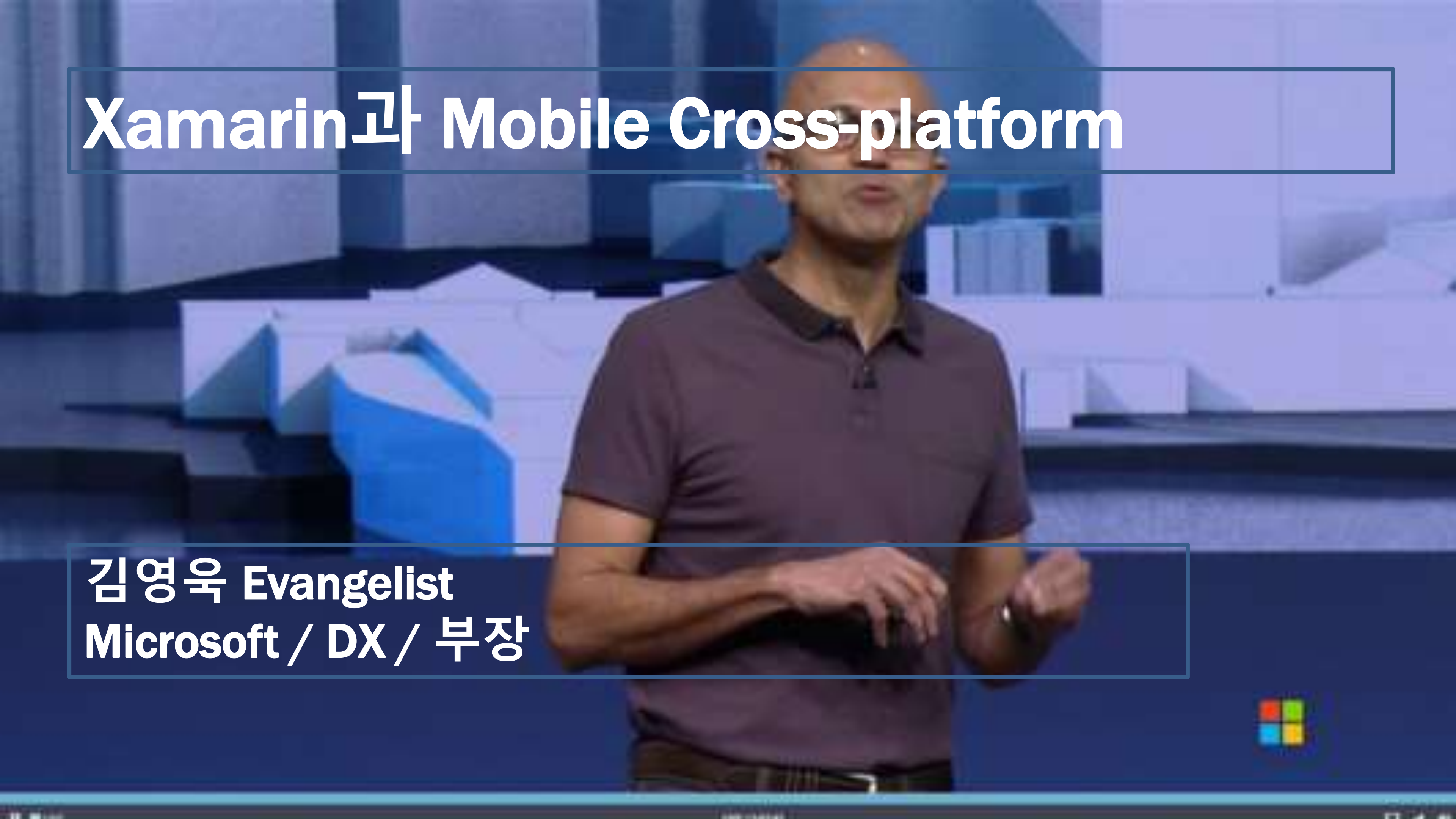
# Xamarin과 Mobile Cross-platform



김영욱 Evangelist
Microsoft / DX / 부장

/build
Microso

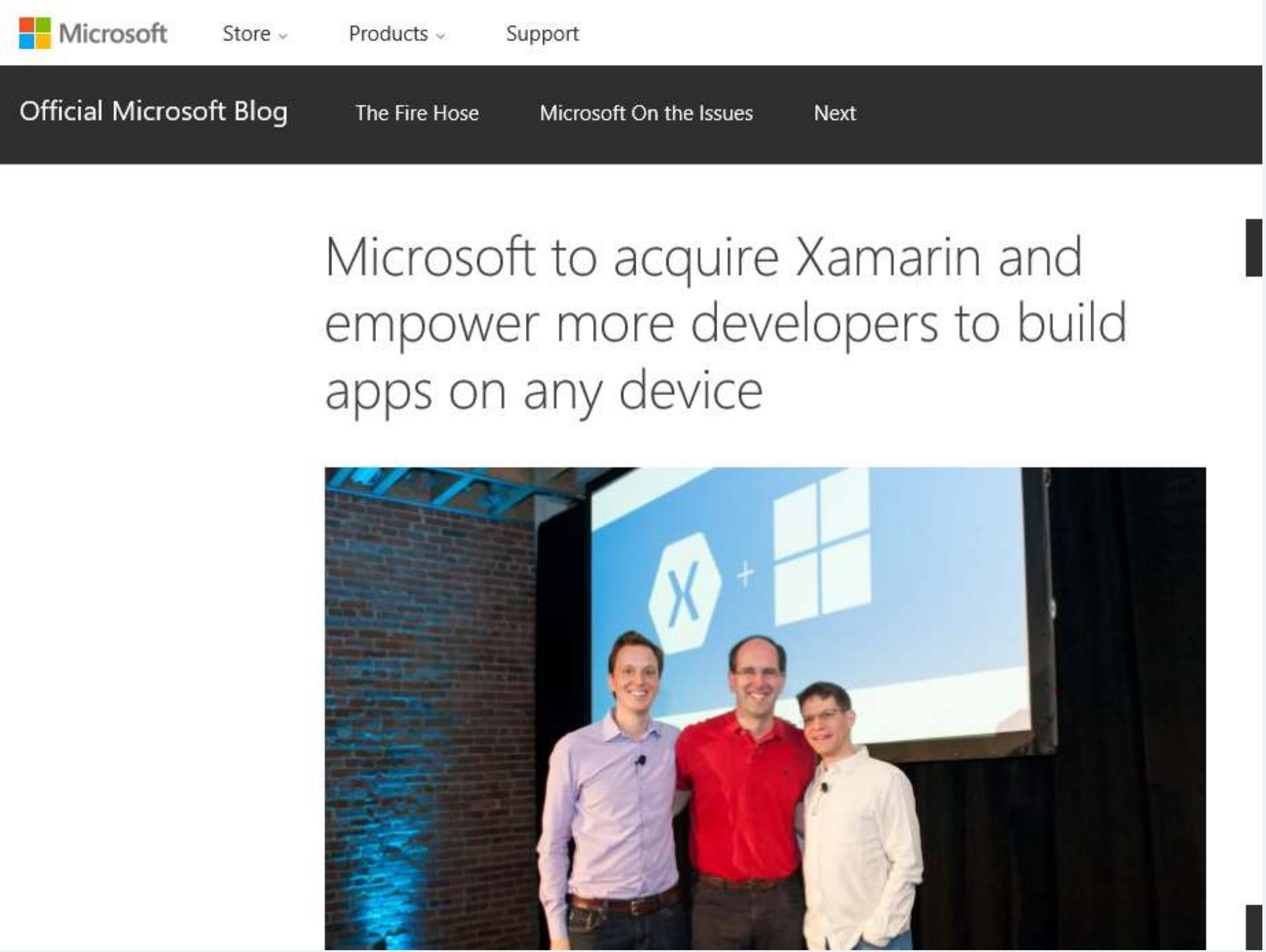
# Mobile-first, cloud-first

# Microsoft LOVE ♥

Announcing

# Open sourcing of the Xamarin runtime

# Microsoft ♥ iOS, Android developers



# Microsoft to acquire Xamarin and empower more developers to build apps on any device

Microsoft
Xamarin
인수
발표

2월 24일

Now Included for FREE

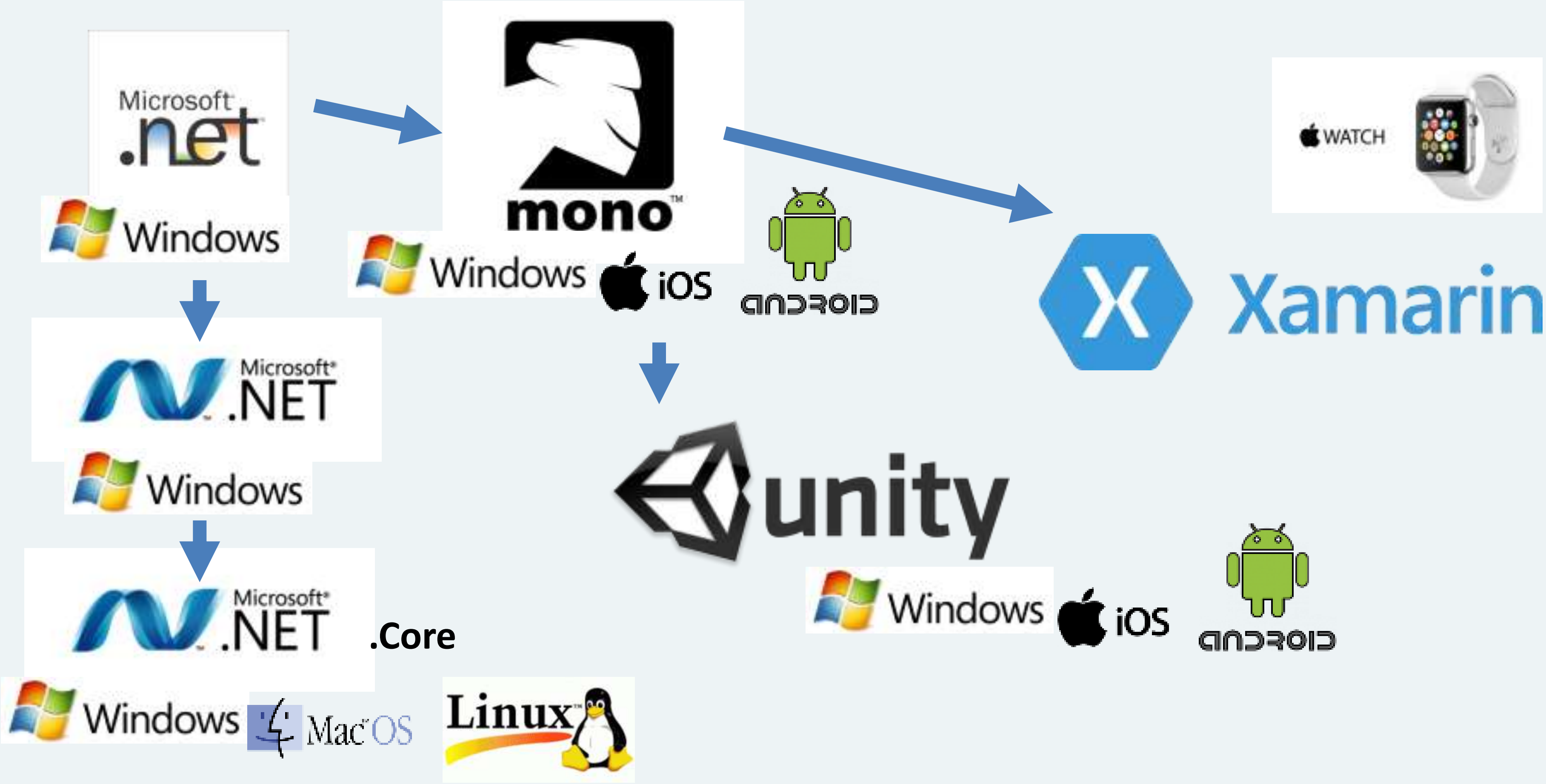
# Announcing

## Xamarin for Visual Studio

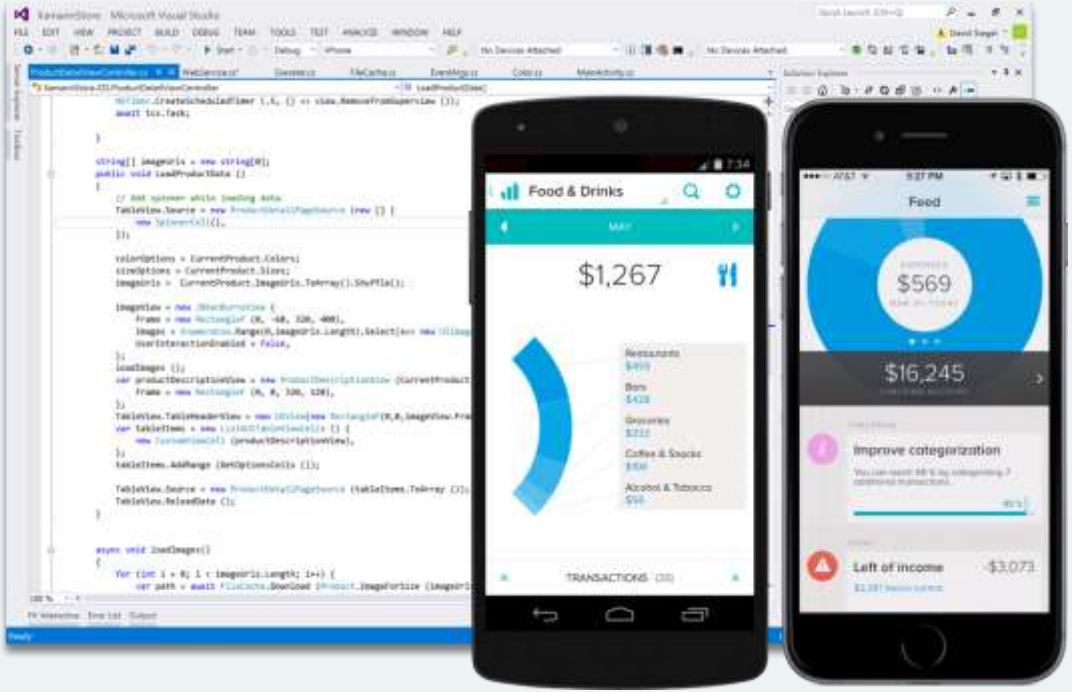
Visual Studio Enterprise

Visual Studio Professional

**Visual Studio Community**


Microsoft .net
Windows
mono
Windows
iOS
android
WATCH
Xamarin
Microsoft .NET
Windows
unity
Windows
iOS
android
Microsoft .NET
.Core
Windows
Mac OS
Linux

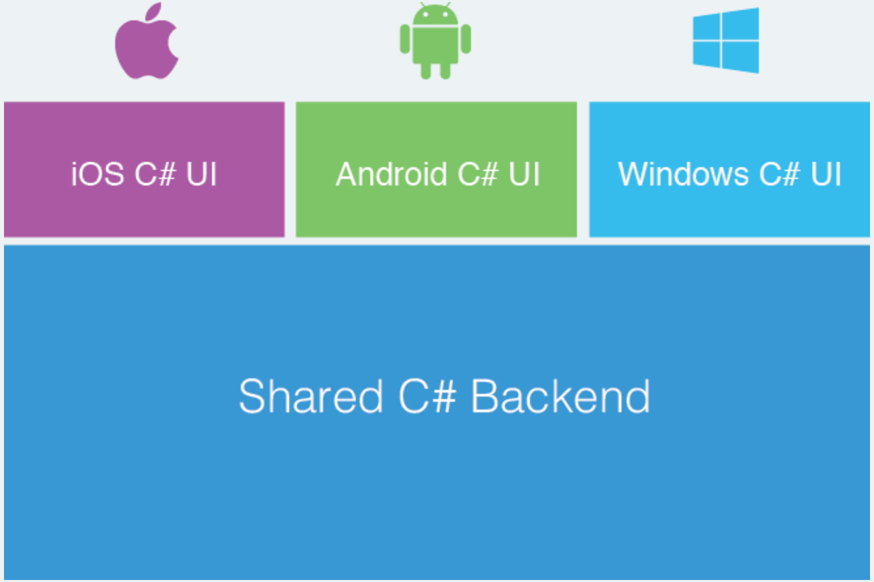
# Xamarin Platform



```
    NSTimer.CreateScheduledTimer (1.5, () => view.RemoveFromSuperview ());
    await tcs.Task;

}

string[] imageUrls = new string[0];
public void LoadProductData ()
{
    // Add spinner while loading data.
    TableView.Source = new ProductDetailPageSource (new [] {
