Why companies like Google, Alibaba and UOL choose Flutter

Geison Goes, ThoughtWorks
@geisonfgfg

What is Flutter

- Is a mobile app SDK for building high-performance, high-fidelity, apps for iOS and Android, from a single codebase
- Able to deliver ARM binaries for Android and iOS
- Have a robust and customizable Widget set
- Native Plugins
- Package Manager
- Reactive Framework

Dart the Flutter programming language

Flutter

To develop mobile apps, get the Flutter SDK.

Web

To develop web apps, get the Dart SDK.

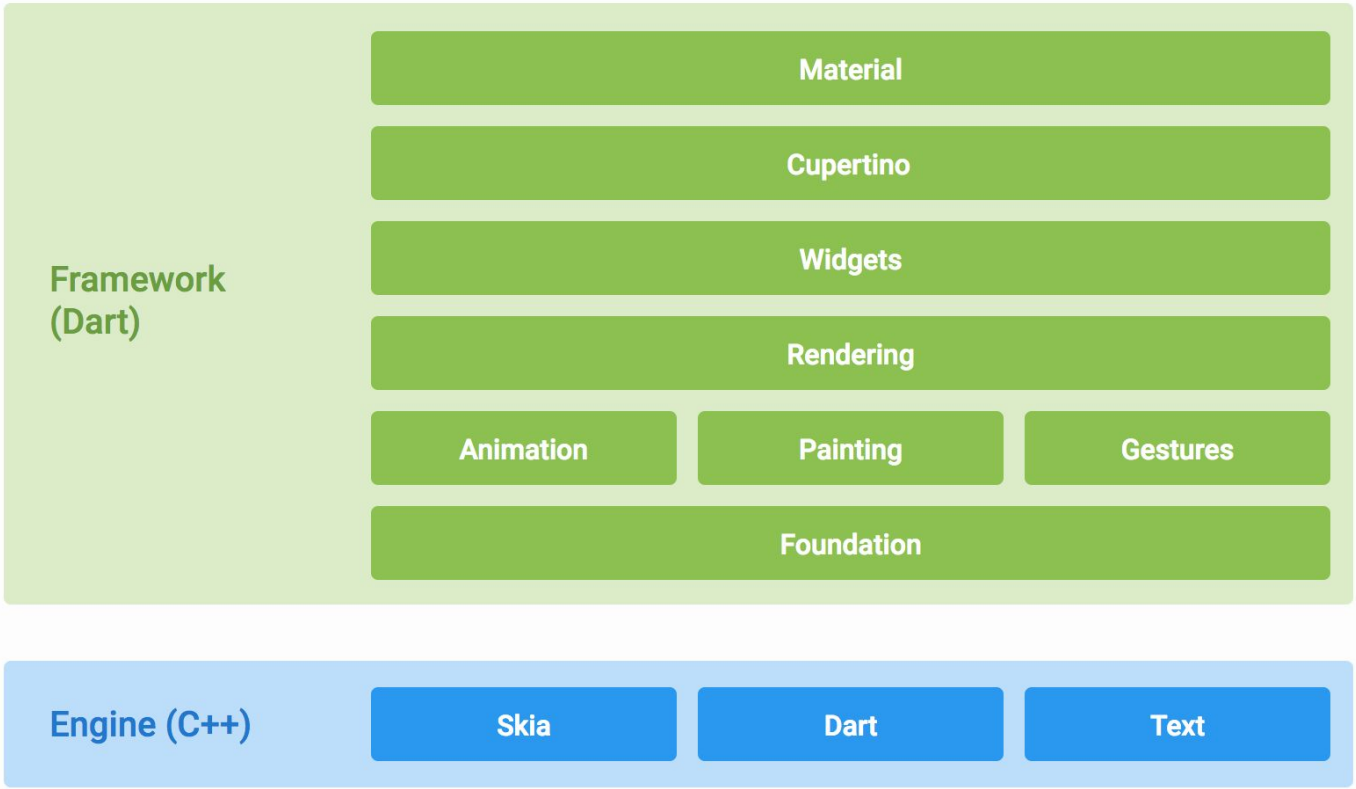
Server

To develop command-line or server-side apps, get the Dart SDK.

