Flutter

How to make a beautiful app in an hour?

Paulina Szklarska

Android Developer
Flutter Developer
Toast Wrocław (Android) meetups co-org
GDG Wrocław co-org
Women Techmakers Wrocław lead
cat owner
travel enthusiast


@pszklarska


@pszklarska


@p_szklarska

Agenda

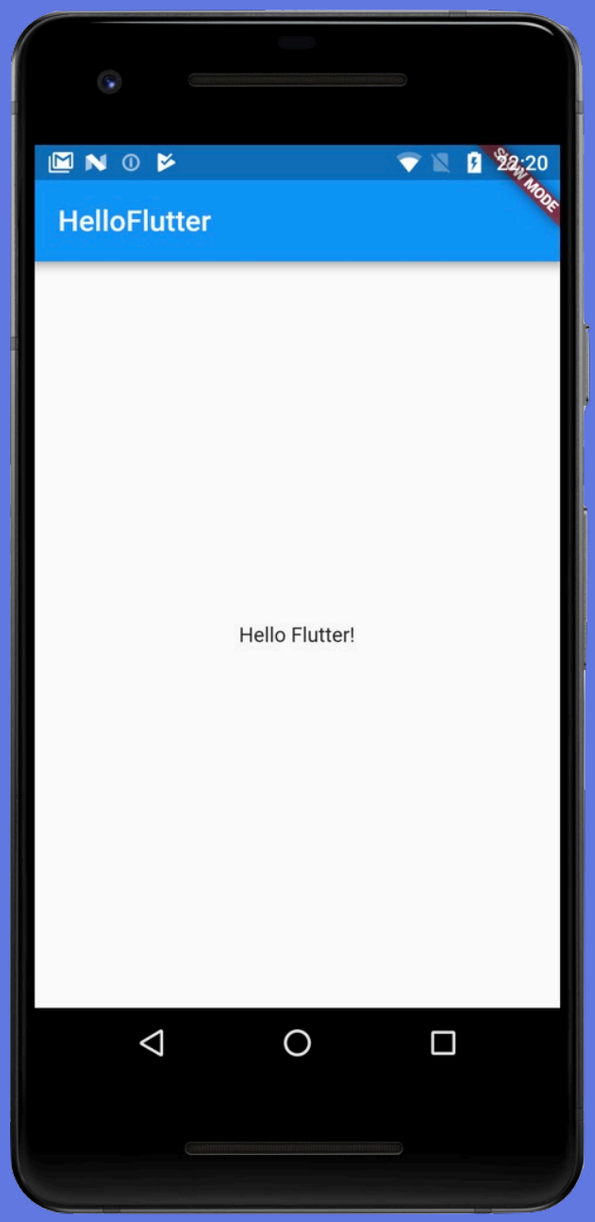
1. What is Flutter?
2. How is Flutter different from other solutions?
3. How to make an app?
4. How does it work?
5. Cool, but... What about X/Y/Z?

What is Flutter?

What is Flutter?

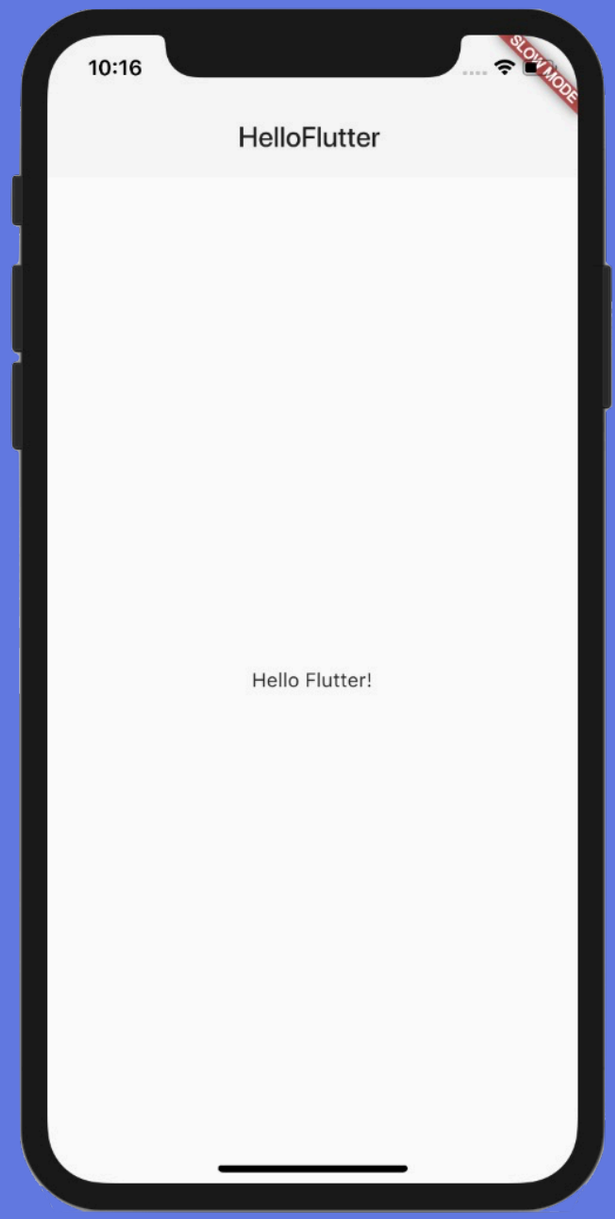
Flutter is Google's mobile UI framework for **crafting** high-quality native interfaces on iOS and Android in record time.

What is Flutter?



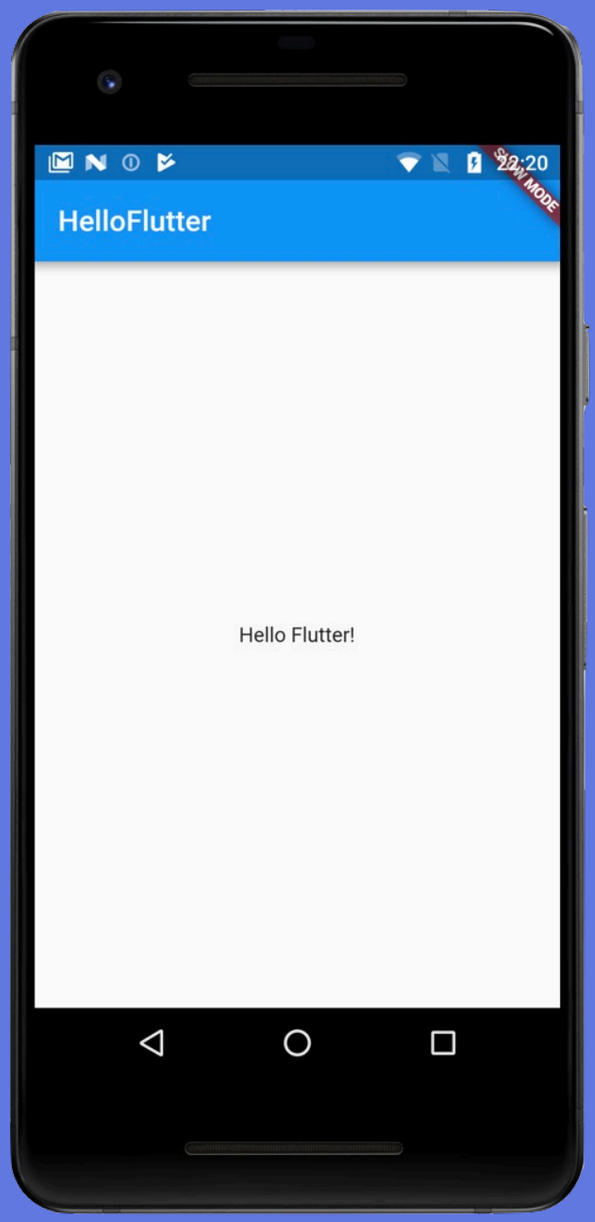
Android

Flutter



iOS

What is Flutter?

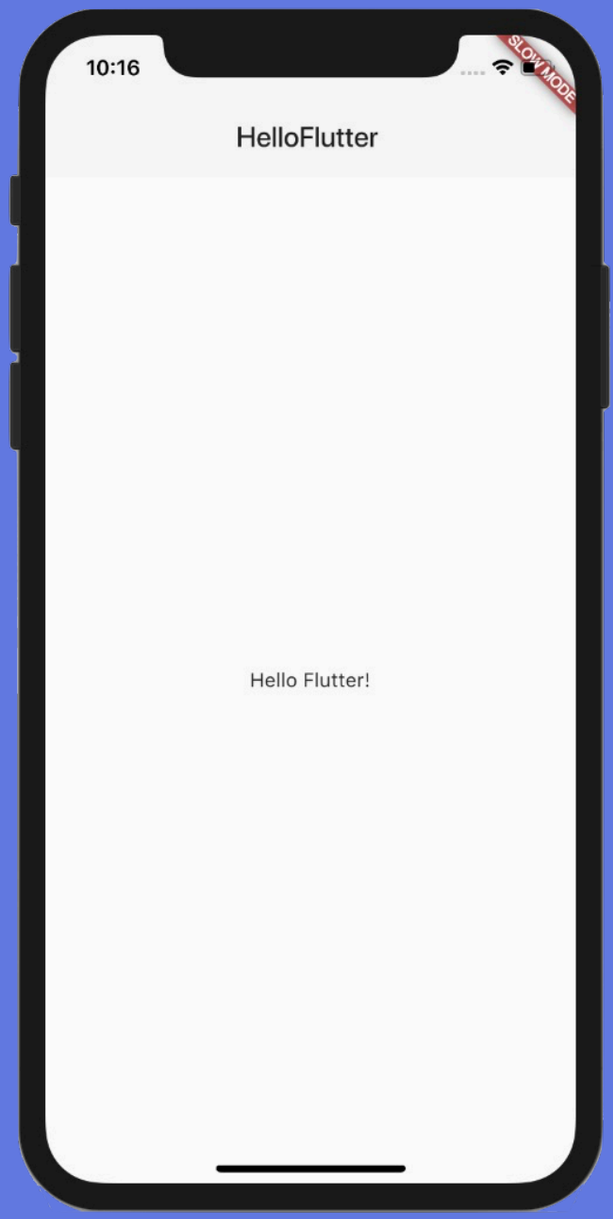


Android

Flutter

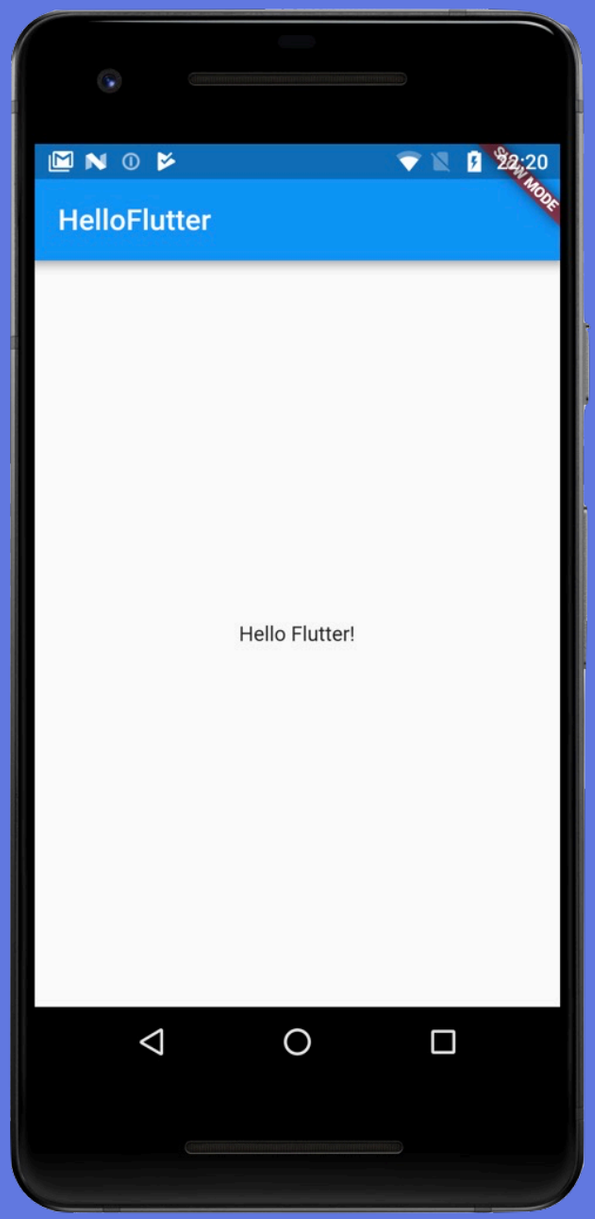
❤️

Dart



iOS

What is Flutter?

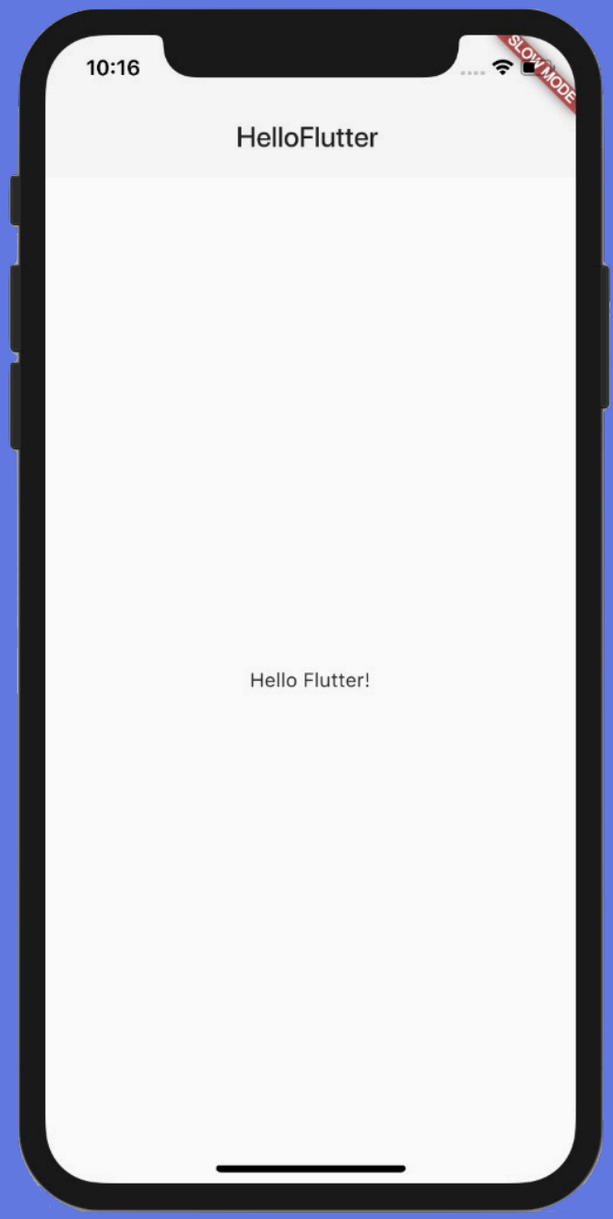


Android

Flutter

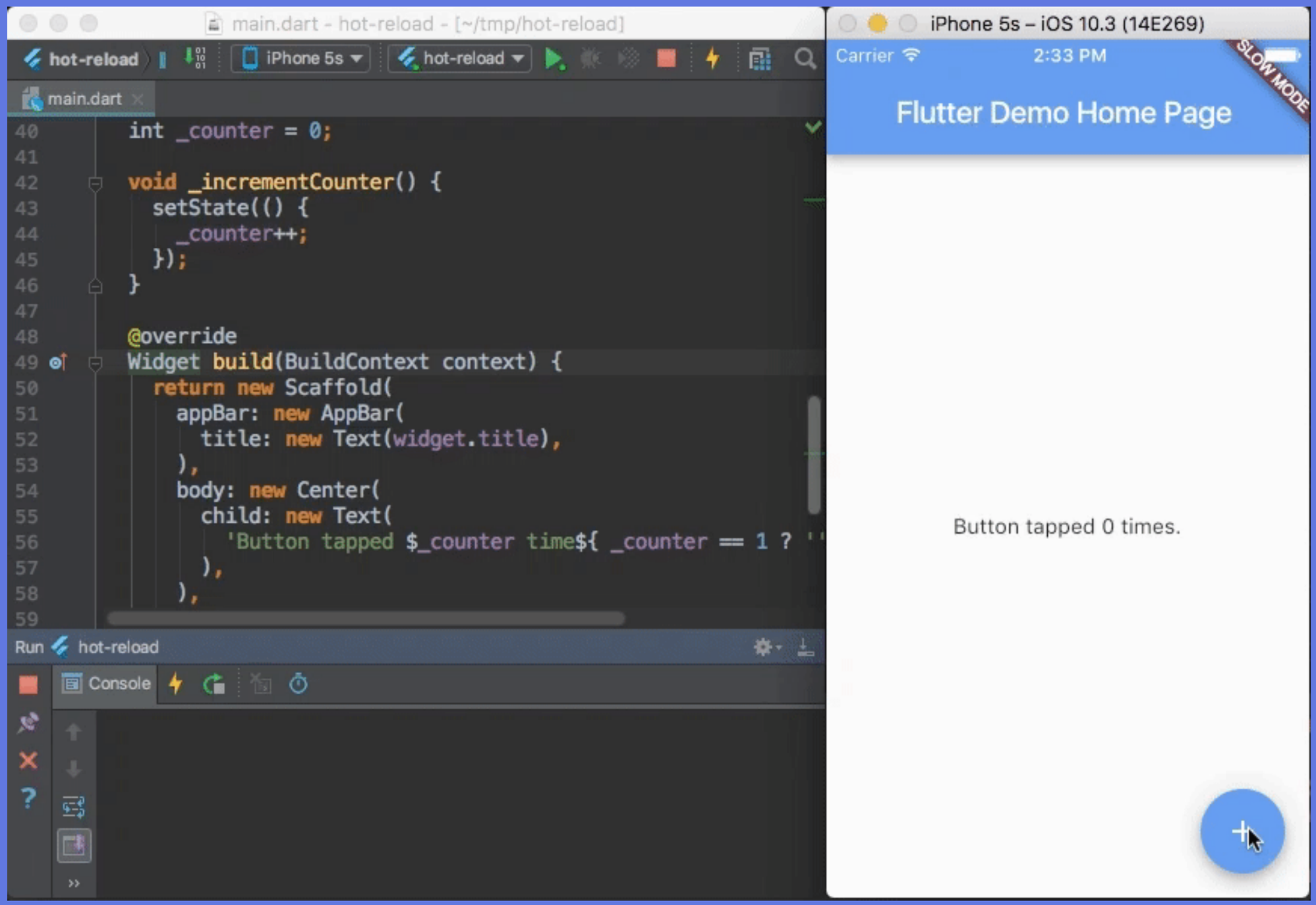
1.0

🎉🎉🎉

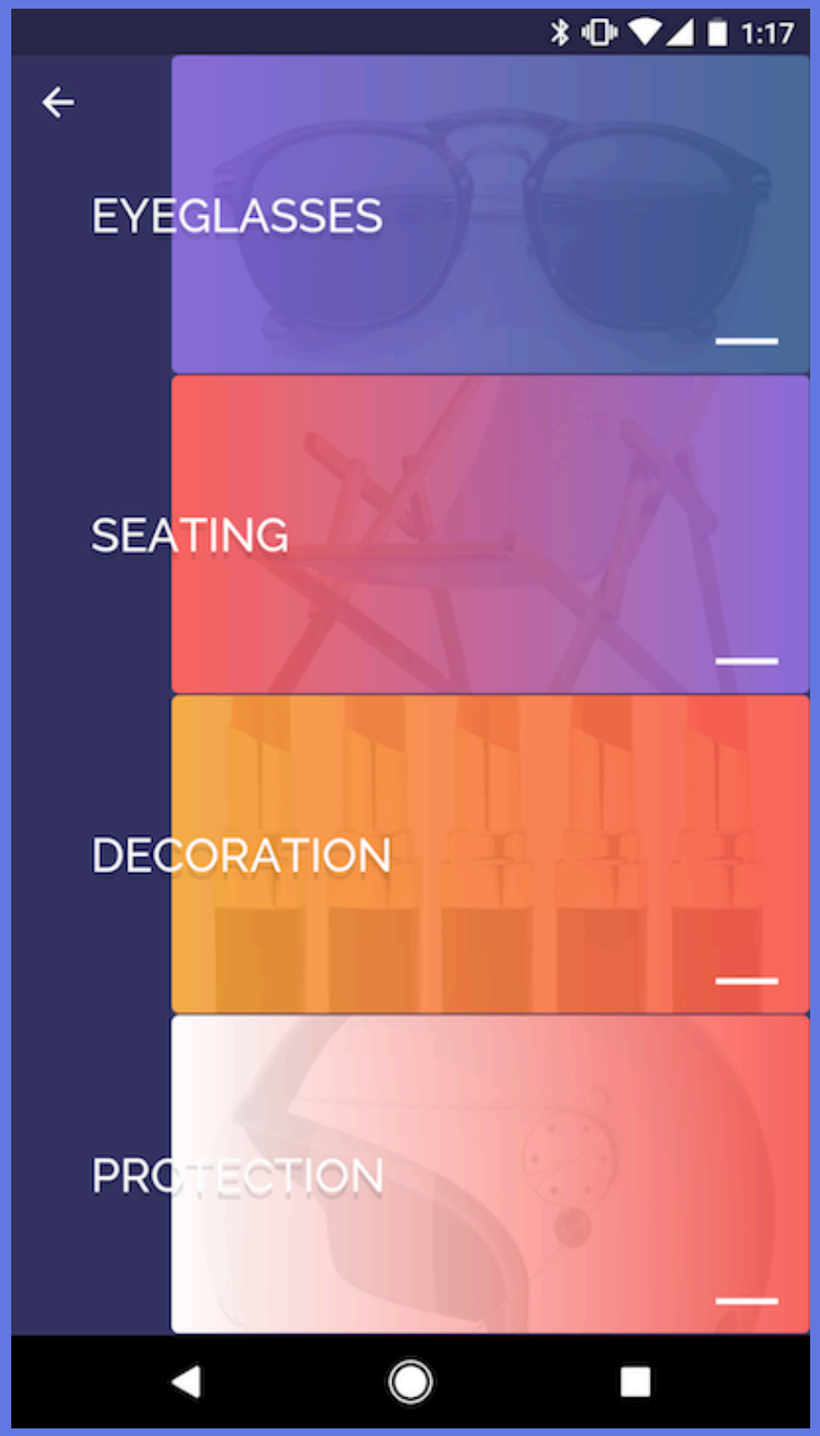


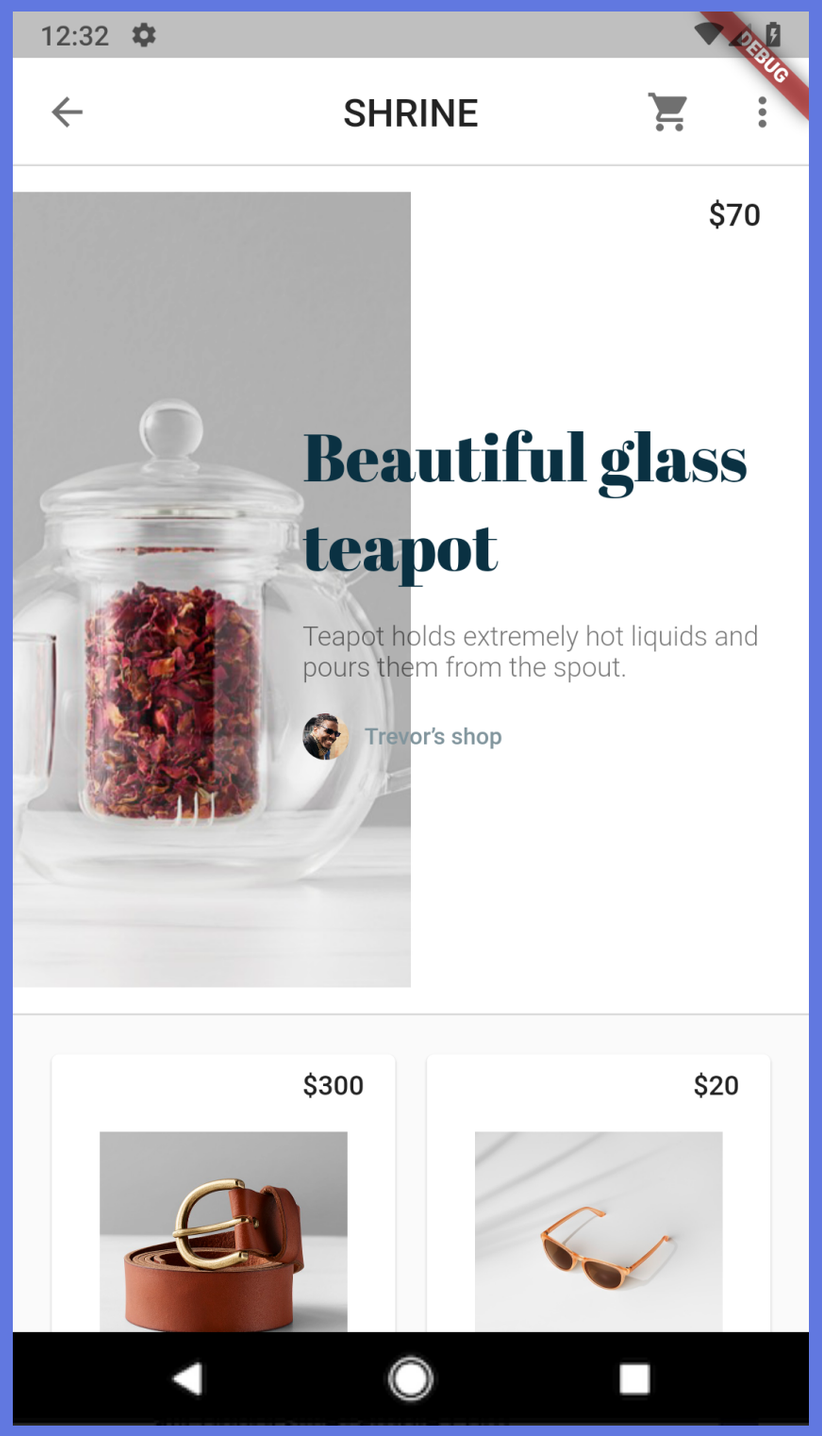
iOS

Fast development



Beautiful UI





Native Performance

Date & Time Pickers

Date pickers use a dialog window to select a single date on mobile. Time pickers use a dialog to select a single time (in the hours:minutes format) on mobile.