        new SpinnerCell(),
    });

    colorOptions = CurrentProduct.Colors;
    sizeOptions = CurrentProduct.Sizes;
    imageUrls =  CurrentProduct.ImageUrls.ToArray().Shuffle();

    imageView = new JBKenBurnsView {
        Frame = new RectangleF (0, -60, 320, 400),
        Images = Enumerable.Range(0,imageUrls.Length).Select(x=> new UIImage
        UserInteractionEnabled = false,
    };
    loadImages ();
    var productDescriptionView = new ProductDescriptionView (CurrentProduct
        Frame = new RectangleF (0, 0, 320, 120),
    };
    TableView.TableHeaderView = new UIView(new RectangleF(0,0,imageView.Fra
    var tableItems = new List<UITableViewCell> () {
        new CustomViewCell (productDescriptionView),
    };
    tableItems.AddRange (GetOptionsCells ());

    TableView.Source = new ProductDetailPageSource (tableItems.ToArray ());
    TableView.ReloadData ();
}

async void loadImages()
{
    for (int i = 0; i < imageUrls.Length; i++) {
        var path = await FileCache.Download (Product.ImageForSize (imageUrl
```

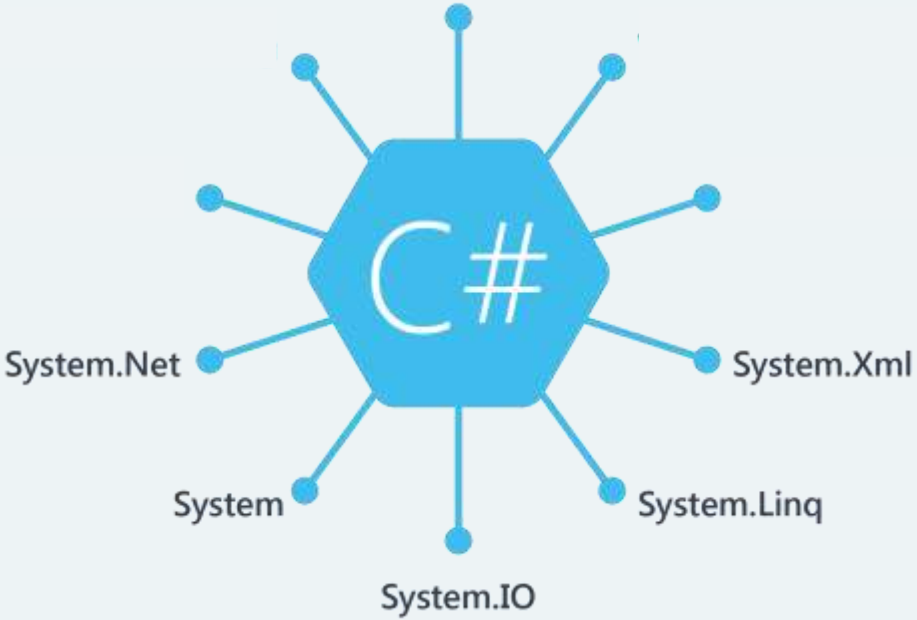
Create native iOS, Android, Mac and Windows apps in Visual Studio and C#