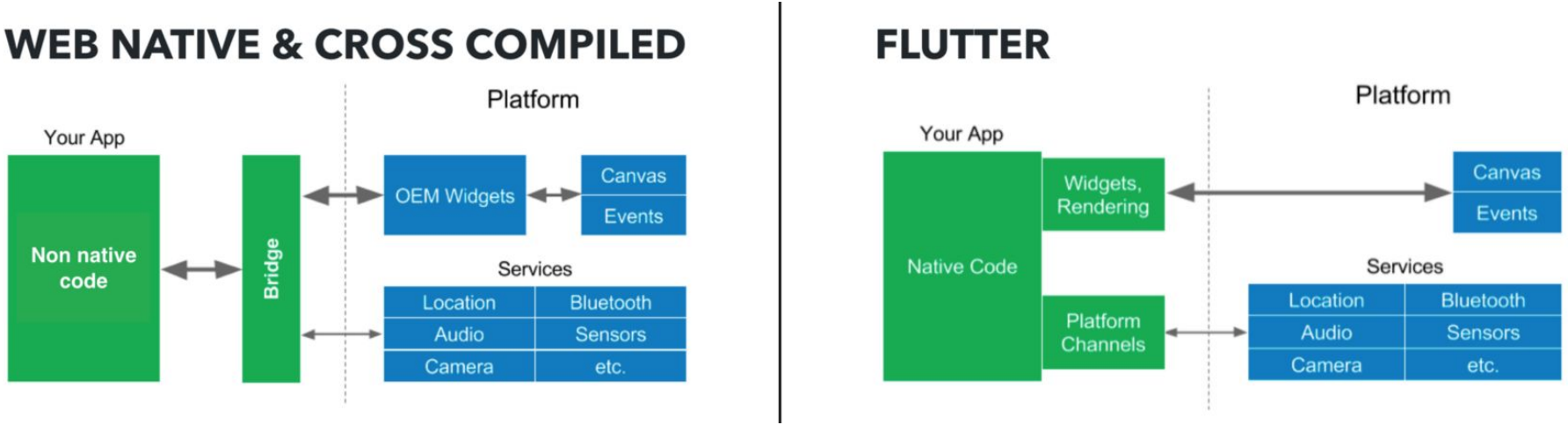
Dart the Flutter programming language

Dart is well-suited to reactive programming, with support for managing short-lived objects—such as UI widgets—through Dart's fast object allocation and generational garbage collector. Dart supports asynchronous programming through language features and APIs that use **Future** and **Stream** objects.

