Студенческая
ИТ-лаборатория

Введение в кроссплатформенную мобильную разработку на Flutter

Дмитрий Дружин
Product Manager в Effective

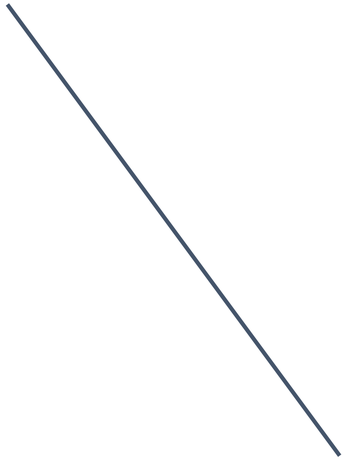
Студенческая
ИТ-лаборатория

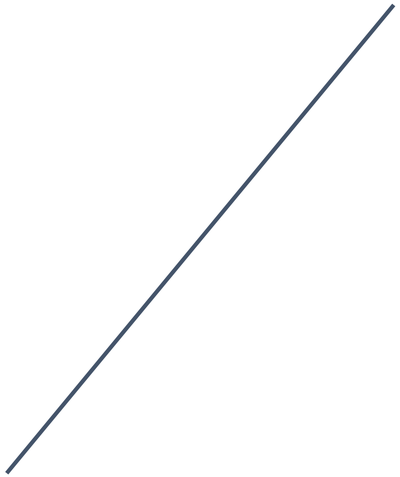
effectiveband

Что такое Flutter и для чего его используют?

Что такое Flutter?

effectiveband





А что, если не Flutter?

effectiveband

Технология	Flutter	PWA	React Native	KMP	Compose Multiplatform
Технология отрисовки UI	Skia (Canvas)	HTML	Нативно	Jetpack Compose / SwiftUI	Skiko (Canvas)
Скорость работы	Быстрый	Медленный	Средний	Быстрый	Средний
Сложность изучения	Средний	Простой	Средний	Сложный	Средний
Доступ к нативному API	Полный	Отсутствует	Частичный	Полный	Полный

Специфика Dart

effectiveband

```dart
String? nullableString; // Может быть null
String nonNullableString = "Hello"; // Не может быть null


String nullName;
print(nullName.length); // Не скомпилируется, потому что nullName = null

String? name;
print(name.length);  // Не скомпилируется, потому что length может и не быть
print(name!.length);  // Упадёт, если name == null


String? myName;
print(myName ?? "Default Name"); // Ставит "Default Name", если там null
```

Null Safety

```dart
// Одинаково?
const int value = 10;
value = 11; // Ошибка

final String girlName = "Alice";
girlName = "Bob"; // Ошибка


// Не совсем:
const String word; // Нужно сразо проинициализировать

final String maleName; // Можно проинициализировать позже
maleName = "Bob";
maleName = "Alice";
```

