

第 10 章

高速で保守性の高いアプリを開発するためのコツ285

10.1 パフォーマンスと保守性、どちらを優先すべきか286

高速でないアプリとは286

高速だが保守性が低い実装286

10.2 高速で保守性の高い実装287

buildメソッドで高コストな計算をしない287

buildメソッドで大きなウィジェットツリーを構築しない289

ウィジェットツリーの階層が浅くなるようウィジェットの選択を見なおす.. 289

const修飾子を付与する291

const修飾子が見えようウィジェットの選択を見なおす292

独自のウィジェットクラスにconstantコンストラクタを実装する292

状態を末端のウィジェットに移す297

Riverpodの状態監視は末端のウィジェットで行う300

Tips アプリのパフォーマンスを計測する303

10.3 まとめ304

第 11 章

Flutterアプリ開発に必要な
ネイティブの知識305

11.1 ネイティブAPIのバージョンと最低サポートOSのバージョン306

最低サポートOSのバージョン306

iOSの最低サポートOSバージョンを設定する308

Tips XcodeのBuild Settings308

Androidの最低サポートOSバージョンを設定する310

ビルドSDKバージョン310

iOSのビルドSDKバージョンの設定310

AndroidのビルドSDKバージョンの設定310

ターゲットSDKバージョン311

11.2 アプリの設定変更312

アプリ名312

iOSのアプリ名を変更する	312
Androidのアプリ名を変更する	312
アプリアイコン	313
iOSのアプリアイコンを変更する	313
Androidのアプリアイコンを変更する	314
アプリアイコンを手軽に生成するパッケージ	315
スプラッシュ画面	316
iOSとAndroidで異なるスプラッシュ画面の位置付け	316
iOSのスプラッシュ画面	316
Androidのスプラッシュ画面	317
スプラッシュ画面を手軽に実現するパッケージ	317
アプリのID	319
iOSのアプリIDを設定する	319
AndroidのアプリIDを設定する	320
11.3 アプリの配布とコード署名	320
iOSのコード署名	321
管理の難しい秘密鍵	321
アプリに署名する	322
Androidのコード署名	326
apkファイルとaabファイル	327
アプリに署名する	327
aabファイルをアップロードする	330
11.4 まとめ	334
参考Web情報	335
著者プロフィール	335
索引	336

第 1 章

環境構築とアプリの実行

Flutter SDK、Android Studio、Xcode

本章では、Flutterの特徴を簡単に紹介したのち、環境構築の手順を解説します。Flutter公式がアナウンスしている環境構築の手順に加え、筆者が普段から利用しているツールなども紹介します。最後には、テンプレートのFlutterアプリを起動するところまでを実践します。

1.1

なぜFlutterが注目を集めているのか

FlutterはGoogleを中心としたオープンソースコミュニティによって開発されているマルチプラットフォームフレームワークです^{注1}。これまでのマルチプラットフォーム技術とは一線を画した実現方式を持ち、高速な実行速度と優れた開発者体験が特徴です。このFlutterがスマートフォンアプリの開発において、シェアを伸ばしています。

マルチプラットフォーム

Flutterは1つのコードベースからさまざまなプラットフォームへアプリケーションを提供することができます。そのサポート環境は、公式発表でモバイル(iOSおよびAndroid)、デスクトップ(Windows、macOSおよびLinux)、Webフロントエンドと多岐にわたります。

また、複数のプラットフォームで安定した動作を実現する構成になっている点も特徴です。マルチプラットフォーム技術の中にはXamarin^{注2}のようにネイティブUI(*User Interface*)をラップした形式のものが 있습니다。こうしたフレームワークでは、複数プラットフォームのAPIをラップする過程で、吸収しきれない差異が発生するケースがあります。一方でFlutterはネイティブのUIを使わない、独自のレンダリングのしくみを持っています。そのため「iOSでデバッグしたあと、Androidでデバッグすると想定したUIになっていない」といった動作差異が限りなく小さくなります。

なお、本書ではiOSとAndroidを対象としたモバイル開発についてのみ取り扱います。

注1 <https://flutter.dev/>

注2 <https://docs.microsoft.com/ja-jp/xamarin/>

高速な実行速度

Flutterで開発したアプリケーションはネイティブコードにコンパイルされます。Cordova^{注3}などのようにWebView上で動作するマルチプラットフォームフレームワークよりも、高速に動作することが期待できます。開発元のGoogleは、前項でも触れた独自のレンダリングのしくみによる高いパフォーマンスをアピールしています。筆者の体感としても、他のいくつかのマルチプラットフォームフレームワークと比較して、Flutterは安定して高いパフォーマンスを発揮しているように感じます。

優れた開発者体験

フレームワークの浸透、発展において優れた開発者体験は重要であり、Flutterはその要素を十分に持っています。筆頭に挙げられるのはホットリロードで、ソースコードの変更を実行中のアプリに即座に反映するしくみです。プログラムでUIを微調整したらすぐにアプリの画面に反映されるため、トライ&エラーをすばやく繰り返すことが可能です。また、コードラボ(チュートリアル)^{注4}やクックブック(実装例)^{注5}が充実しています。パッケージ管理ツールやコードの静的解析ツールなどが統合されている点も開発者にとってうれしいポイントの一つです。

1.2

Flutterの環境構築

それではFlutterの開発環境を構築しましょう。Flutterの環境構築は大きく2つの段階があります。

1つ目はFlutter SDK(*Software Development Kit*)のインストール、2つ目は実行するプラットフォームであるiOSとAndroidの開発環境のインストールです。

