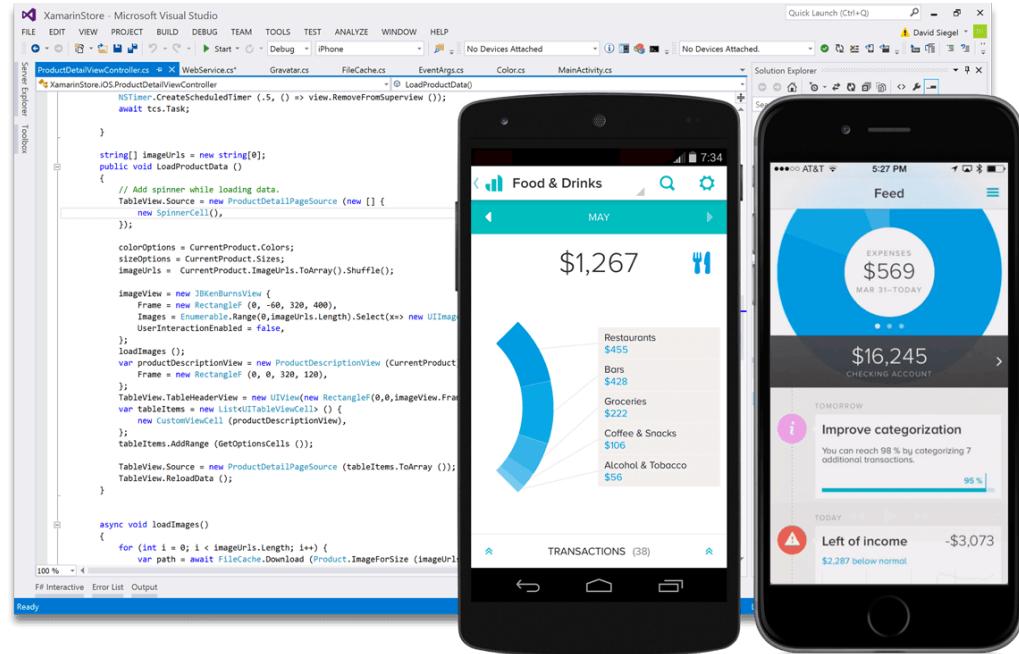


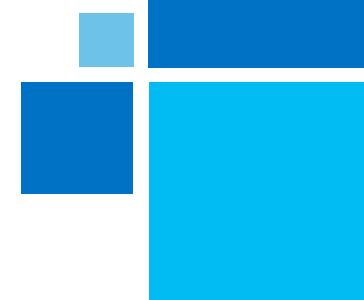
Xamarin

ことはじめ

Ralph (Tamaki Hidetsugu)



Xamarin



Microsoft Student Partners

自己紹介



玉木 英嗣（ラルフ）

明治大学理工学部情報科学科3年

アプリ・ウェブエンジニア

MSP内の活動：Minecraft・Web制作

 @r_ralph_h

 r-ralph

 TamakiHidetsugu

目次

- 概要
 - Xamarinとは
 - 必要な開発環境
 - 競合製品と比べて
 - 無料でできること、できないこと
- 開発入門
 - 開発環境構築方法 (Windows & Mac)
 - 新しくプロジェクトを作る (Windows & Mac)
 - 実行方法 (Windows & Mac with Android & iOS)
 - デモ
- さらに学ぶには
 - 公式サイトの見方
 - 見るべき入門サイト、情報源
 - ユーザーコミュニティ

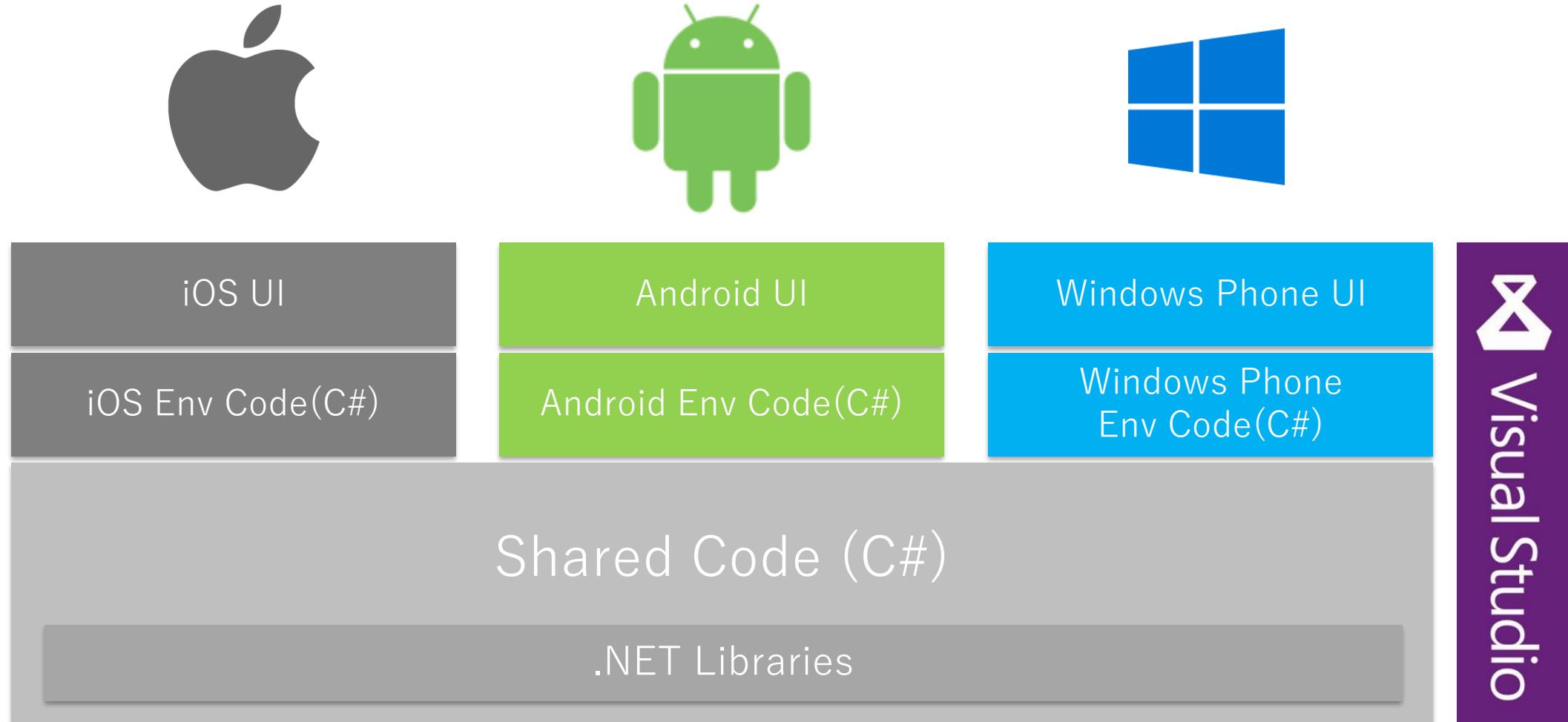
概要



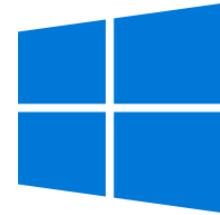
Xamarin (ざまりん)とは

- クロスプラットフォーム開発ツール
- C#/.NETで記述するだけで、iOS, Android, WindowsPhone で動作できる
- もちろん、NuGetを使って既存のライブラリを使用できる
- 今年3月末にあった Build2016 で無償化が発表された

Xamarin Native とは



Xamarin.Forms とは



iOS Env Code(C#)

Android Env Code(C#)

Windows Phone
Env Code(C#)



Visual Studio

Shared UI Code (XAML/C#)

Shared Code (C#)

.NET Libraries

PCLとShared Asset

- PCL (Portable Class Library)
共通コード 자체も独立した1プロジェクトとして管理する
コンパイルが行われた時の成果物が各環境のプロジェクトで使われる
NuGetによるライブラリ参照ができる
- Shared Asset (Shared Projects)
共通コード部分は、それを使うプロジェクトのビルド時に同時にビルドされる
C#内的一般的なAPIへのみにしかアクセスできない
`#if __ANDROID__`などで分岐可能

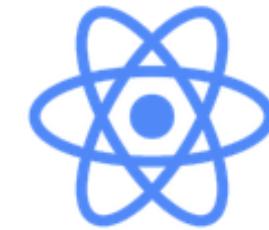
必要な開発環境

- 開発環境
Windows: Visual Studio
Mac: Xamarin Studio
- Windows上で、iOSアプリのStoryboard編集や、ビルドを行うためには、Xamarin StudioをインストールしたMacが必要

競合製品と比べて



APACHE
CORDOVA™



React Native

A framework for building native apps with React.

競合製品と比べて

- Xamarinのいい点
 - Cordovaのような、Hybrid HTMLを使用したものでは、実行時の遅さが目立つたり、システムが提供するハードウェアアクセラレーションや、外部装置へのアクセスが制限される。
 - コードはコンパイルされ、完全にプラットフォームごとのネイティブ動作になるため早い
 - それぞれのプラットフォームでのクラス名・関数名がほぼ同じ名前でAPIとして提供されているので、いずれかのプラットフォーム上でアプリ経験があると慣れやすい。
 - C#/.NETで制作された数多くのライブラリ資産を活用できる。

競合製品と比べて

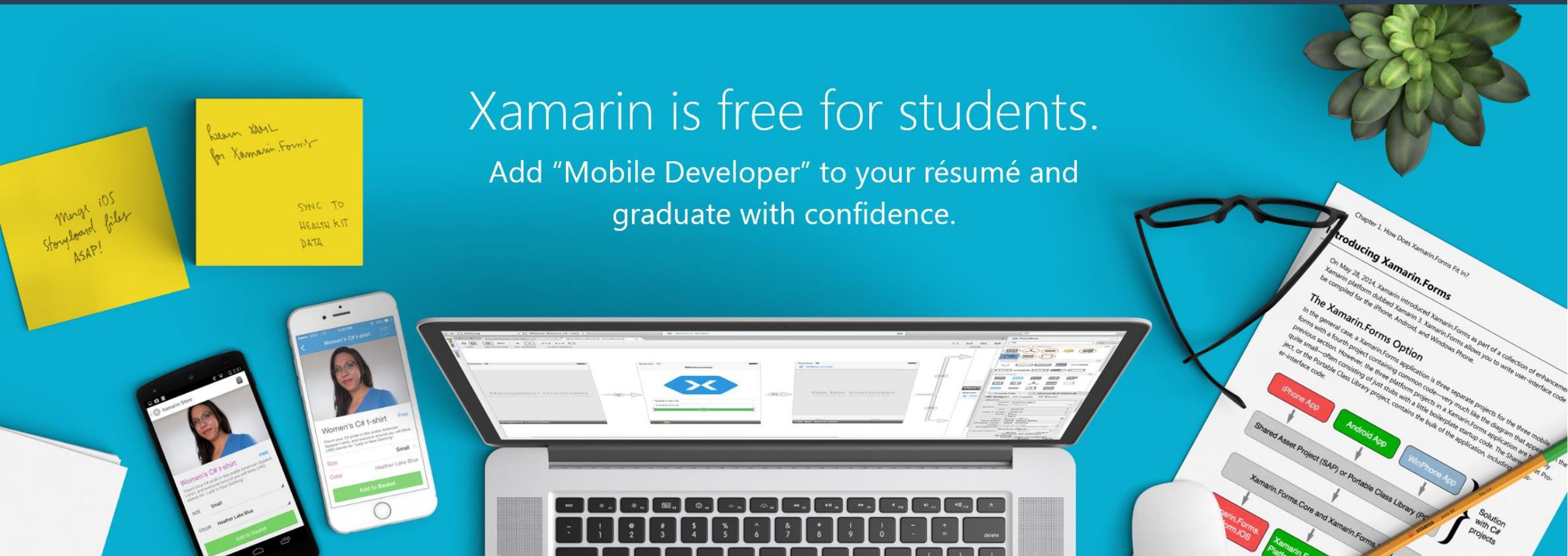
- Xamarinの悪い点
 - HTML/JSとC#では、前者のほうが触ったことがある人が多い
 - 各々のプラットフォームの固有動作やAPIなどについて知らないといけない。
 - 最近になって無償化されたため、ユーザーが少なく、Web上の情報も少ない
 - 共有ライブラリにも制限があり、使えない場合がある。

無料でできること、できないこと

[Products ▾](#)[Customers](#)[Pricing](#)[Developers](#)[Support ▾](#)[Resources ▾](#)[Sign in](#)

Xamarin is free for students.