# Xamarin's Unique Approach

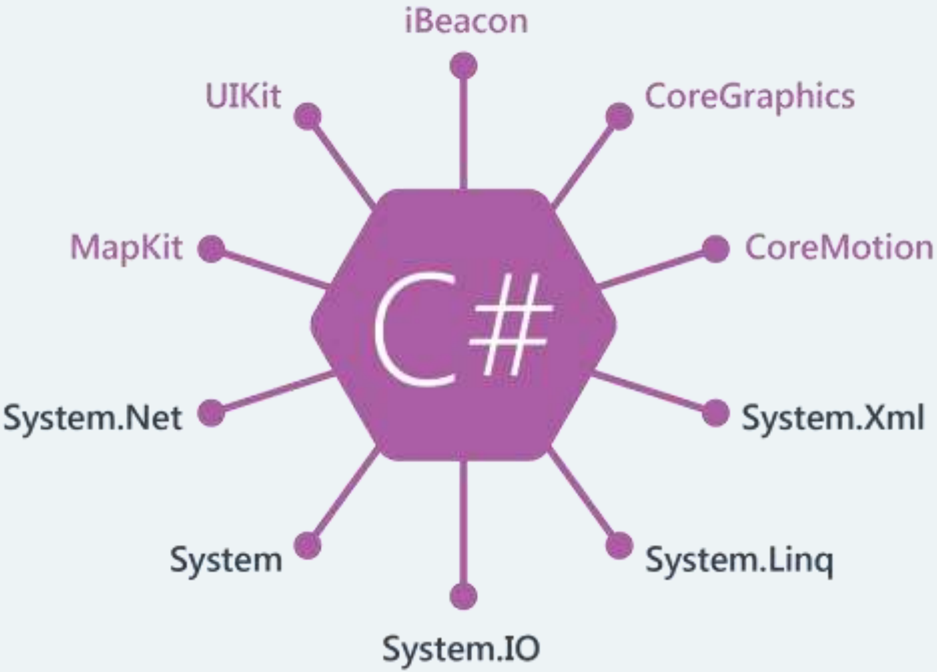


- Shared C# codebase
- 100% native API access
- High performance

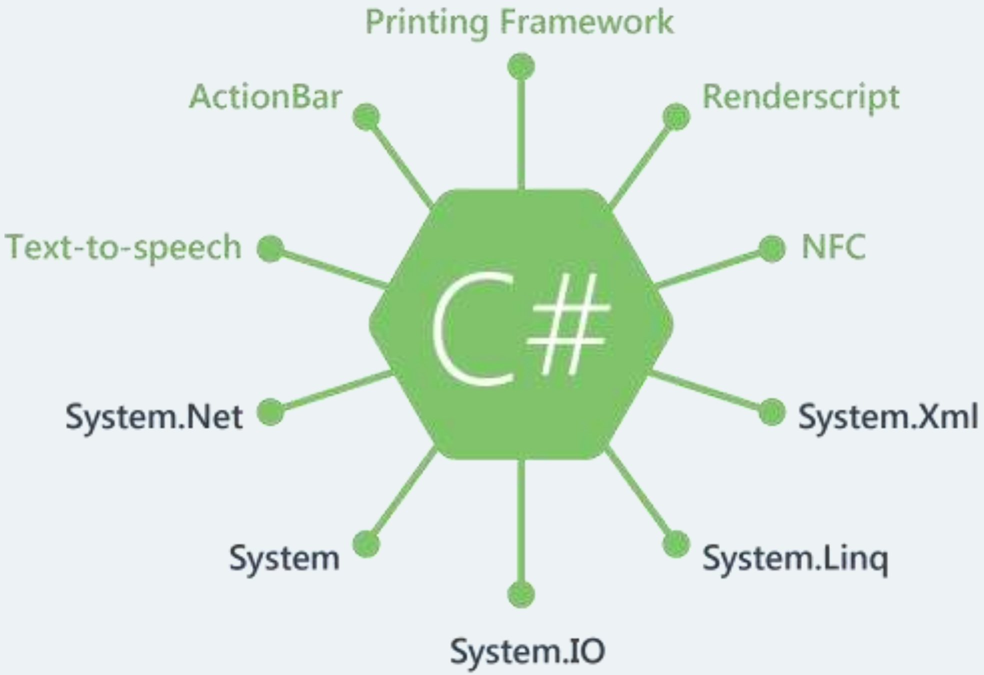
# Windows APIs



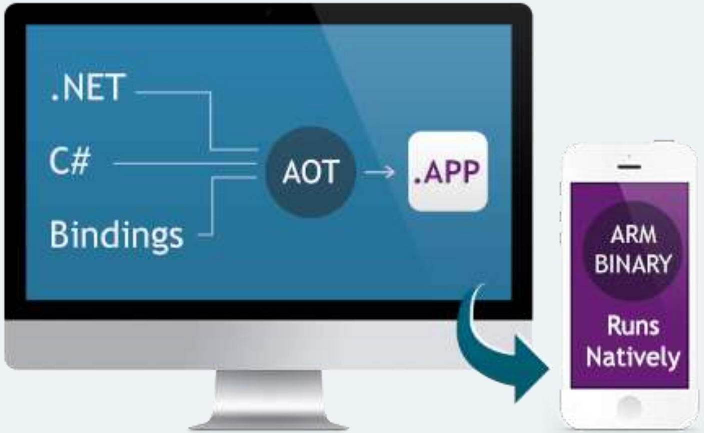
# .NET + iOS APIs | 100% Coverage



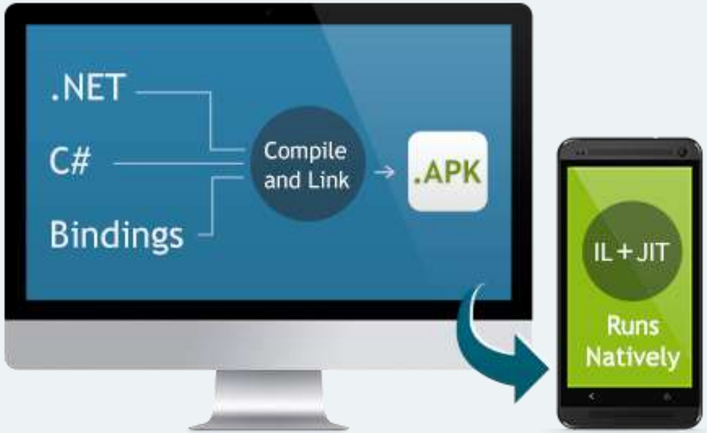
# .NET Android APIs | 100% Coverage



# Native Performance

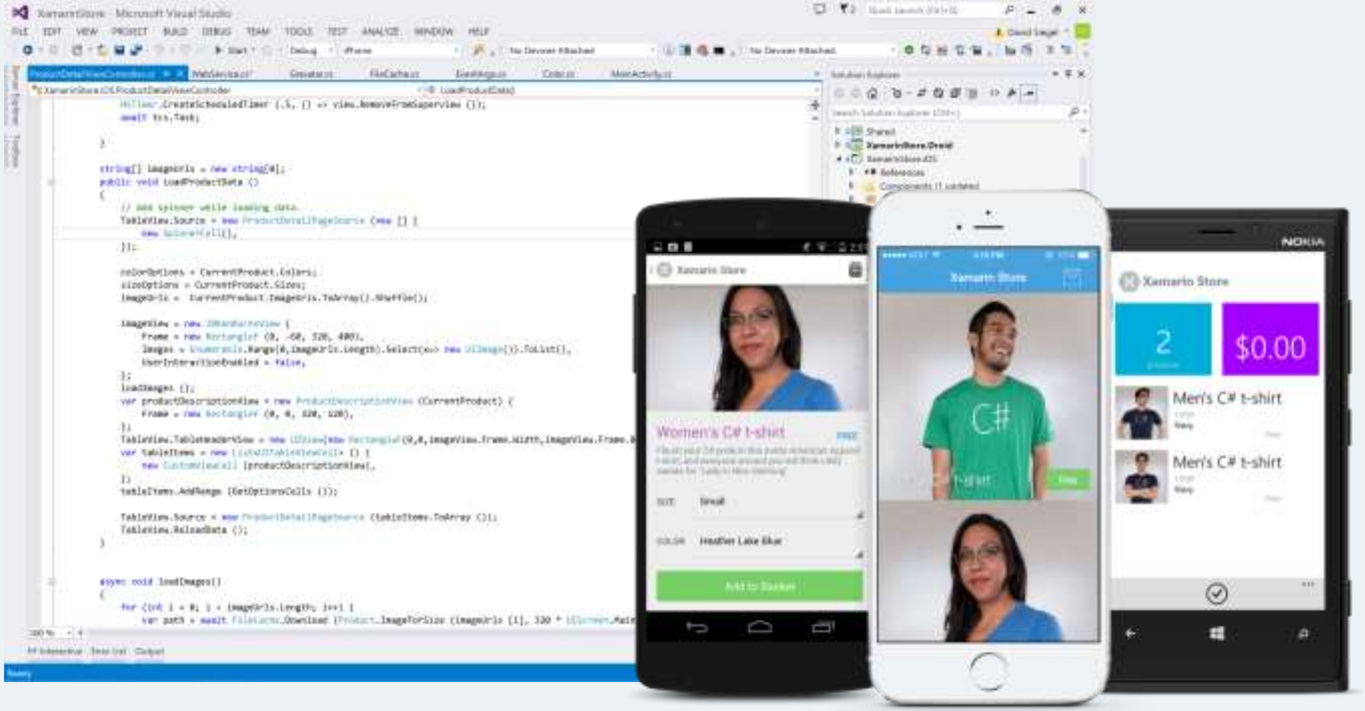


Xamarin.iOS does full Ahead Of Time (AOT) compilation to produce an ARM binary for Apple's App Store.



Xamarin.Android takes advantage of Just In Time (JIT) compilation on the Android device.

# Anything you can do in Objective-C, Swift, or Java can be done in C# with Xamarin and Visual Studio.



```
        Helpers.CreateScheduledTimer (.5, () => view.RemoveFromSuperview ());
        await tcs.Task;

    }

    string[] imageUrls = new string[0];
    public void LoadProductData ()
    {
        // Add spinner while loading data.
        TableView.Source = new ProductDetailPageSource (new [] {
            new SpinnerCell(),
        });

        colorOptions = CurrentProduct.Colors;
        sizeOptions = CurrentProduct.Sizes;
        imageUrls =  CurrentProduct.ImageUrls.ToArray().Shuffle();

        imageView = new JBKenBurnsView {
            Frame = new RectangleF (0, -60, 320, 400),
            Images = Enumerable.Range(0,imageUrls.Length).Select(x=> new UIImage()).ToList(),
            UserInteractionEnabled = false,
        };
        loadImages ();
        var productDescriptionView = new ProductDescriptionView (CurrentProduct) {
            Frame = new RectangleF (0, 0, 320, 120),
        };
        TableView.TableHeaderView = new UIView(new RectangleF(0,0,imageView.Frame.Width,imageView.Frame.B
        var tableItems = new List<UITableViewCell> () {
            new CustomViewCell (productDescriptionView),
        };
        tableItems.AddRange (GetOptionsCells ());

        TableView.Source = new ProductDetailPageSource (tableItems.ToArray ());
        TableView.ReloadData ();
    }


    async void loadImages()
    {
        for (int i = 0; i < imageUrls.Length; i++) {
            var path = await FileCache.Download (Product.ImageForSize (imageUrls [i], 320 * UIScreen.Mai
```

# Always Up-to-Date

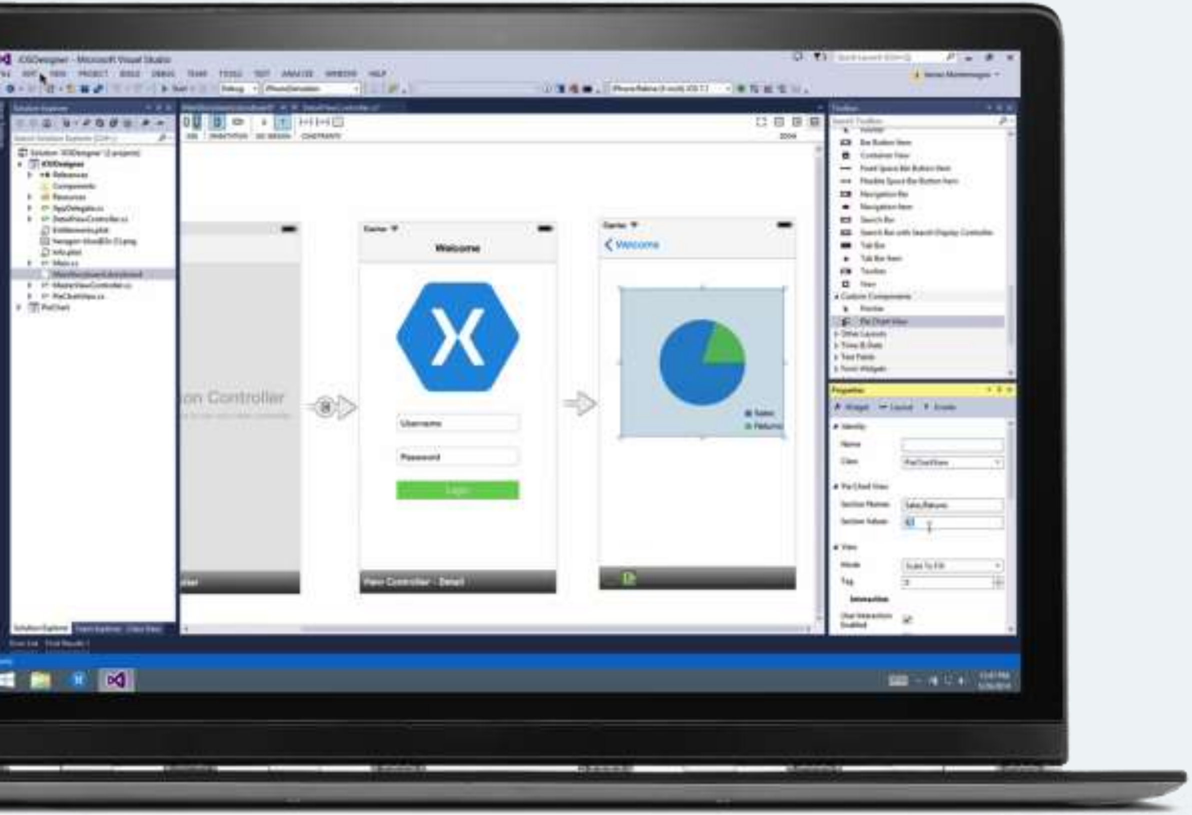
## Track record of offering same-day support:

iOS 5, iOS 6, iOS 7, iOS 7.1, iOS 8

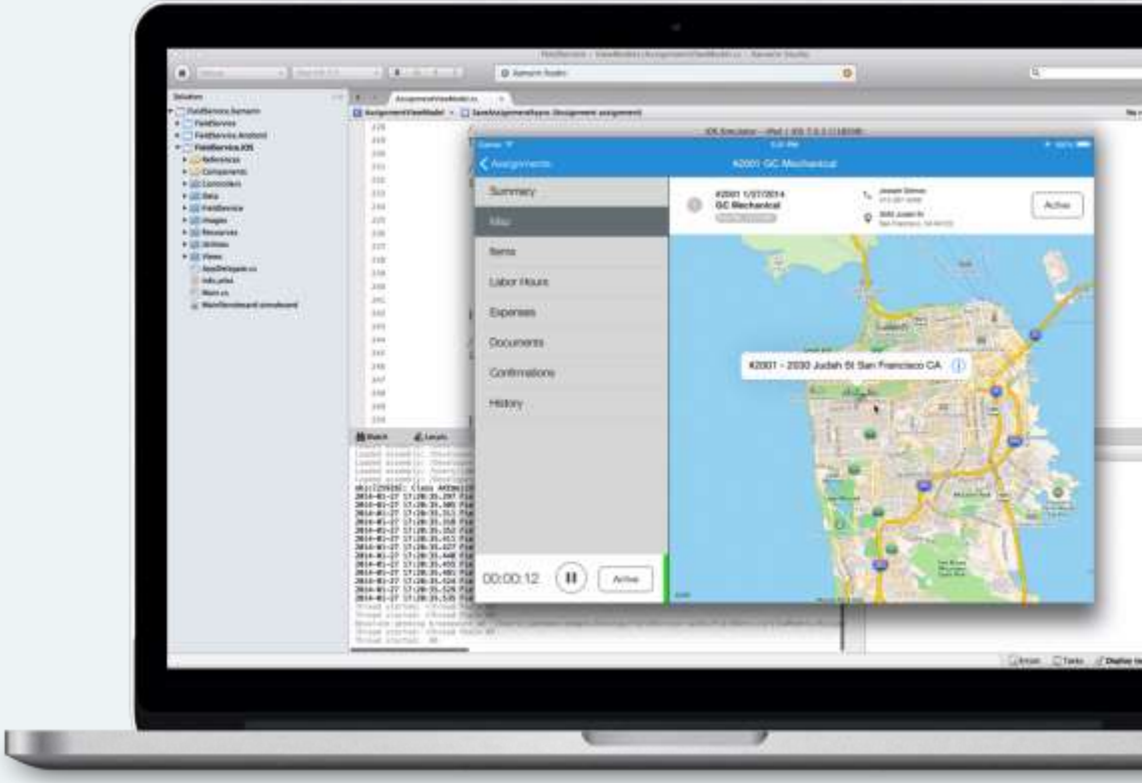
## Full support for:

- Google Glass
- Android Wear
- Apple Watch
- Amazon Fire TV
- More!

