GDG Indy

Flutter Study Jam

Our Sponsors:

Google

zWORKS

Agenda:

6:30p - 6:45p	Network/Food
6:45p - 7:00p	Flutter Overview
7:00p - 8:00p	Flutter Setup

Slides adopted from Martin Aguinis's 'Intro to Flutter' and Introduction to Flutter event by Codemate Ltd.

Flutter

Build beautiful native apps in record time

Ibrahim Sardar

Organizer
Recent Graduate from IUPUI

Jayden Thrasher

Co-organizer
VP of Computer Science Club at IUPUI

What is Flutter?

- SDK for building cross-platform mobile apps, built by Google
- Targets Android, iOS and Fuchsia
- Consistent UIs across devices and manufacturers
- Superb performance
- Interfaced with the Dart Programming Language

Challenges of mobile development today

“To the metal” approaches

- ✓ **High-quality apps**
  Platform and system integrations
- ✓ **High-performance UIs**
  Native code, GPU accelerated
- ✗ **Must fund two apps**
  Two teams, codebases, & investments
- ✗ **Inconsistent brand, features**
  Different across devices & OEMs

“Cross platform” approaches

- ✓ **Fast development**
  Quick iterations, hot reload
- ✓ **Portability, reach**
  Single codebase
- ✗ **Poor Performance**
  Slow, jerky, unpredictable
- ✗ **Non-Native Look/Feel**
  Users can tell the difference

