العطاء الرقمي
Attaa Digital

تطوير التطبيقات باستخدام Flutter

Eng. Mouaz M. Al-Shahmeh
Google Dev Library Contributor

Google Developers

Flutter

العطاء الرقمي
Attaa Digital

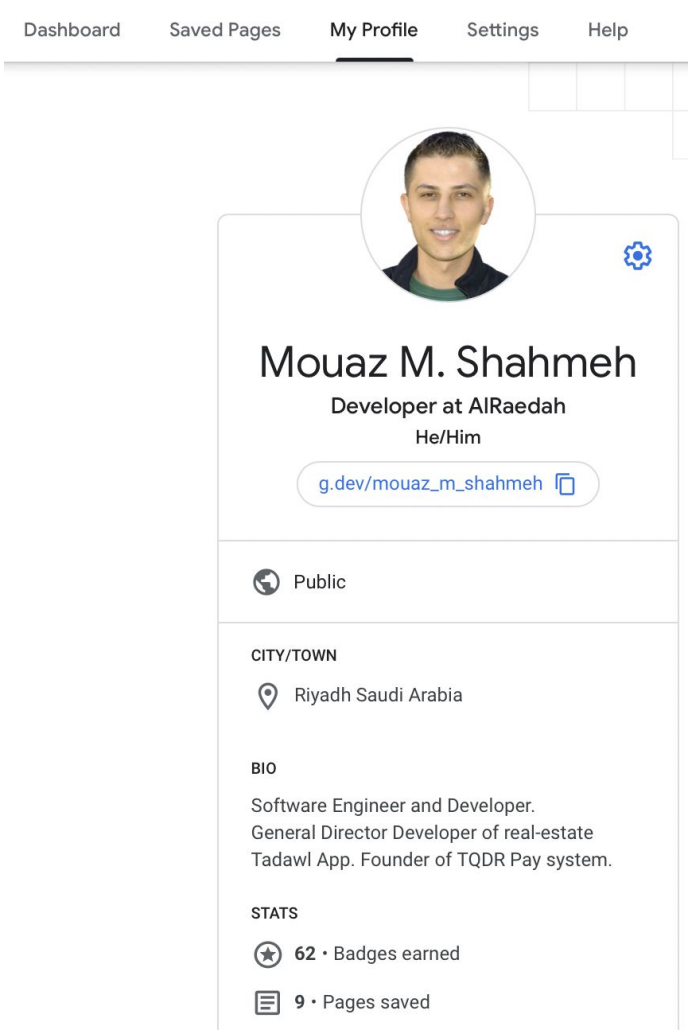
Ask Me Anything

m.m.shahmeh@gmail.com

g.dev/mouaz_m_shahmeh

LINKS

- github.com/Eng-Mouaz-M-AlShahmeh
- linkedin.com/in/mouaz-shahmeh
- stackoverflow.com/users/18449528
- twitter.com/mouaz_m_shahmeh



Flutter

العطاء الرقمي
Attaa Digital

أهداف اللقاء

Flutter

العطاء الرقمي
Attaa Digital

أهداف اللقاء

1. بناء تطبيقات بشفرة كود واحدة بلغة البرمجة DART ولمنصات عديدة
2. بناء تطبيقات تفاعلية ومتجاوبة لمقاسات الشاشات
3. التعرف على البرمجة الآمنة للقيم الفارغة
4. بناء تطبيقات جميلة وسريعة وبأداء أصلي بتقنية Flutter

Flutter

العطاء الرقمي
Attaa Digital

أهداف اللقاء

5. استخدام تنقل فعال بين شاشات التطبيق

6. اختبار التطبيقات بشكل حقيقي وتجهيزها للنشر في متاجر التطبيقات

7. إدارة البيانات في التطبيق بفاعلية والتعامل مع قواعد بيانات علائقية وغير علائقية

8. برمجة التطبيقات بأكواد نظيفة عن طريق إدارة الحالة

Flutter

العطاء الرقمي
Attaa Digital

محاور اللقاء

Flutter

العطاء الرقمي
Attaa Digital

محاور اللقاء

1. مقدمة في Flutter
2. تهيئة بيئة التطوير
3. تشغيل أول تطبيق Flutter
4. Declarative vs Imperative

Flutter

العطاء الرقمي
Attaa Digital

محاور اللقاء

5 التصميم المتجاوب

6 Null Safety

7 State Management

8 المفاتيح - التنقل

Flutter

العطاء الرقمي
Attaa Digital

محاور اللقاء

9 خيارات قواعد البيانات

10 اختبارات الأداء

11 أنماط بناء التطبيقات

12 المشروع - ملخص ومراجع

Flutter

العطاء الرقمي
Attaa Digital

1

مقدمة في Flutter

Flutter العطاء الرقمي Attaa Digital

مقدمة في Flutter

What is Flutter?

Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source.

Flutter

العطاء الرقمي
Attaa Digital

مقدمة في Flutter

Hot Reload - Hot Restart

One Code Base (Dart)

بيئات تطوير منافسة cross-platform
(KMM - Xamarin - React Native)

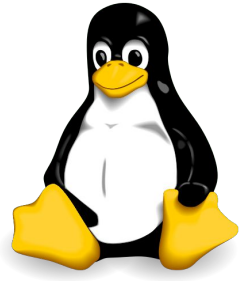
تطوير التطبيقات الأصلي:
اندرويد (Kotlin - Java)
ابل (Objective C - Swift)

تطبيقات ثلاثية الأبعاد (Unity)

Flutter

العطاء الرقمي
Attaa Digital

مقدمة في Flutter



تطبيقات الجوال (اندرويد - ابل)

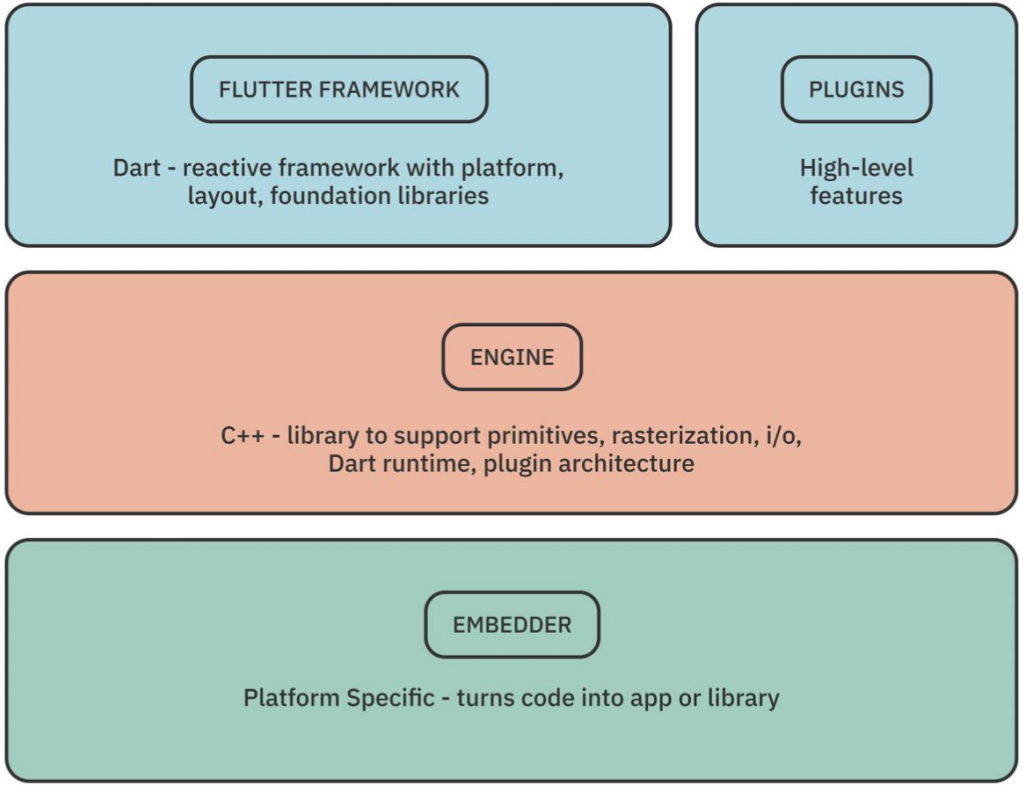
تطبيقات سطح المكتب (ويندوز - ماك - لينكس)

تطبيقات الويب

Flutter

العطاء الرقمي
Attaa Digital

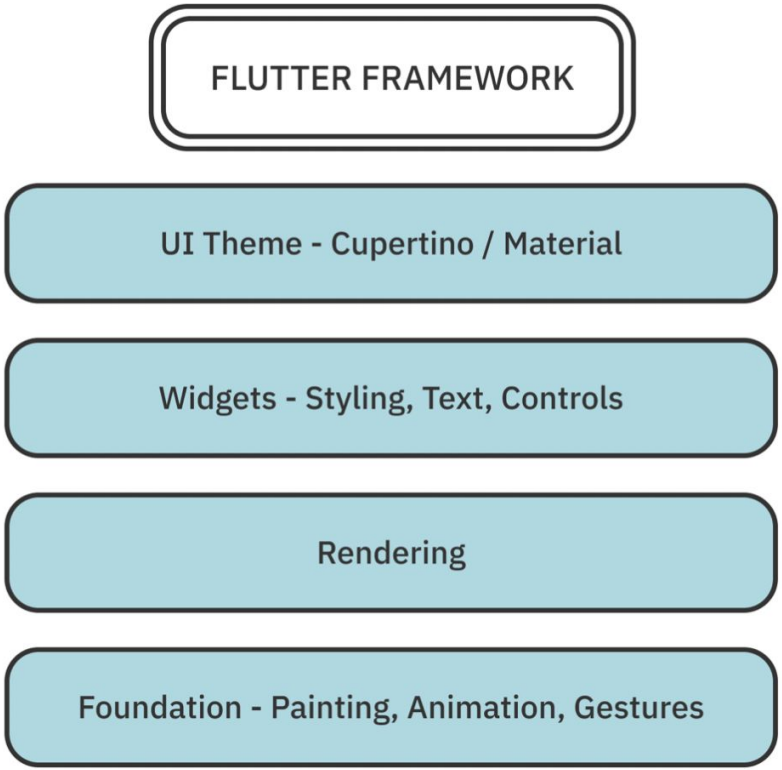
مقدمة في Flutter



Flutter

العطاء الرقمي
Attaa Digital

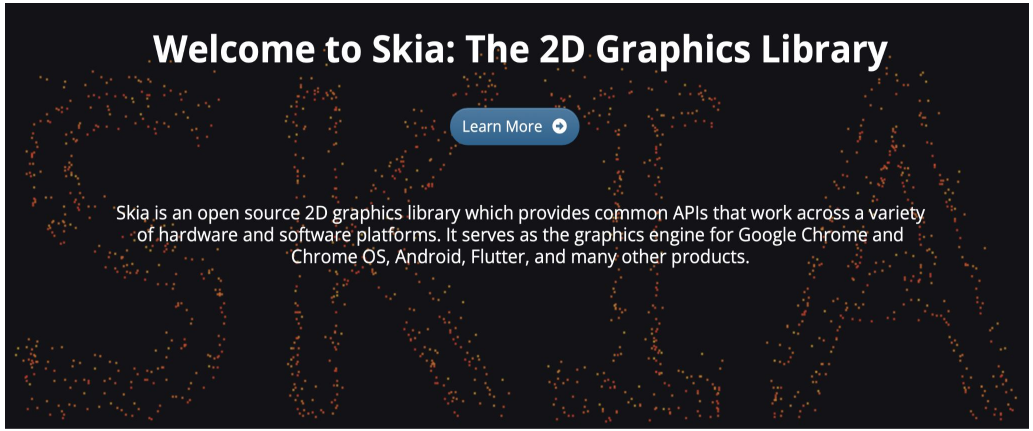
مقدمة في Flutter



Flutter

العطاء الرقمي
Attaa Digital

مقدمة في Flutter



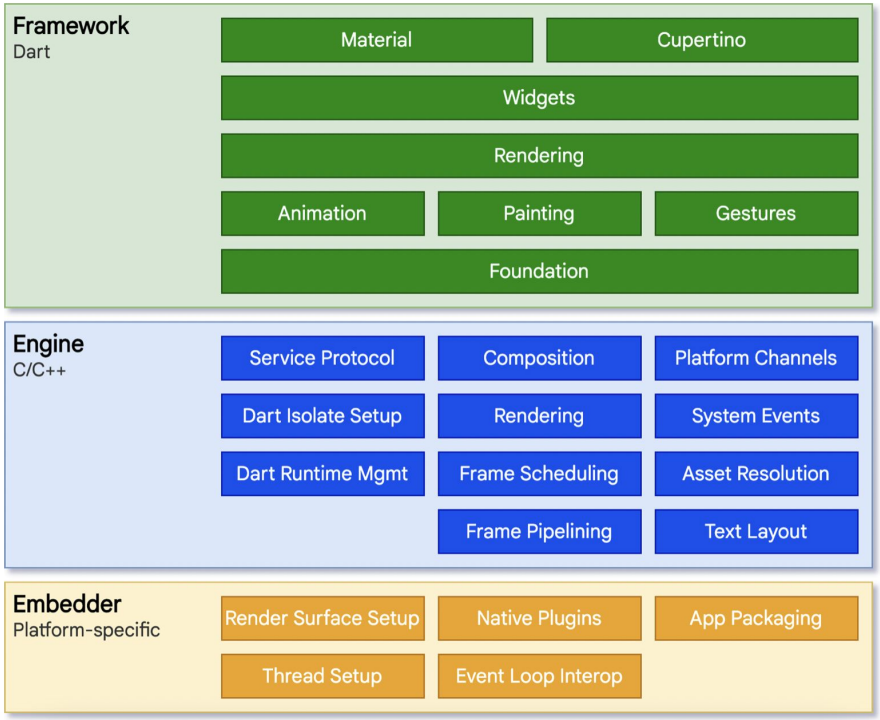
Supported Platforms

Windows 7, 8, 8.1, 10
macOS 10.10.5 or later
iOS 11 or later
Android 4.1 (JellyBean) or later
Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, or Fedora Linux 32+

