(Source Code)

Generating code from code to create Flutter apps

The Presenter

A **brief** personal intro

- For Education: BSc in Computer Science, actively doing a Data Science MSc
- Work: Full-stack developer with a focus on Flutter consulting
- Interests: Flutter, Python, Node.js, React, DX, ML, Visual data analysis
- W Handles:
 - **Solution** @peter_gyarmati

 - in @petergy

The Presentation Things we'll discuss tonight

- What is code generation and how would it help me?
- How does code generation work in Flutter?
- Common code gen. use cases and best practices in Flutter.
- Hands-on example
- Discussion & Wrap-up

What is code generation and how would it help me?

TL;DR: You write code to automatically produce more code

- The problem at hand: boilerplate code is useful, but
 - Writing getters, setters, serializers, etc. is tedious
 - Makes the codebase more difficult to maintain
 - A universal, language- and framework-agnostic problem
- Two main techniques to the rescue:
 - Reflection: allows a program to inspect and modify its own behavior at runtime
 - Code gen: auto generating code based on some input at build time
 - Description The source code you write will be the source of other source code (Source Code)



What is code generation and how would it help me?

"Talk is cheap. Show me the code." ~ Linus Torvalds

Before

```
Map<String, Object?> toJson() {
@immutable
                                                              return {
class Person {
                                                                 'firstName': firstName,
 const Person({
                                                                'lastName': lastName,
   required this.firstName,
                                                                 'age': age,
   required this.lastName,
                                                              };
   required this.age,
 });
                                                            @override
 factory Person.fromJson(Map<String, Object?> json) {
                                                            String toString() {
   return Person(
                                                              return 'Person('
     firstName: json['firstName'] as String,
                                                                 'firstName: $firstName,
     lastName: json['lastName'] as String,
                                                                'lastName: $lastName, '
     age: json['age'] as int,
                                                                'age: $age'
                                                              ')';
 final String firstName;
                                                            @override
 final String lastName;
                                                            bool operator =(Object other) {
 final int age;
                                                              return other is Person &&
                                                                  person.runtimeType = runtimeType &&
 Person copyWith({
                                                                  person.firstName = firstName &&
   String? firstName,
                                                                  person.lastName = lastName &&
   String? lastName,
                                                                  person.age = age;
   int? age,
 }) {
   return Person(
                                                            @override
     firstName: firstName,
                                                            int get hashCode {
     lastName: lastName,
                                                               return Object.hash(
      age: age,
                                                                runtimeType,
                                                                firstName,
                                                                lastName,
                                                                age,
```

After

```
@freezed
class Person with _$Person {
  const factory Person({
    required String firstName,
    required String lastName,
    required int age,
  }) = _Person;
  factory Person.fromJson(Map<String, Object?> json)
      ⇒ _$PersonFromJson(json);
```

Source: https://pub.dev/packages/freezed

How does code generation work in Flutter?

- No access to reflection in Flutter
 - dart:mirrors is not in the SDK due to app size considerations
 - Our only way to get rid of boilerplate is code gen.
- Dart has a first-class tooling for code gen. (build_runner, source_gen, ...)
- On a **very** high-level
 - Pick a code gen package from pub.dev
 - Install build_runner
 - Install the package-specific dependencies
 - Run flutter pub run build_runner watch --delete-conflicting-outputs

Common code gen. use-cases and best practices in Flutter

- JSON serialozation / deserialization: json_serializable
- Dataclasses: freezed
- Fully typed asset paths: flutter_gen
- Routing: auto_route
- Dependency Injection: injectable
- HTTP Client: chopper
- Internationalization (i18n): flutter_i18n, easy_localization



Let's live-code a simple app, using: json_serializable, freezed and flutter_gen

Project source:

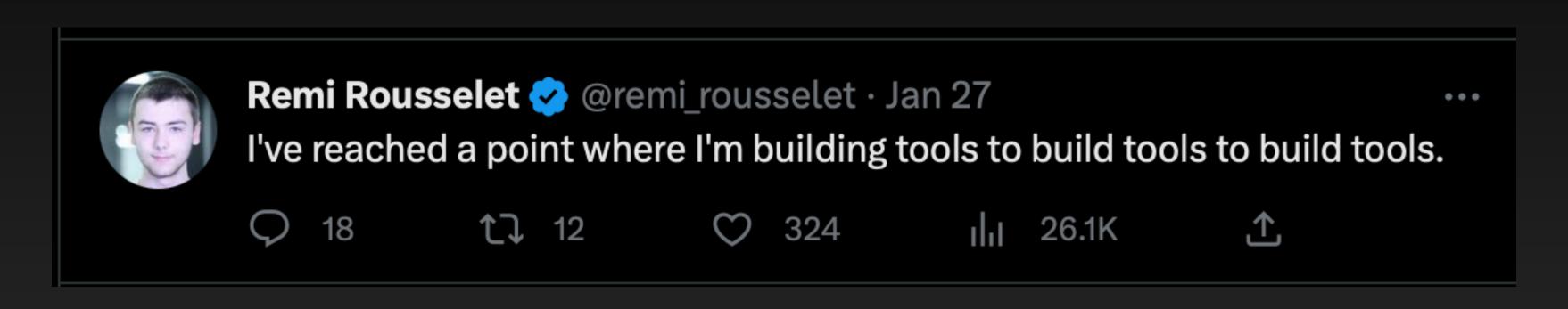
dub.sh/fltr



or if you are the type of guy who likes to type: github.com/peter-gy/source-code-squared

Discussion & Wrap-up

The code gen. rabbit hole is deep



Author of Riverpod, Provider, Freezed and many other Flutter packages