Introduction to Flutter

And cross-platform mobile development

Burkina Institute of Technology (BIT)

Instructor: W.E. Kevin Yanogo

Outline

1. Native vs Cross-Platform Development
2. What is Flutter ?
3. Flutter vs other Cross-Platform Frameworks (React Native, Xamarin…)
4. Introduction to Dart
5. Useful Flutter and Dart Resources
6. Getting Started with Android Studio
7. Flutter “Hello World” Application
8. Next Steps: Building our first Flutter App (Flutter News)
9. Q&A

About the Instructor

- Full Name: W.E. Kevin Yanogo
- Background:
  - Computer Science (B.S.c) National Dong Hwa University
  - Computer Science (M.S) The University of Texas at Arlington
- Interests:
  - Mobile Development: Android(Java), Flutter
  - Web Development: React JS, Html5/CSS3, Php, …
  - Games Development: Unity3D(webGL)
  - Cloud Data: GCP, AWS (current work)
  - ML/AI: Tensorflow, Microsoft AirSim, …
- Notable Works:
  - Google Play Apps: UniTutor, CSIEB0020, UList
  - KevappsGaming: http://kevappsgaming.com/
  - Research Work: Algorithms, SERC Lab (UT Arlington)
- Contacts:
  - GitHub: https://github.com/yankev07
  - LinkedIn: https://www.linkedin.com/in/kevinyann/
  - Email: kevin.yanogo@gmail.com
  - Website: https://yankev07.github.io/

Native vs Cross-Platform Development

Building a mobile app exclusively for a single platform

Native App Development

Pros

- Broad Functionality
- Better Store Support

Increased Scalability
High Performance And Great UX

Cons

- Costly
- Time Consuming

Creating an app that works on several platforms

Cross-Platform App Development

Pros

- Less Costly
- Faster Development
- Single Code Base

Cons

- Slower App
- Limited Functionality
- Limited UX

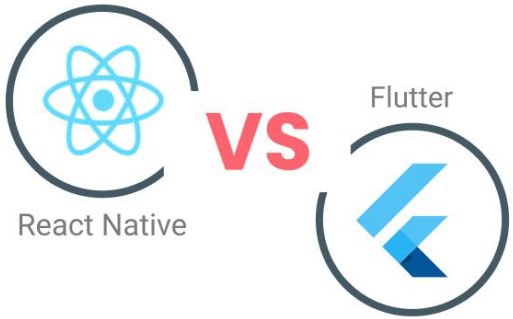
What is Flutter ?

- Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.
- Flutter is free and open source.
- Flutter applications can be deployed to multiple devices: mobile, web, desktop and embedded devices.
- Flutter is approachable to programmers familiar with object-oriented concepts (classes, methods, variables, etc) and imperative programming concepts (loops, conditionals, etc).
- Flutter is built with C, C++, Dart, and Skia (a 2D rendering engine)
- Exporting Flutter App to Android: The engine's C and C++ code are compiled with Android's NDK. The Dart code (both the SDK's and yours) are ahead-of-time (AOT) compiled into native, ARM, and x86 libraries. Those libraries are included in a "runner" Android project, and the whole thing is built into an .apk.
- Exporting Flutter App to IOS: The engine's C and C++ code are compiled with LLVM. The Dart code (both the SDK's and yours) are ahead-of-time (AOT) compiled into a native, ARM library. That library is included in a "runner" iOS project, and the whole thing is built into an .ipa.

Flutter vs other Cross-Platform Frameworks (React Native, Xamarin…)

React Native is an open-source mobile application framework created by Facebook, that uses JavaScript. It is therefore a direct competitor of Flutter.

Let’s compare both frameworks, and analyze the pros and cons of each.



