

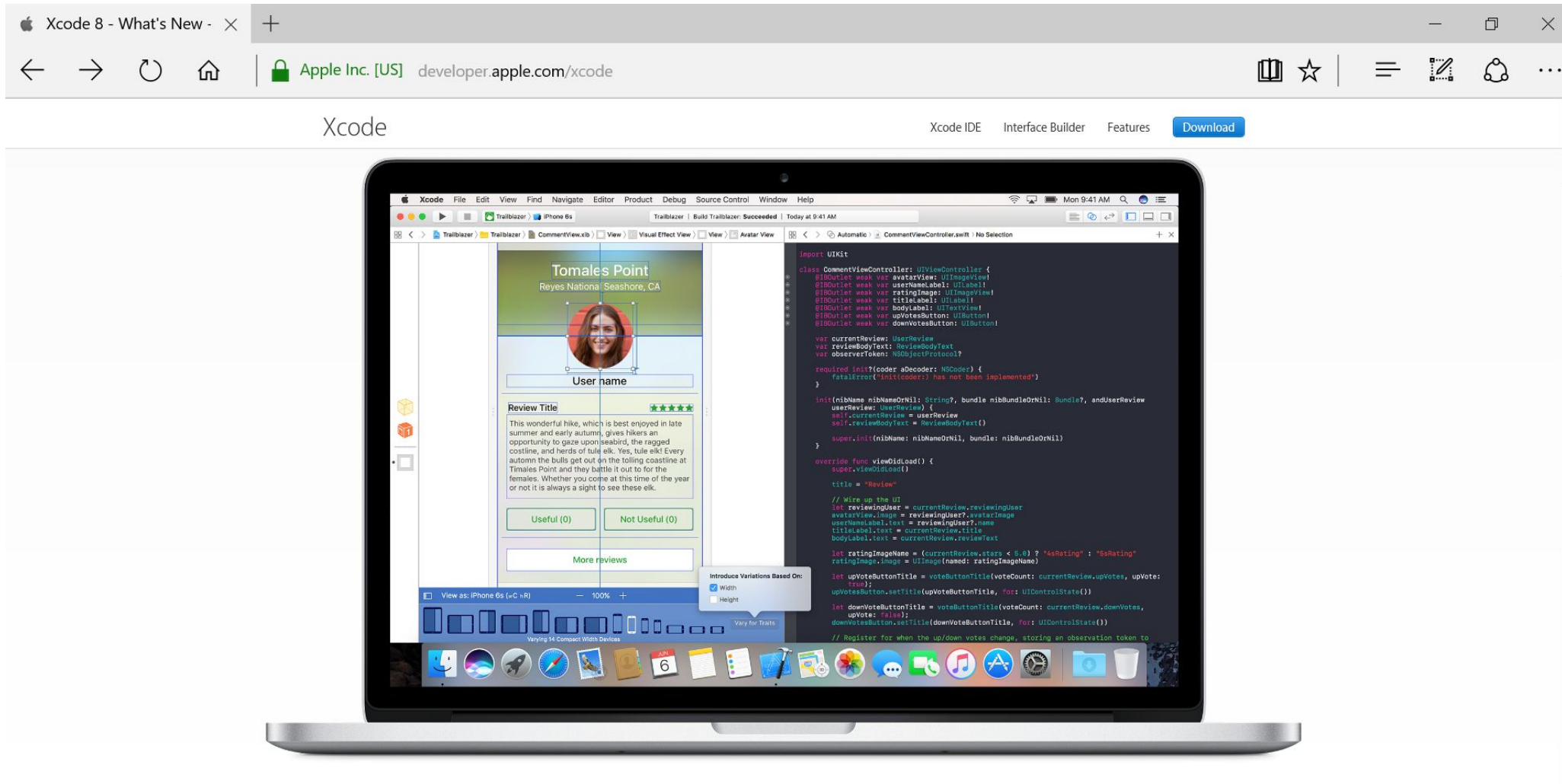
# Хатагин: разработка кроссплатформенных мобильных приложений

Алексей Кизин, Синерджи Софт

[www.synsol.ru](http://www.synsol.ru)

[www.facebook.com/akizin](https://www.facebook.com/akizin)