Add “Mobile Developer” to your résumé and graduate with confidence.



無料でできること、できないこと

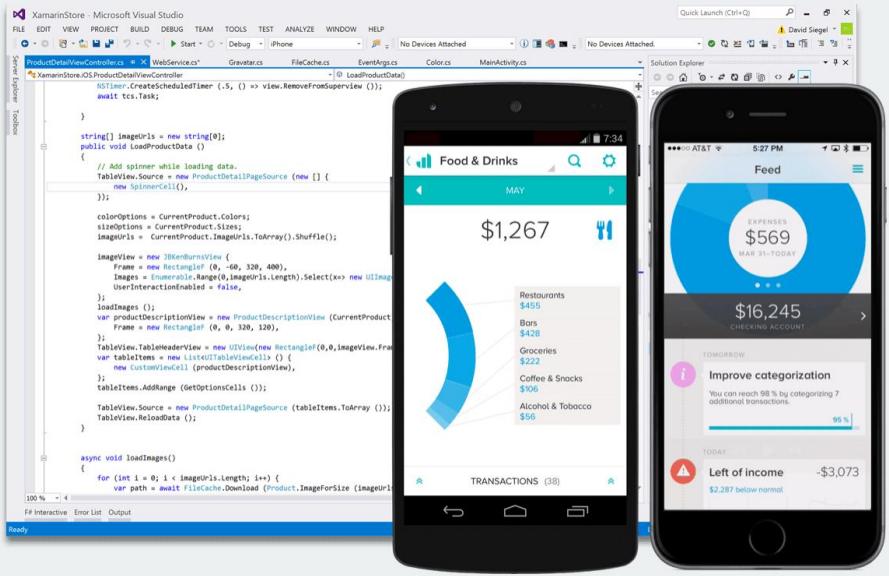
- 学生はすべて無料でできる
- 学生だとむしろ得なことが増える
 - Free枠以上で提供されるプログラムへの参加権 (Xamarin University)
 - Xamarin Test Cloud の利用権
 - などなど
- <https://www.xamarin.com/student>

開発入門



開発環境構築方法

1. Xamarin のサイトへアクセス
2. スクロールすると **Download Xamarin** というリンクがあるのでクリック



Xamarin

Deliver native Android, iOS, and Windows apps, using existing skills, teams, and code.

Build native apps for multiple platforms on a shared C# codebase. Use the same IDE, language, and APIs everywhere.

- ✓ Native UI, native API access, and native performance
- ✓ Anything you can do in Objective-C, Swift, or Java you can do in C# with Xamarin
- ✓ Ship cutting-edge apps with same-day support for new OS releases

[Download Xamarin](#) [Learn More](#)

開発環境構築方法

Full name

Email

Company

I already have Visual Studio installed

Please select

My company has more than 5 developers

Please select

I agree to the [Terms & Conditions](#)

Download Now

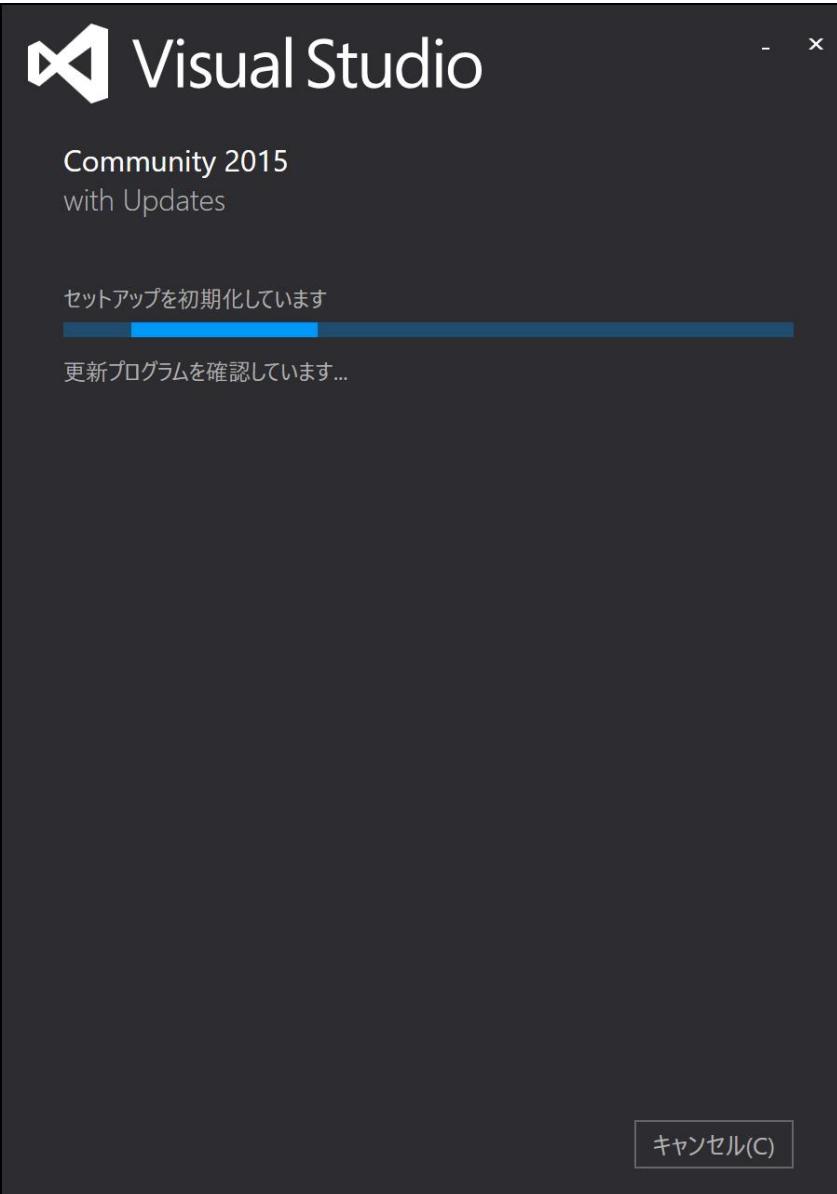
or download [Xamarin Studio for OS X](#)

名前・メールアドレス・大学名を入力

既にVisual Studioをインストールしている場合はYes

Noを選択

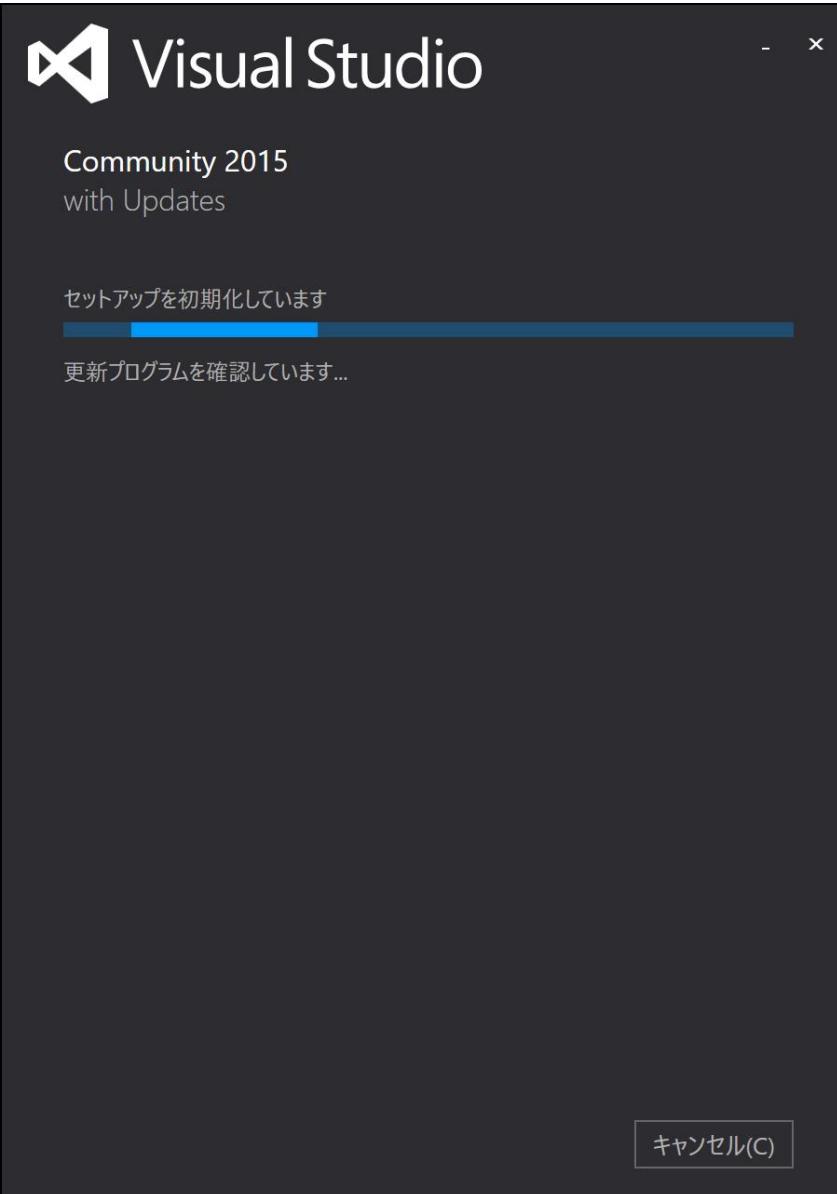
開発環境構築方法（Windows）



Visual Studioが入っていない場合は
Community 2015のインストールが始まる。

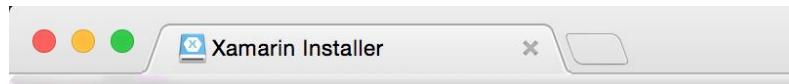
既にインストールされている場合は
Xamarinのパッケージのみがインストール
される。

開発環境構築方法（Windows）



- Visual Studioが入っていない場合はCommunity 2015のインストールが始まる。
- 既にインストールされている場合はXamarinのパッケージのみがインストールされる。
- インストールするコンポーネントはデフォルトのままでOK

開発環境構築方法（Mac）



A better way to build apps.