Documentation

TabBar

A Material Design widget that displays a horizontal row of tabs.

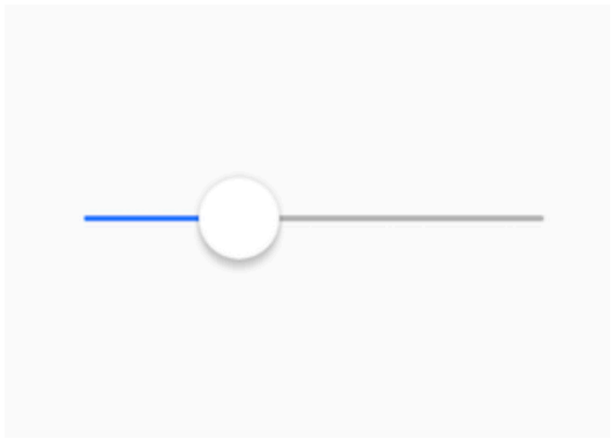
Documentation , Samples

FloatingActionButton

A floating action button is a circular icon button that hovers over content to promote a primary action in the application. Floating action buttons are...

Documentation

Native Performance



CupertinoSlider

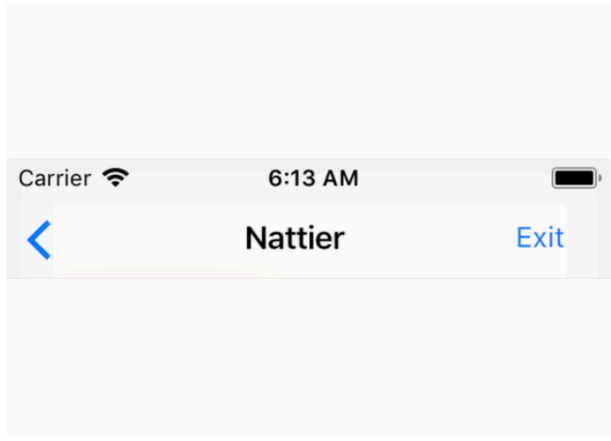
Used to select from a range of values.

Documentation

CupertinoPicker

An iOS-style picker control. Used to select an item in a short list.

Documentation



CupertinoNavigationBar

An iOS-style top navigation bar. Typically used with CupertinoPageScaffold.

Documentation

How is Flutter different?

How is Flutter different?

Application Code (JS)

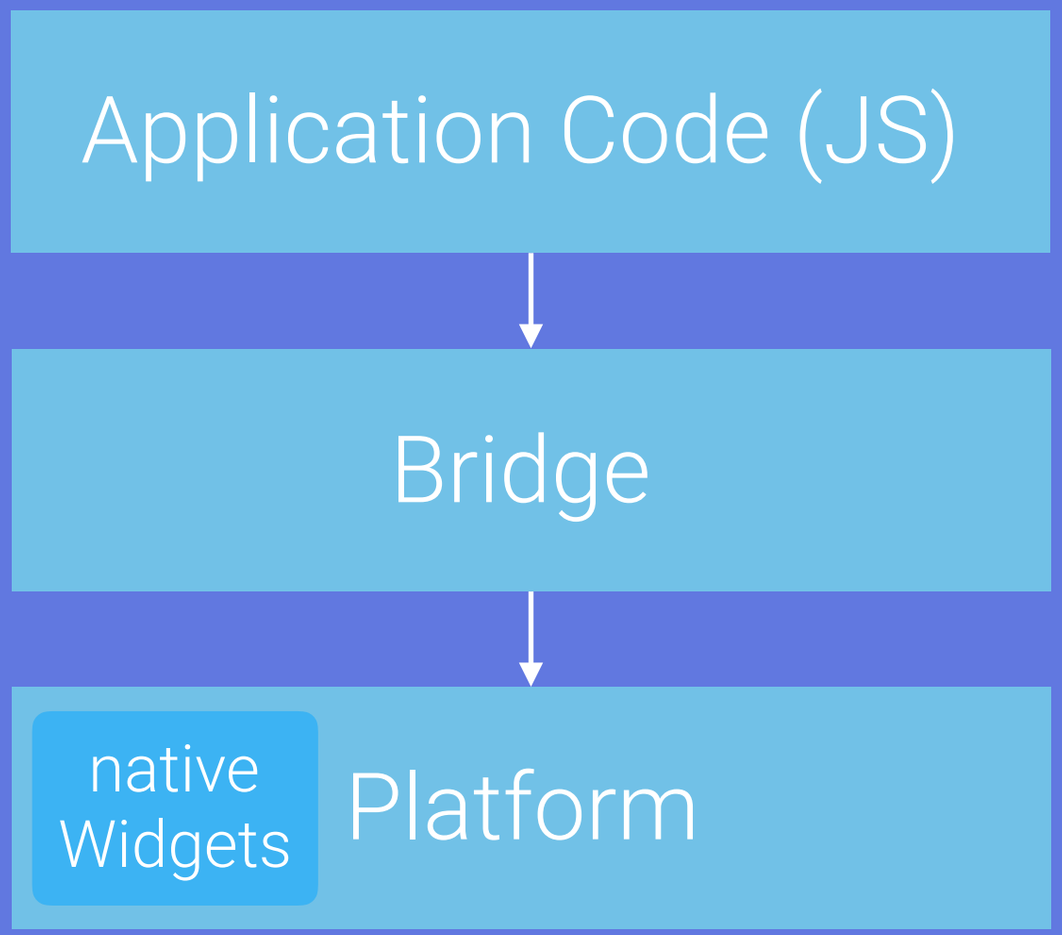
React Native

How is Flutter different?

Application Code (JS)

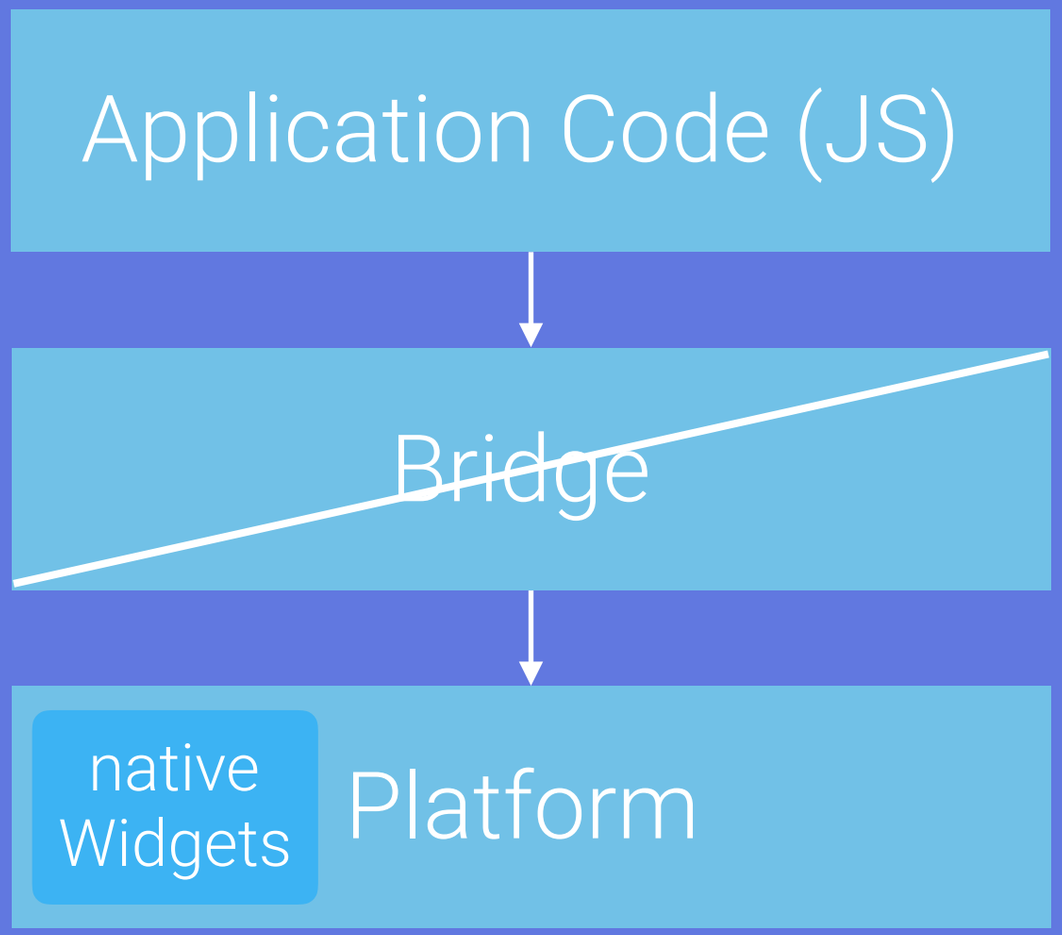
React Native

How is Flutter different?



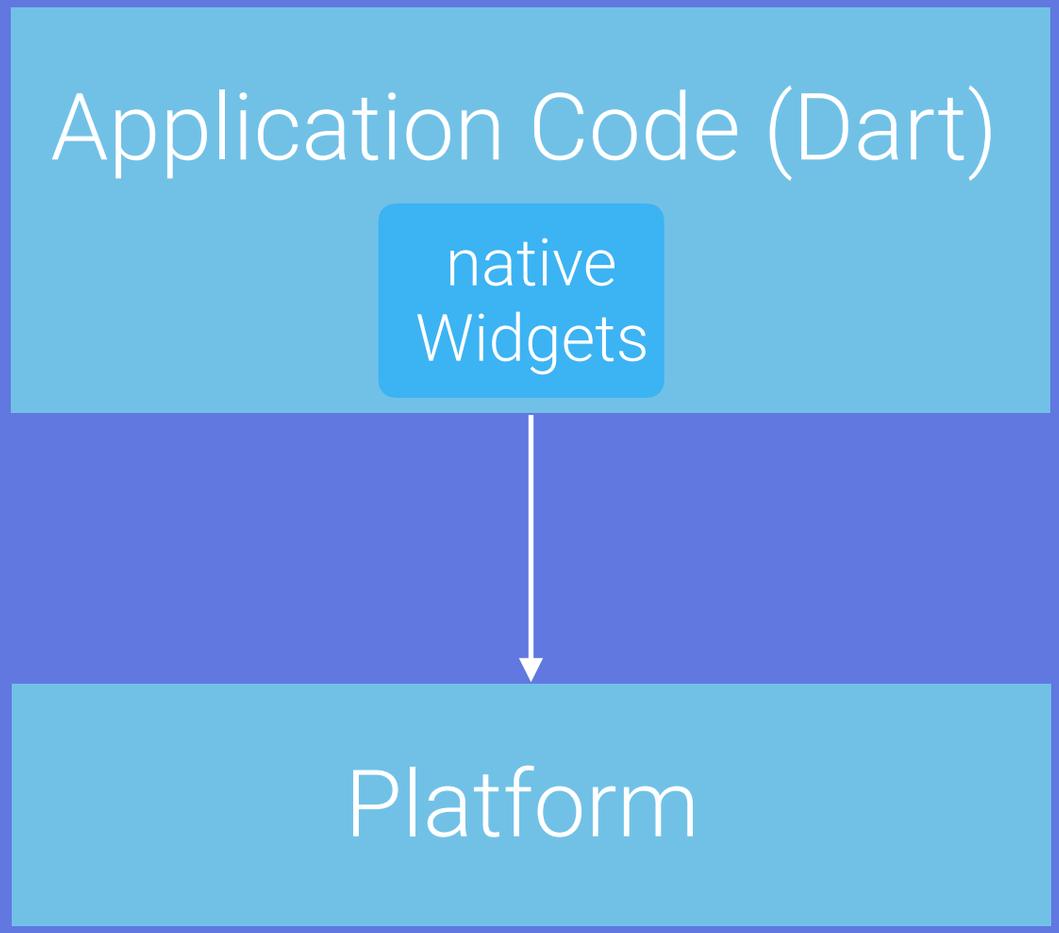
React Native

How is Flutter different?

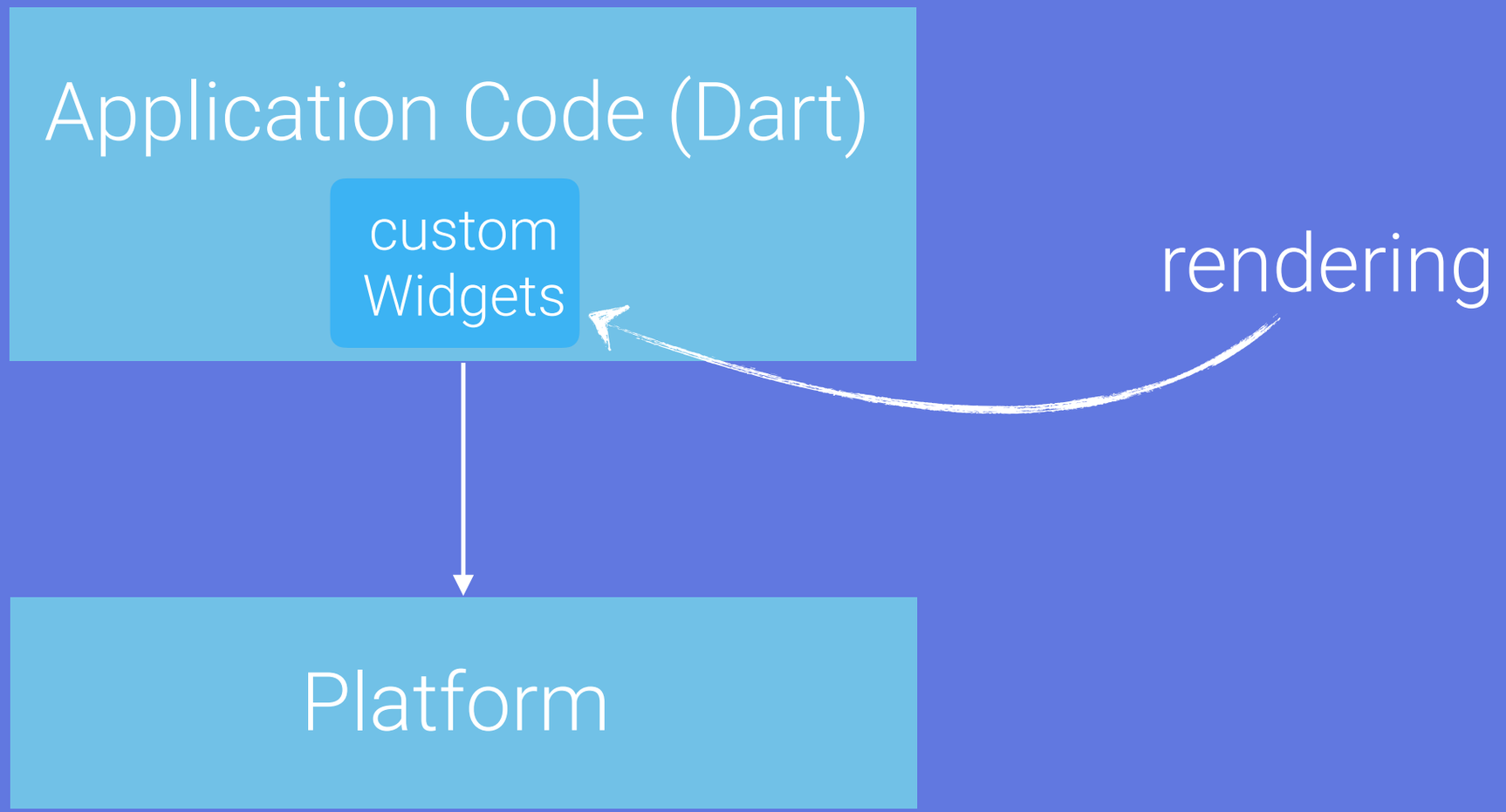


React Native

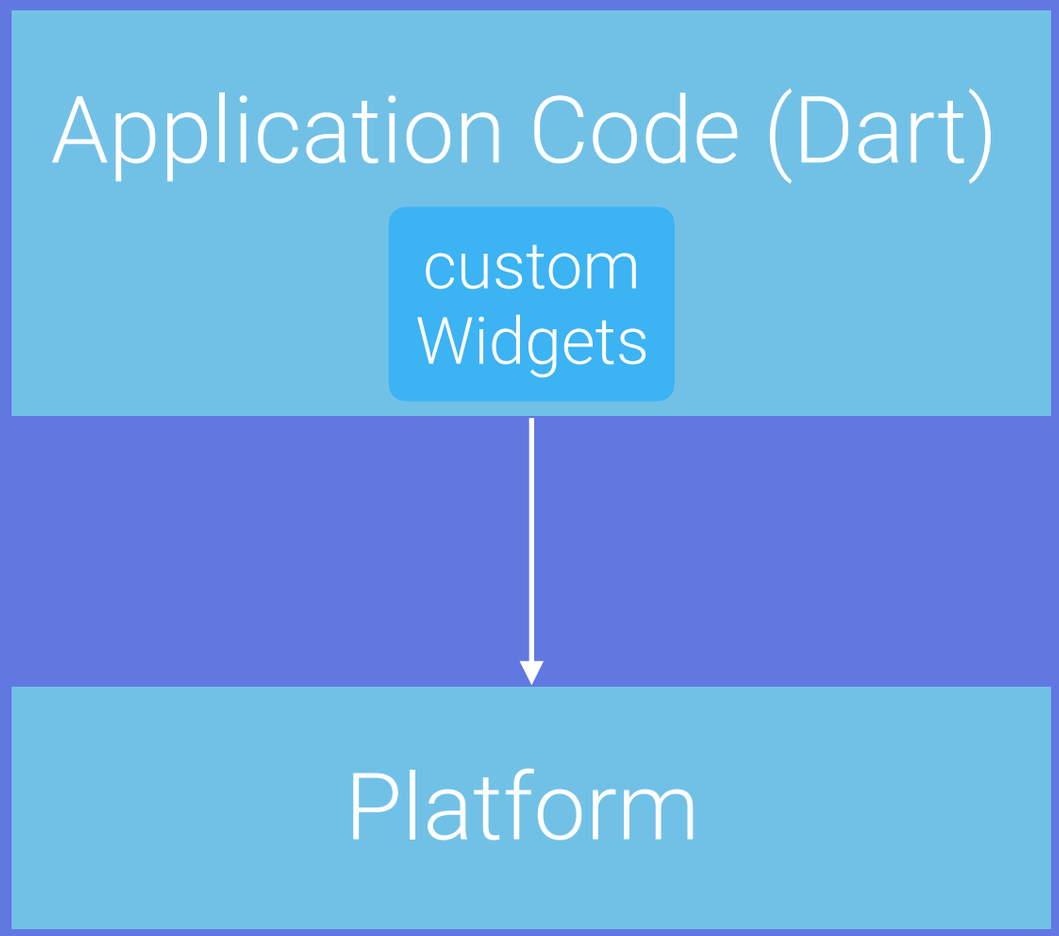
How is Flutter different?



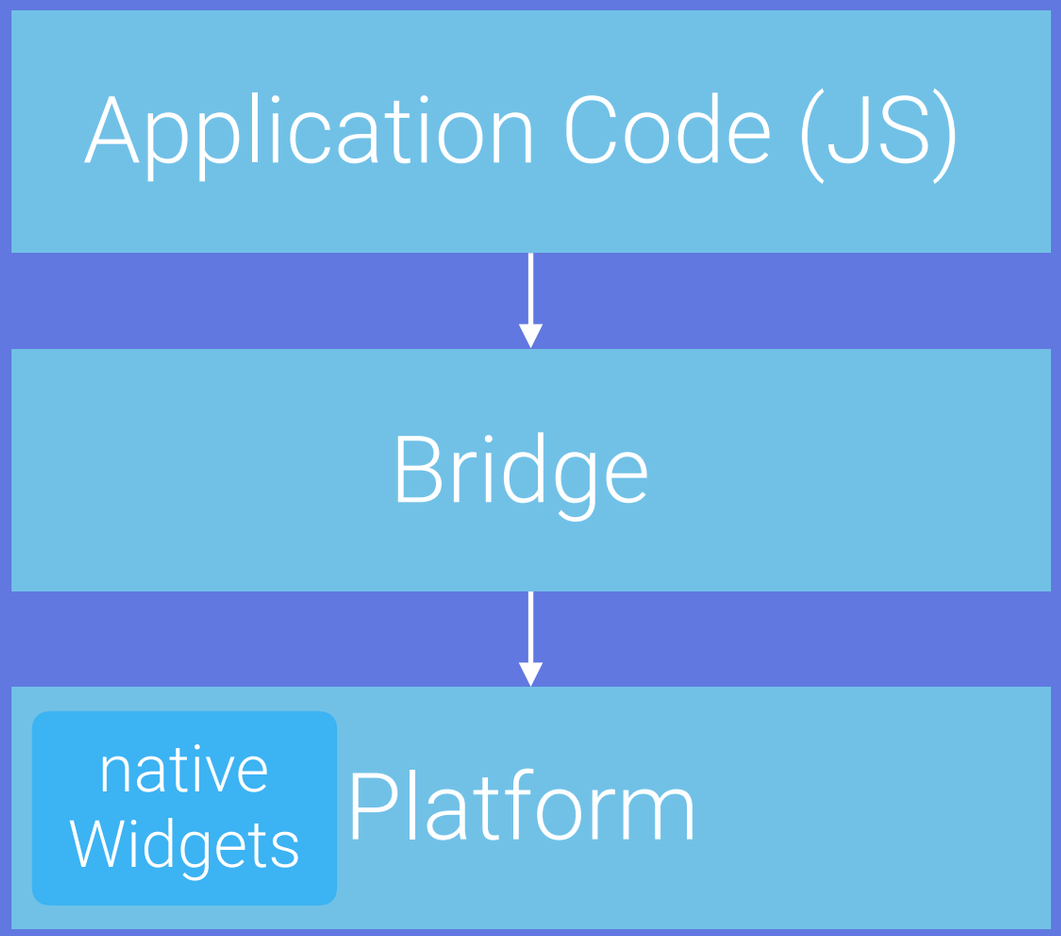
How is Flutter different?



How is Flutter different?