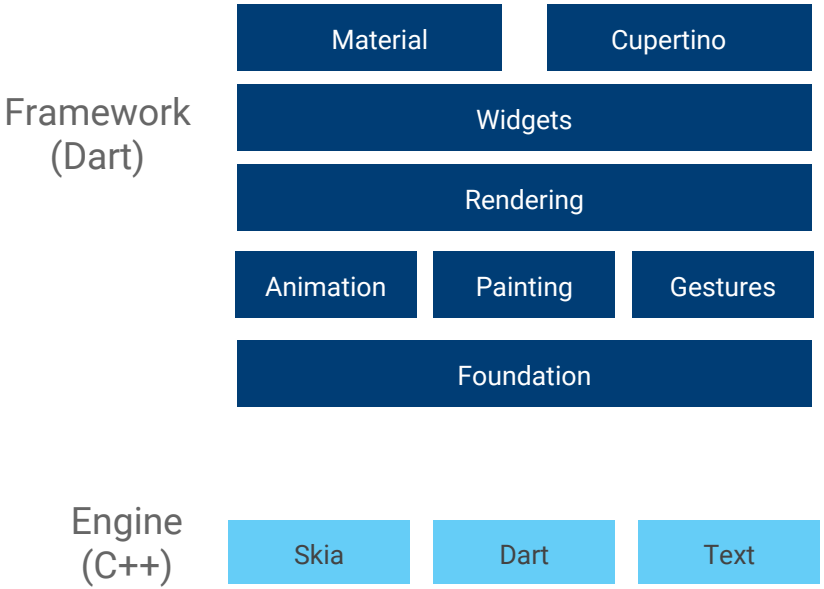
Flutter offers the best of both worlds

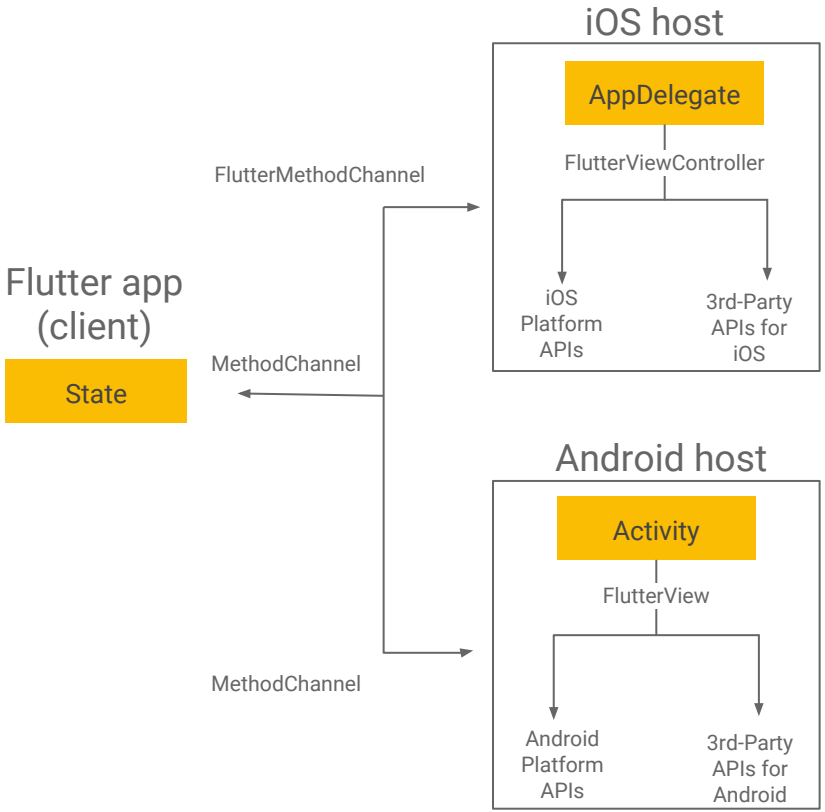
High-Velocity Development

Expressive and Flexible Toolkit

Native iOS and Android App

Flutter is a modern UI toolkit for native apps





High-velocity development

Sub-second reload times

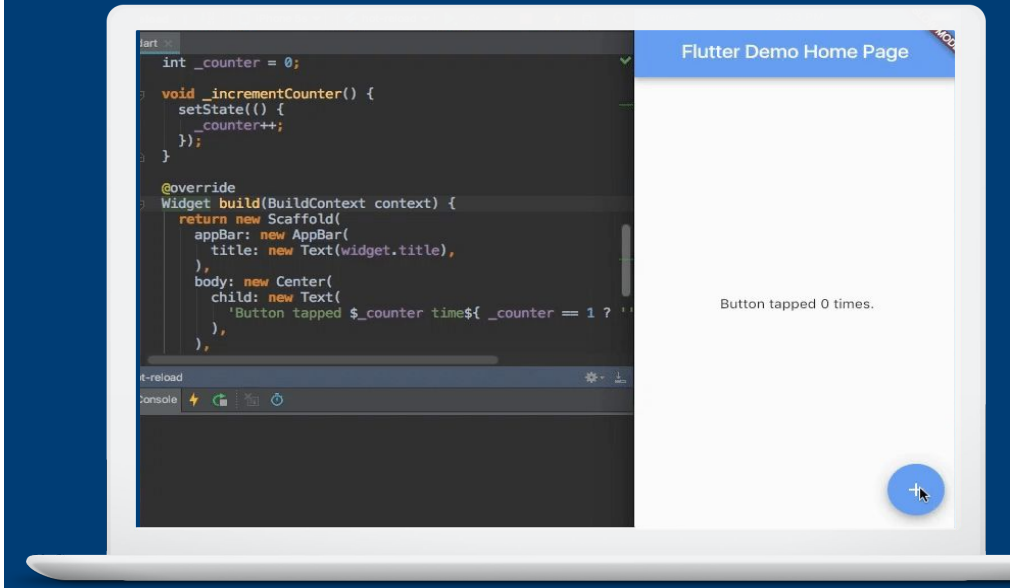
Paint your app to life

Iterate rapidly on features

Test hypotheses quicker than ever

More time to experiment & test features

Single-codebase for faster collab



Flexibility and Control for beautiful UI's

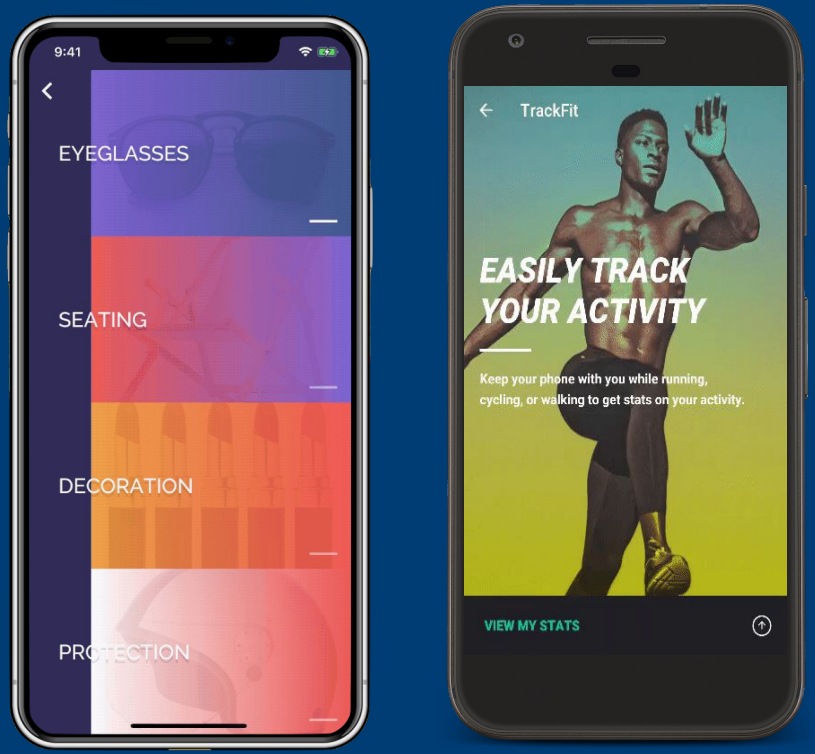
Control every pixel on the screen

Make your brand come to life

Never say "no" to your designer

Stand out in the marketplace

Win awards with beautiful UI



Native Apps for iOS and Android

Compiles directly to native ARM code Does not use a JavaScript bridge

60fps, GPU accelerated

