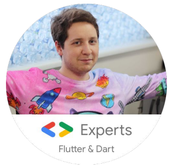
Google Developers

Make the code work for you: Flutter Code Generation

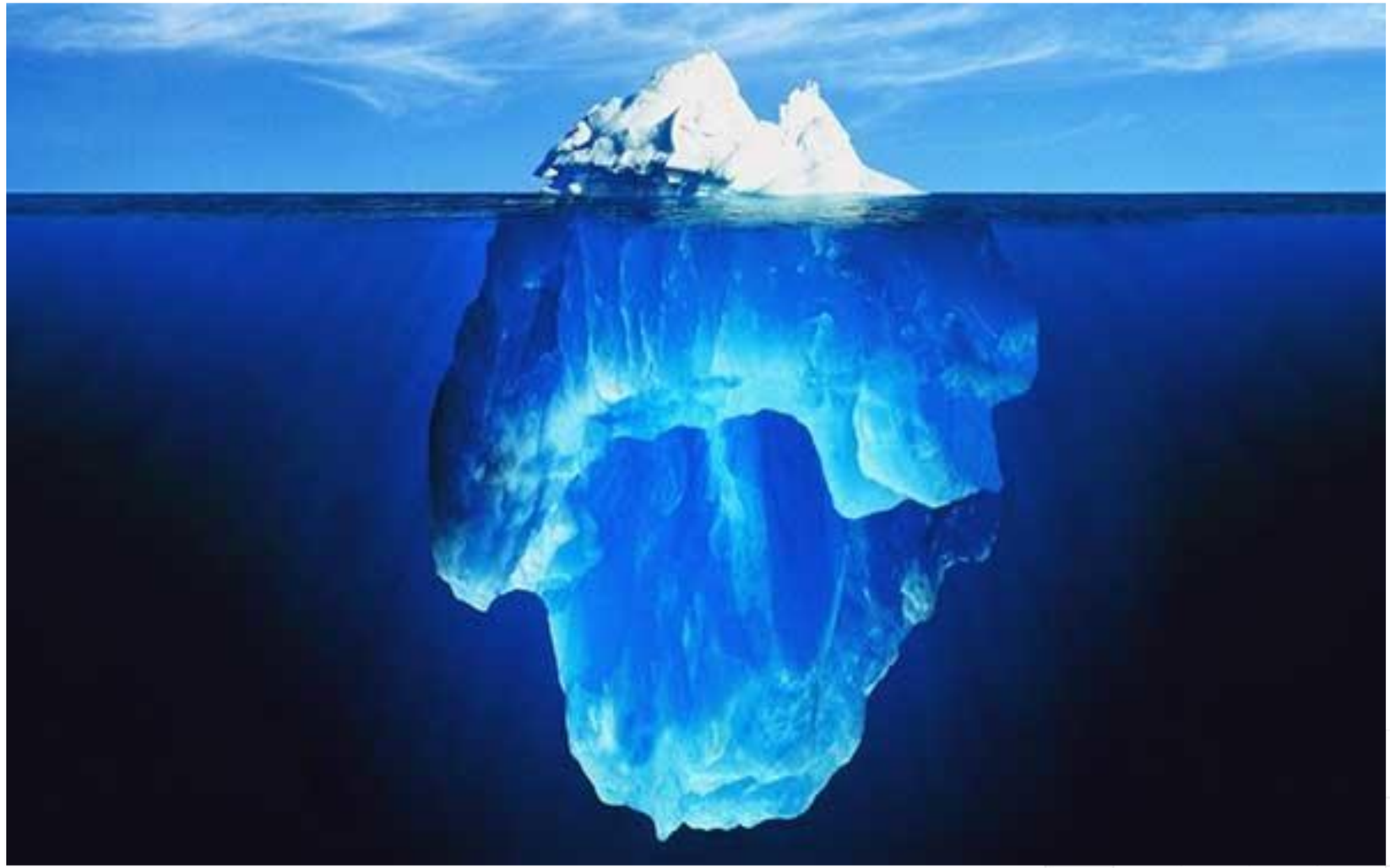


Mangirdas Kazlauskas
GDE for Flutter & Dart
@mkobuolys

Expectations

This talk

Flutter Code Generation



Agenda

- Localization
- Assets
- Objects & State
- Widgets

About me

- Software Engineer from Lithuania 🇱🇹
- Mobile Tech Lead @ Billo
- Google Developer Expert for Flutter & Dart 💙
- Organiser @ Flutter Lithuania
- Using Flutter since v0.10.2

billo

What is Code Generation

Basic development flow

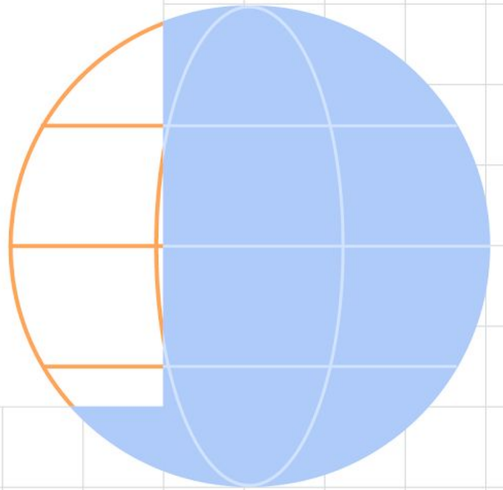
What is Code Generation

Basic development flow

Development flow with code generation

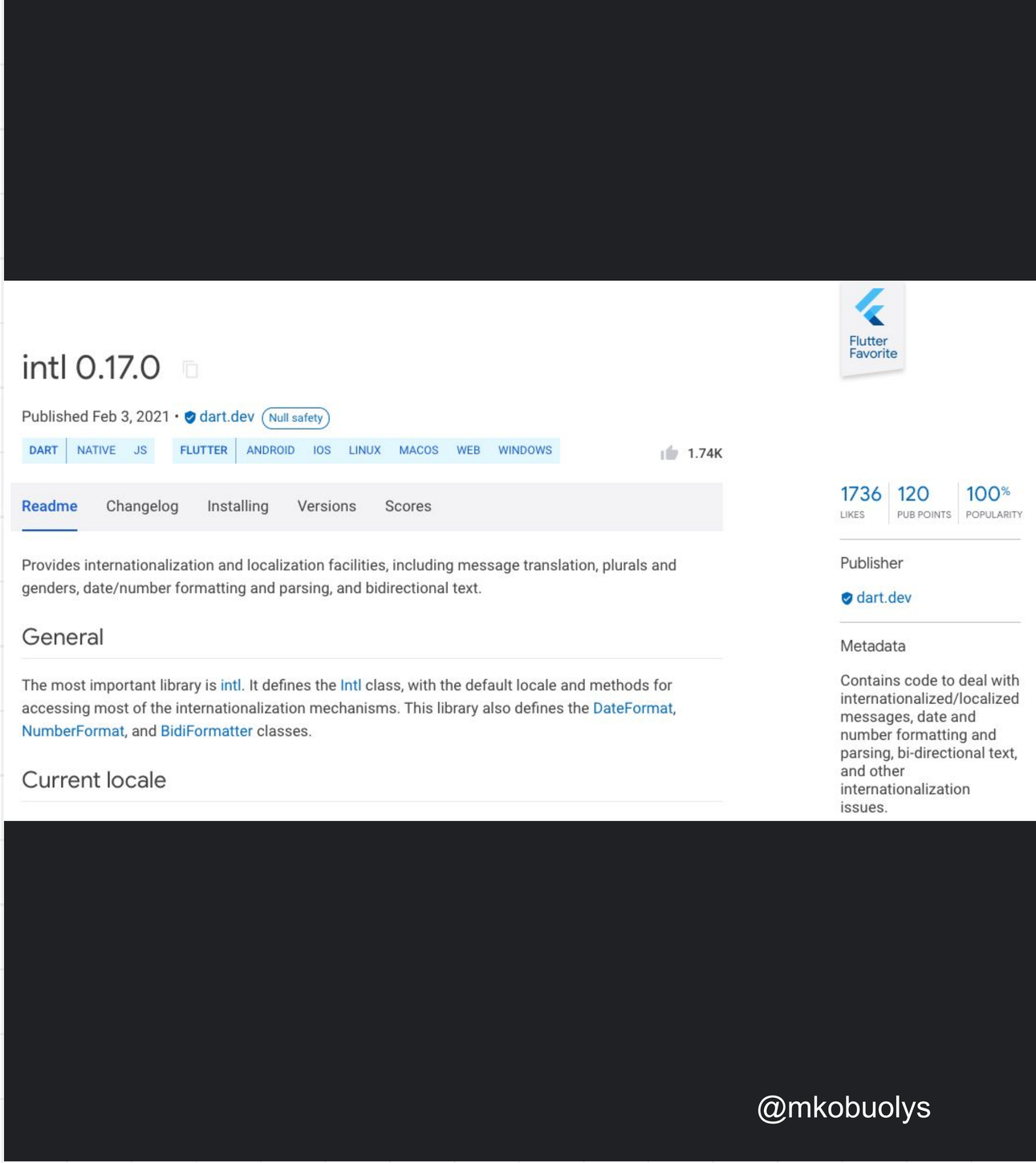
Experts

Localization



intl

Provides internationalization and localization facilities, including message translation, plurals and genders, date/number formatting and parsing, and bidirectional text.



Experts

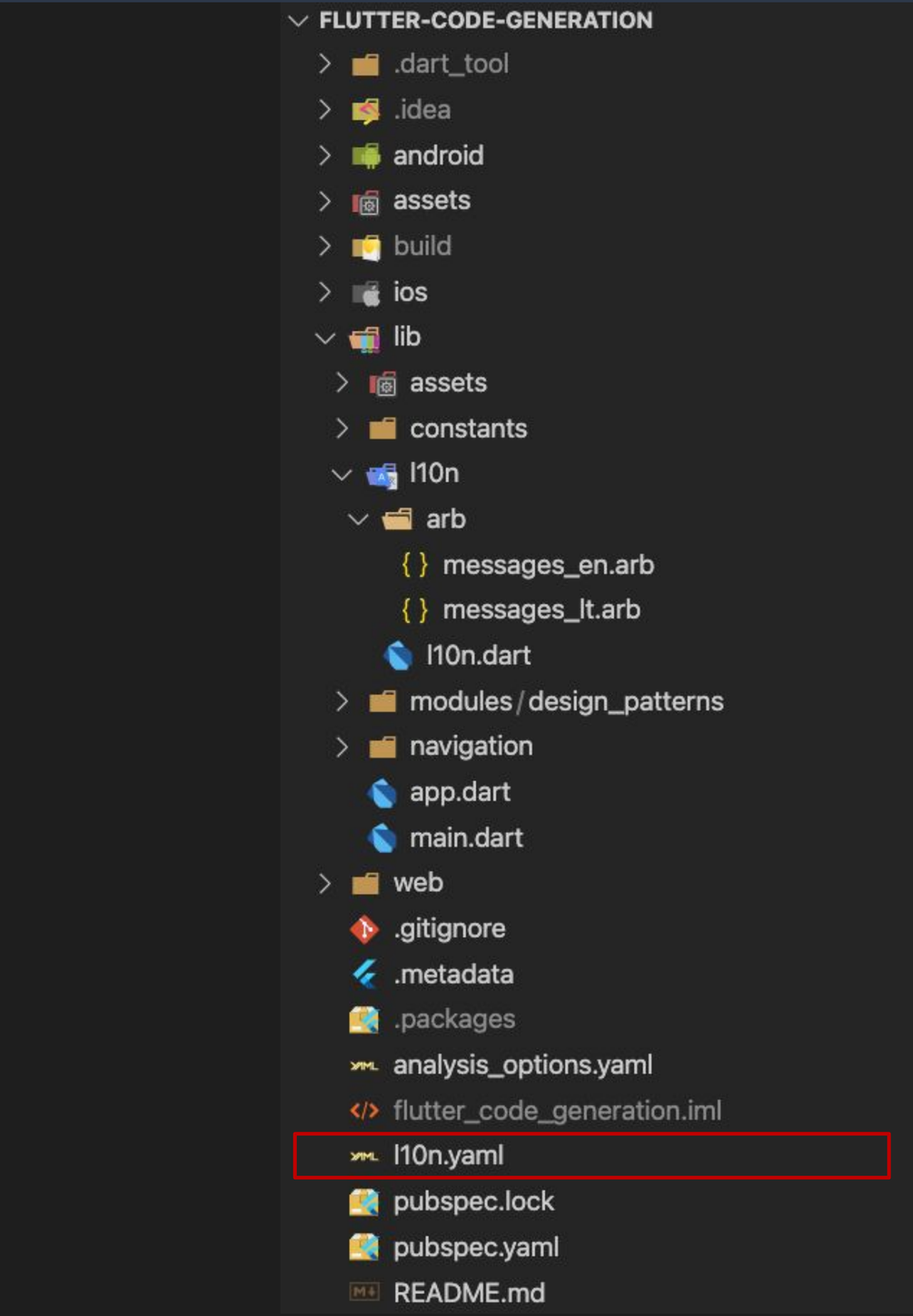
@mkobuolys

pubspec.yaml

```yaml
name: flutter_code_generation
description: Examples of Flutter Code Generation.
version: 1.0.0+1

environment:
  sdk: ">=2.12.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter
  # Internationalization support.
  flutter_localizations:
    sdk: flutter
  intl: ^0.17.0

flutter:
  # Adds code generation (synthetic package) support
  generate: true
```

l10n.yaml

```
1  arb-dir: lib/l10n/arb
2  template-arb-file: messages_en.arb
3  output-localization-file: app_localizations.dart
```