Flutter

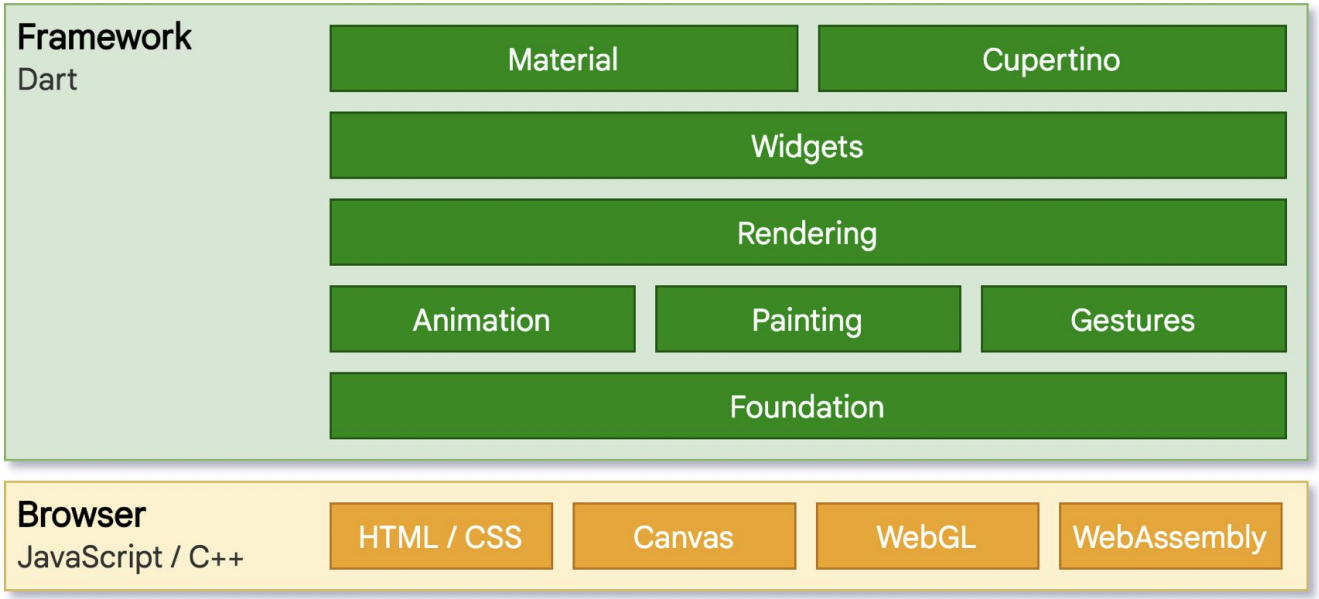
العطاء الرقمي
Attaa Digital

مقدمة في Flutter



مقدمة في Flutter

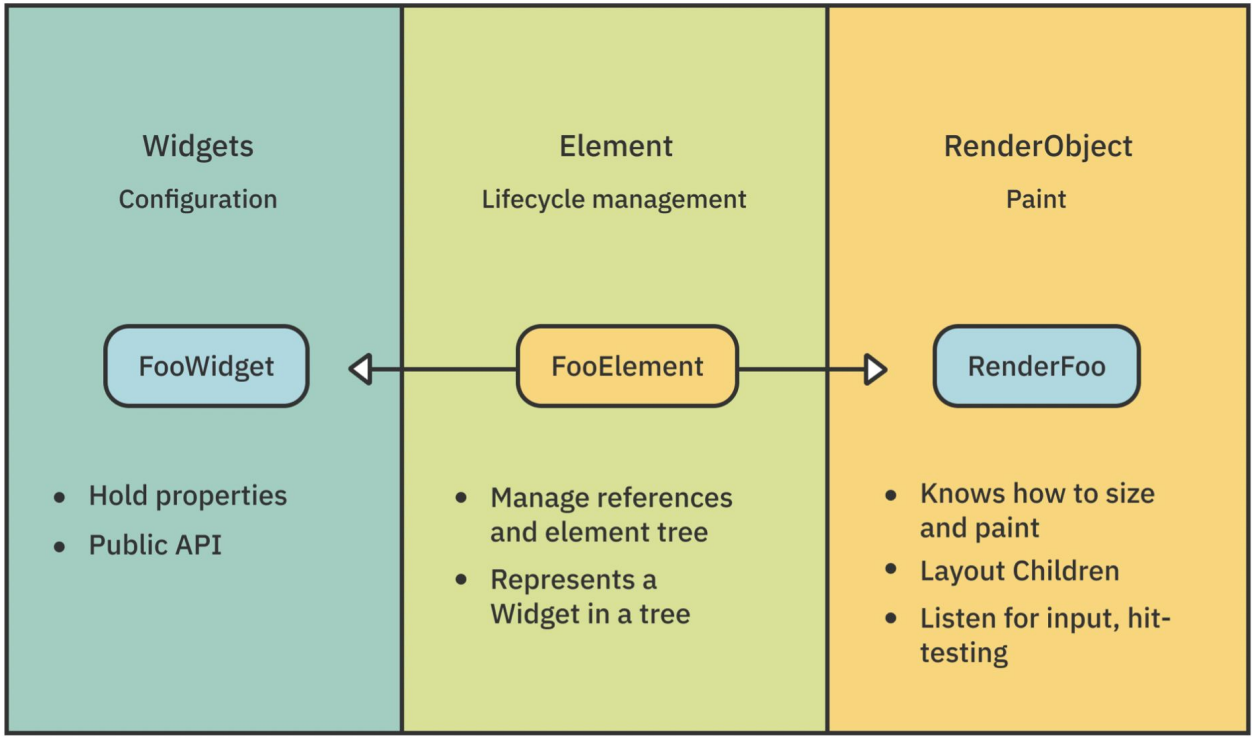
The web version of the architectural layer diagram is as follows:



Flutter

العطاء الرقمي
Attaa Digital

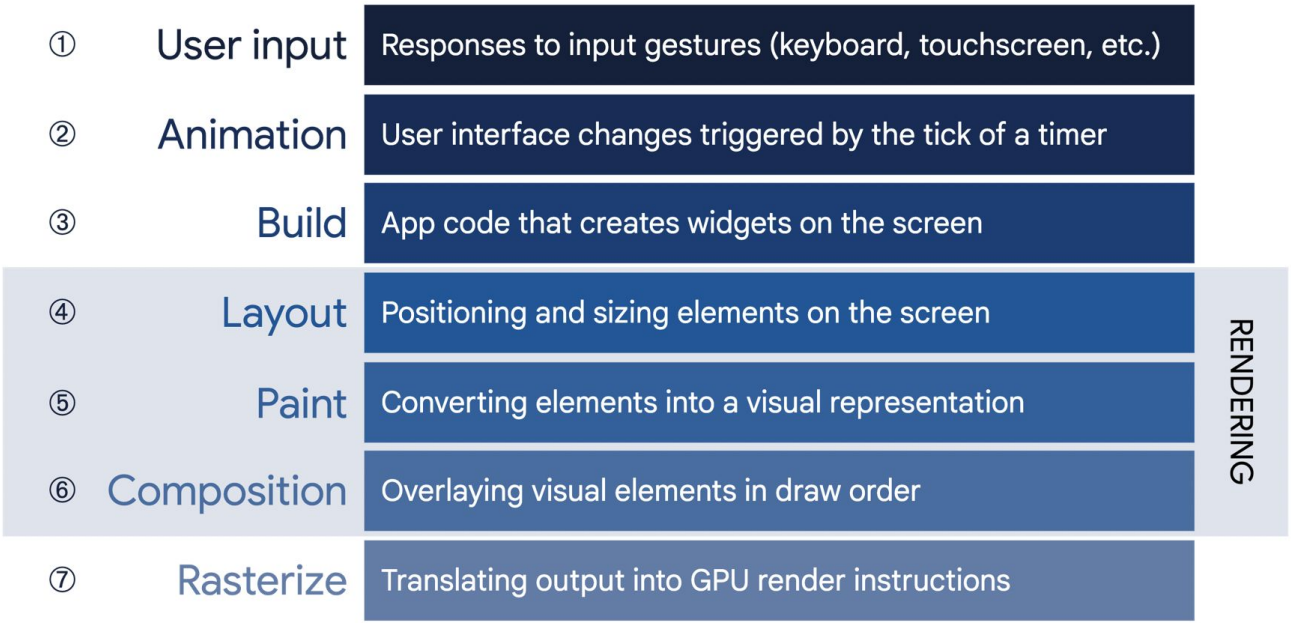
مقدمة في Flutter



Flutter

العطاء الرقمي
Attaa Digital

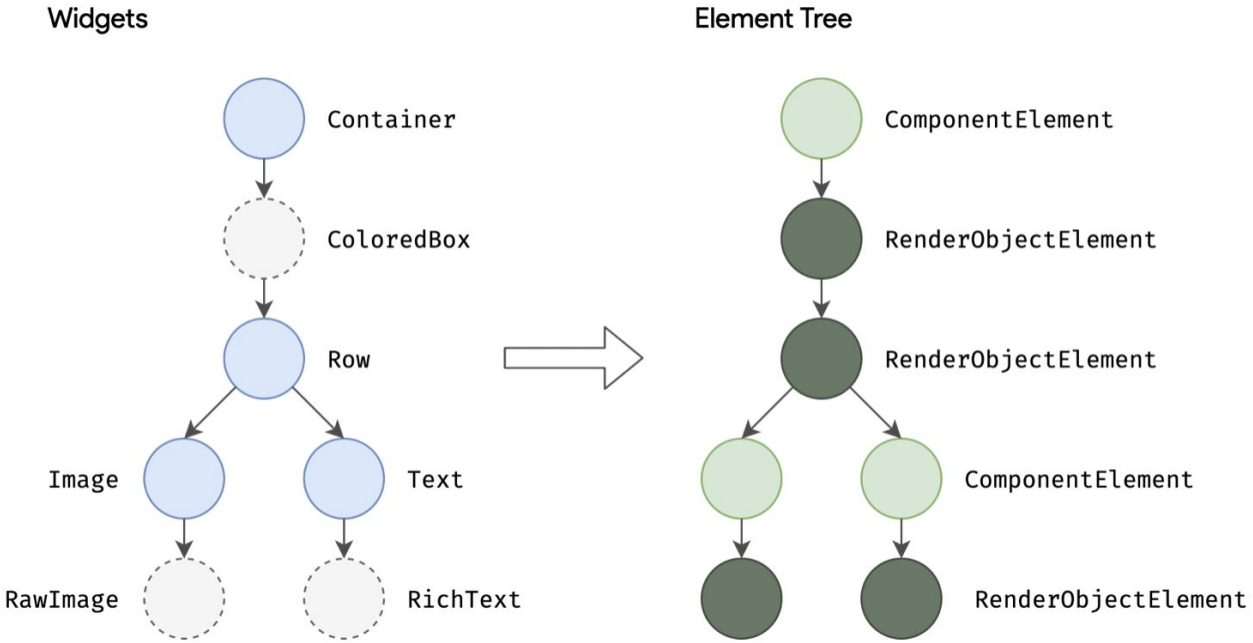
مقدمة في Flutter



Flutter

العطاء الرقمي
Attaa Digital

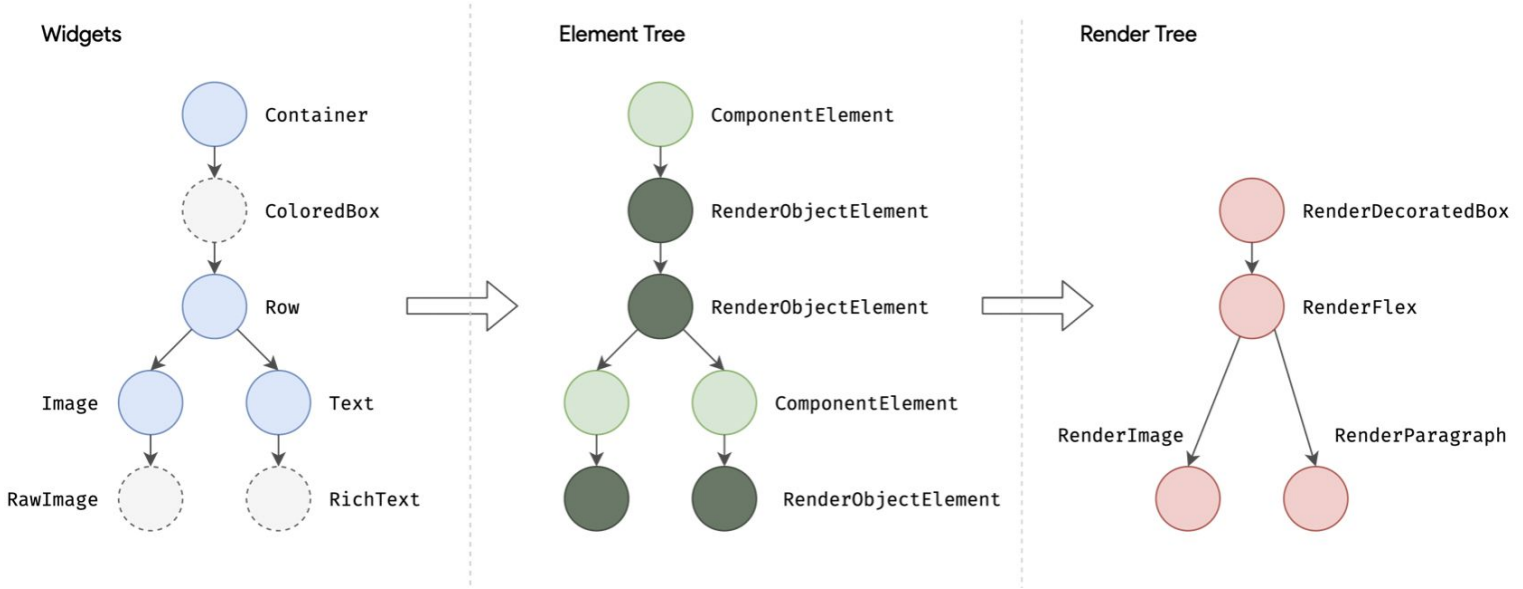
مقدمة في Flutter



Flutter

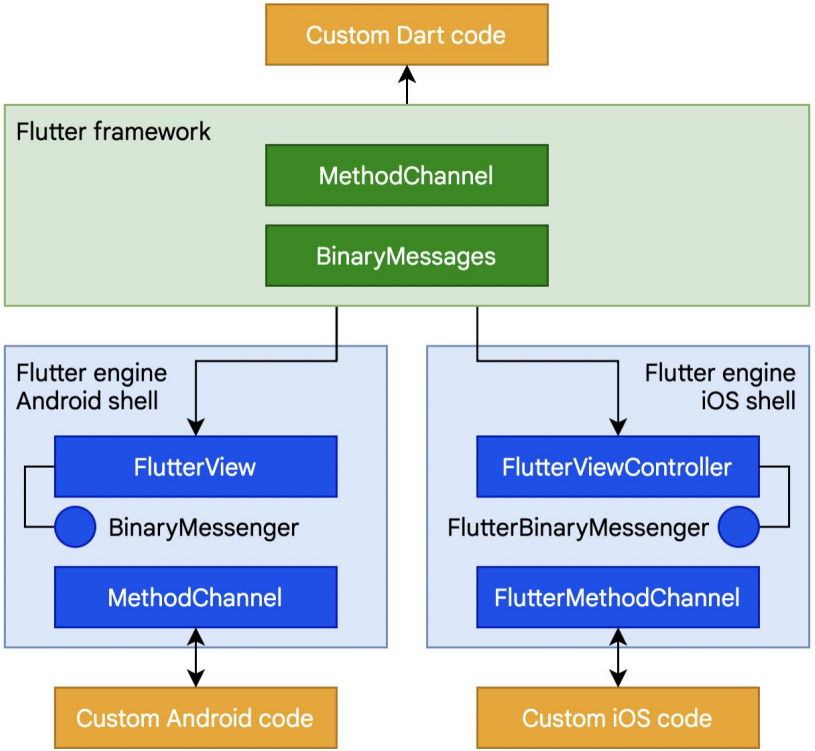
العطاء الرقمي
Attaa Digital

مقدمة في Flutter



Flutter العطاء الرقمي Attaa Digital

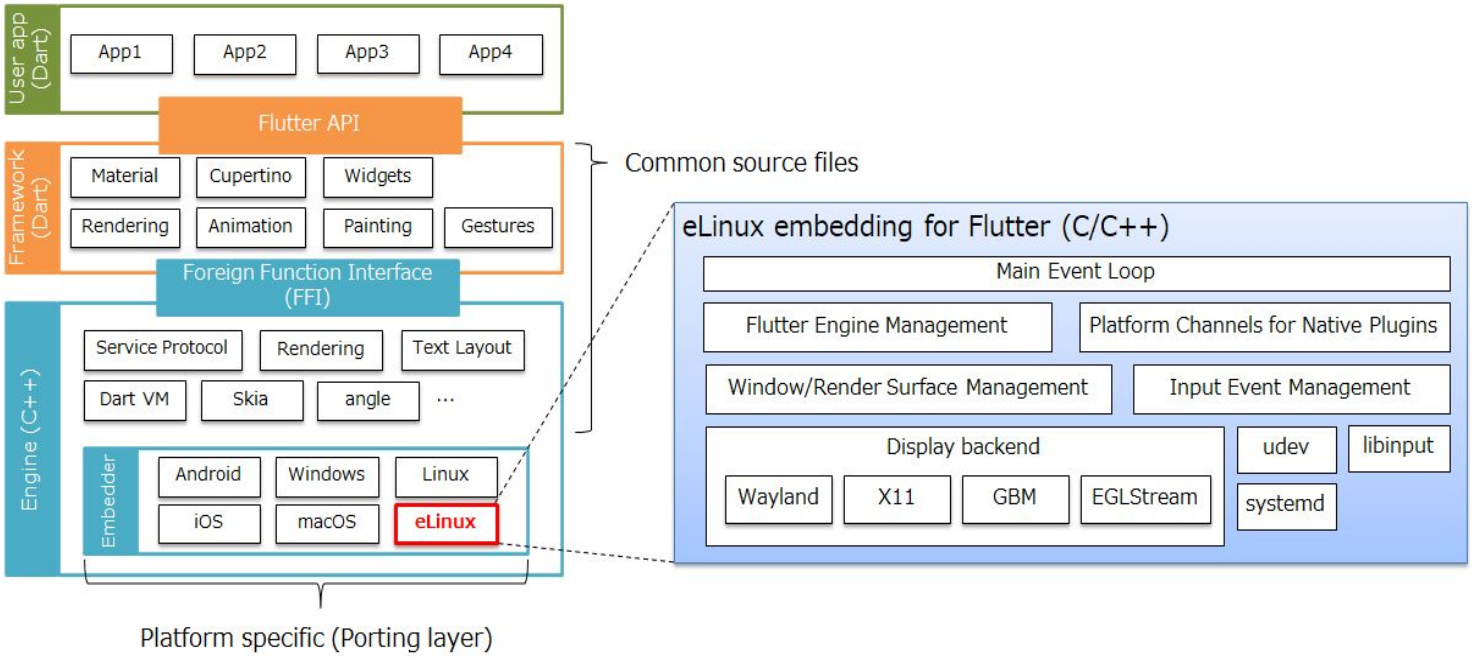
مقدمة في Flutter



Flutter العطاء الرقمي Attaa Digital

مقدمة في Flutter

جاري العمل في Flutter الآن في
أنظمة Embedder



Flutter

العطاء الرقمي
Attaa Digital

مقدمة في Flutter

جاري العمل في Flutter الآن في أنظمة Embedder

Tested devices

Board / SoC	Vendor	OS / BSP	Backend	Status
Jetson Nano	NVIDIA	JetPack 4.3	Wayland	✓
Jetson Nano	NVIDIA	JetPack 4.3	DRM	✓ (#1)
Raspberry Pi 4 Model B	Raspberry Pi Foundation	Ubuntu 20.10	Wayland	✓
Raspberry Pi 4 Model B	Raspberry Pi Foundation	Ubuntu 20.10	DRM	✓ (#9)
i.MX 8MQuad EVK	NXP	Sumo (kernel 4.14.98)	Wayland	✓
i.MX 8M Mini EVKB	NXP	Zeus (kernel 5.4.70)	Wayland	✓
RB5 Development Kit	Qualcomm	Ubuntu 18.04.05	DRM	✓
Zynq	Xilinx	-	-	Not tested
Desktop (x86_64)	Intel	Ubuntu 20.04	Wayland	✓
Desktop (x86_64)	Intel	Ubuntu 20.04	DRM	✓
Desktop (x86_64)	Intel	Ubuntu 20.04	X11	✓
QEMU (x86_64)	QEMU	AGL (Automotive Grade Linux) koi	Wayland	✓
QEMU (x86_64)	QEMU	AGL (Automotive Grade Linux) koi	DRM	✓

Flutter

العطاء الرقمي
Attaa Digital

مقدمة في Flutter

جاري العمل في Flutter الآن في أنظمة Embedder

TIZEN DEVICES مثال TOYOTA AGL

TIZEN™

Flutter

العطاء الرقمي
Attaa Digital

2

تهيئة بيئة التطوير

Flutter العطاء الرقمي Attaa Digital

تهيئة بيئة التطوير

مواصفات جهاز جيدة

تحميل Flutter SDK

تحميل برنامج أكواد

ابل (cocoapods) - اندرويد (JDK Java)

DartPad

GNU Emacs

Vim

Eclipse

android studio

Visual Studio Code

IntelliJ IDEA

Xcode

Flutter

العطاء الرقمي
Attaa Digital

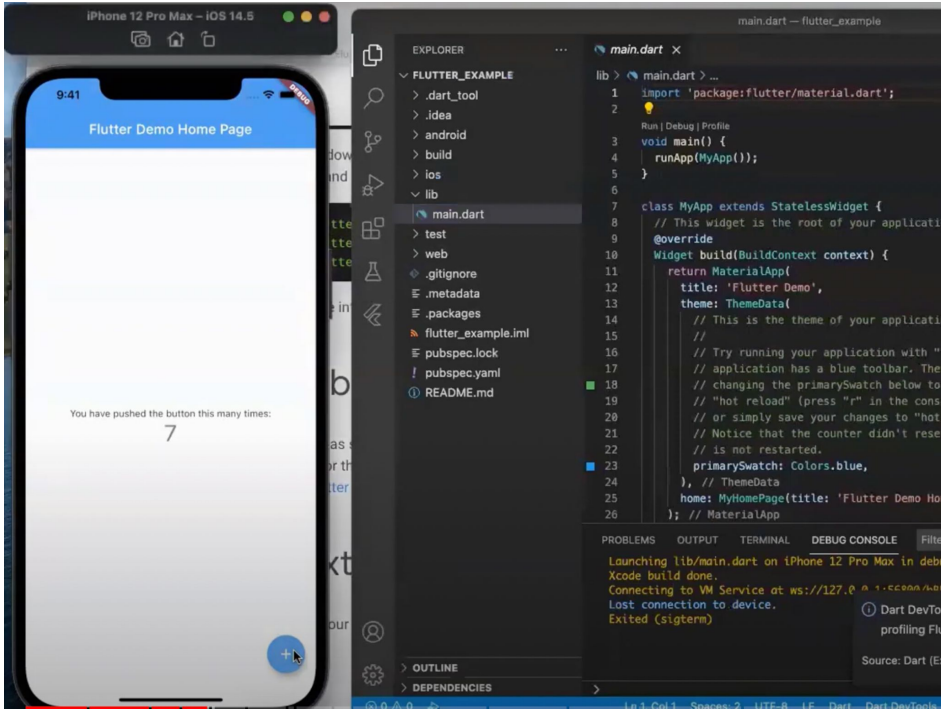
3

تشغيل أول تطبيق

Flutter

العطاء الرقمي
Attaa Digital

تشغيل أول تطبيق



Flutter

العطاء الرقمي
Attaa Digital

4

Declarative VS Imperative

Flutter

العطاء الرقمي
Attaa Digital

Declarative VS Imperative

Enum enhancement

Before: have to use extensions

```
enum Water {
  frozen,
...
  boiling;
}

extension Members on Water {
  int waterToTemp(Water water) {
    switch (water) {
      case Water.frozen:
        return 32;
...
    }
  }

  String convertToString() =>
    "The $name water is ${waterToTemp(this)} F.";
}

void main() { print(Water.frozen.convertToString()); }
```

Now: directly on enum. Can override.

```
enum Water {
  frozen(32),
...
  boiling(212);

  final int tempInFahrenheit;
  const Water(this.tempInFahrenheit);

  @override
  String toString() =>
    "The $name water is $tempInFahrenheit F.";
}

void main() { print(Water.frozen); }
```

Flutter

العطاء الرقمي
Attaa Digital

5

التصميم المتجاوب

Flutter

العطاء الرقمي
Attaa Digital

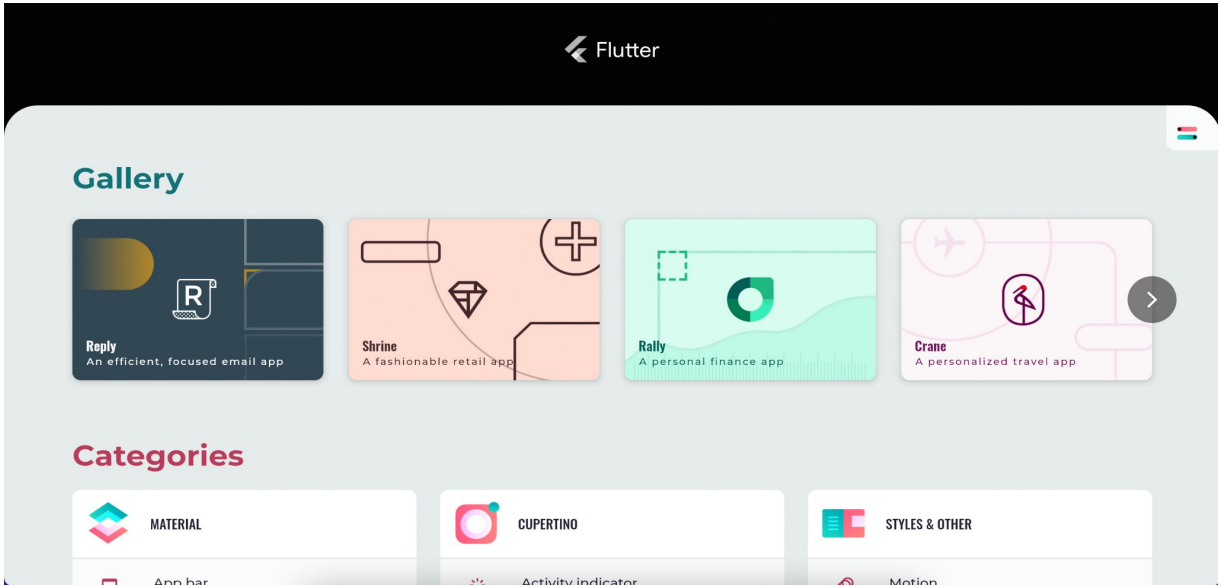
التصميم المتجاوب

MediaQuery.of(context)

Cupertino (apple)

Material (android) 3

gallery.flutter.dev



Flutter

العطاء الرقمي
Attaa Digital

6

Null Safety

Flutter

العطاء الرقمي
Attaa Digital

Null Safety

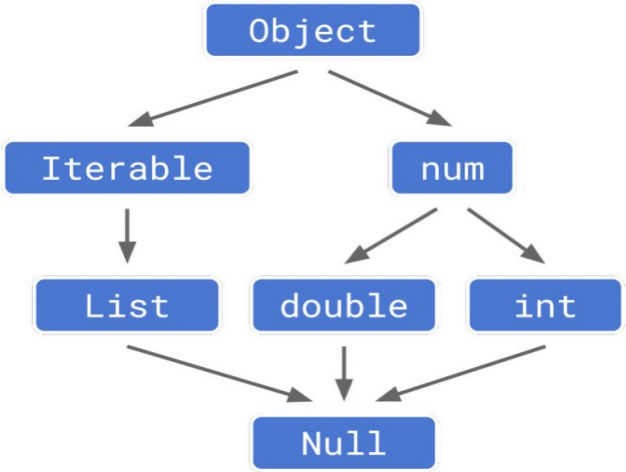
Sound null safety is available in Dart 2.12 and Flutter 2.

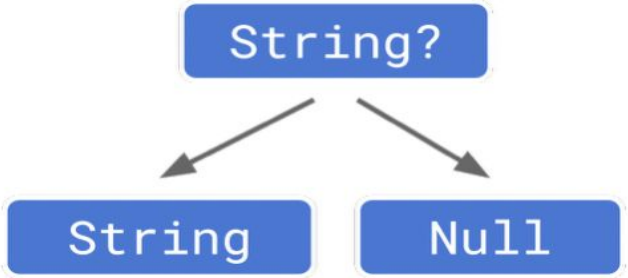
Flutter

العطاء الرقمي
Attaa Digital

Null Safety

In type theory lingo, the `Null` type was treated as a subtype of all types:

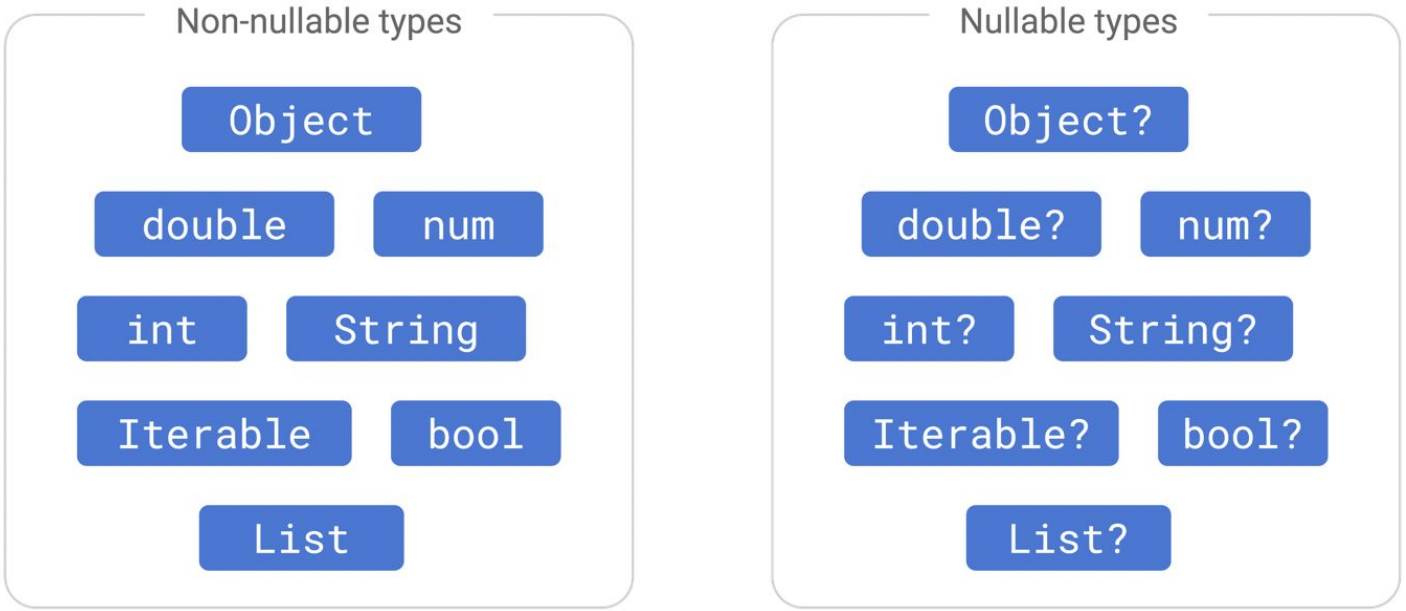




Flutter

العطاء الرقمي
Attaa Digital

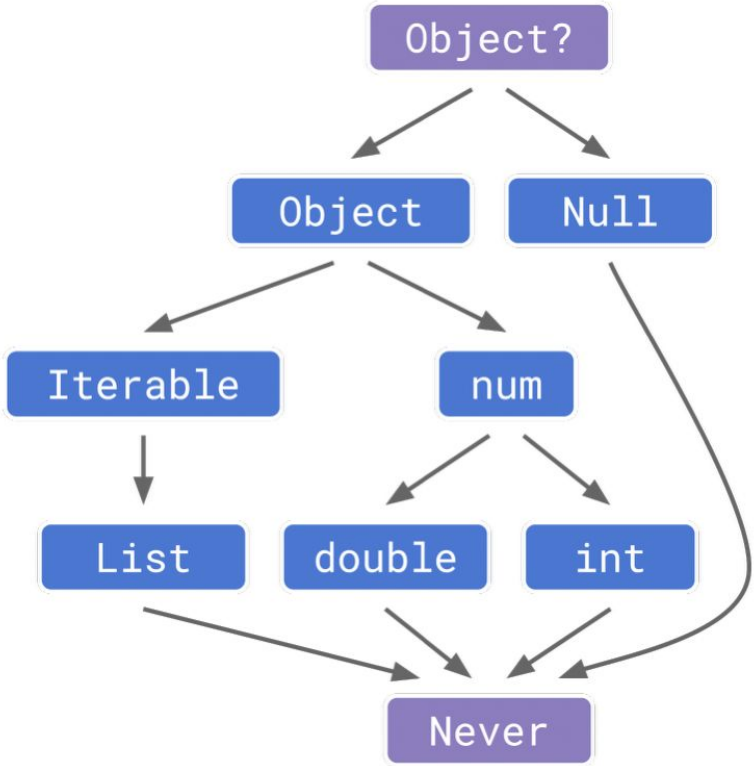
Null Safety



Flutter

العطاء الرقمي
Attaa Digital

Null Safety



Flutter

العطاء الرقمي
Attaa Digital

Null Safety

- Types are non-nullable by default and made nullable by adding `?`.
- Optional parameters must be nullable or have a default value. You can use `required` to make named parameters non-optional. Non-nullable top-level variables and static fields must have initializers. Non-nullable instance fields must be initialized before the constructor body begins.
- Method chains after null-aware operators short circuit if the receiver is `null`. There are new null-aware cascade (`?..`) and index (`?[]`) operators. The postfix null assertion “bang” operator (`!`) casts its nullable operand to the underlying non-nullable type.
- Flow analysis lets you safely turn nullable local variables and parameters into usable non-nullable ones. The new flow analysis also has smarter rules for type promotion, missing returns, unreachable code, and variable initialization.
- The `late` modifier lets you use non-nullable types and `final` in places you otherwise might not be able to, at the expense of runtime checking. It also gives you lazy-initialized fields.
- The `List` class is changed to prevent uninitialized elements.