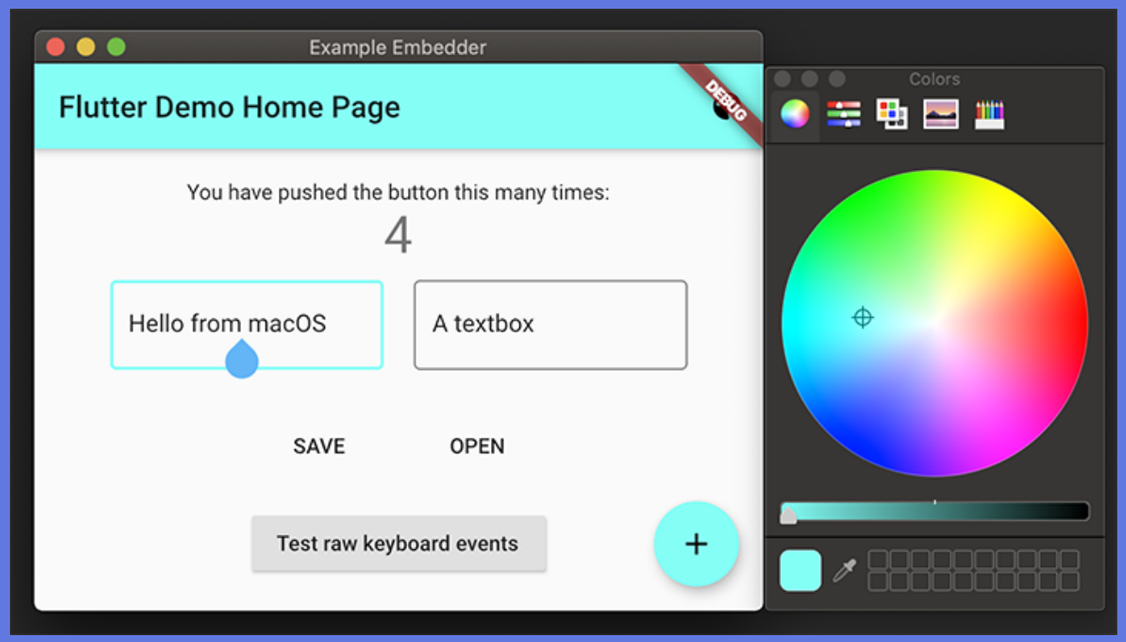


Flutter



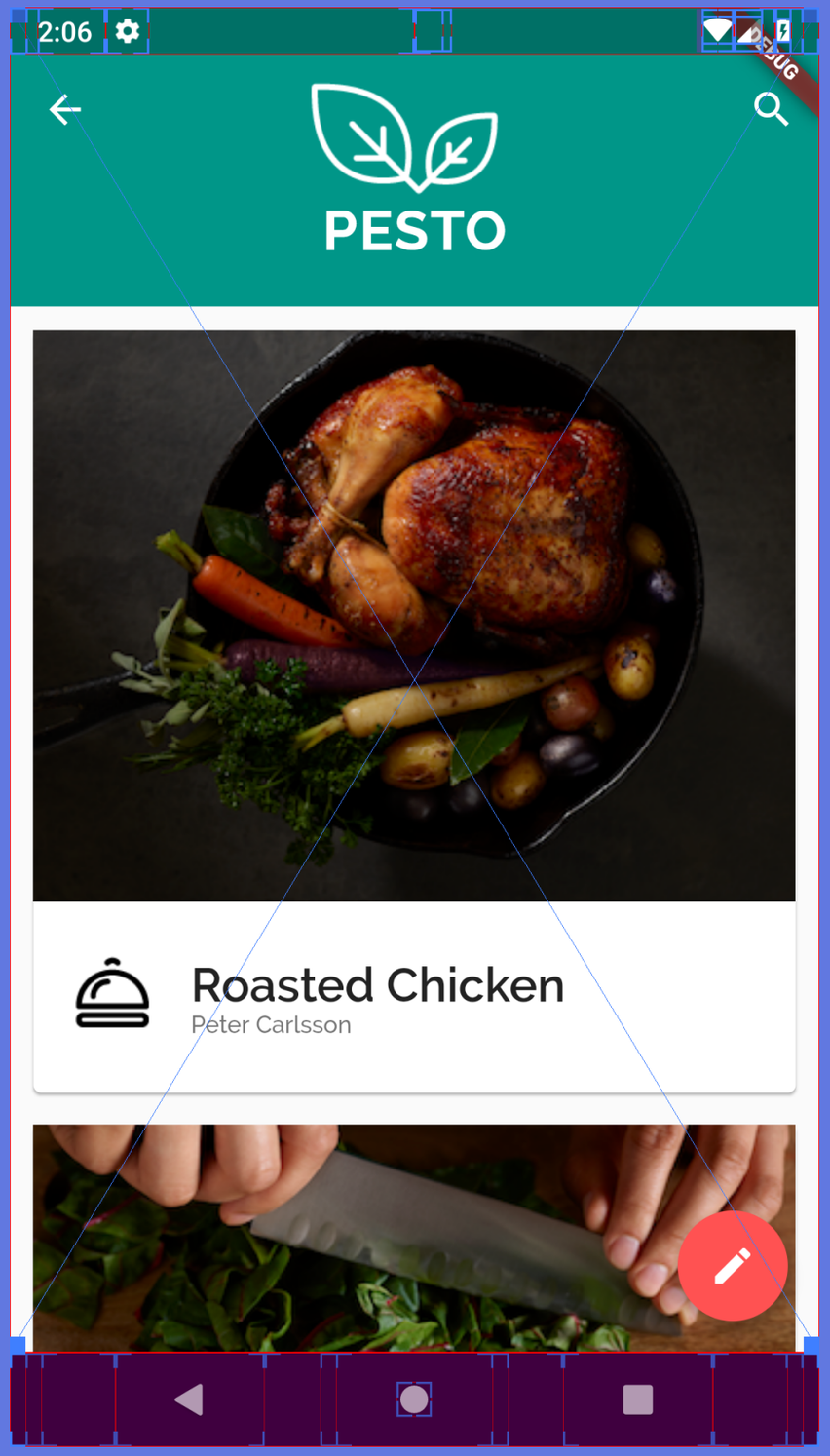
React Native

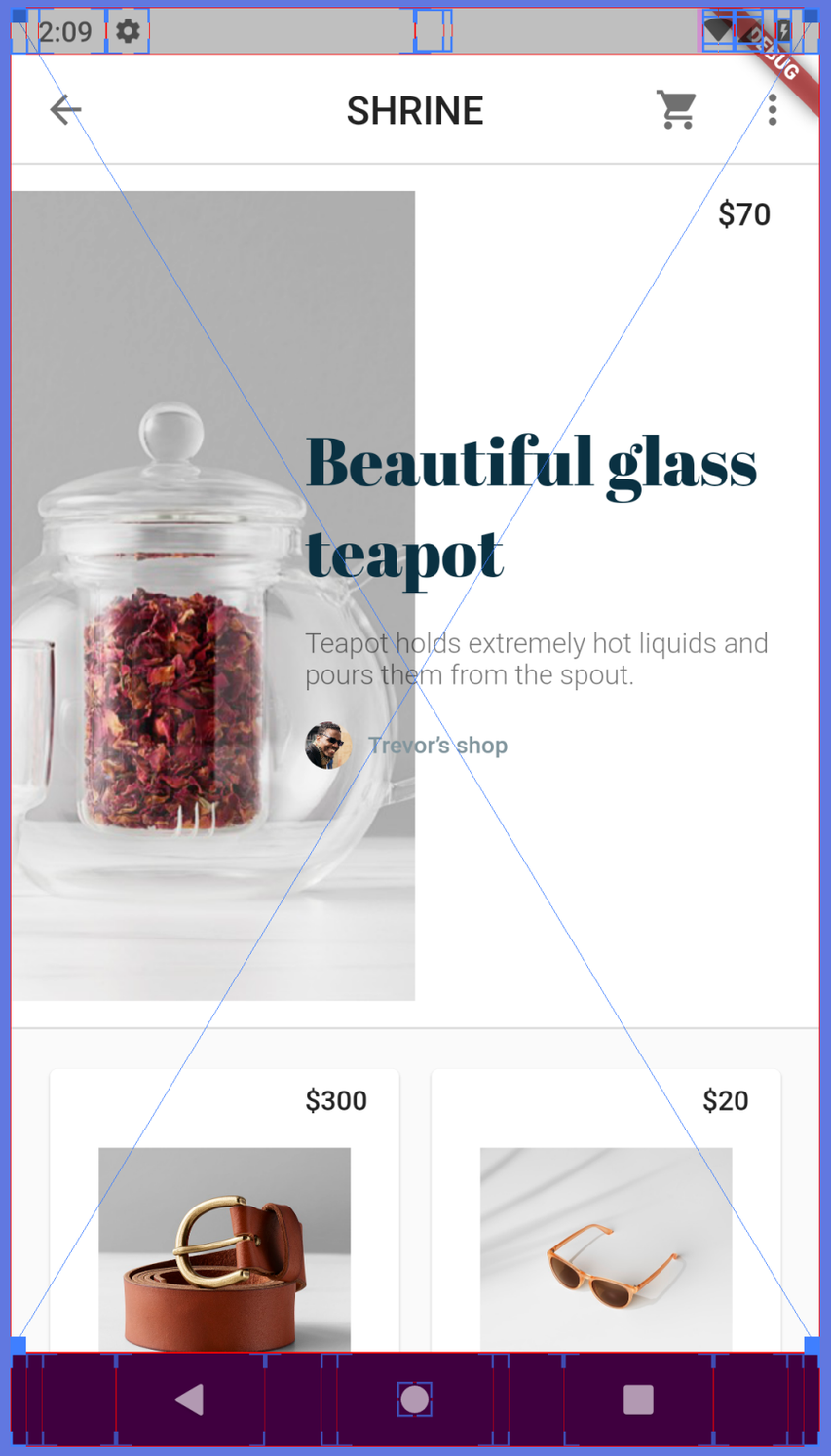
How is Flutter different?



Photos from: https://medium.com/flutter-io/https-medium-com-flutter-io-pitching-flutter-2d4f494e47d1

How is Flutter different?





How is Flutter different?

FAST

60 fps

How is Flutter different?

FAST

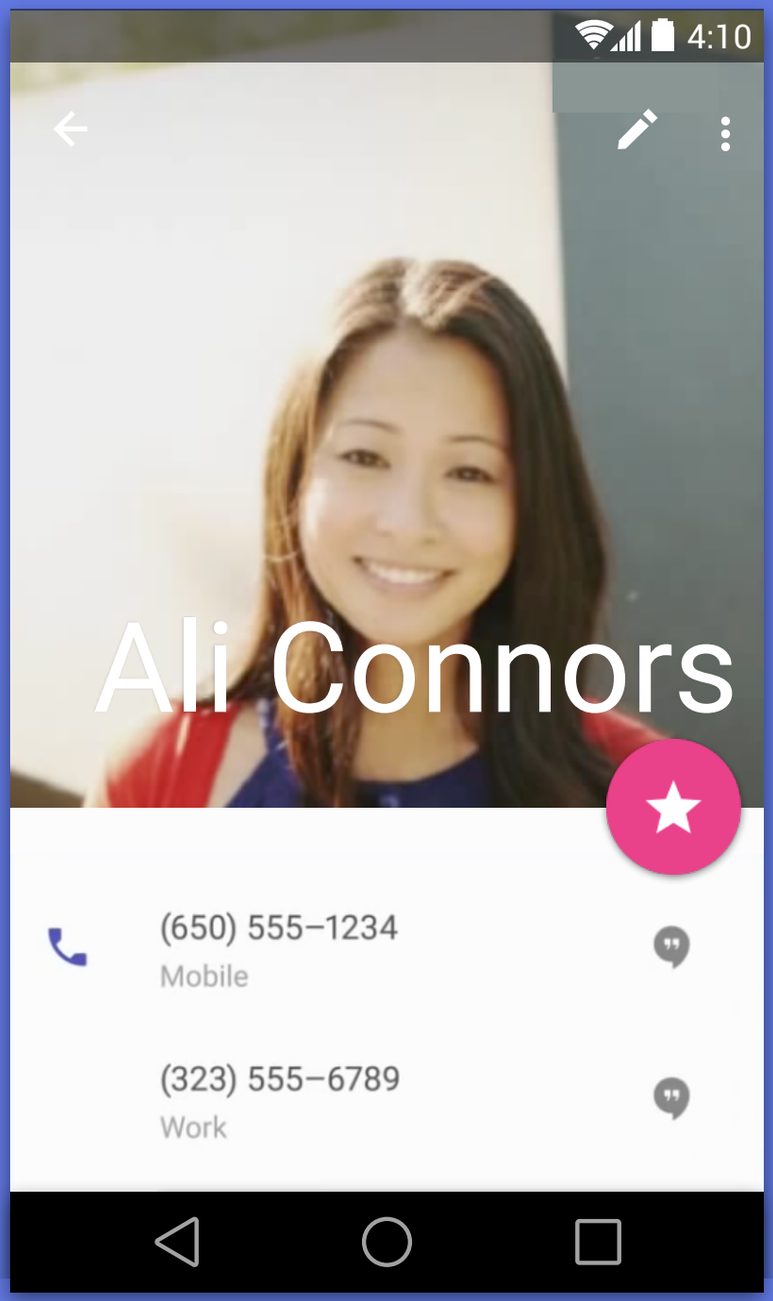
60 fps

FLEXIBLE

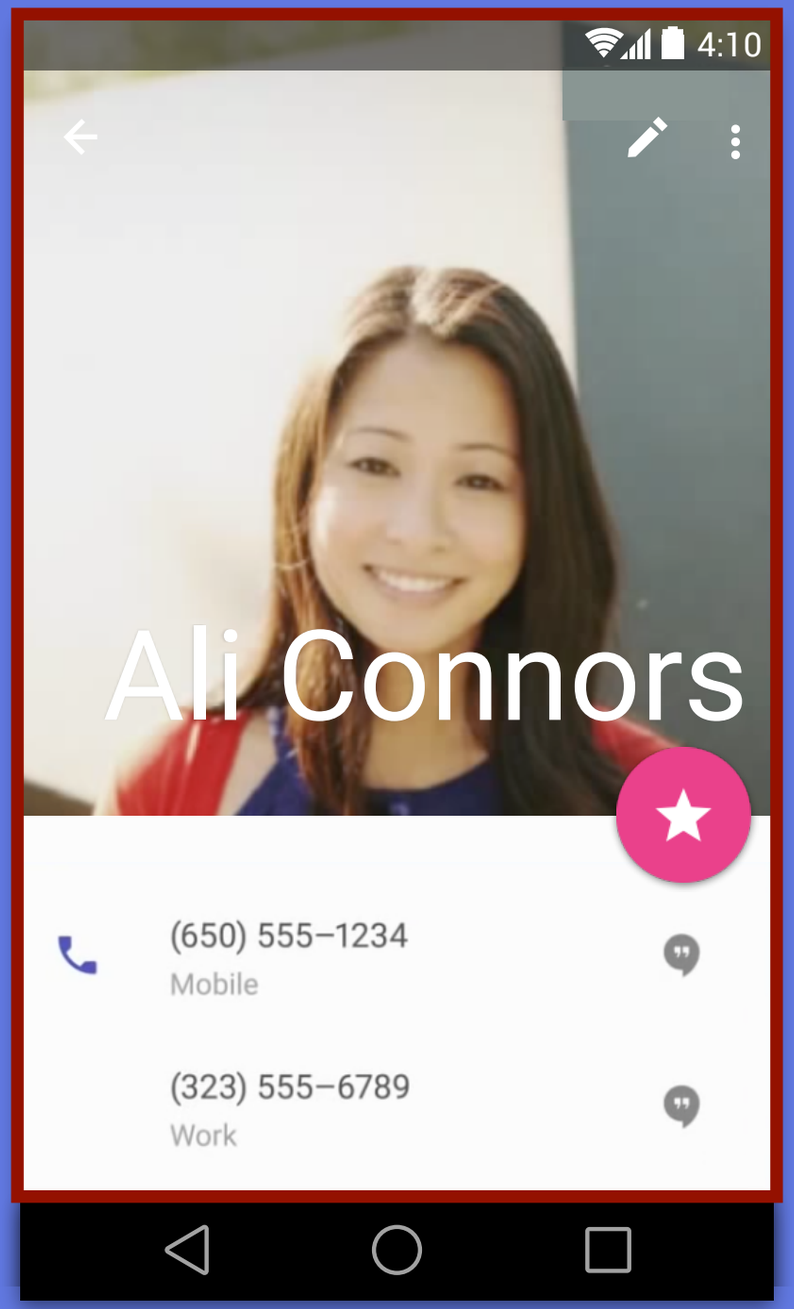
WIDGETS

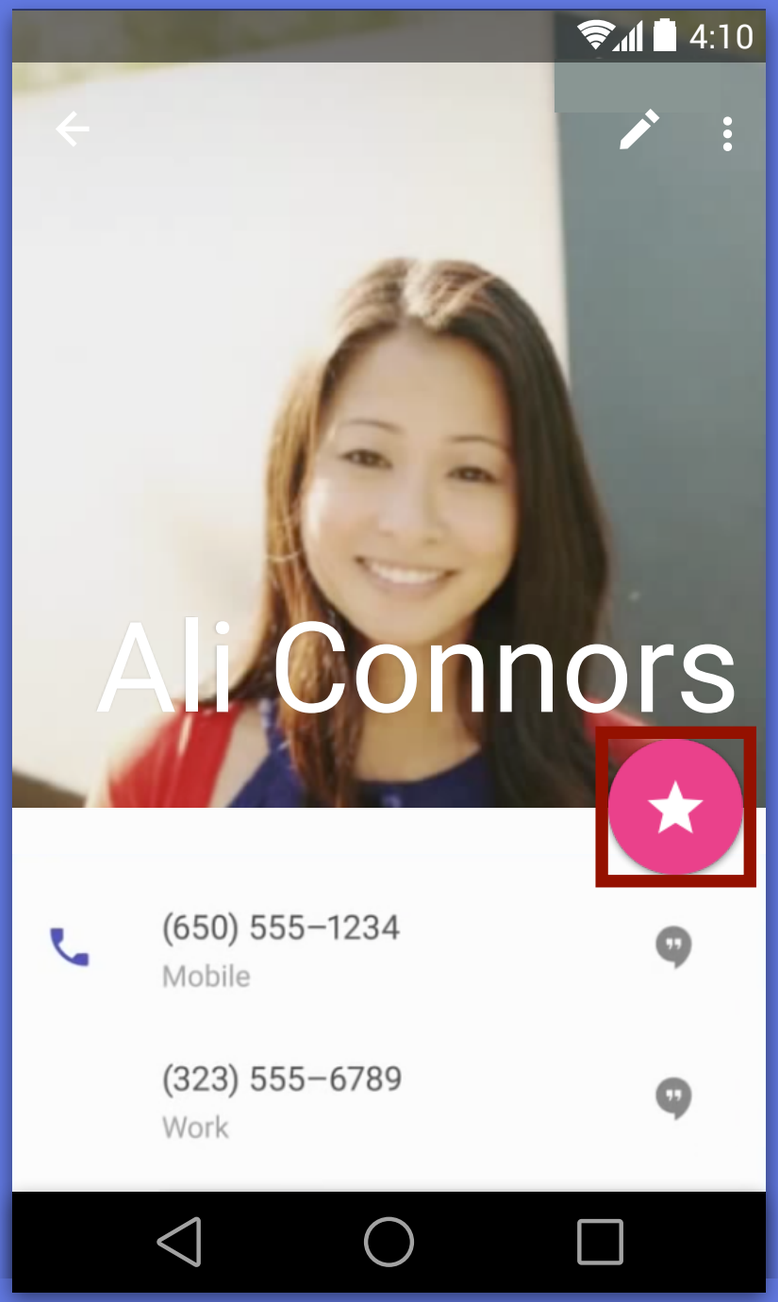
How is Flutter different?

EVERYTHING IS A WIDGET


4:10
Ali Connors
(650) 555–1234
Mobile
(323) 555–6789
Work

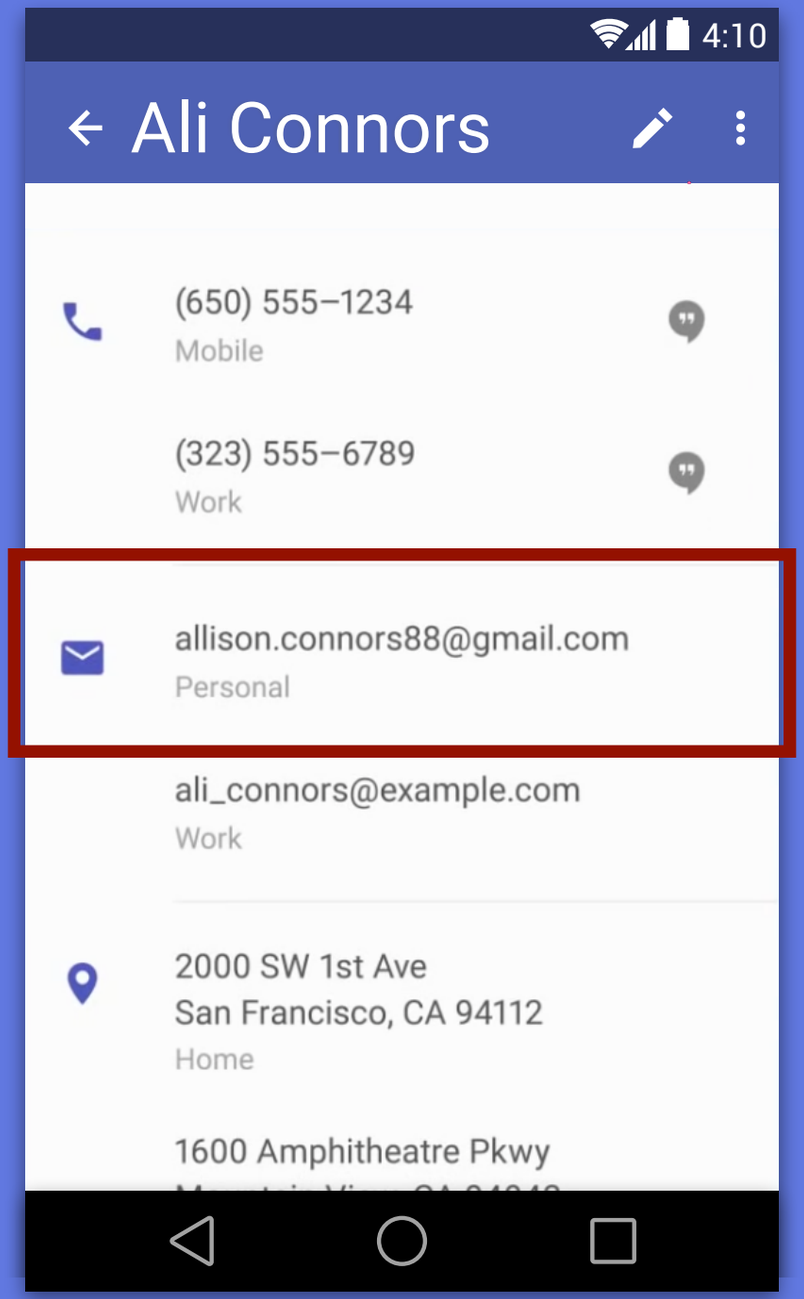
Scaffold



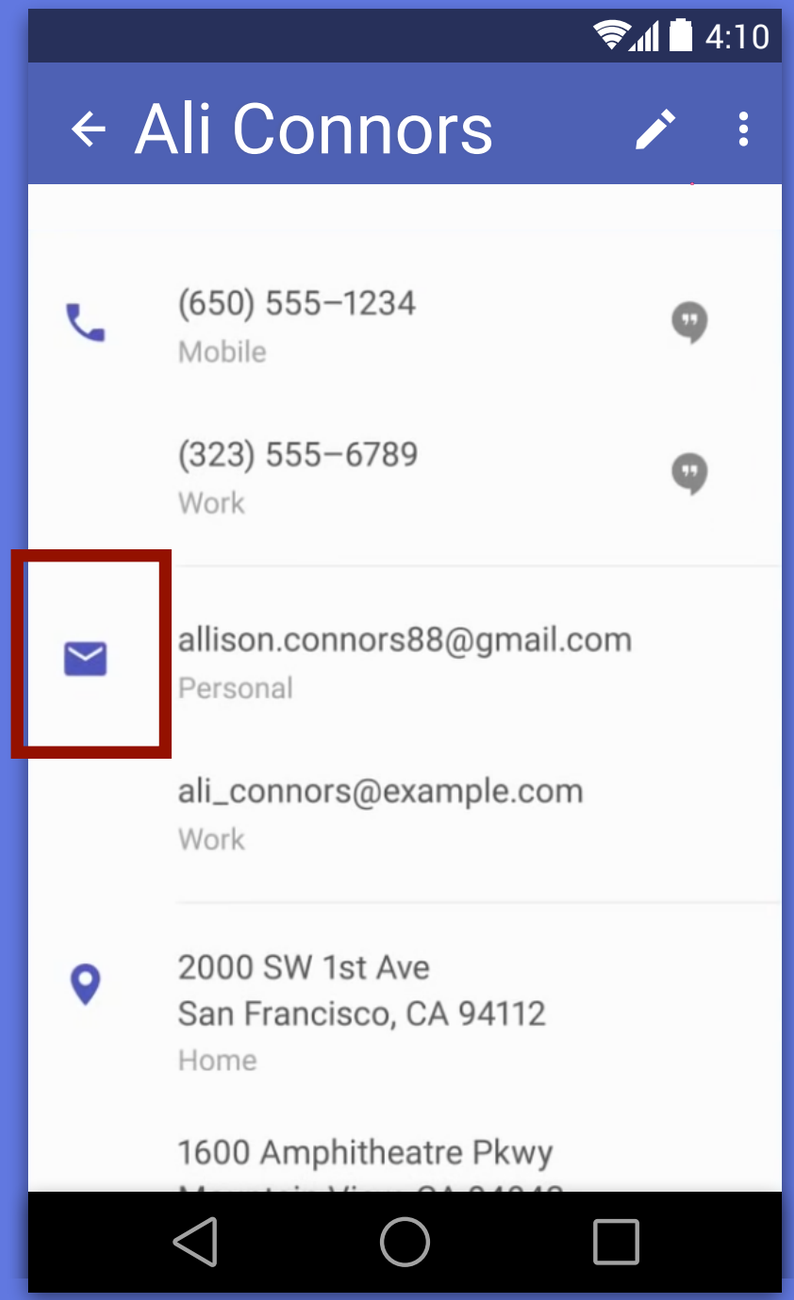


Floating ActionButton

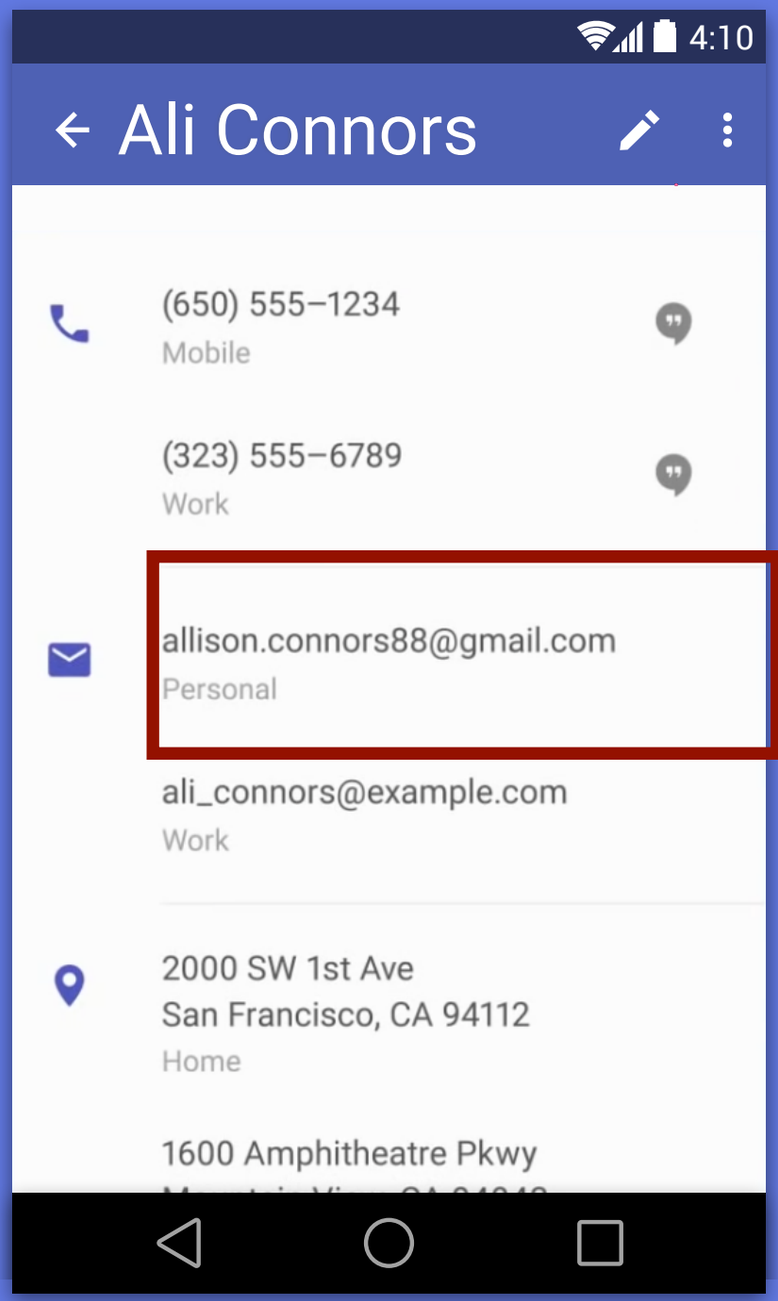
Row


4:10
Ali Connors
(650) 555–1234
Mobile
(323) 555–6789
Work
allison.connors88@gmail.com
Personal
ali_connors@example.com
Work
2000 SW 1st Ave
San Francisco, CA 94112
Home
1600 Amphitheatre Pkwy

Center


4:10
Ali Connors
(650) 555–1234
Mobile
(323) 555–6789
Work
allison.connors88@gmail.com
Personal
ali_connors@example.com
Work
2000 SW 1st Ave
San Francisco, CA 94112
Home
1600 Amphitheatre Pkwy

Padding


4:10
Ali Connors
(650) 555–1234
Mobile
(323) 555–6789
Work
allison.connors88@gmail.com
Personal
ali_connors@example.com
Work
2000 SW 1st Ave
San Francisco, CA 94112
Home
1600 Amphitheatre Pkwy

Android Equivalents

Android	Flutter
LinearLayout	Column
RelativeLayout	Column, Row, Stack
RecyclerView	ListView
gestures	GestureDetector
padding	Padding

How is Flutter different?