# Инструменты разработки Apple <https://developer.apple.com/xcode>



**OS:**

- Mac OS X

**IDE:**

- Xcode

**ЯЗЫКИ:**

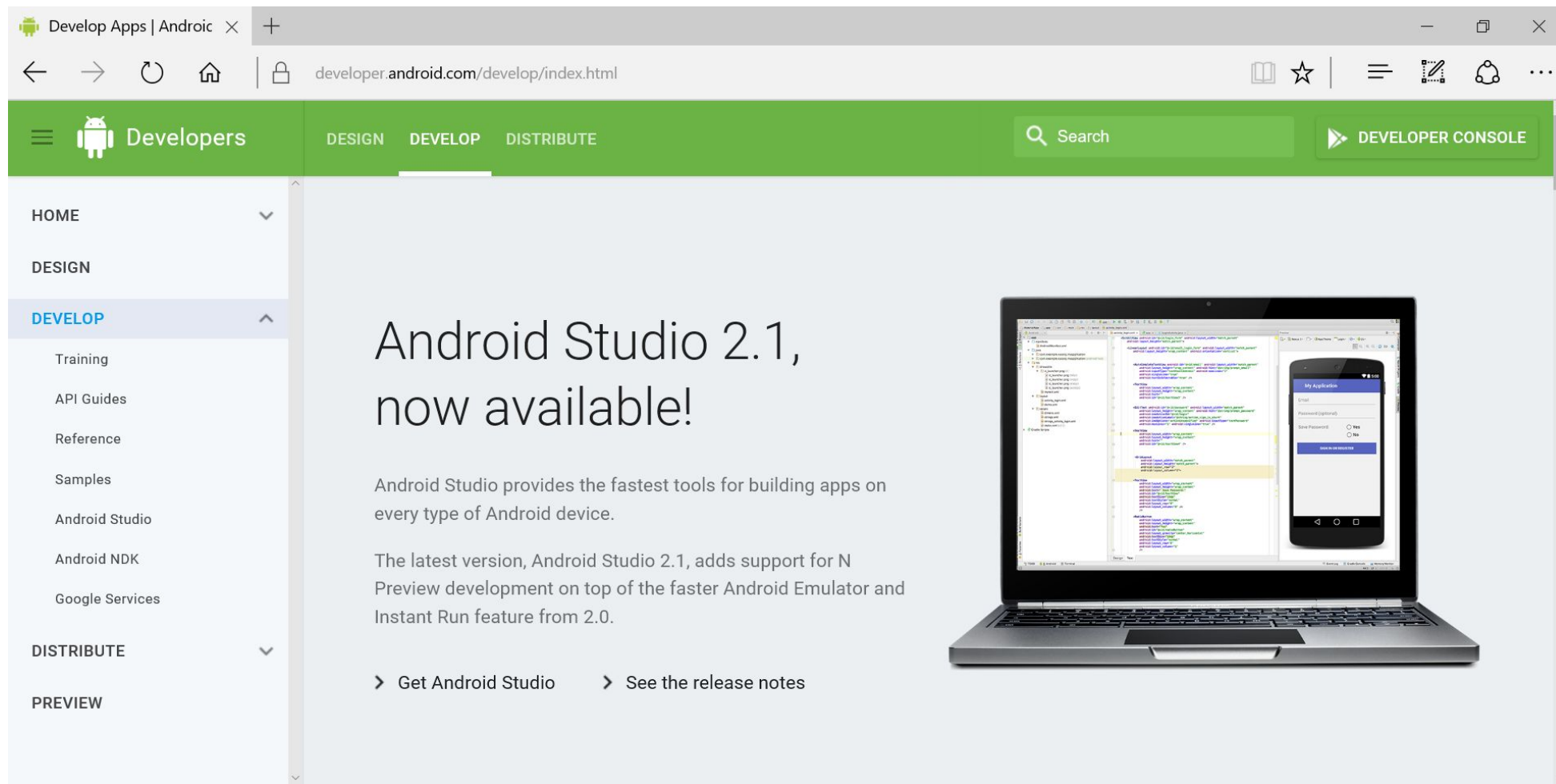
- Objective-C
- Swift

**Платформа:**

- iOS

# Инструменты разработки Google

<https://developer.android.com/studio>



The screenshot shows the developer.android.com website. The main heading reads "Android Studio 2.1, now available!". Below the heading, it states: "Android Studio provides the fastest tools for building apps on every type of Android device." and "The latest version, Android Studio 2.1, adds support for N Preview development on top of the faster Android Emulator and Instant Run feature from 2.0." There are two links: "Get Android Studio" and "See the release notes". To the right, a laptop displays the Android Studio IDE interface with code and a virtual device emulator.