Flutter vs other Cross-Platform Frameworks (React Native, Xamarin…)

Flutter

- Hot Reload (milliseconds between refresh)
- One codebase, for 2 mobile platforms and more (Flutter for Web, etc …)
- Up to 50% less testing (QA Testing)
- Faster apps (Skia GL and 60fps)
- Beautiful Widgets (Lovely Design)
- Same app UI, even on older devices
- Perfect for MVPs (Minimum Viable Products)
- Size of the developer community (& Dart)
- Libraries & support – impressive, but still not as rich as native development
- Continuous Integration support (ex: Jenkins)
- Platform risk (ex: ARCore/Sceneform)
- App's size

React Native

- Fast refresh (inject code into a running app)
- One codebase, 2 mobile platforms and more
- It uses a wildly popular language: JavaScript
- Developer freedom of choice (ex: libraries)
- Relative maturity (released in 2015)
- An active and vast community
- Easy to learn for React developers
- Up to 50% less testing
- It isn't really Native (UI/Performance- wise)
- Fewer components out of box (third parties mostly)
- Developer freedom of choice (could be a disadvantage)
- Lots of abandoned packages
- Fragile UI (UI inconsistency after OS updates)
- Apps are bigger than native ones

When to choose Flutter ?

In reality, customers, don’t really care whether an app works on React Native or Flutter.

Instead, they highly value the user experience that the mobile product provides. For example, do transitions between screens and other animations feel fluid? Does the app load fast? Can we use familiar gestures to navigate around the app?

Combined, these small things can easily make the audience fall in love with a mobile product or prompt us to remove it. Choose Flutter when:

- When you have talented Dart developers on your team.
- When you expect your application to include many complex animations and calculation-intensive algorithms (heavy load on CPU/GPU)
- When the UI will need a lot of detailed customization
- When there’s a chance the app will go beyond mobile, or if the next app will benefit from the experience of the current team (reusability)
- When the App is fit for development on wearable and embedded devices, PCs, and vehicle infotainment systems

Introduction to Dart

Dart is a programming language designed for client (UI) development, such as for the web and mobile apps.

It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C-style syntax.

For loops

You can iterate with the standard `for` loop. For example:

```
var message = StringBuffer('Dart is fun');
for (var i = 0; i < 5; i++) {
  message.write('!');
}
```

Useful Flutter and Dart Resources

- ❑ Dart Codelabs (Intro to Dart for Java Developers)
- ❑ Dart Tutorials
- ❑ Dart Language Tour
- ❑ Dart Textbook
- ❑ Dart Online Editor DartPad
- ❑ Dart Packages Pub.Dev
- ❑ Dart DevTools
- ❑ Flutter Tutorials
- ❑ Flutter Samples
- ❑ Flutter UI Builder FlutterFlow

Getting Started with Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on the IntelliJ IDEA by Jet Brains

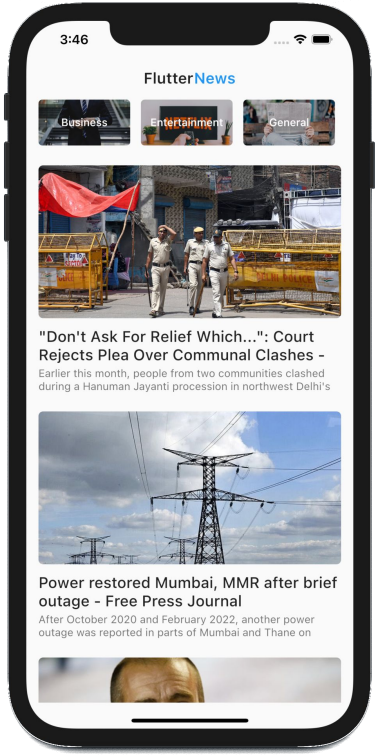
We will use Android Studio as our main IDE for this course (more precisely Android Studio Arctic Fox 2020.3.1 released on July 28, 2021)

Flutter “Hello World” Application

- Setting up Flutter with Android Studio
- Writing our First Flutter App

Next Steps: Building our first Flutter App (Flutter News)

News API: https://newsapi.org/



Q&A