FAST

60 fps

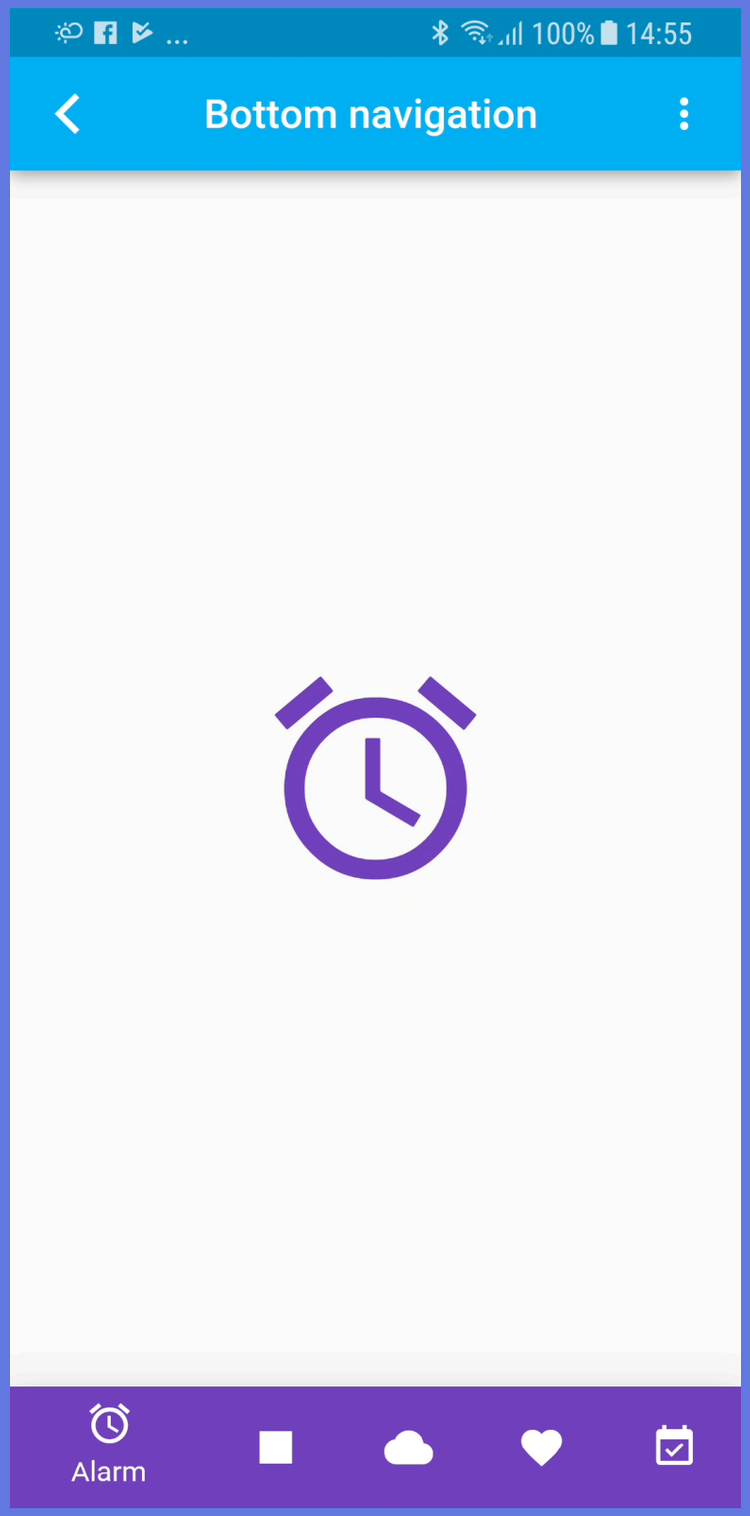
FLEXIBLE

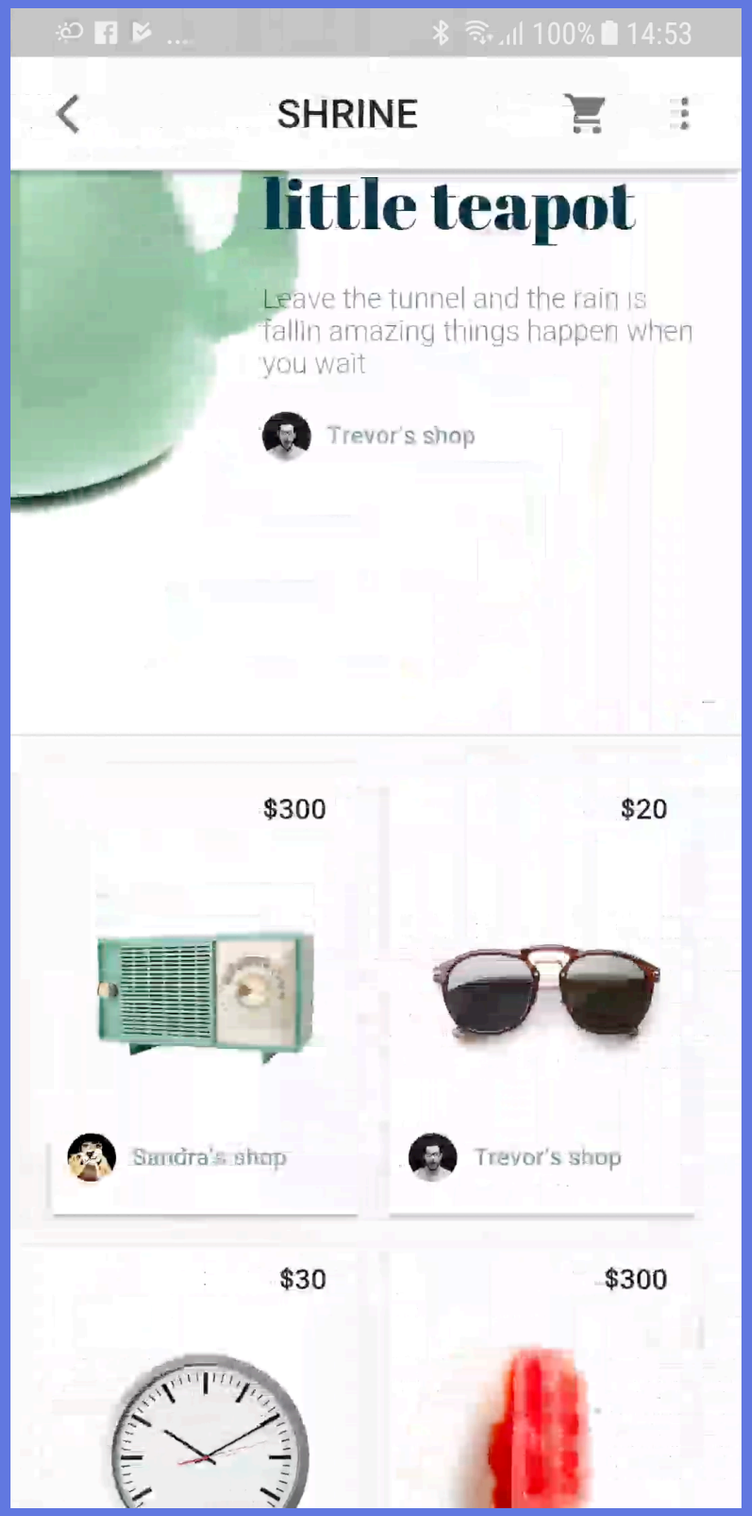
WIDGETS

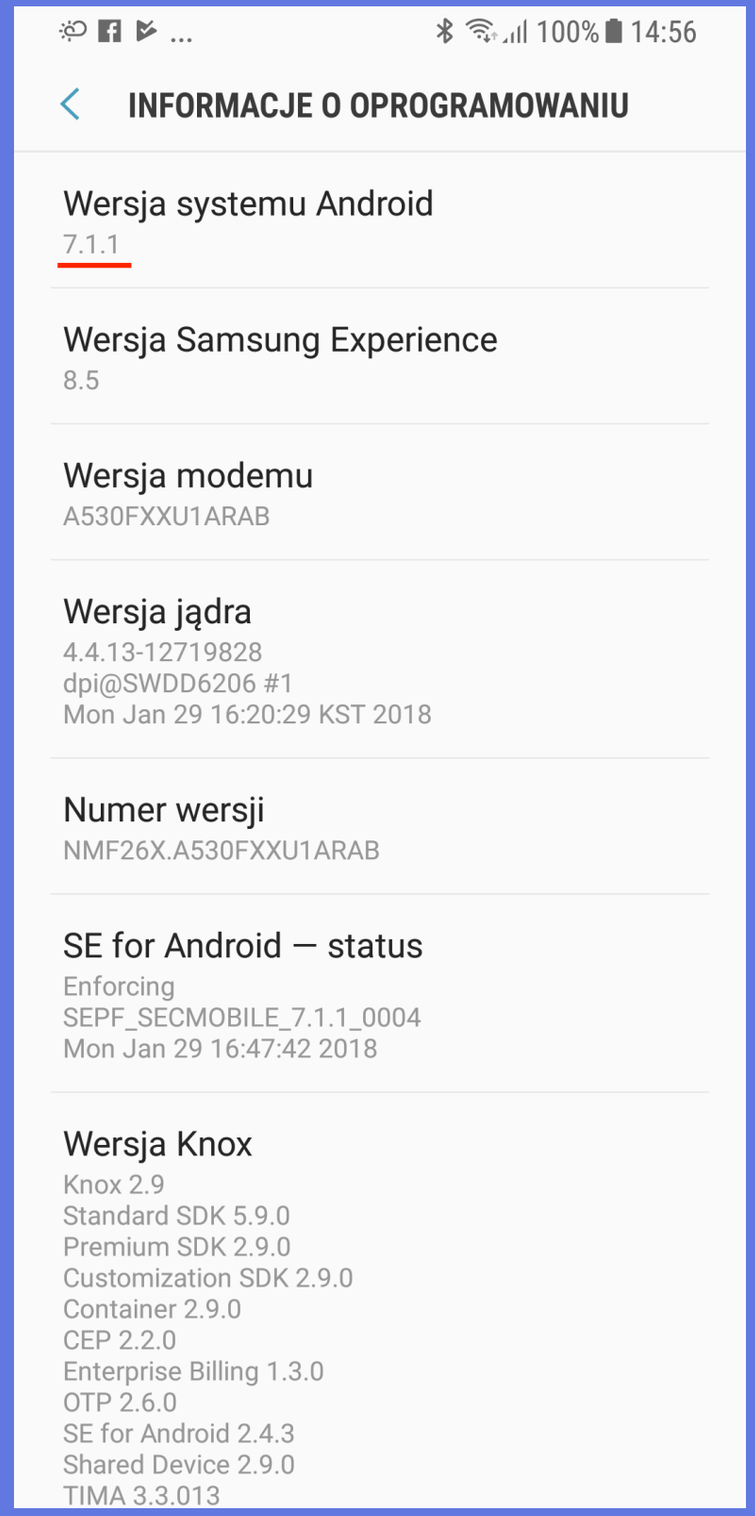
NATIVE

LOOK&FEEL

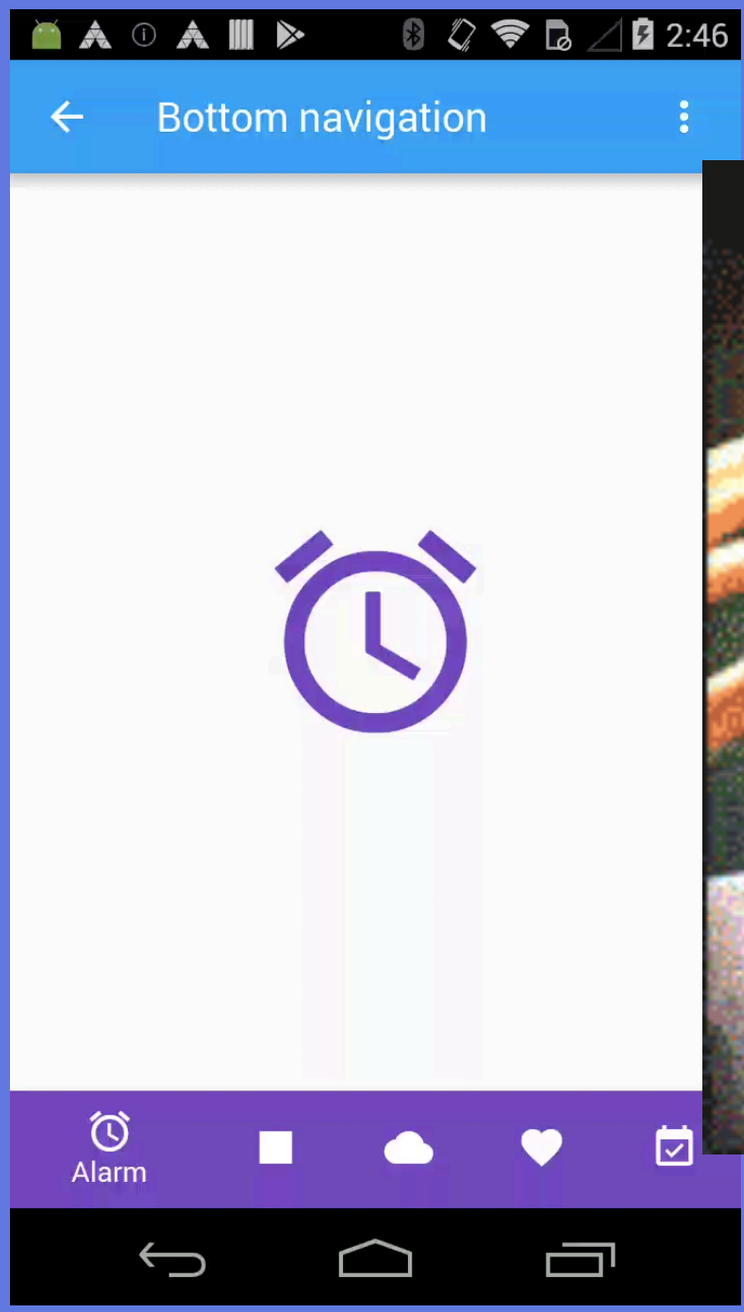
How is Flutter different?

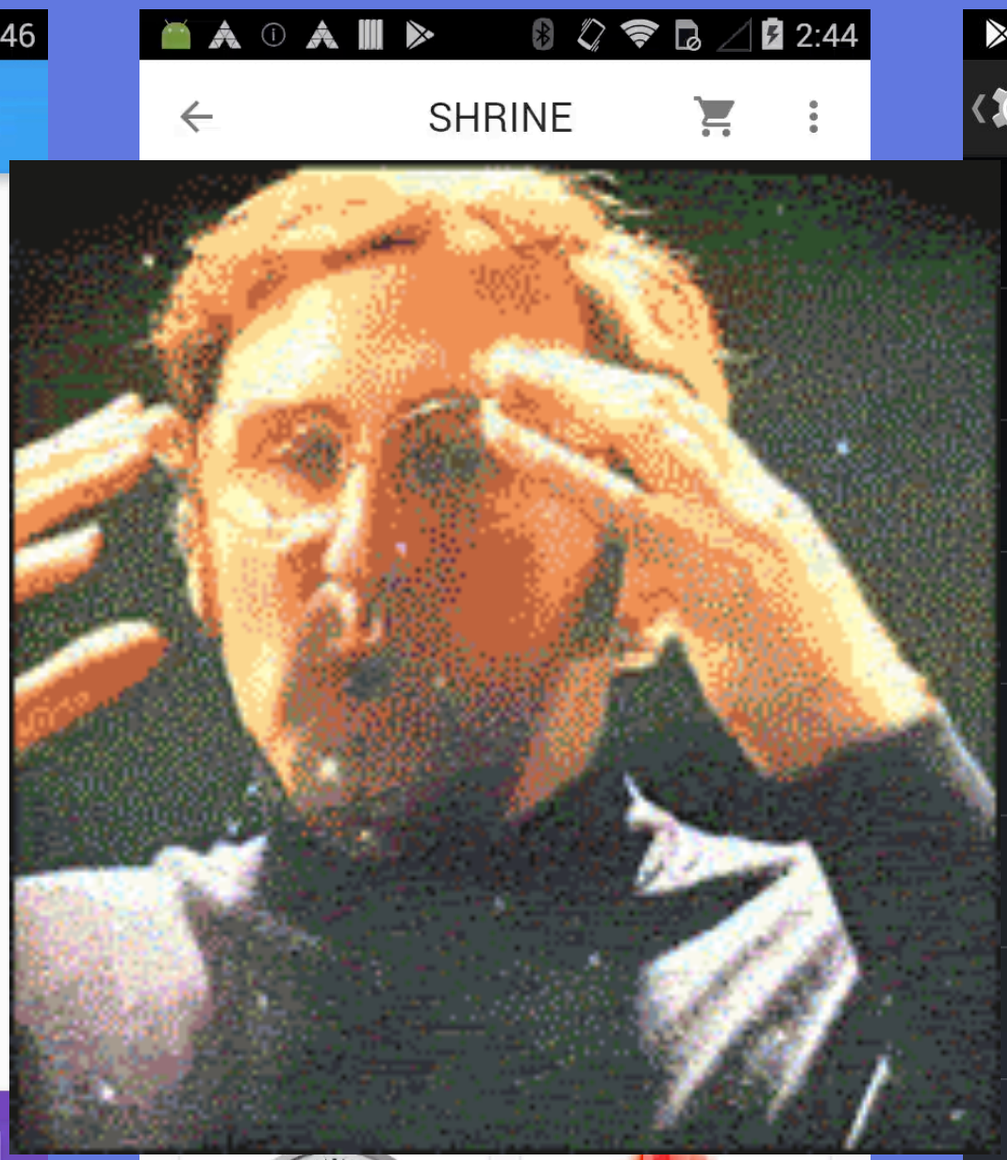


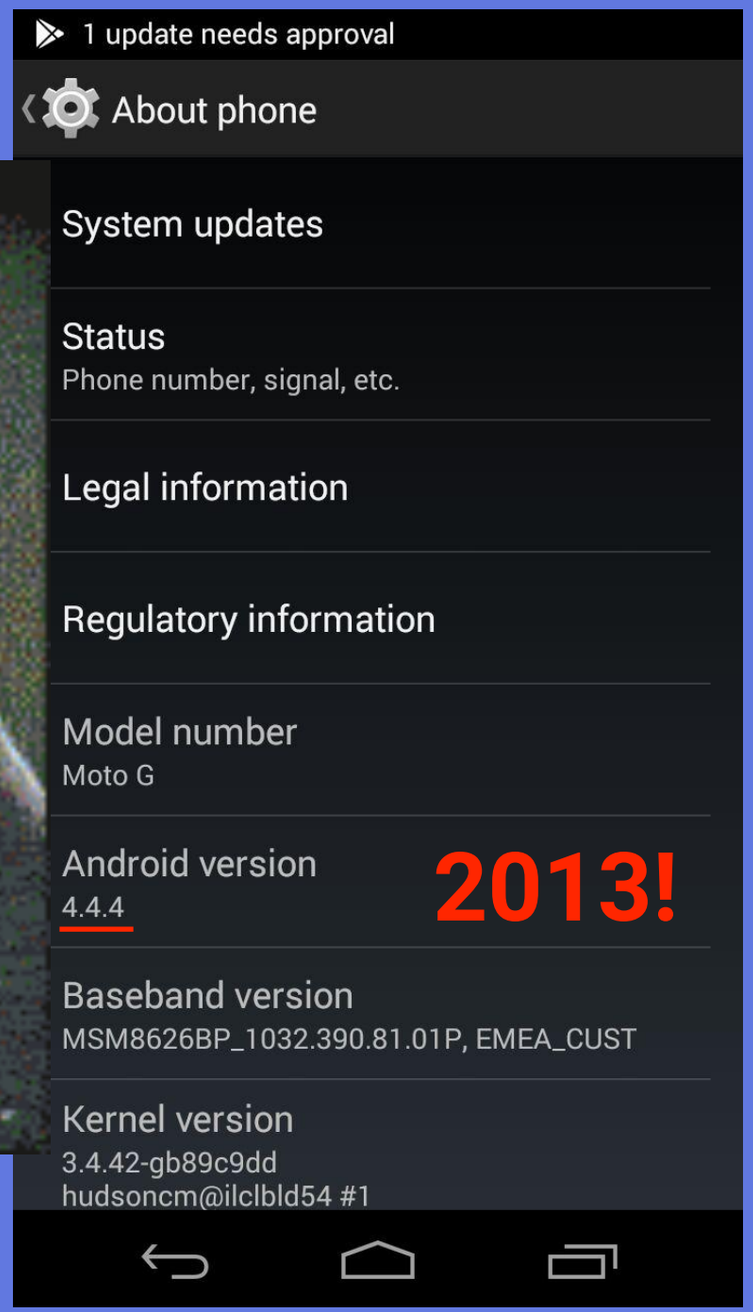




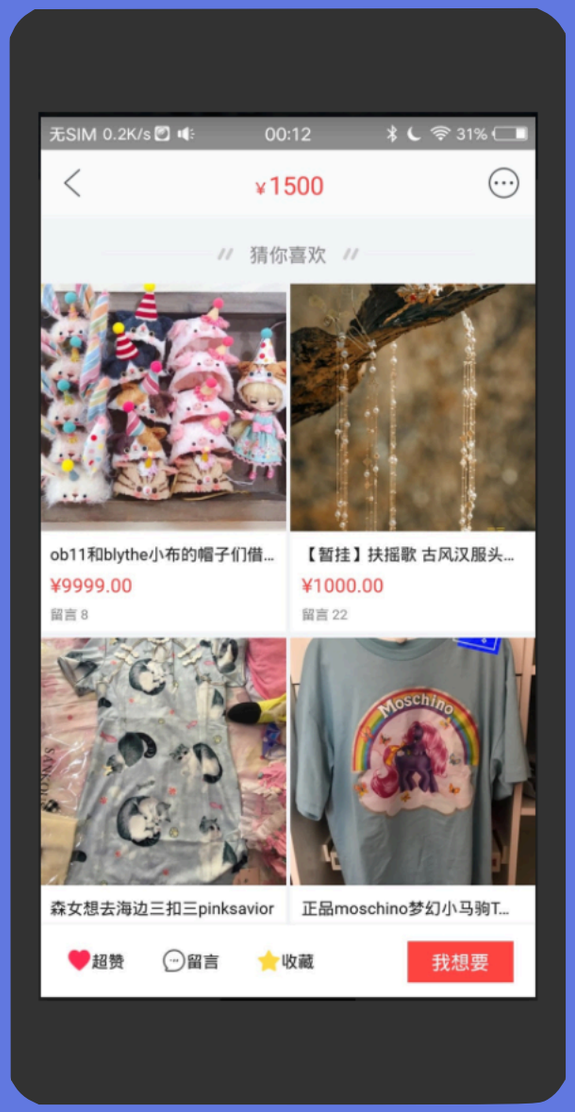
How is Flutter different?