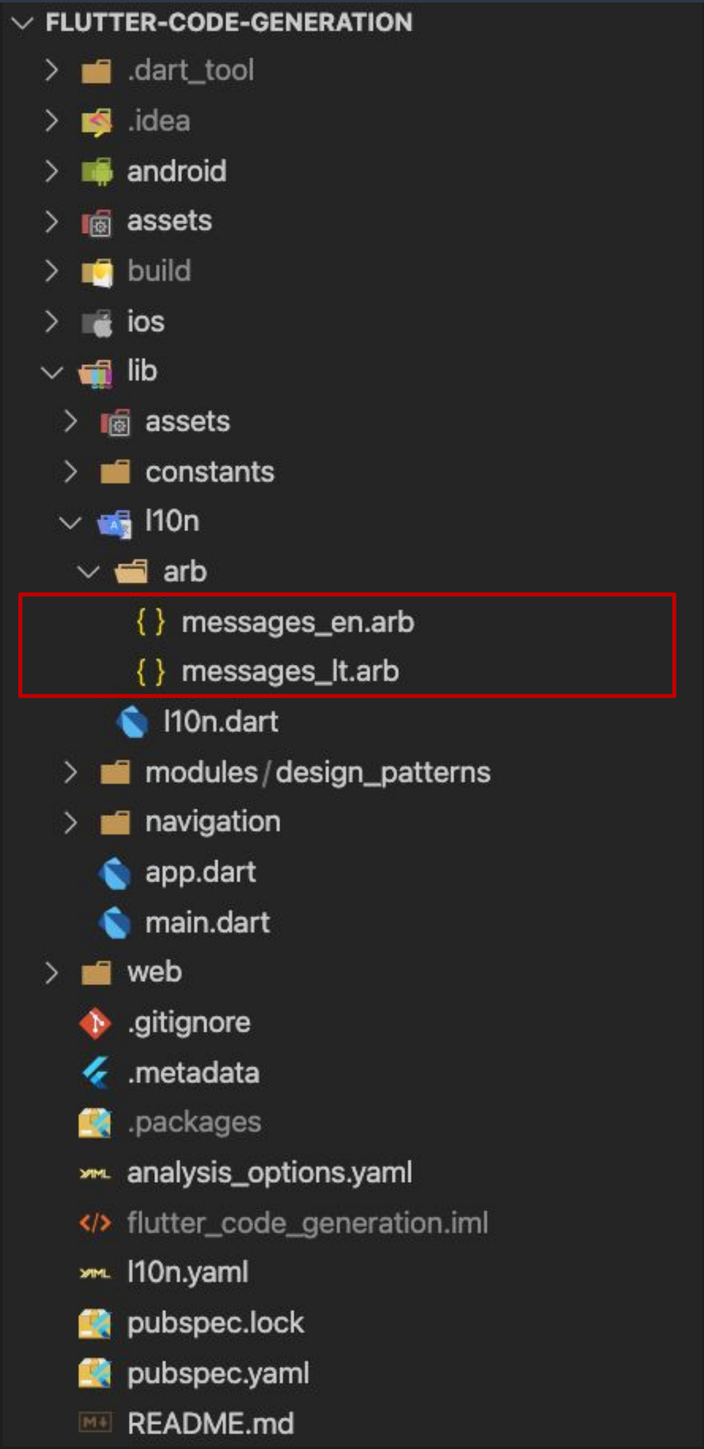


ARB (.arb) files

```arb
{
  "@@locale": "en",
  "appTitle": "Flutter Code Generation",
  "@appTitle": {
    "description": "Main menu page title."
  },
  "localizationExamplesTitle": "Localization Examples",
  "@localizationExamplesTitle": {
    "description": "Localization examples page title."
  },
  ...
}
```

```arb
{
  "@@locale": "lt",
  "appTitle": "Flutter Code Generation",
  "localizationExamplesTitle": "Lokalizacijos pavyzdžiai",
  "localizationExamplesLocaleText": "Lokalė: {locale}",
  "localizationExamplesCurrentDateText": "Šiandienos data: {date}",
  "localizationExamplesCurrencyText": "Kaina: {price}",
  "localizationExamplesButtonPressedText": "{count, plural, =0 {Jūs dar nepaspaudėte mygtuko}
=1 {Jūs paspaudėte mygtuką vieną kartą} other {Jūs paspaudėte mygtuką tiek kartų: {count}}}",
  ...
}
```

@mkobuolys



l10n.dart (optional)

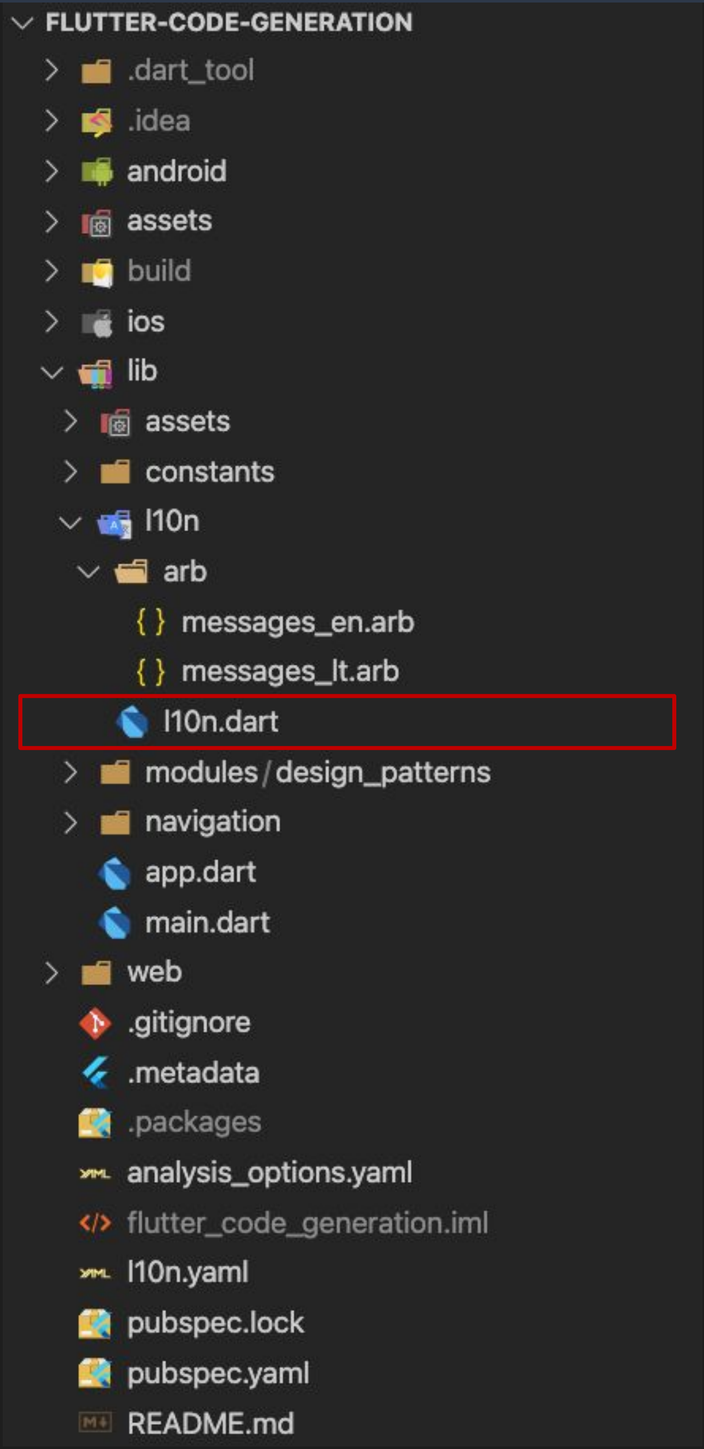
```dart
import 'package:flutter/widgets.dart';

import 'package:flutter_gen/gen_l10n/app_localizations.dart';
export 'package:flutter_gen/gen_l10n/app_localizations.dart';

extension AppLocalizationsExtension on BuildContext {
  AppLocalizations get l10n => AppLocalizations.of(this)!;
}
```

```dart
AppBar(
  title: Text(context.l10n.appTitle),
),
```

- w/ extension

```dart
AppBar(
  title: Text(AppLocalizations.of(context)!.appTitle),
),
```

- w/o extension



ARB -> Dart: Simple parameter

```
"localizationExamplesTitle": "Localization Examples",
"@localizationExamplesTitle": {
  "description": "Localization examples page title."
},
"localizationExamplesLocaleText": "Locale: {locale}",
"@localizationExamplesLocaleText": {
  "description": "Localization examples locale text.",
  "placeholders": {
    "locale": {
      "type": "String",
      "example": "en"
    }
  }
},
```

```
/// The translations for English (`en`).
class AppLocalizationsEn extends AppLocalizations {
  AppLocalizationsEn([String locale = 'en']) : super(locale);

  @override
  String get localizationExamplesTitle => 'Localization Examples';

  @override
  String localizationExamplesLocaleText(String locale) {
    return 'Locale: $locale';
  }
}
```

ARB -> Dart: Date, currency

```
"localizationExamplesCurrentDateText": "Today's date: {date}",
"@localizationExamplesCurrentDateText": {
  "description": "Localization examples current date and time text.",
  "placeholders": {
    "date": {
      "type": "DateTime",
      "format": "yMMMMd"
    }
  }
},
"localizationExamplesCurrencyText": "Price: {price}",
"@localizationExamplesCurrencyText": {
  "description": "Localization examples current date and time text.",
  "placeholders": {
    "price": {
      "type": "double",
      "format": "simpleCurrency",
      "optionalParameters": {
        "decimalDigits": 2
      }
    }
  }
},
```

```
@override
String localizationExamplesCurrentDateText(DateTime date) {
  final intl.DateFormat dateDateFormat = intl.DateFormat.yMMMMd(localeName);
  final String dateString = dateDateFormat.format(date);

  return 'Today\'s date: $dateString';
}

@override
String localizationExamplesCurrencyText(double price) {
  final intl.NumberFormat priceNumberFormat = intl.NumberFormat.simpleCurrency(
    locale: localeName,
    decimalDigits: 2
  );
  final String priceString = priceNumberFormat.format(price);

  return 'Price: $priceString';
}
```

@mkobuolys

ARB -> Dart: Plurals

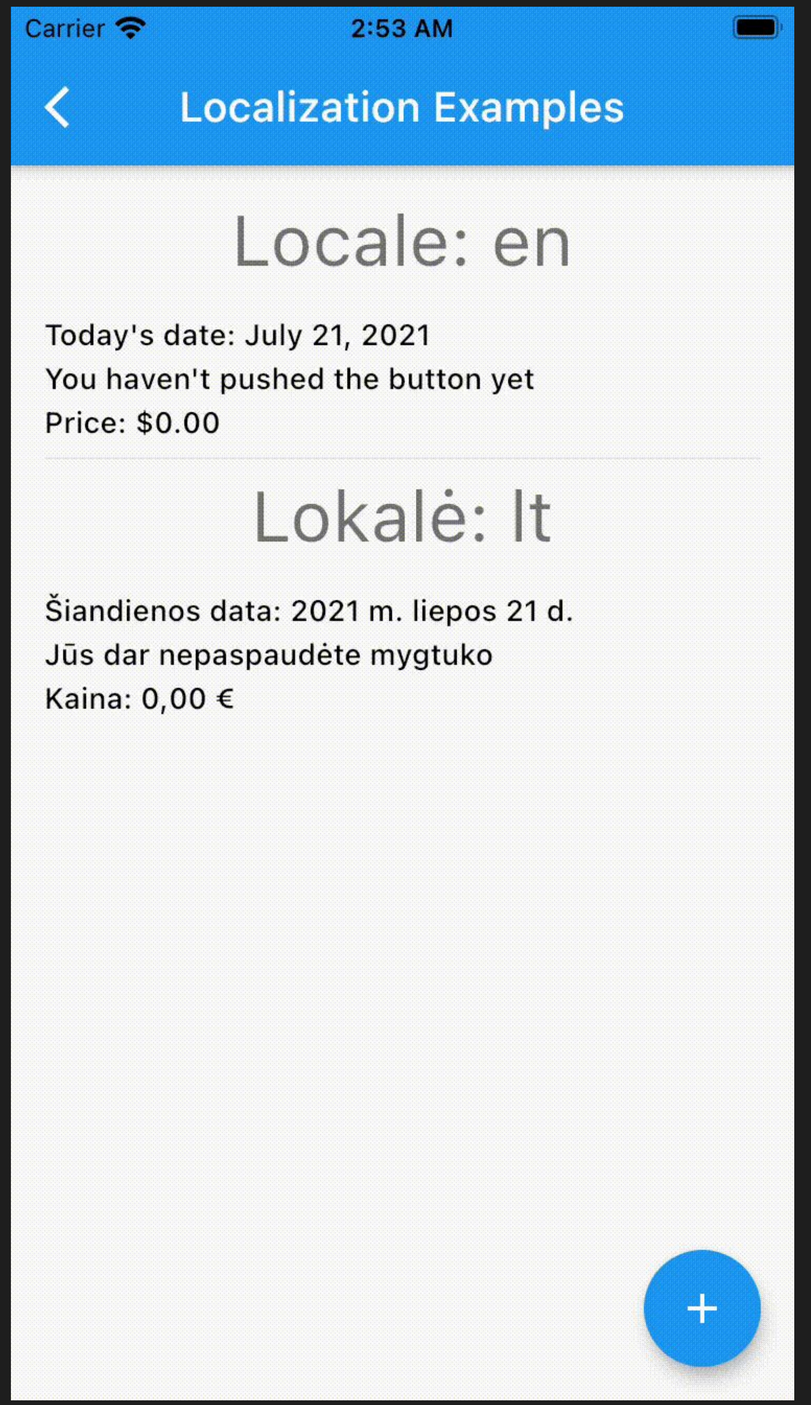
```
"localizationExamplesButtonPressedText": "{count, plural, =0 {You haven't pushed
the button yet} =1 {You have pushed the button once} other {You have pushed the
 button this many times: {count}}}",
"@localizationExamplesButtonPressedText": {
  "description": "Localization examples button pressed example text.",
  "placeholders": {
    "count": {
      "type": "int",
      "format": "decimalPattern",
      "example": "10"
    }
  }
},
```

```
@override
String localizationExamplesButtonPressedText(int count) {
  final intl.NumberFormat countNumberFormat = intl.NumberFormat.decimalPattern(localeName);
  final String countString = countNumberFormat.format(count);

  return intl.Intl.pluralLogic(
    count,
    locale: localeName,
    zero: 'You haven\'t pushed the button yet',
    one: 'You have pushed the button once',
    other: 'You have pushed the button this many times: $count',
  );
}
```