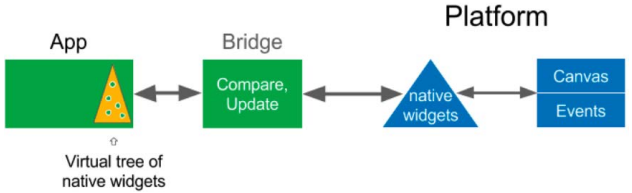
Smooth animations

Deep platform integrations

Natural look and feel

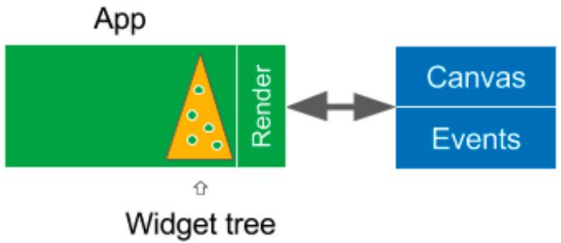
Critical platform differences (scrolling, navigation, fonts)

(React Native)



VS

(Flutter)



Works with popular tools and platforms

Android Studio

Xcode

VS Code

Firebase

3rd-party Android SDKs

Android APIs

iOS APIs

Material Design

Redux

3rd-party iOS SDKs

Rich ecosystem and community

- 1,100+ other packages in repository
- 24,000+ Github stars
- 500+ apps in Play Store
- startflutter.com, flutter.rocks, flutter.institute, and more
- Open source (250+ contributors), BSD license

Four ways to use Flutter today

Start a new app from scratch

Build your new idea in Flutter, and reach both iOS and Android at the same time.

Prototype a new app idea

Use Flutter to test out an app concept or idea in record time.