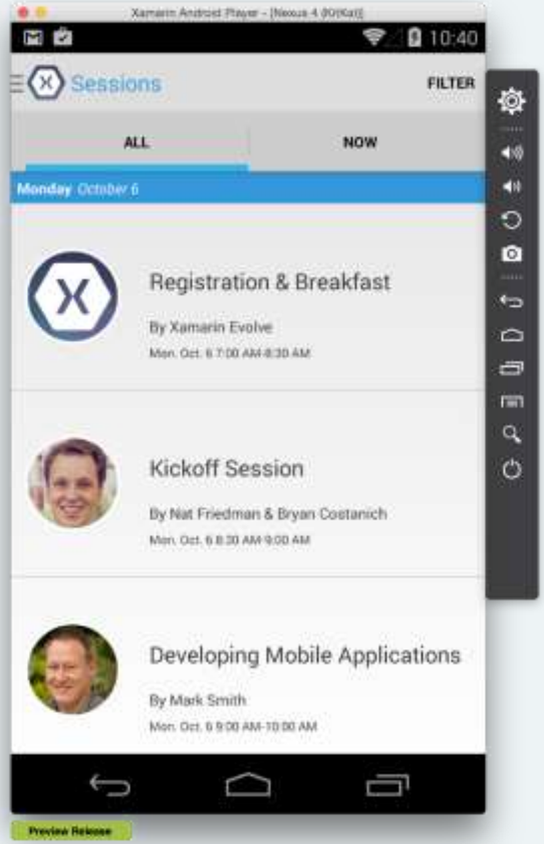
# Visual Studio – 2010+



# Xamarin Studio – Mac or PC



# Xamarin Android Player



Monday October 6

Registration & Breakfast
By Xamarin Evolve
Mon. Oct. 6 7:00 AM-8:30 AM

Kickoff Session
By Nat Friedman & Bryan Costanich
Mon. Oct. 6 8:30 AM-9:00 AM

Developing Mobile Applications
By Mark Smith
Mon. Oct. 6 9:00 AM-10:00 AM

- Rapidly demo and test app builds on Mac or PC
- Hardware-accelerated virtualization
- Runs side-by-side Windows Phone Emulator
- Free for Xamarin Subscribers

**xamarin.com/Android-Player**

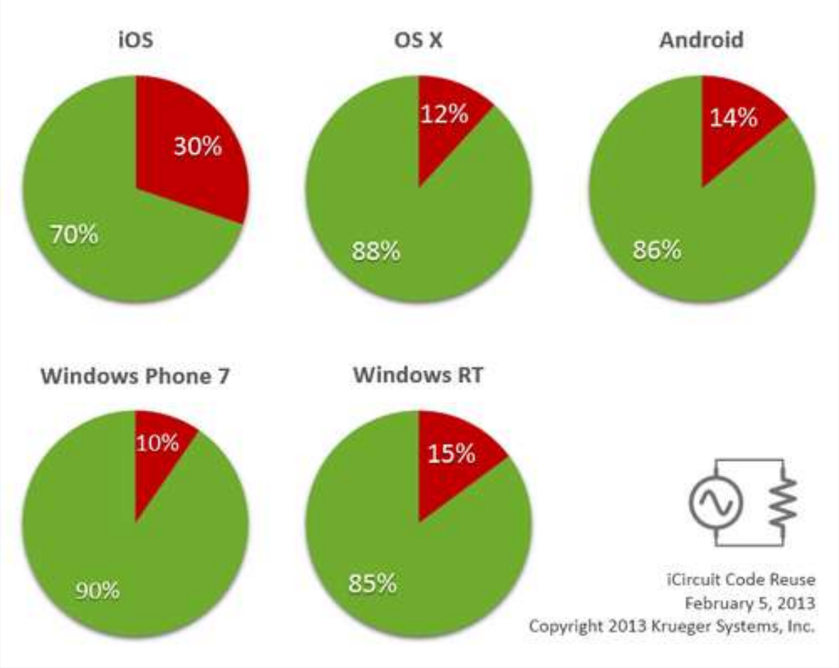
# Xamarin Native HOL

# Hello, Android Quickstart

**http://1drv.ms/1oLrBpN**

# Code Sharing

# Code Sharing: Accelerate Development



iCircuit Code Reuse
February 5, 2013
Copyright 2013 Krueger Systems, Inc.

- Up to 90% Code Reuse
- Leverage existing libraries

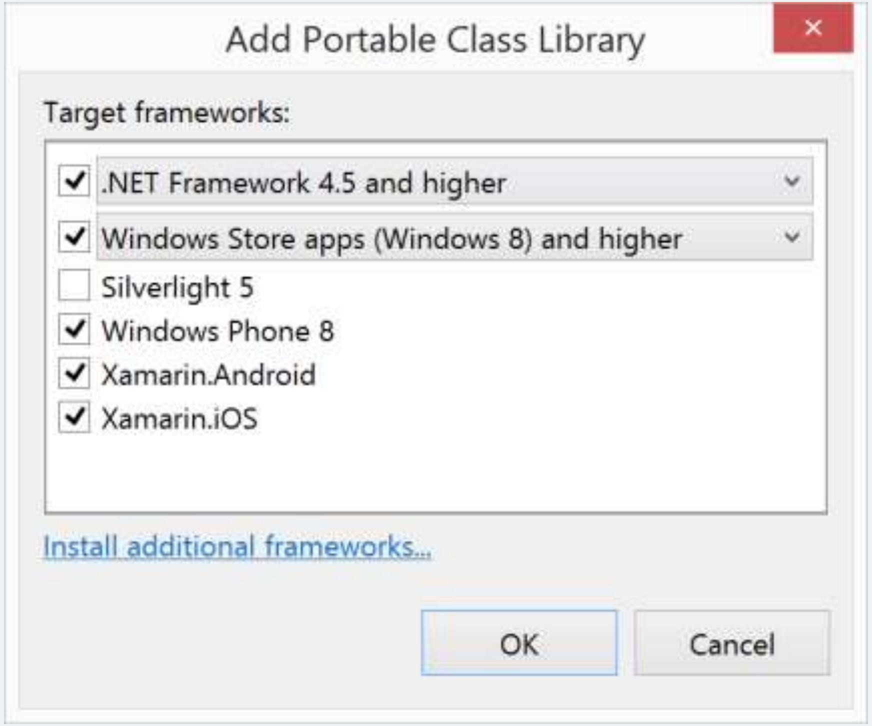
🟩 App Logic 🟥 User Interface Code

Code-sharing stats from iCircuit, built with Xamarin and featured on the App Store.

Source: http://praeclarum.org/post/42378027611/icircuit-code-reuse-part-cinq

# Portable Class Libraries

- 1 Assembly
- Multiple Platforms
- **Including**
  - **Xamarin.Android**
  - **Xamarin.iOS**



Add Portable Class Library

Target frameworks:

☑ .NET Framework 4.5 and higher

☑ Windows Store apps (Windows 8) and higher

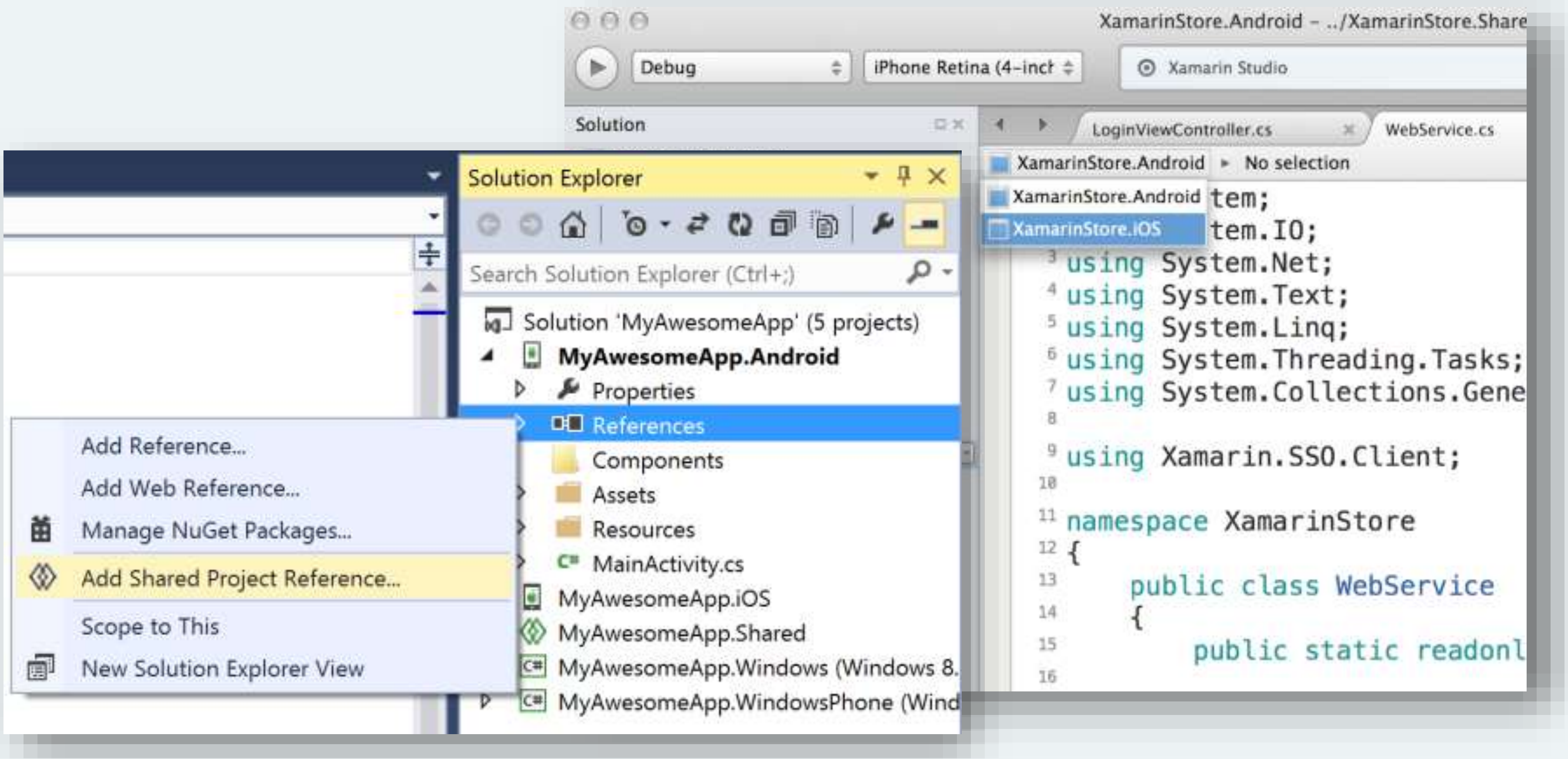
☐ Silverlight 5

☑ Windows Phone 8

☑ Xamarin.Android

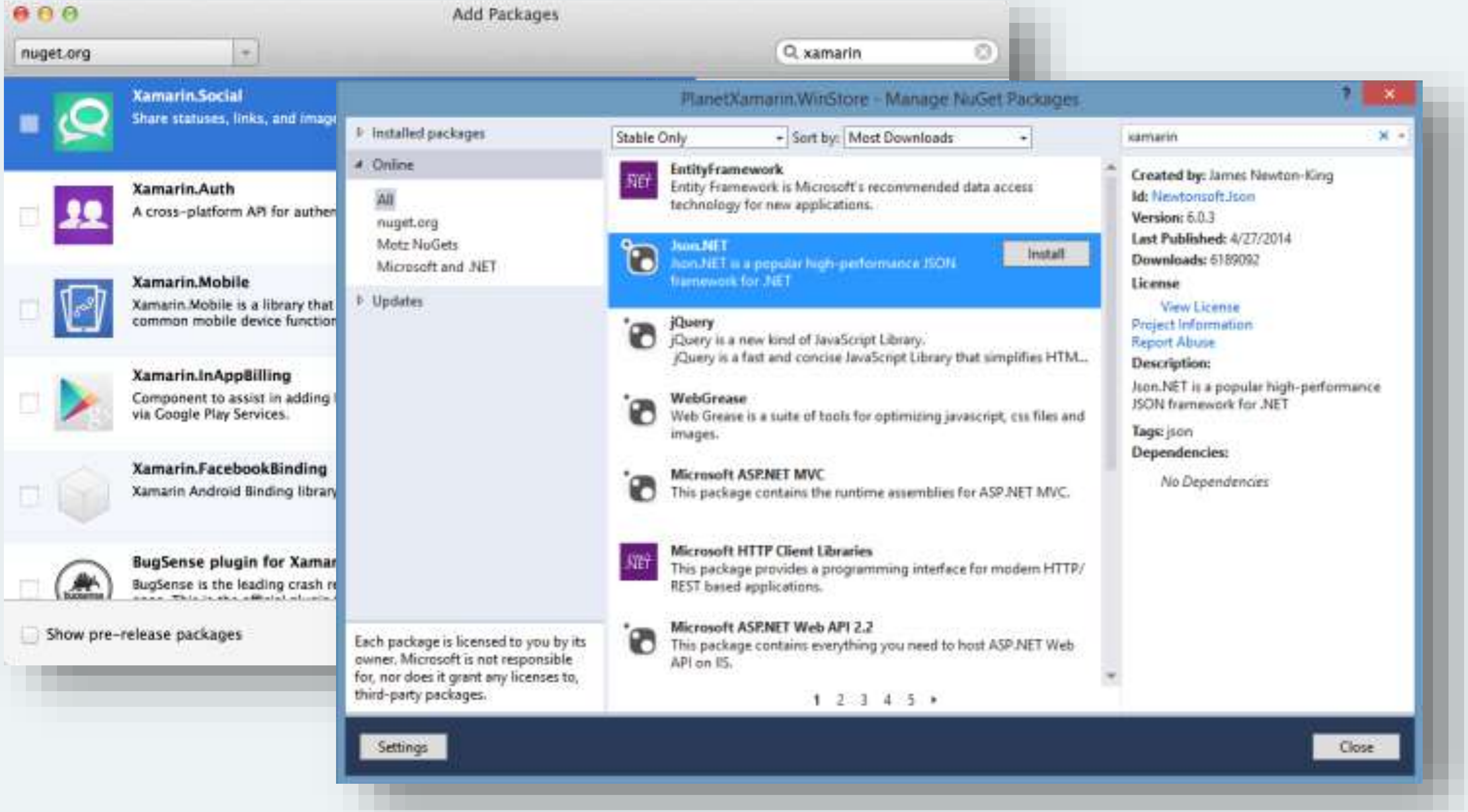
☑ Xamarin.iOS

# Shared Projects



```
tem;
tem.IO;
using System.Net;
using System.Text;
using System.Linq;
using System.Threading.Tasks;
using System.Collections.Gene

using Xamarin.SSO.Client;

namespace XamarinStore
{
    public class WebService
    {
        public static readonl
```

# NuGet



Add Packages

nuget.org

xamarin

Xamarin.Social
Share statuses, links, and image

Xamarin.Auth
A cross-platform API for authen

Xamarin.Mobile
Xamarin.Mobile is a library that
common mobile device function

Xamarin.InAppBilling
Component to assist in adding
via Google Play Services.

Xamarin.FacebookBinding
Xamarin Android Binding library

BugSense plugin for Xamar
BugSense is the leading crash re

□ Show pre-release packages

PlanetXamarin.WinStore - Manage NuGet Packages

- Installed packages
- Online
  - All
  - nuget.org
  - Motz NuGets
  - Microsoft and .NET
- Updates

Stable Only | Sort by: Most Downloads

xamarin

**EntityFramework**
Entity Framework is Microsoft's recommended data access technology for new applications.

**Json.NET** Install
Json.NET is a popular high-performance JSON framework for .NET

**jQuery**
jQuery is a new kind of JavaScript Library.
jQuery is a fast and concise JavaScript Library that simplifies HTM...

**WebGrease**
Web Grease is a suite of tools for optimizing javascript, css files and images.

**Microsoft ASP.NET MVC**
This package contains the runtime assemblies for ASP.NET MVC.

**Microsoft HTTP Client Libraries**
This package provides a programming interface for modern HTTP/REST based applications.

**Microsoft ASP.NET Web API 2.2**
This package contains everything you need to host ASP.NET Web API on IIS.

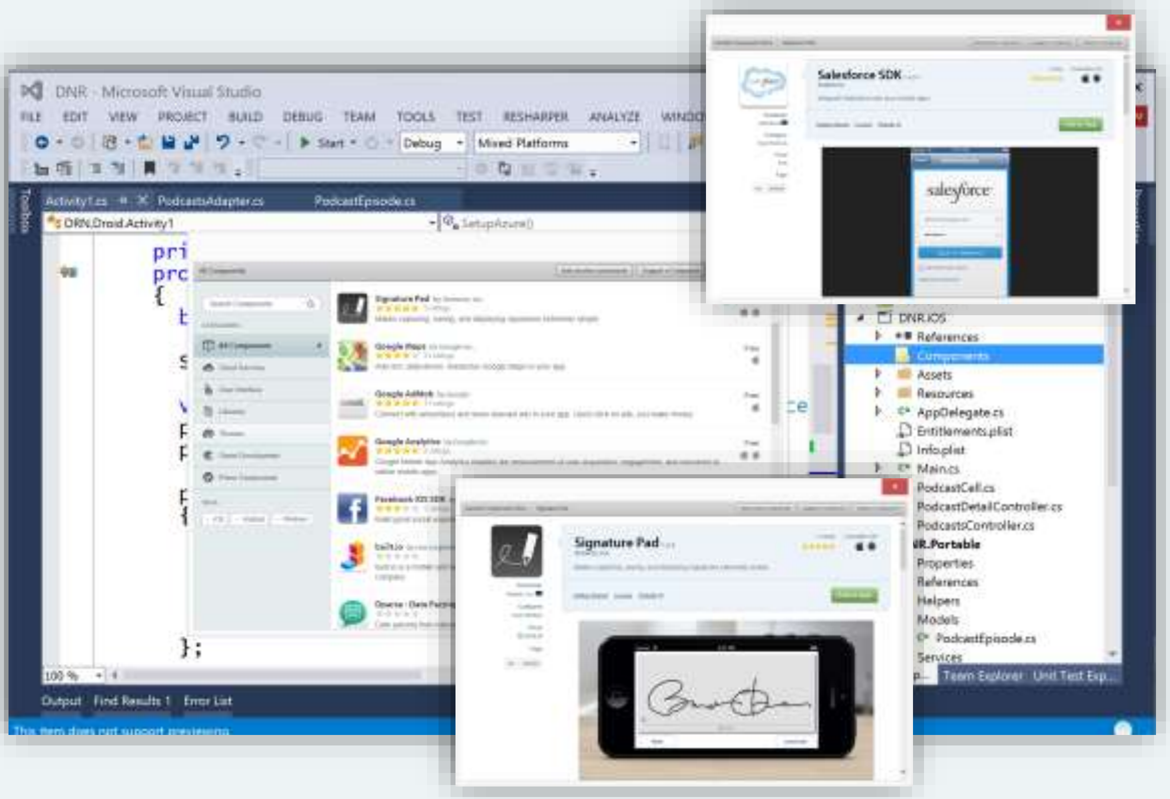
1 2 3 4 5 ▸

**Created by:** James Newton-King
**Id:** Newtonsoft.Json
**Version:** 6.0.3
**Last Published:** 4/27/2014
**Downloads:** 6189092
**License**
View License
Project Information
Report Abuse
**Description:**
Json.NET is a popular high-performance JSON framework for .NET
**Tags:** json
**Dependencies:**
*No Dependencies*

Each package is licensed to you by its owner. Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.

Settings | Close

# Xamarin Component Store



## Build Apps Faster

- Add high quality pre-built app components directly from Visual Studio and Xamarin Studio

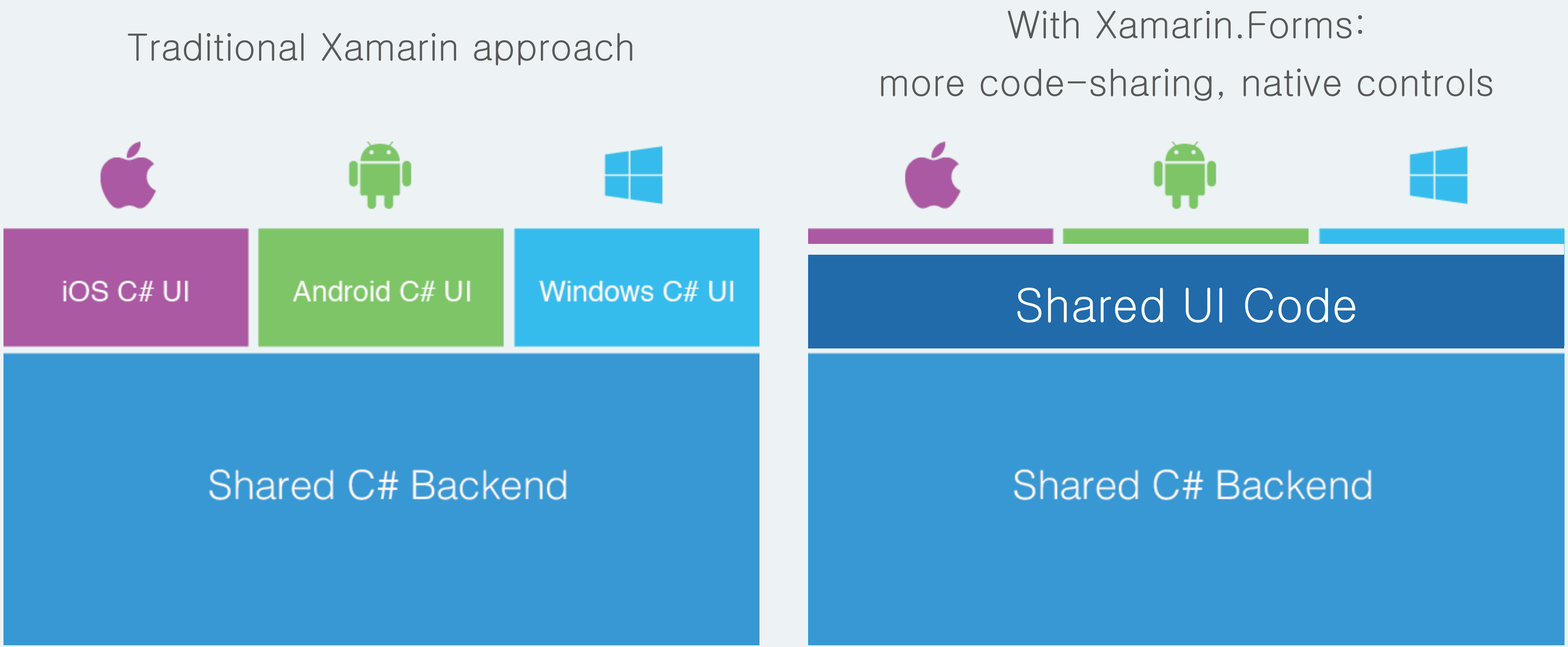
- Beautiful cross-platform UI controls, cloud services and enterprise backend integrations are just a few clicks away

# Xamarin.Forms

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

My name is Scott Hanselman. I'm a programmer, teacher, and speaker. I work out of my home office in Portland, Oregon for the Web Platform Team at Microsoft, but this blog, its content and opinions are my own. I blog about technology, culture, gadgets, diversity, code, the web, where we're going and where we've been. I'm excited about community, social equity, media, entrepreneurship and above all, the open web.