@mkobuolys

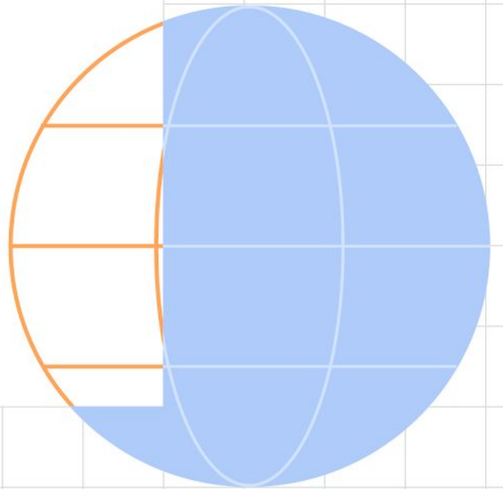
Usage

```dart
Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: [
    Center(
      child: Text(
        context.l10n.localizationExamplesLocaleText(languageCode),
        style: Theme.of(context).textTheme.headline4,
      ),
    ),
    const SizedBox(height: LayoutConstants.spaceL),
    Text(
      context.l10n.localizationExamplesCurrentDateText(DateTime.now()),
      style: textStyle,
    ),
    const SizedBox(height: LayoutConstants.spaceS),
    Text(
      context.l10n.localizationExamplesButtonPressedText(counter),
      style: textStyle,
    ),
    const SizedBox(height: LayoutConstants.spaceS),
    Text(
      context.l10n.localizationExamplesCurrencyText(counter * 345.67),
      style: textStyle,
    ),
  ],
);
```



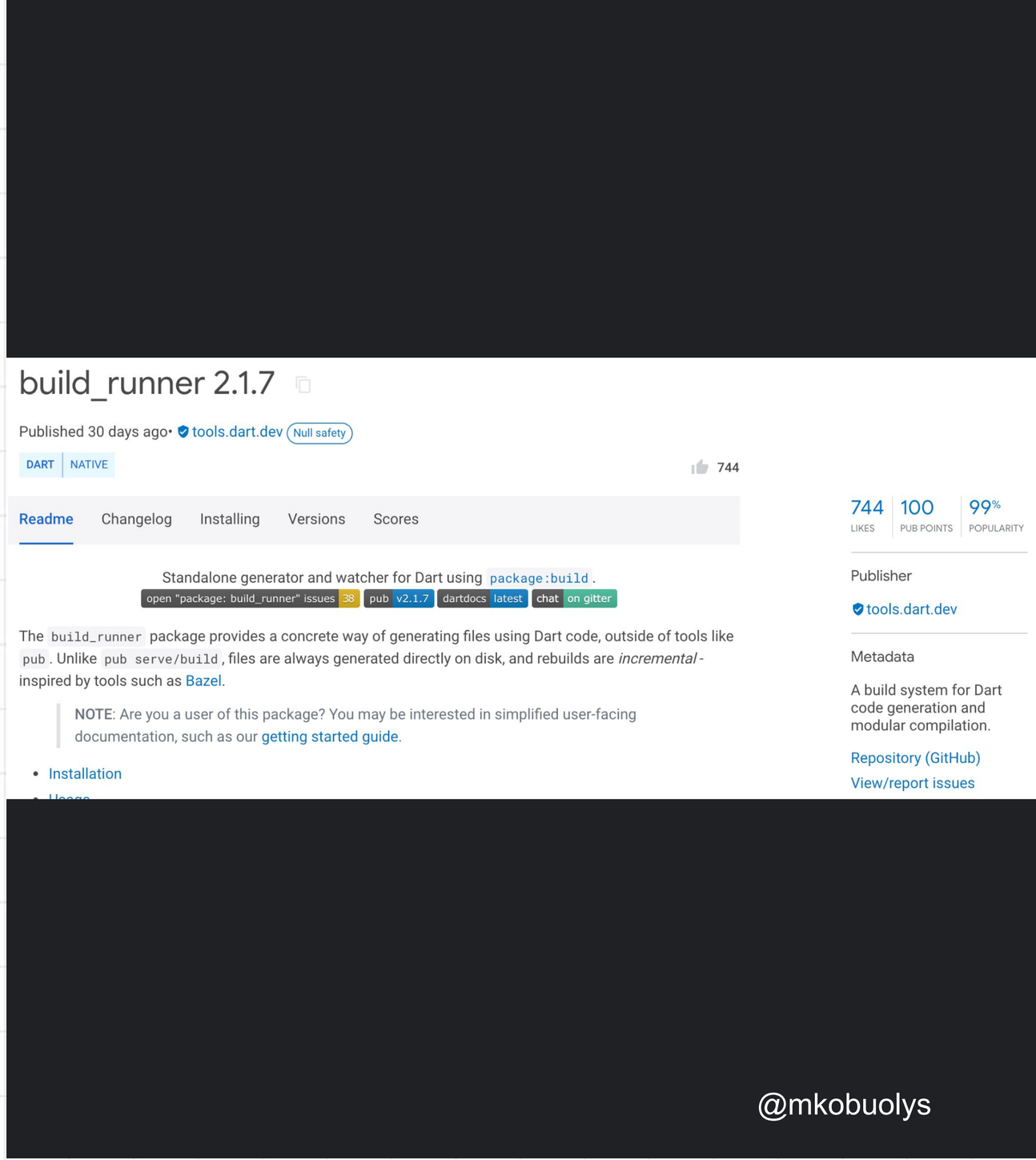
Experts

Assets



build_runner

Provides a concrete way of generating files using Dart code, outside of tools like pub.

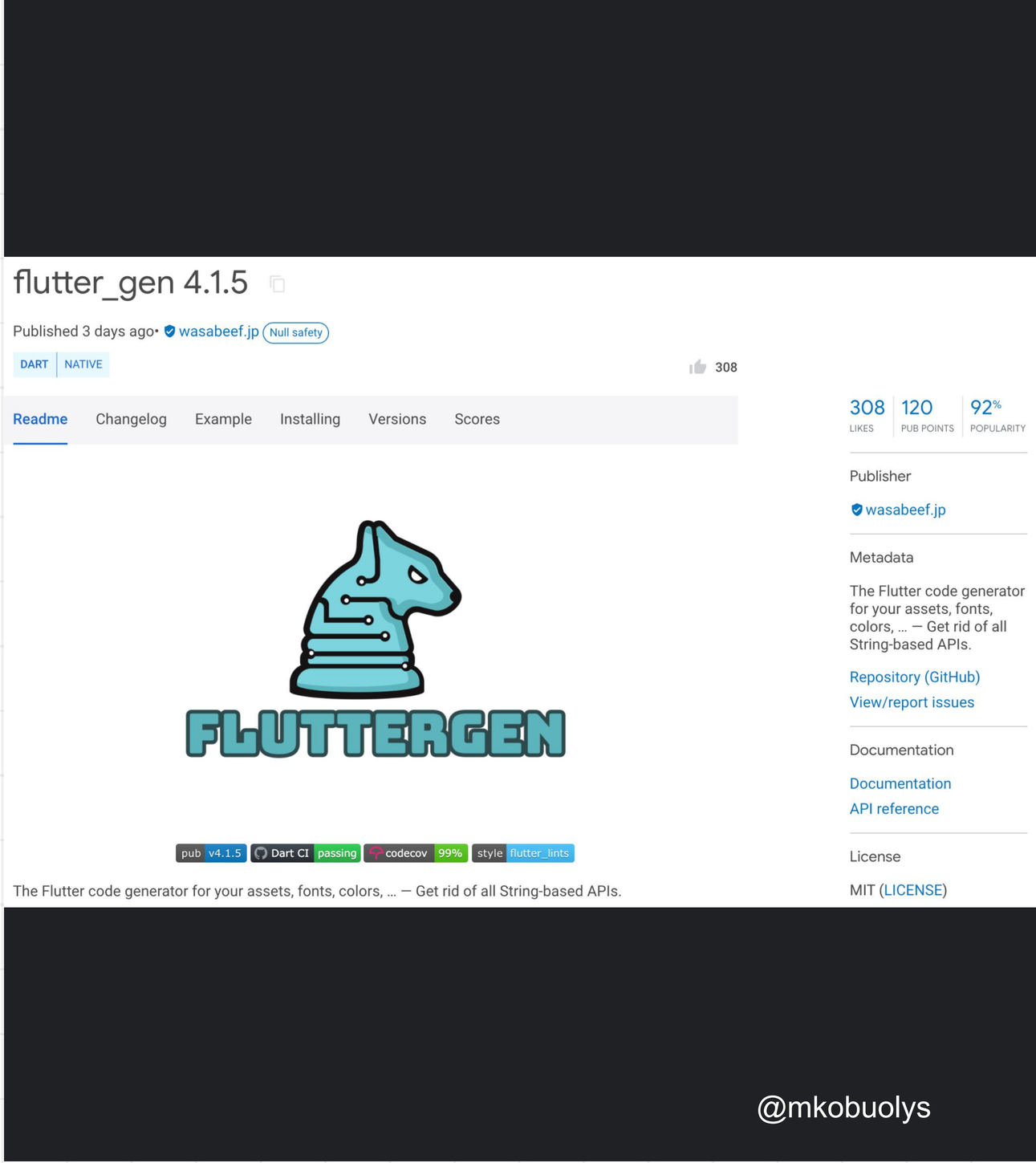


Experts

@mkobuolys

flutter_gen

The Flutter code generator for your assets, fonts, colors, … - Get rid of all String-based APIs.



Experts

@mkobuolys

pubspec.yaml – dependencies/assets

```yaml
dependencies:
  flutter:
    sdk: flutter
  flare_flutter: ^3.0.2
  flutter_svg: ^1.0.1
  rive: ^0.8.1

dev_dependencies:
  build_runner: ^2.1.7
  flutter_gen_runner: ^4.1.5
```

```yaml
flutter:
  uses-material-design: true
  # Adds code generation (synthetic package) support
  generate: true
  assets:
    - assets/color/
    - assets/flare/
    - assets/images/
    - assets/images/icons/
    - assets/json/
    - assets/rive/
  fonts:
    - family: Fuggles
      fonts:
        - asset: assets/fonts/Fuggles-Regular.ttf
```

pubspec.yaml – flutter_gen

```yaml
flutter_gen:
  output: lib/assets/gen # Optional (default: lib/gen/)
  line_length: 80 # Optional (default: 80)
  null_safety: true # Optional (default: true)

  integrations:
    flutter_svg: true
    flare_flutter: true
    rive: true

  colors:
    enabled: true
    inputs:
      - assets/color/colors.xml

  assets:
    enabled: true
    # Assets.imagesDash
    # style: camel-case

    # Assets.images_dash
    # style: snake-case

    # Assets.images.dash (default style)
    # style: dot-delimiter

  fonts:
    enabled: true
```

- w/o integration

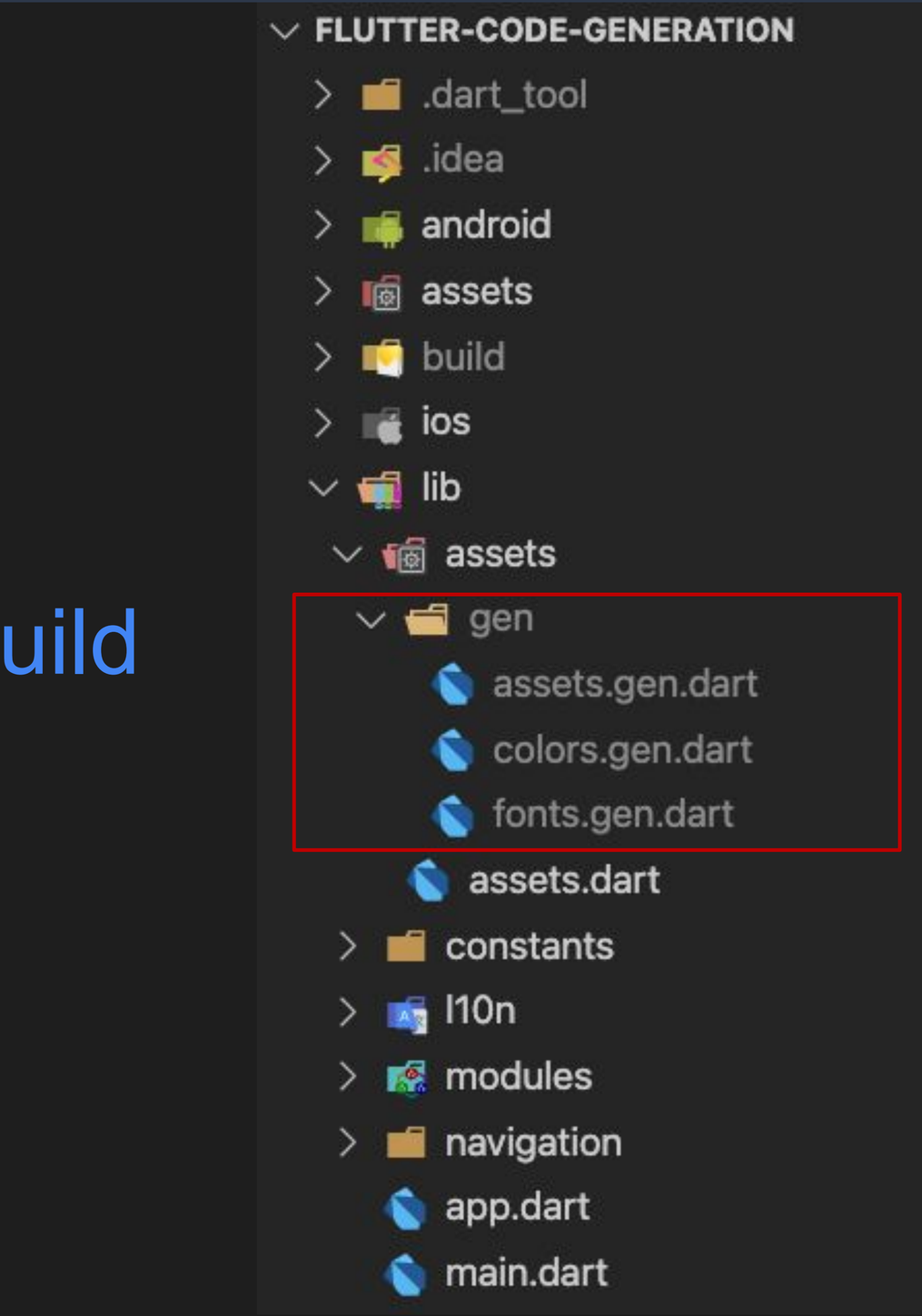
```dart
class $AssetsImagesIconsGen {
  const $AssetsImagesIconsGen();

  String get githubLogo => 'assets/images/icons/github_logo.svg';
  String get mediumLogo => 'assets/images/icons/medium_logo.svg';
  String get twitterLogo => 'assets/images/icons/twitter_logo.svg';
}
```

