# Re:ゼロから始めるアプリ開発

— Xamarin ってなぁに？ —

ここあ　著

2019 年 9 月 22 日　 ver 1.0

**■免責**

本書は情報の提供のみを目的としています。
本書の内容を実行・適用・運用したことで何が起きようとも、それは実行・適用・運用した人自身の責任であり、著者や関係者はいかなる責任も負いません。

**■商標**

本書に登場するシステム名や製品名は、関係各社の商標または登録商標です。
また本書では、™、®、© などのマークは省略しています。

# 目次

# 第1章 アプリを作るための準備

## 1.1 開発環境を整える

### ♣ 1.1.1 PC 側での準備

Android のアプリを作ろうと思うと，今は色々な環境を選択することができる．Java とか，Kotlin とか，もしかしたら Cocoa とかも聞いたことがあるかもしれない．

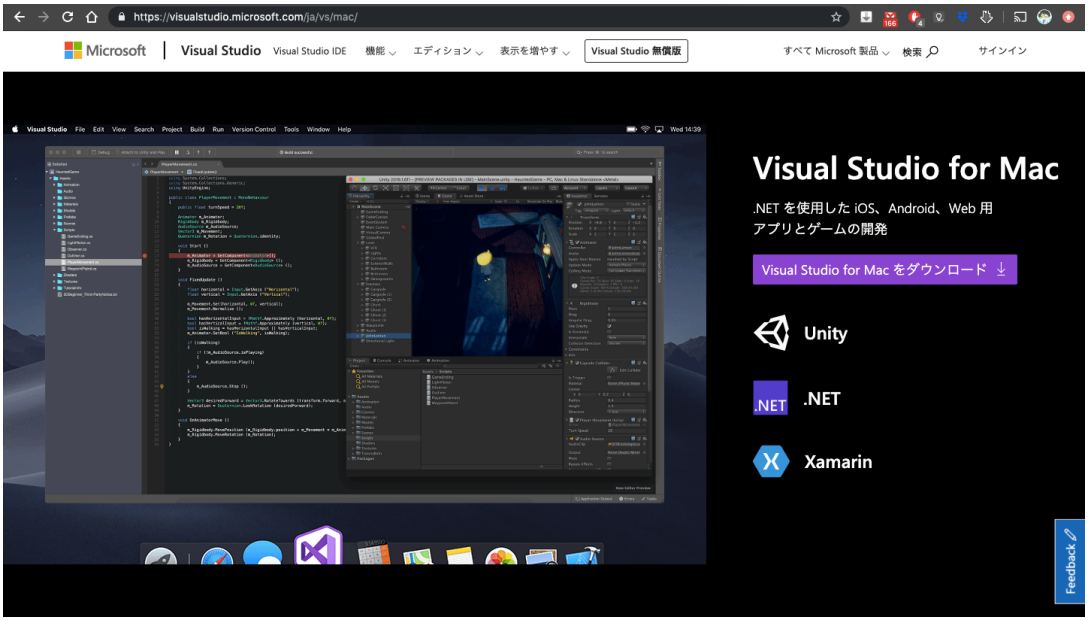
今回は，C#を用いて Android を開発することのできる `Xamarin` というフレームワークに焦点を合わせてみたいと思う．

