Add Flutter to existing app

Contents

Add-to-app

It's sometimes not practical to rewrite your entire application in Flutter all at once. For those situations, Flutter can be integrated i your existing application piecemeal, as a library or module. That module can then be imported into your Android or iOS (currently supported platforms) app to render a part of your app's UI in Flutter. Or, just to run shared Dart logic.

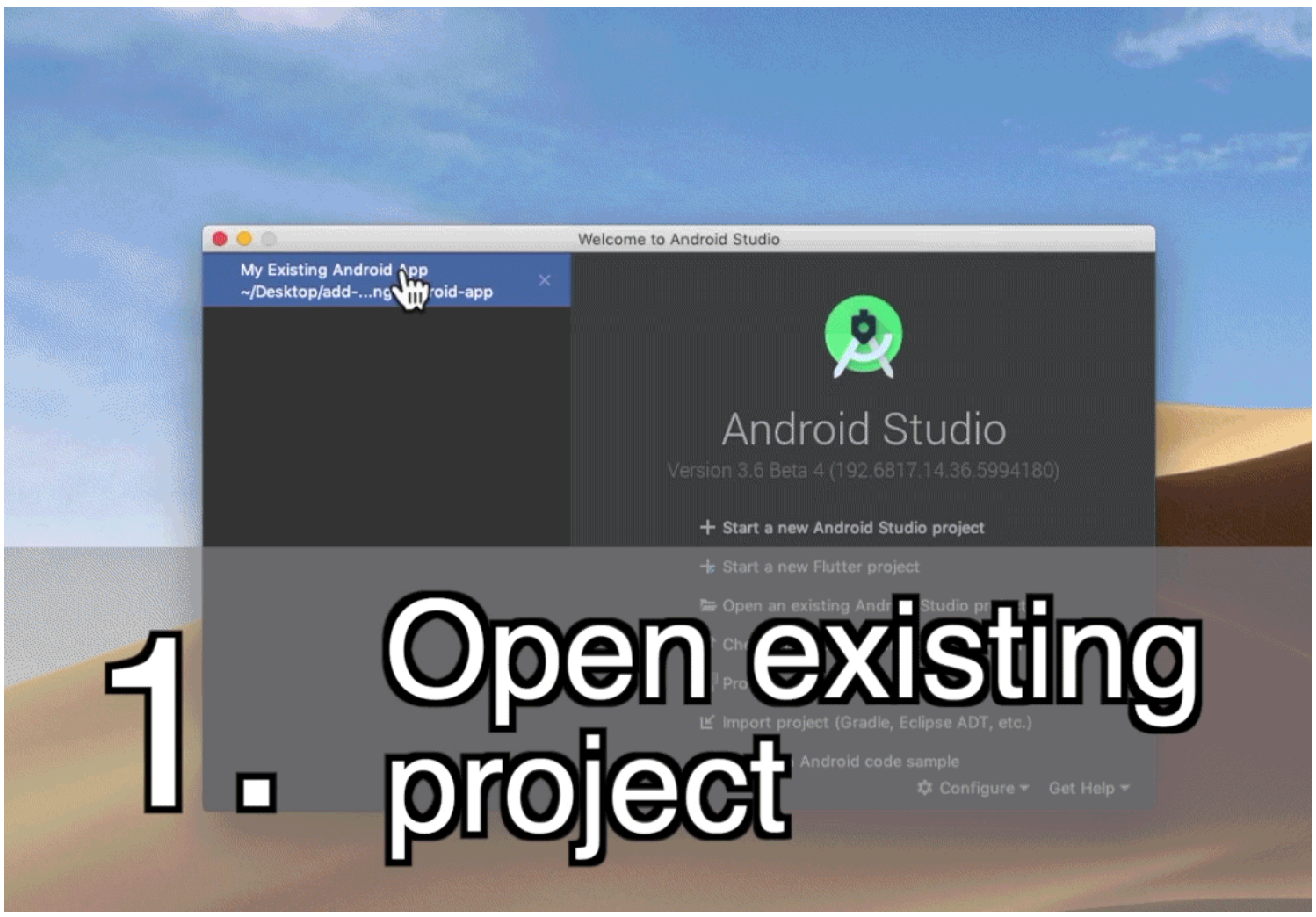
In a few steps, you can bring the productivity and the expressiveness of Flutter into your own app.

As of Flutter v1.12, add-to-app is supported for the basic scenario of integrating one full-screen Flutter instance at a time per app currently has the ***following limitations***:

- Running multiple Flutter instances or running in partial screen views may have undefined behavior.
- Using Flutter in background mode is still a WIP.
- Packing a Flutter library into another sharable library or packing multiple Flutter libraries into an application isn't supported.
- Plugins used in add-to-app on Android should undergo flutter.dev/go/android-plugin-migration and use the FlutterPlugin-bas APIs. Plugins that don't support FlutterPlugin may have unexpected behaviors if they make assumptions that are untenable add-to-app (such as assuming that a Flutter Activity is always present).

Supported features

Add to Android applications



- Auto-build and import the Flutter module by adding a Flutter SDK hook to your Gradle script.
- Build your Flutter module into a generic Android Archive (AAR) for integration into your own build system and for better Jeti interoperability with AndroidX.

- `FlutterEngine` API for starting and persisting your Flutter environment independently of attaching a `FlutterActivity`/`FlutterFragment` etc.
- Android Studio Android/Flutter co-editing and module creation/import wizard.
- Java and Kotlin host apps are supported.
- Flutter modules can use Flutter plugins to interact with the platform. Android plugins should be migrated to the V2 plugins A for best add-to-app correctness. As of Flutter v1.12, most of the plugins maintained by the Flutter team as well as FlutterFir have been migrated.
- Support for Flutter debugging and stateful hot reload by using `flutter attach` from IDEs or the command line to connect t app that contains Flutter.

Add to iOS applications

- Auto-build and import the Flutter module by adding a Flutter SDK hook to your CocoaPods and to your Xcode build phase.
- Build your Flutter module into a generic iOS Framework for integration into your own build system.
- `FlutterEngine` API for starting and persisting your Flutter environment independently of attaching a `FlutterViewControll`
- Objective-C and Swift host apps supported.
- Flutter modules can use Flutter plugins to interact with the platform.
- Support for Flutter debugging and stateful hot reload by using `flutter attach` from IDEs or the command line to connect t app that contains Flutter.

See our add-to-app GitHub Samples repository for sample projects in Android and iOS that import a Flutter module for UI.

Get started

To get started, see our project integration guide for

Android | iOS

API usage

After Flutter is integrated into your project, see our API usage guides for

Android | iOS