final поля

Прочие особенности языка

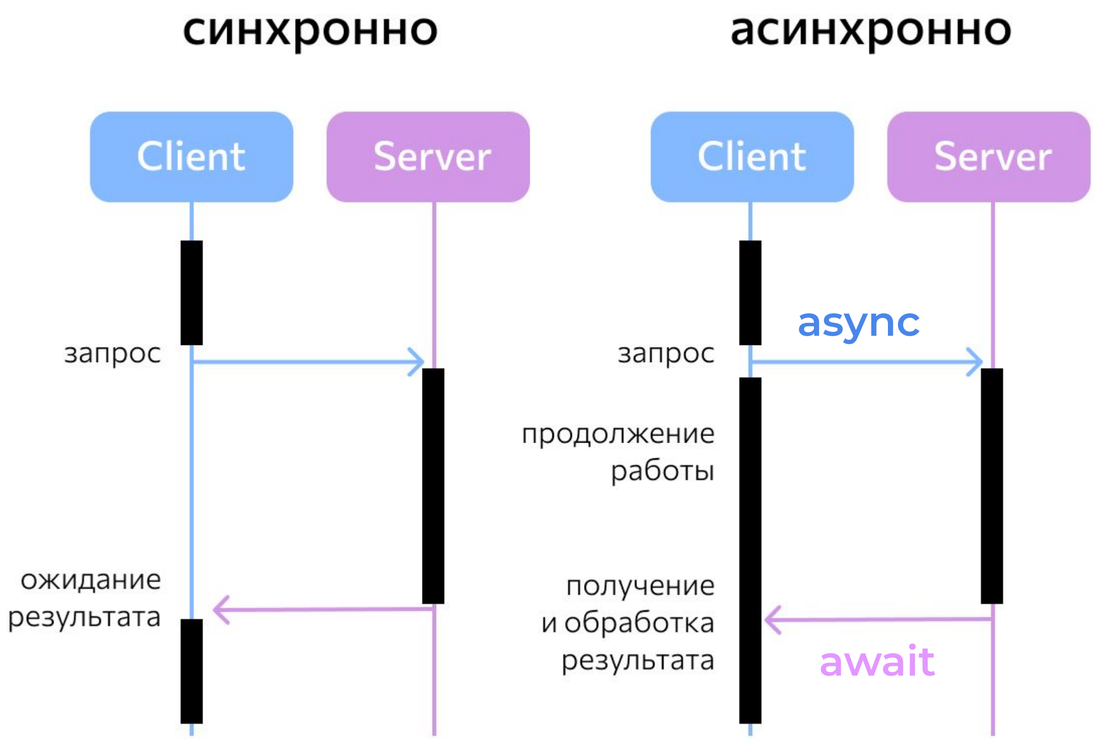
Студенческая
ИТ-лаборатория

effectiveband

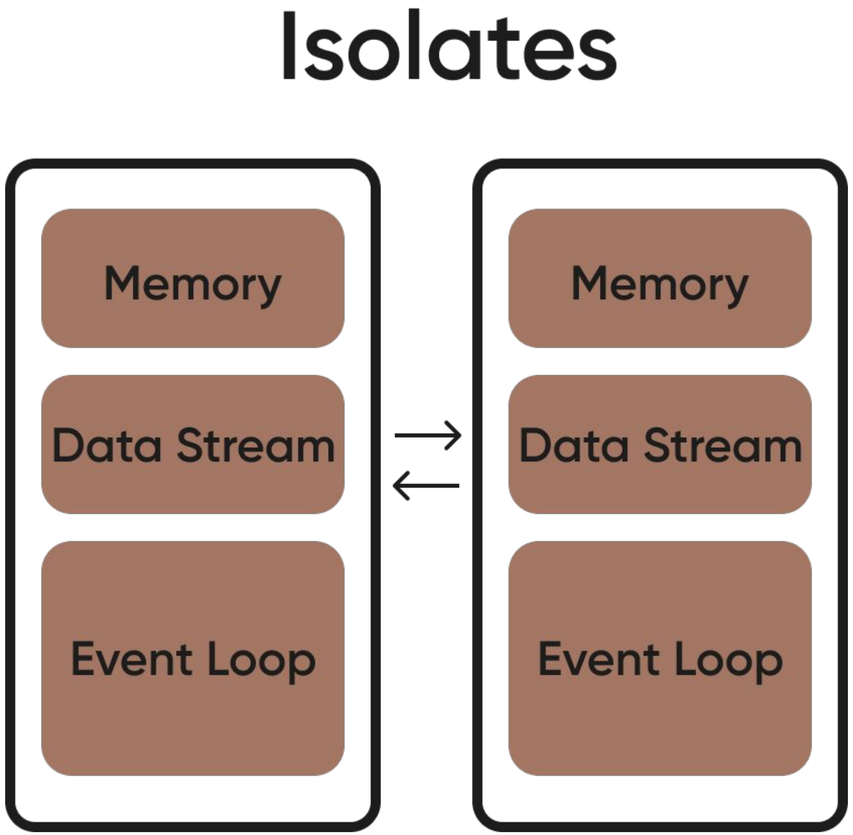
Асинхронность и многопоточность

Асинхронность и многопоточность

effectiveband



async-await



isolates(compute)

Студенческая
ИТ-лаборатория

effectiveband

Вёрстка и виджеты

Основные виджеты

Stateless

```dart
class MyStatelessWidget extends StatelessWidget {
  const MyStatelessWidget({super.key});

  @override
  Widget build(BuildContext context) {
    return const Text('Привет, Flutter!');
  }
}
```

Stateful

```dart
class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({super.key});

  @override
  State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Column(
      children: [
        Text('Счетчик: $_counter'),
        ElevatedButton(
          onPressed: _incrementCounter,
          child: const Text('Увеличить'),
        ), // ElevatedButton
      ],
    ); // Column
  }
}
```

Основные виджеты

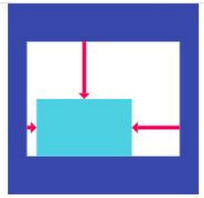
effectiveband

Layout widgets

UI > Widgets > Layout

Arrange other widgets columns, rows, grids, and many other layouts.

Single-child layout widgets

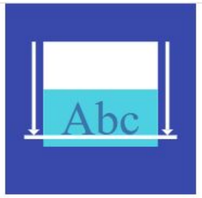


Align

A widget that aligns its child within itself and optionally sizes itself based on the child's size.

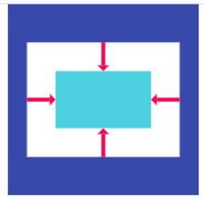
AspectRatio

A widget that attempts to size the child to a specific aspect ratio.



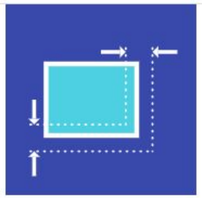
Baseline

Container that positions its child according to the child's baseline.



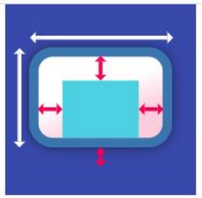
Center

Alignment block that centers its child within itself.



ConstrainedBox

A widget that imposes additional constraints on its child.



Container

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

Basic widgets

UI > Widgets > Basics

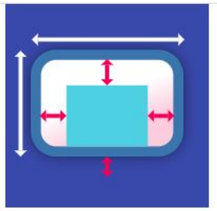
Widgets to know before building your first Flutter app.

AppBar

Container that displays content and actions at the top of a screen.

Column

Layout a list of child widgets in the vertical direction.



Container

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

ElevatedButton

A Material Design elevated button. A filled button whose

FlutterLogo

The Flutter logo, in widget form. This widget respects the

Icon

A Material Design icon.