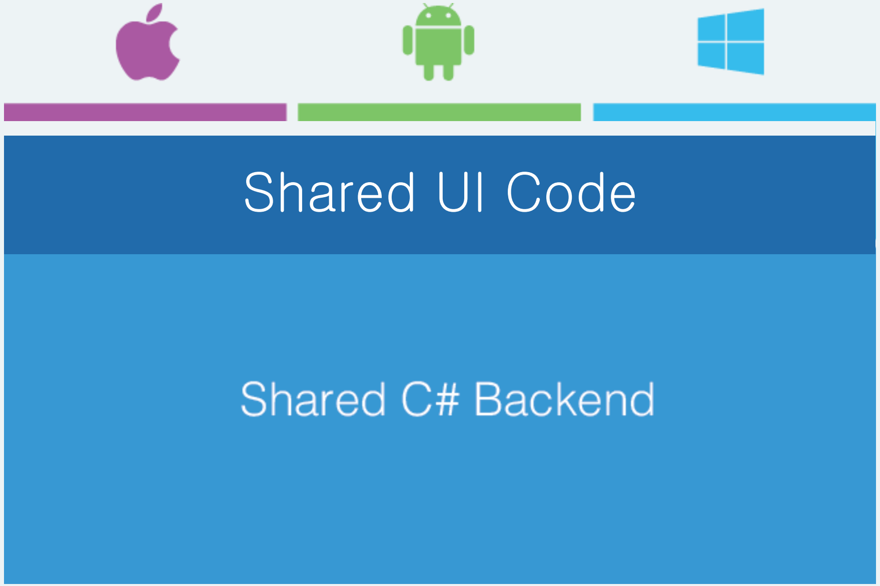
# Xamarin + Xamarin.Forms



# What's Included

- 40+ Pages, layouts, and controls

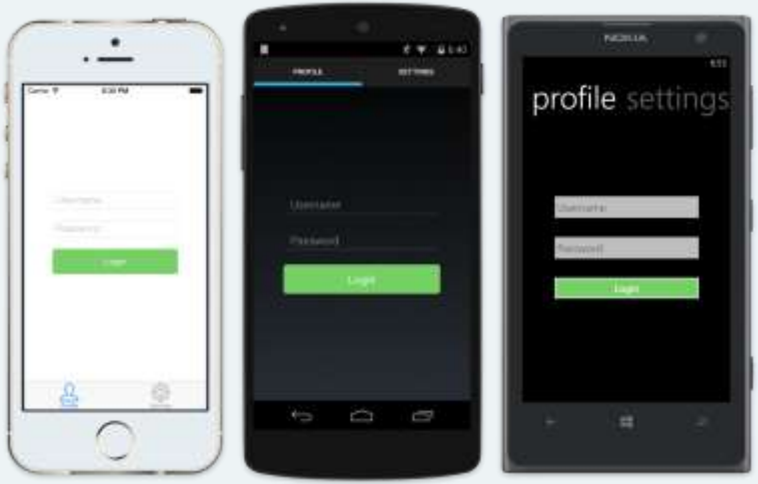
  (Build from code behind or XAML)
- Two-way data binding
- Navigation
- Animation API
- Dependency Service
- Messaging Center



Shared UI Code

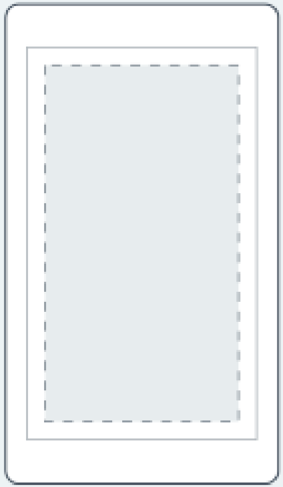
Shared C# Backend

# Native UI from shared code

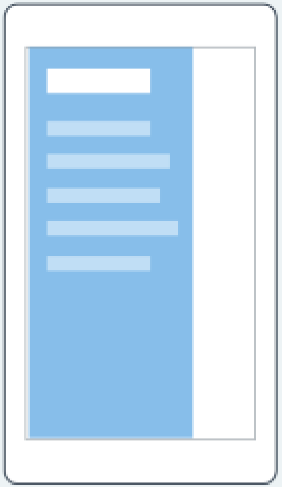


```
<?xml version="1.0" encoding="UTF-8"?>
<TabbedPage xmlns="http://xamarin.com/schemas/2014/forms"
            xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
            x:Class="MyApp.MainPage">
<TabbedPage.Children>
<ContentPage Title="Profile" Icon="Profile.png">
   <StackLayout Spacing="20" 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>
```

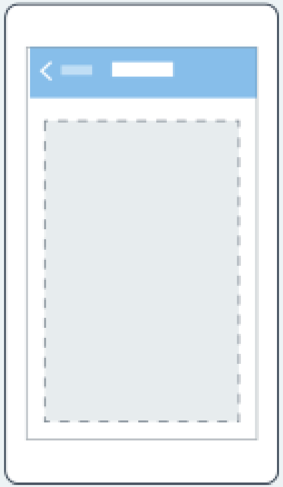
# Pages



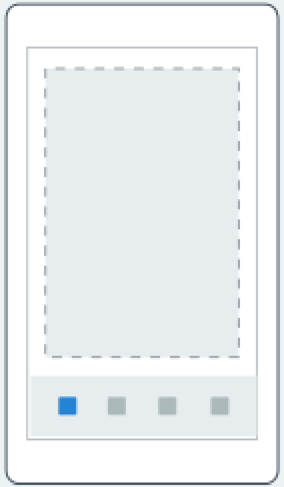
Content



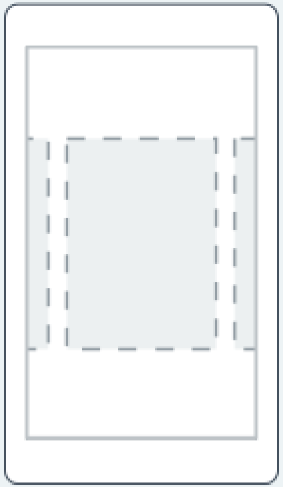
MasterDetail



Navigation

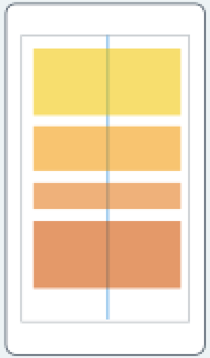


Tabbed

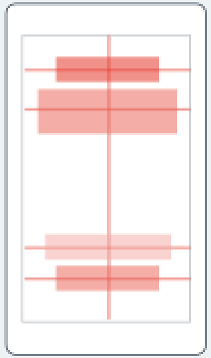


Carousel

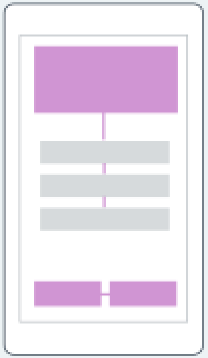
# Layouts



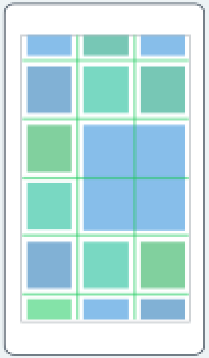
Stack



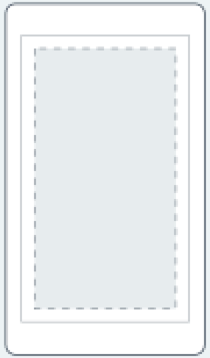
Absolute



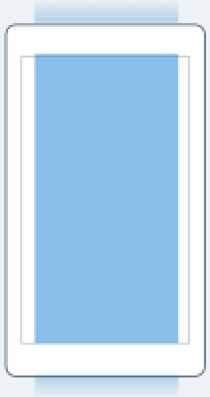
Relative



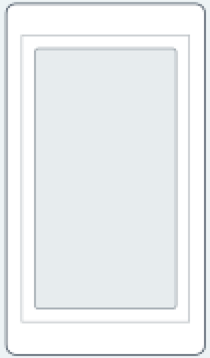
Grid



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

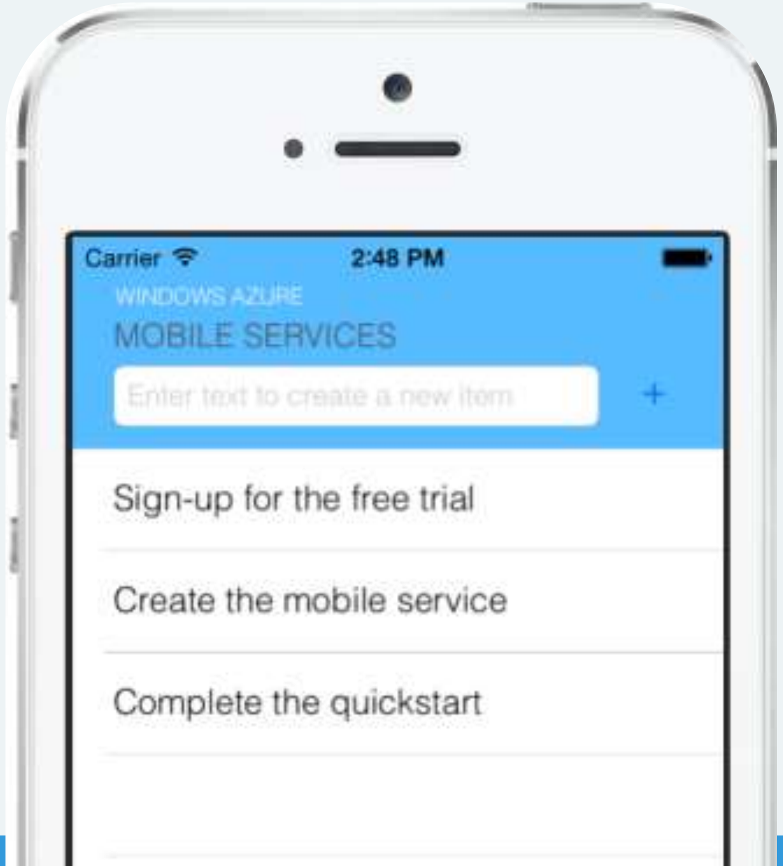
# Control Comparison

| 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 |

# Microsoft Azure



Sign-up for the free trial

Create the mobile service

Complete the quickstart

- Build amazing mobile apps in C# with Xamarin
- Build amazing backends using Azure
- Integrate Xamarin & Azure
  - Azure Mobile Services
  - Azure Messaging
  - Azure Storage

감사합니다

Thank you~!