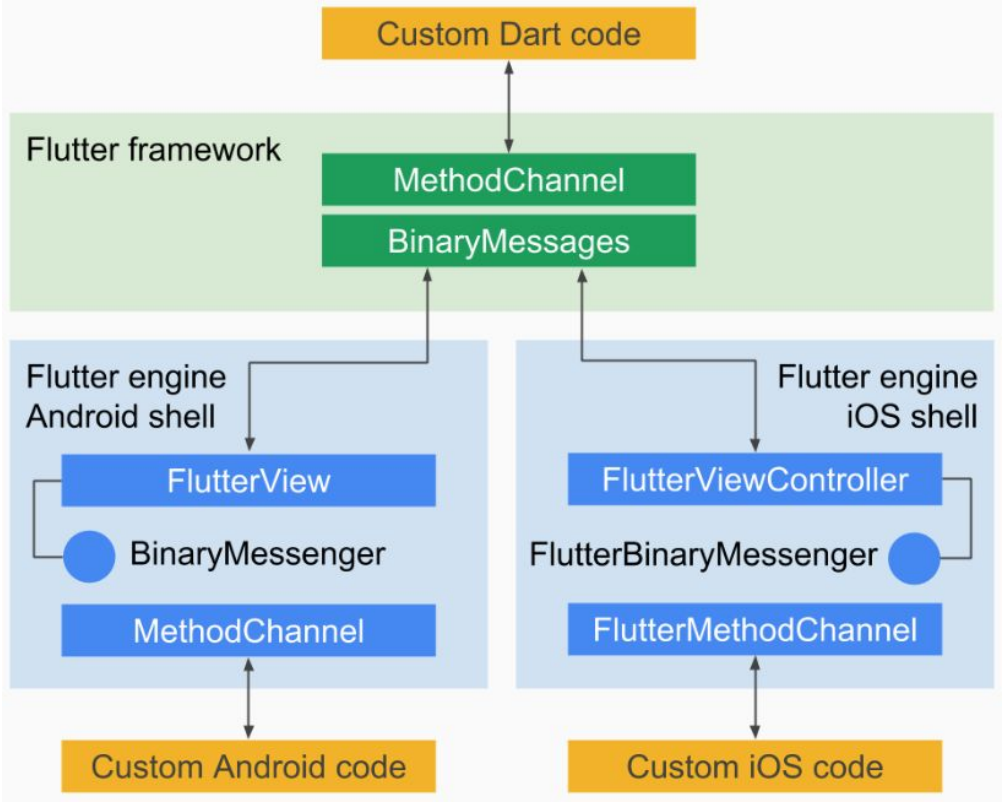
The Flutter architecture



The Flutter architecture



The Flutter Platform Channels



The Flutter Platform Channels

Android Java Channel Side

```java
package com.gawkat.flutterplatforminteractions;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import java.util.Map;
import io.flutter.app.FlutterActivity;
import io.flutter.plugin.common.MethodCall;
import io.flutter.plugin.common.MethodChannel;
import io.flutter.plugins.GeneratedPluginRegistrant;

public class MainActivity extends FlutterActivity {
    private static final String CHANNEL = "demo.gawkat.com/info";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        GeneratedPluginRegistrant.registerWith(this);

        new MethodChannel(getFlutterView(), CHANNEL).setMethodCallHandler(
            new MethodChannel.MethodCallHandler() {
                @Override
                public void onMethodCall(MethodCall methodCall, MethodChannel.Result result) {
                    final Map<String, Object> arguments = methodCall.arguments();

                    if (methodCall.method.equals("getMessage")) {
                        String from = (String) arguments.get("from");
                        String message = "Android say hi " + from;
                        result.success(message);
                    }
                }
            }
        );
    }
}
```

The Flutter Platform Channels

iOS Objective-C Channel Side

```objc
#include "AppDelegate.h"
#include "GeneratedPluginRegistrant.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [GeneratedPluginRegistrant registerWithRegistry:self];
    // Override point for customization after application launch.

    FlutterViewController* controller = (FlutterViewController*)self.window.rootViewController;
    FlutterMethodChannel* channel = [FlutterMethodChannel
                                     methodChannelWithName:@"demo.gawkat.com/info"
                                     binaryMessenger:controller];

    [channel setMethodCallHandler:^(FlutterMethodCall* call, FlutterResult result) {
        NSSring *from = call.arguments[@"from"];

        if ([@"getMessage" isEqualToString:call.method]) {
            NSString *message = @"iOS says greetings";
            NSString *returnMessage = [message stringByAppendingString:from];
            result(returnMessage);
        }
    }];

    return [super application:application didFinishLaunchingWithOptions:launchOptions];
}
@end
```

The Flutter Platform Channels

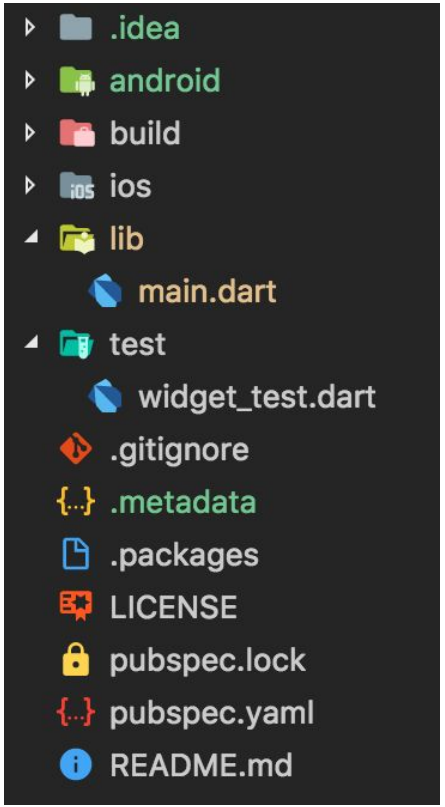
Flutter Channel Usage

```dart
import 'dart:async';

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: 'Flutter Demo',
      theme: new ThemeData(
        primarySwatch: Colors.lightGreen,
      ),
      home: new MyHomePage(),
    );
  }
}

class MyHomePage extends StatefulWidget {
  @override
  _MyHomePageState createState() => new _MyHomePageState();
}
```

```dart
class _MyHomePageState extends State<MyHomePage> {
  static const platform = const MethodChannel('demo.gawkat.com/info');
  String _message = "No messages yet...";

  @override
  void initState() {

    _getMessage().then((String message) {
      setState(() {
        _message = message;
      });
    });

    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text("Home"),
      ),
      body: new ListView(
        children: <Widget>[
          new ListTile(
            title: new Text(_message),
          )
        ],
      ),
    );
  }