Bring your app to the the *other* platform

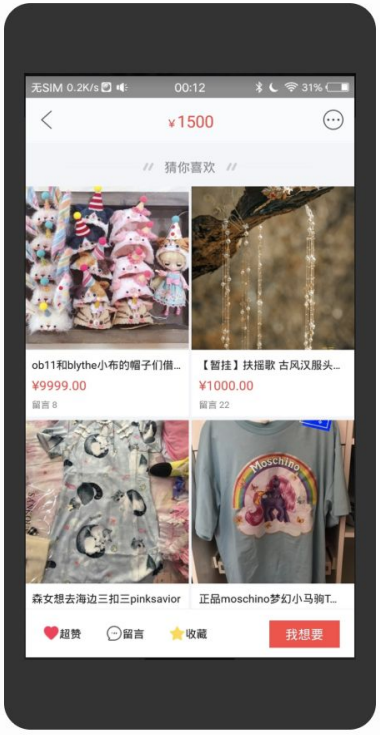
You already have an iOS or Android app? Use Flutter to build for the other platform. Combine codebases when you've proven your Flutter app.

Use Flutter for a part of your app

Test Flutter in production with one or two screens in your existing app.

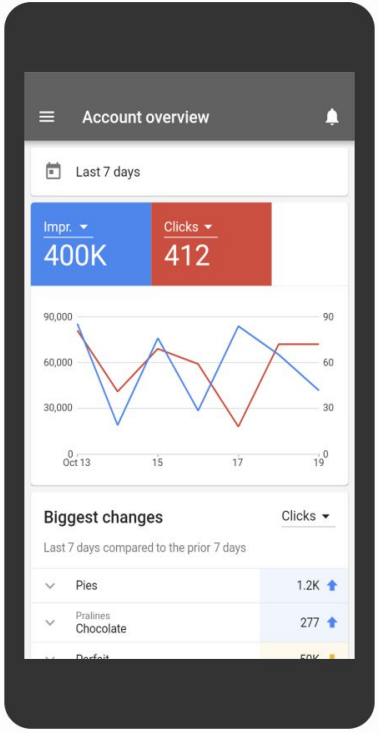
Built and used by Google, developers and companies around the world

Alibaba



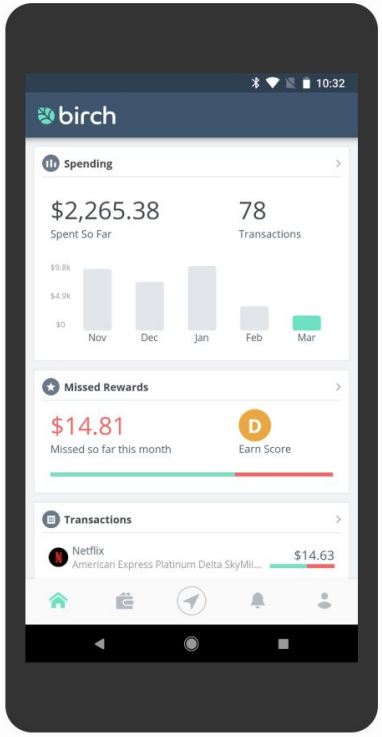
Alibaba's app incorporates Flutter to power parts of their app.

Google AdWords



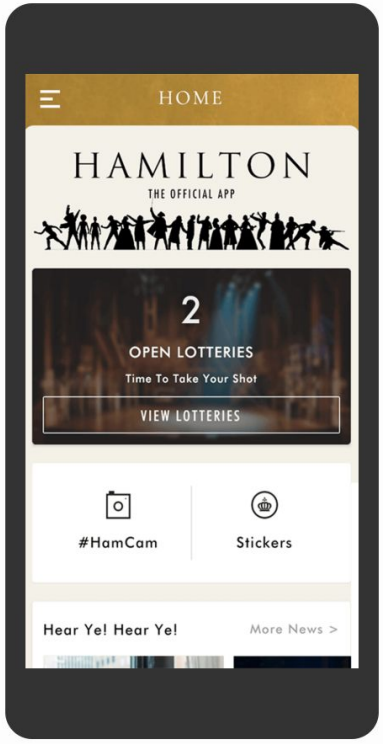
The Google AdWords app helps you keep your ad campaigns running smoothly — no matter where your business takes you.