● Install Xamarin

1. ダウンロードしたXamarinInstaller.dmgを開く
2. 出てきた画面の「Install Xamarin」をダブルクリック

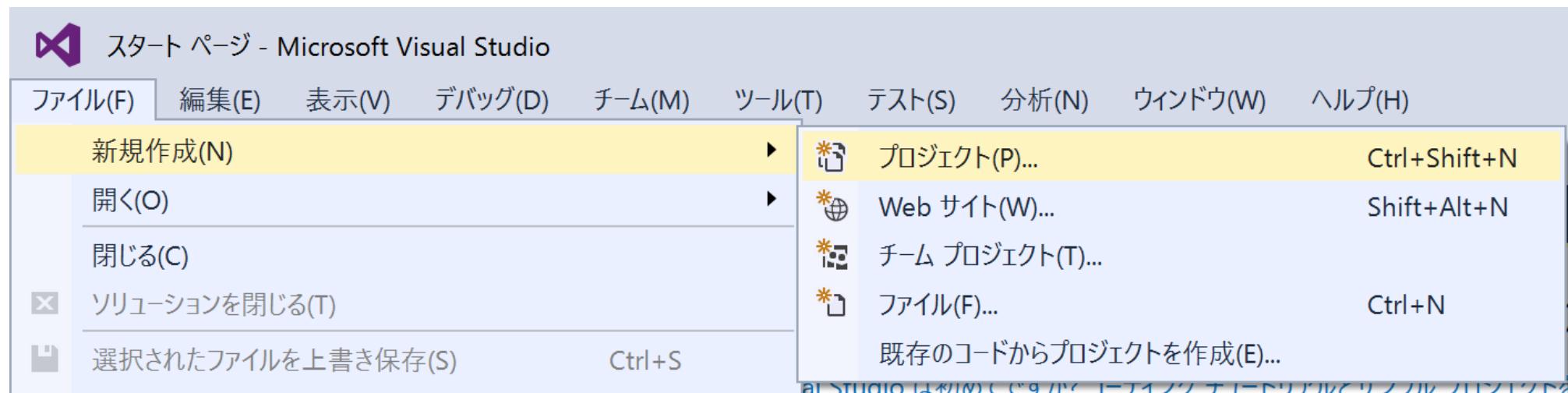
開発環境構築方法（Mac）

基本的に Continue を押していくばOK

The image shows two screenshots of the Xamarin Installer on a Mac. The left screenshot displays the 'Read and accept this license in order to proceed with the installation' screen. It features the Xamarin logo and a large 'X' icon. On the left sidebar, there are navigation links: 'Xamarin License', 'Prerequisites', 'Installation', and 'Summary'. The main content area contains the 'Xamarin Software License Agreement' (Last updated: November 12, 2015), sections for 'ACCEPTANCE' and 'DEFINITIONS', and two checkboxes at the bottom: 'I accept the terms of the License Agreement' (unchecked) and 'Automatically send errors and usage information to help Xamarin improve your experience' (checked). Buttons for 'Go back' and 'Continue' are at the bottom. The right screenshot shows the 'Installation in progress' screen. It also features the Xamarin logo and a large 'X' icon. The sidebar has the same navigation links. The main content area displays the status of various components: Mono Framework 4.4.0 (downloading), Android SDK 24.4.1, Xamarin Studio 6.0.0, Xamarin.Android 6.1.0, and Xamarin.iOS 9.8.0. A progress bar indicates the download of Mono Framework is 34.08MB (current: 302.59MB; total 2.33GB) at 955.60KB/s. Buttons for 'Go back' and 'Continue' are at the bottom.

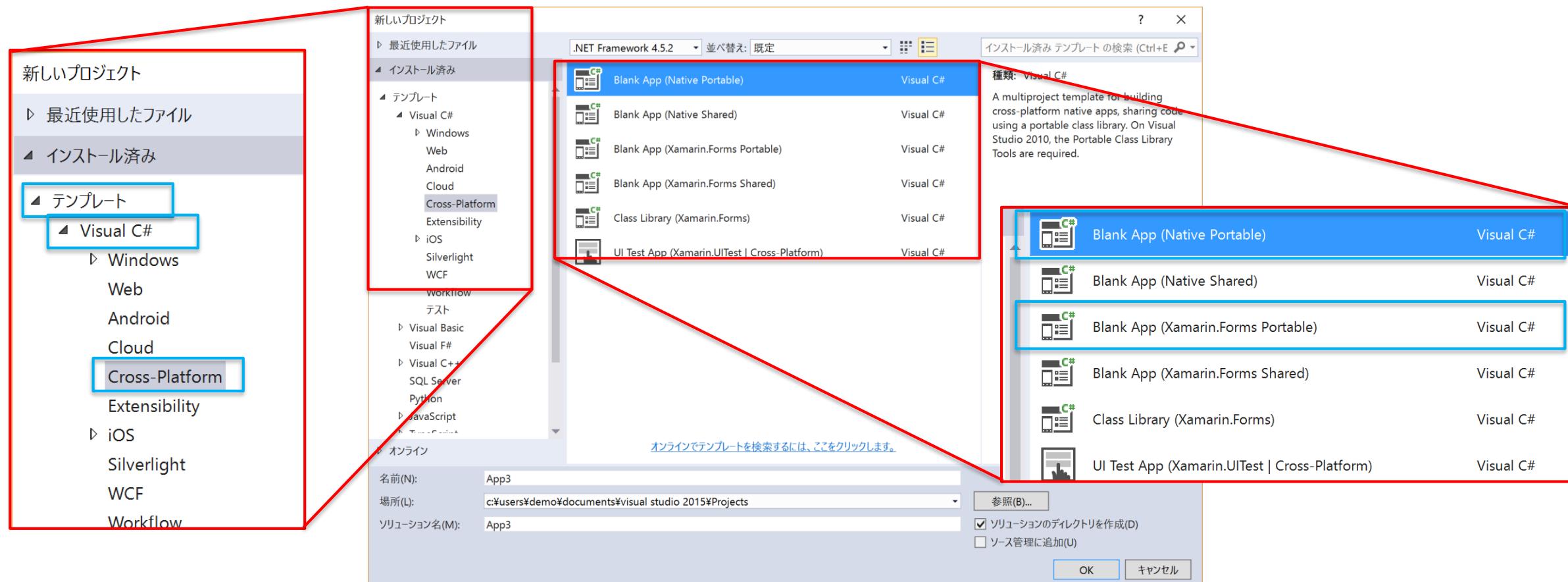
新しくプロジェクトを作る (Windows)

- ファイル→新規作成→プロジェクト



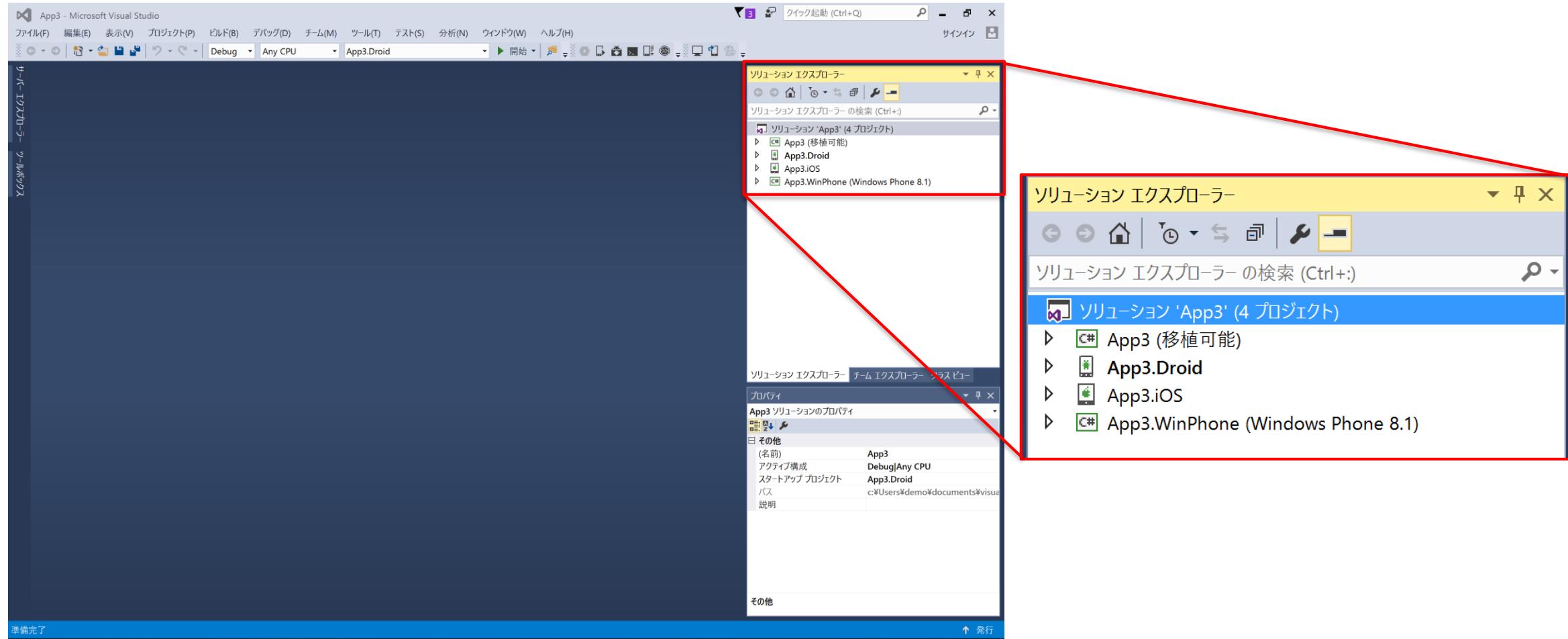
新しくプロジェクトを作る (Windows)

- テンプレート → Visual C# → Cross Platform → Blank App
 - Xamarin Native なら Blank App (Native Portable)
 - Xamarin.Forms なら Blank App (Xamarin.Forms Portable)



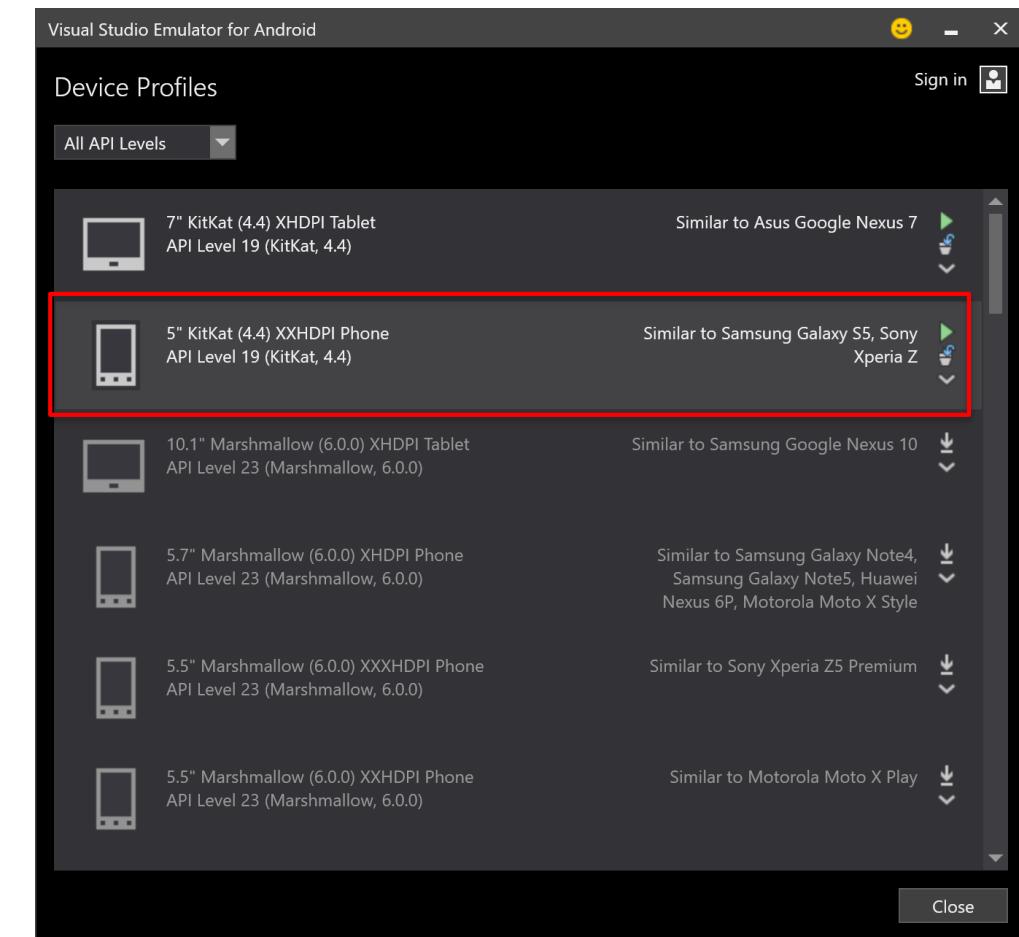
新しくプロジェクトを作る (Windows)