## OS:

- Windows
- Mac OS X
- Linux

## IDE:

- Android Studio

## Язык:

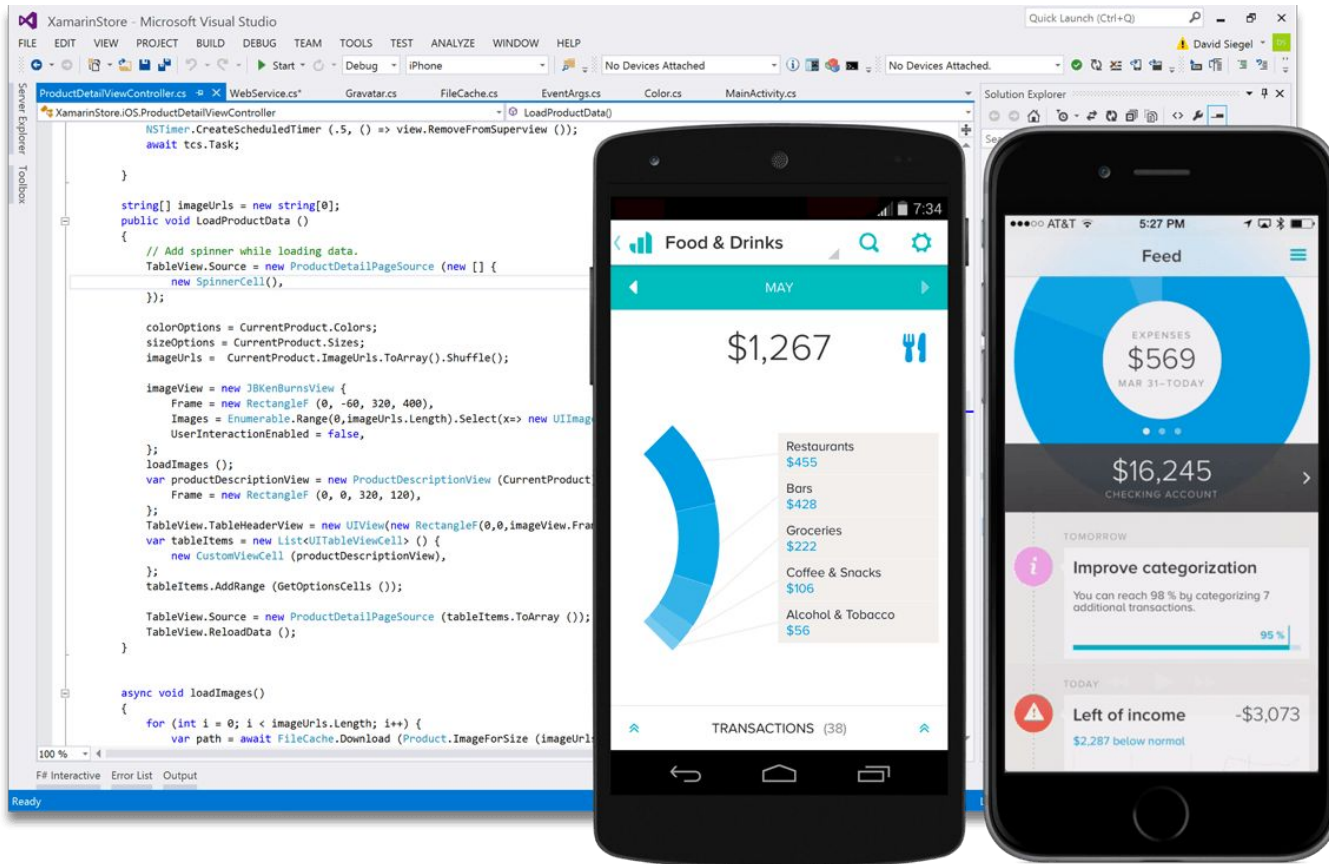
- Java

## Платформа:

- Android

# Платформа Xamarin

- **Нативные** приложения для iOS, Android, & Windows на **C#**



# Фишки C# : var

## Меньше кода, порядок с типами

Локальная переменная с неявным типом имеет строгую типизацию, как если бы тип был задан явно, только тип определяет компилятор.

```
public Dictionary<string, UIImage> GetImages(string directory) {
    var images = new Dictionary<string, UIImage>();
    var paths = Directory.EnumerateFiles(directory, "*.png");
    foreach (var path in paths) {
        images[path] = UIImage.FromFile(path);
    }
    return images;
}

public async Task<List<FeedItem>> GetFeedItems(DateTime date) {
    var feed = "http://planet.xamarin.com/feed/";
    var response = await httpClient.GetStringAsync(feed);
    var items = await ParseFeedAsync(response);
    return items.Where(item => item.Published.Date == date).ToList();
}
```

# Фишки C# : async + await

## Async на уровне языка

Асинхронное программирование позволяет обеспечить отзывчивость приложений. В Objective-C, Swift и Java асинхронные операции требуют написания вспомогательного кода. В C# поддержка асинхронных операций реализована на уровне языка.

```
public Dictionary<string, UIImage> GetImages(string directory) {  
    var images = new Dictionary<string, UIImage>();  
    var paths = Directory.EnumerateFiles(directory, "*.png");  
    foreach (var path in paths) {  
        images[path] = UIImage.FromFile(path);  
    }  
    return images;  
}  
  
public async Task<List<FeedItem>> GetFeedItems(DateTime date) {  
    var feed = "http://planet.xamarin.com/feed/";  
    var response = await httpClient.GetStringAsync(feed);  
    var items = await ParseFeedAsync(response);  
    return items.Where(item => item.Published.Date == date).ToList();  
}
```

# Фишки C# : Генерики

## Более строгие типы, более умные инструменты

Генерики гарантируют, что коллекции и прочие составные типы используются безопасно, без необходимости приведения типов или создания избыточных комментариев. Кроме всего прочего это позволяет делать более умное автозавершение кода

```
public Dictionary<string, UIImage> GetImages(string directory) {
    var images = new Dictionary<string, UIImage>();
    var paths = Directory.EnumerateFiles(directory, "*.png");
    foreach (var path in paths) {
        images[path] = UIImage.FromFile(path);
    }
    return images;
}

public async Task<List<FeedItem>> GetFeedItems(DateTime date) {
    var feed = "http://planet.xamarin.com/feed/";
    var response = await httpClient.GetStringAsync(feed);
    var items = await ParseFeedAsync(response);
    return items.Where(item => item.Published.Date == date).ToList();
}
```