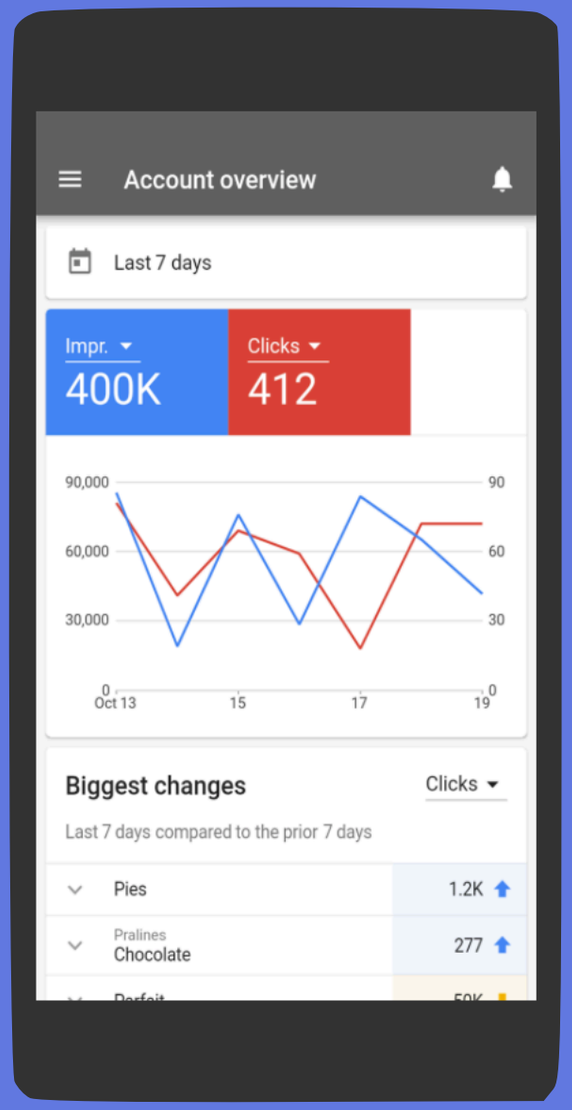




Who uses Flutter?



Alibaba
50+ million users



Google Ads,
5+ milion users

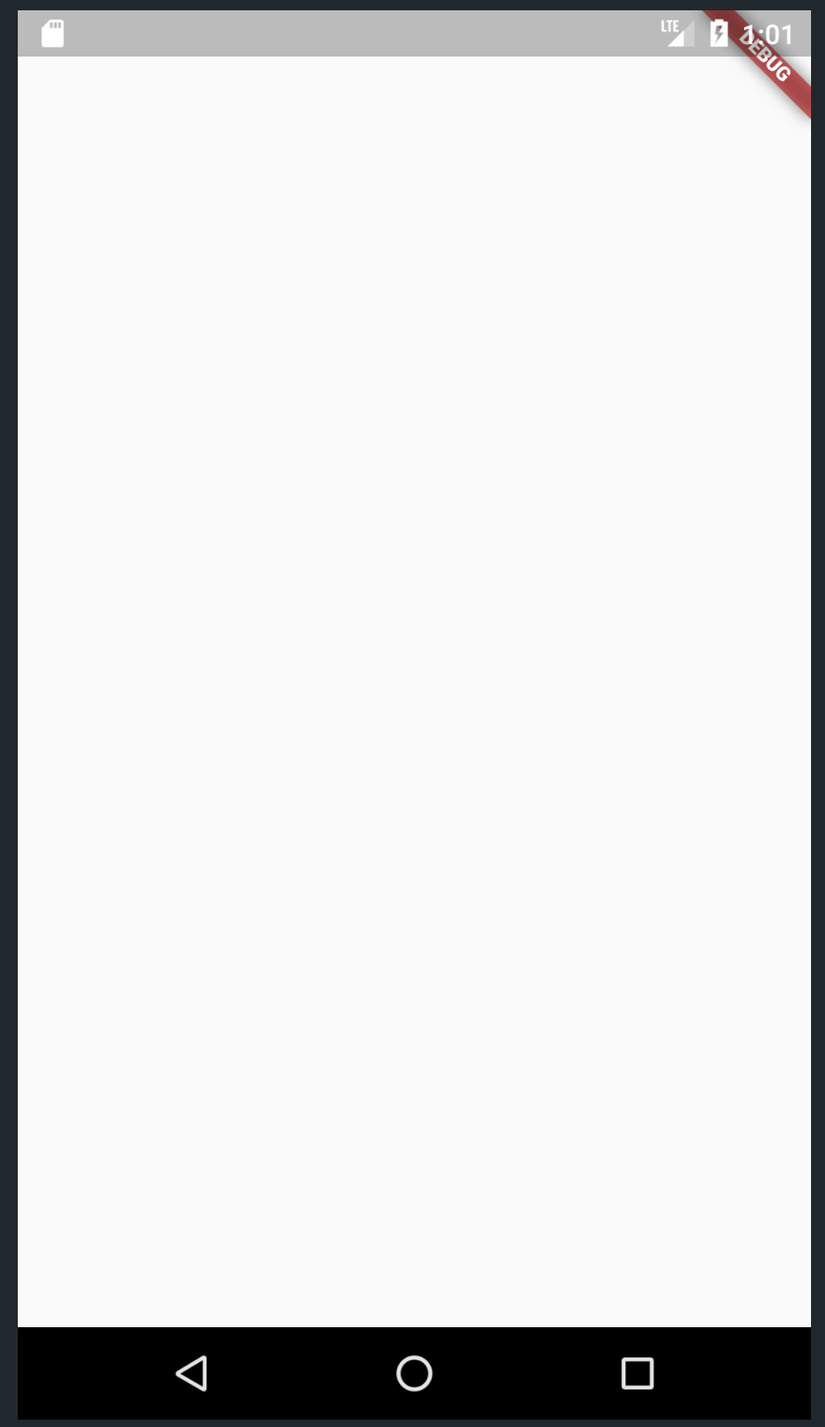
Who uses Flutter?

Reflectly, 500k+ users

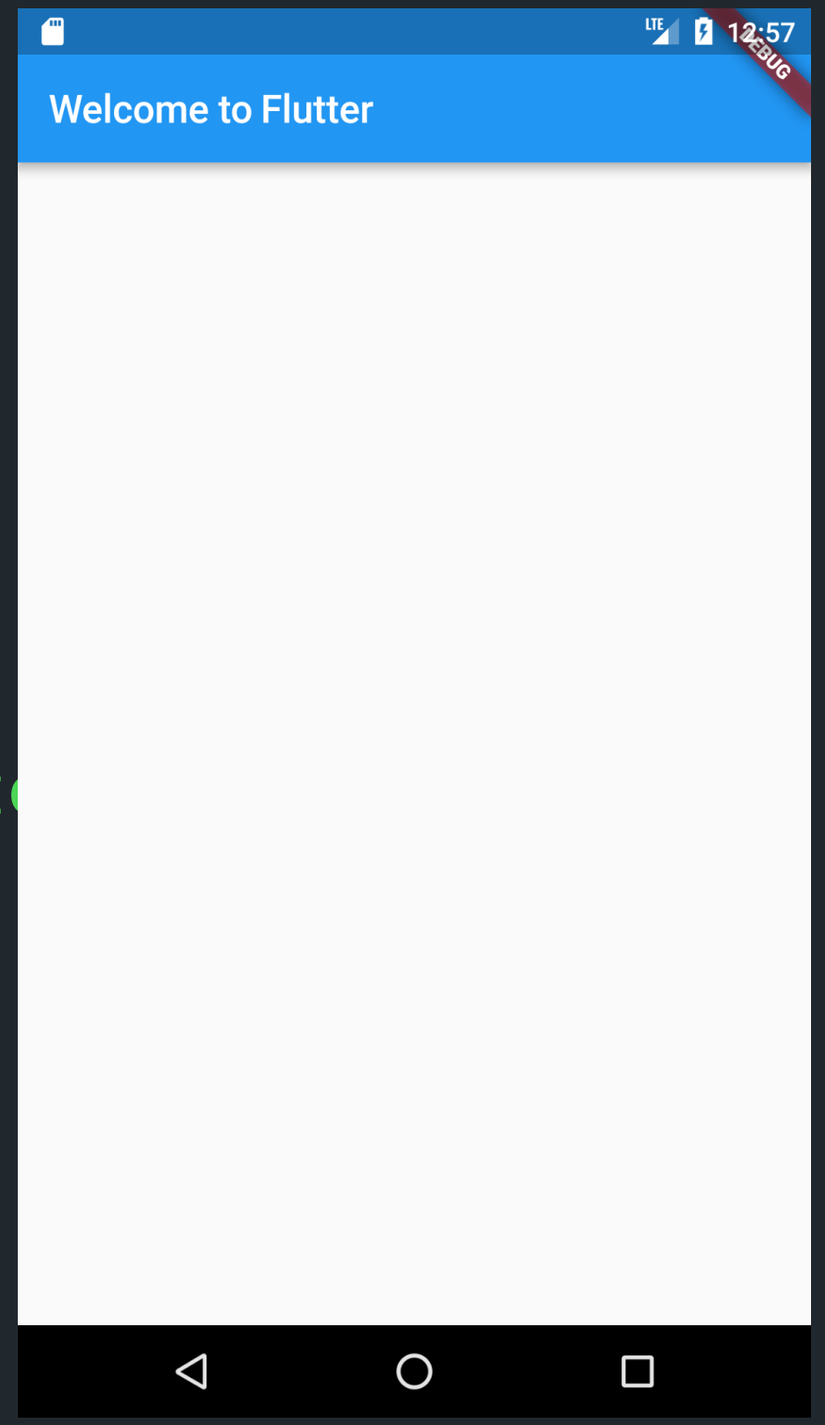
How to make an app?

```dart
import 'package:flutter/material.dart';

void main() => runApp(MyApp());
```

```dart
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'FlutterApp',
      home: Scaffold(

      ),
    );
  }
}
```



```dart
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'FlutterApp',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Welcome to Flutte
        ),
      ),
    );
  }
}
```



```dart
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'FlutterApp',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Welcome to Flutter'),
        ),
        body: MyAppBody(),
      ),
    );
  }
}
```

```dart
class MyAppBody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
  }
}
```

```dart
class MyAppBody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
      ],
    );
  }
}
```

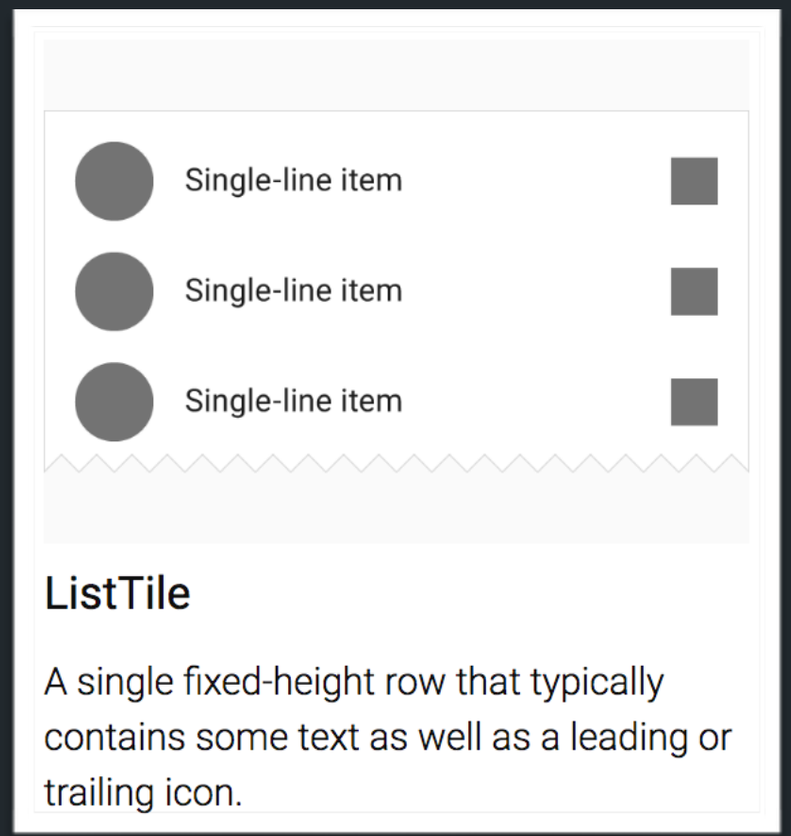
```dart
class MyAppBody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Flexible(
        ),
      ],
    );
  }
}
```

```dart
class MyAppBody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Flexible(
          child: ListView.builder(
          ),
        ),
      ],
    );
  }
}
```

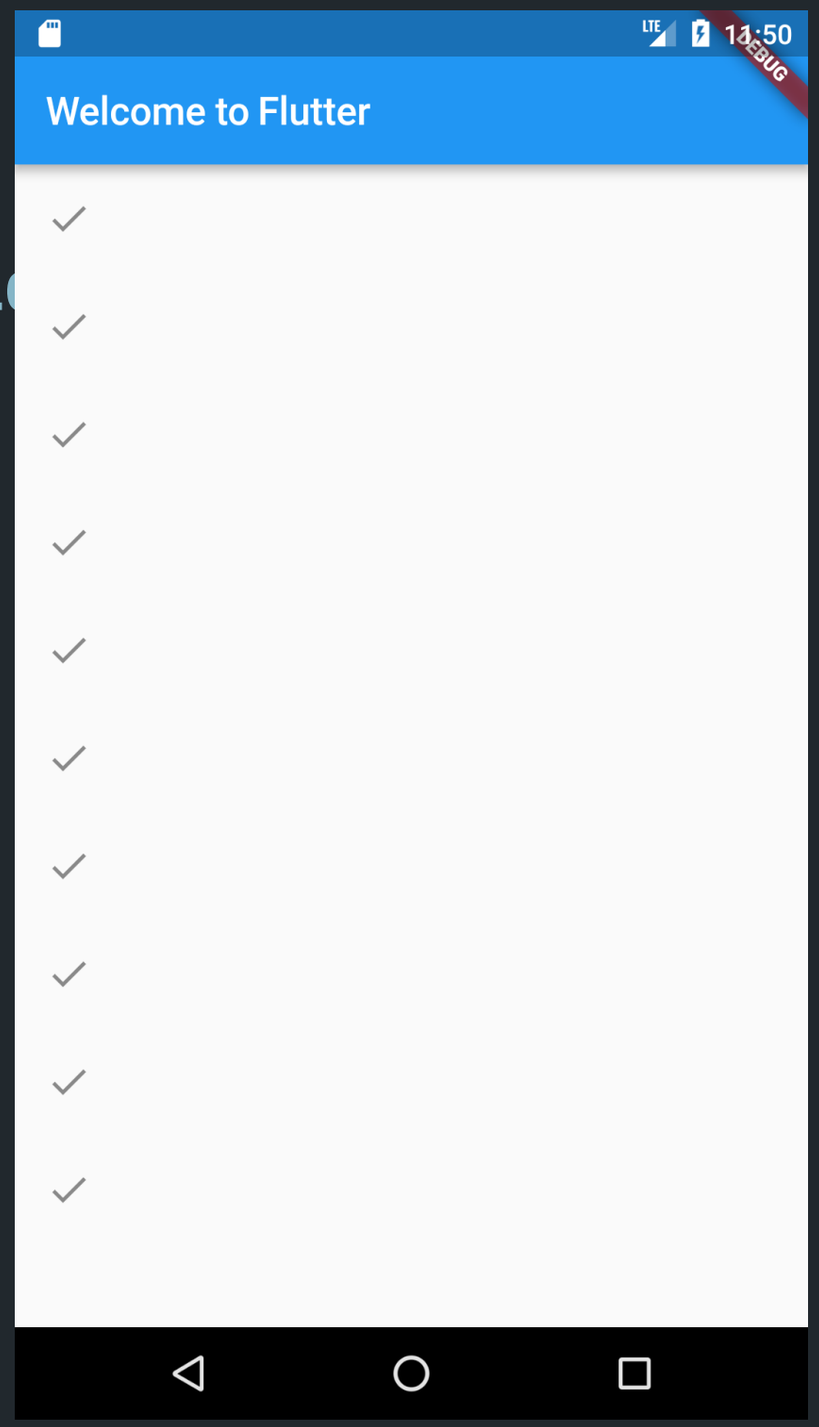
```dart
class MyAppBody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Flexible(
          child: ListView.builder(
            itemCount: 10,
          ),
        ),
      ],
    );
  }
}
```

```dart
class MyAppBody extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        Flexible(
          child: ListView.builder(
            itemCount: 10,
            itemBuilder: (_, position) =>
                MyAppListItem(position),
          ),
        ),
      ],
    );
  }
}
```

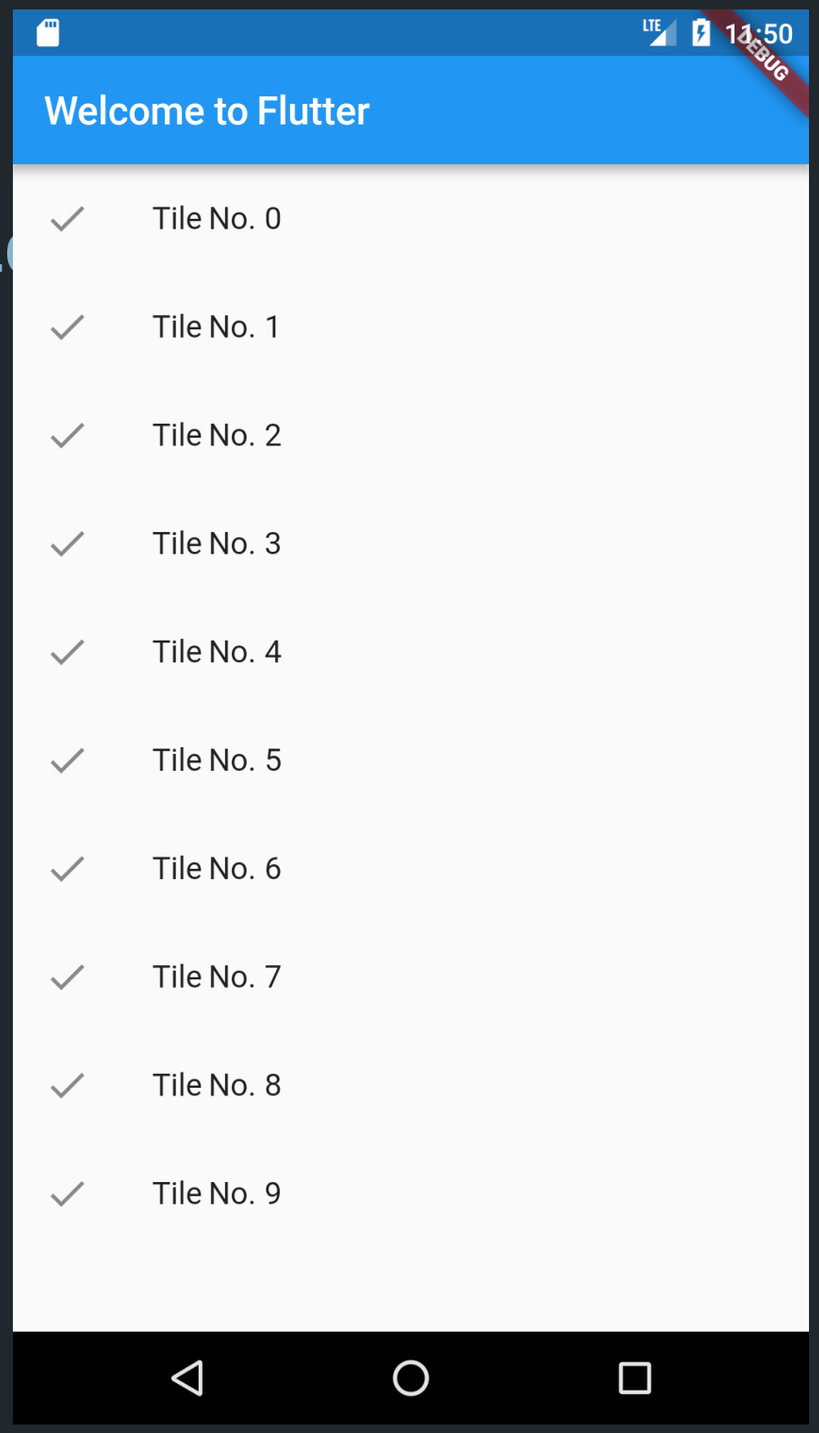
```dart
class MyAppListItem extends StatelessWidget {
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return ListTile(

    );
  }
}
```



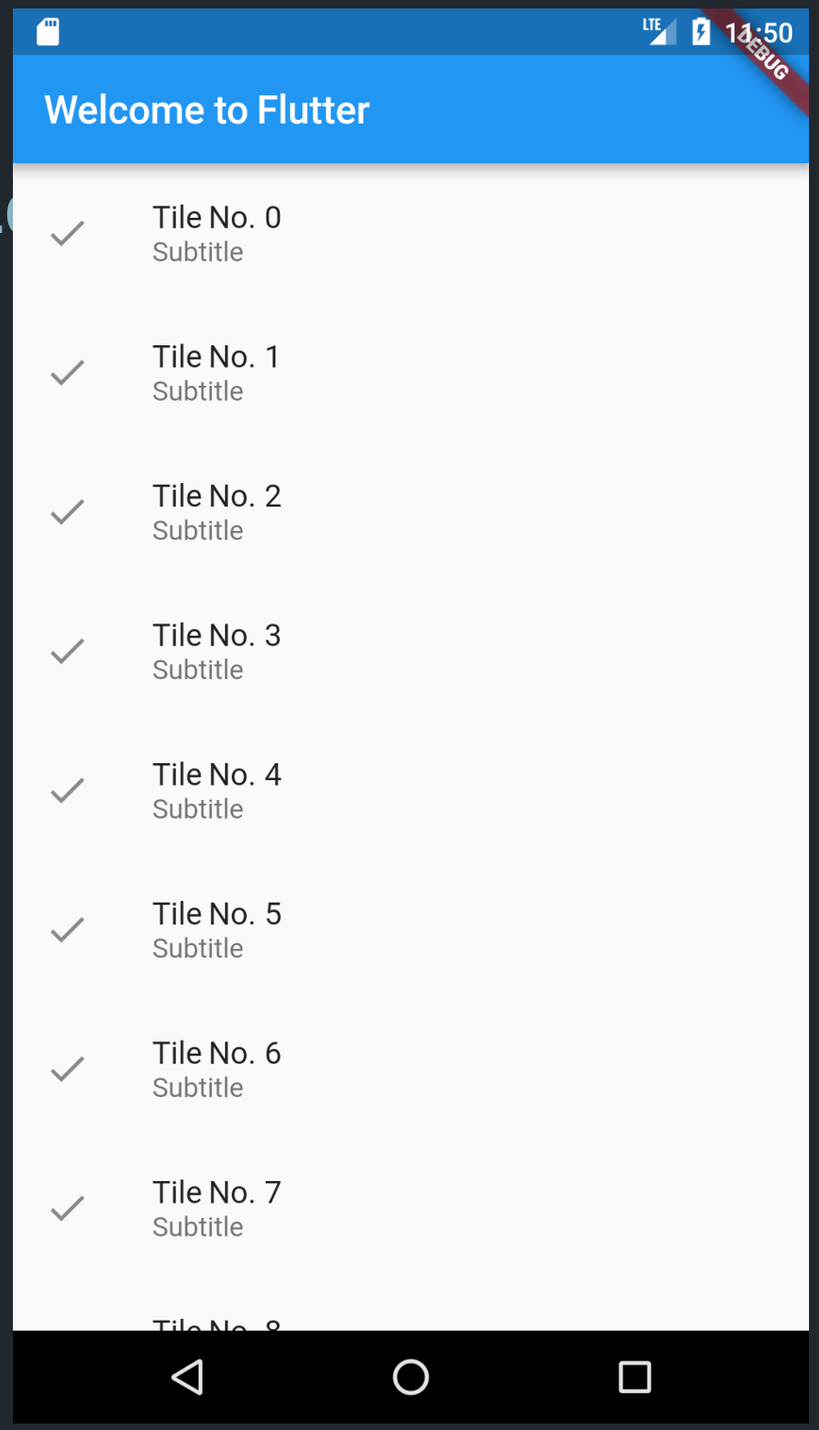
```dart
class MyAppListItem extends StatelessWi
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return ListTile(
      leading: Icon(Icons.done),
    );
  }
}
```