- 完成！



実行方法（Windows & Android）

- ・ ツール → Visual Studio Emulator for Android
- ・ 「5“ KitKat ~」をダブルクリック
- ・ 起動するまで待機

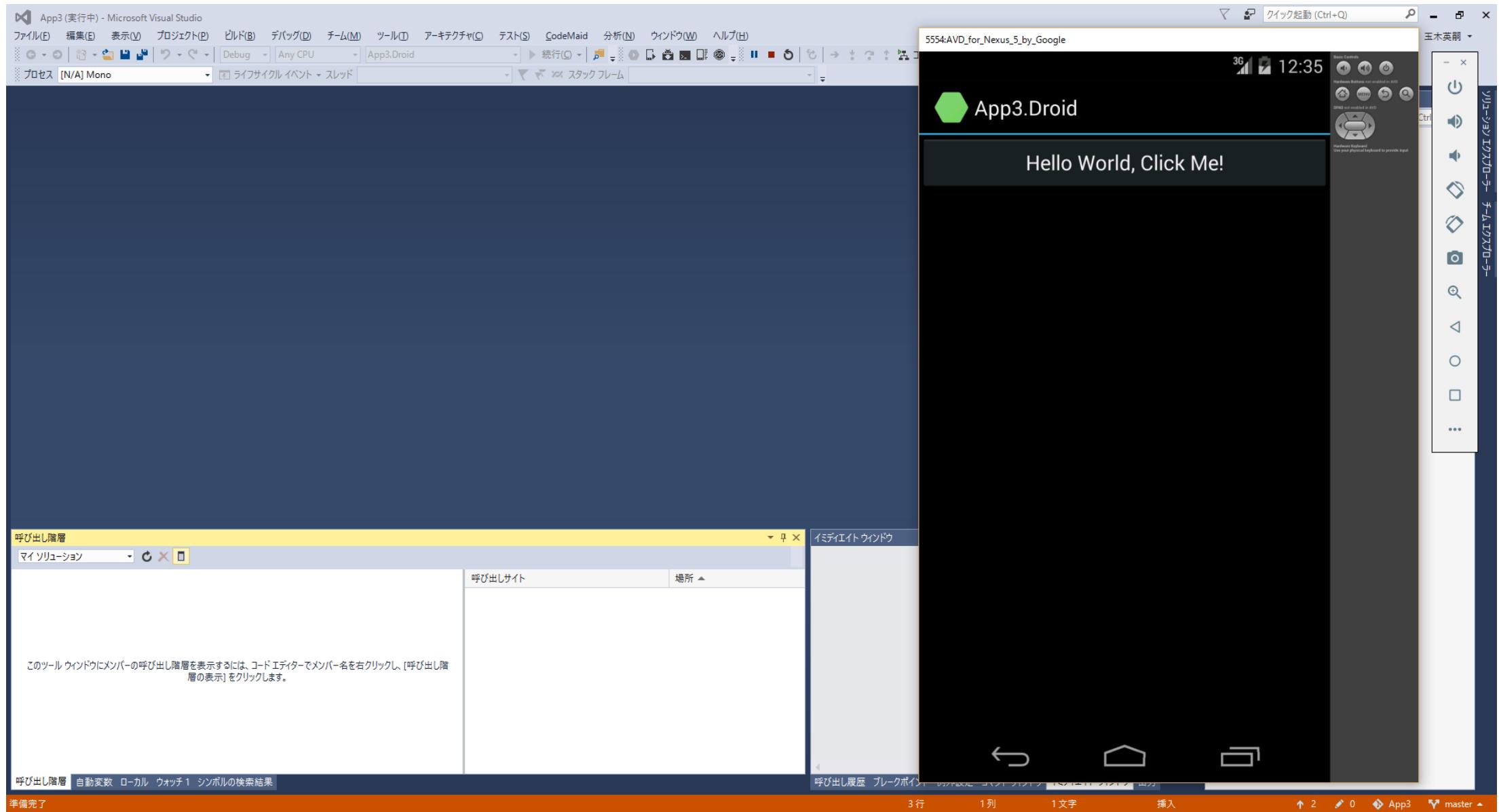


実行方法（Windows & Android）

- 「ソリューション名.Droid」が選択されていることを確認して実行ボタン ➤ をクリック

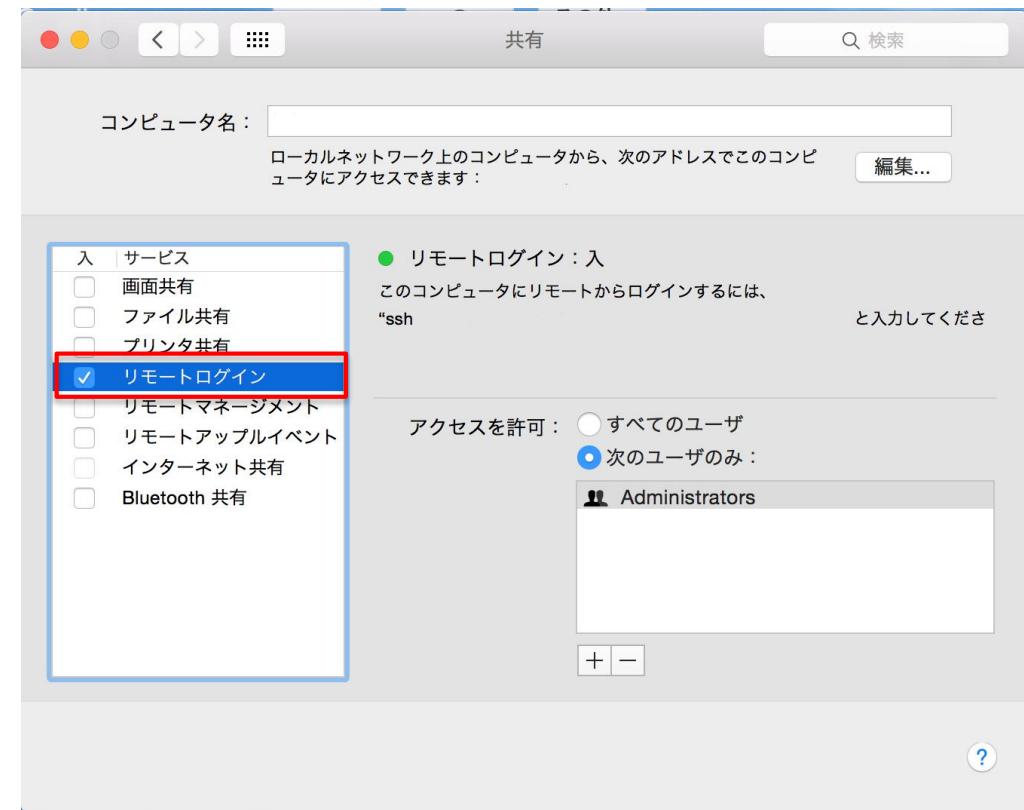
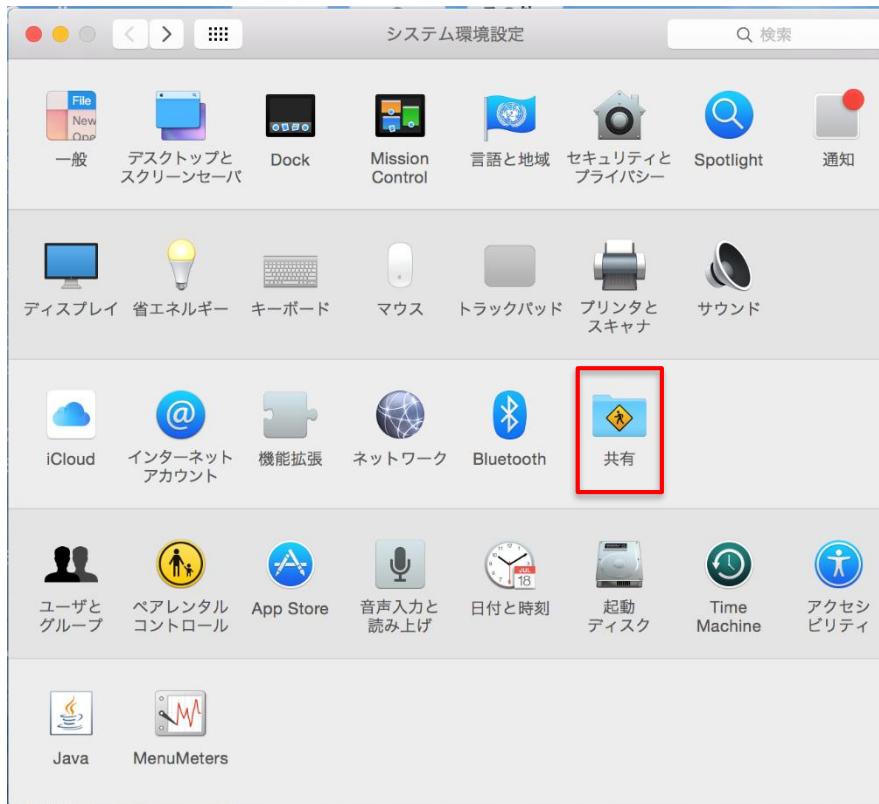


実行方法 (Mac & Android)