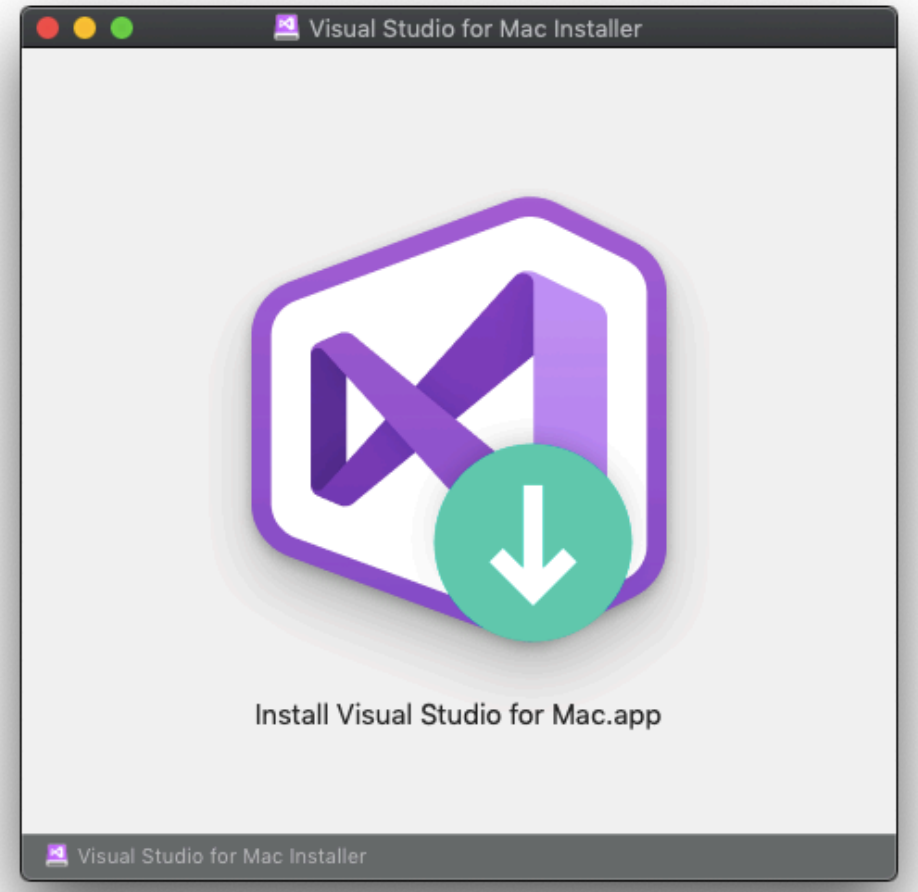
`Xamarin` というのは，Android アプリを作る上で欠かすことのできない `Android SDK` を C#でも使えるようにしたもの．

この，`Xamarin` というフレームワークは `Visual Studio` を使って開発することができる．

以降，インストールの手順を見ていきたいと思う．

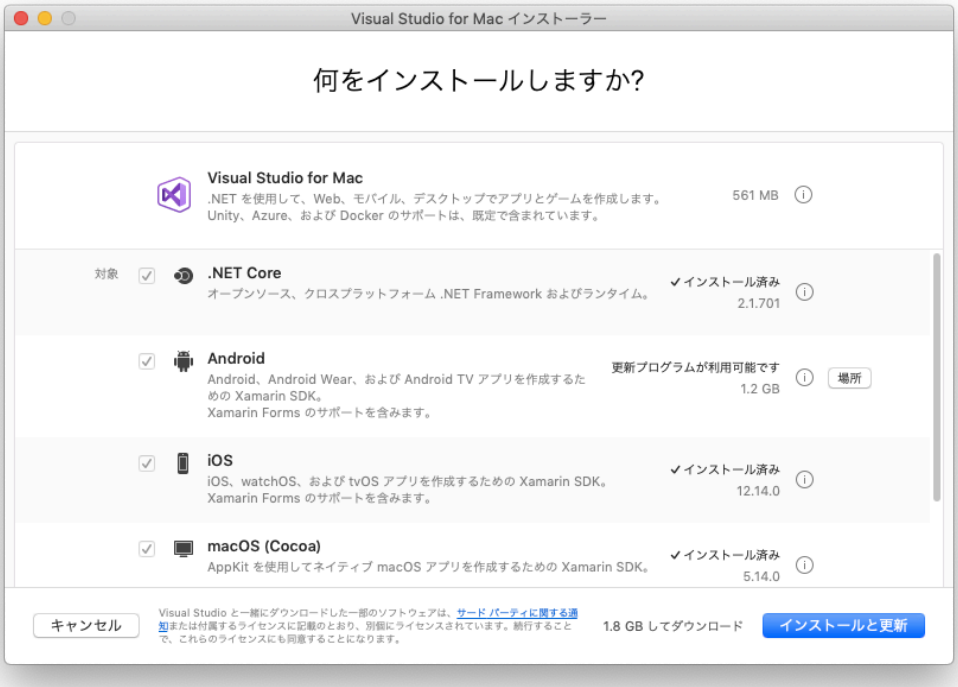
~~正直なところこんな節はなくてもいいんじゃないかなぁなんて思ってたり~~




Visual Studio for Mac Installer
Install Visual Studio for Mac.app
Visual Studio for Mac Installer