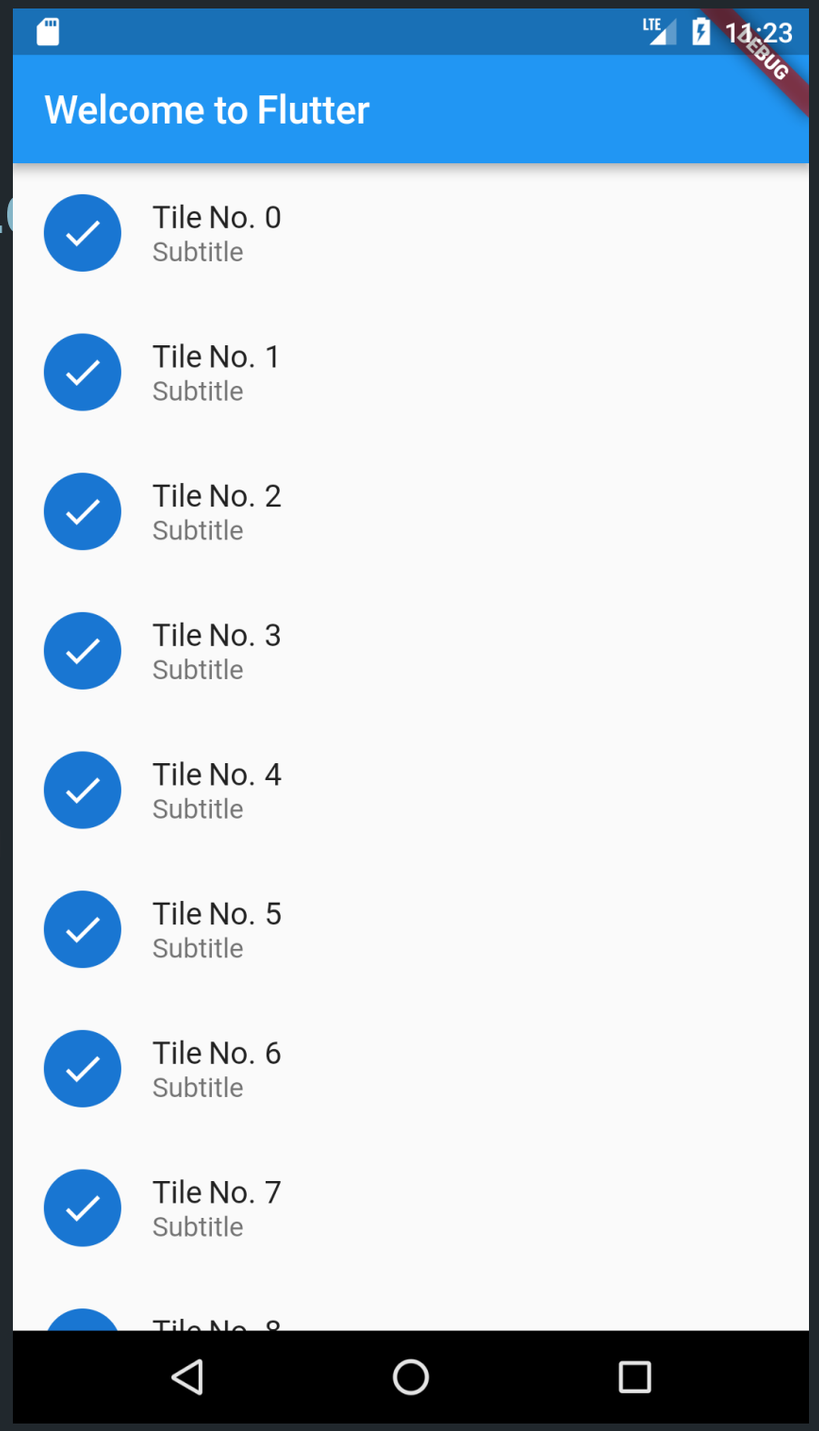
```dart
class MyAppListItem extends StatelessWi
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return ListTile(
      leading: Icon(Icons.done),
      title: Text("Tile No. $position")
    );
  }
}
```



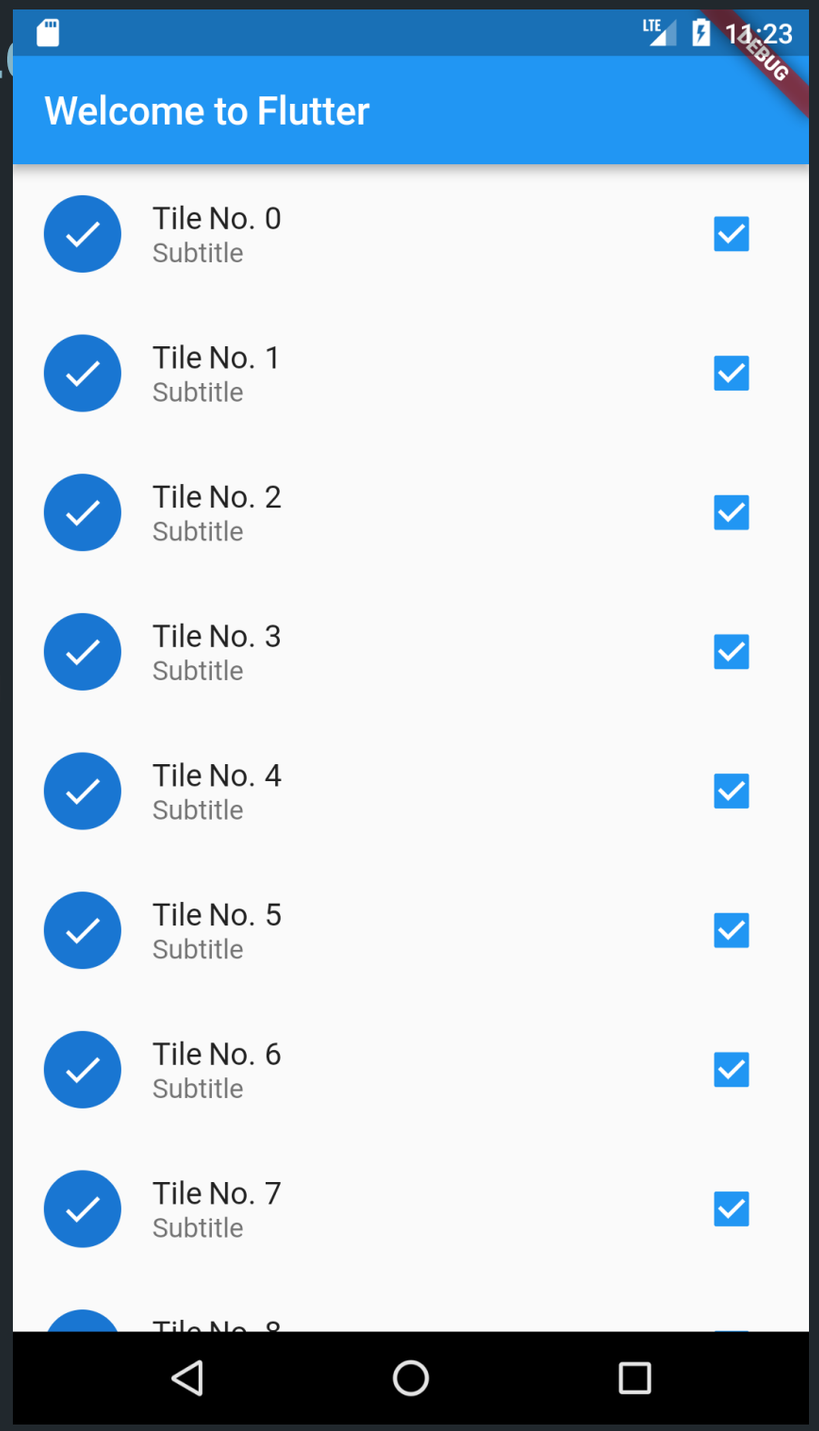
```dart
class MyAppListItem extends StatelessWi
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return ListTile(
      leading: Icon(Icons.done),
      title: Text("Tile No. $position")
      subtitle: Text("Subtitle"),
    );
  }
}
```



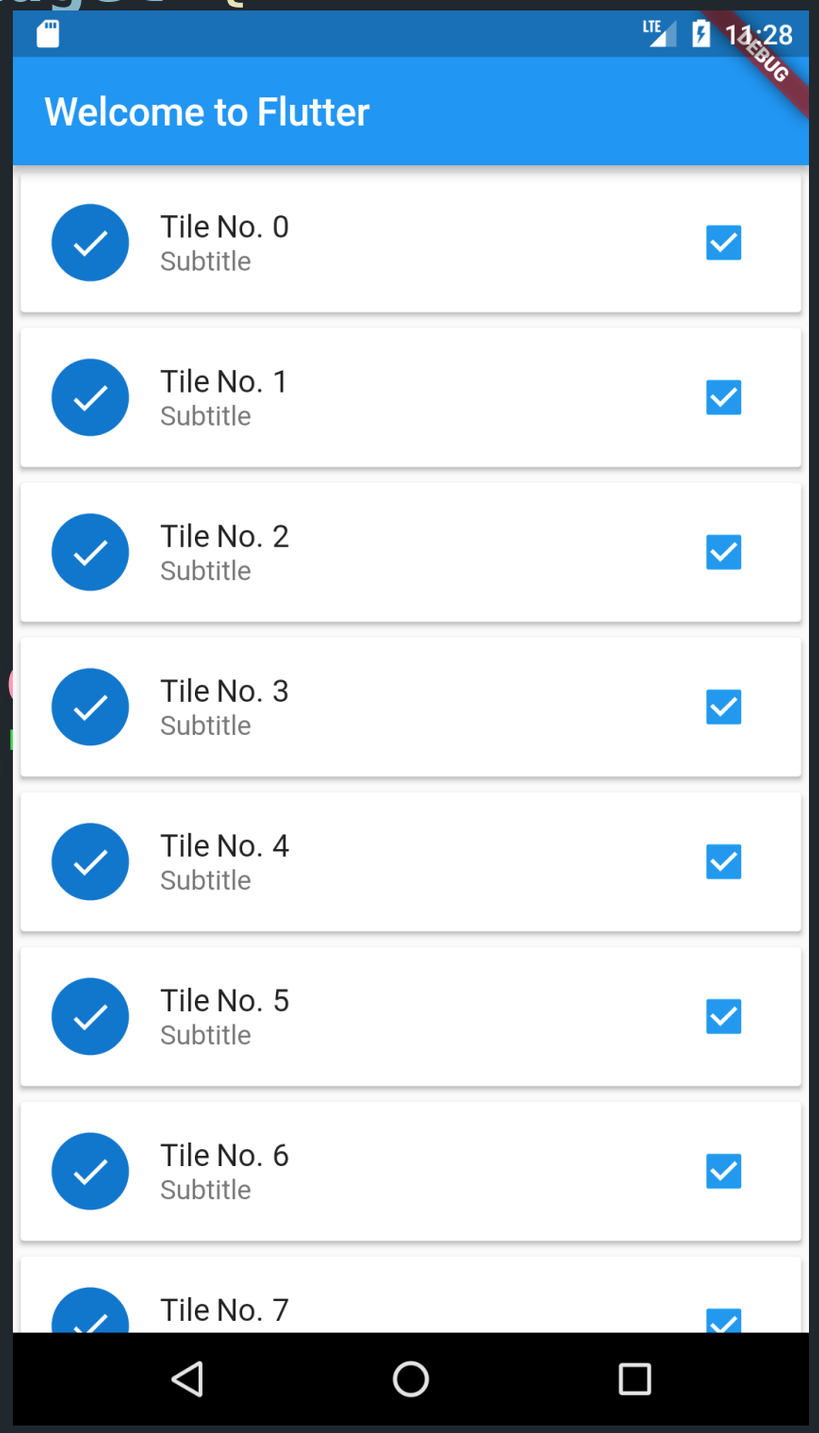
```dart
class MyAppListItem extends StatelessWi
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return ListTile(
      leading: CircleAvatar(child: Icon
      title: Text("Tile No. $position")
      subtitle: Text("Subtitle"),
    );
  }
}
```



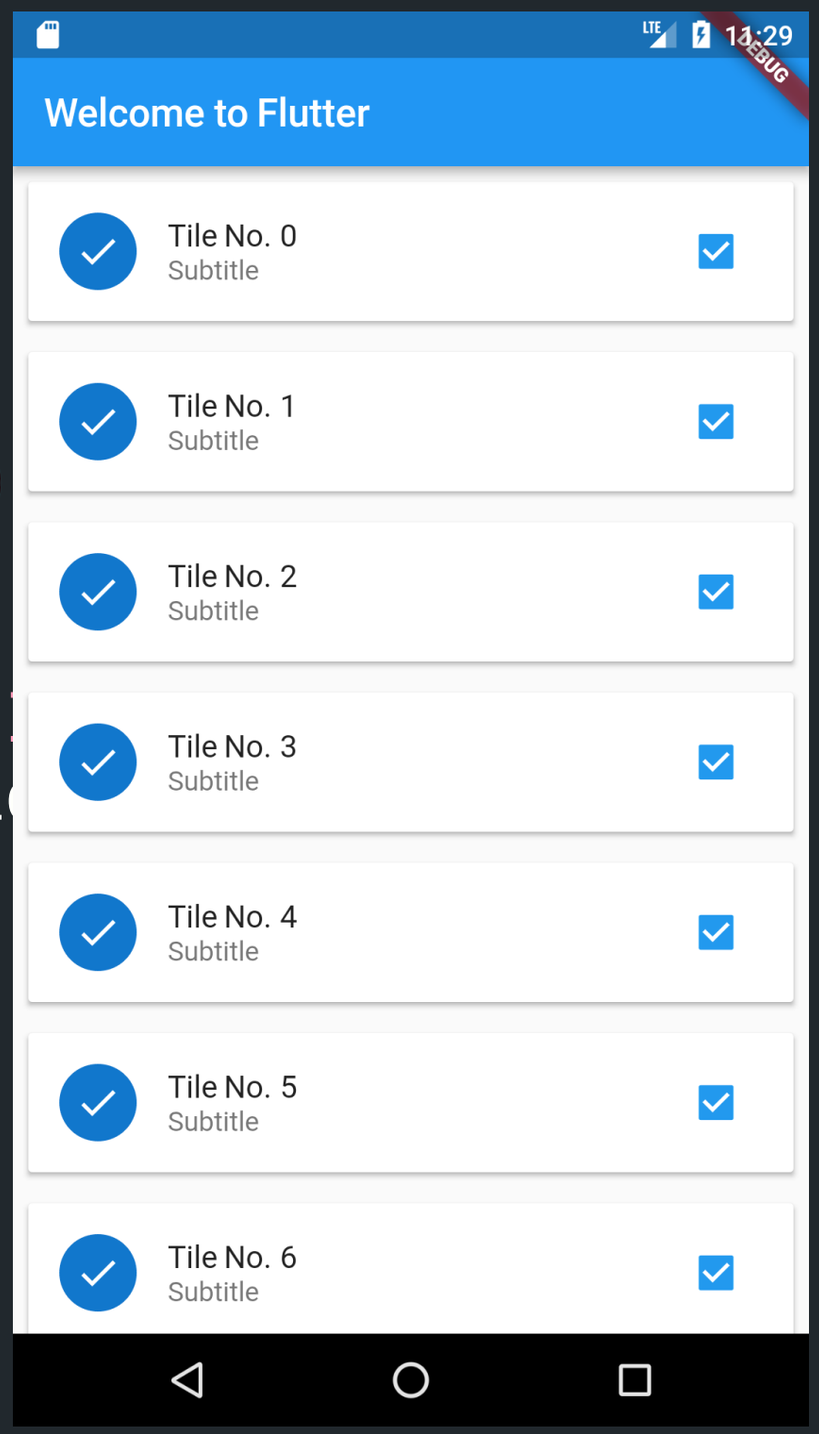
```dart
class MyAppListItem extends StatelessWi
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return ListTile(
      leading: CircleAvatar(child: Icon
      title: Text("Tile No. $position")
      subtitle: Text("Subtitle"),
      trailing: Checkbox(
        value: true,
        onChanged: handleCheck,
      ),
    );
  }
}
```



```dart
class MyAppListItem extends StatelessWidget {
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return Card(
      child: ListTile(
        leading: CircleAvatar(child: Ic
        title: Text("Tile No. $position
        subtitle: Text("Subtitle"),
        trailing: Checkbox(
          value: true,
          onChanged: handleCheck,
        ),
      ),
    );
  }
}
```

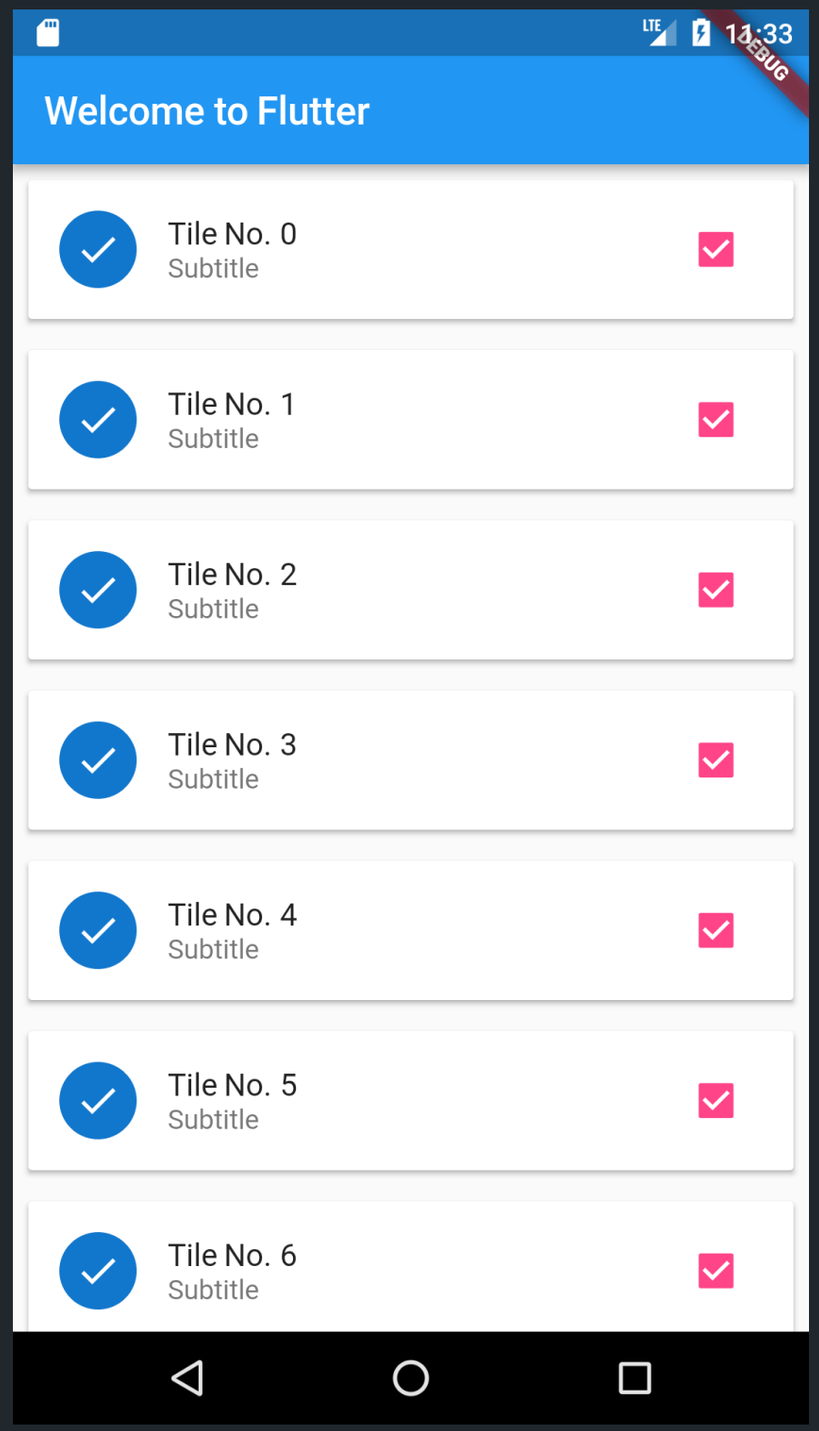


```dart
class MyAppListItem extends StatelessWidget {
  final int position;

  MyAppListItem(this.position);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(4.0
      child: Card(
        child: ListTile(
          leading: CircleAvatar(child:
          title: Text("Tile No. $positi
          subtitle: Text("Subtitle"),
          trailing: Checkbox(
            value: true,
            onChanged: handleCheck,
          ),
        ),
      ),
    );
  }
}
```

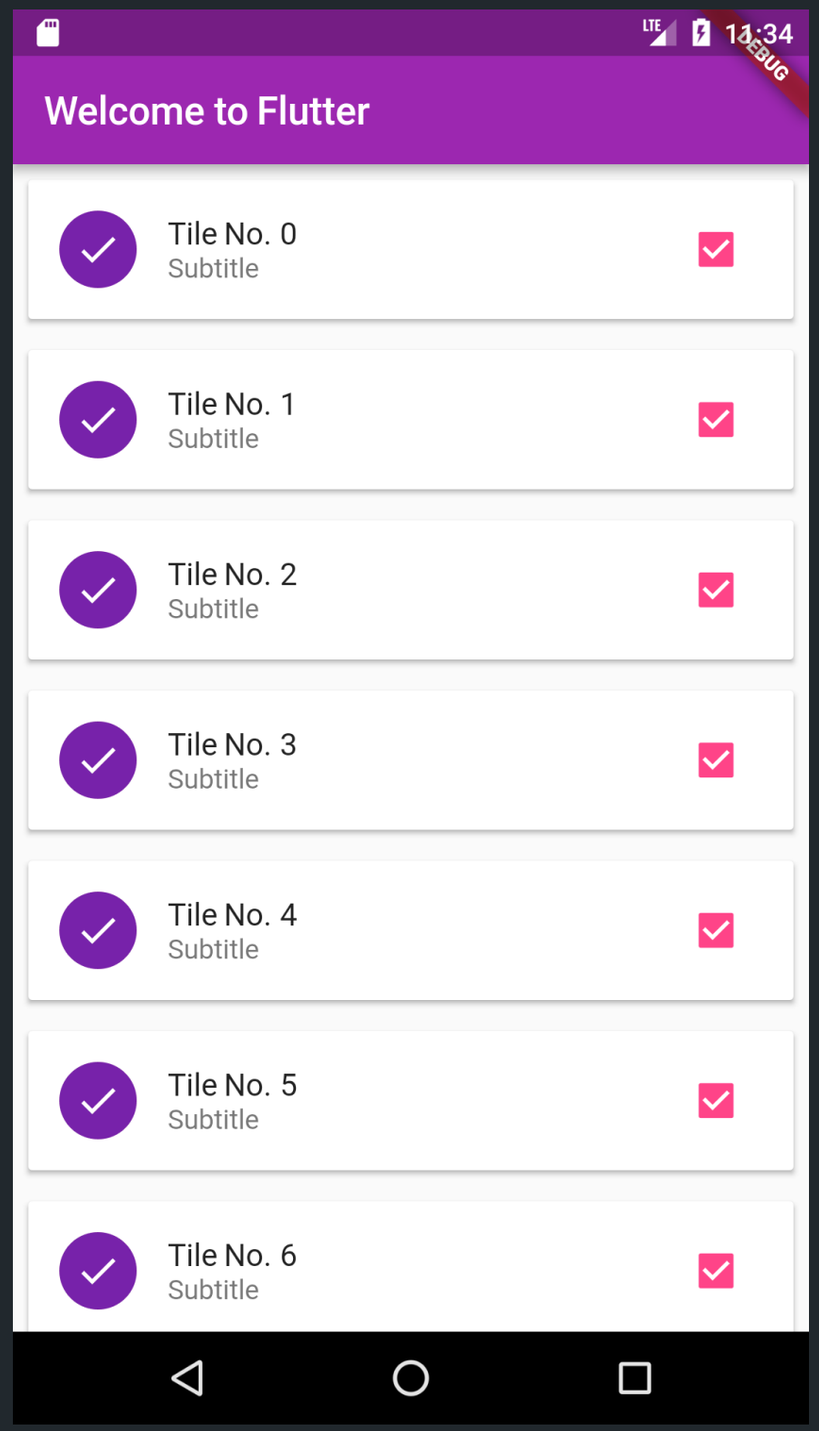


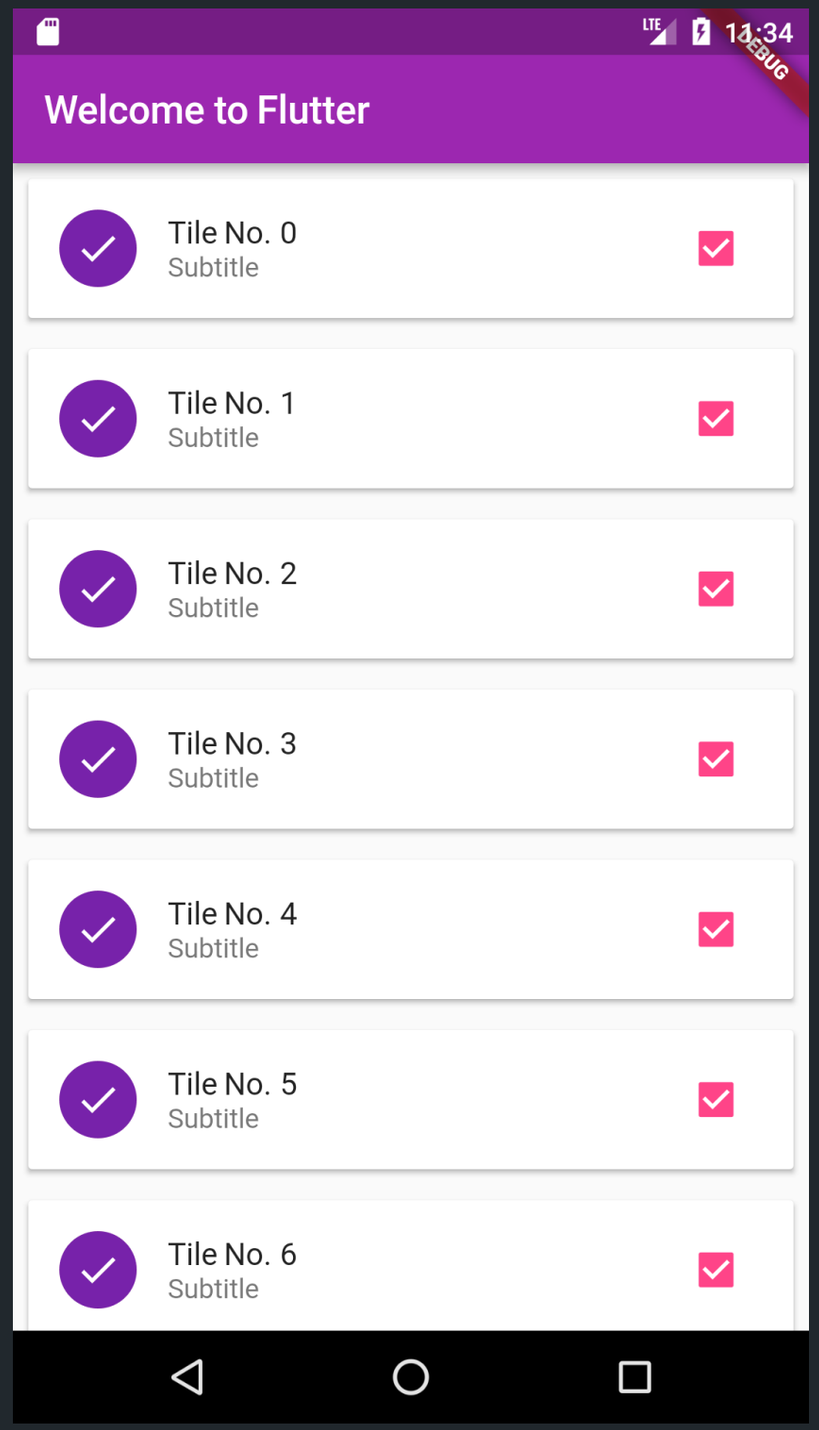
```dart
return MaterialApp(
  title: 'FlutterApp',
  home: Scaffold(
    ...
  ),
);
```

```dart
return MaterialApp(
  title: 'FlutterApp',
  theme: ThemeData(
    primarySwatch: Colors.blue,
    accentColor: Colors.pinkAccent,
  ),
  home: Scaffold(
    ...
 ),
);
```