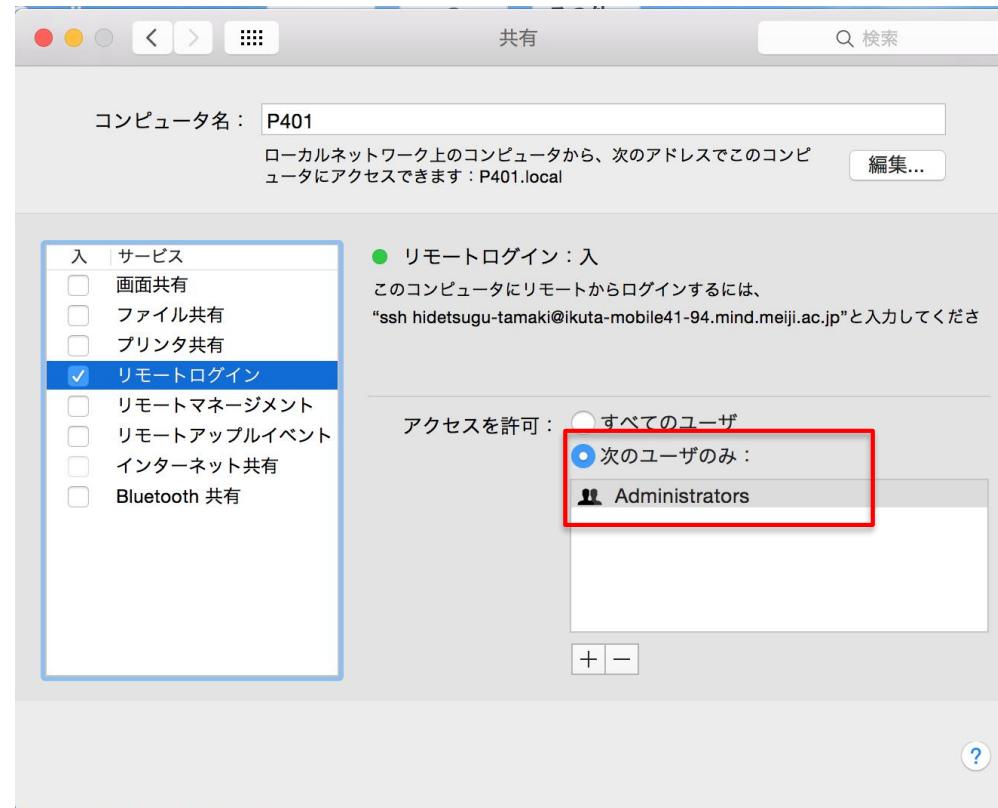
実行方法（Windows & iOS）

- 初めにMac側で設定が必要
- システム環境設定 → 共有 → 「リモートログイン」にチェックを入れる



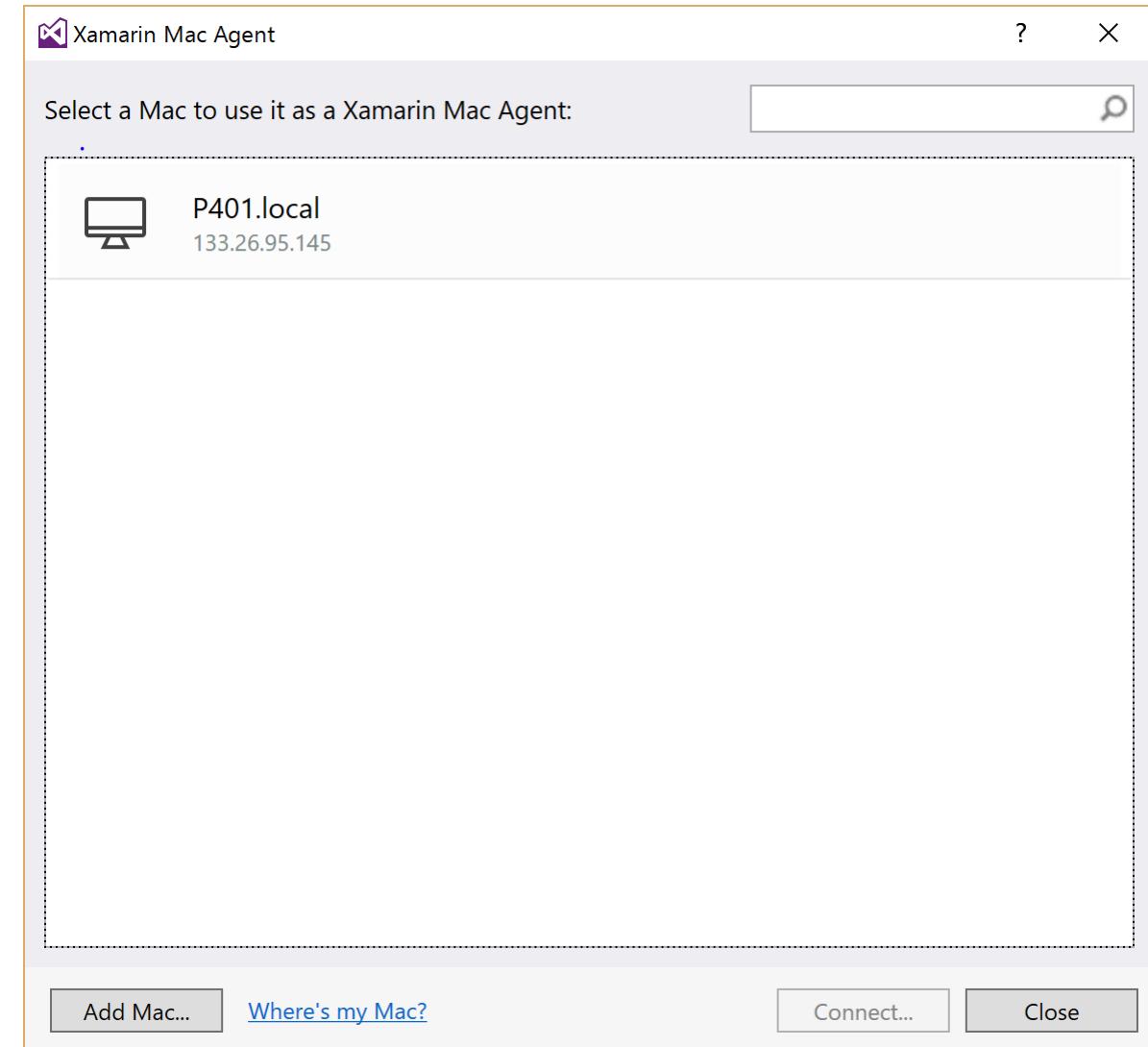
実行方法（Windows & iOS）

- 「次のユーザーのみ」にチェックを入れ、Xamarin Studioをインストールしたユーザーorグループを選択



実行方法（Windows & iOS）

- iOSプロジェクトを開こうとすると右の画面が出てくる
- 同じネットワーク内にMacが見つかればリストに出てくる
- ユーザー名とパスワードを入力して完了



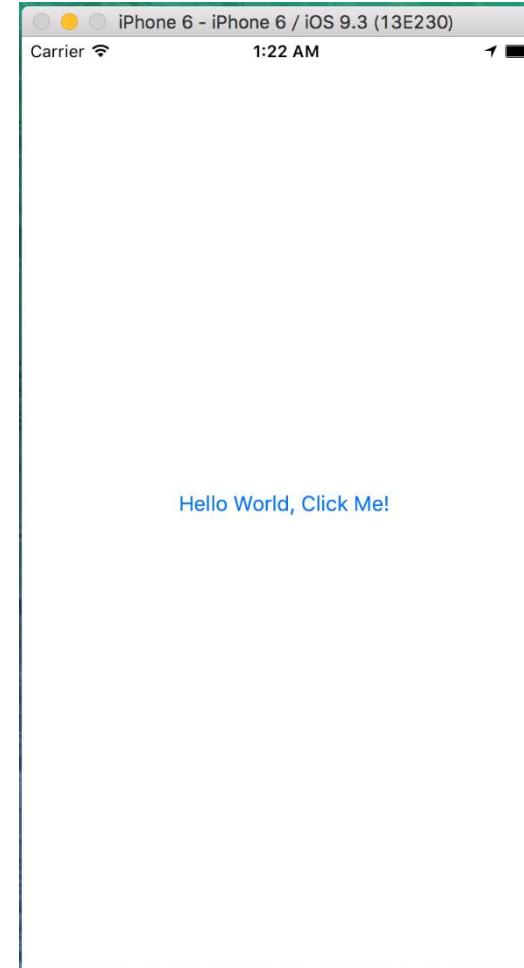
実行方法（Windows & iOS）

- 「iPhoneSimulator」「ソリューション名.iOS」が選択されていることを確認して実行ボタン ▶ をクリック
 - ▶ の右側では、起動するシミュレーターの環境を設定可能



実行方法（Windows & iOS）

- シミュレーターが立ち上がってアプリが実行された！
(つなげたMacの上で)



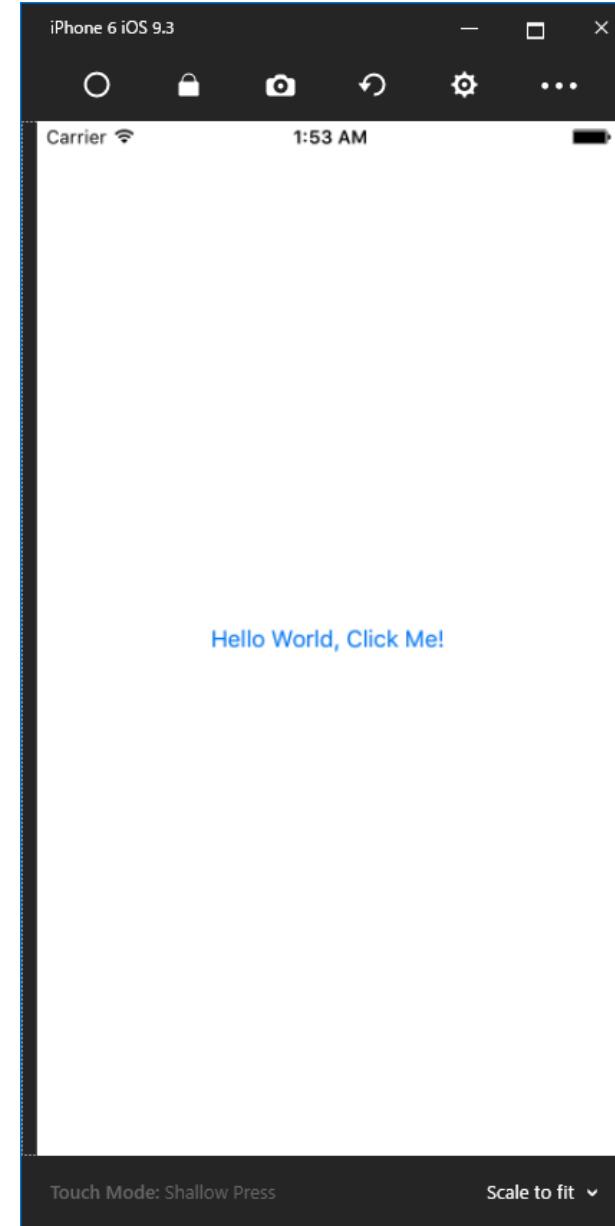
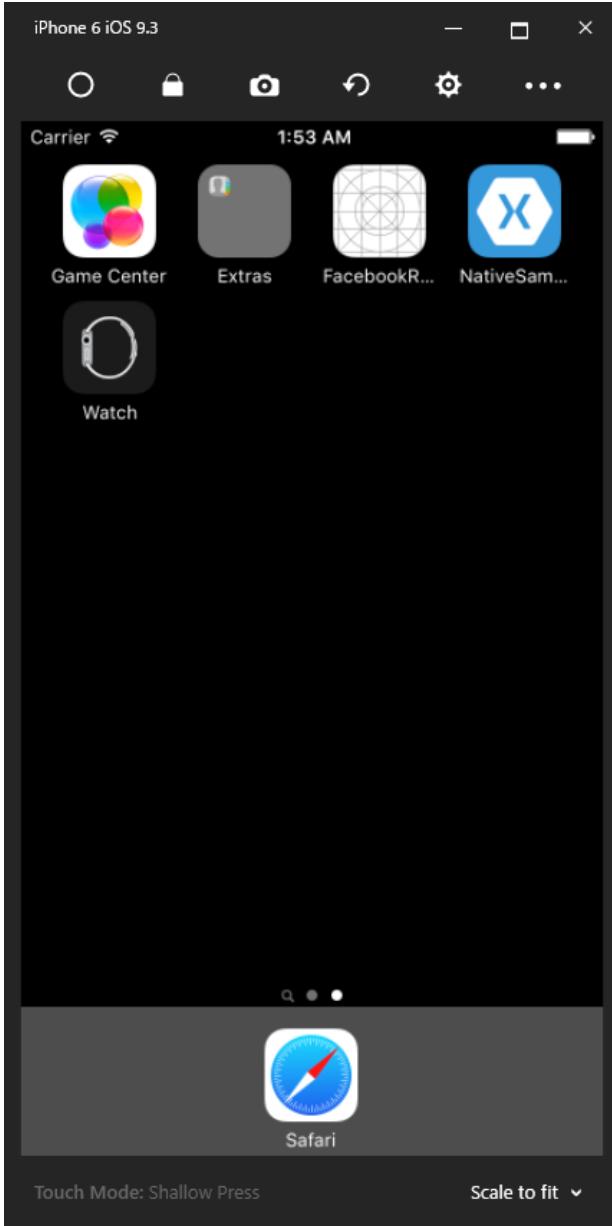
実行方法（Windows & iOS）

- Windows上でシミュレーターを実行させたい場合
- <https://developer.xamarin.com/guides/cross-platform/windows/ios-simulator/>

