Source Code Generate on Dart

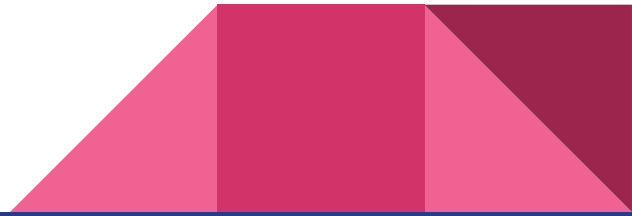
Dartのソースコード生成事情について

@sh4869sh

[宣伝]Flutter Pluginの作り方の記事が出ます

多分（絶賛執筆中）

- 技術書展4でTechBoosterにて出るはず
- FlutterのPluginの書き方を説明したもの



結論

dart:mirrorsを使うな
source_genを使おう

dart:mirrorsとは

- Dartでリフレクション等をサポートするためのライブラリ

```dart
import 'dart:mirrors';

class MyClass {
  int i, j;
  int sum() => i + j;

  MyClass(this.i, this.j);
}

InstanceMirror myClassInstanceMirror = reflect(new MyClass(3, 4));
InstanceMirror f = myClassInstanceMirror.invoke(#sum, []); // Returns an InstanceMirror on 7
```

dart:mirrorsの問題

- dart2jsでmirrorsを使ったプログラムをトランスパイルするとひどい
  - 参考 リフレクション (dart:mirrors) を使ったライブラリを dart2js する - Qiita
  - 2MBとかになる……
  - dartdevcだとそもそもが大きいのであまりわからなかった (デフォルトで2MB)
- そもそもフロントエンドでは使わないことが鉄板とされてきた
- サーバーサイドでソースコード生成に使うぐらい？
  - あまり利用例がない
  - JSONオブジェクト変換用ライブラリ dartsonなどは使ってる
  - https://github.com/eredo/dartson

dart:mirrorsの問題を緩和する

リフレクションを取得するクラスに@MirrorUsedを使う

- バイナリサイズが抑えられる
- リフレクションを取得するクラスすべてに使わないといけない

dart:mirrorsが使えなくなる

- Flutterではすでに使えない
- おそらく2.0で使えなくなる
  - 参照:
  - https://github.com/flutter/flutter/issues/1150
  - https://www.dartlang.org/articles/dart-vm/reflection-with-mirrors に
  - *This article applies only to the standalone VM under the 1.x Dart SDK. We don't recommend using mirrors in web applications, and the Flutter SDK does not support the dart:mirrors library.*
  - とあるので

dart:mirrorsの代替

reflectable package

- Dart Team公式のライブラリ https://github.com/dart-lang/reflectable
- 概ねやりたいことはできる
- 一部できないことも
  - Private宣言はサポートされてない
  - 関数自体もReflectionできない

リフレクションは諦めよう

静的にコード生成するほうが楽

- source_genライブラリを使おう
  - 静的にソースコードを生成するためのライブラリ
  - ツールとして使う場合にはビルドサイズに影響ない
  - mirrorsも使ってないので安心して使える
  - コード生成なのでどのプラットフォームでも共通して使えるのが旨味
- 使用例：json_serializable
  - https://github.com/dart-lang/json_serializable
  - JSONシリアライズとデシリアライズ用のコードを生成してくれる

source_gen

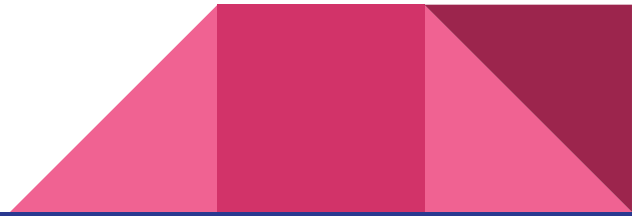
GeneratorとGeneratorForAnnotationがある

Generator

- ライブラリに対してソースコードを生成する

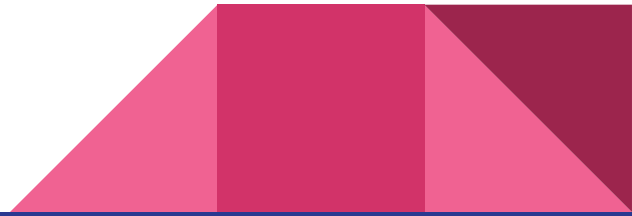
GeneratorForAnnotation

- Annotationがついたコードに対してソースコードを生成する



source_gen

- GeneratorForAnnotationを使うとJSON・CSV等のデータフォーマットに対応するクラスの自動生成ができる
- サンプル
  - https://github.com/sh4869/csv_code_generator.dart