“Install Visual Studio for Mac.app”はインターネットからダウンロードされたアプリケーションです。開いてもよろしいですか?
"Install Visual Studio for Mac.app"はディスクイメージ"visualstudioformacinstaller.dmg"内にあります。 このディスクイメージは"Chrome"により今日の6:29にvisualstudio.microsoft.comからダウンロードされました。 Appleによるチェックで悪質なソフトウェアは検出されませんでした。
このディスクイメージ内のアプリケーションを開くときには警告しない
キャンセル
Webページを表示
開く


Visual Studio for Mac インストーラー
Visual Studio for Mac をダウンロードしていただき、ありがとうございます
プライバシーについて詳しくは、Microsoft のプライバシーに関する声明をご 覧ください
続行すると、ライセンス条項 に同意したものと見なされます
終了
続行
キャンセル


Visual Studio for Mac インストーラー
何をインストールしますか?
Visual Studio for Mac
.NET を使用して、Web、モバイル、デスクトップでアプリとゲームを作成します。
Unity、Azure、および Docker のサポートは、既定で含まれています。
561 MB
対象
.NET Core
オープンソース、クロスプラットフォーム .NET Framework およびランタイム。
✓インストール済み
2.1.701
Android
Android、Android Wear、および Android TV アプリを作成するための Xamarin SDK。
Xamarin Forms のサポートを含みます。
更新プログラムが利用可能です
1.2 GB
場所
iOS
iOS、watchOS、および tvOS アプリを作成するための Xamarin SDK。
Xamarin Forms のサポートを含みます。
✓インストール済み
12.14.0
macOS (Cocoa)
AppKit を使用してネイティブ macOS アプリを作成するための Xamarin SDK。
✓インストール済み
5.14.0
キャンセル
Visual Studio と一緒にダウンロードした一部のソフトウェアは、サード パーティに関する通知または付属するライセンスに記載のとおり、別個にライセンスされています。続行することで、これらのライセンスにも同意することになります。
1.8 GB してダウンロード
インストールと更新