実行方法（Windows & iOS）

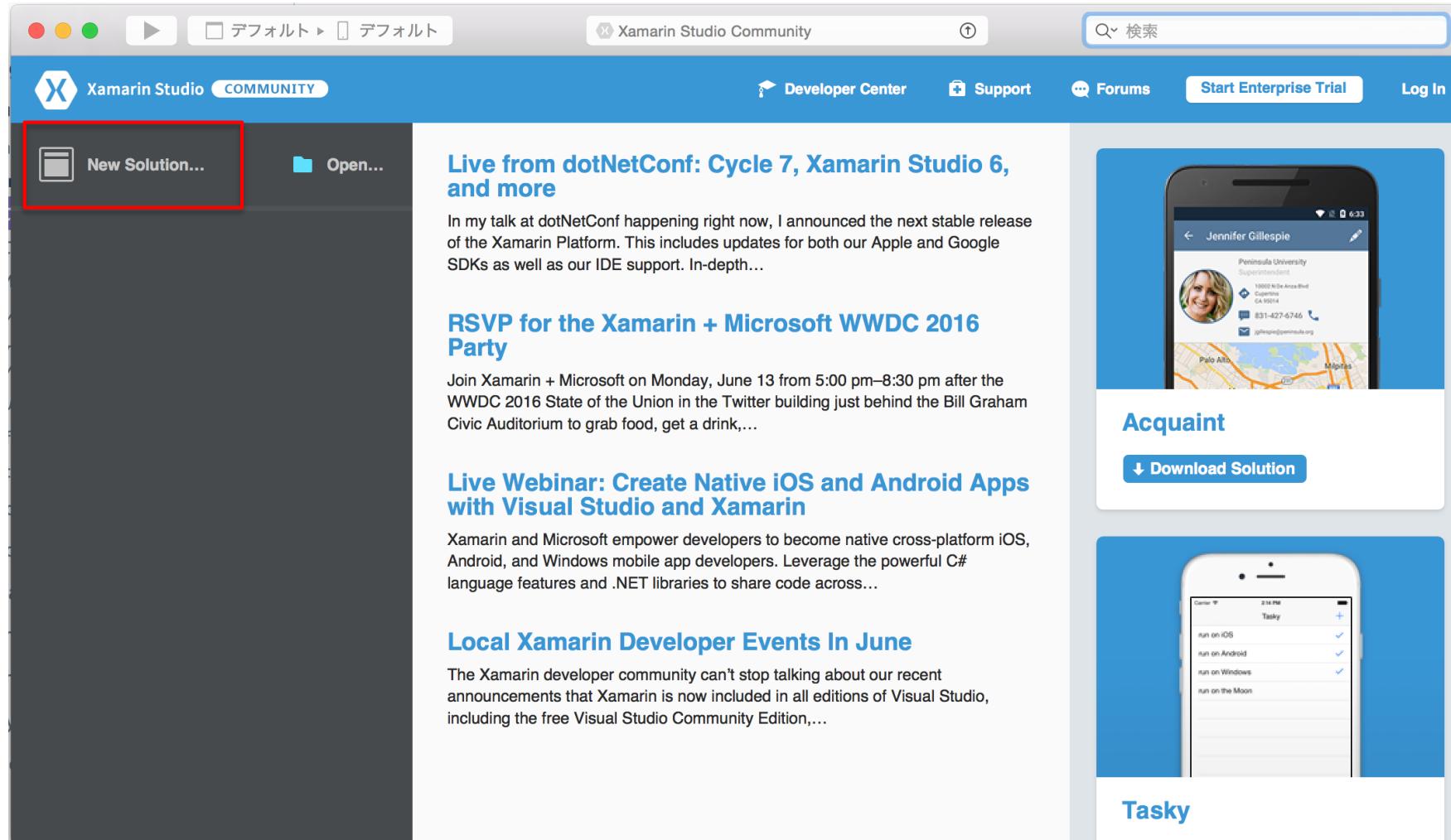
1. WindowsとMac両方のXamarinをBetaもしくはAlphaにする
2. Windows上でシミュレーターをインストールする
3. Visual Studio上のオプションからiOSシミュレーターを有効にする

実行方法 (Windows & iOS)



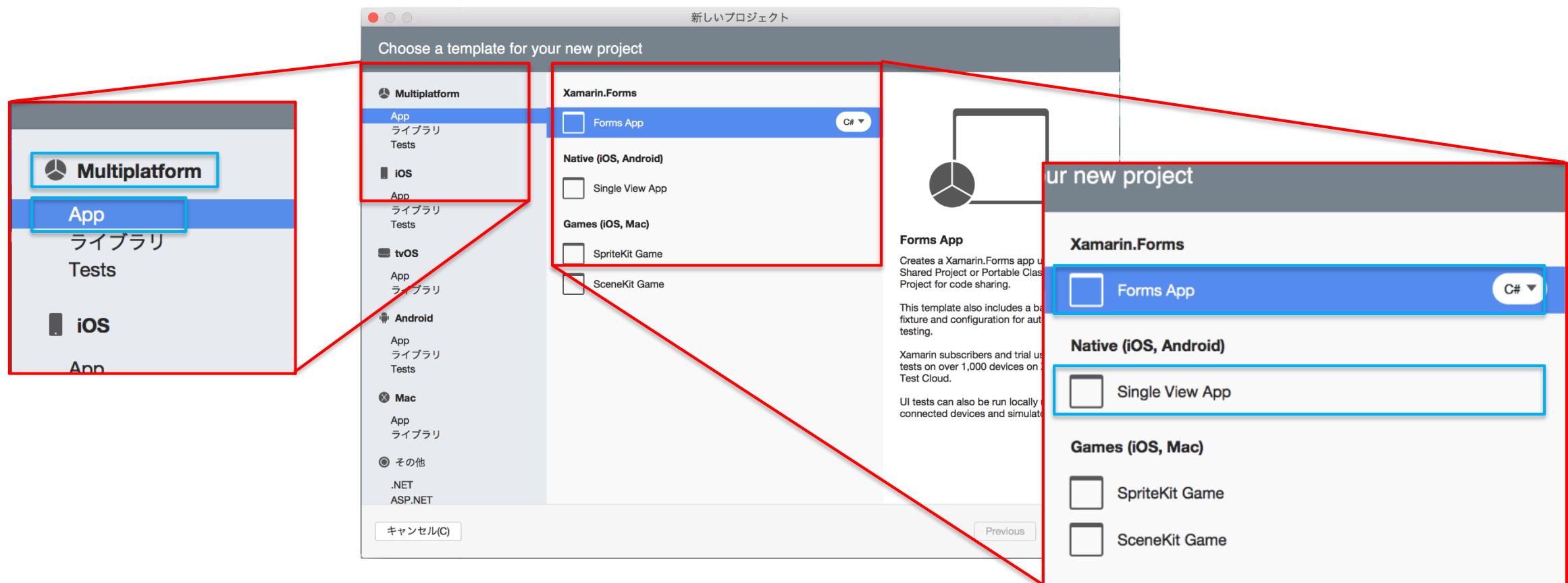
新しくプロジェクトを作る (Mac)

- New Solution を選択



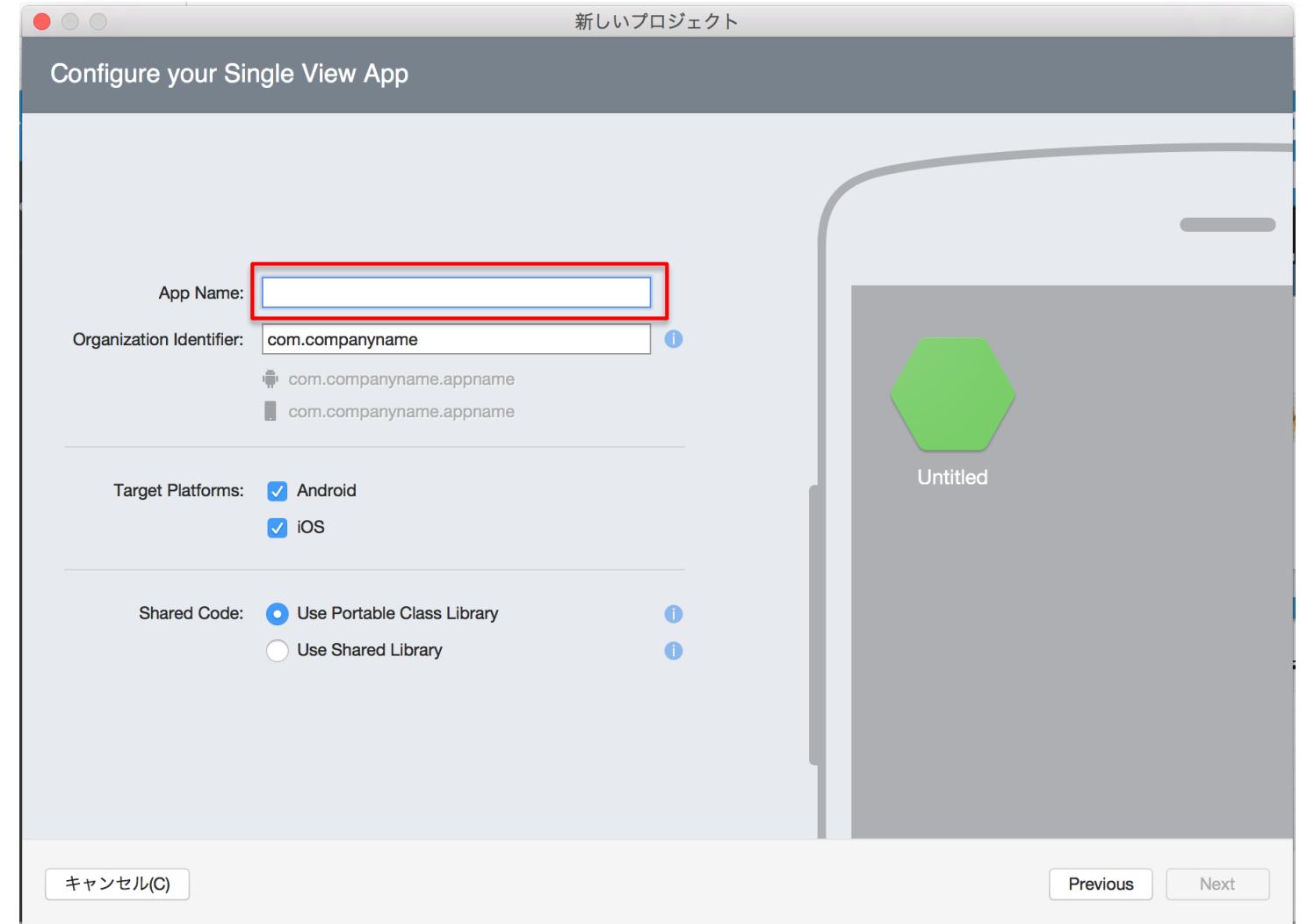
新しくプロジェクトを作る (Mac)

- Multiplatform→ App
 - Xamarin Native なら Single View App
 - Xamarin.Forms ならForms App