Flutter

العطاء الرقمي
Attaa Digital

7

State Management

Flutter

العطاء الرقمي
Attaa Digital

State Management

Provider
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_provider_example

Riverpod
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_riverpod_example

setState
(StateFulWidget, StatelessWidget, InheritedWidget)

GetX

Redux

Fish - Redux

BloC / RX
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_bloc_example

GetIt

MobX

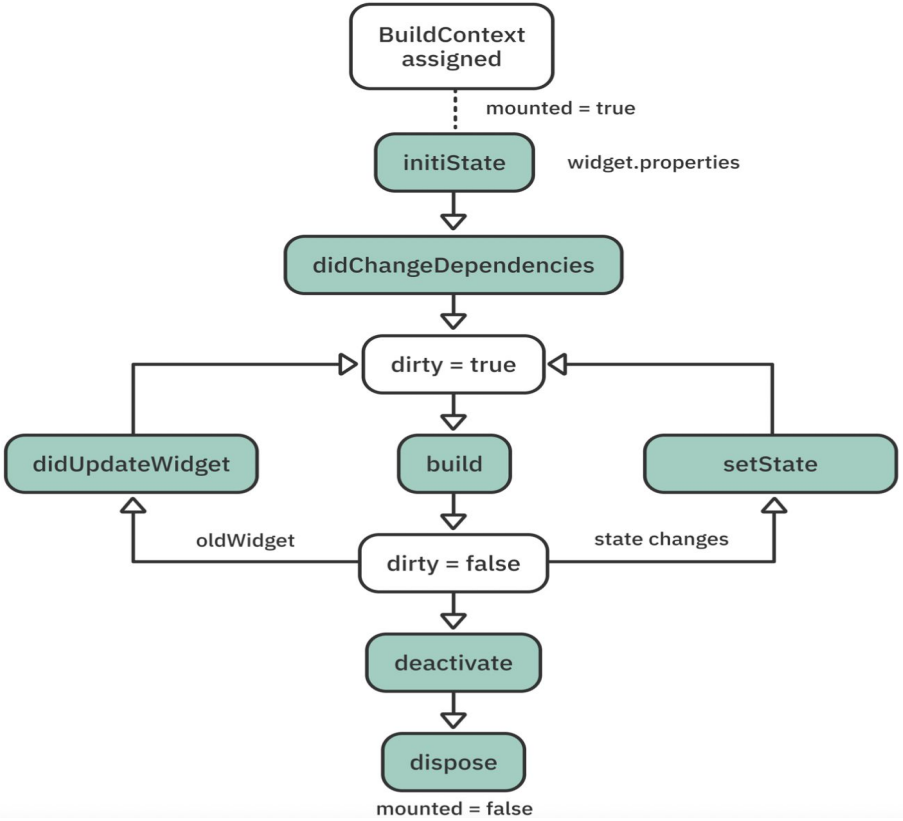
Binder

Flutter

العطاء الرقمي
Attaa Digital

State Management

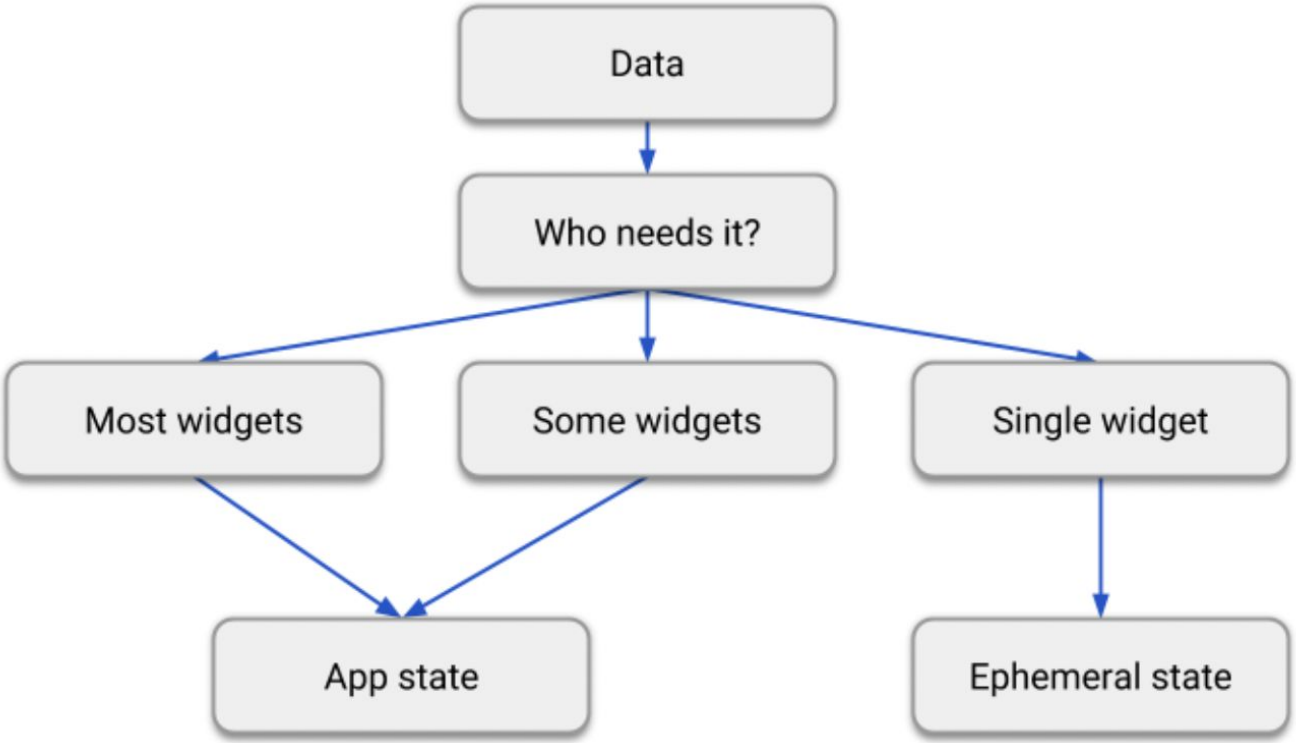
State Object Lifecycle



Flutter

العطاء الرقمي
Attaa Digital

State Management



State Management

Ephemeral state

Ephemeral state (sometimes called *UI state* or *local state*) is the state you can neatly contain in a single widget.

This is, intentionally, a vague definition, so here are a few examples.

- current page in a `PageView`
- current progress of a complex animation
- current selected tab in a `BottomNavigationBar`

State Management

App state

State that is not ephemeral, that you want to share across many parts of your app, and that you want to keep between user sessions, is what we call application state (sometimes also called shared state).

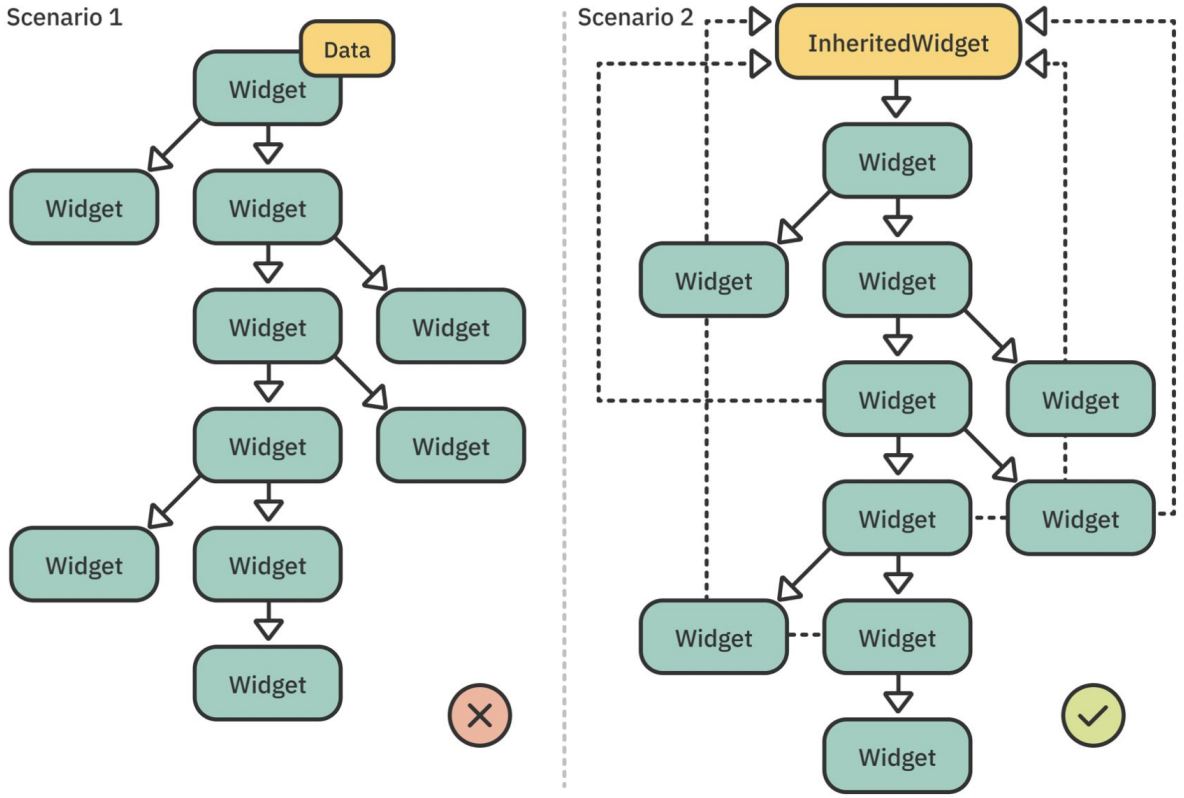
Examples of application state:

- User preferences
- Login info
- Notifications in a social networking app
- The shopping cart in an e-commerce app
- Read/unread state of articles in a news app