### ♣ 1.1.2 端末側での準備

実機で Android アプリの開発を行おうと思うと，開発者向けオプションを選択できるようにしなければいけない．

**端末設定 > システム > 端末情報**と進んで**ビルド番号**を 7 回タップすると，**端末設定 > システム**に開発者向けオプションという項目が追加されるはずだ．

## 1.2 1.2

# 第2章 初めてのAndroidアプリケーション

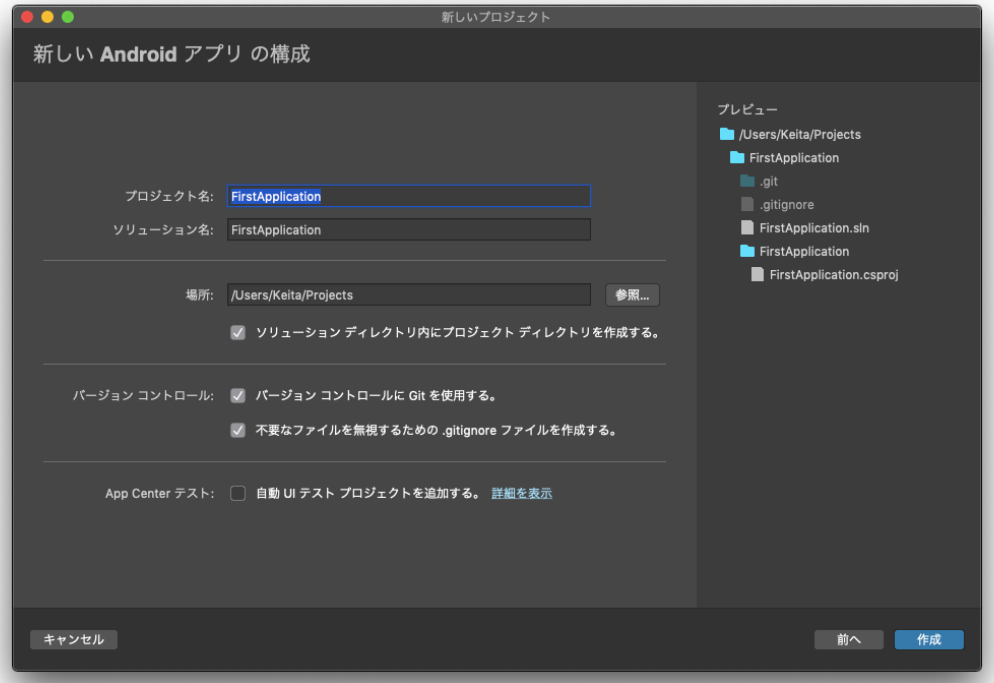
## 2.1 プロジェクトの作成

第 1 章で開発環境の作成を終えたら，早速 Android アプリの作成に挑戦してみたいと思う．


▲図 2.1: 形式の選択


新しいプロジェクト
Android アプリの構成
アプリ名:
FirstApplication
組織の識別子:
com.koron0902.ddns.net
パッケージ名:
com.koron0902.ddns.net.firstapplication
ターゲット プラットフォーム:
最大の互換性
最小: 2.3 "Gingerbread" (API 10)
最新の開発
最小: 4.1 "Jelly Bean" (API 16)
最新および最高
テーマ:
既定
FirstApplic...
キャンセル
前へ
次へ

▲図 2.2: 構成:アプリ




▲図 2.3: 構成:プロジェクト

プロジェクトが出来上がったら，まずは何も考えずに実行してみよう．

## 2.2 出力されたコードを見てみる

勝手に作られたプログラムをじっくりとみていこう．

`Resources/layout/activity_main.xml` というファイルがあるかと思う．これがアプリのレイアウトを決めているファイルだ．

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
</RelativeLayout>
```

それともう一つ，`MainActivity.cs` を見てみよう．

あとで詳しく解説するが，Android アプリには `Activity` という動作単位が存在する，このファイルにはそんな Activity の動作を決める記述がなされている．

```
using Android.App;
using Android.OS;
using Android.Support.V7.App;
using Android.Runtime;
using Android.Widget;

namespace test {
 [Activity(Label = "@string/app_name",
  Theme = "@style/AppTheme",
  MainLauncher = true)]
 public class MainActivity : AppCompatActivity {
  protected override void OnCreate(Bundle savedInstanceState) {
   base.OnCreate(savedInstanceState);
   Xamarin.Essentials.Platform.Init(this, savedInstanceState);
   // Set our view from the "main" layout resource
   SetContentView(Resource.Layout.activity_main);  // (1)
  }

  // (2)
  public override void OnRequestPermissionsResult(int requestCode,
     string[] permissions,
     [GeneratedEnum] Android.Content.PM.Permission[] grantResults) {
   Xamarin.Essentials.Platform.OnRequestPermissionsResult(requestCode,
        permissions,
        grantResults);

   base.OnRequestPermissionsResult(requestCode,
        permissions,
        grantResults);
  }
 }
}
```

**(1)** では，アプリケーションのレイアウトを設定している．

自動生成されたアプリケーションでは `Resources/layout/activity_main.axml` で定義されている．

**(2)** では，Android6 から実装されたいわゆる **M パーミッション**に関する設定を行うためのコールバックである．

今回は特に気にしなくても良い．必要になったら説明したいと思う．

さて，自動生成されたプログラムを実行したところで特に何が表示されるわけでもなく，タイトルと白い画面のみのアプリだったことだろう．

次回以降の章では，このアプリケーションに手を加えてアプリらしいアプリを作るための技術を磨いていこう．

# 第3章 データの入出力

## 3.1 出力するための手段

### ♣ 3.1.1 TextView

画面に情報を表示する際，TextView は有用な手段であると言える．
まずはレイアウトファイルとそれに付随するプログラムを見て欲しい．