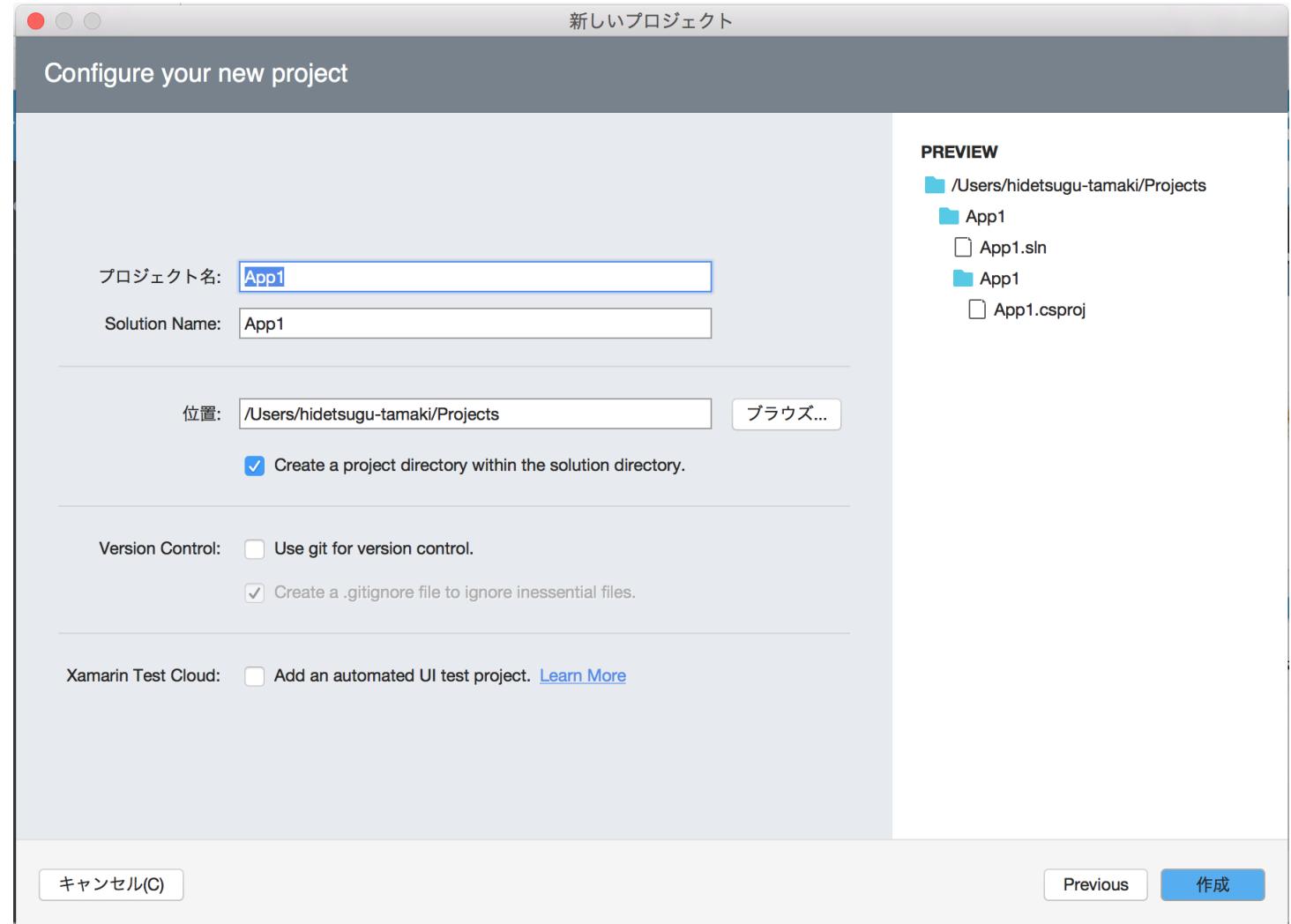
新しくプロジェクトを作る (Mac)

- アプリ名を入力する



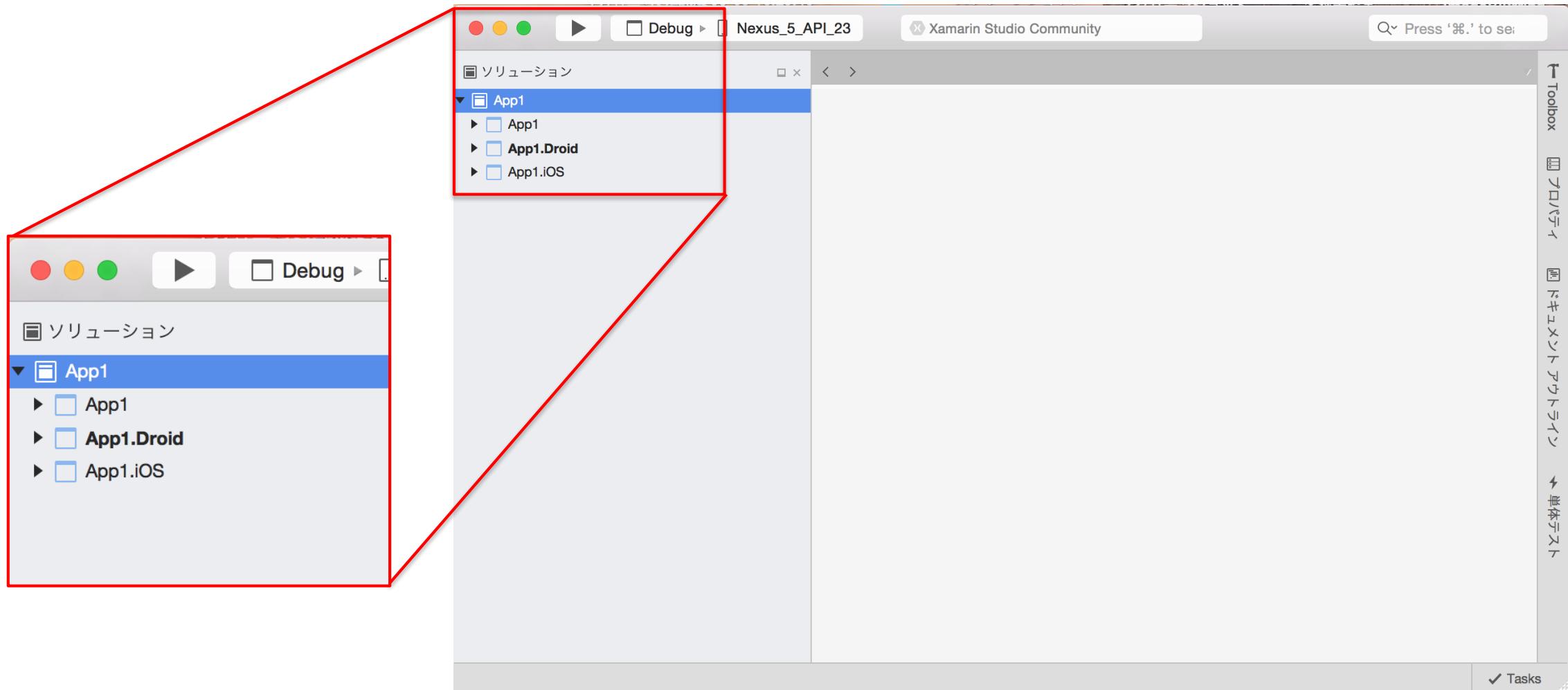
新しくプロジェクトを作る (Mac)

- プロジェクト名や保存場所などの確認



新しくプロジェクトを作る (Mac)