# Xamarin Native HOL

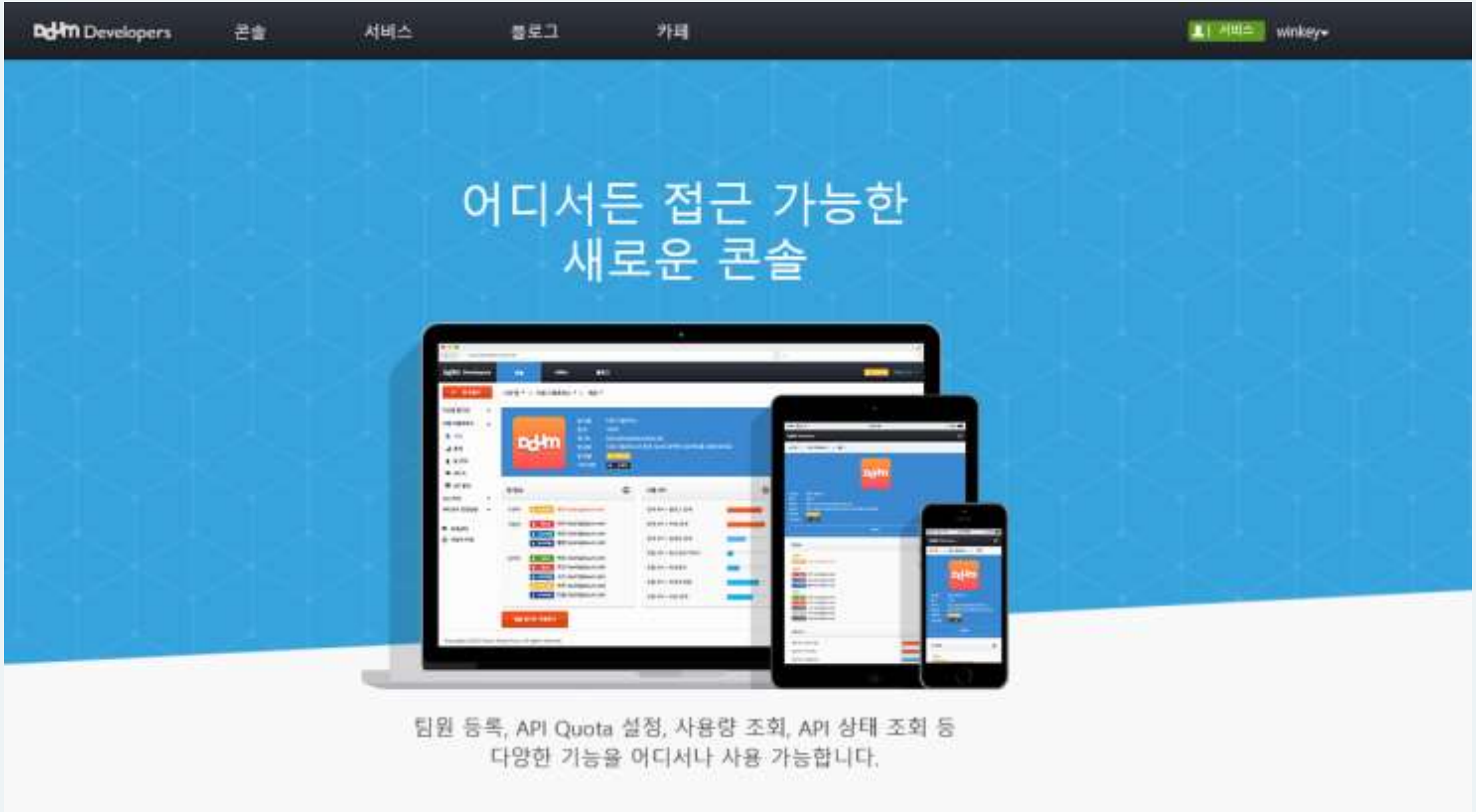
# Daum Book API

[http://1drv.ms/1oLrBpN](http://1drv.ms/1oLrBpN)

# Daum 검색 API

http://developers.daum.net/

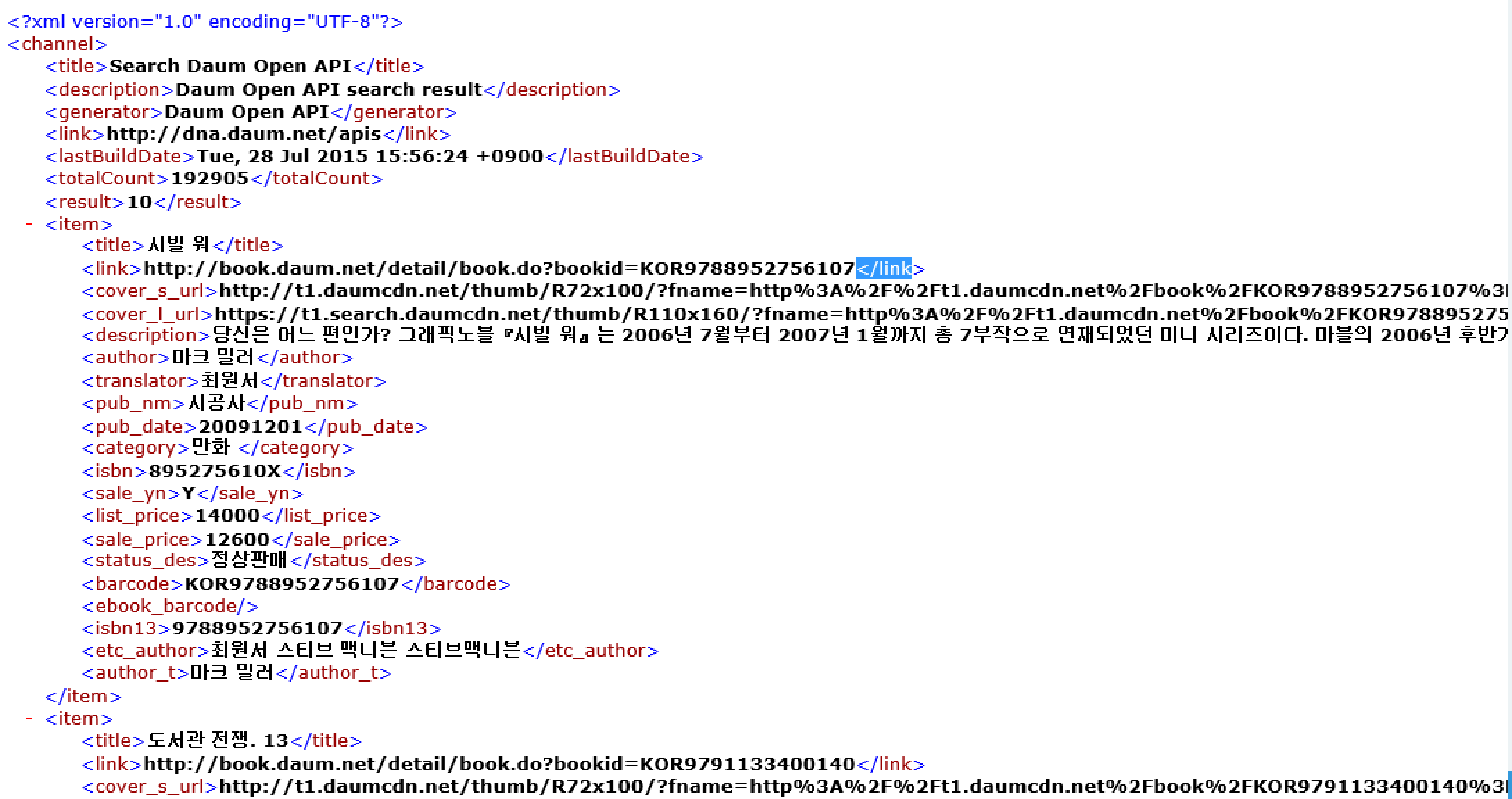
# Daum Open APIs



어디서든 접근 가능한
새로운 콘솔

팀원 등록, API Quota 설정, 사용량 조회, API 상태 조회 등
다양한 기능을 어디서나 사용 가능합니다.

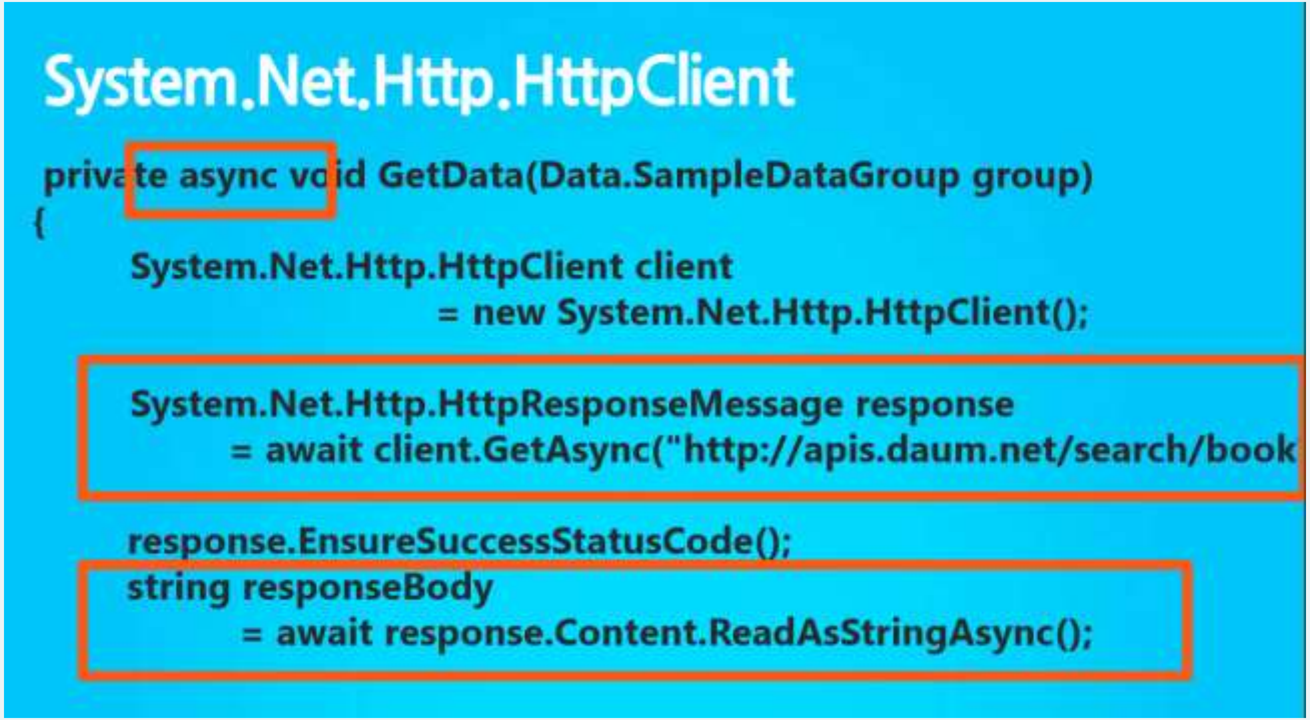
# Daum Open APIs



```
<?xml version="1.0" encoding="UTF-8"?>
<channel>
    <title>Search Daum Open API</title>
    <description>Daum Open API search result</description>
    <generator>Daum Open API</generator>
    <link>http://dna.daum.net/apis</link>
    <lastBuildDate>Tue, 28 Jul 2015 15:56:24 +0900</lastBuildDate>
    <totalCount>192905</totalCount>
    <result>10</result>
  - <item>
        <title>시빌 워</title>
        <link>http://book.daum.net/detail/book.do?bookid=KOR9788952756107</link>
        <cover_s_url>http://t1.daumcdn.net/thumb/R72x100/?fname=http%3A%2F%2Ft1.daumcdn.net%2Fbook%2FKOR9788952756107%3
        <cover_l_url>https://t1.search.daumcdn.net/thumb/R110x160/?fname=http%3A%2F%2Ft1.daumcdn.net%2Fbook%2FKOR978895275
        <description>당신은 어느 편인가? 그래픽노블 『시빌 워』는 2006년 7월부터 2007년 1월까지 총 7부작으로 연재되었던 미니 시리즈이다. 마블의 2006년 후반
        <author>마크 밀러</author>
        <translator>최원서</translator>
        <pub_nm>시공사</pub_nm>
        <pub_date>20091201</pub_date>
        <category>만화 </category>
        <isbn>895275610X</isbn>
        <sale_yn>Y</sale_yn>
        <list_price>14000</list_price>
        <sale_price>12600</sale_price>
        <status_des>정상판매</status_des>
        <barcode>KOR9788952756107</barcode>
        <ebook_barcode/>
        <isbn13>9788952756107</isbn13>
        <etc_author>최원서 스티브 맥니븐 스티브맥니븐</etc_author>
        <author_t>마크 밀러</author_t>
    </item>
  - <item>
        <title>도서관 전쟁. 13</title>
        <link>http://book.daum.net/detail/book.do?bookid=KOR9791133400140</link>
        <cover_s_url>http://t1.daumcdn.net/thumb/R72x100/?fname=http%3A%2F%2Ft1.daumcdn.net%2Fbook%2FKOR9791133400140%3
```