- w/ integration

```dart
class $AssetsImagesIconsGen {
  const $AssetsImagesIconsGen();

  SvgGenImage get githubLogo =>
      const SvgGenImage('assets/images/icons/github_logo.svg');
  SvgGenImage get mediumLogo =>
      const SvgGenImage('assets/images/icons/medium_logo.svg');
  SvgGenImage get twitterLogo =>
      const SvgGenImage('assets/images/icons/twitter_logo.svg');
}
```

Generating assets

> flutter pub run build_runner build



assets.gen.dart - Assets

```dart
class Assets {
  Assets._();

  static const $AssetsColorGen color = $AssetsColorGen();
  static const $AssetsFlareGen flare = $AssetsFlareGen();
  static const $AssetsImagesGen images = $AssetsImagesGen();
  static const $AssetsJsonGen json = $AssetsJsonGen();
  static const $AssetsRiveGen rive = $AssetsRiveGen();
}
```

assets.gen.dart - specific classes

```dart
class $AssetsColorGen {
  const $AssetsColorGen();

  /// File path: assets/color/colors.xml
  String get colors => 'assets/color/colors.xml';
}

class $AssetsFlareGen {
  const $AssetsFlareGen();

  /// File path: assets/flare/penguin.flr
  FlareGenImage get penguin => const FlareGenImage('assets/flare/penguin.flr');
}

class $AssetsImagesGen {
  const $AssetsImagesGen();

  /// File path: assets/images/dash-1.png
  AssetGenImage get dash1 => const AssetGenImage('assets/images/dash-1.png');

  /// File path: assets/images/dash-2.jpeg
  AssetGenImage get dash2 => const AssetGenImage('assets/images/dash-2.jpeg');

  $AssetsImagesIconsGen get icons => const $AssetsImagesIconsGen();
}
```

```dart
class $AssetsJsonGen {
  const $AssetsJsonGen();

  /// File path: assets/json/design_patterns.json
  String get designPatterns => 'assets/json/design_patterns.json';
}

class $AssetsRiveGen {
  const $AssetsRiveGen();

  /// File path: assets/rive/vehicles.riv
  RiveGenImage get vehicles => const RiveGenImage('assets/rive/vehicles.riv');
}
```

```dart
class $AssetsImagesIconsGen {
  const $AssetsImagesIconsGen();

  /// File path: assets/images/icons/github_logo.svg
  SvgGenImage get githubLogo =>
      const SvgGenImage('assets/images/icons/github_logo.svg');

  /// File path: assets/images/icons/medium_logo.svg
  SvgGenImage get mediumLogo =>
      const SvgGenImage('assets/images/icons/medium_logo.svg');

  /// File path: assets/images/icons/twitter_logo.svg
  SvgGenImage get twitterLogo =>
      const SvgGenImage('assets/images/icons/twitter_logo.svg');
}
```

@mkobuolys

colors.gen.dart

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#000000</color>
    <color name="gray_70">#EEEEEE</color>
    <color name="gray_410">#979797</color>
    <color name="crimson_red" type="material">#CF2A2A</color>
    <color name="yellow_ocher" type="material material-accent">#DF9527</color>
</resources>
```

```dart
class ColorName {
  ColorName._();

  static const Color black = Color(0xFF000000);
  static const MaterialColor crimsonRed = MaterialColor(
    0xFFCF2A2A,
    <int, Color>{
      50: Color(0xFFF9E5E5),
      100: Color(0xFFF1BFBF),
      200: Color(0xFFE79595),
      300: Color(0xFFDD6A6A),
      400: Color(0xFFD64A4A),
      500: Color(0xFFCF2A2A),
      600: Color(0xFFCA2525),
      700: Color(0xFFC31F1F),
      800: Color(0xFFBD1919),
      900: Color(0xFFB20F0F),
    },
  );
  static const Color gray410 = Color(0xFF979797);
  static const Color gray70 = Color(0xFFEEEEEE);
  static const MaterialColor yellowOcher = MaterialColor(
    0xFFDF9527,
    <int, Color>{
      50: Color(0xFFFBF2E5),
      100: Color(0xFFF5DFBE),
      200: Color(0xFFEFCA93),
      300: Color(0xFFE9B568),
      400: Color(0xFFE4A547),
      500: Color(0xFFDF9527),
      600: Color(0xFFDB8D23),
      700: Color(0xFFD7821D),
      800: Color(0xFFD27817),
      900: Color(0xFFCA670E),
    },
  );
  static const MaterialAccentColor yellowOcherAccent = MaterialAccentColor(
    0xFFFFFBCA3,
    <int, Color>{
      100: Color(0xFFFFE8E0),
      200: Color(0xFFFFFBCA3),
      400: Color(0xFFFFA989),
      700: Color(0xFFFF9E7A),
    },
  );
}
```

@mkobuolys

fonts.gen.dart

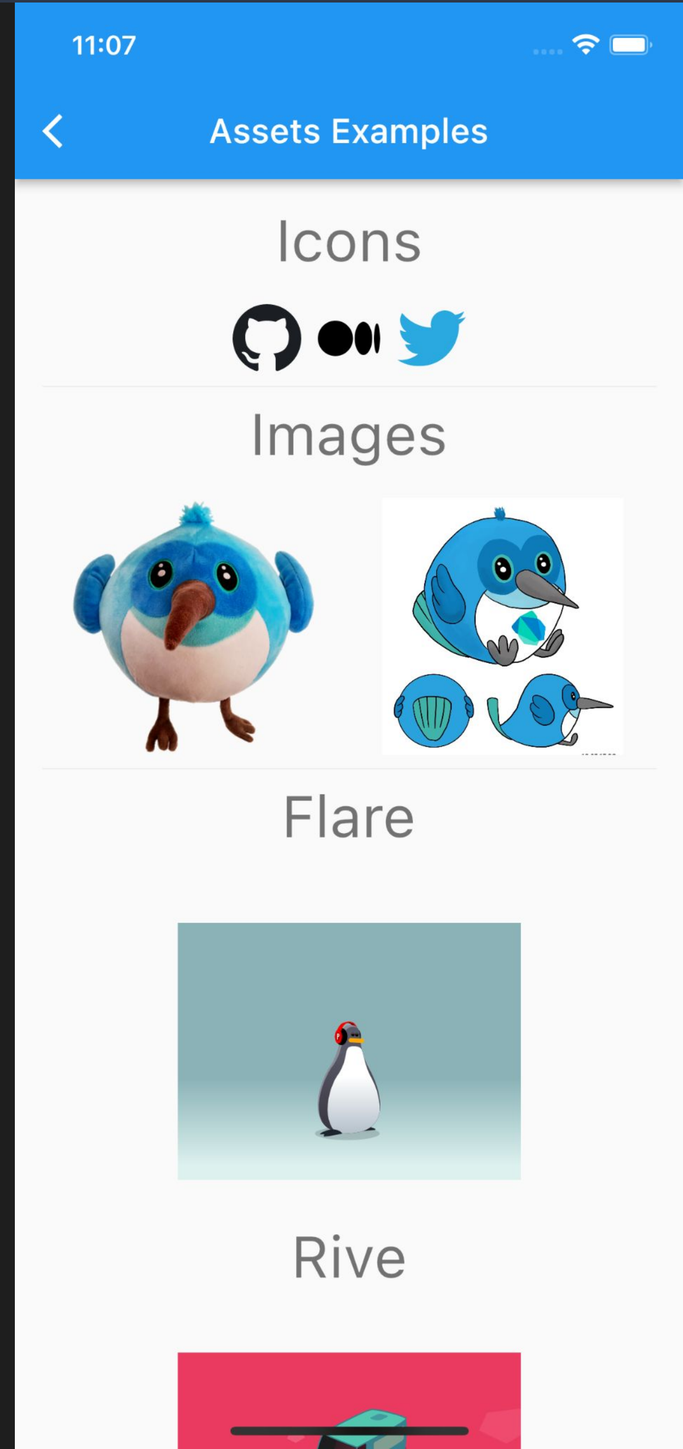
```dart
class FontFamily {
  FontFamily._();

  /// Font family: Fuggles
  static const String fuggles = 'Fuggles';
}
```

Usage: Icons/images

```dart
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Assets.images.icons.githubLogo.svg(width: _iconWidth),
    const SizedBox(width: LayoutConstants.spaceM),
    Assets.images.icons.mediumLogo.svg(width: _iconWidth),
    const SizedBox(width: LayoutConstants.spaceM),
    Assets.images.icons.twitterLogo.svg(width: _iconWidth),
  ],
);
```

```dart
Row(
  mainAxisAlignment: MainAxisAlignment.spaceAround,
  children: [
    Assets.images.dash1.image(
      height: _imageHeight,
      width: _imageWidth,
    ),
    Assets.images.dash2.image(
      height: _imageHeight,
      width: _imageWidth,
    ),
  ],
);
```

@mkobuolys



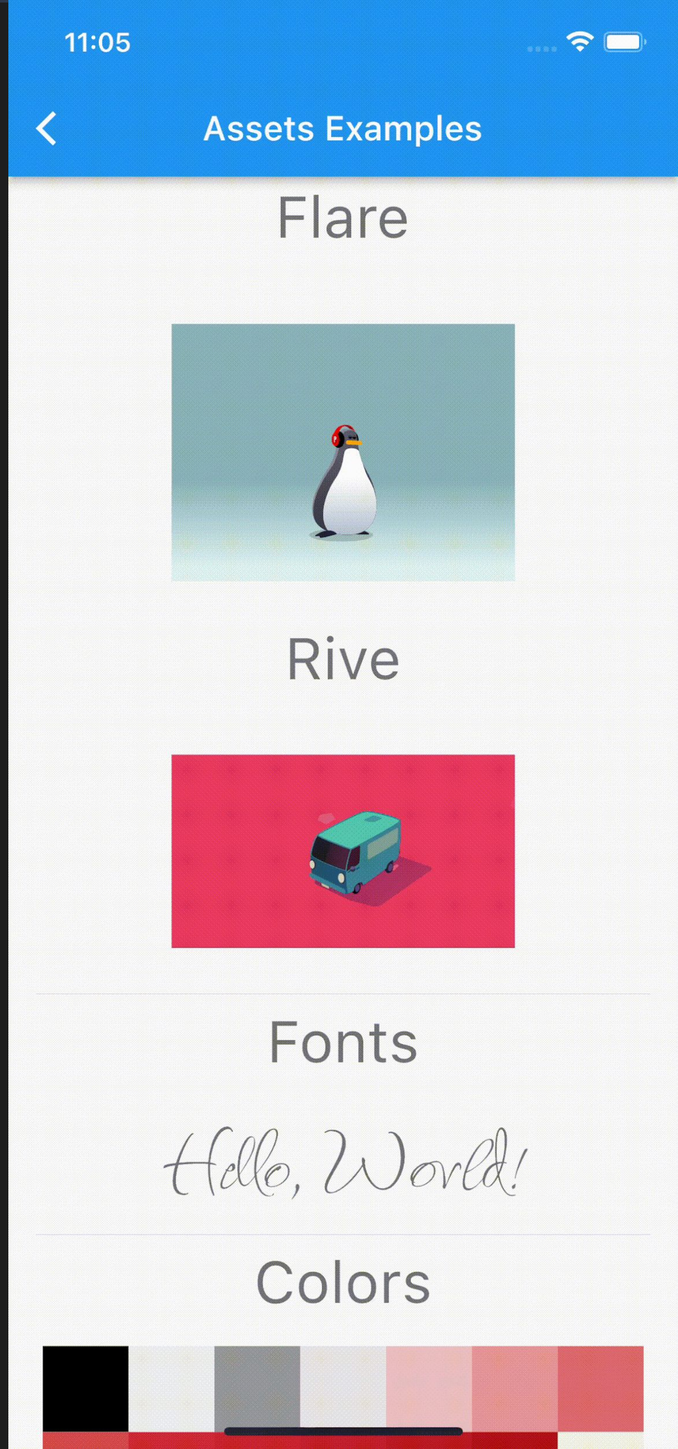
Usage: Flare/Rive/fonts

```dart
Center(
  child: SizedBox(
    height: 200.0,
    width: 200.0,
    child: Assets.flare.penguin.flare(
      animation: 'walk',
    ),
  ),
);
```