```

```dart
  Future<String> _getMessage() async {
    var sendMap = <String, dynamic> { 'from' : 'Geison', };
    String value;

    try {
      value = await platform.invokeMethod('getMessage', sendMap);
    } catch (e) {
      print(e);
    }

    return value;
  }
}
```

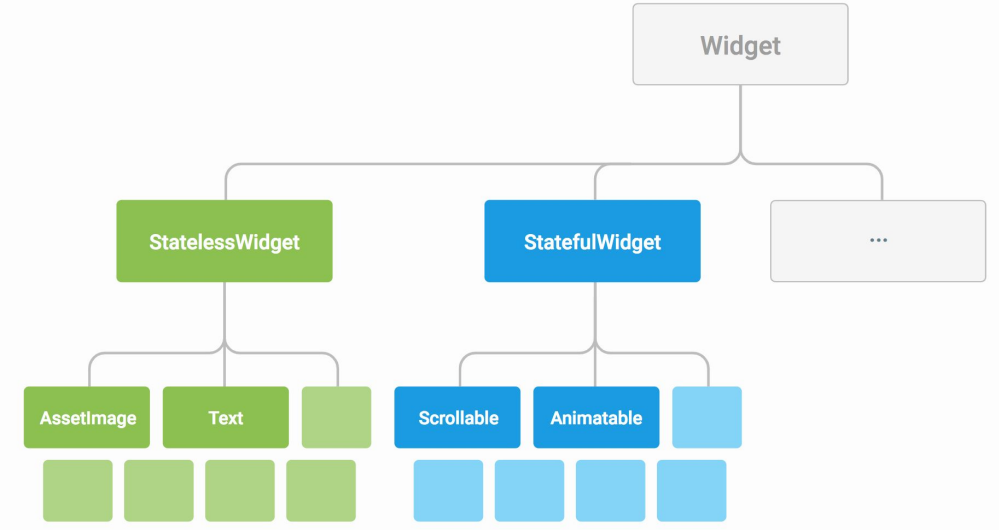
The Flutter project structure



The Flutter Widget System

Everything in Flutter is a Widget

Widgets are divided in two categories
StatefulWidgets and **StatelessWidgets**



A Stateless Widget

A widget that does not require mutable state

Widgets that don't need to manage any form of internal state, used at times where only the data that is initially passed into the object is necessary

- Text
- Button

```
class BigText extends StatelessWidget {
  // a property on this class
  final String text;

  // a constructor for this class
  BigText(this.text);

  Widget build(context) {
    // Pass the text down to another widget
    return new Text(
      text,
      // Even changing font-style is done through a Dart class.
      textStyle: new TextStyle(fontSize: 20.0),
    );
  }
}
```

A Stateful Widget

A widget that has mutable state

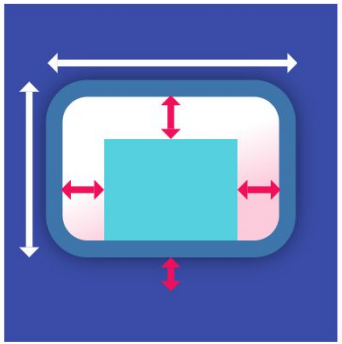
They allow us to create widgets which can dynamically change their content over time and don't rely on static states which are passed in during their instantiation

- Slider
- Checkbox

```
class Counter extends StatefulWidget {
  Counter({Key key, this.title}) : super(key: key);

   // Stateful Widgets don't have build methods.
   // They have createState() methods.
   // Create State returns a class that extends Flutters State class.
  @override
  _MyHomePageState createState() => new _MyHomePageState();

  // Stateful Widgets are rarely more complicated than this.
}

class _MyHomePageState extends State<MyHomePage> {
  int counter = 0;

  void increaseCount() {
    // setState is a special method that tells Flutter to repaint
    // the view because state has been updated!
    setState(() {
      this.counter++;
    }
  }

  // gotta have that build method!
  Widget build(context) {
    return new RaisedButton(
      onPressed: increaseCount,
      child: new Text('Tap to Increase'),
    );
  }
}
```

A Layout Widget

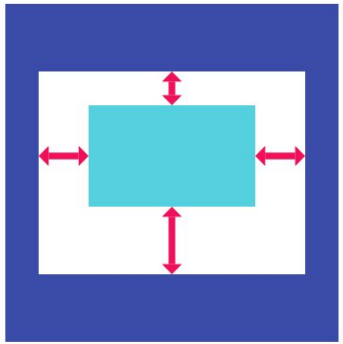
Single-child layout widgets



Container

A convenience widget that combines common painting, positioning, and sizing widgets.