Birch Finance



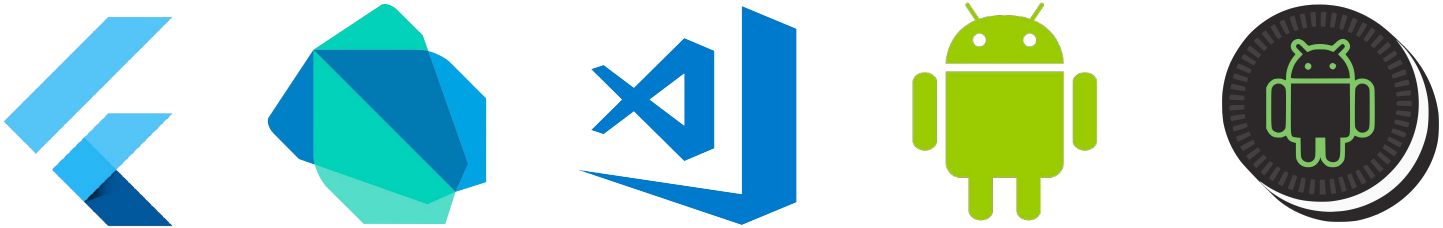
Credit card rewards app to manage and optimize your existing cards.

Hamilton Musical

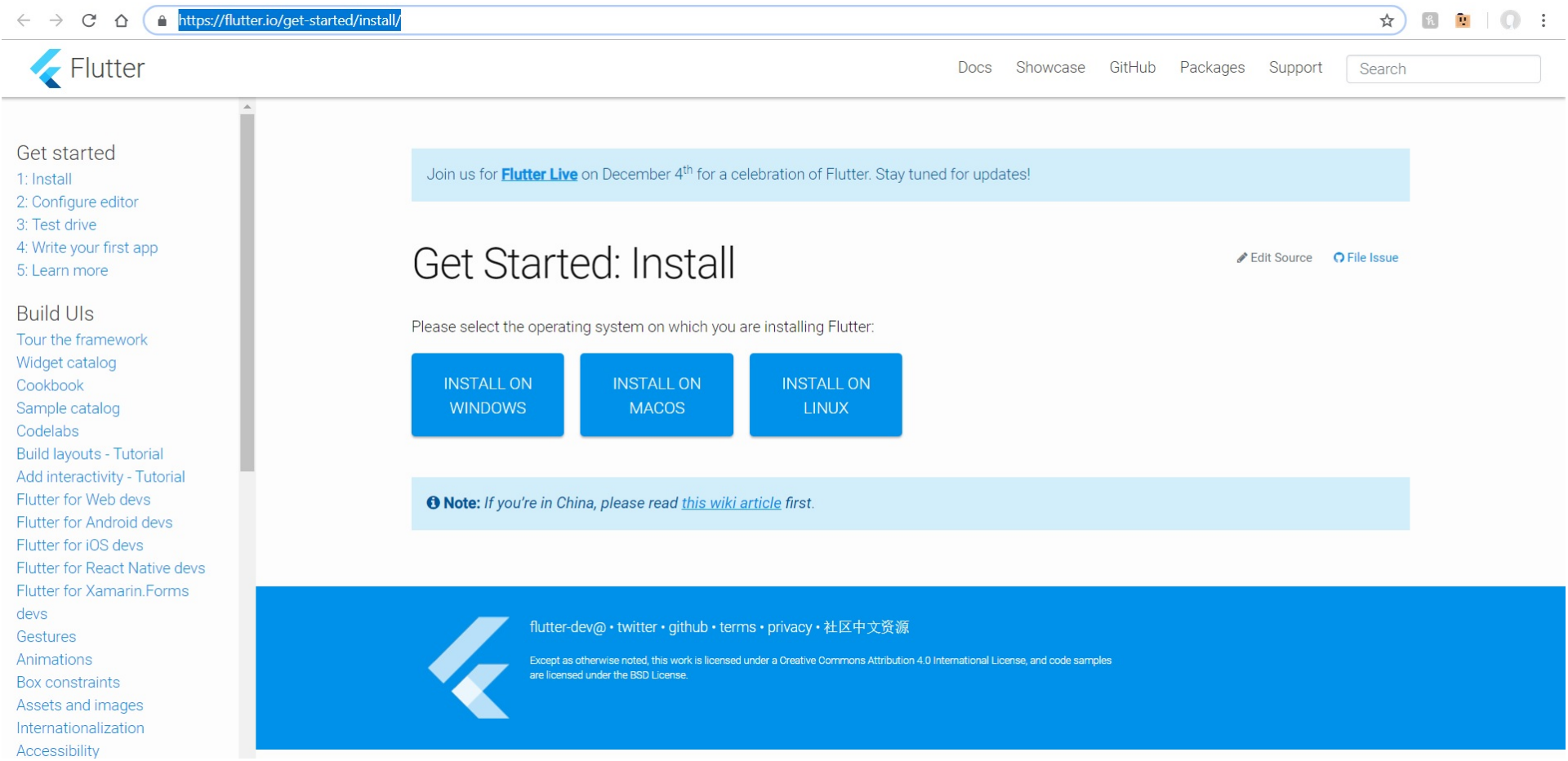


Official app of the hit Broadway musical, Hamilton. Includes daily lotteries, exclusive news and videos, a trivia game, merchandise store, and more.

Setting Up Flutter



Using Flutter, Dart, VS Code, Android, and Android Emulator API 26 Oreo



https://flutter.io/get-started/install/

Get Started: Install on Windows

System requirements

To install and run Flutter, your development environment must meet these minimum requirements:

- **Operating Systems**: Windows 7 SP1 or later (64-bit)
- **Disk Space**: 400 MB (does not include disk space for IDE/tools).
- **Tools**: Flutter depends on these tools being available in your environment.
  - PowerShell 5.0 or newer
  - Git for Windows (with the "Use Git from the Windows Command Prompt" option)

    If Git for Windows is already installed, make sure you can run `git` commands from the Command Prompt or PowerShell.

Get the Flutter SDK

1. Download the following installation bundle to get the latest beta release of the Flutter SDK (for other release channels, and older builds, see the SDK archive page.):
   - flutter_windows_v0.9.4-beta.zip
2. Extract the zip-file and place the contained `flutter` in the desired installation location for the Flutter SDK (eg. `C:\src\flutter`; do not install flutter in a directory like `C:\Program Files\` that requires elevated privileges).

https://flutter.io/setup-windows/

Android setup

Note: Flutter relies on a full installation of Android Studio to supply its Android platform dependencies. However, you can write your Flutter apps in a number of editors; a later step will discuss that.

Install Android Studio

1. Download and install Android Studio.