```dart
Center(
  child: SizedBox(
    height: 150.0,
    width: 200.0,
    child: Assets.rive.vehicles.rive(),
  ),
);
```

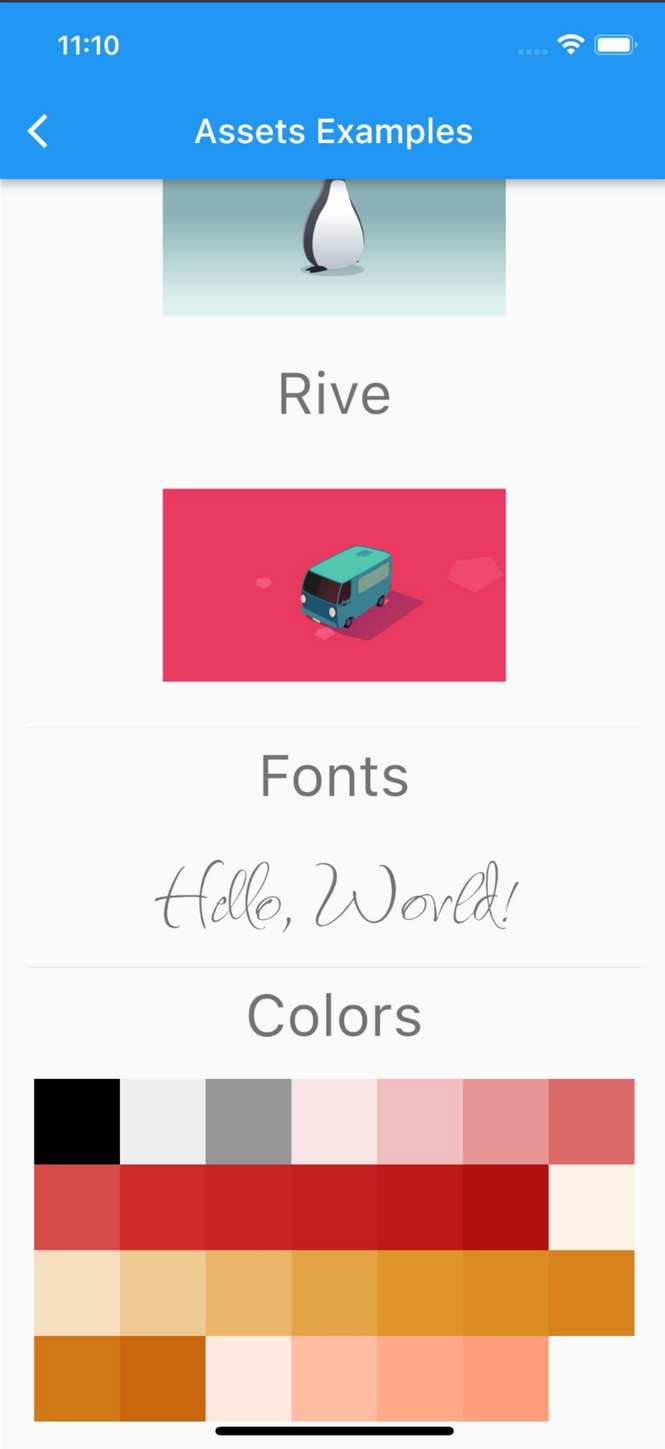
```dart
Text(
  context.l10n.assetsExamplesFontsHelloWorldText,
  style: Theme.of(context)
      .textTheme
      .headline2!
      .copyWith(fontFamily: FontFamily.fuggles),
);
```

@mkobuolys



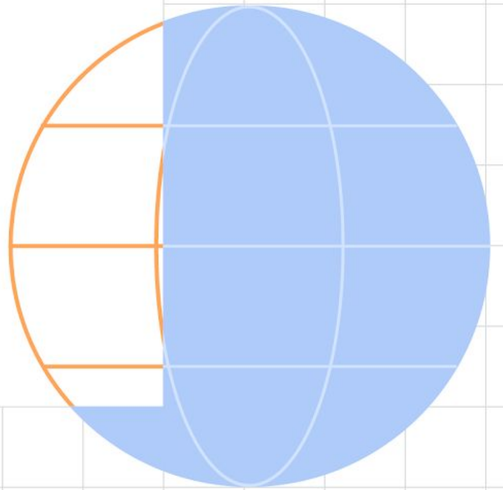
Usage: Colors

```dart
Wrap(
  children: [
    const _ColorCube(color: ColorName.black),
    const _ColorCube(color: ColorName.gray70),
    const _ColorCube(color: ColorName.gray410),
    _ColorCube(color: ColorName.crimsonRed.shade50),
    _ColorCube(color: ColorName.crimsonRed.shade100),
    _ColorCube(color: ColorName.crimsonRed.shade200),
    _ColorCube(color: ColorName.crimsonRed.shade300),
    _ColorCube(color: ColorName.crimsonRed.shade400),
    _ColorCube(color: ColorName.crimsonRed.shade500),
    _ColorCube(color: ColorName.crimsonRed.shade600),
    _ColorCube(color: ColorName.crimsonRed.shade700),
    _ColorCube(color: ColorName.crimsonRed.shade800),
    _ColorCube(color: ColorName.crimsonRed.shade900),
    _ColorCube(color: ColorName.yellowOcher.shade50),
    _ColorCube(color: ColorName.yellowOcher.shade100),
    _ColorCube(color: ColorName.yellowOcher.shade200),
    _ColorCube(color: ColorName.yellowOcher.shade300),
    _ColorCube(color: ColorName.yellowOcher.shade400),
    _ColorCube(color: ColorName.yellowOcher.shade500),
    _ColorCube(color: ColorName.yellowOcher.shade600),
    _ColorCube(color: ColorName.yellowOcher.shade700),
    _ColorCube(color: ColorName.yellowOcher.shade800),
    _ColorCube(color: ColorName.yellowOcher.shade900),
    _ColorCube(color: ColorName.yellowOcherAccent.shade100),
    _ColorCube(color: ColorName.yellowOcherAccent.shade200),
    _ColorCube(color: ColorName.yellowOcherAccent.shade400),
    _ColorCube(color: ColorName.yellowOcherAccent.shade700),
  ],
);
```



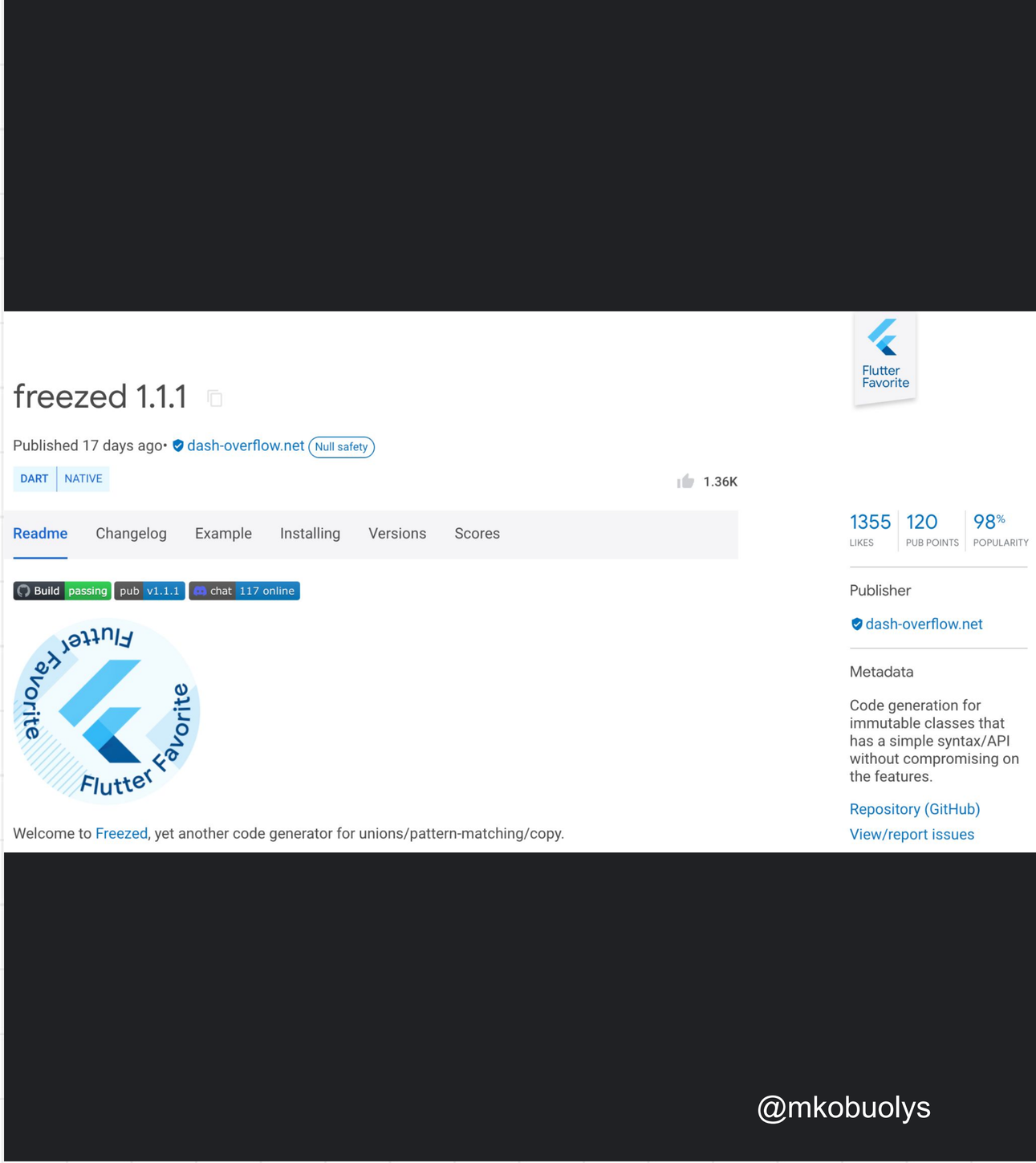
Experts

Objects & State



freezed

"… yet another code generator for unions/pattern-matching/copy."

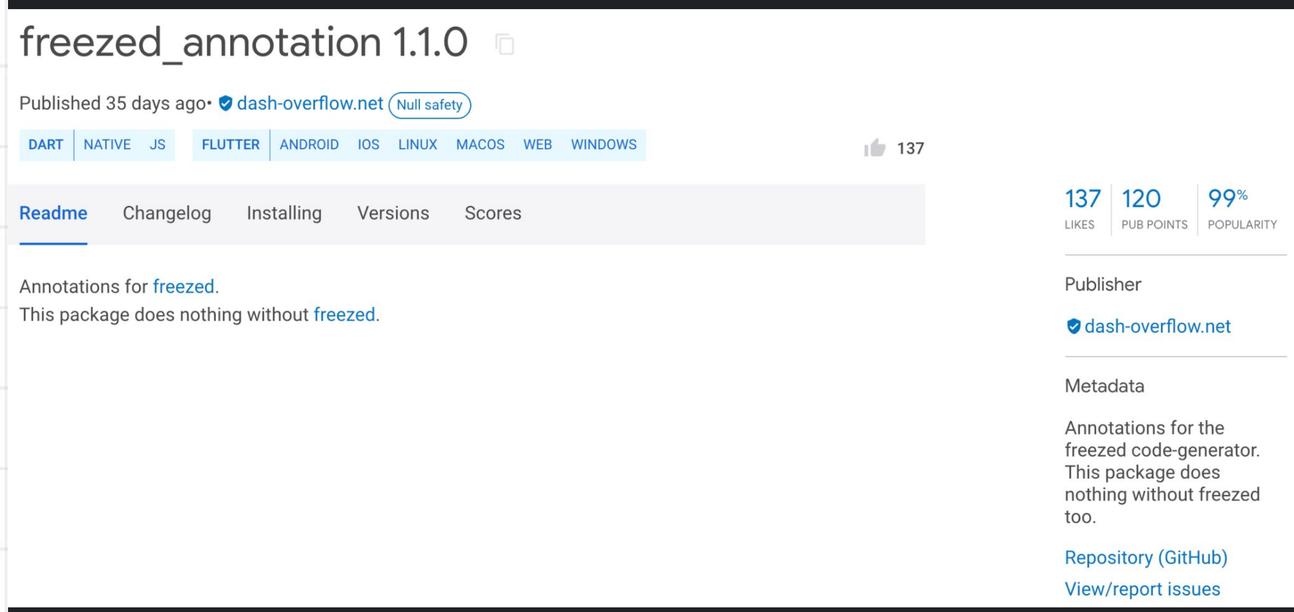


Experts

@mkobuolys

freezed_annotation

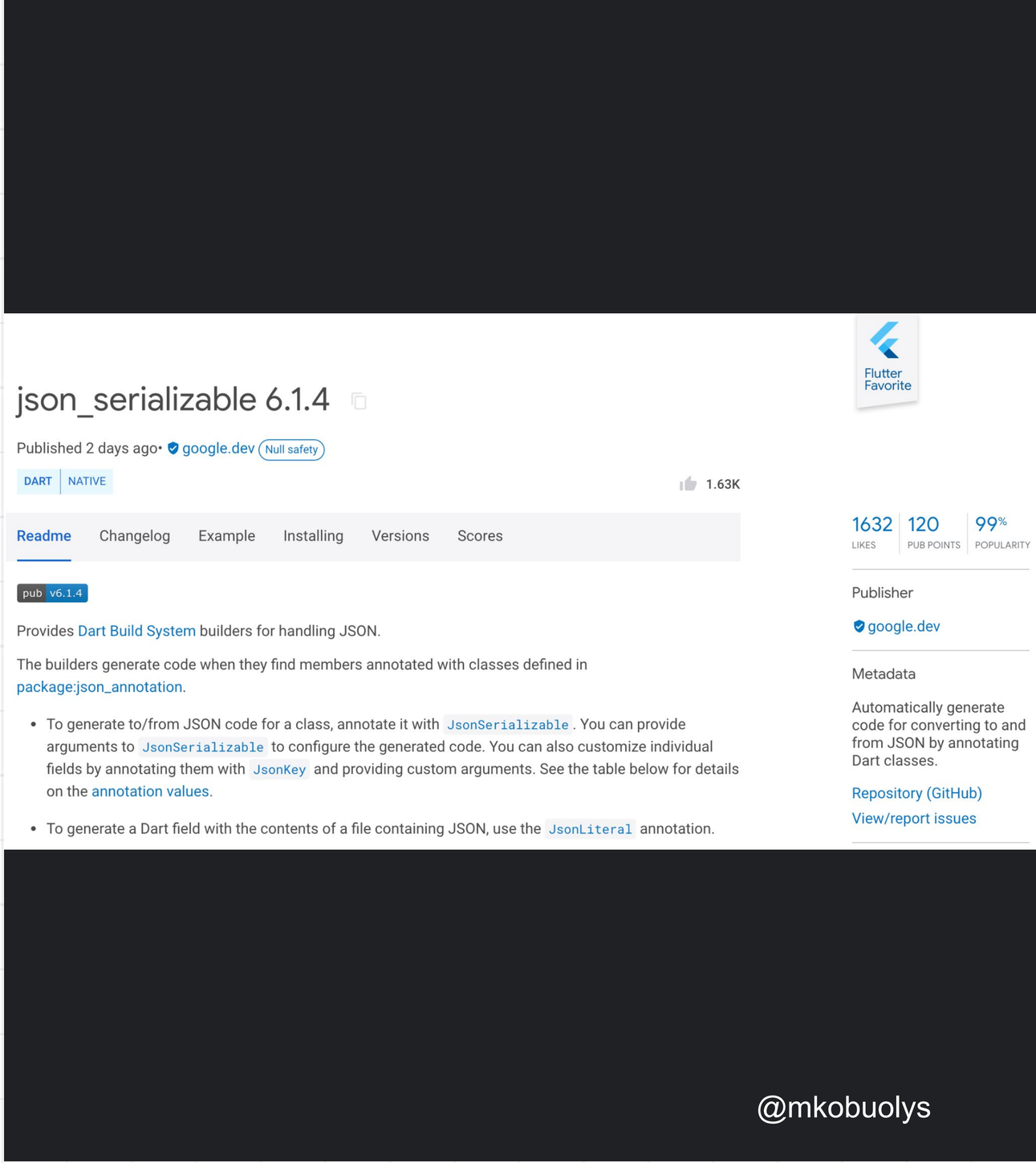
Annotations for freezed package.



Experts

json_serializable

Automatically generate code for converting to and from JSON by annotating Dart classes.



Experts

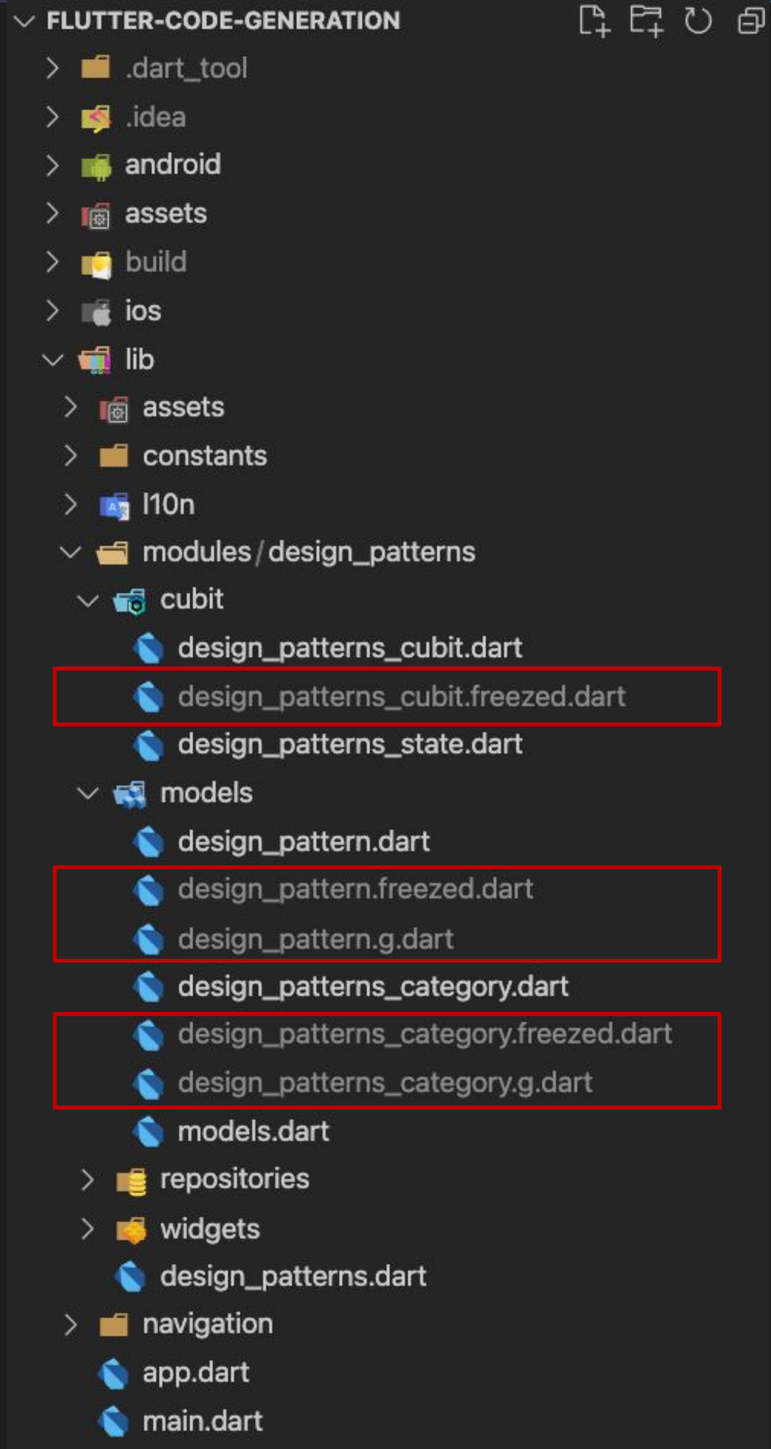
@mkobuolys

pubspec.yaml