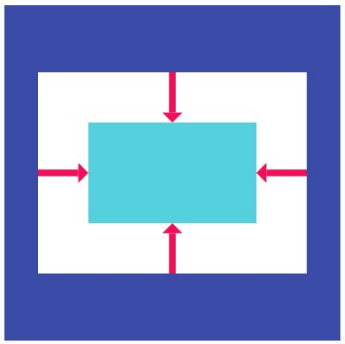
Documentation



Padding

A widget that insets its child by the given padding.

Documentation



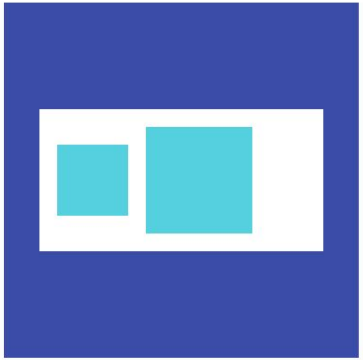
Center

A widget that centers its child within itself.

Documentation

A Layout Widget

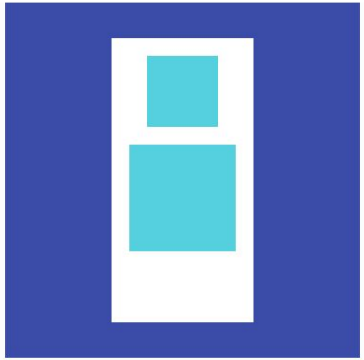
Multi-child layout widgets



Row

Layout a list of child widgets in the horizontal direction.

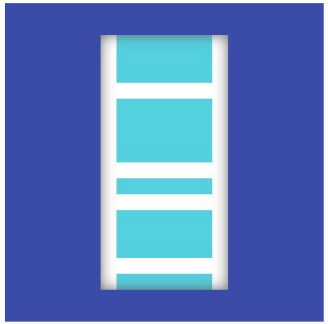
Documentation



Column

Layout a list of child widgets in the vertical direction.

Documentation

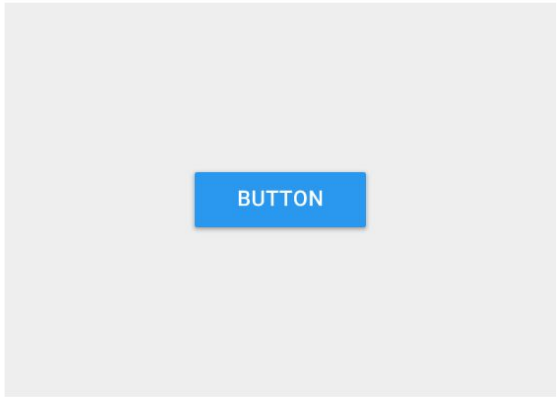


ListView

A scrollable, linear list of widgets. ListView is the most commonly used scrolling widget. It displays its children one after another in the scroll direction. In the cros...

Documentation , Samples

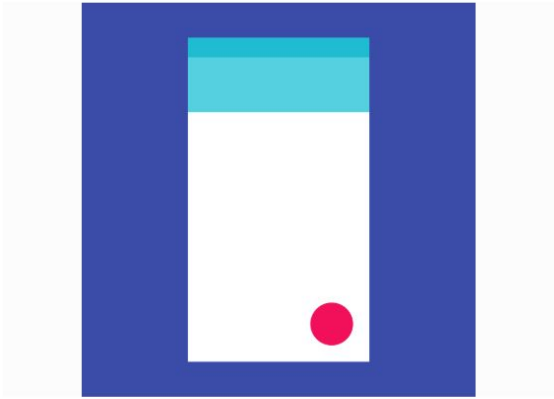
Android Material Widgets



RaisedButton

A Material Design raised button. A raised button consists of a rectangular piece of material that hovers over the interface.