# Фишки C# : лямбда-выражения

## Внимание на логику, не на процесс

В Java для Android появились недавно, в Objective-C слишком громоздкие. В C# работа с лямбда-операторами «проще не бывает»

```
public Dictionary<string, UIImage> GetImages(string directory) {
    var images = new Dictionary<string, UIImage>();
    var paths = Directory.EnumerateFiles(directory, "*.png");
    foreach (var path in paths) {
        images[path] = UIImage.FromFile(path);
    }
    return images;
}

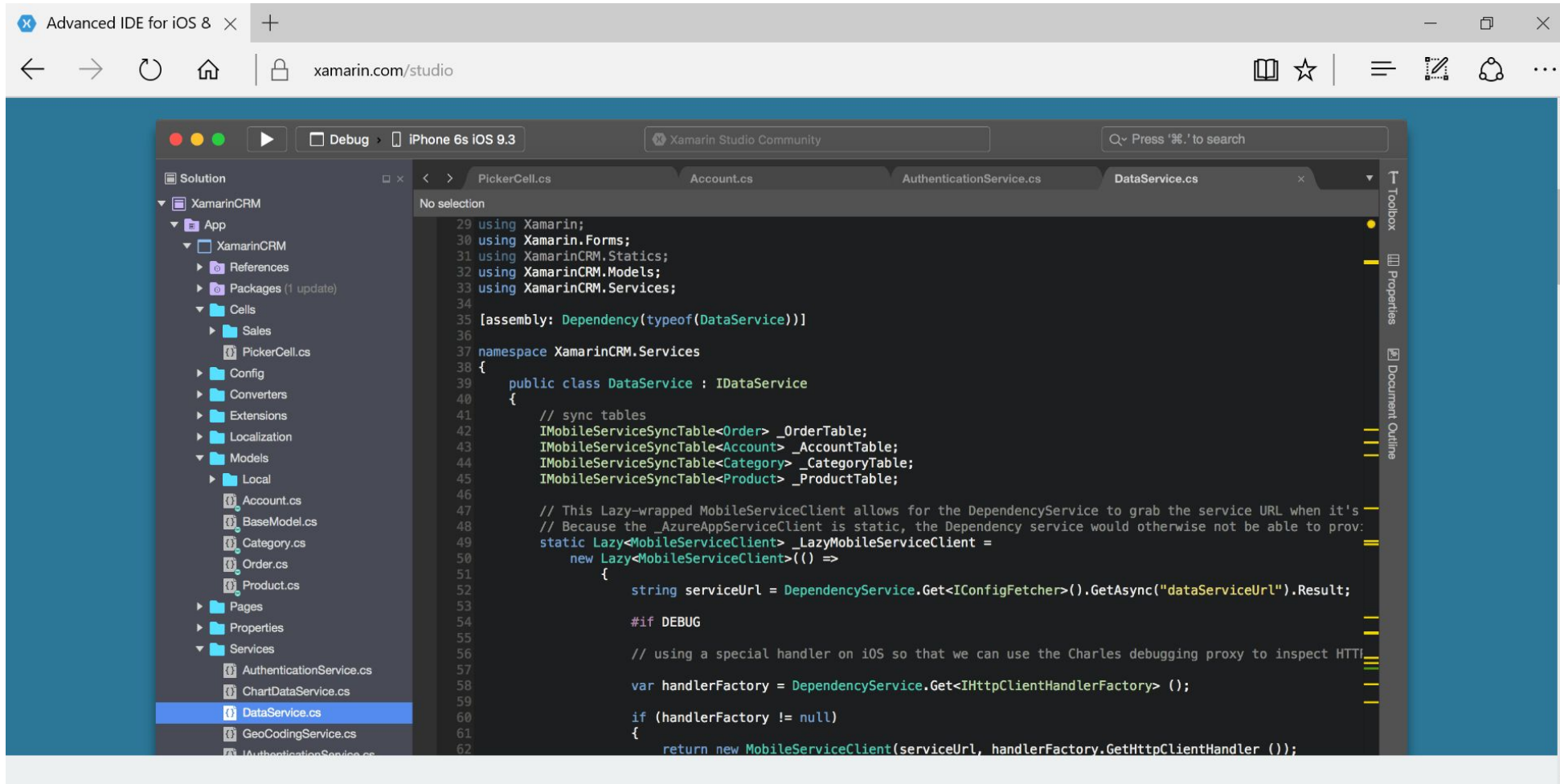
public async Task<List<FeedItem>> GetFeedItems(DateTime date) {
    var feed = "http://planet.xamarin.com/feed/";
    var response = await httpClient.GetStringAsync(feed);
    var items = await ParseFeedAsync(response);
    return items.Where(item => item.Published.Date == date).ToList();
}
```



# Немного истории

- Июнь 2000 – представлен .NET Framework
- Июль 2001 – запуск проекта Mono (Ximian, Miguel de Icaza)
- Август 2003 – Novel покупает Ximian
- Апрель 2011 – Attachmate покупает Novel, будущее Mono под вопросом
- Июль 2011 – права на Mono, Mono Touch и Mono for Android переданы Xamarin
- Февраль 2013 – Выпуск Xamarin 2.0 (ранее Monodevelop)
- Февраль 2016 – Microsoft покупает Xamarin