Полный каталог

Material Widgets

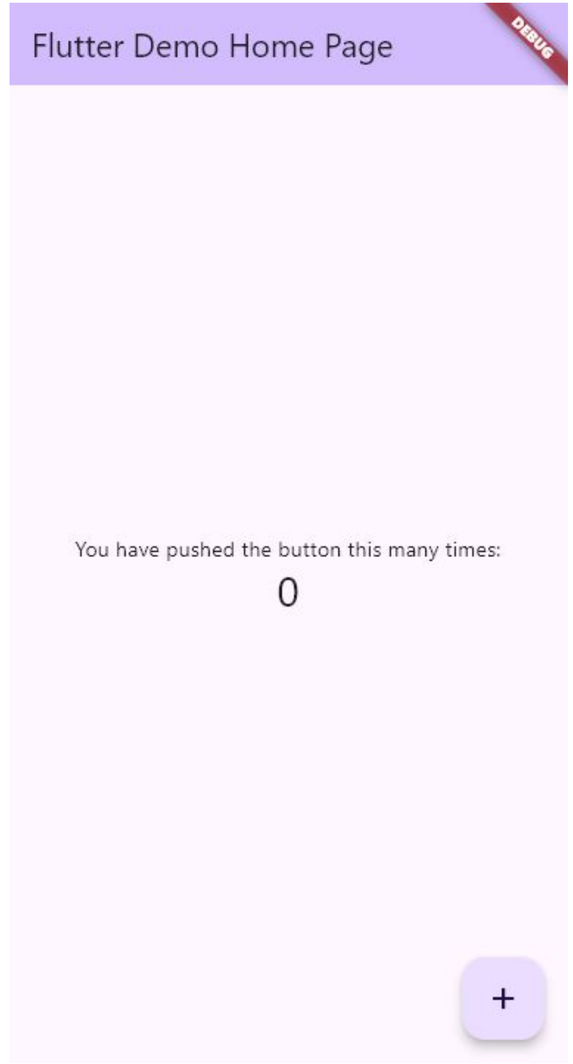
Декларативная вёрстка приложения

effectiveband

```dart
class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: Text(widget.title),
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text('You have pushed the button this many times:'),
            Text('$_counter', style: Theme.of(context).textTheme.headlineMedium)
          ], // <Widget>[]
        ), // Column
      ), // Center
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ), // FloatingActionButton
    ); // Scaffold
  }
}
```



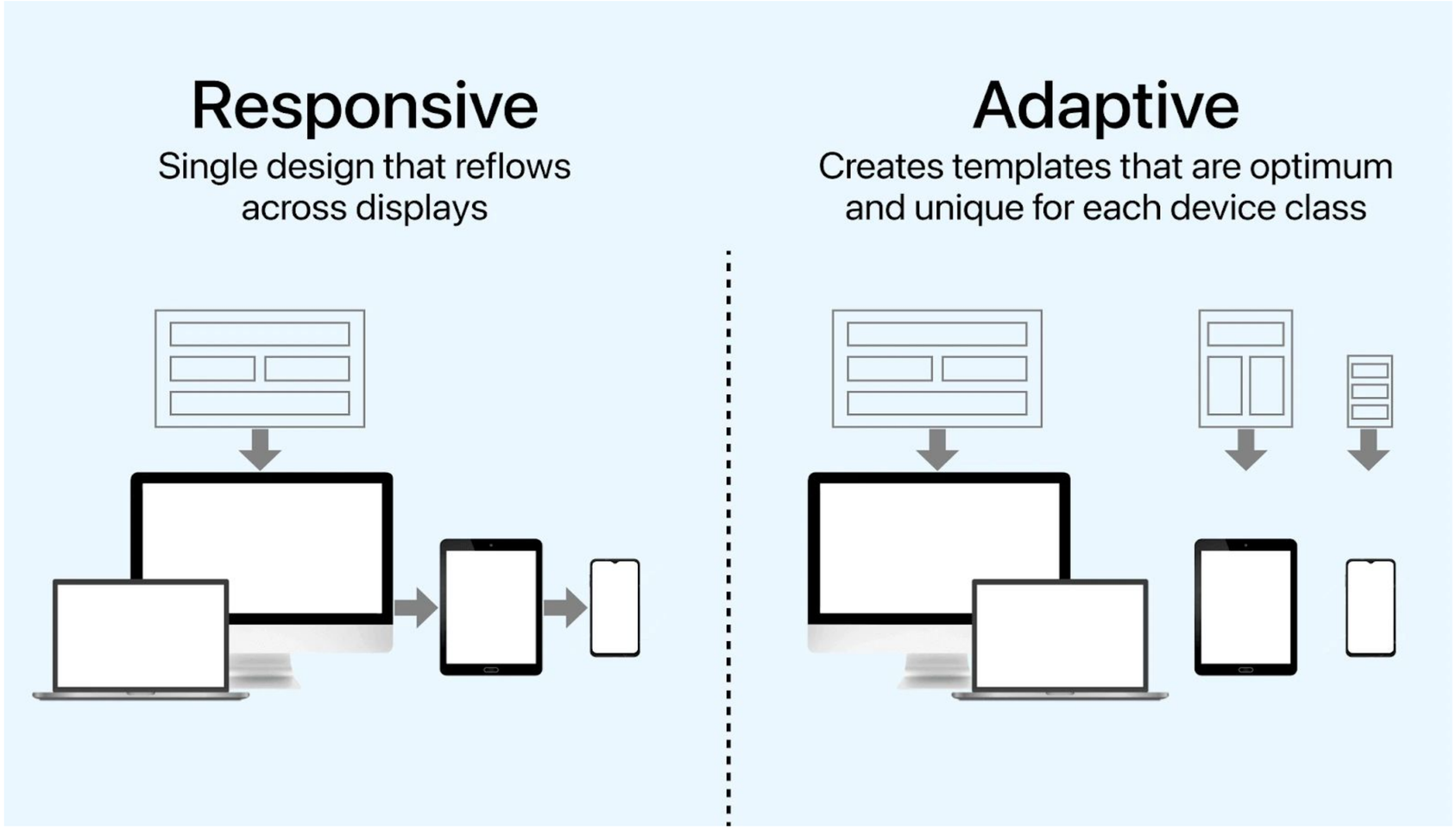
Студенческая
ИТ-лаборатория

effectiveband

Принципы вёрстки (Adaptive/Responsive)