2. Start Android Studio, and go through the 'Android Studio Setup Wizard'. This will install the latest Android SDK, Android SDK Platform-Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.

Set up your Android device

To prepare to run and test your Flutter app on an Android device, you'll need an Android device running Android 4.1 (API level 16) or higher.

1. Enable **Developer options** and **USB debugging** on your device. Detailed instructions are available in the Android documentation.
2. Windows-only: Install the Google USB Driver
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the `flutter devices` command to verify that Flutter recognizes your connected Android device.

By default, Flutter uses the version of the Android SDK where your `adb` tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the `ANDROID_HOME` environment variable to that installation directory.

Set up the Android emulator

To prepare to run and test your Flutter app on the Android emulator, follow these steps:

1. Enable VM acceleration on your machine.
2. Launch **Android Studio>Tools>Android>AVD Manager** and select **Create Virtual Device**. (The **Android** submenu is only present when inside an Android project.)
3. Choose a device definition and select **Next**.
4. Select one or more system images for the Android versions you want to emulate, and select **Next**. An *x86* or *x86_64* image is recommended.

The Android setup takes very long if you havn't got android dependencies and an emulator set up already

Get Started: Configure Editor

Edit Source File Issue

You can build apps with Flutter using any text editor combined with our command-line tools. However, we recommend using one of our editor plugins for an even better experience. With our editor plugins, you get code completion, syntax highlighting, widget editing assists, run & debug support, and more.