# System.Net.Http.HttpClient

- Namespace: System.Net.Http.HttpClient
  Assembly: System.Net.Http (in System.Net.Http.dll)



## System.Net.Http.HttpClient

```
private async void GetData(Data.SampleDataGroup group)
{
    System.Net.Http.HttpClient client
                    = new System.Net.Http.HttpClient();

    System.Net.Http.HttpResponseMessage response
            = await client.GetAsync("http://apis.daum.net/search/book

    response.EnsureSuccessStatusCode();
    string responseBody
            = await response.Content.ReadAsStringAsync();
```

# System.Net.Http.HttpClient

## DEMO

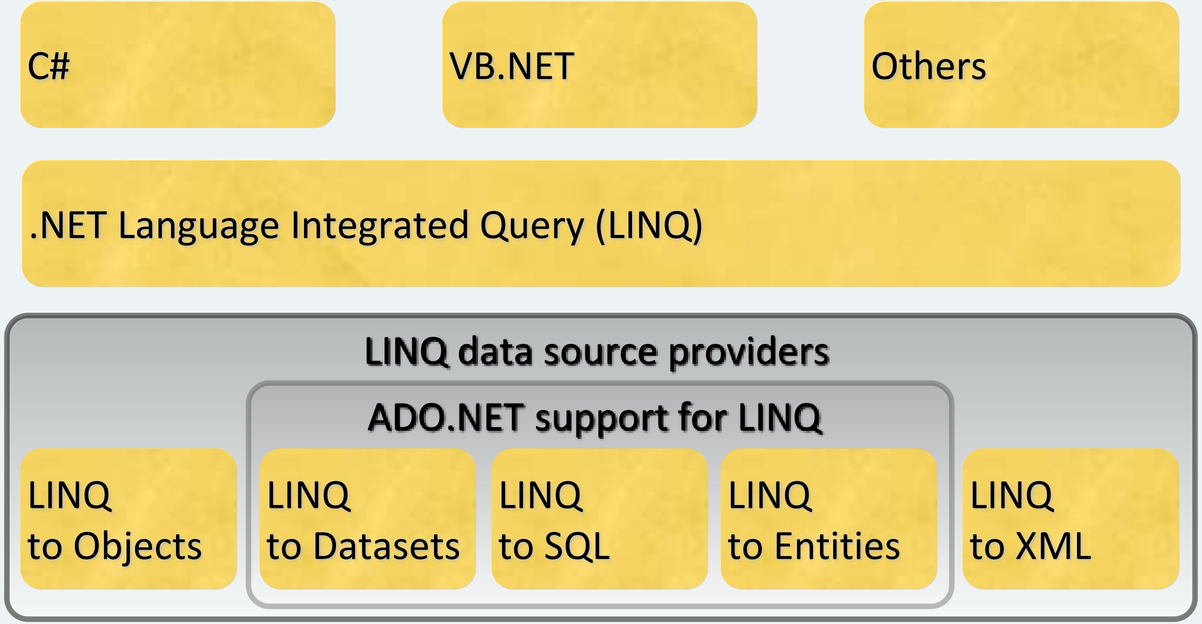
Windows

# Entity Class

```
class BookItem
{
 public Int64 Id { get; set; }
 public string Title { get; set; }
 public string Category { get; set; }
 public string ImageUrl { get; set; }
 public string Description { get; set; }
}
```

# Linq

# Architecture

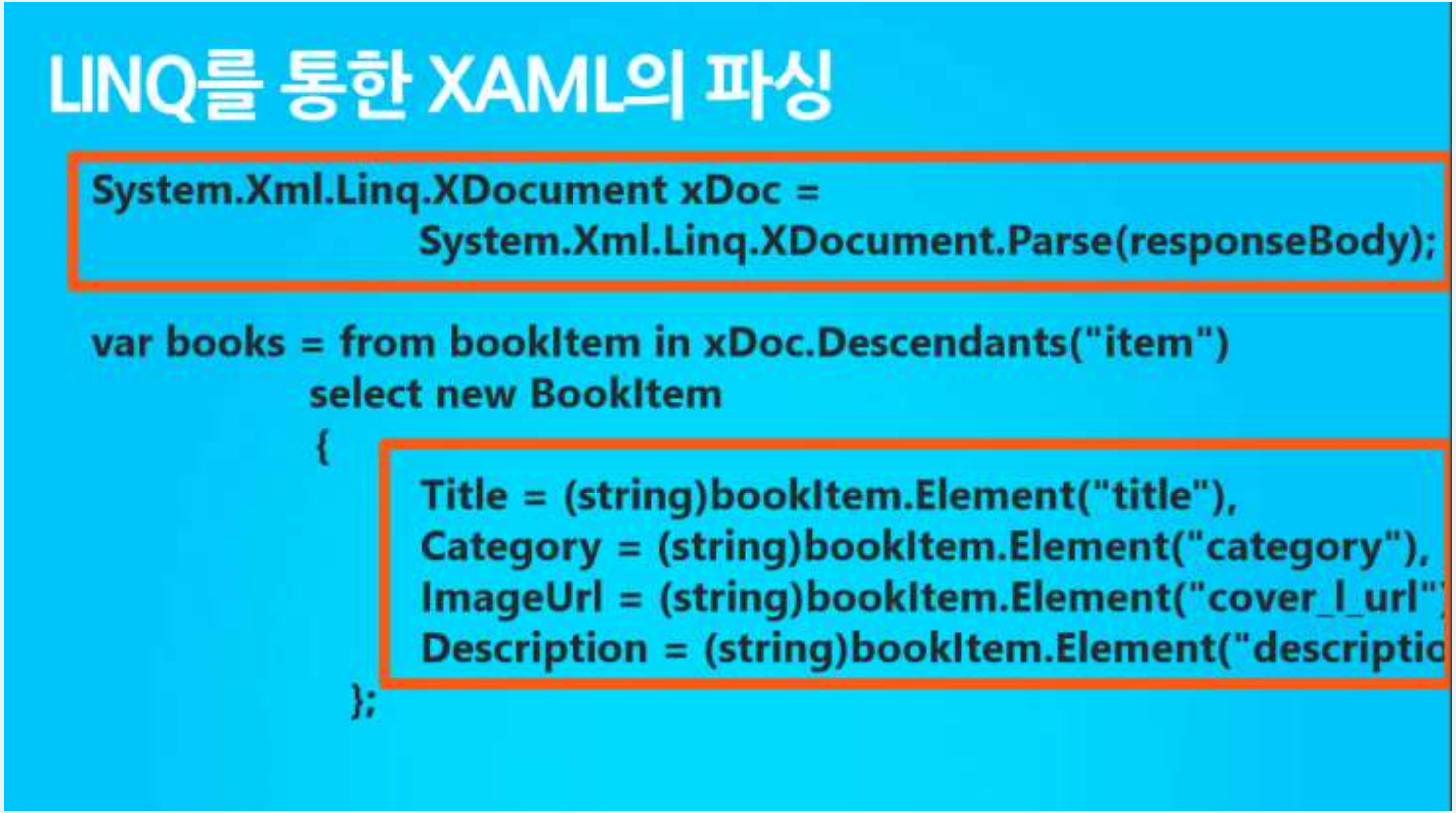


# LINQ to Objects

- C#

```
int[] nums = new int[] {0,4,2,6,3,8,3,1};
double average = nums.Take(6).Average();
var above = from n in nums
            where n > average
            select n;
```

# System.Net.Http.HttpClient



## LINQ를 통한 XAML의 파싱

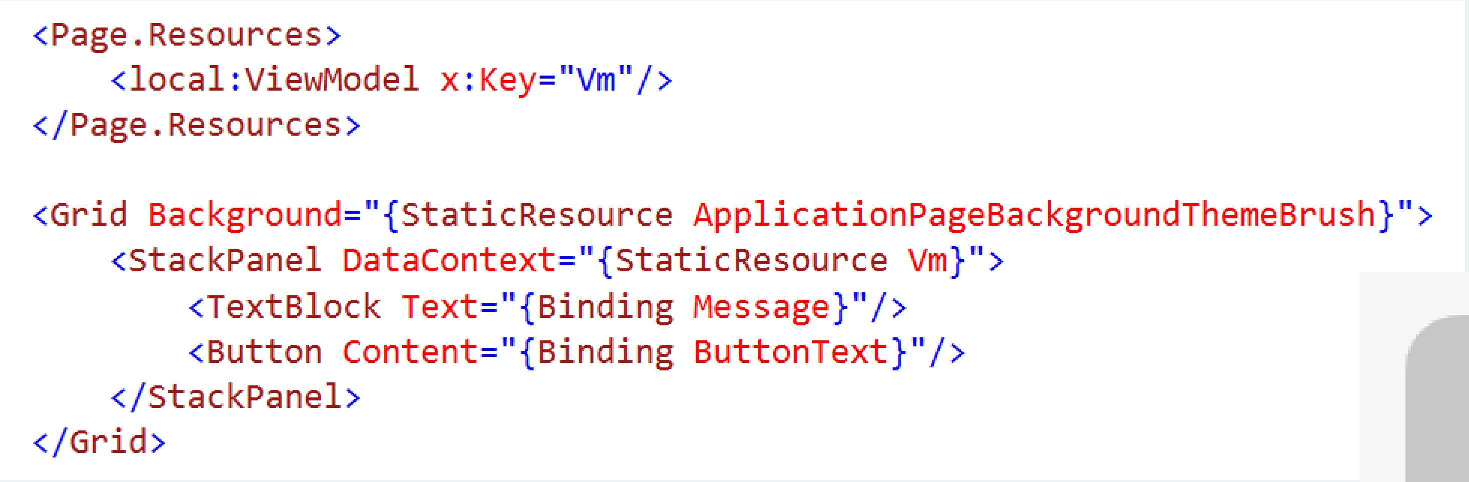
```
System.Xml.Linq.XDocument xDoc =
                    System.Xml.Linq.XDocument.Parse(responseBody);

var books = from bookItem in xDoc.Descendants("item")
              select new BookItem
              {
                   Title = (string)bookItem.Element("title"),
                   Category = (string)bookItem.Element("category"),
                   ImageUrl = (string)bookItem.Element("cover_l_url"
                   Description = (string)bookItem.Element("descriptio
               };
```

# Data binding

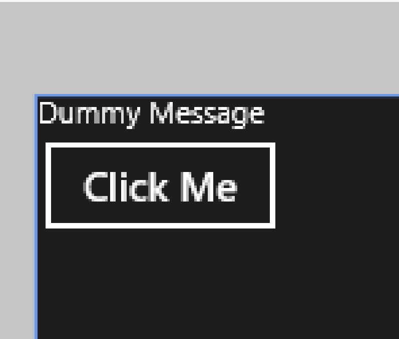
# Data binding basics

# Dynamic Data

- Use data binding to connect to a data source



```
<Page.Resources>
    <local:ViewModel x:Key="Vm"/>
</Page.Resources>

<Grid Background="{StaticResource ApplicationPageBackgroundThemeBrush}">
    <StackPanel DataContext="{StaticResource Vm}">
        <TextBlock Text="{Binding Message}"/>
        <Button Content="{Binding ButtonText}"/>
    </StackPanel>
</Grid>
```



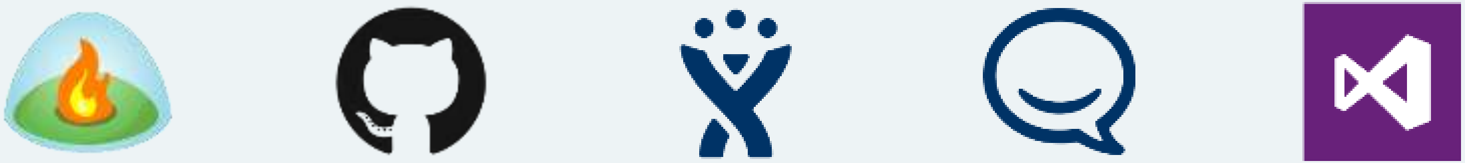
# Data Binding

DEMO

Windows

# Integrate

Simple integrations into popular services for mission critical notifications.



감사합니다

Thank you~!