Flutter

العطاء الرقمي
Attaa Digital

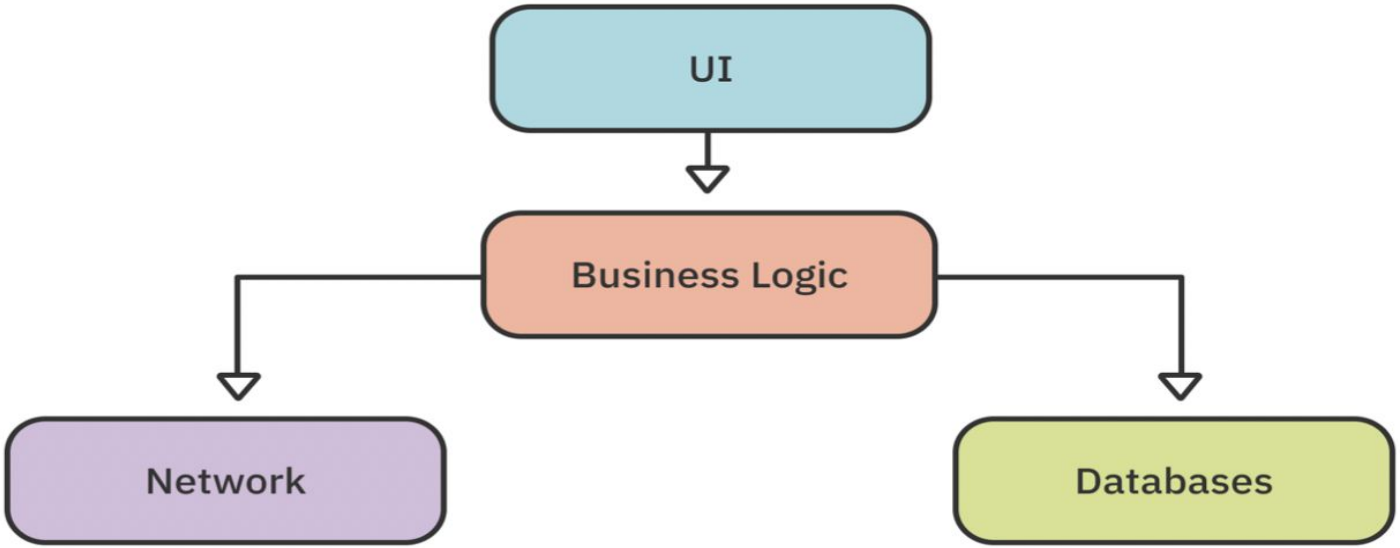
State Management



Flutter

العطاء الرقمي
Attaa Digital

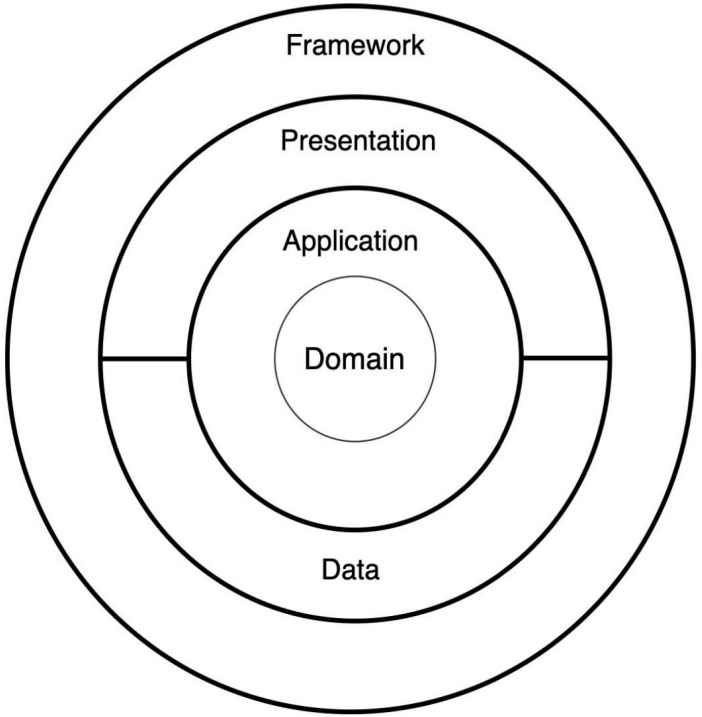
State Management



Flutter العطاء الرقمي Attaa Digital

State Management

Clean Architecture

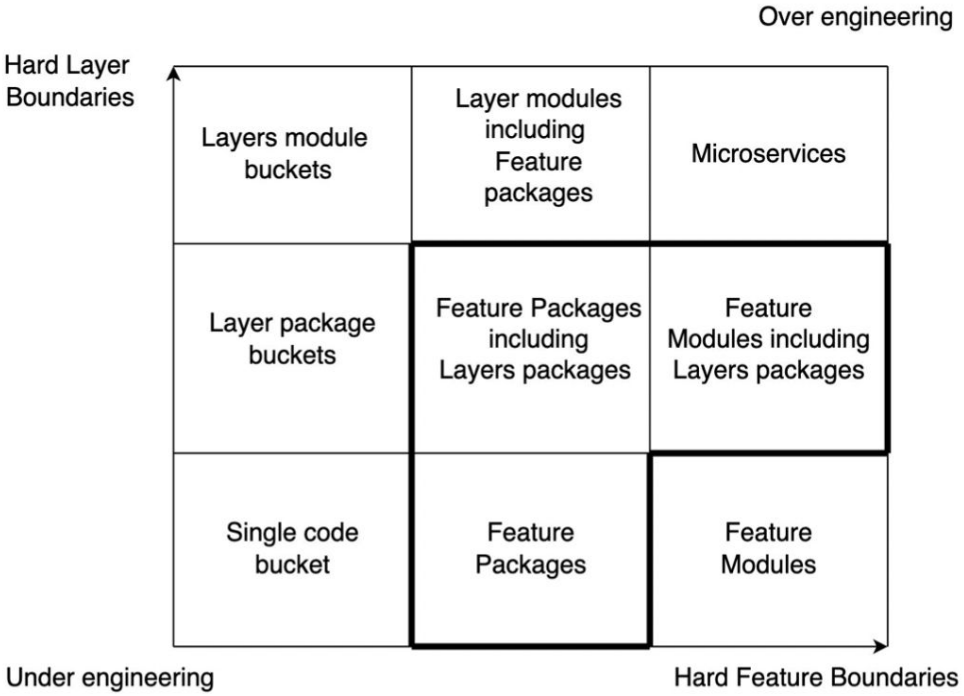


Enterprise Mobile Layering

Flutter العطاء الرقمي Attaa Digital

State Management

Clean Architecture



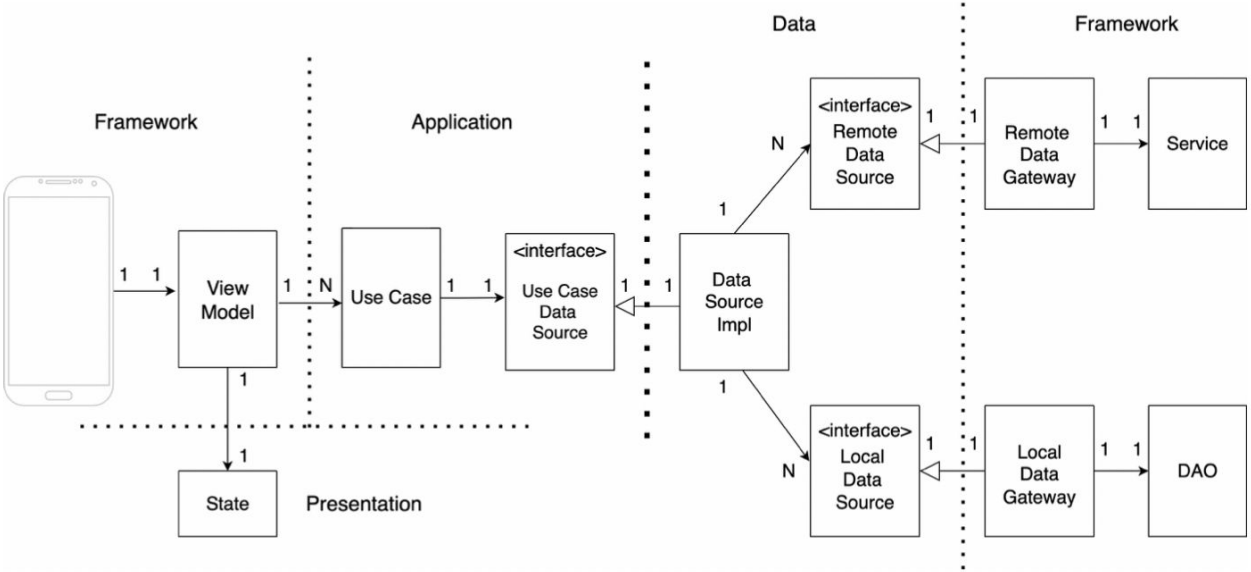
Architecture categorization based on boundaries.

Flutter

العطاء الرقمي
Attaa Digital

State Management

Clean Architecture

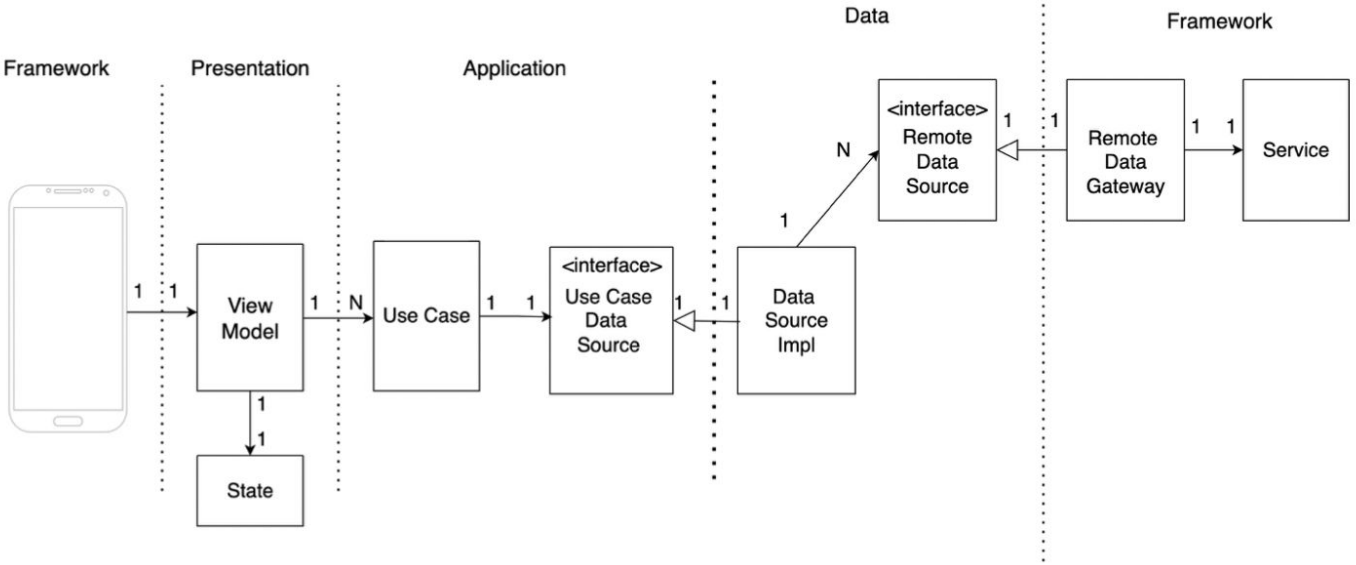


Microservices Architecture.

Flutter العطاء الرقمي Attaa Digital

State Management

Clean Architecture



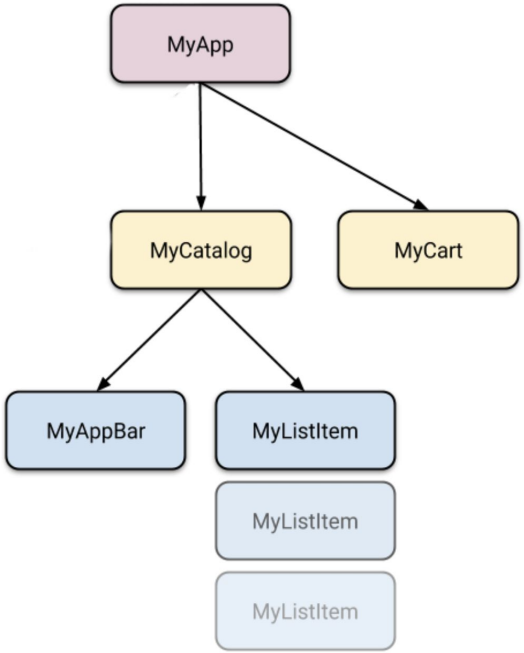
Pragmatic Clean Architecture.

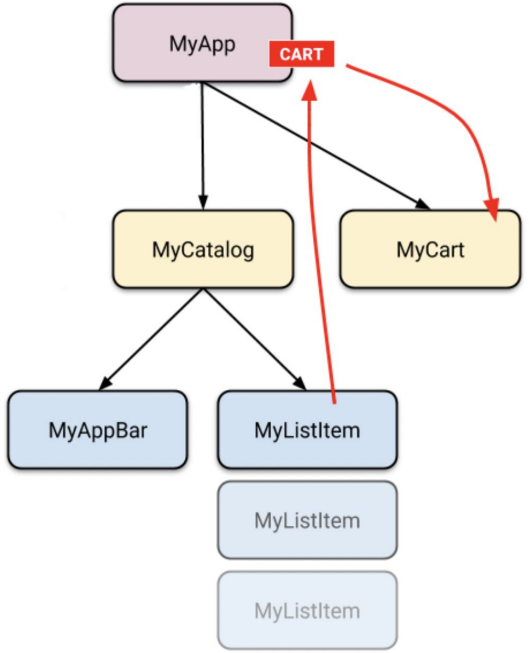
Flutter

العطاء الرقمي
Attaa Digital

State Management

Provider





Flutter

العطاء الرقمي
Attaa Digital

State Management

Provider

```dart
class CartModel extends ChangeNotifier {
  /// Internal, private state of the cart.
  final List<Item> _items = [];

  /// An unmodifiable view of the items in the cart.
  UnmodifiableListView<Item> get items => UnmodifiableListView(_items);

  /// The current total price of all items (assuming all items cost $42).
  int get totalPrice => _items.length * 42;

  /// Adds [item] to cart. This and [removeAll] are the only ways to modify the
  /// cart from the outside.
  void add(Item item) {
    _items.add(item);
    // This call tells the widgets that are listening to this model to rebuild.
    notifyListeners();
  }

  /// Removes all items from the cart.
  void removeAll() {
    _items.clear();
    // This call tells the widgets that are listening to this model to rebuild.
    notifyListeners();
  }
}
```

Flutter

العطاء الرقمي
Attaa Digital

State Management

Provider

```
void main() {
  runApp(
    ChangeNotifierProvider(
      create: (context) => CartModel(),
      child: const MyApp(),
    ),
  );
}
```

```
void main() {
  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (context) => CartModel()),
        Provider(create: (context) => SomeOtherClass()),
      ],
      child: const MyApp(),
    ),
  );
}
```

Flutter

العطاء الرقمي
Attaa Digital

State Management

Provider

```
// DON'T DO THIS
return Consumer<CartModel>(
  builder: (context, cart, child) {
    return HumongousWidget(
      // ...
      child: AnotherMonstrousWidget(
        // ...
        child: Text('Total price: ${cart.totalPrice}'),
      ),
    );
  },
);
```

```
// DO THIS
return HumongousWidget(
  // ...
  child: AnotherMonstrousWidget(
    // ...
    child: Consumer<CartModel>(
      builder: (context, cart, child) {
        return Text('Total price: ${cart.totalPrice}');
      },
    ),
  ),
);
```