source_genの使い方

`build_runner`パッケージを使う

- 公式が提供しているDartプロジェクト用のビルドシステム
- ソースコード生成等ができる
- 詳しい説明は省略
- 簡単に説明をします

参考: https://github.com/dart-lang/json_serializable/tree/master/example

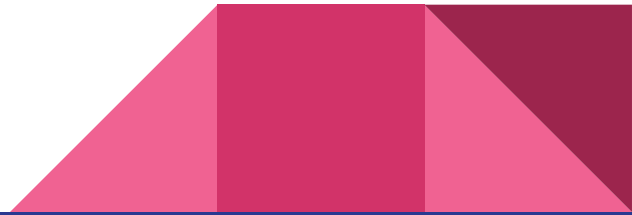
build_runner

- dev_dependenciesにbuild_runnerを追加してpub get
- build.yamlを編集
- targetに対してどのBuilderを利用するかを記述する

ビルダーのオプションとは

Builderとして提供するクラスもbuild.yamlを書く必要がある

- おそらくbuildシステムがdepndenciesを解決するときにそのbuild.yamlを見るのだと思う
- このbuild.yamlの中にどのDartファイルがBuilderを提供するかどうかを記述する
- そのBuilderオブジェクトの中にオプションが書いてある



json_serializable

生成用コード

```dart
import 'package:json_annotation/json_annotation.dart';
part 'test.g.dart';

@JsonSerializable()
class Person extends Object with _$PersonSerializerMixin {
  final String firstName;
  final String lastName;

  @JsonKey(name: 'date-of-birth', nullable: false)
  final DateTime dateOfBirth;

  Person(this.firstName, this.lastName, this.dateOfBirth);
  factory Person.fromJson(Map<String, dynamic> json) => _$PersonFromJson(json);
}
```

JSON用のシリアライザを生成したい
クラスにつける

json_serializable

生成後

```dart
// GENERATED CODE - DO NOT MODIFY BY HAND

part of 'test.dart';

// **************************************************************************
// Generator: JsonSerializableGenerator
// **************************************************************************

Person _$PersonFromJson(Map<String, dynamic> json) => new Person(
    json['firstName'] as String,
    json['lastName'] as String,
    DateTime.parse(json['date-of-birth'] as String));

abstract class _$PersonSerializerMixin {
  String get firstName;
  String get lastName;
  DateTime get dateOfBirth;
  Map<String, dynamic> toJson() => <String, dynamic>{
        'firstName': firstName,
        'lastName': lastName,
        'date-of-birth': dateOfBirth.toIso8601String()
      };
}
```

他のコード生成ツール

- swagger-codegen
  - https://github.com/swagger-api/swagger-codegen Dart対応済み
- protbuf
  - https://github.com/dart-lang/dart-protoc-plugin
- Discovery Document対応ライブラリもあります
  - https://github.com/dart-lang/discoveryapis_generator
- code_builderというライブラリもある
  - https://github.com/dart-lang/code_builder/
  - ソースコードを生成するためのライブラリ
- AnalyzerからASTを触ることもできる
  - https://github.com/dart-lang/sdk/tree/master/pkg/analyzer

おまけ

json_serializeable パッケージの環境

```
environment:
  sdk: '>=2.0.0-dev.9 <2.0.0'
```

build_runner パッケージの環境

```
environment:
  sdk: '>=2.0.0-dev.20 <2.0.0'
```

2.0系使ってるやつが多いので注意

おまけ2

prtobuf の compiler for dartはsource_gen使ってなかった……

lines (17 sloc) | 410 Bytes — Raw | Blame | History

```
name: protoc_plugin
version: 0.7.10
author: Dart Team <misc@dartlang.org>
description: Protoc compiler plugin to generate Dart code
homepage: https://github.com/dart-lang/dart-protoc-plugin
environment:
  sdk: '>=1.21.0 <2.0.0'
dependencies:
  fixnum: ^0.10.5
  path: ^1.0.0
  protobuf: ^0.7.1
  dart_style: ^1.0.6
dev_dependencies:
  browser: any
  test: ^0.12.0
executables:
  protoc-gen-dart: protoc_plugin
```