```yaml
dependencies:
  flutter:
    sdk: flutter
  freezed_annotation: ^1.1.0

dev_dependencies:
  build_runner: ^2.1.7
  freezed: ^1.1.1
  json_serializable: ^6.1.4
```

Generating files

> flutter pub run build_runner build
or
> flutter pub run build_runner watch



Freezed class

```dart
part 'design_patterns_category.freezed.dart';
part 'design_patterns_category.g.dart';

@freezed
class DesignPatternsCategory with _$DesignPatternsCategory {
  const factory DesignPatternsCategory({
    required String id,
    required String title,
    required String color,
    required List<DesignPattern> patterns,
  }) = _DesignPatternsCategory;

  factory DesignPatternsCategory.fromJson(Map<String, dynamic> json) =>
      _$DesignPatternsCategoryFromJson(json);
}
```

Generated *.g.dart

```dart
_$_DesignPatternsCategory _$$_DesignPatternsCategoryFromJson(
        Map<String, dynamic> json) =>
    _$_DesignPatternsCategory(
      id: json['id'] as String,
      title: json['title'] as String,
      color: json['color'] as String,
      patterns: (json['patterns'] as List<dynamic>)
          .map((e) => DesignPattern.fromJson(e as Map<String, dynamic>))
          .toList(),
    );

Map<String, dynamic> _$$_DesignPatternsCategoryToJson(
        _$_DesignPatternsCategory instance) =>
    <String, dynamic>{
      'id': instance.id,
      'title': instance.title,
      'color': instance.color,
      'patterns': instance.patterns,
    };
```

Generated *.freezed.dart

```dart
@JsonSerializable()
class _$_DesignPatternsCategory implements _DesignPatternsCategory {
  const _$_DesignPatternsCategory(
      {required this.id,
      required this.title,
      required this.color,
      required this.patterns});

  factory _$_DesignPatternsCategory.fromJson(Map<String, dynamic> json) =>
      _$$_DesignPatternsCategoryFromJson(json);

  @override
  final String id;
  @override
  final String title;
  @override
  final String color;
  @override
  final List<DesignPattern> patterns;

  @override
  String toString() {
    return 'DesignPatternsCategory(id: $id, title: $title, '
      'color: $color, patterns: $patterns)';
  }
  ...
}
```

```dart
  ...
  @override
  bool operator ==(dynamic other) {
    return identical(this, other) ||
        (other.runtimeType == runtimeType &&
            other is _DesignPatternsCategory &&
            const DeepCollectionEquality().equals(other.id, id) &&
            const DeepCollectionEquality().equals(other.title, title) &&
            const DeepCollectionEquality().equals(other.color, color) &&
            const DeepCollectionEquality().equals(other.patterns, patterns));
  }

  @override
  int get hashCode => Object.hash(
      runtimeType,
      const DeepCollectionEquality().hash(id),
      const DeepCollectionEquality().hash(title),
      const DeepCollectionEquality().hash(color),
      const DeepCollectionEquality().hash(patterns));

  @JsonKey(ignore: true)
  @override
  _$DesignPatternsCategoryCopyWith<_DesignPatternsCategory> get copyWith =>
      __$DesignPatternsCategoryCopyWithImpl<_DesignPatternsCategory>(
          this, _$identity);

  @override
  Map<String, dynamic> toJson() {
    return _$$_DesignPatternsCategoryToJson(this);
  }
}
```

@mkobuolys

State management: State

```dart
abstract class DesignPatternsState extends Equatable {
  const DesignPatternsState();

  @override
  List<Object> get props => [];
}

class DesignPatternsInitial extends DesignPatternsState {}

class DesignPatternsLoadInProgress extends DesignPatternsState {}

class DesignPatternsLoadSuccess extends DesignPatternsState {
  final List<DesignPatternsCategory> categories;

  const DesignPatternsLoadSuccess([this.categories = const []]);

  @override
  List<Object> get props => [categories];

  @override
  String toString() => 'DesignPatternsLoadSuccess { categories: $categories }';
}

class DesignPatternsLoadFailure extends DesignPatternsState {}
```

```dart
@freezed
class DesignPatternsState with _$DesignPatternsState {
  const factory DesignPatternsState.initial() =
    _DesignPatternsStateInitial;
  const factory DesignPatternsState.loadInProgress() =
    _DesignPatternsStateLoadInProgress;
  const factory DesignPatternsState.loadSuccess({
    required List<DesignPatternsCategory> categories,
  }) = _DesignPatternsStateLoadSuccess;
  const factory DesignPatternsState.loadFailure() =
      _DesignPatternsStateLoadFailure;
}
```

State management: Cubit

```dart
class DesignPatternsCubit extends Cubit<DesignPatternsState> {
  final DesignPatternsRepository repository;

  DesignPatternsCubit({
    required this.repository,
  }) : super(const DesignPatternsState.initial());

  Future<void> load() async {
    emit(const DesignPatternsState.loadInProgress());

    try {
      final categories = await repository.getCategories();

      emit(DesignPatternsState.loadSuccess(categories: categories));
    } on DesignPatternsJsonNotFoundException {
      emit(const DesignPatternsState.loadFailure());
    }
  }
}
```

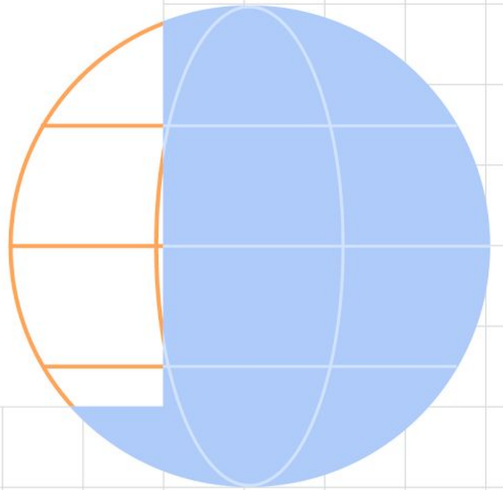
State management: Usage