# Инструменты разработки Xamarin <https://www.xamarin.com/studio>



## OS:

- Windows
- Mac OS X

## IDE:

- Xamarin Studio

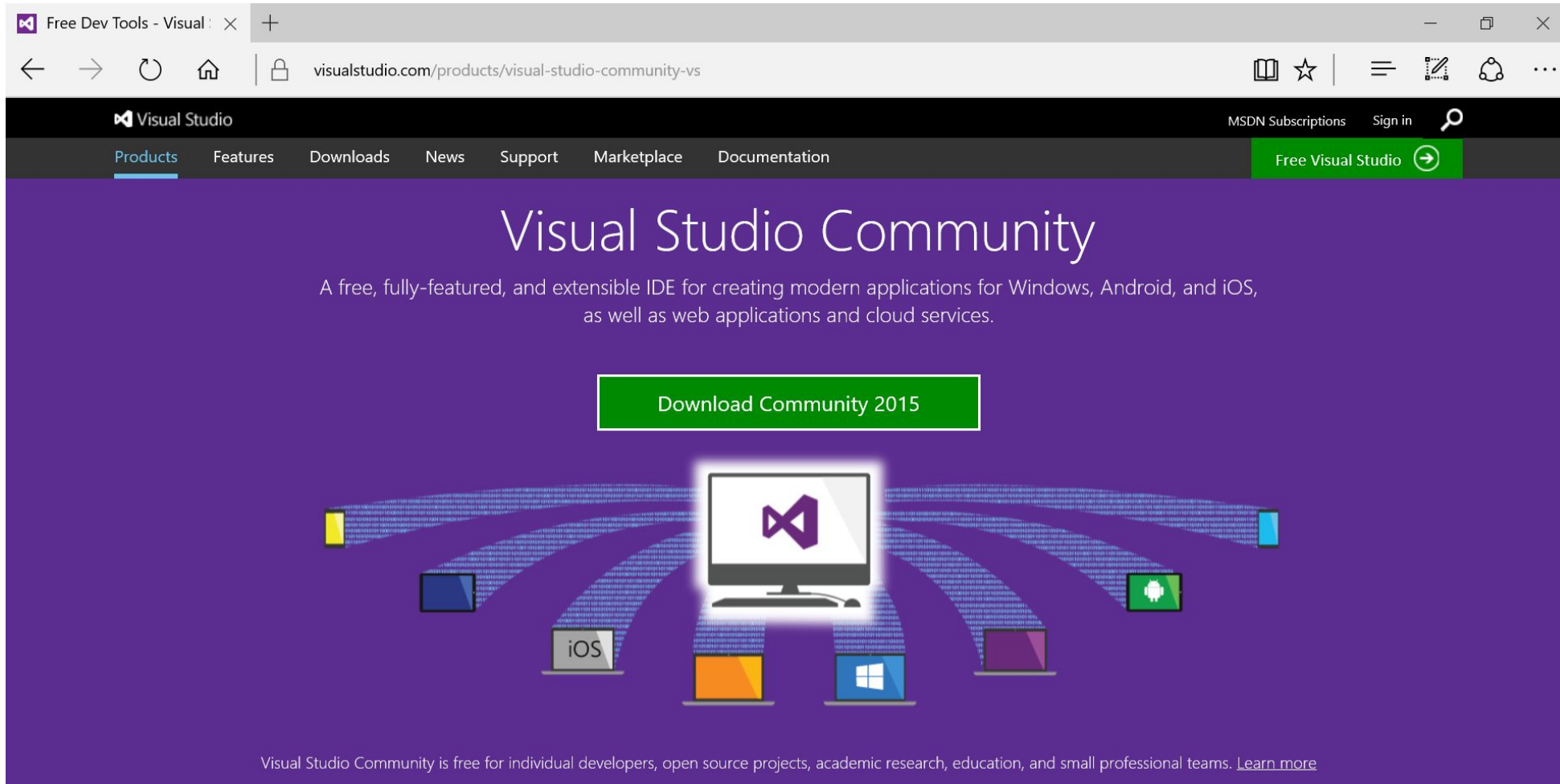
## Язык:

- C#

## Платформы:

- iOS
- Android
- Windows

# Инструменты разработки Microsoft <https://www.xamarin.com/studio>



The image shows a screenshot of the Visual Studio Community website. The browser address bar displays 'visualstudio.com/products/visual-studio-community-vs'. The website header includes the Visual Studio logo, navigation links for 'Products', 'Features', 'Downloads', 'News', 'Support', 'Marketplace', and 'Documentation', and a 'Free Visual Studio' button. The main content area features the title 'Visual Studio Community' and a description: 'A free, fully-featured, and extensible IDE for creating modern applications for Windows, Android, and iOS, as well as web applications and cloud services.' A prominent green button labeled 'Download Community 2015' is centered below the text. Below the button is a graphic showing a central computer monitor with the Visual Studio logo, connected by blue lines to various mobile device icons representing iOS, Android, and Windows. At the bottom of the page, a small text line states: 'Visual Studio Community is free for individual developers, open source projects, academic research, education, and small professional teams. [Learn more](#)'

