# $(\text{Source Code})^2$

## Generating code from code to create Flutter apps

Péter Ferenc Gyarmati, 2023. 04. 20.

*10th Flutter Vienna Meetup in Vienna*

# The Presenter

## A **brief** personal intro

- 🎓 **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
- 👋 **Handles:**
  - @peter_gyarmati
  - @peter-gy
  - @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 <u>at runtime</u>*
  - Code gen: *auto generating code based on some input <u>at build time</u>*
  - ➡️ The source code you write will be the source of other source code **$(\text{Source Code})^2$**

# 🤖 What is code generation and how would it help me?

*“Talk is cheap. Show me the code.”* ~ Linus Torvalds

## Before

```dart
@immutable
class Person {
  const Person({
    required this.firstName,
    required this.lastName,
    required this.age,
  });

  factory Person.fromJson(Map<String, Object?> json) {
    return Person(
      firstName: json['firstName'] as String,
      lastName: json['lastName'] as String,
      age: json['age'] as int,
    );
  }

  final String firstName;
  final String lastName;
  final int age;

  Person copyWith({
    String? firstName,
    String? lastName,
    int? age,
  }) {
    return Person(
      firstName: firstName,
      lastName: lastName,
      age: age,
    );
  }
```

```dart
  Map<String, Object?> toJson() {
    return {
      'firstName': firstName,
      'lastName': lastName,
      'age': age,
    };
  }

  @override
  String toString() {
    return 'Person('
      'firstName: $firstName, '
      'lastName: $lastName, '
      'age: $age'
    ')';
  }

  @override
  bool operator ==(Object other) {
    return other is Person &&
        person.runtimeType == runtimeType &&
        person.firstName == firstName &&
        person.lastName == lastName &&
        person.age == age;
  }

  @override
  int get hashCode {
    return Object.hash(
      runtimeType,
      firstName,
      lastName,
      age,
    );
  }
}
```

## After

```dart
@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 serialization / 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

## 💻 Hands-on Example

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

Project source:

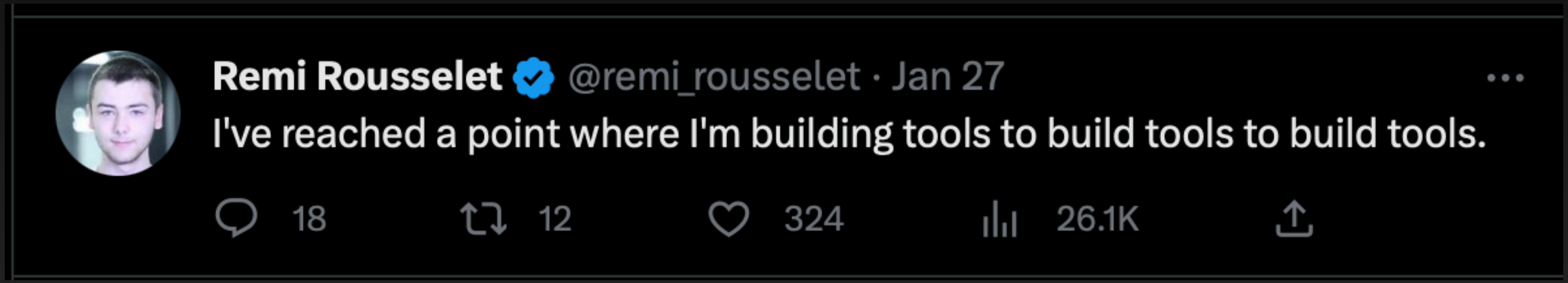
dub.sh/fltr

*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



Remi Rousselet @remi_rousselet · Jan 27

I've reached a point where I'm building tools to build tools to build tools.

18 12 324 26.1K

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