Flutter

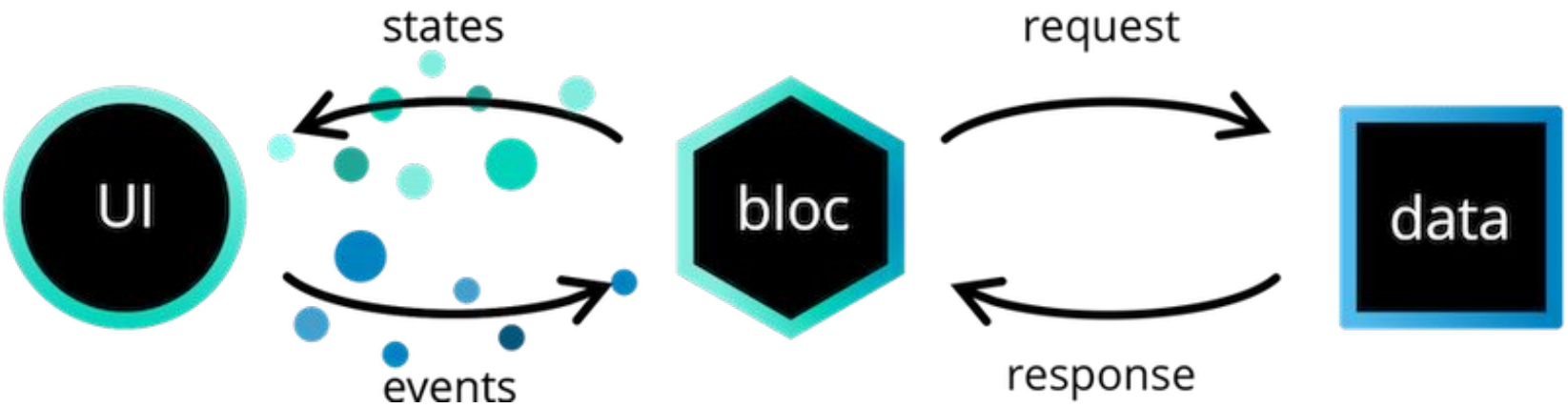
العطاء الرقمي
Attaa Digital

State Management

Bloc

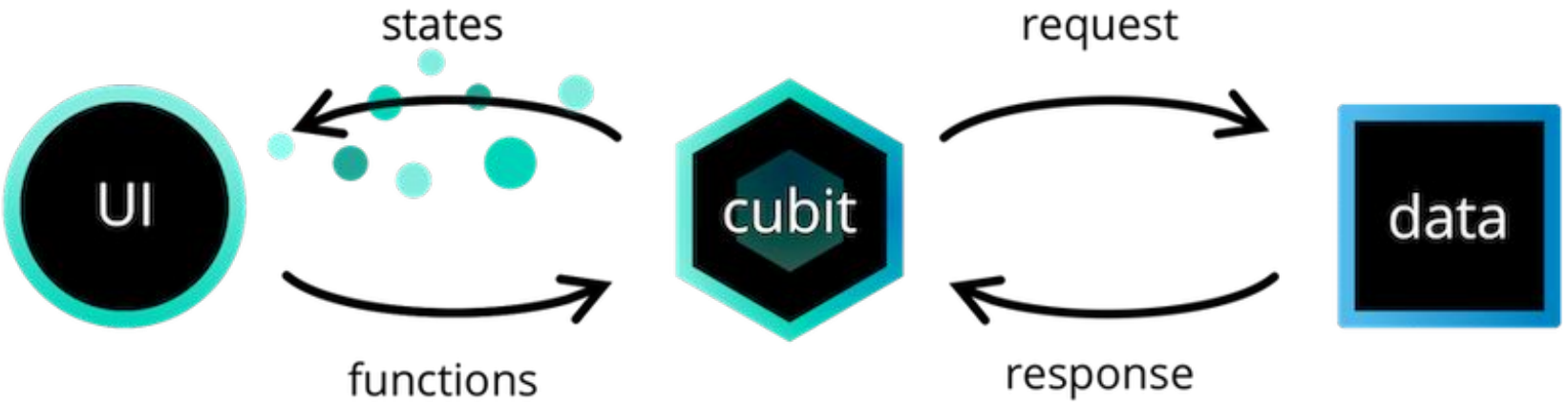
State Management

Bloc



State Management

Bloc

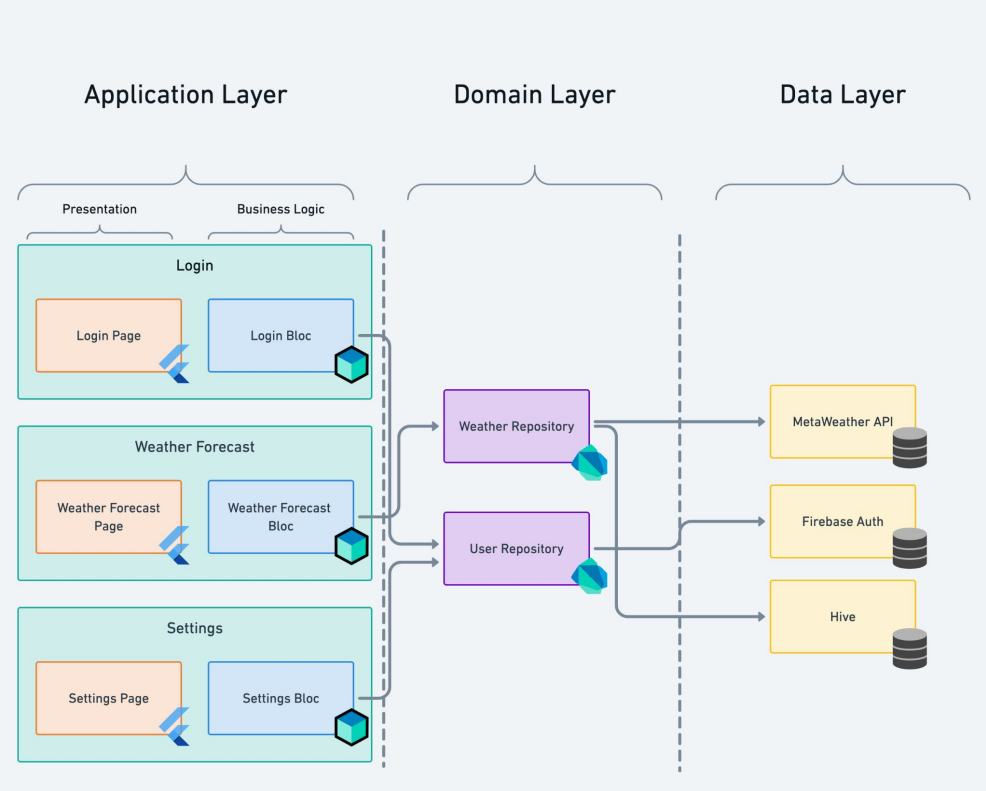


Flutter

العطاء الرقمي
Attaa Digital

State Management

Bloc



Flutter

العطاء الرقمي
Attaa Digital

State Management

Riverpod

Provider Type	Provider Create Function	Example Use Case
Provider	Returns any type	A service class / computed property (filtered list)
StateProvider	Returns any type	A filter condition / simple state object
FutureProvider	Returns a Future of any type	A result from an API call
StreamProvider	Returns a Stream of any type	A stream of results from an API
StateNotifierProvider	Returns a subclass of StateNotifier	A complex state object that is immutable except through an interface
ChangeNotifierProvider	Returns a subclass of ChangeNotifier	A complex state object that requires mutability

Flutter

العطاء الرقمي
Attaa Digital

State Management

Riverpod

The `.family` modifier has one purpose: Getting a unique provider based on external parameters.

Some common use-cases for `family` would be:

- Combining FutureProvider with `.family` to fetch a `Message` from its ID
- Passing the current `Locale` to a provider, so that we can handle translations

Flutter

العطاء الرقمي
Attaa Digital

State Management

Riverpod

.autoDispose

A common use case is to destroy the state of a provider when it is no-longer used.

There are multiple reasons for doing so, such as:

- When using Firebase, to close the connection and avoid unnecessary cost.
- To reset the state when the user leaves a screen and re-enters it.

Providers come with built-in support for this use case, through the `.autoDispose` modifier.

Flutter العطاء الرقمي Attaa Digital

State Management

Riverpod

- **ref.read()**

As the name implies, we read a notifier and or its value and not trigger for changes. This is exactly the same thing we do when we use the Provider counterpart of context.read() or Provider.of(context, listen: false). This is very important as it gives us the possibility of accessing the provider and make changes outside of the build context that is, in the onTap() etc. Never call the ref.read() inside build context as much as it does not trigger rebuild of the widget when changes occur and could be seen as a way of preserving and reducing the number of useless UI rebuilds, it is highly contested against as the official docs for riverpod will say

State Management

Riverpod

- **ref.watch()**

This is probably the most versatile of them all. It is used to get the value of a provider and also watch for changes. It is typically used only inside the build method or inside another provider. Never use it inside any of the lifecycle methods like initstate or setstate as there will be deadlock and conflict since both may be triggering a rebuild of the widget and still depending on each other

State Management

Riverpod

- **ref.listen()**

This is a very useful one as it is similar in every way to ref.watch, but that it does not trigger a rebuild of the widget but instead it watches the provider and when a change occurs you can call a function. This is useful perhaps if for navigation etc.

State Management

Riverpod

- **provider.select()**

The other day I was having a conversation with a friend who brought to my attention from logs that riverpod with ChangeNotifier rebuilds a few more times than provider and bloc and more rebuilds when used am immutable state like with statenotifier. This is understandable as few numbers of rebuilds is not a stronghold that riverpod is proud of, but the solution it provides is geared at different difficulties. But if you are concerned about the number of rebuilds you can mitigate this with the select method. This works in that, in a state or class with more than one property, you can decide to listen to changes only from one or a few properties (the provider listens to the whole object by default).

Flutter

العطاء الرقمي
Attaa Digital

State Management

Riverpod

```dart
class BlueSquare extends ConsumerWidget {
  const BlueSquare({
    Key? key,
  }) : super(key: key);

  @override
  Widget build(BuildContext context, WidgetRef ref) {
    // But if you want only the size and dont want unnecessary rebuild when
    // color changes
    final size =
        ref.watch(blueSquareProvider.select((blueSquare) => blueSquare.size));

    return Container(
      width: size,
      height: size,
      color: Colors.blue,
    );
  }
}
```

8

المفاتيح

Flutter

العطاء الرقمي
Attaa Digital

المفاتيح

When? Adding - Removing - Reordering (collection of widgets of the same type of some state)

GlobalKey

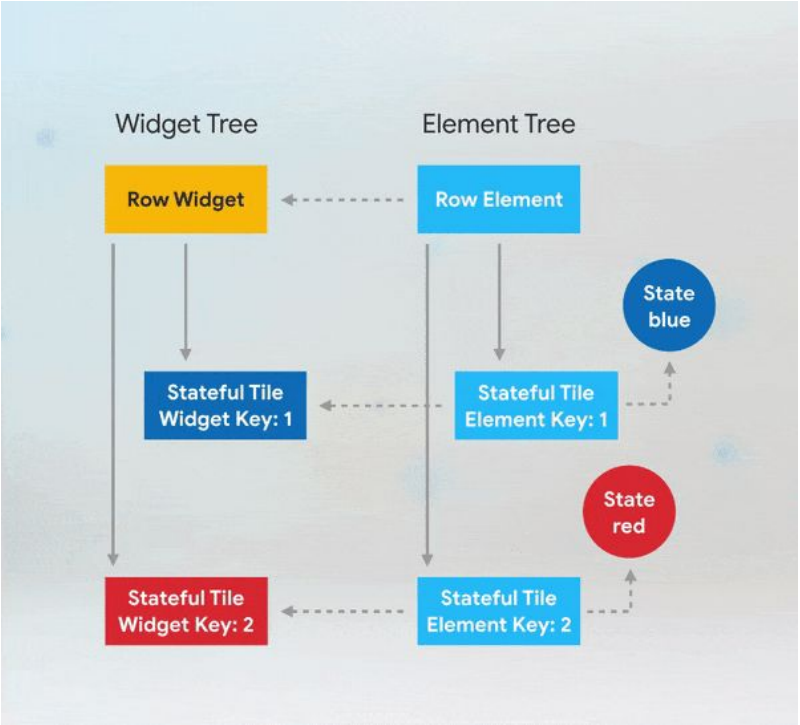
LocalKey (
ValueKey('Mouaz Shahmeh') , ObjectKey(1, 2, 3, 4), UniqueKey(), PageStorageKey()scrollLocation)

Examples: ScaffoldKey and FormKey
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_keys_example

Flutter

العطاء الرقمي
Attaa Digital

المفاتيح



9

التنقل

Flutter العطاء الرقمي Attaa Digital

التنقل

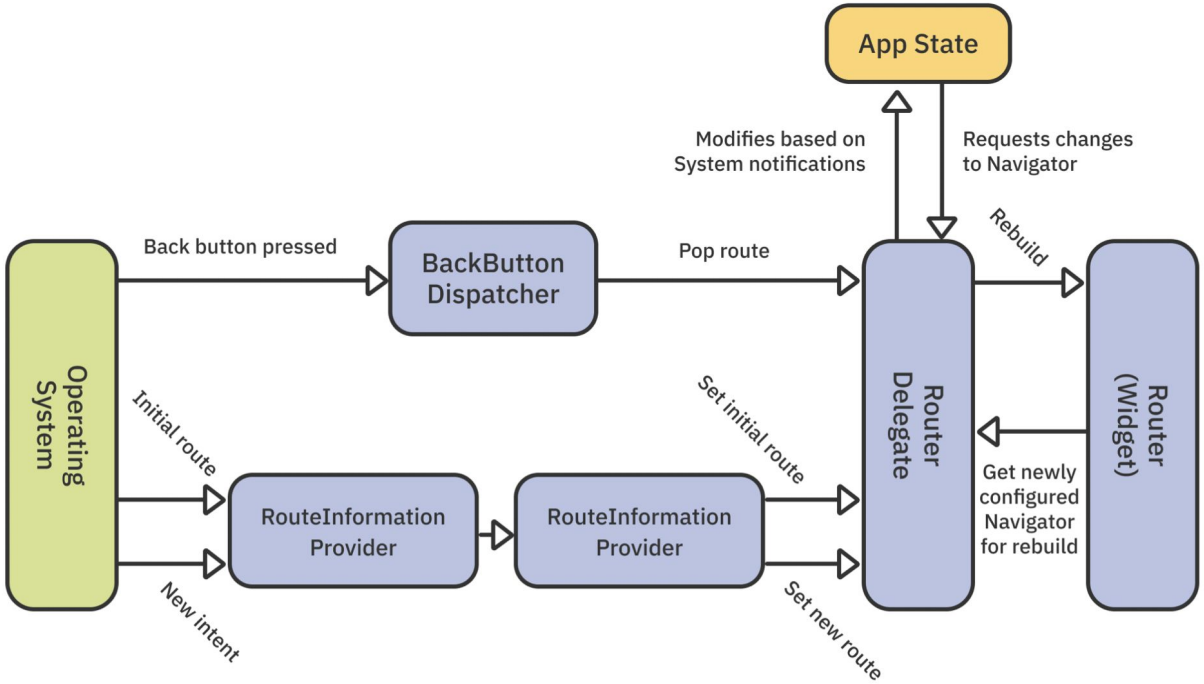
GoRouter (push - go) (Declarative navigation - no context)
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_gorouter_example

Navigator 2 (push - pop - pushReplacement - pushNamed - popUntill ...)

- https://pub.dev/packages/beamer
- https://pub.dev/packages/flow_builder
- https://pub.dev/packages/fluro
- https://pub.dev/packages/vrouter
- https://pub.dev/packages/auto_route

التنقل

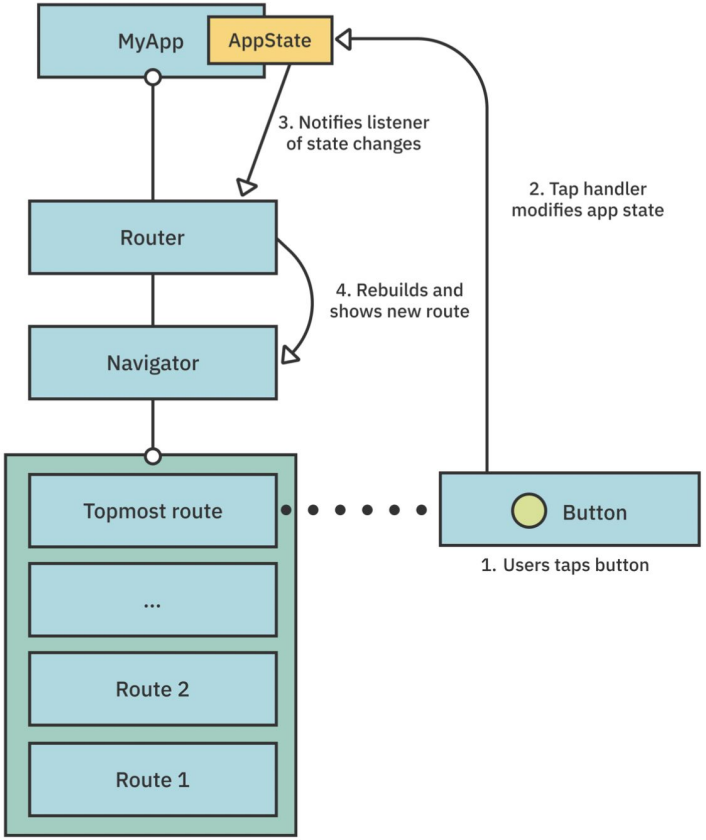
Here are the new abstractions that make up Navigator 2.0's declarative API:



Flutter

العطاء الرقمي
Attaa Digital

التنقل



Flutter

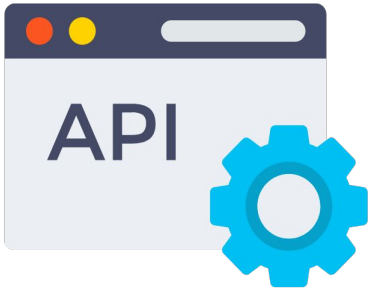
العطاء الرقمي
Attaa Digital

10

خيارات قواعد البيانات

خيارات قواعد البيانات

Firebase



Shared
Preferences

Flutter

العطاء الرقمي
Attaa Digital

خيارات قواعد البيانات

Firebase (FlutterFire) - BAAS (Backend As A Service)
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_firebase_crud_example

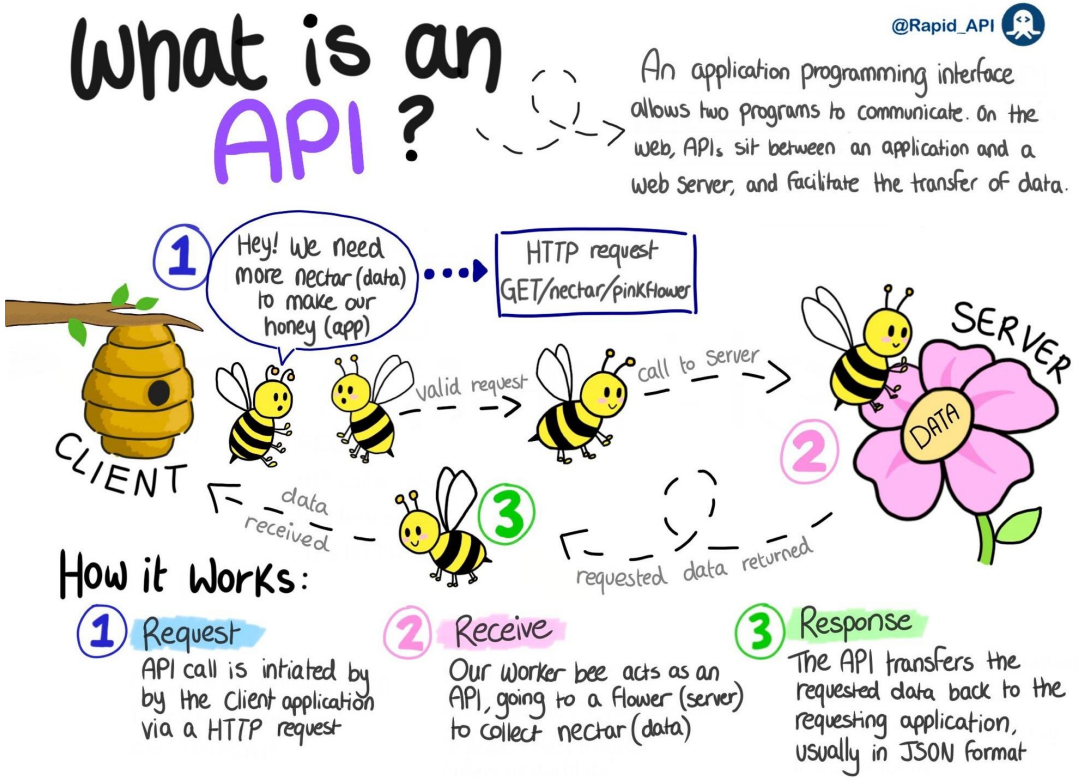
Supabase (PostgreSql) - BAAS (Backend As A Service)
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_supabase_crud_example