Принципы вёрстки

effectiveband



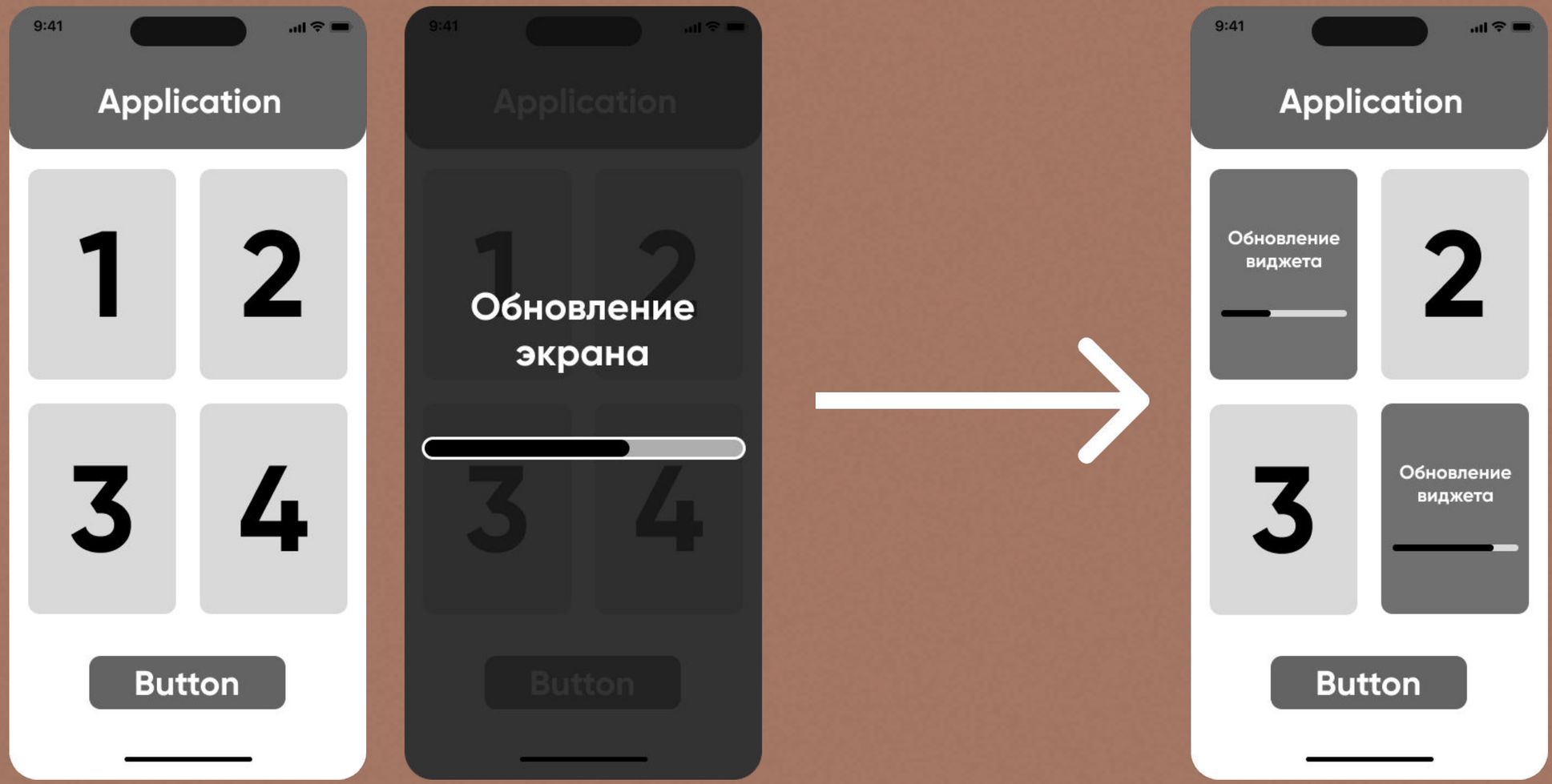
Студенческая
ИТ-лаборатория

effectiveband

Реактивность и стейт-менеджмент

Что такое реактивность?