```dart
@override
Widget build(BuildContext context) {
  return BlocBuilder<DesignPatternsCubit, DesignPatternsState>(
    builder: (_, state) {
      if (state is DesignPatternsStateLoadInProgress) {
        return const _LoaderView();
      } else if (state is DesignPatternsStateLoadFailure) {
        return const _ErrorView();
      } else if (state is DesignPatternsStateLoadSuccess) {
        return DesignPatternsListView(
          categories: state.categories,
        );
      } else {
        return const SizedBox();
      }
    },
  );
}
```

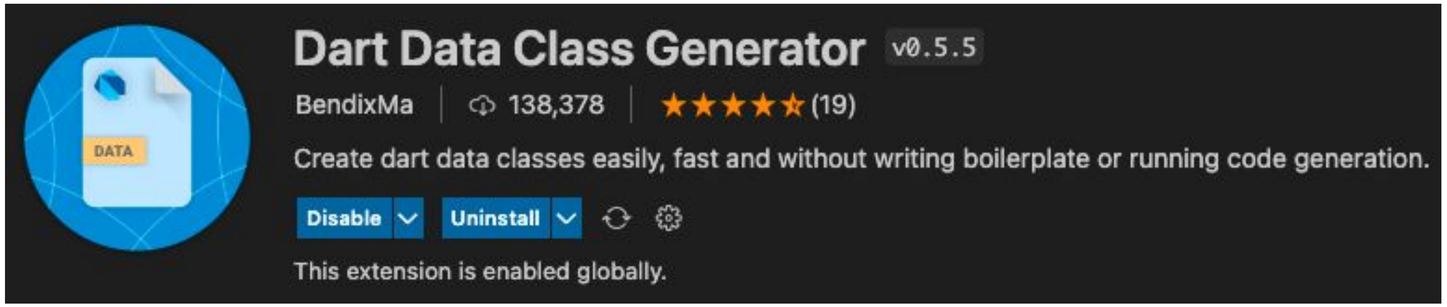
```dart
@override
Widget build(BuildContext context) {
  return BlocBuilder<DesignPatternsCubit, DesignPatternsState>(
    builder: (_, state) => state.maybeWhen(
      loadInProgress: () => const _LoaderView(),
      loadFailure: () => const _ErrorView(),
      loadSuccess: (categories) => DesignPatternsListView(
        categories: categories,
      ),
      orElse: () => const SizedBox(),
    ),
  );
}
```

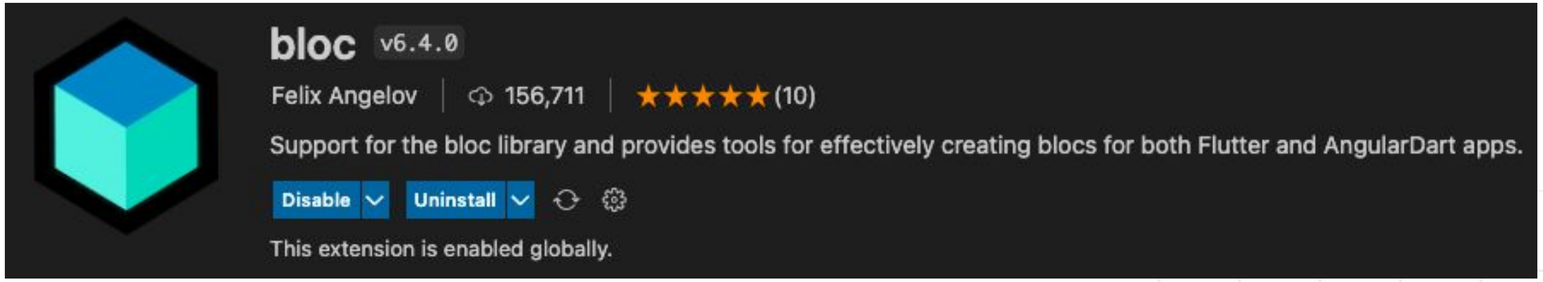
Experts

Widgets

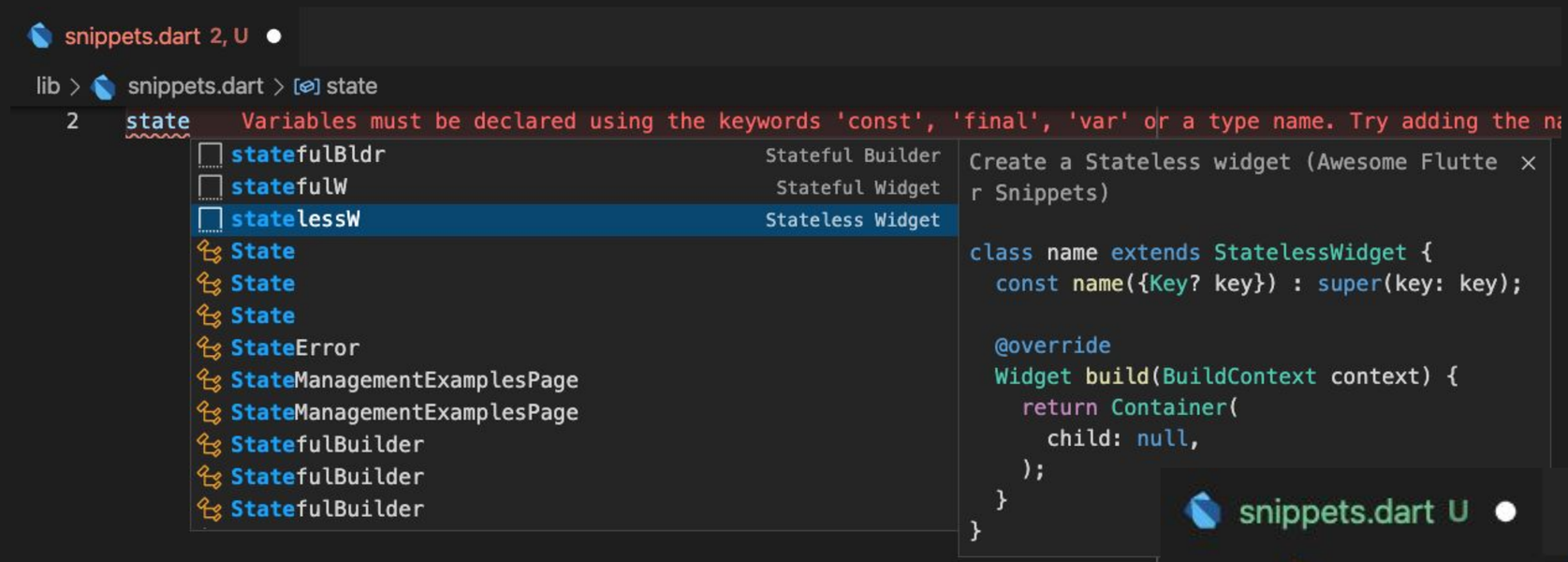


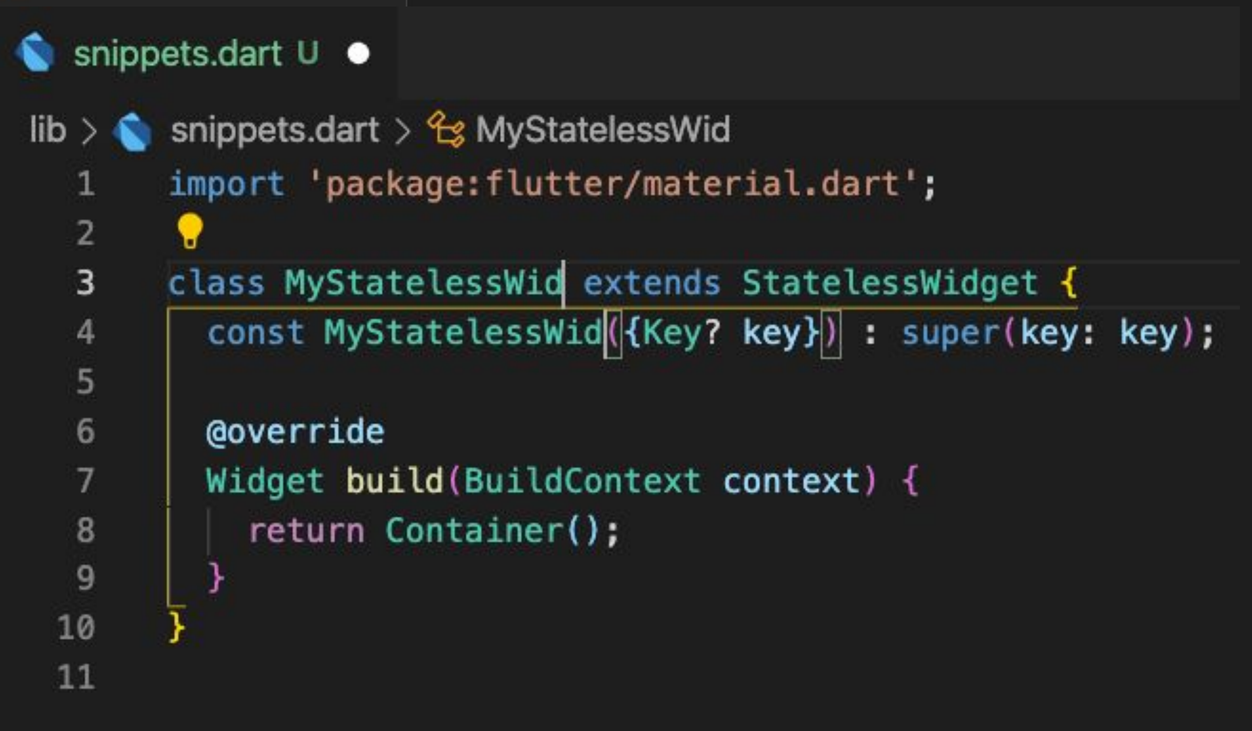
IDE Extensions





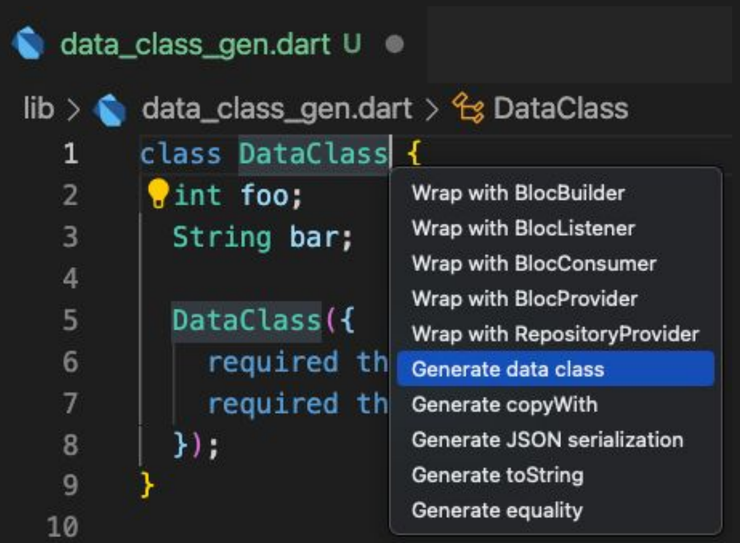
Awesome Flutter Snippets





@mkobuolys

Dart Data Class Generator



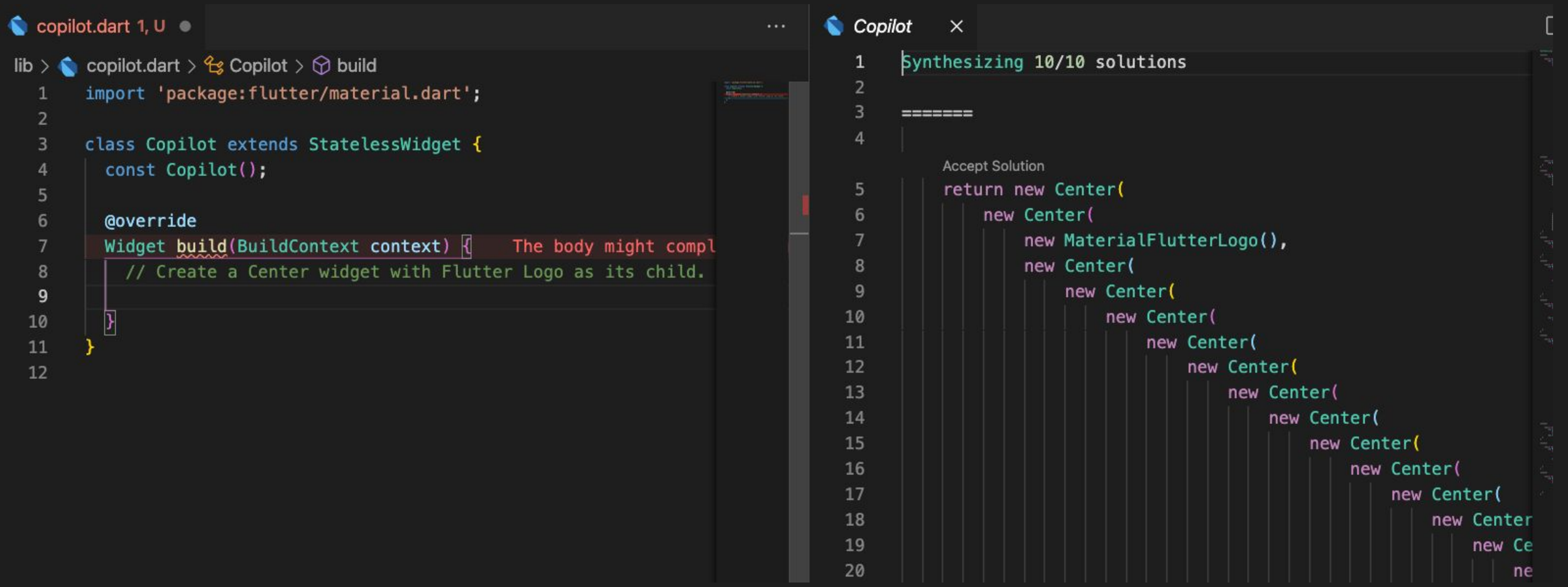

```dart
class DataClass {
  int foo;
  String bar;

  DataClass({
    required this.foo,
    required this.bar,
  });

  DataClass copyWith({
    int? foo,
    String? bar,
  }) {
    return DataClass(
      foo: foo ?? this.foo,
      bar: bar ?? this.bar,
    );
  }

  Map<String, dynamic> toMap() {
    return {
      'foo': foo,
      'bar': bar,
    };
  }

  factory DataClass.fromMap(Map<String, dynamic> map) {
    return DataClass(
      foo: map['foo'],
      bar: map['bar'],
    );
  }
  ...
}
```

```dart
  ...
  String toJson() => json.encode(toMap());

  factory DataClass.fromJson(String source) =>
      DataClass.fromMap(json.decode(source));

  @override
  String toString() => 'DataClass(foo: $foo, bar: $bar)';