Follow the steps below to add an editor plugin for Android Studio, IntelliJ, or VS Code. If you want to use a different editor, that's OK, simply skip ahead to Next step: Create and run your first app.

Android Studio | VS Code

Visual Studio Code (VS Code) setup

VS Code: A light-weight editor with Flutter run and debug support.

Install VS Code

- VS Code, latest stable version.

Install the Flutter plugin

1. Start VS Code
2. Invoke **View>Command Palette...**
3. Type 'install', and select the **'Extensions: Install Extension'** action
4. Enter `flutter` in the search field, select 'Flutter' in the list, and click **Install**
5. Select 'OK' to reload VS Code

Validate your setup with the Flutter Doctor

1. Invoke **View>Command Palette...**
2. Type 'doctor', and select the **'Flutter: Run Flutter Doctor'** action
3. Review the output in the 'OUTPUT' pane for any issues

Next step

Let's take Flutter for a test drive: create a first project, run it, and experience 'hot reload'.

Next step: Test drive Flutter

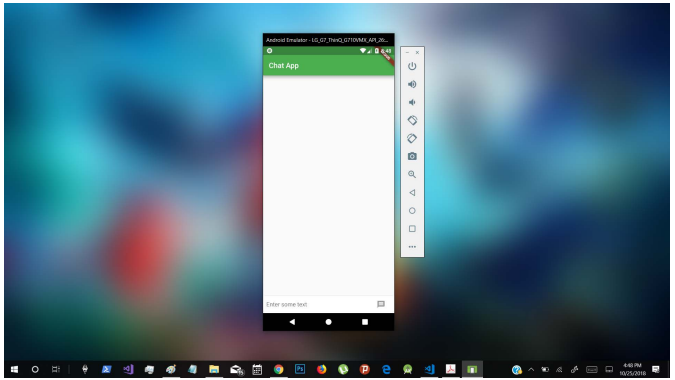
https://flutter.io/get-started/editor/

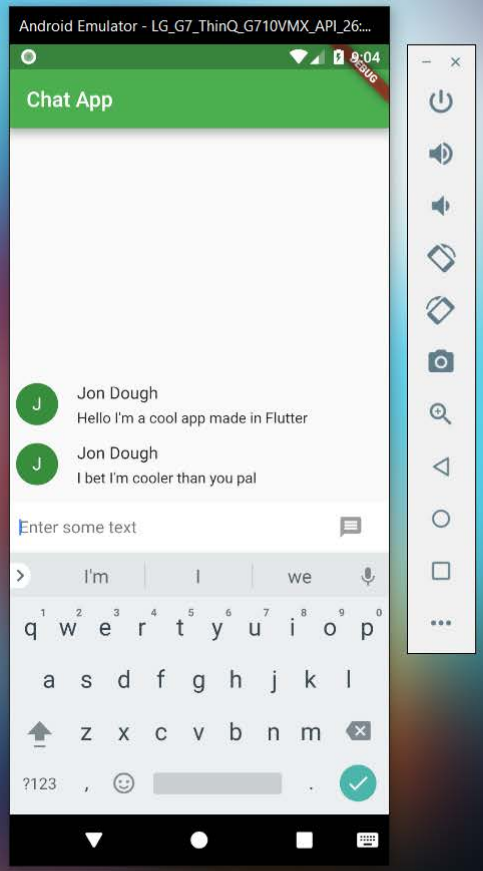
Navigate to the Github repo:
https://github.com/Ibsardar/indyGDG-FlutterStudyJam-Demo

Create a new project
(VS Code: CTRL+SHIFT+P, "Flutter: New Project")

Replace "main.dart" with the main.dart under the "lib" folder in your newly created project

Run the project (F5 in VS Code)
and it should look like this:




Android Emulator - LG_G7_ThinQ_G710VMX_API_26:...
9:04
DEBUG
Chat App
J
Jon Dough
Hello I'm a cool app made in Flutter
J
Jon Dough
I bet I'm cooler than you pal
Enter some text
I'm
I
we
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
,
.

Thank you!

flutter.io

Confidential + Proprietary