Documentation

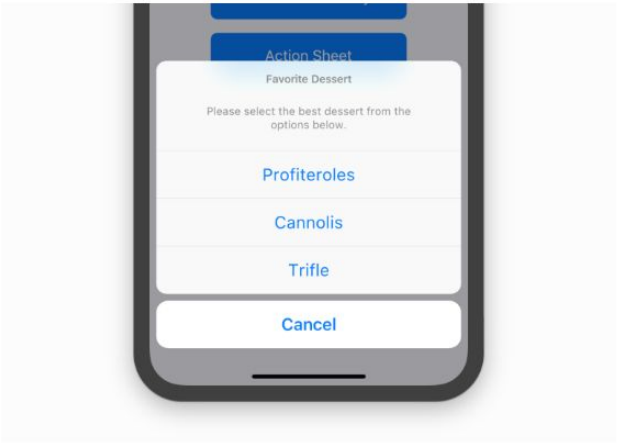


FloatingActionButton

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

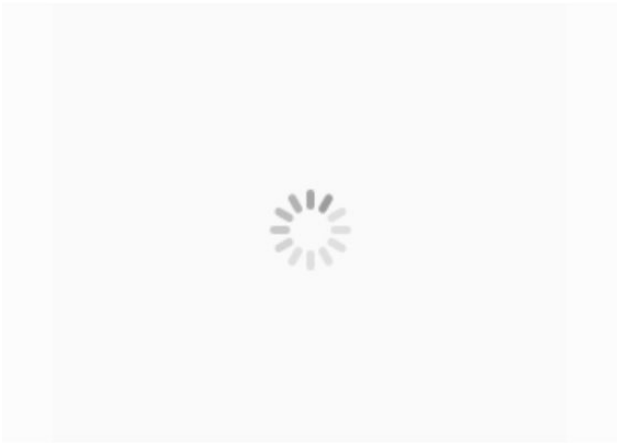
iOS Copertino Widgets



CupertinoActionSheet

An iOS-style bottom action sheet.

Documentation



CupertinoActivityIndicator

An iOS-style activity indicator. Displays a circular 'spinner'.

Documentation

Flutter main advantages

It is possible develop without IDEs just commandline and a text editor is enough

+

Share Code/Libs to both platforms and maybe web and apis

+

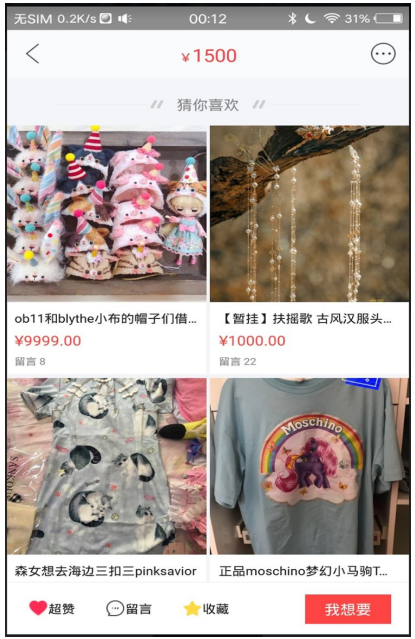
Hot Reload

+

Developer Happiness

Flutter Showcase

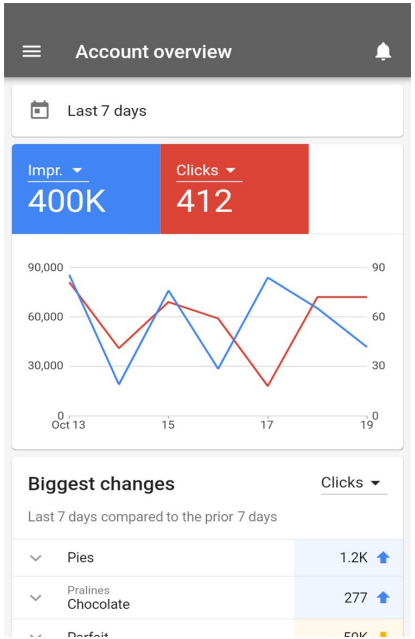
Alibaba



[Video Case]

(Chinese audio only, turn on subtitle)

Google Ads



UOL

References

- https://flutter.io
- https://flutter.io/showcase/
- https://github.com/iampawan/FlutterExampleApps
- https://www.youtube.com/watch?v=qWL1lGchpRA&list=PLR2qQy0Zxs_UdqAcaipPR3CG1Ly57UlhV
- https://flutterbyexample.com/flutter-widgets/#stateless-and-statefulwidgets

Thank You

Geison Goes

twitter.com/@geisonfgfg

- Senior Consultant na Thoughtworks
- Software Engineer for 12 years
- Husband
- Father
- Dog trainer
- Read more at about.me/geisonfgf