```
<TextView
  androud:id="@+id/textview1"
  android:text="これはTextViewです"
/>
```

```
  var tv = FindViewById<TextView>(Resource.Id.textview1);
  tv.text = "テキストを変更しました";
```

### ♣ 3.1.2 Progress Bar

### ♣ 3.1.3 Toast 通知

ボタンを押した時とかに，何かしらの情報が画面下にポップアップ表示されるような経験は誰しもがしたことがあるだろう．その時に現れるポップアップ表示のことを Toast 通知という．
Toast 通知の表示のさせ方を見てみよう

```
  Toast.MakeText(ApplicationContext,
       "Toast通知",
       ToastLength.Short).Show();
```

このように記述するだけでデータをポップアップ表示することができるので，ぜひ知っておいて欲しい．
また，Toast の表示スタイルは様々な形式にカスタムすることができる．

### ♣ 3.1.4 通知バー

## 3.2 入力するための手段

### ♣ 3.2.1 Button

何かしらのアクションのトリガーとして非常に有用な手段であるボタンの実装について解説する．
まずは，サンプルコードを見て欲しい

```
<Button
  android:id="@+id/button1"
  android:text="押してみてね"
/>
```

```
  var button = FindViewById<Button>(Resource.Id.button1);
  button.Click += (sender, e) => {
    Toast.MakeText(ApplicationContext,
         "ボタンが押されたよ",
         ToastLength.Short).Show();
  };
```

### ♣ 3.2.2 Text

### ♣ 3.2.3 CheckBox

### ♣ 3.2.4 SeekBar

### ♣ 3.2.5 Toggle Button

# 第4章 ハードウェアの利用

## 4.1 センサー

### ♣ 4.1.1 加速度センサー

### ♣ 4.1.2 ジャイロセンサー

### ♣ 4.1.3 近接センサー

## 4.2 GPS : Global Positioning System

位置情報を使ったアプリケーションを作ろうと思った場合，GPS の利用は必要不可欠であると言える．そこで，本章では GPS の使用方法について解説したいと思う．

### ♣ 4.2.1 GPS の有効化

## 4.3 カメラ

## 4.4 NFC : Near Field Communication

## 4.5 ストレージの利用

# 第5章 様々なActivity

## 5.1 Activity のライフサイクル

## 5.2 Activity の遷移

### ♣ 5.2.1 データの受け渡しをしない場合

### ♣ 5.2.2 データの受け渡しをする場合

### ♣ 5.2.3 外部のアプリケーションを呼び出す

例えばリンクをクリックした時，勝手にブラウザが立ち上がったり，はたまたどのブラウザを使用するか選択させられたりといった経験はあるのではないだろうか．

このように外部のアプリケーションを立ち上げるのは，一種の Activity 遷移に分類されると筆者は考えている．

そこで，外部のアプリケーションを呼び出す方法をいくつか紹介する．

### ♣ 明示的 Intent

### ♣ 暗黙的 Intent

# 第6章 Background Service

# 第7章 Design Support Libraryを使う

## 7.1 TabbedLayout なアプリケーション

本節では，Tabbed Layout なアプリケーションの作り方を解説していく．

### ♣ 7.1.1 Fragment の実装

### ♣ 7.1.2 Adapter の実装

### ♣ 7.1.3 Adapter と Layout の紐付け

## 7.2 Swipe Refresh Layout

某 SNS とかで，下にスワイプして手を離した時にクルクルしたものが表示されるのを見たことがあるかと思う．
本節ではそのような機能の実装に挑戦してみたいと思う．

# 第8章 第8章 Android Support Libraryを使う

## 8.1 ToolBar

# 第9章 あとがき

# Re:ゼロから始めるアプリ開発

Xamarin ってなぁに？

2019 年 9 月 22 日　ver 1.0

著　者　ここあ
印刷所　○○印刷所



*(powered by Re:VIEW Starter)*