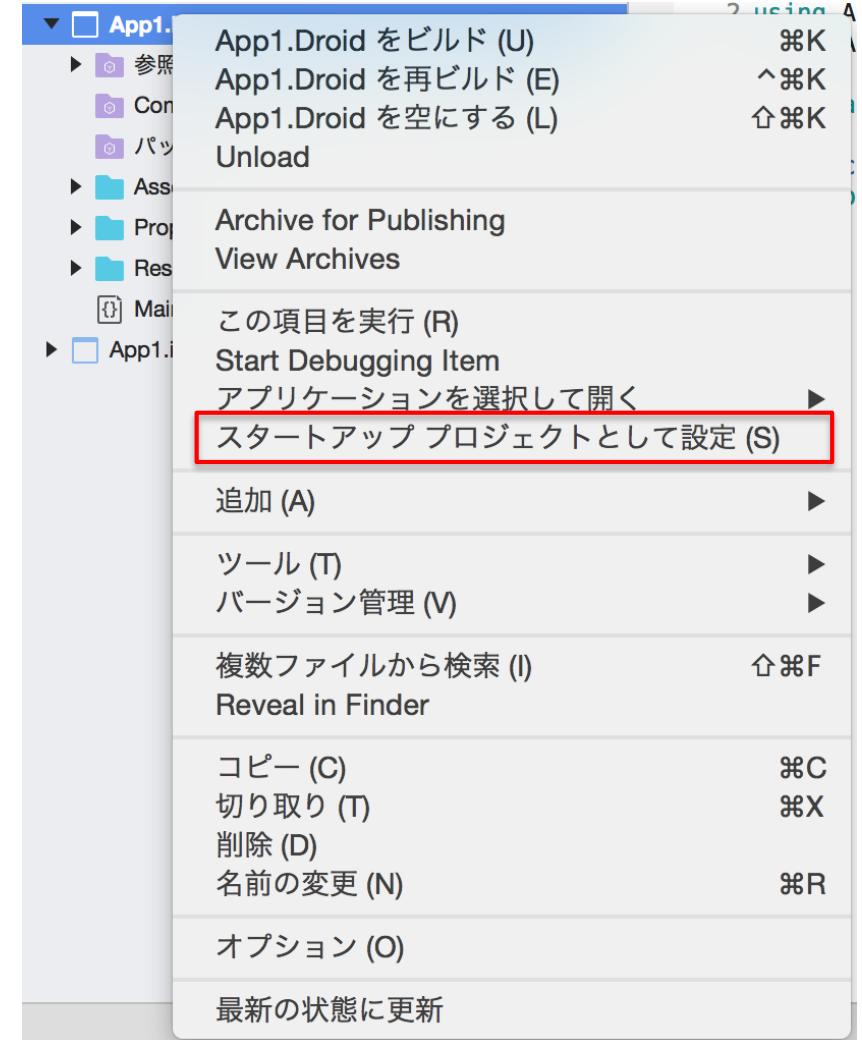
- 完成！



実行方法（Mac & Android）

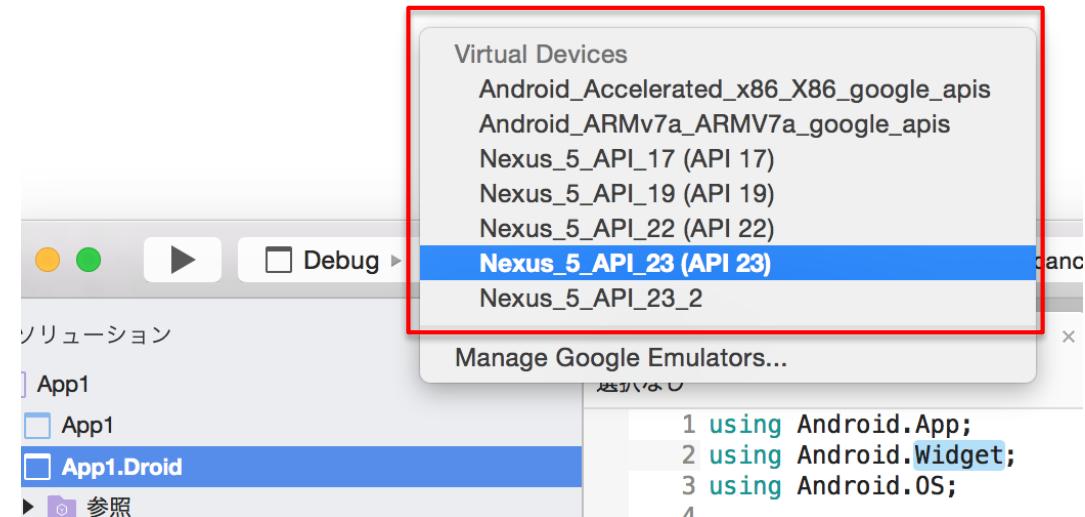
1. ソリューション名.Droidを右クリック

2. 「スタートアッププロジェクトとして設定」をクリック

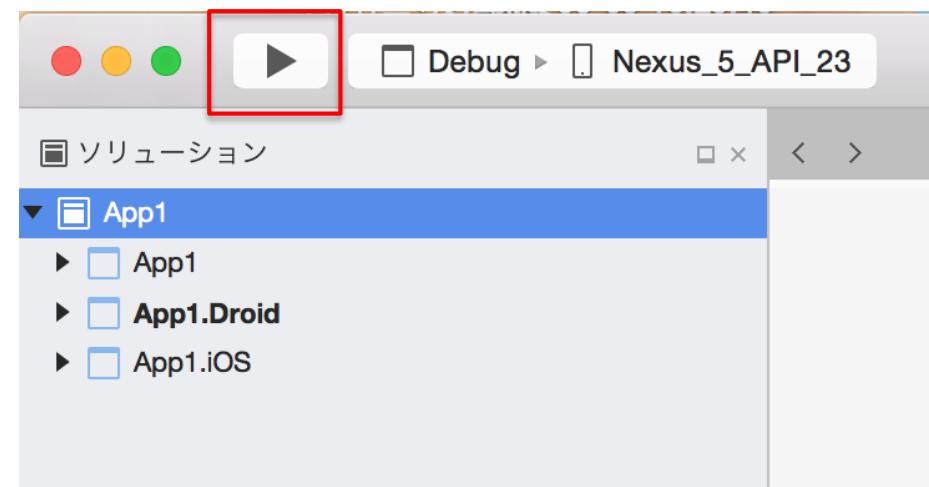


実行方法（Mac & Android）

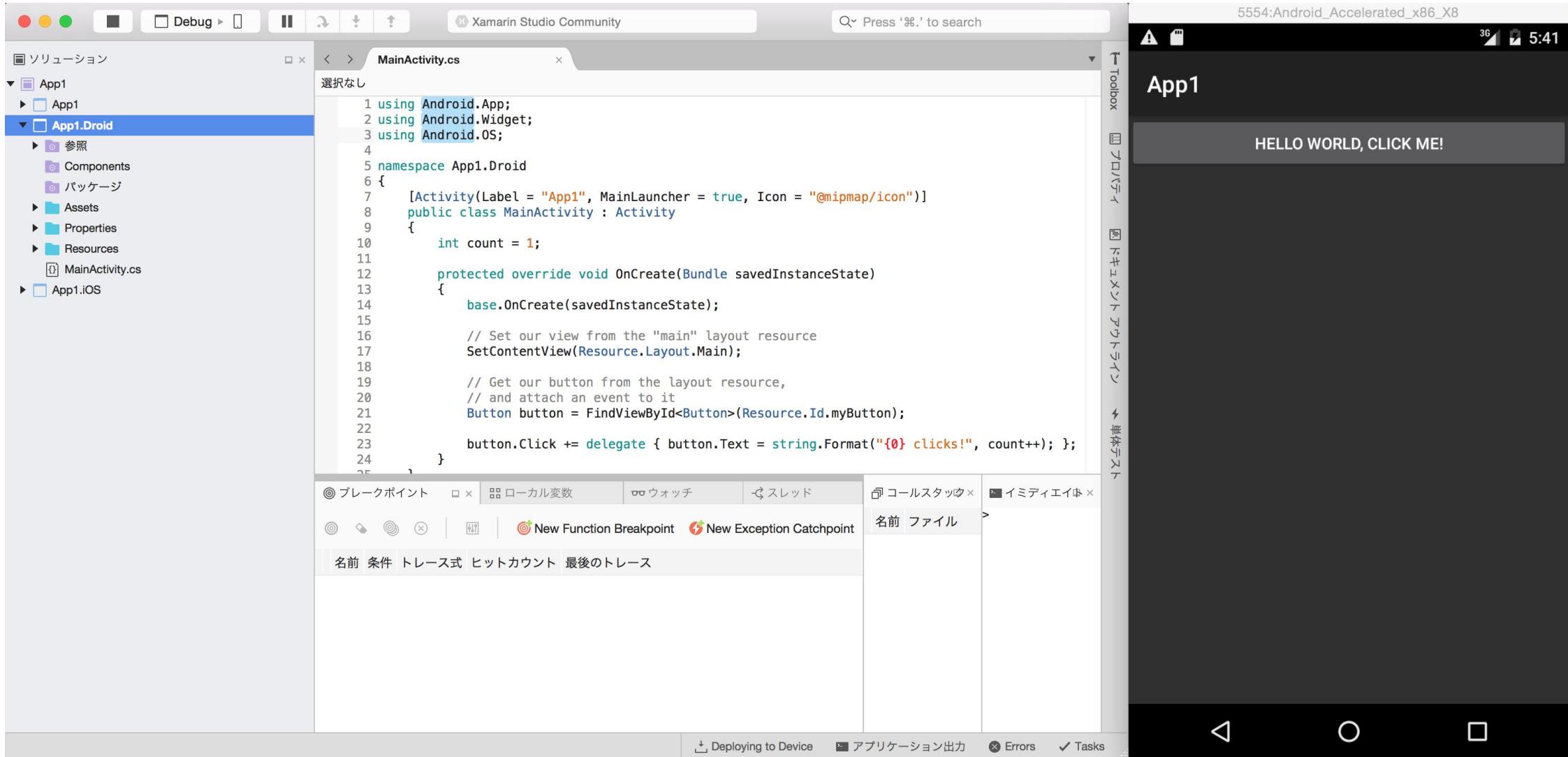
1. 起動するエミュレーターを選択



2. ▶ をクリック



実行方法 (Mac & Android)



実行方法（Mac & iOS）

1. ソリューション名.iOSを右クリック

2. 「スタートアッププロジェクトとして設定」をクリック

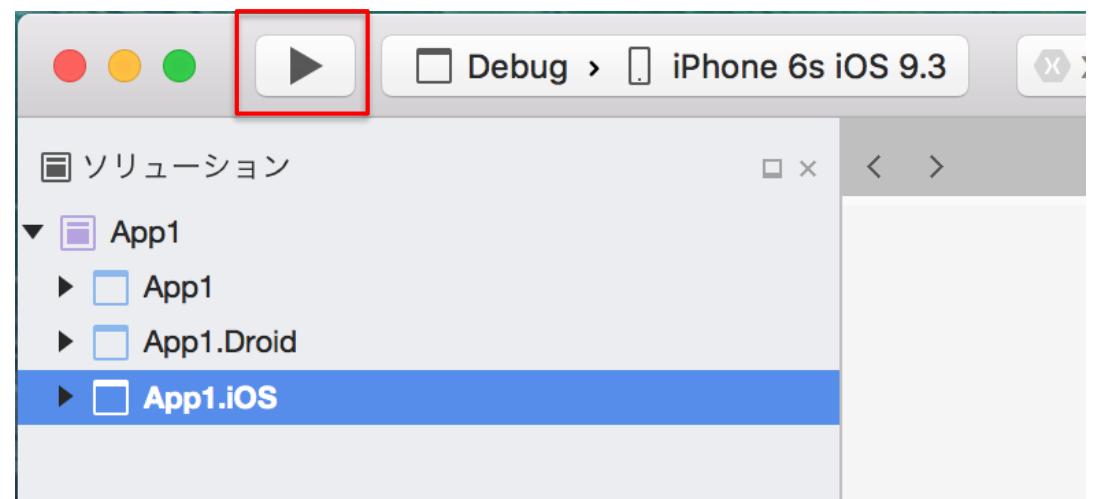


実行方法（Mac & Android）

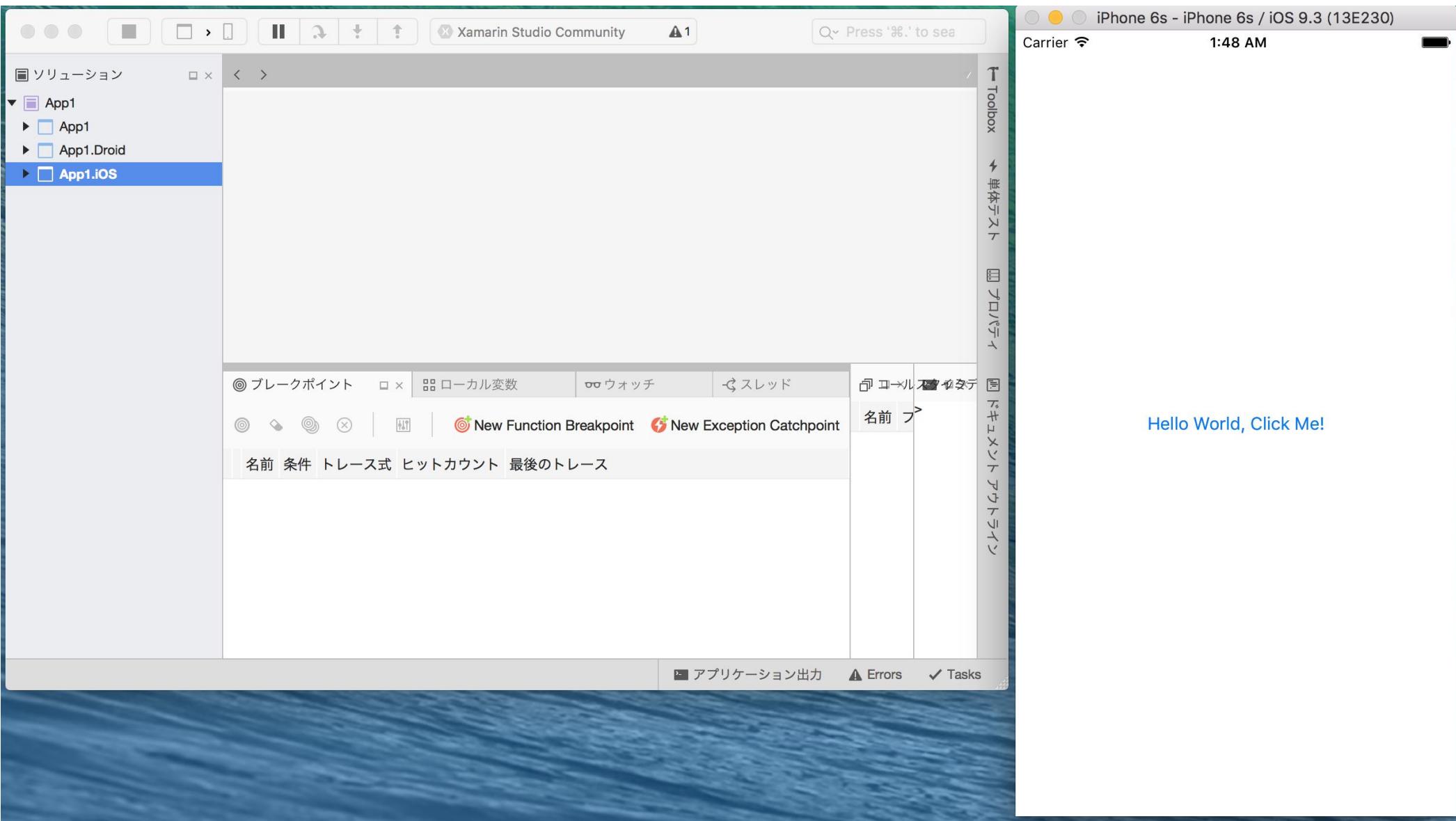
1. 起動するエミュレーターを選択



2. ▶ をクリック

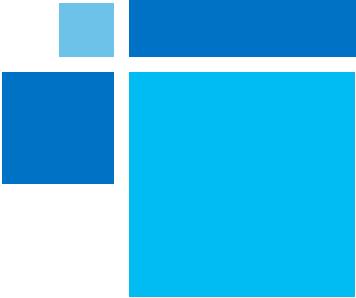


実行方法 (Mac & iOS)



デモ

- Github上からデータをダウンロードし、表示するアプリ
- データを取得する部分を共有コードとして書いてみる



さらに学ぶには

公式サイトの見方

- API References
<https://developer.xamarin.com/api/>
共通APIやプラットフォーム固有のAPIの説明が書かれている
- Guides
<https://developer.xamarin.com/guides/>
- Recipes
<https://developer.xamarin.com/recipes/>
各プラットフォームのAPIを使用したサンプルなどが置かれている

見るべき入門サイト、情報源

- まだまだ日本語の情報が不足しています(ノド')
- 無償化される前の敷居が高すぎた
- 日本のXamarin界で有名なのはエクセルソフトの田淵さんと、MSのXamarinエバンジェリストのちょまどさん
- お二方のブログに結構つまっている
- @ytabuchi / @chomado

ユーザーコミュニティ

- JXUG (Japan Xamarin User Group)
<http://jxug.org/>
- たぶん日本唯一のXamarinコミュニティ
- 月一程度で勉強会が開かれている（東京）
- 地方でも時々開かれている

まとめ

Xamarinはいいぞ