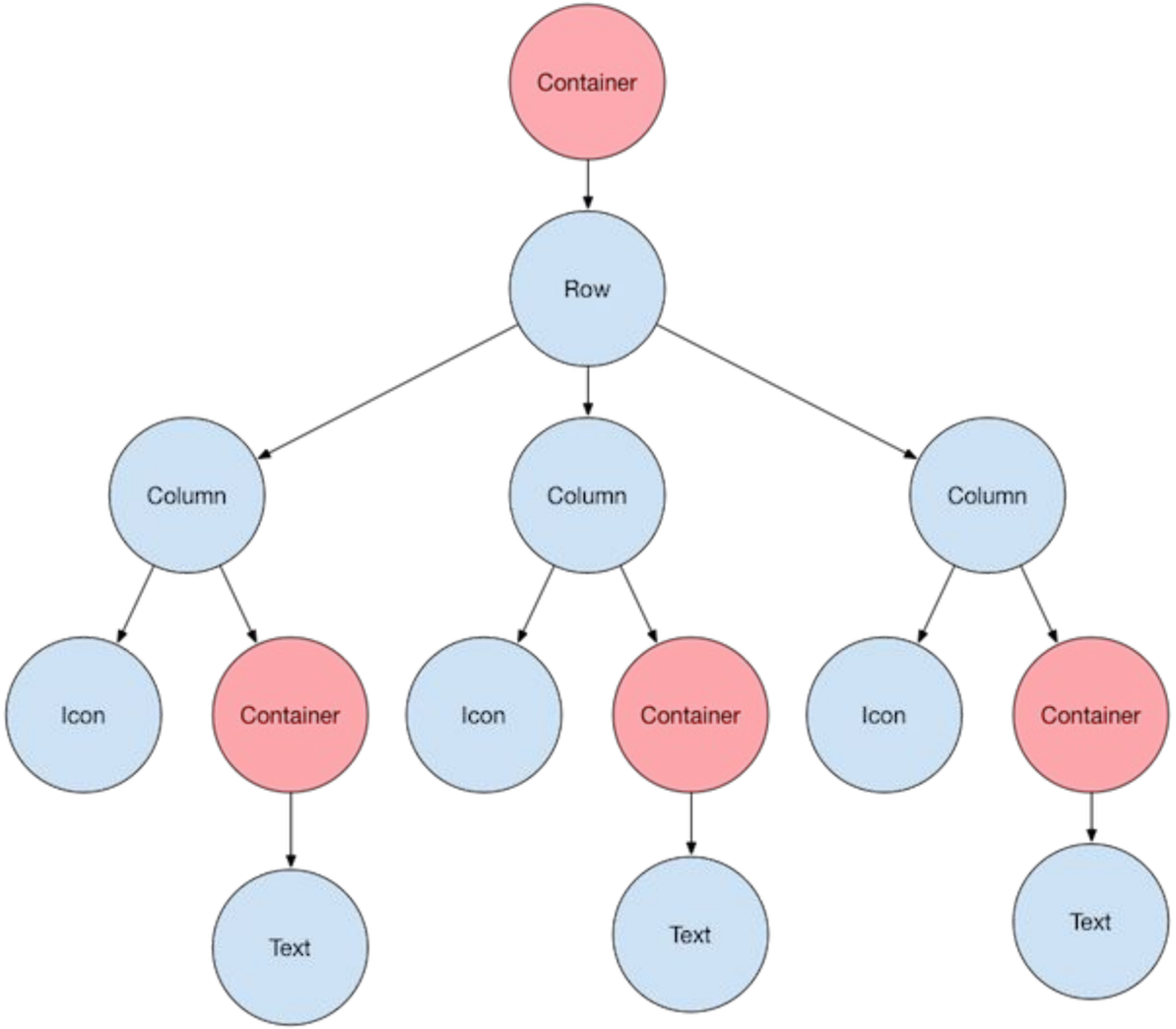
effectiveband

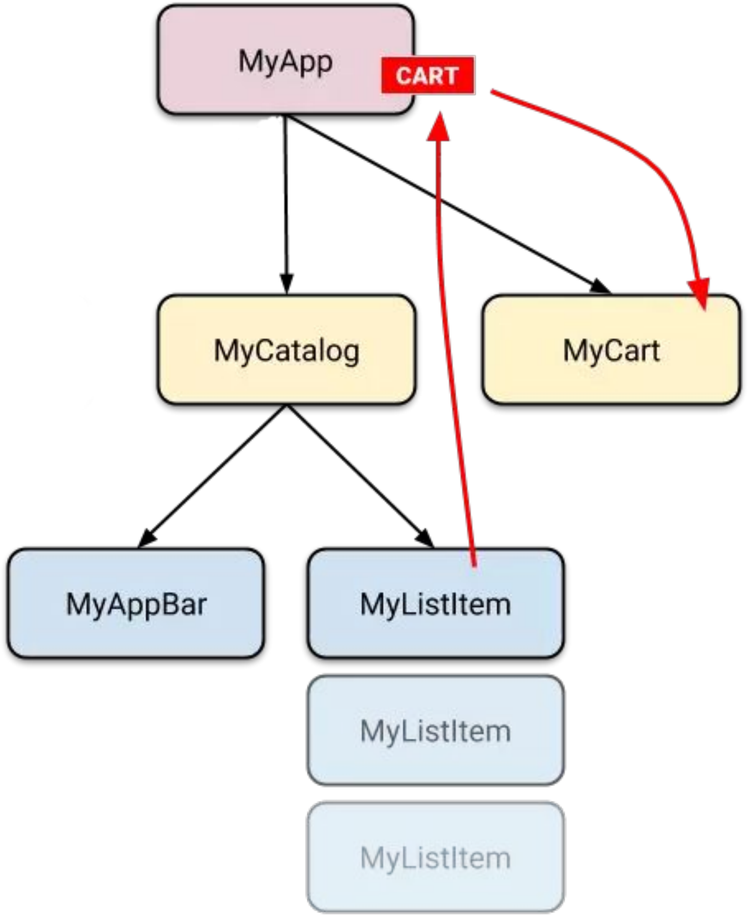


Студенческая
ИТ-лаборатория

Flutter Widget Tree

effectiveband





Стейт-менеджмент

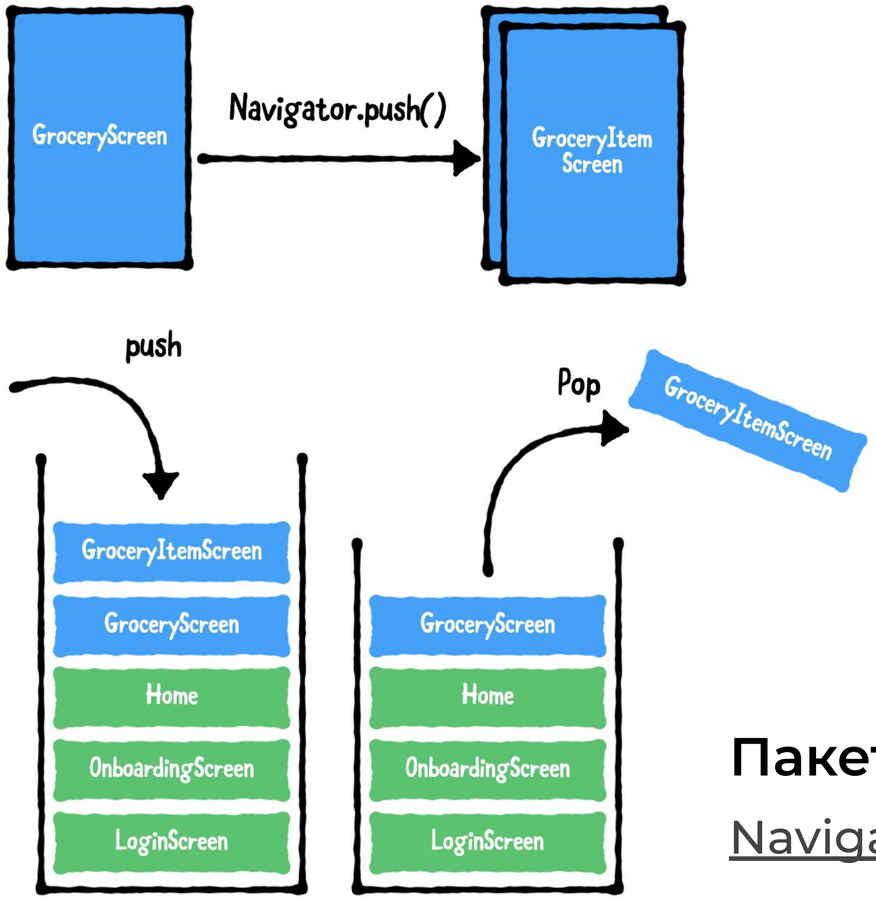
effectiveband