## OS:

- Windows

## IDE:

- Visual Studio

## Язык:

- C#

## Платформы:

- iOS
- Android
- Windows

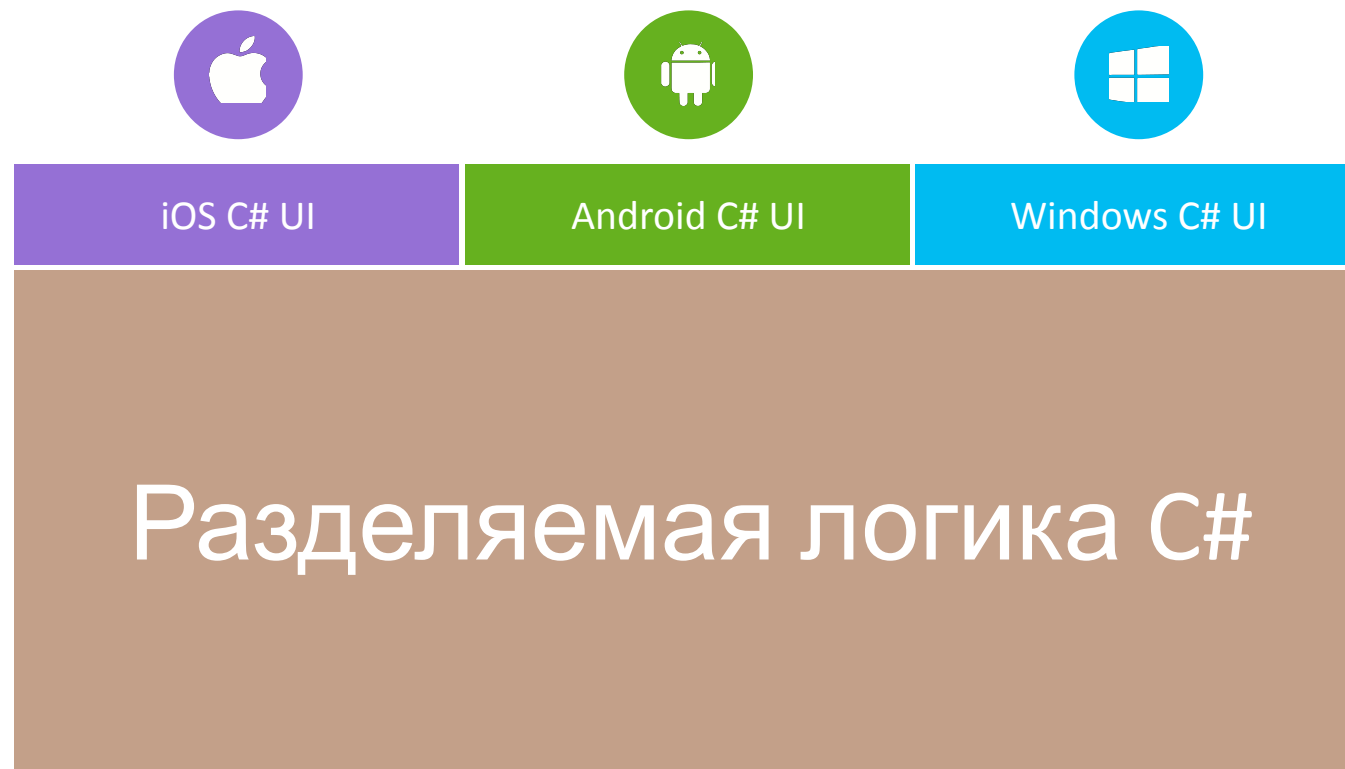
# Среды разработки Xamarin

## Development Environments

This table shows which platforms can be built with different development tool & operating system combinations:

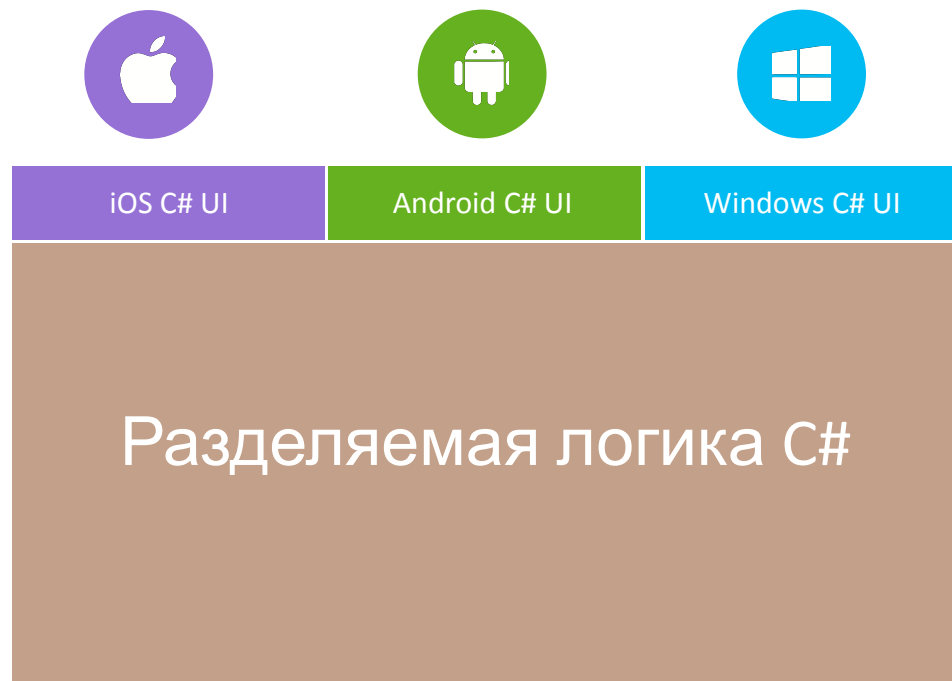
	MAC OS X	WINDOWS	
Development Environment	XAMARIN STUDIO	VISUAL STUDIO	XAMARIN STUDIO
Xamarin.iOS	Yes	Yes (with Mac computer)	No
Xamarin.Android	Yes	Yes	Yes
Xamarin.Forms	iOS & Android only	Android, Windows Phone, Windows (iOS with Mac computer)	Android only
Xamarin.Mac	Yes	No	No