SqfLite - SqfEntity (local)
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_sqflite_crud_example

Shared Preferences (path)
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_shared_preferences_crud_example

API http (json)
https://github.com/Eng-Mouaz-M-AlShahmeh/flutter_development_api_http_crud_example

خيارات قواعد البيانات



Flutter

العطاء الرقمي
Attaa Digital

خيارات قواعد البيانات

DIFFERENT TYPES OF API

RapidAPI.com/hub
@Rapid_API

	REST	GRAPHQL	SOAP	RPC
STRUCTURE	FOLLOWS SIX ARCHITECTURAL CONSTRAINTS	SCHEMA AND TYPE	MESSAGE STRUCTURE	LOCAL PROCEDURAL CALLS
FORMAT	JSON, XML, HTML, PLAIN TEXT	JSON	XML	JSON, XML, FLATBUFFERS, ETC
ADVANTAGES	FLEXIBLE IN TERMS OF DATA FORMAT AND STRUCTURE	SOLVES OVER-FETCHING AND UNDER-FETCHING	HIGHLY SECURE AND EXTENSIBLE	LIGHTWEIGHT PAYLOADS MAKE IT HIGH PERFORMING
USE CASES	RESOURCES BASED APPS	MOBILE APIS	PAYMENT GATEWAYS	COMMAND-FOCUSED SYSTEMS

Flutter

العطاء الرقمي
Attaa Digital

خيارات قواعد البيانات

Flutter

العطاء الرقمي
Attaa Digital

11

اختبارات الأداء

Flutter

العطاء الرقمي
Attaa Digital

اختبارات الأداء

Dev Tools

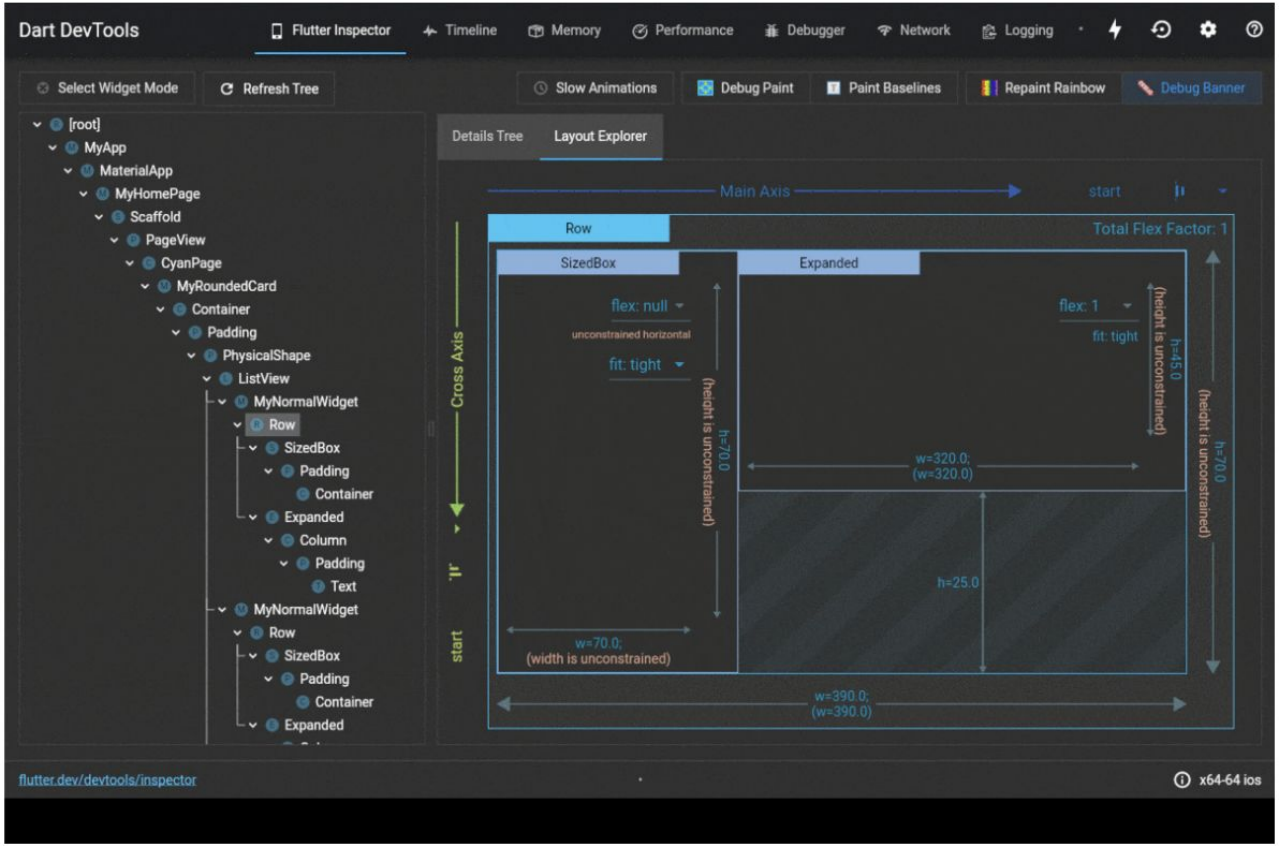
Flutter Doctor

Flutter Debug Mode (Code Trace)