Студенческая
ИТ-лаборатория

effectiveband

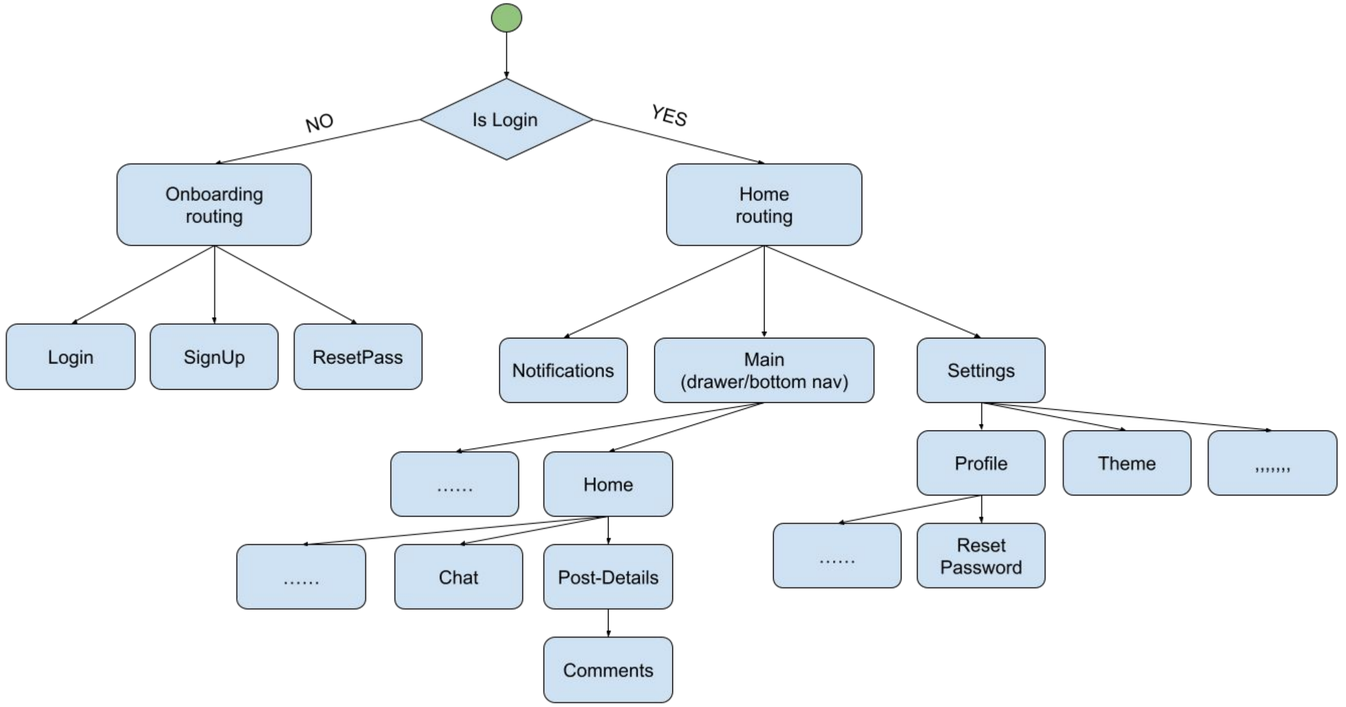
Навигация и роутинг

Навигация



Роутинг

effectiveband



Пакеты:

Navigator

Go_Router

AutoRoute

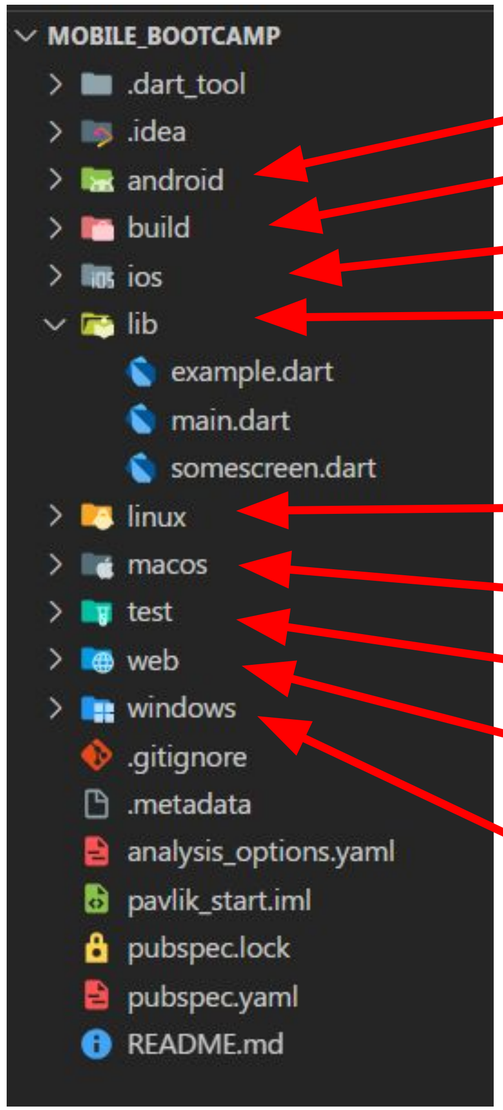
Студенческая
ИТ-лаборатория

effectiveband

Файловая структура проекта

Ваш проект:

effectiveband



Нативные модули (для Android)

Сборки (под системы)

Нативные модули (для iOS)

Ваши файлы (на Dart)

Нативные модули (для Linux)

Нативные модули (для MacOS)