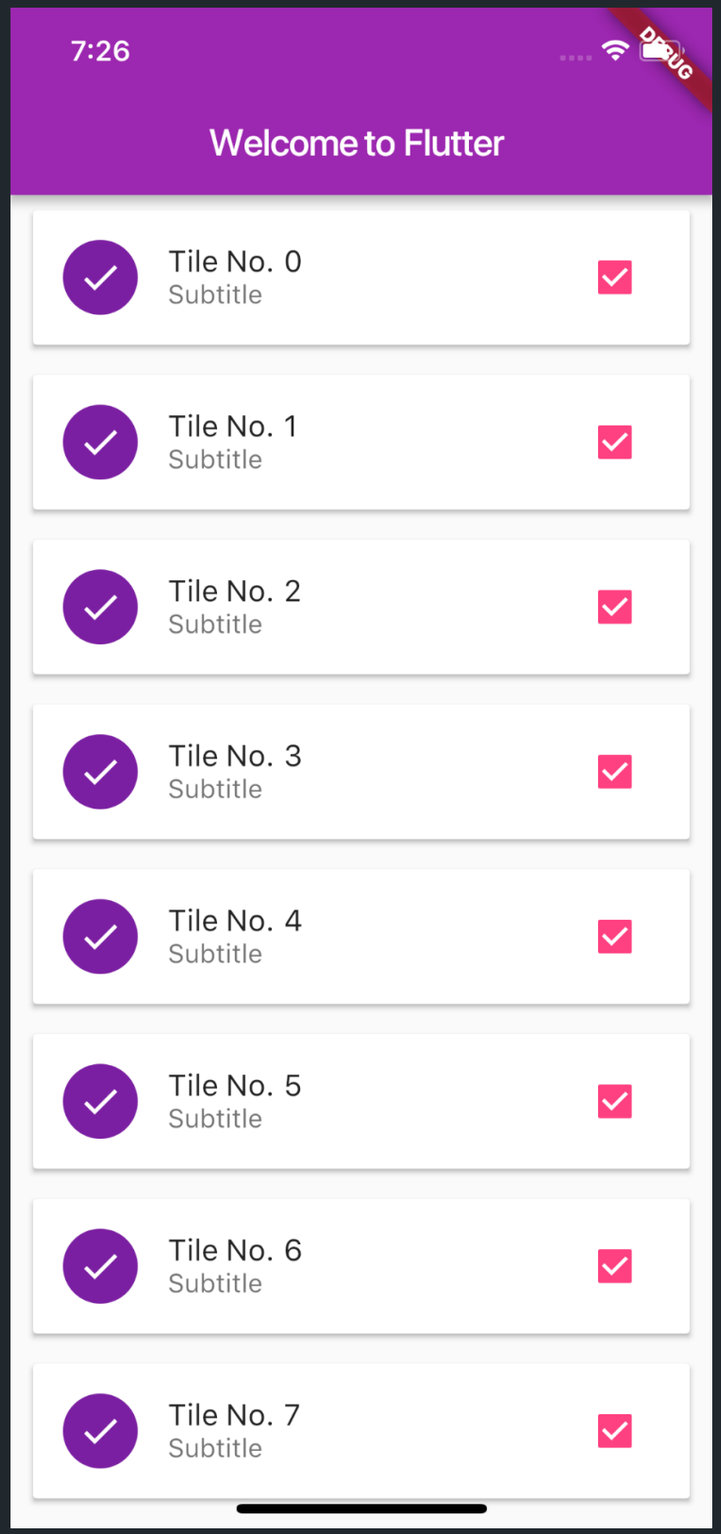


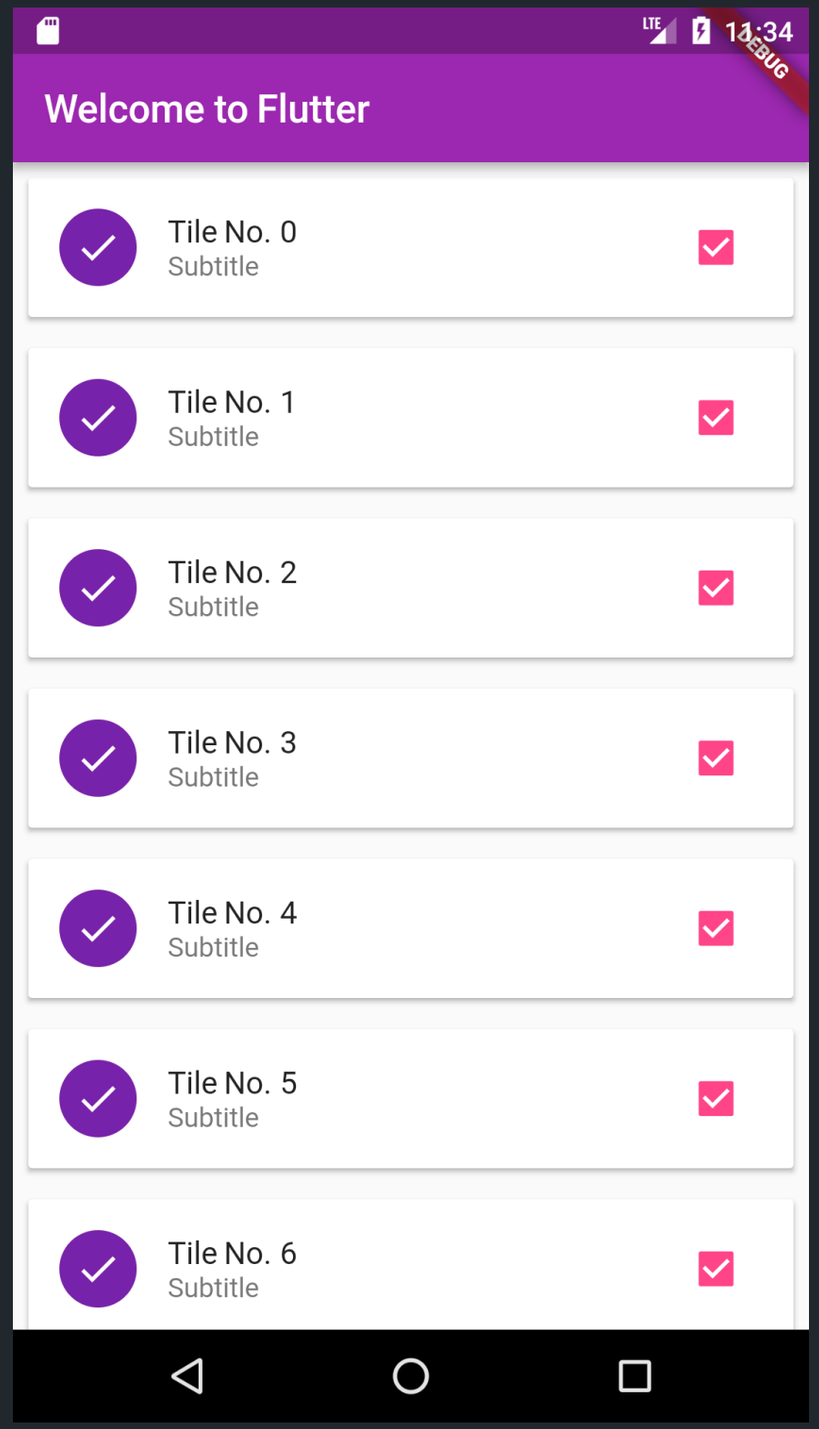
```dart
return MaterialApp(
  title: 'FlutterApp',
  theme: ThemeData(
    primarySwatch: Colors.purple,
    accentColor: Colors.pinkAccent,
  ),
  home: Scaffold(
    ...
 ),
);
```



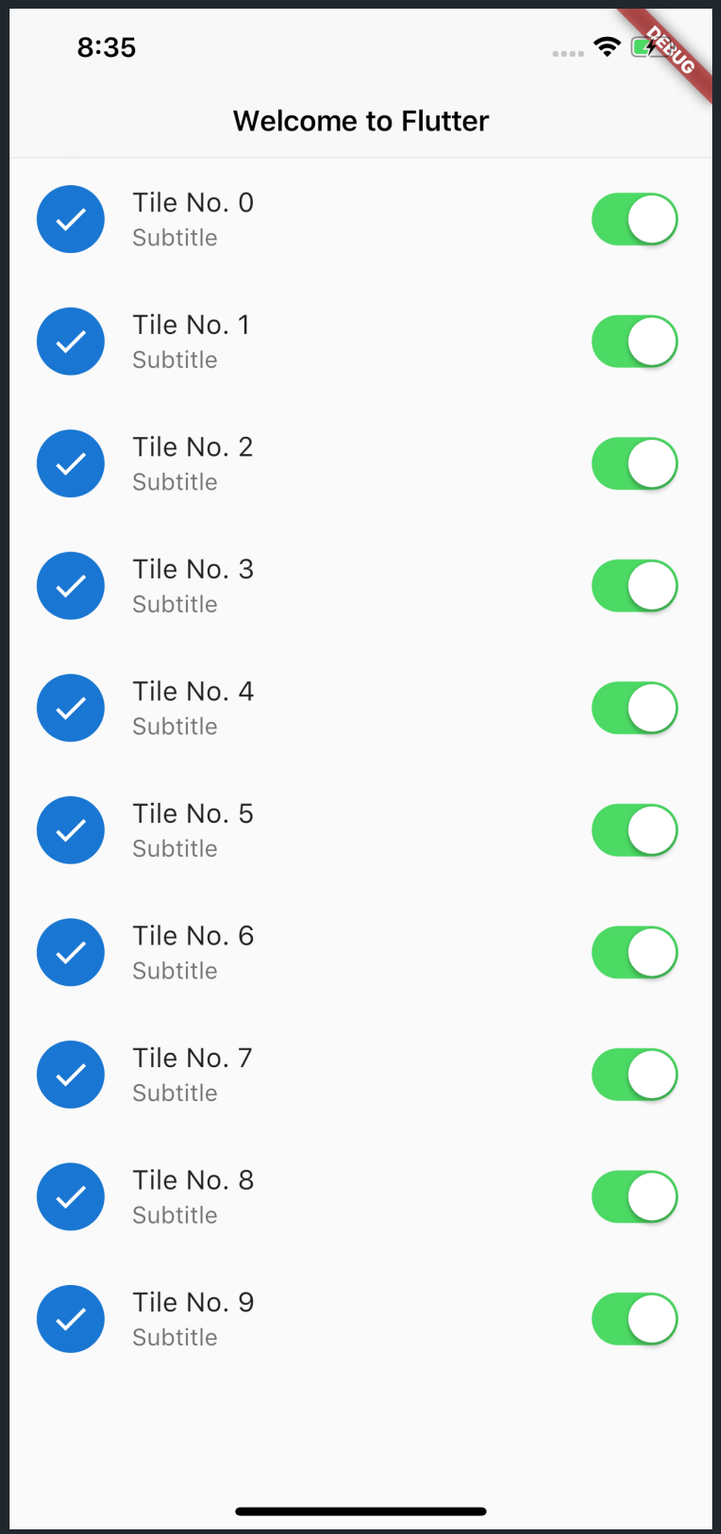

11:34
DEBUG
Welcome to Flutter
Tile No. 0
Subtitle
Tile No. 1
Subtitle
Tile No. 2
Subtitle
Tile No. 3
Subtitle
Tile No. 4
Subtitle
Tile No. 5
Subtitle
Tile No. 6
Subtitle

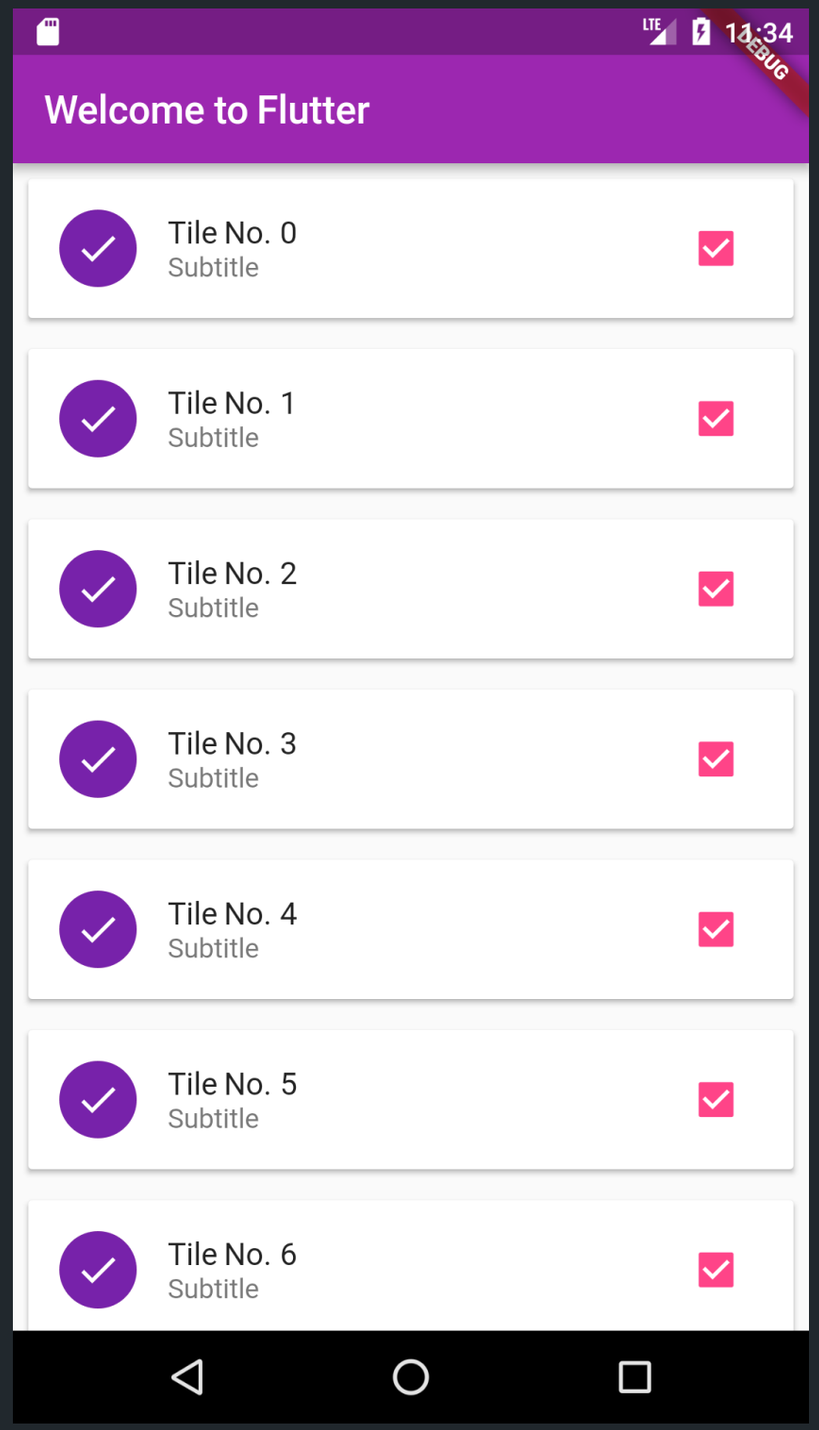
@p_szklarska


7:26
DEBUG
Welcome to Flutter
Tile No. 0
Subtitle
Tile No. 1
Subtitle
Tile No. 2
Subtitle
Tile No. 3
Subtitle
Tile No. 4
Subtitle
Tile No. 5
Subtitle
Tile No. 6
Subtitle
Tile No. 7
Subtitle


11:34
DEBUG
Welcome to Flutter
Tile No. 0
Subtitle
Tile No. 1
Subtitle
Tile No. 2
Subtitle
Tile No. 3
Subtitle
Tile No. 4
Subtitle
Tile No. 5
Subtitle
Tile No. 6
Subtitle

@p_szklarska


8:35
DEBUG
Welcome to Flutter
Tile No. 0
Subtitle
Tile No. 1
Subtitle
Tile No. 2
Subtitle
Tile No. 3
Subtitle
Tile No. 4
Subtitle
Tile No. 5
Subtitle
Tile No. 6
Subtitle
Tile No. 7
Subtitle
Tile No. 8
Subtitle
Tile No. 9
Subtitle


11:34
DEBUG
Welcome to Flutter
Tile No. 0
Subtitle
Tile No. 1
Subtitle
Tile No. 2
Subtitle
Tile No. 3
Subtitle
Tile No. 4
Subtitle
Tile No. 5
Subtitle
Tile No. 6
Subtitle

@p_szklarska

Testing?

@p_szklarska

```dart
void main() {
  testWidgets('widget has first and last elements',
          (WidgetTester tester) async {

    await tester.pumpWidget(MyApp());
    expect(find.byType(ListView), findsOneWidget);
    expect(find.text("Title No. 0"), findsOneWidget);

    await tester.drag(find.byType(ListView), Offset(0.0, -200.0));
    await tester.pump();

    expect(find.text("Title No. 9"), findsOneWidget);
  });

}
```

How does it work?

How does it work?

source: https://flutter.io/technical-overview/

Building app process

~ 2.8MB

Flutter engine
(C/C++)

~ 4.2MB

APK file

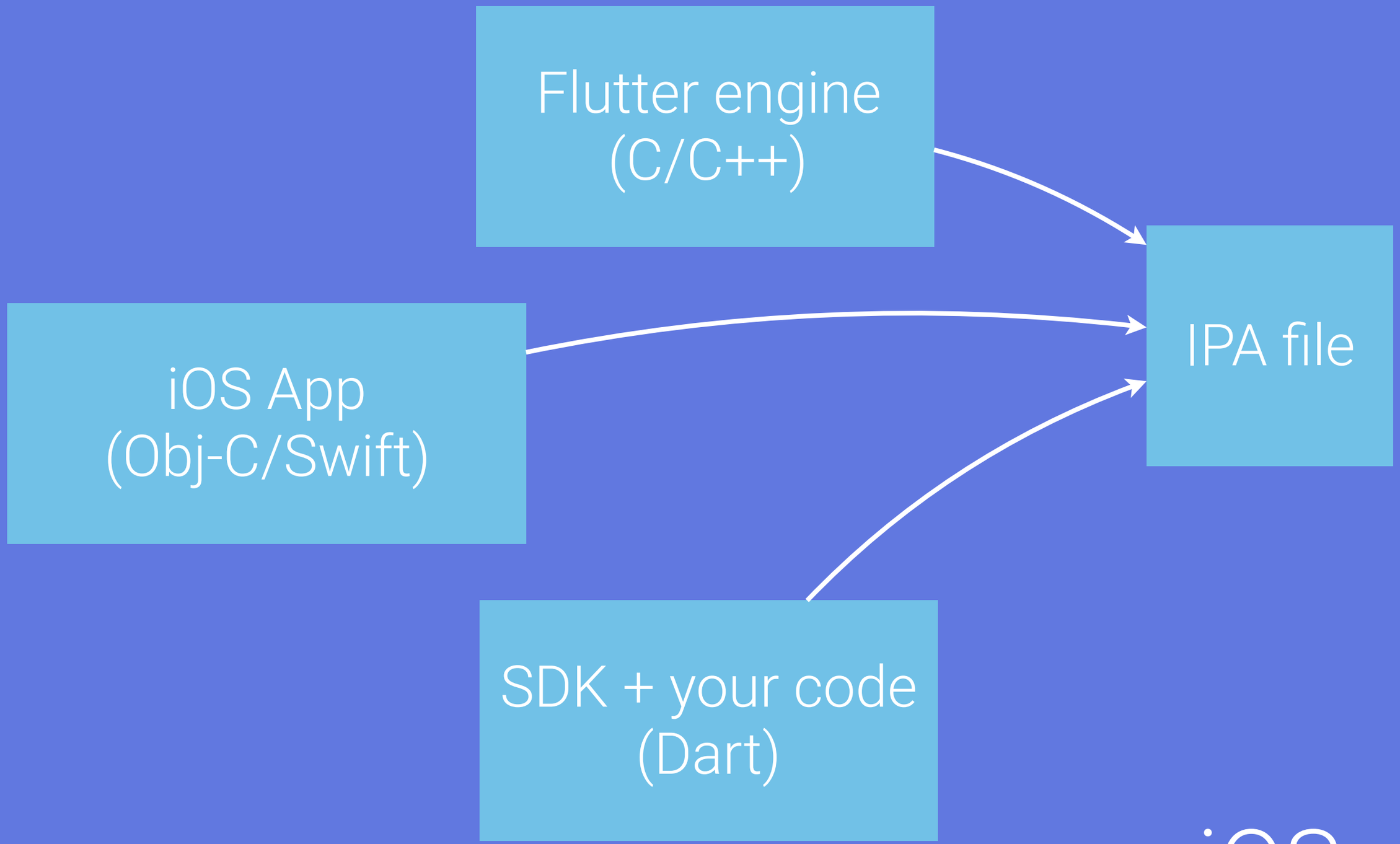
Android App
(Java/Kotlin)

~ 1MB

SDK + your code
(Dart)