# Подход Xamarin

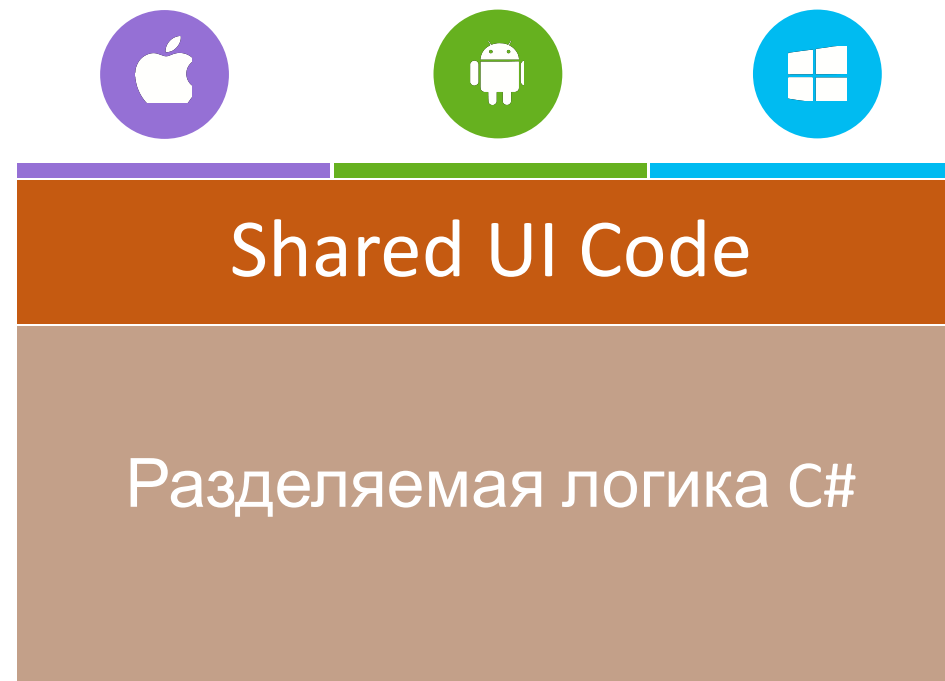


Общая база кода C# • 100% доступ к нативным API • Высокая производительность

# Xamarin + Xamarin.Forms



Традиционный подход



Xamarin.Forms:  
Больше разделяемого кода,  
все нативно

# Windows API

Microsoft.Phone	Microsoft.Networking	Windows.Storage	Windows.Foundation	Microsoft.Devices
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

# iOS – 100% покрытие API

UIKit	UIKit	iBeacon	CoreGraphics	CoreMotion
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