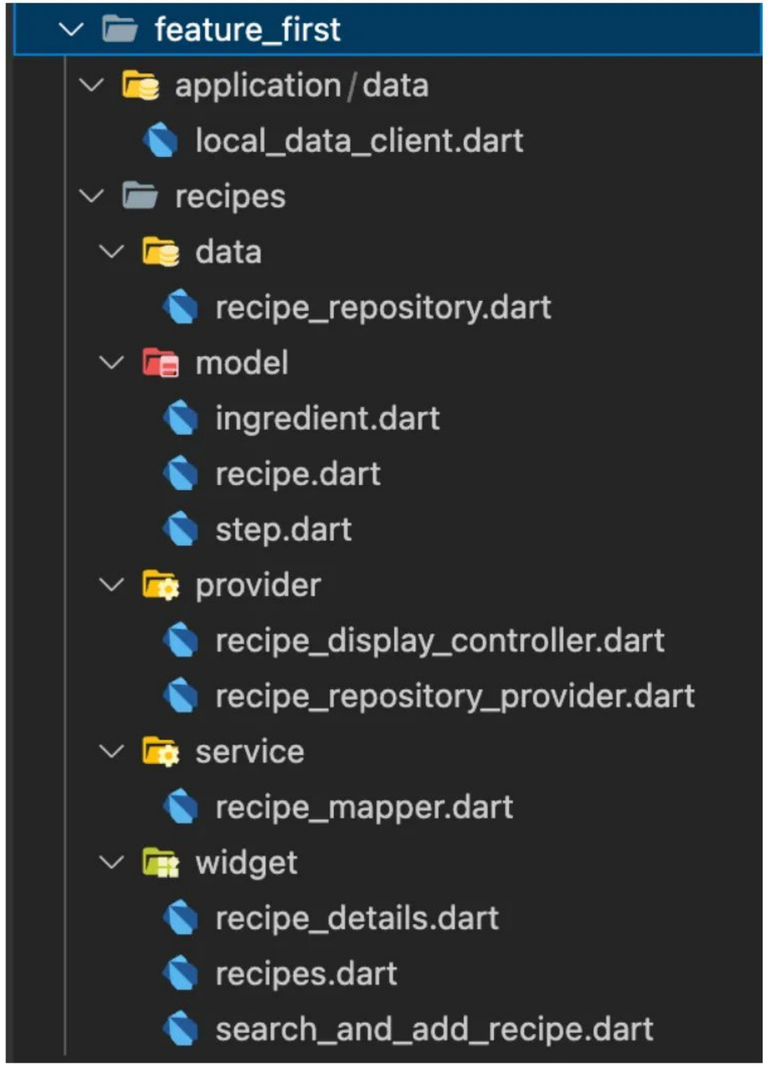
Папка для тестов

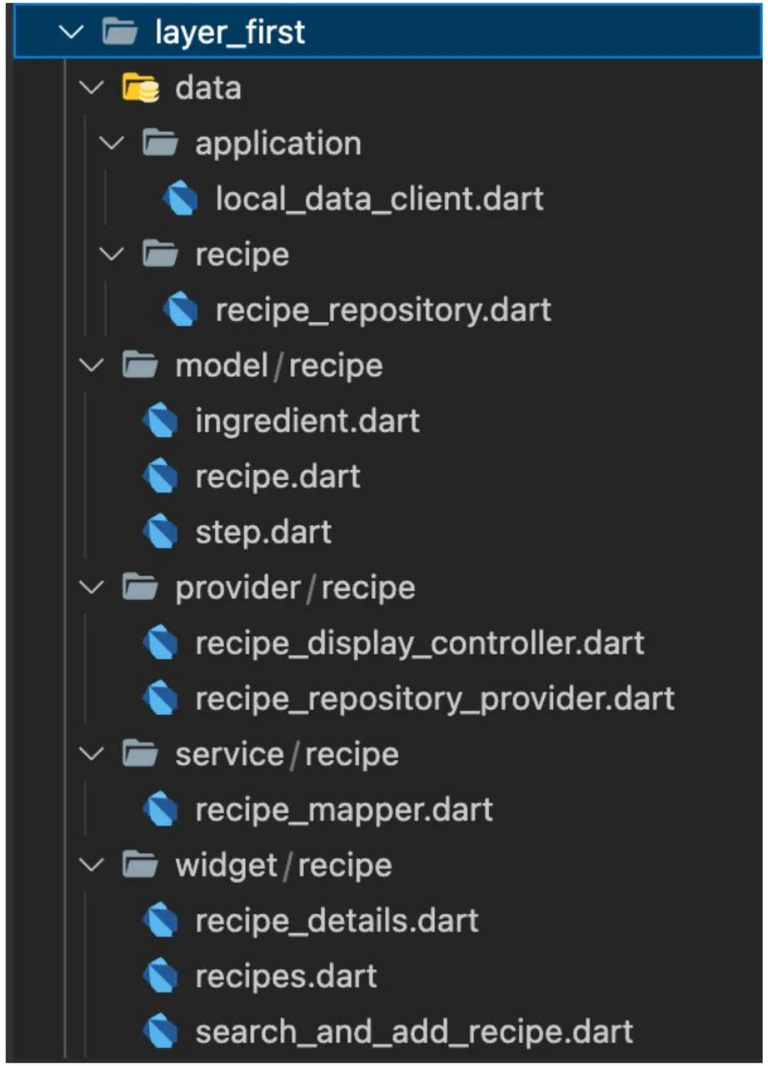
Нативные модули (для ВЕБа)

Нативные модули (для Windows)

Подходы к организации файлов

effectiveband





Студенческая
ИТ-лаборатория

effectiveband

DartSDK, FlutterSDK: консольные команды

Основные команды

DartSDK

```
dart --version      # Проверка установленной версии Dart
dart create app_name  # Создание нового Dart-проекта
dart run main.dart  # Запуск Dart-программы
dart compile exe main.dart  # Компиляция в исполняемый файл
dart analyze        # Анализ кода на ошибки
dart format .       # Форматирование кода
dart test           # Запуск тестов проекта
dart pub get        # Установка зависимостей
dart pub upgrade    # Обновление зависимостей
```

Подробная документация

Основные команды

FlutterSDK

```
flutter doctor      # Проверка установки Flutter и зависимостей
flutter create app_name  # Создание нового Flutter-проекта
flutter run         # Запуск приложения на подключенном устройстве или эмуляторе
flutter build apk   # Сборка APK-файла для Android
flutter build ios   # Сборка iOS-приложения (только на macOS)
flutter pub get     # Установка зависимостей из pubspec.yaml
flutter pub upgrade  # Обновление зависимостей
flutter clean       # Очистка кэша сборки (полезно при ошибках)
flutter analyze     # Анализ кода на наличие ошибок и предупреждений
flutter test        # Запуск тестов проекта
flutter format .    # Форматирование кода во всем проекте
```

Подробная документация

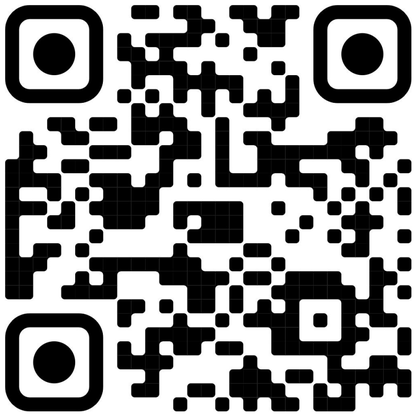
Студенческая
ИТ-лаборатория

effectiveband

Немного полезностей

Полезные материалы

effectiveband



Документация
Dart

Документация
Flutter

(Книга)
Основы Dart

(YouTube)
LazyLoad

Если остались вопросы:

Мой Telegram:

@d11m40

Дмитрий Дружин
Product MAnager в Effective