Flutter

العطاء الرقمي
Attaa Digital

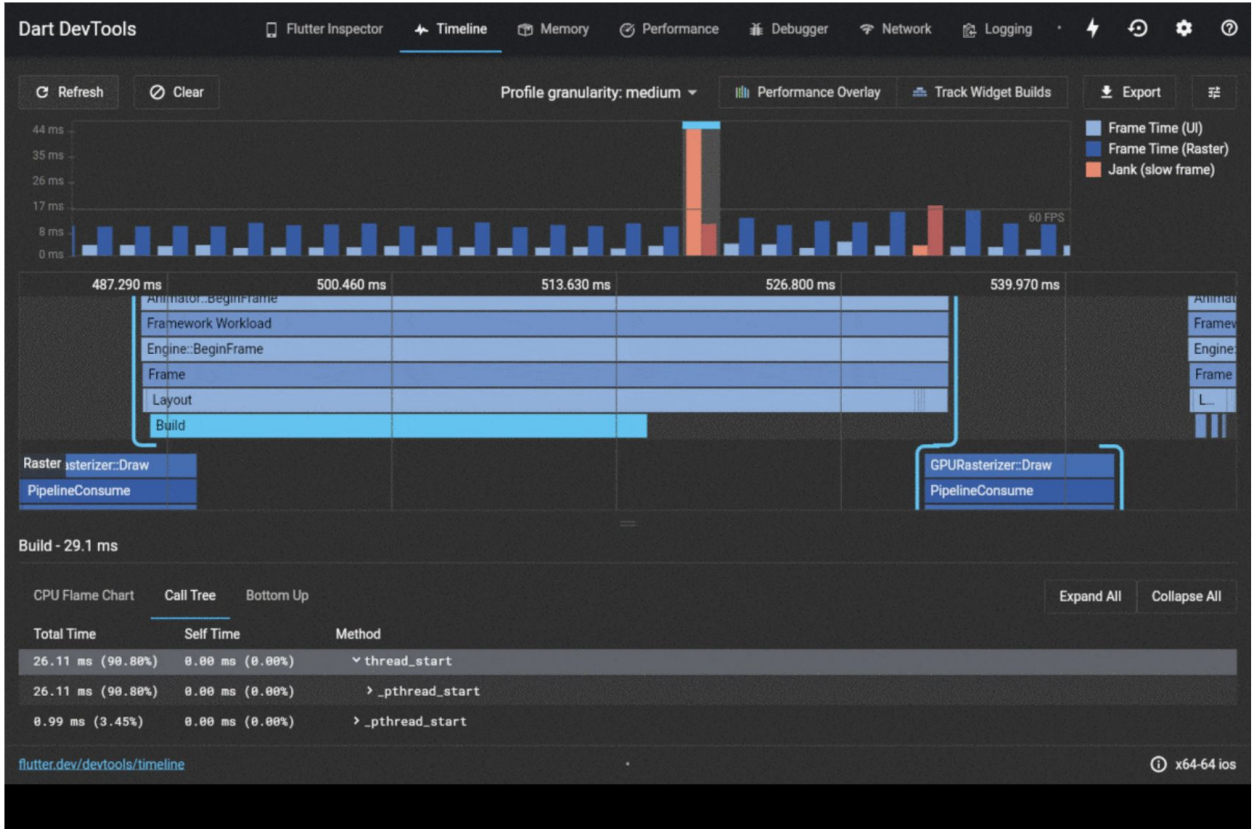
اختبارات الأداء



Flutter

العطاء الرقمي
Attaa Digital

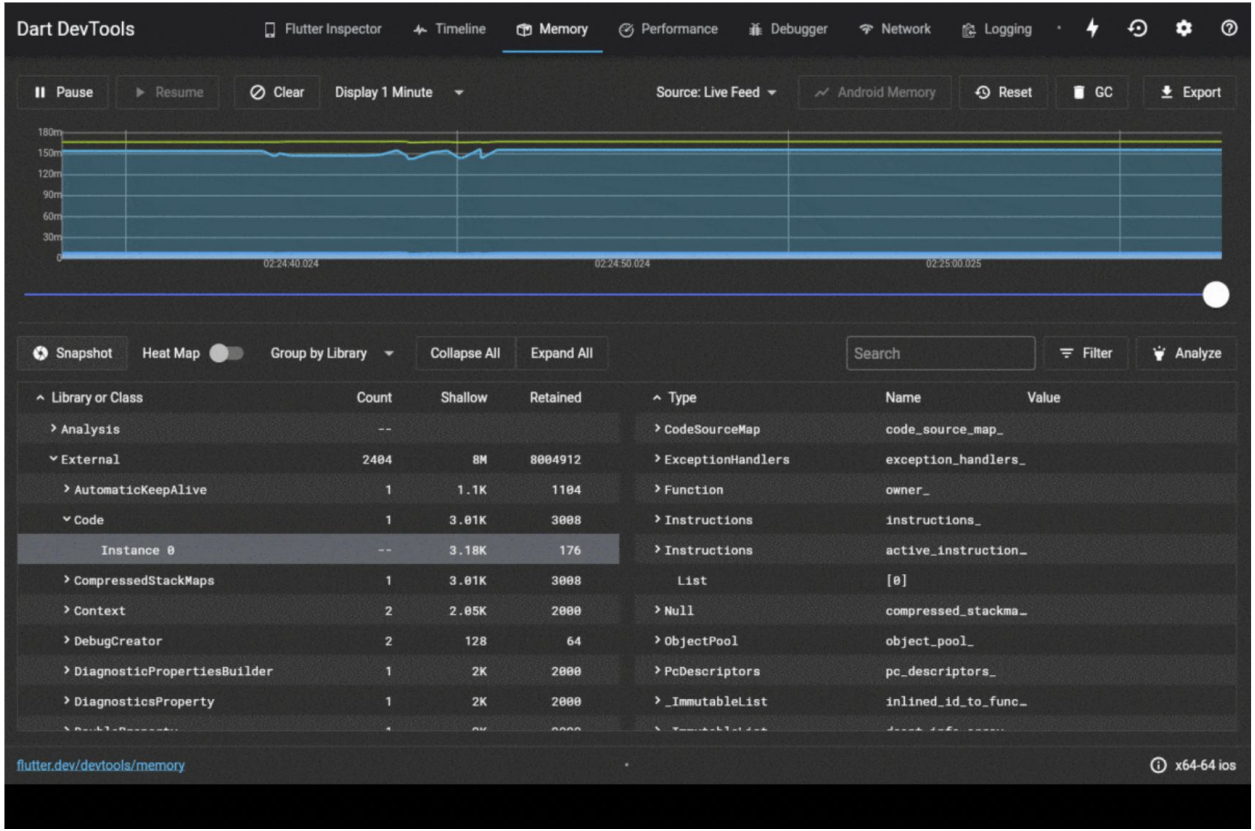
اختبارات الأداء



Flutter

العطاء الرقمي
Attaa Digital

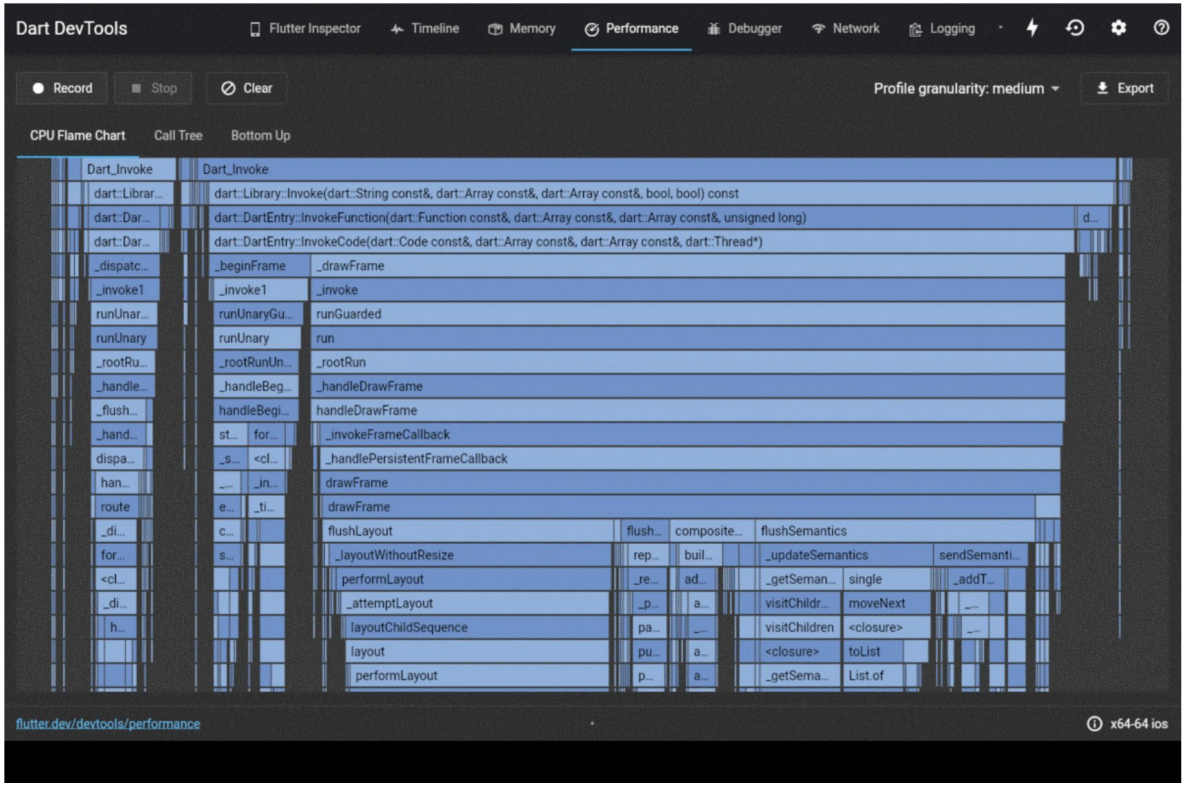
اختبارات الأداء



Flutter

العطاء الرقمي
Attaa Digital

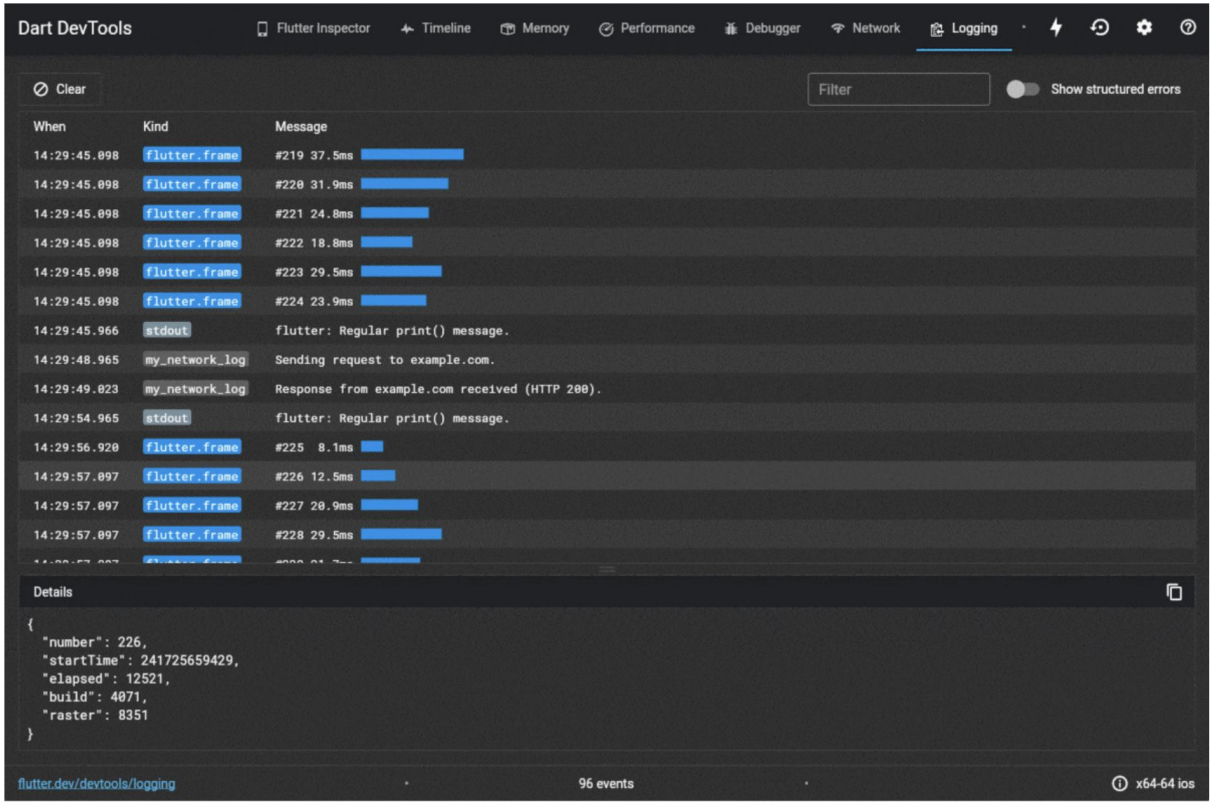
اختبارات الأداء



Flutter

العطاء الرقمي
Attaa Digital

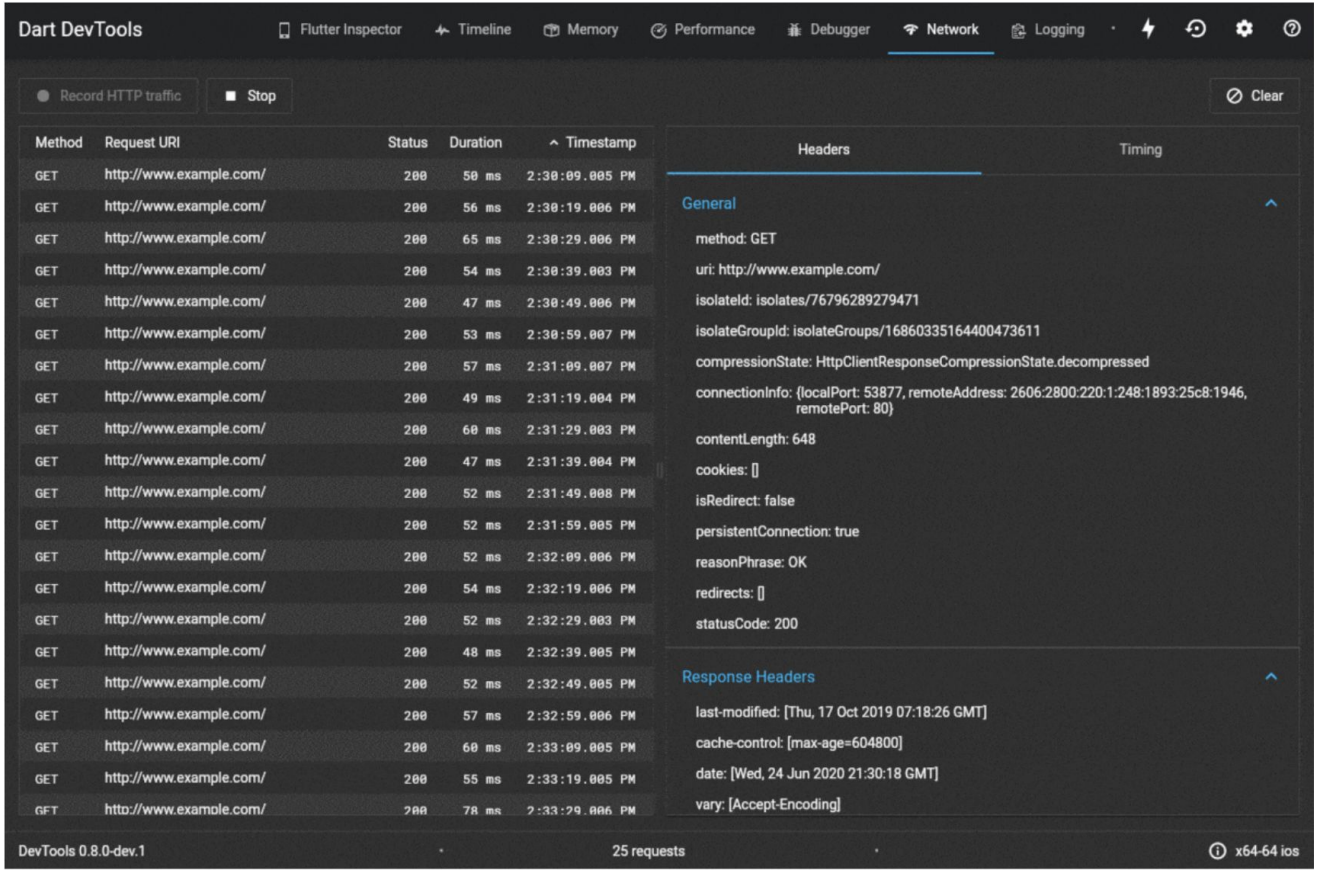
اختبارات الأداء



Flutter

العطاء الرقمي
Attaa Digital

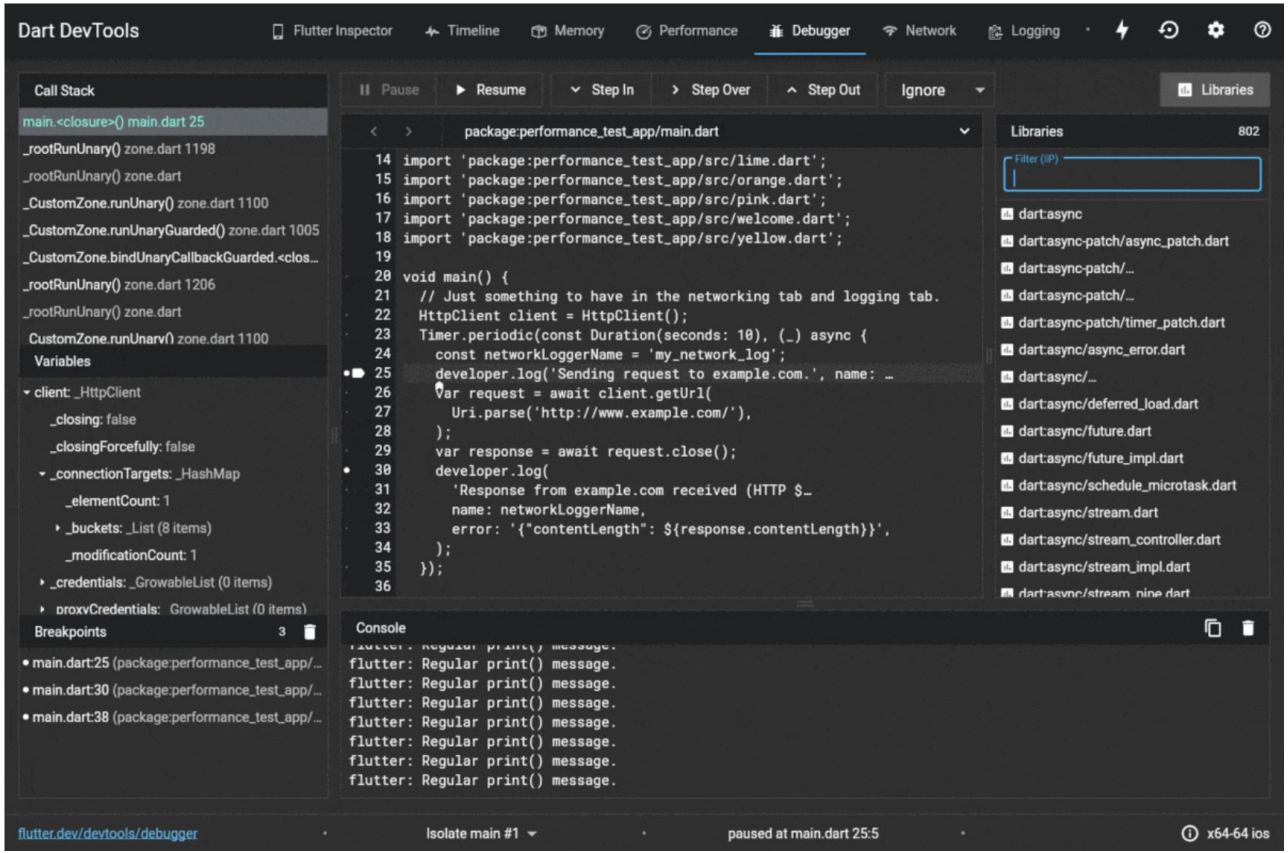
اختبارات الأداء



Flutter

العطاء الرقمي
Attaa Digital

اختبارات الأداء



Flutter

العطاء الرقمي
Attaa Digital

12

أنماط بناء التطبيقات

Flutter

العطاء الرقمي
Attaa Digital

أنماط بناء التطبيقات

Debug (print - debugShowCheckedModeBanner - run - hot reload & restart - debug mode)

Profile (run - real performance - real device - hot reload & restart)

Release (archive in xcode - apk or bundle in android - app signing - unique id - play store - app store)

تطبيق عملي

Flutter

العطاء الرقمي
Attaa Digital

المشروع

Flutter

العطاء الرقمي
Attaa Digital

تفاصيل المشروع

- عمل مشروع Flutter جديد باستخدام برنامج الأكواد واخر اصدار لـ Flutter SDK بدعم Null Safety.
- استخدام MediaQuery.of(context) في شاشات المشروع لدعم مقاسات الشاشات.
- استخدام واحدة من ادارة الحالة في التطبيق (provider - riverpod - bloc)
- استخدام GoRouter للتنقل بين الشاشات في التطبيق
- استخدام واحدة من خيارات قواعد البيانات لعمل عملية أو أكثر من عمليات CRUD (Firebase - Supabase - Sqfite or Sqfentity - api http - shared preferences)
- رفع المشروع في ريبو Github جديد .
- تصوير 3 شاشات من التطبيق على الأقل
- إرسال ايميل بعنوان "دورة تطوير التطبيقات باستخدام Flutter - الاسم الثلاثي" الى m.m.shahmeh@gmail.com

  إرفاق رابط الـ github ضمن الايميل

  إرفاق صور التطبيق بشكل ملف مضغوط (rar - zip)

13

ملخص ومراجع

Flutter

العطاء الرقمي
Attaa Digital

ملخص ومراجع

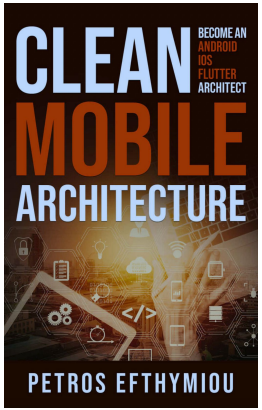
flutter.dev

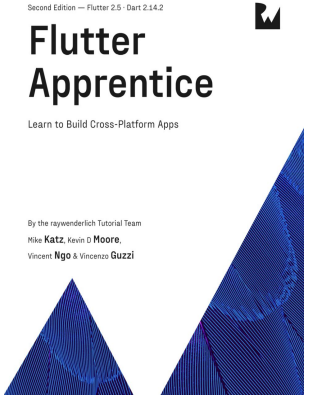
The Boring Flutter Development Show (Youtube)

Widget of the Week (Youtube)

pub.dev

fuchsia.dev (Zircon is Dart package)





Flutter

العطاء الرقمي
Attaa Digital

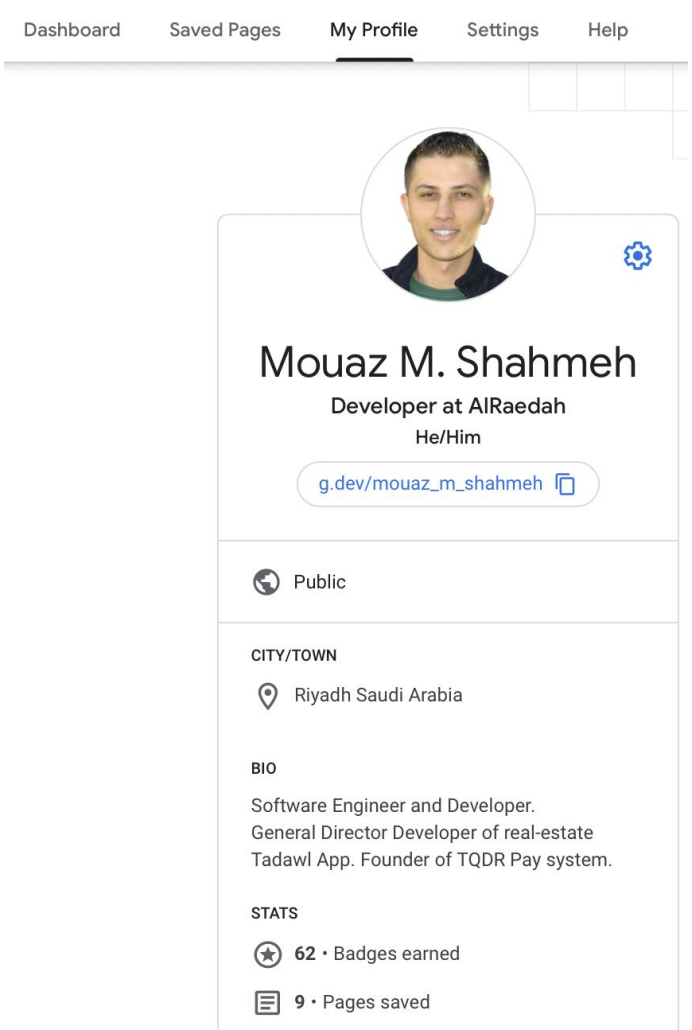
Ask Me Anything

m.m.shahmeh@gmail.com

g.dev/mouaz_m_shahmeh

LINKS

- github.com/Eng-Mouaz-M-AlShahmeh
- linkedin.com/in/mouaz-shahmeh
- stackoverflow.com/users/18449528
- twitter.com/mouaz_m_shahmeh



شكراً لكم