  @override
  bool operator ==(Object other) {
    if (identical(this, other)) return true;

    return other is DataClass && other.foo == foo && other.bar == bar;
  }

  @override
  int get hashCode => foo.hashCode ^ bar.hashCode;
}
```

@mkobuolys

AI-powered tools

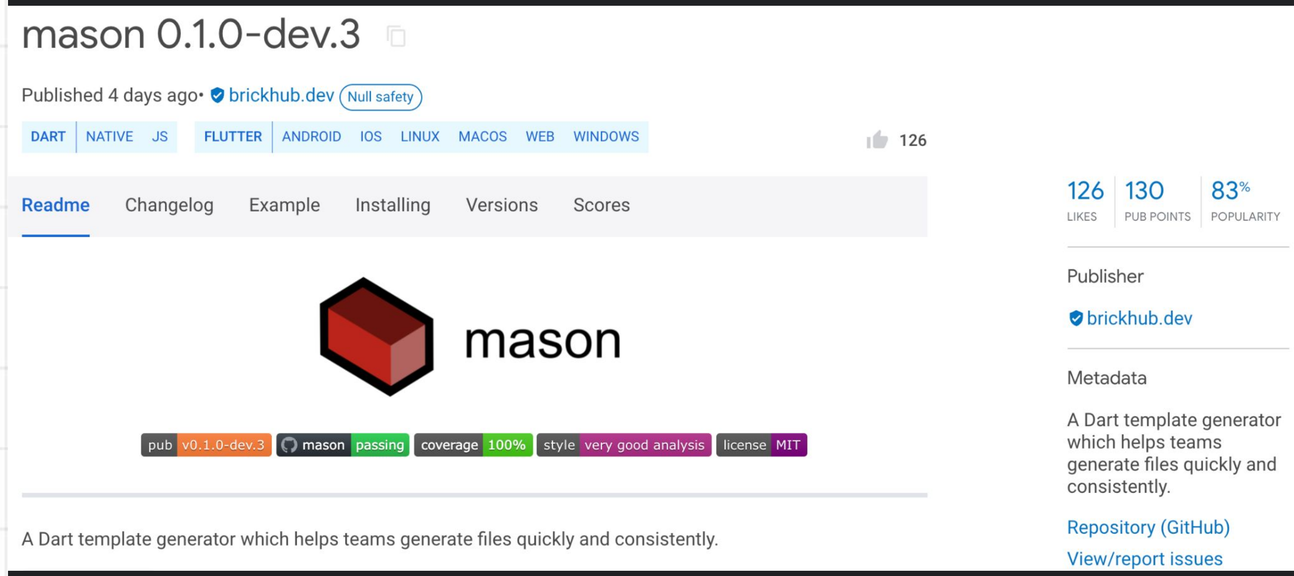
Too smart?



@mkobuolys

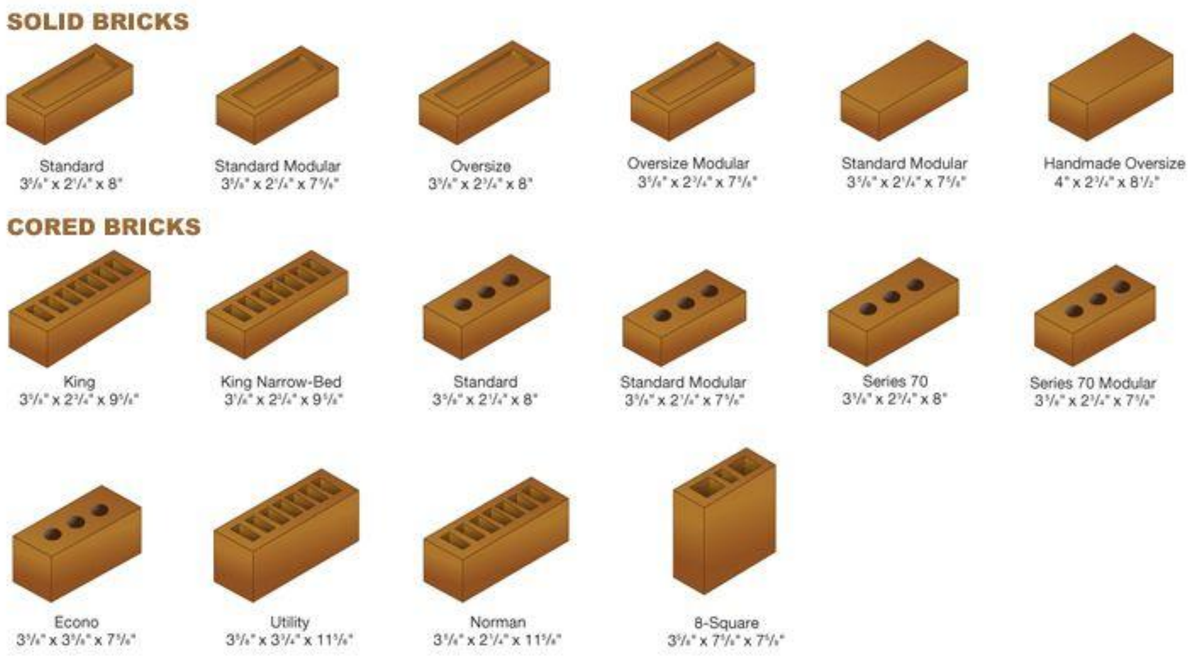
mason

A Dart template generator which helps teams generate files quickly and consistently.



Experts

Mason: Concept



Mason: Expected result

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

class CustomPage extends StatelessWidget {
  static PageRoute route() {
    return MaterialPageRoute(builder: (context) => const CustomPage());
  }

  const CustomPage();

  @override
  Widget build(BuildContext context) {
    return const _CustomPageView();
  }
}

class _CustomPageView extends StatelessWidget {
  const _CustomPageView();

  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

Mason: Expected result

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

class CustomPage extends StatelessWidget {
  static PageRoute route() {
    return MaterialPageRoute(builder: (context) => const CustomPage());
  }

  const CustomPage();

  @override
  Widget build(BuildContext context) {
    return const _CustomPageView();
  }
}

class _CustomPageView extends StatelessWidget {
  const _CustomPageView();

  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

Any name

Optional

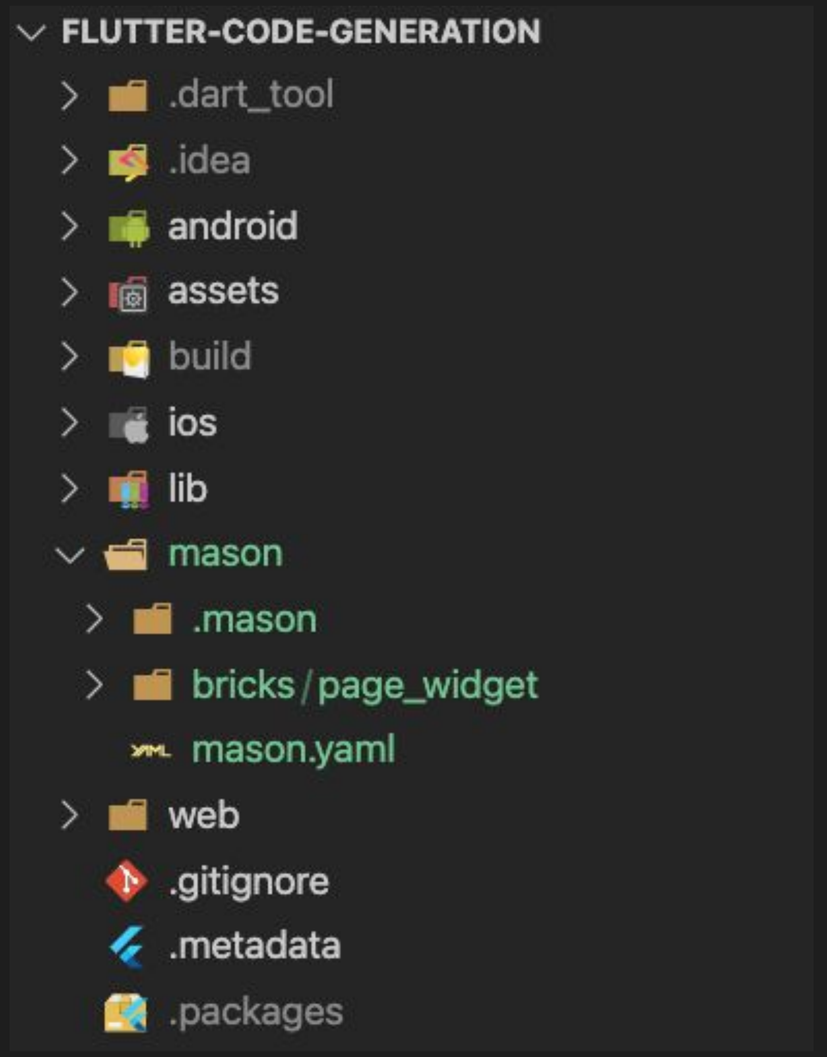
Mason: Initial configuration

> mason init
> mason new page_widget
> mason get

mason.yaml U

mason > mason.yaml > ...

```yaml
bricks:
  page_widget:
    path: bricks/page_widget
```



Mason: Brick structure/dynamic name

brick.yaml U

mason > bricks > page_widget > brick.yaml > [ ] vars

```yaml
name: page_widget
description: A custom page widget brick.
vars:
  - name
  - routable
```

{{#snakeCase}}{{name}}_page{{/snakeCase}}.dart

Mason: Brick

Checks routable flag

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

class {{#pascalCase}}{{name}} page {{/pascalCase}} extends StatelessWidget {
  {{#routable}}
  static PageRoute route() {
    return MaterialPageRoute(builder: (context) => const {{#pascalCase}}{{name}} page {{/pascalCase}}());
  }
  {{/routable}}

  const {{#pascalCase}}{{name}} page {{/pascalCase}}();

  @override
  Widget build(BuildContext context) {
    return const _{{#pascalCase}} {{name}} page view {{/pascalCase}}();
  }
}

class _{{#pascalCase}}{{name}} page view {{/pascalCase}} extends StatelessWidget {
  const _{{#pascalCase}}{{name}} page view {{/pascalCase}}();

  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

Dynamic name in PascalCase

Mason: Generating code

> mason make page_widget –o …

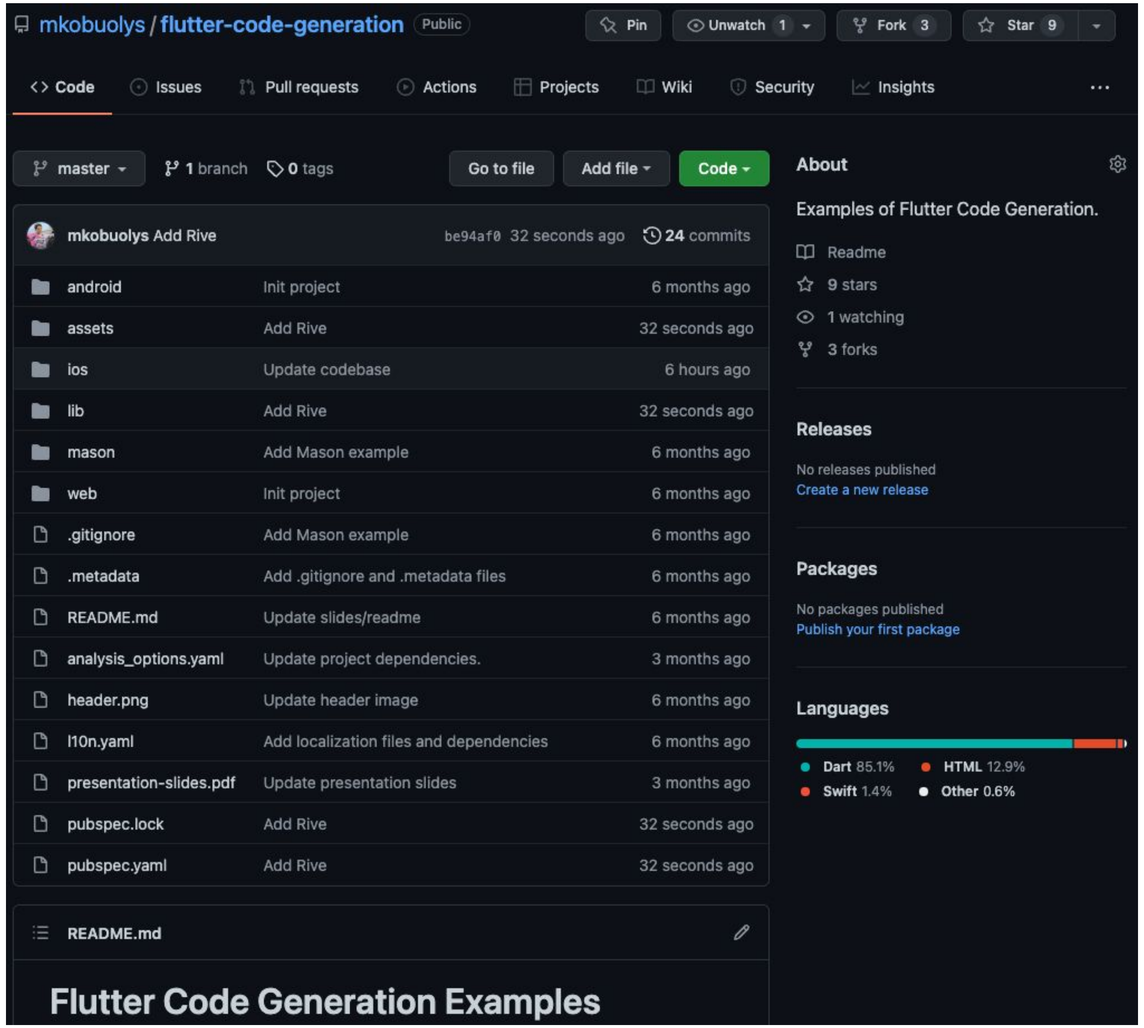
```
Mangirdass-MBP:mason mkobuolys$ mason make page_widget -o ../lib/
name: Custom
routable: true
✓ Made brick page_widget (7.7ms)
✓ Generated 1 file(s):
  /Users/mkobuolys/dev/Flutter projects/flutter-code-generation/lib/custom_page.dart (new)
```

Mason: Expected vs Actual result

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

class CustomPage extends StatelessWidget {
  static PageRoute route() {
    return MaterialPageRoute(builder: (context) => const CustomPage());
  }

  const CustomPage();

  @override
  Widget build(BuildContext context) {
    return const _CustomPageView();
  }
}

class _CustomPageView extends StatelessWidget {
  const _CustomPageView();

  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

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

class CustomPage extends StatelessWidget {

  static PageRoute route() {
    return MaterialPageRoute(builder: (context) => const CustomPage());
  }


  const CustomPage();

  @override
  Widget build(BuildContext context) {
    return const _CustomPageView();
  }
}

class _CustomPageView extends StatelessWidget {
  const _CustomPageView();

  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

@mkobuolys



https://github.com/mkobuolys/flutter-code-generation

Resources

- Localization
  - Internationalizing Flutter apps – https://bit.ly/3z945Xu
  - Internationalization User Guide - https://bit.ly/3iHdOOv
  - intl - https://pub.dev/packages/intl
- Assets
  - build_runner - https://pub.dev/packages/build_runner
  - flutter_gen - https://pub.dev/packages/flutter_gen

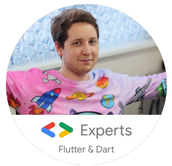
Resources

- Objects & State
  - freezed – https://pub.dev/packages/freezed
  - freezed_annotation - https://pub.dev/packages/freezed_annotation
  - json_serializable - https://pub.dev/packages/json_serializable
- Widgets
  - GitHub Copilot - https://copilot.github.com/
  - mason - https://pub.dev/packages/mason

Google Developers

Experts

Save trees. Stay SOLID. Thank you!



Mangirdas Kazlauskas
GDE for Flutter & Dart
@mkobuolys