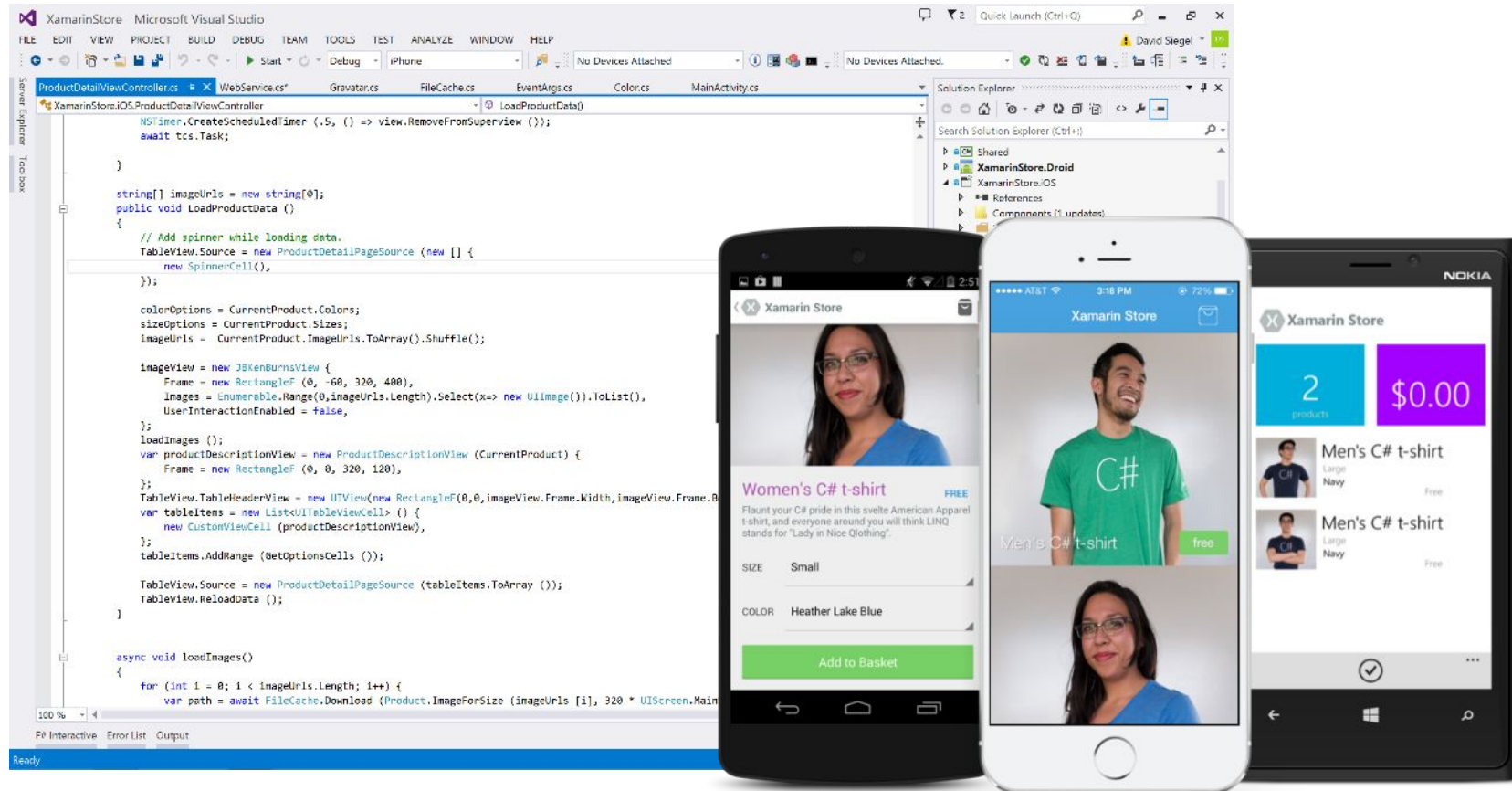


# Android – 100% покрытие API

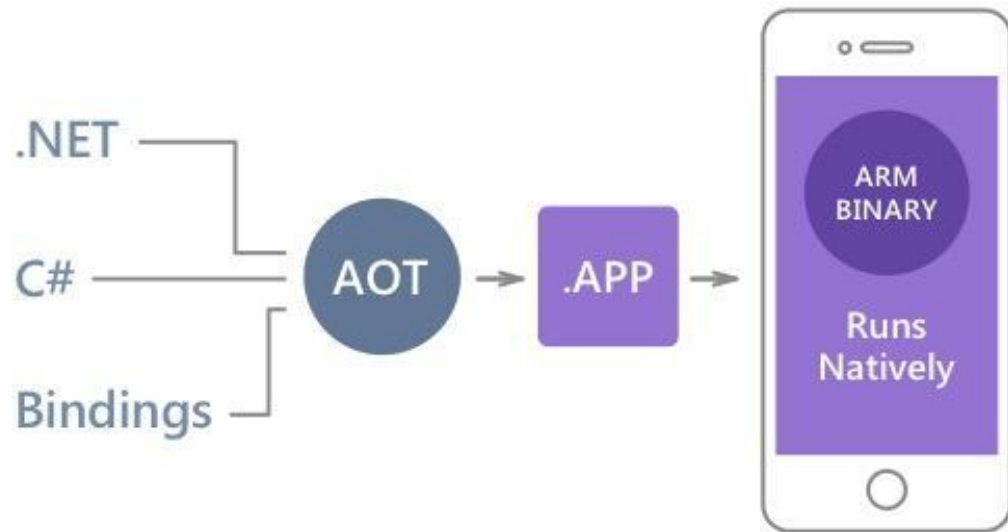
Text-to-speech	ActionBar	Printing Framework	RenderScript	NFC
System.Net	System	System.IO	System.Linq	System.Xml
System.Data	System.Windows	System.Numerics	System.Core	System.ServiceModel

C#

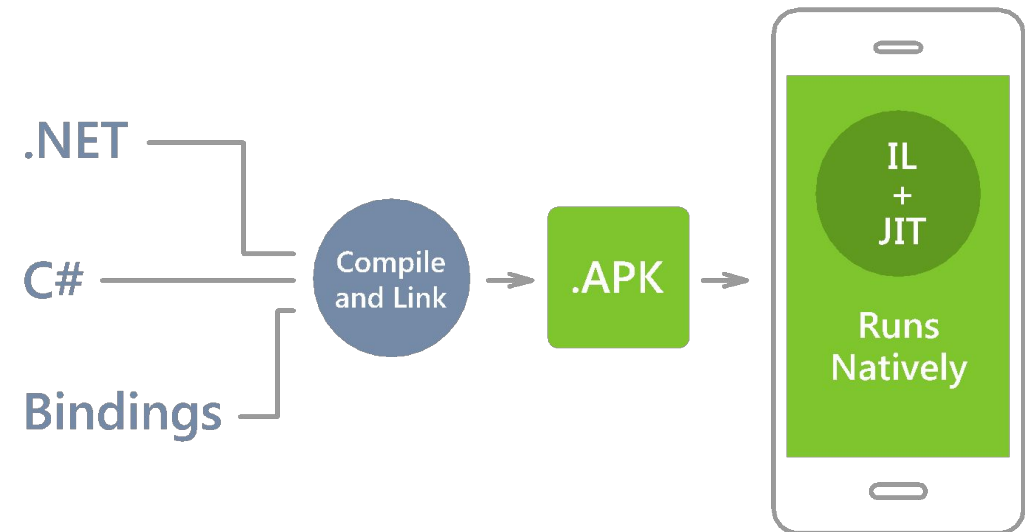
# Все, что можно сделать на Objective-C, Swift или Java можно сделать на C# и Visual Studio / Xamarin.



# Нативная производительность



[Xamarin.iOS](#) выполняет полную Ahead Of Time (AOT) компиляцию для создания бинарника ARM для App Store Apple .



[Xamarin.Android](#) использует Just In Time (JIT) компиляцию на устройствах Android.

✓ Всегда  
актуально

Поддержка «день в день»:

- iOS 5
- iOS 6
- iOS 7
- iOS 7.1
- iOS 8
- iOS 9
- iOS 10

Полная поддержка:

- Apple Watch
- Google Glass
- Android Wear
- Amazon Fire TV
- и т.д. и т.п.

Демо