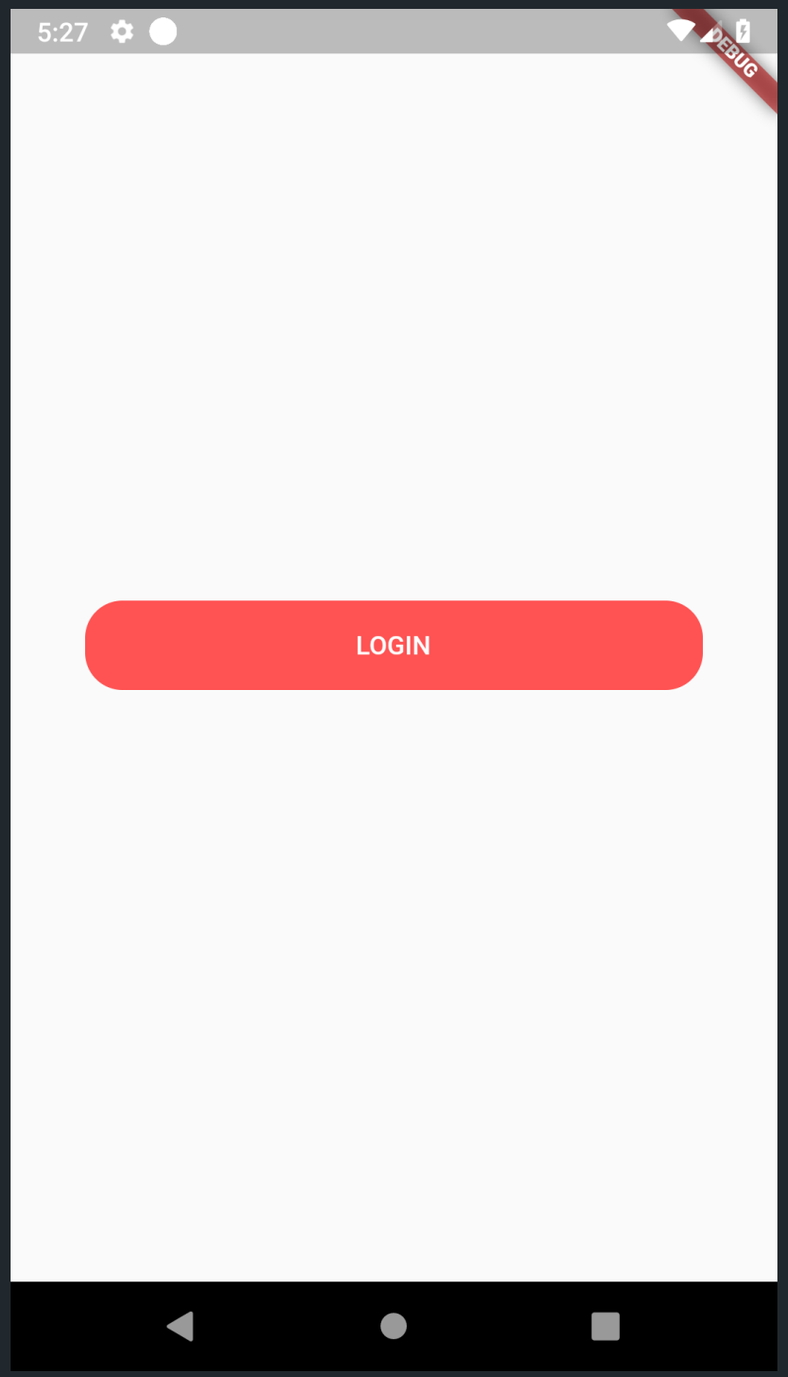
Android

Building app process



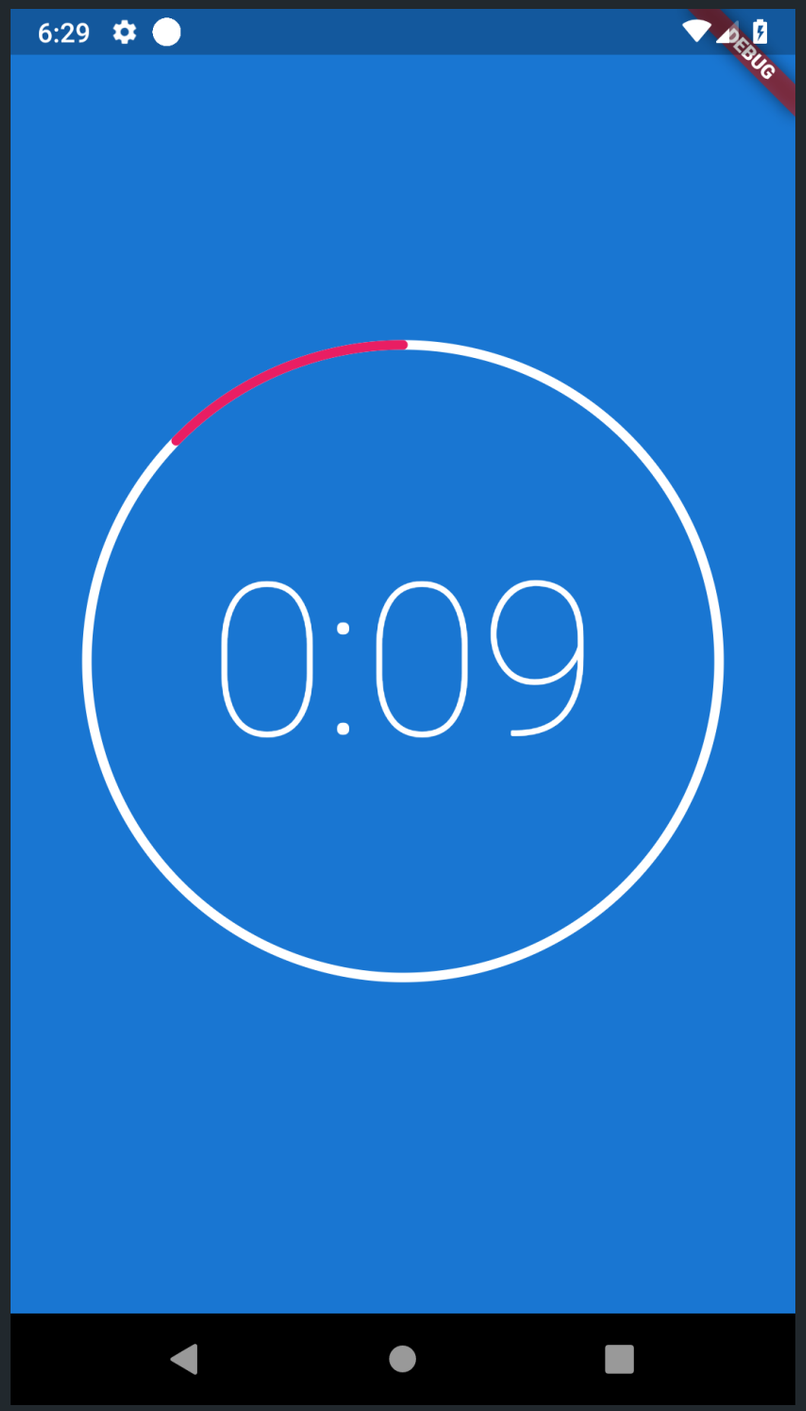
Cool, but...

What about custom UI?


5:27
DEBUG
LOGIN

```dart
class SimpleButton extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return FlatButton(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(20.0),
      ),
      color: Colors.redAccent,
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Text(
          "LOGIN",
          style: TextStyle(color: Colors.white),
        ),
      ),
      onPressed: () => {},
    );
  }
}
```

```dart
class SimpleButton extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return FlatButton(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(20.0),
      ),
      color: Colors.redAccent,
      child: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Text(
          "LOGIN",
          style: TextStyle(color: Colors.white),
        ),
      ),
      onPressed: () => {},
    );
  }
}
```


6:29
DEBUG
0:09

```dart
class ProgressPainter extends CustomPainter {

  @override
  void paint(Canvas canvas, Size size) {
    Paint paint = new Paint()
      ..color = Colors.white
      ..strokeWidth = 5.0
      ..strokeCap = StrokeCap.round
      ..style = PaintingStyle.stroke;

    Offset center = size.center(Offset.zero);
    double radius = size.width / 2.0;
    canvas.drawCircle(center, radius, paint);

    paint.color = Colors.pink;
    paint.style = PaintingStyle.stroke;
    canvas.drawArc(
        Rect.fromCircle(center: center, radius: radius),
        math.pi * 1.5, -0.8, false, paint);
  }

}
```

What about camera, BLE, other services?

Accessing external services

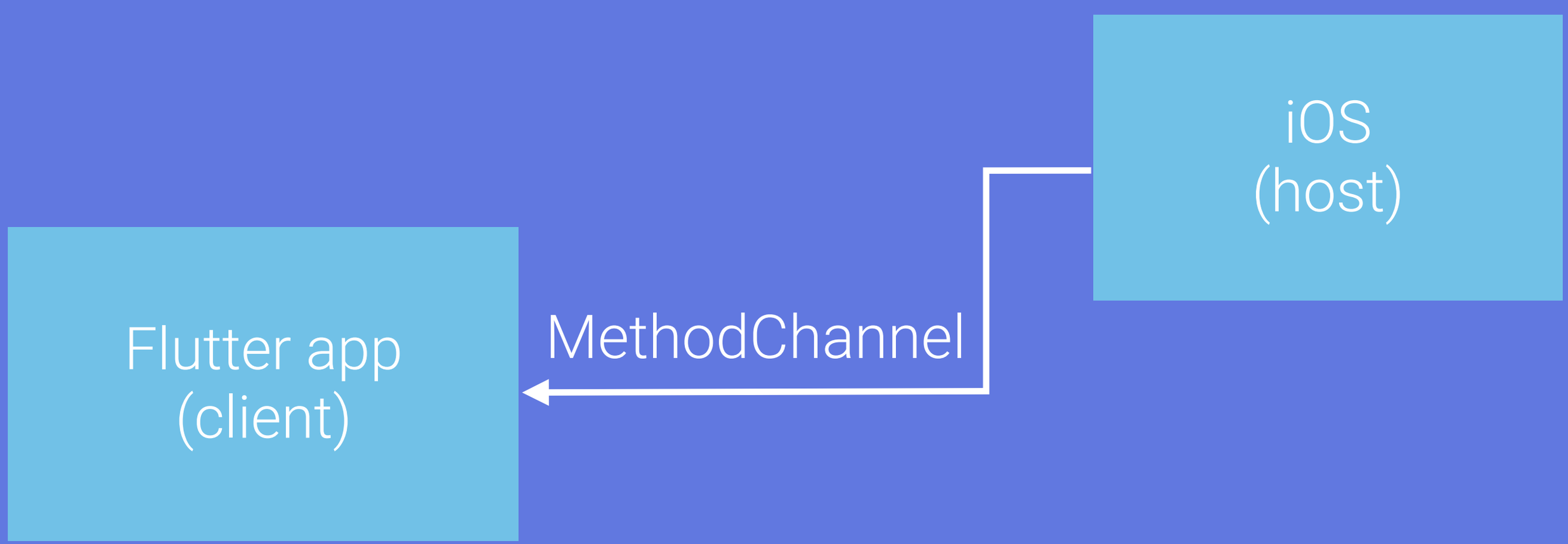
MethodChannel

Accessing external services

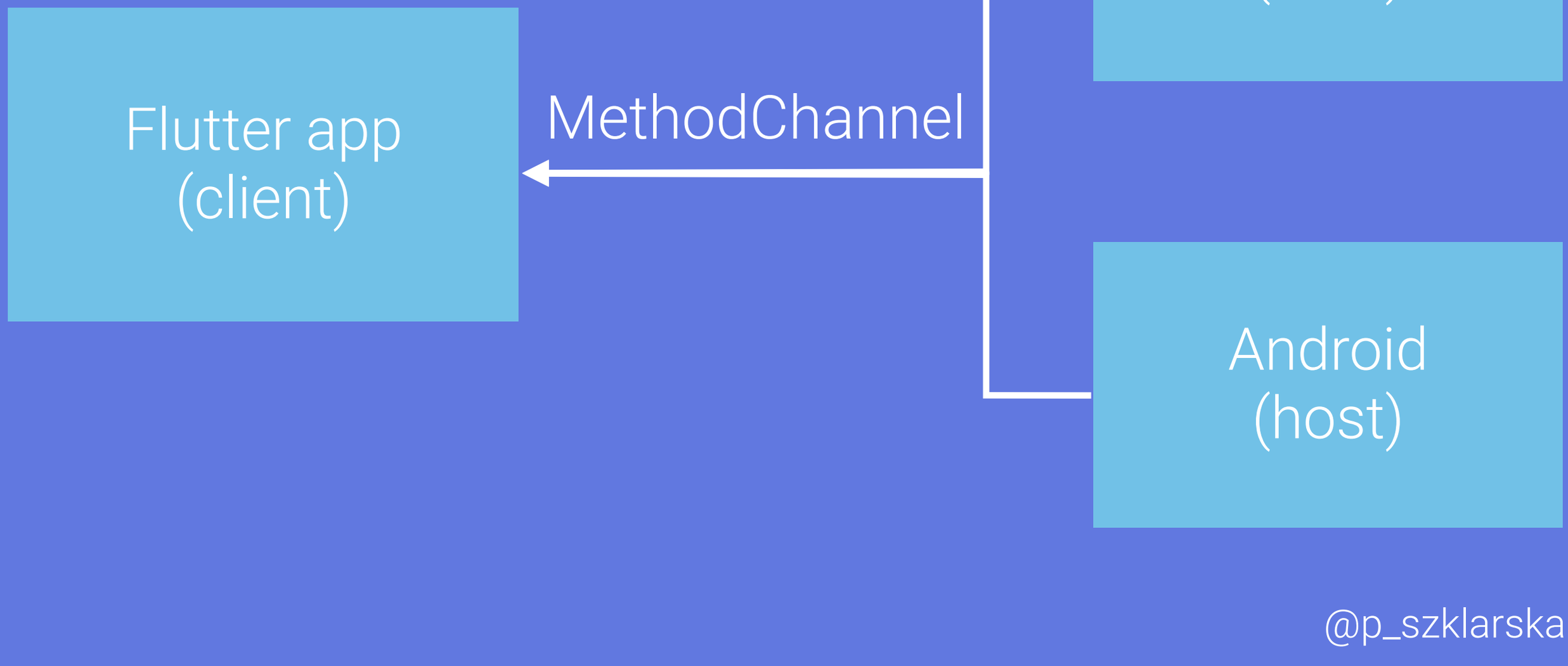
Flutter app
(client)

MethodChannel

Accessing external services



Accessing external services



```
class MainActivity() : FlutterActivity() {
  private val CHANNEL = „samples.flutter.io/battery”

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    GeneratedPluginRegistrant.registerWith(this)

    MethodChannel(flutterView, CHANNEL)
            .setMethodCallHandler { call, result ->
              // TODO
            }
  }
}
```

```
class MainActivity() : FlutterActivity() {
  private val CHANNEL = „samples.flutter.io/battery”

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    GeneratedPluginRegistrant.registerWith(this)

    MethodChannel(flutterView, CHANNEL)
            .setMethodCallHandler { call, result ->
              // TODO
            }
  }
}
```

```
if (call.method.equals("getBatteryLevel")) {
  int batteryLevel = getBatteryLevel();
  if (batteryLevel != -1) {
    result.success(batteryLevel);
  } else {
    result.error("UNAVAILABLE",
            "Battery level not available.", null);
  }
} else {
  result.notImplemented();
}
```

```
if (call.method.equals("getBatteryLevel")) {
  int batteryLevel = getBatteryLevel();
  if (batteryLevel != -1) {
    result.success(batteryLevel);
  } else {
    result.error("UNAVAILABLE",
            "Battery level not available.", null);
  }
} else {
  result.notImplemented();
}
```

```
if (call.method.equals("getBatteryLevel")) {
  int batteryLevel = getBatteryLevel();
  if (batteryLevel != -1) {
    result.success(batteryLevel);
  } else {
    result.error("UNAVAILABLE",
            "Battery level not available.", null);
  }
} else {
  result.notImplemented();
}
```

```
if (call.method.equals("getBatteryLevel")) {
  int batteryLevel = getBatteryLevel();
  if (batteryLevel != -1) {
    result.success(batteryLevel);
  } else {
    result.error("UNAVAILABLE",
        "Battery level not available.", null);
  }
} else {
  result.notImplemented();
}
```

```
const platform =
        const MethodChannel('samples.flutter.io/battery');
final int result =
        await platform.invokeMethod('getBatteryLevel');
String batteryLevel = "Battery level at $result % .";
```

Dart Packages

firebase_auth

FLUTTER

Flutter plugin for Firebase Auth, enabling Android and iOS authentication using passwords, and identity providers like Google, Facebook, and Twitter.

path_provider

FLUTTER

Flutter plugin for getting commonly used locations on the Android & iOS file systems, such as the temp and app data directories.

url_launcher

FLUTTER

Flutter plugin for launching a URL on Android and iOS. Supports web, phone, SMS, and email schemes.

google_sign_in

FLUTTER

Flutter plugin for Google Sign-In, a secure authentication system for signing in with a Google account on Android and iOS.

shared_preferences

FLUTTER

Flutter plugin for reading and writing simple key-value pairs. Wraps NSUserDefaults on iOS and SharedPreferences on Android.

firebase_database

FLUTTER

Flutter plugin for Firebase Database, a cloud-hosted NoSQL database with realtime data syncing across Android and iOS clients, and offline access.

image_picker

FLUTTER

Flutter plugin for selecting images from the Android and iOS image library, and taking new pictures with the camera.

firebase_analytics

FLUTTER

Flutter plugin for Google Analytics for Firebase, an app measurement solution that provides insight on app usage and user engagement on Android and iOS.

firebase_storage

FLUTTER

Flutter plugin for Firebase Cloud Storage, a powerful, simple, and cost-effective object storage service for Android and iOS.

connectivity

FLUTTER

Flutter plugin for discovering the state of the network (WiFi & mobile/cellular) connectivity on Android and iOS.

video_player

FLUTTER

Flutter plugin for displaying inline video with other Flutter widgets on Android and iOS.

device_info

FLUTTER

Flutter plugin providing detailed information about the device (make, model, etc.), and Android or iOS version the app is running on.

Dart Packages

shared_preferences 0.4.3

Published Oct 1, 2018 *FLUTTER*

README.md CHANGELOG.md Example Installing Versions 100

Popularity: 100

Health: 100

Maintenance: 100

Overall: 100

Learn more about *scoring*.

We analyzed this package on Oct 26, 2018, and provided a score, details, and suggestions below. Analysis was completed with status *completed* using:

- Dart: 2.1.0-dev.7.1.flutter-b99bcfd309
- pana: 0.12.5
- Flutter: 0.10.1

About

Flutter plugin for reading and writing simple key-value pairs. Wraps NSUserDefaults on iOS and SharedPreferences on Android.

Homepage
Repository (GitHub)
Issue Tracker
Report an issue (GitHub)
API Docs

Author

Flutter Team

Uploader

zarah@google.com
goderbauer@google.com
jackson@google.com
mravn@google.com
bkonyi@google.com

What about limitations?

Limitations

- not possible to use native UI components (Google Maps)
- no web equivalent
- no support for 3D
- ARM only (no support for x86 in release)
- min. 4.2MB

Limitations

- ~~not possible to use native UI components (Google Maps)~~
- ~~no web equivalent~~
- no support for 3D
- ARM only (no support for x86 in release)
- min. 4.2MB

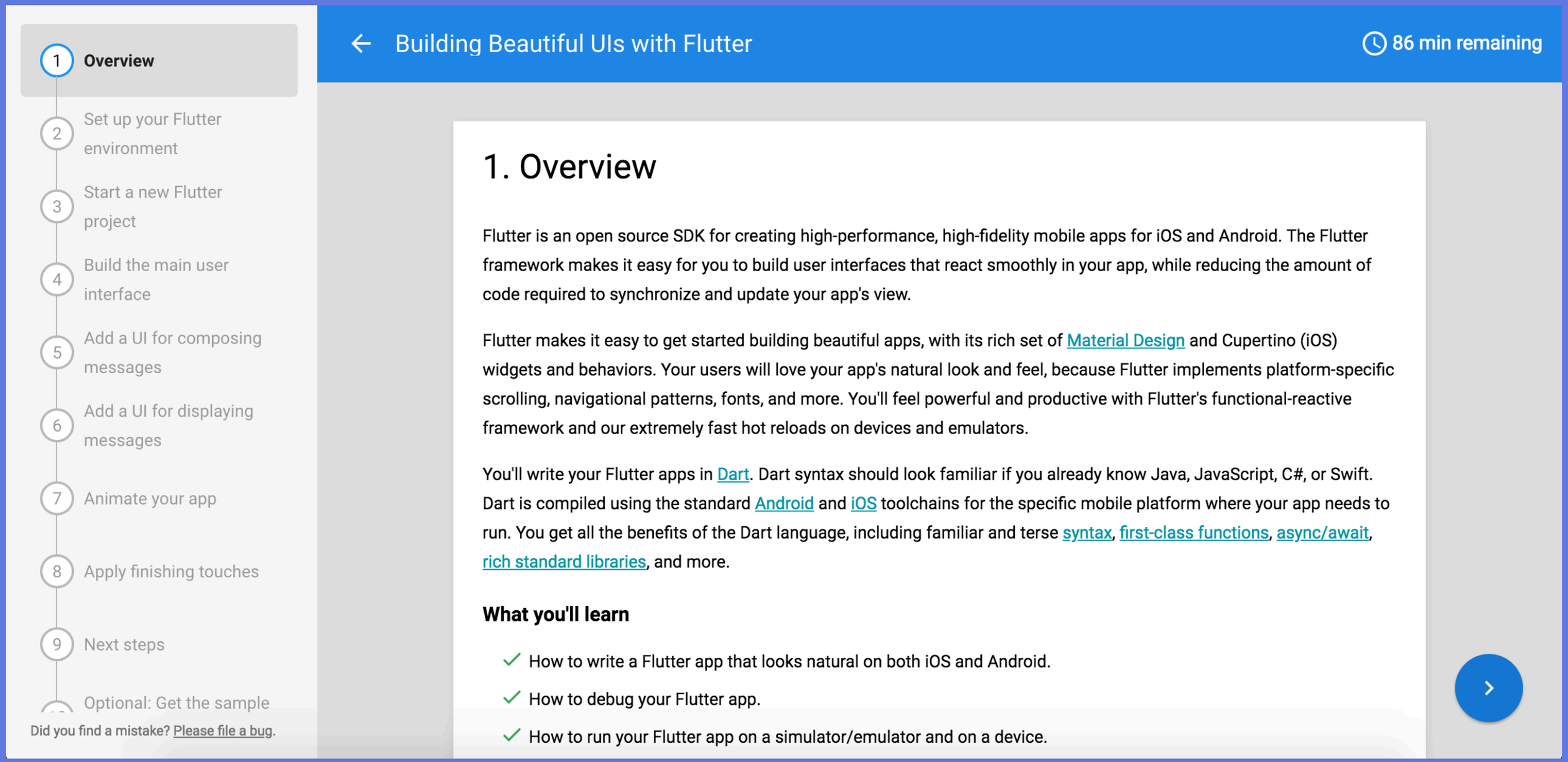
* since Flutter Live, 4th December 2018


ellen

Useful Resources

Flutter UI Codelabs

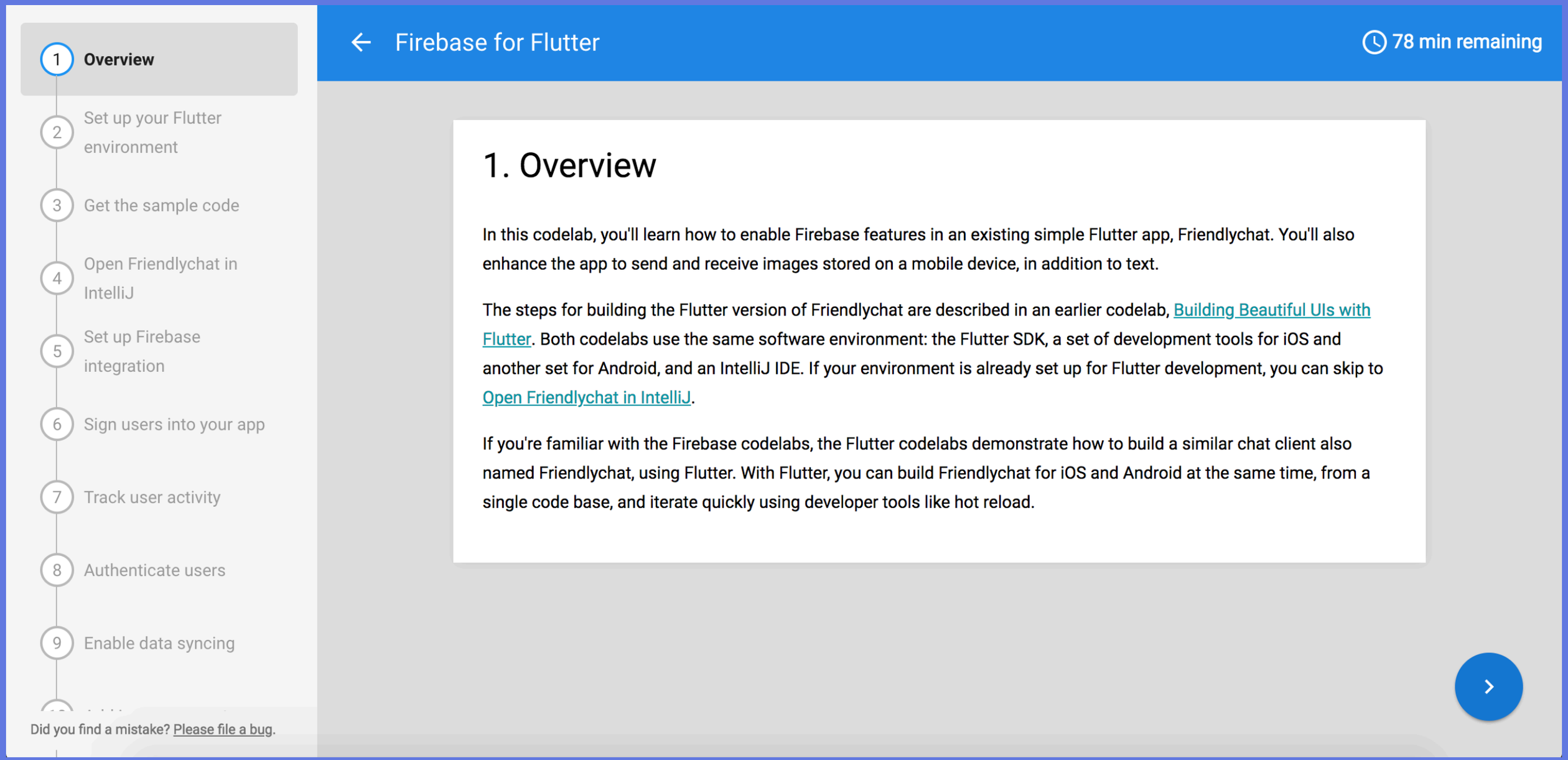
https://codelabs.developers.google.com/codelabs/flutter/



Useful Resources

Flutter Firebase Codelabs

https://codelabs.developers.google.com/codelabs/flutter-firebase



Facts

1. Flutter is cool.
2. Flutter is very fast (development, application).
3. Flutter is super fun.
4. Flutter is easy to start.
5. Flutter is beautiful.

Useful Resources

Do you want to…

See other Flutter applications?
https://itsallwidgets.com

Know more about state management?
https://www.youtube.com/watch?v=zKXz3pUkw9A

Know more about technical details?
https://flutter.io/technical-overview/

Read about 5 reasons why you may love Flutter?
https://hackernoon.com/flutter-5-reasons-why-you-may-love-it-55021fdbf1aa

Thanks!

Questions?