なお、iOSを開発対象にすることもあり、本書ではmacOS環境についての

注3 <https://cordova.apache.org>

注4 <https://docs.flutter.dev/codelabs>

注5 <https://docs.flutter.dev/cookbook>

み取り扱います。執筆時の環境は以下です。

- macOS Sonoma 14.3
- Flutter 3.16.9

Flutterのインストール

本項のいくつかの手順は、後述の「1.3 fvm による Flutter のバージョン管理」の節で解説する fvm というツールを導入することで省略できます。しかし、はじめての環境構築は内容を理解するためにも、手順を踏んでインストールを試みることをお勧めします。

まずは、Flutter の公式 Web サイトから Flutter SDK の ZIP ファイルをダウンロードします。URL の参照先は iOS の環境構築に関するページですが、Flutter SDK の ZIP ファイルは他のプラットフォームと共通です。

- <https://docs.flutter.dev/get-started/install/macos/mobile-ios?tab=download#install-the-flutter-sdk>

ZIP ファイルは Intel CPU 向けと Apple Silicon 向けが用意されていますので、お使いの環境に合わせてダウンロードしてください。

Apple Silicon 搭載の Mac をお使いの場合は、加えて Rosetta 2 をインストールしましょう。SDK に含まれる一部の実行ファイルが x64 アーキテクチャで提供されており、これらを Apple Silicon 搭載の Mac で動作させるために Rosetta 2 が必要になります。Rosetta 2 は以下のコマンドでインストールします。

```
# Rosetta 2 をインストール
$ sudo softwareupdate --install-rosetta --agree-to-license
```

ダウンロードした ZIP ファイルを解凍し、任意のディレクトリに配置します。今回は公式 Web サイトで紹介されているディレクトリ (~/.development) で進めます。

```
~/development
├── flutter
│   ├── bin
│   ├── dev
│   ├── examples
│   ├── packages
│   └── ... (省略)
```

次に、配置した Flutter SDK にパスを通します。以下のコマンドを実行し、

SDKに^{どうこん}同梱されている実行ファイルをどのディレクトリからでも実行できるようにします。

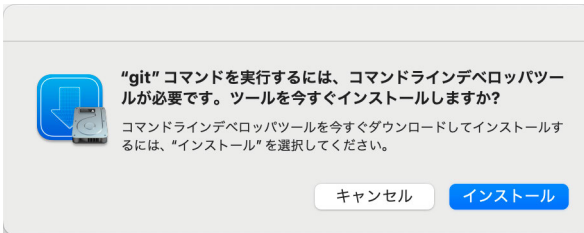
```
# flutterのコマンドラインツールにパスを通す
$ echo "export PATH=\"$PATH:$HOME/development/flutter/bin\"" >> ~/.zshenv
# 実行中のシェルにパスを適用
$ . ~/.zshenv
```

これでFlutterのインストールは完了です。確認のため、Flutterのバージョンをターミナルに出力してみましょう。

```
# flutterのバージョンを出力
$ flutter --version
```

環境にgitがインストールされていない場合は、デベロッパツールのインストールを促すダイアログが表示されます(図1.1)。これはFlutterのコマンドラインツールが、内部でgitコマンドを呼び出しているためです。インストールしておきましょう。

図1.1 デベロッパツールのインストールダイアログ



筆者の環境ではflutter --versionの結果は以下のよう出力されました。

```
Flutter 3.16.9 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 41456452f2 (32 hours ago) • 2024-01-25 10:06:23 -0800
Engine • revision f40e976bed
Tools • Dart 3.2.6 • DevTools 2.28.5
```

プラットフォームごとの環境のインストール

Flutterで開発したプロジェクトをiOSやAndroidへビルドするために、各プラットフォームの開発環境を構築する必要があります。開発環境の情報を出力するコマンド(flutter doctor)を活用しながら各プラットフォームの開発環境をインストールしていきましょう。

```
# flutterの開発環境の情報を出力
$ flutter doctor
```

図 1.2 flutter doctor の出力結果

```
[✓] Flutter (Channel stable, 3.16.9, on macOS 14.3 23D56 darwin-arm64, locale ja-JP) ①
[X] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/docs/get-started/install/macos#android-setup for
      detailed instructions).
      If the Android SDK has been installed to a custom location, please use
      `flutter config --android-sdk` to update to that location.

[X] Xcode - develop for iOS and macOS ②
    X Xcode installation is incomplete; a full installation is necessary for iOS and
    macOS development.
      Download at: https://developer.apple.com/xcode/
      Or install Xcode via the App Store.
      Once installed, run:
        sudo xcode-select --switch /Applications/Xcode.app/Contents/Developer
        sudo xcodebuild -runFirstLaunch
    X CocoaPods not installed.
      CocoaPods is used to retrieve the iOS and macOS platform side's plugin code
      that responds to your
      plugin usage on the Dart side.
      Without CocoaPods, plugins will not work on iOS or macOS.
      For more info, see https://flutter.dev/platform-plugins
      To install see https://guides.cocoapods.org/using/getting-started.html#install
      ation for instructions.

[X] Chrome - develop for the web (Cannot find Chrome executable at /Applications/Google
    Chrome.app/Contents/MacOS/Google Chrome)
    ! Cannot find Chrome. Try setting CHROME_EXECUTABLE to a Chrome executable.

[!] Android Studio (not installed) ③
[X] Connected device (1 available)
[X] Network resources
```

コマンドの出力結果(図 1.2)の①と③が Android の開発環境、②が iOS の開発環境に関わる項目です。

Androidの開発環境をインストールする

あらためて flutter doctor の出力結果から、Android の開発環境に関わる項目を確認します。Android の開発環境がまだ構築されていない場合は、以下