

Fried Flutter: preparing Flutter for commercial development



vladimir Ivanov



vladimir Ivanov

- Solution Architect @ EPAM Systems



vladimir Ivanov

- Solution Architect @ EPAM Systems
- Experienced in Android, React-Native & Flutter



Today we

Today we

- Know about Quality Standards in EPAM

Today we

- Know about Quality Standards in EPAM
- Know the project

Today we

- Know about Quality Standards in EPAM
- Know the project
- Face the issues

Today we

- Know about Quality Standards in EPAM
- Know the project
- Face the issues
- Make the conclusions

Engineering Excellence in EPAM



Engineering Excellence in EPAM

- Unified Code Style



Engineering Excellence in **EPAM**

- Unified Code Style
- Static analysis



Engineering Excellence in **EPAM**

- Unified Code Style
- Static analysis
- Unit tests



Engineering Excellence in EPAM

- Unified Code Style
- Static analysis
- Unit tests
- Tests coverage



Engineering Excellence in EPAM

- Unified Code Style
- Static analysis
- Unit tests
- Tests coverage
- Quality Gates



Our project

Our project

- Innovative way of learning English.

Our project

- Innovative way of learning English.
- AI to help learning

Project Specifics

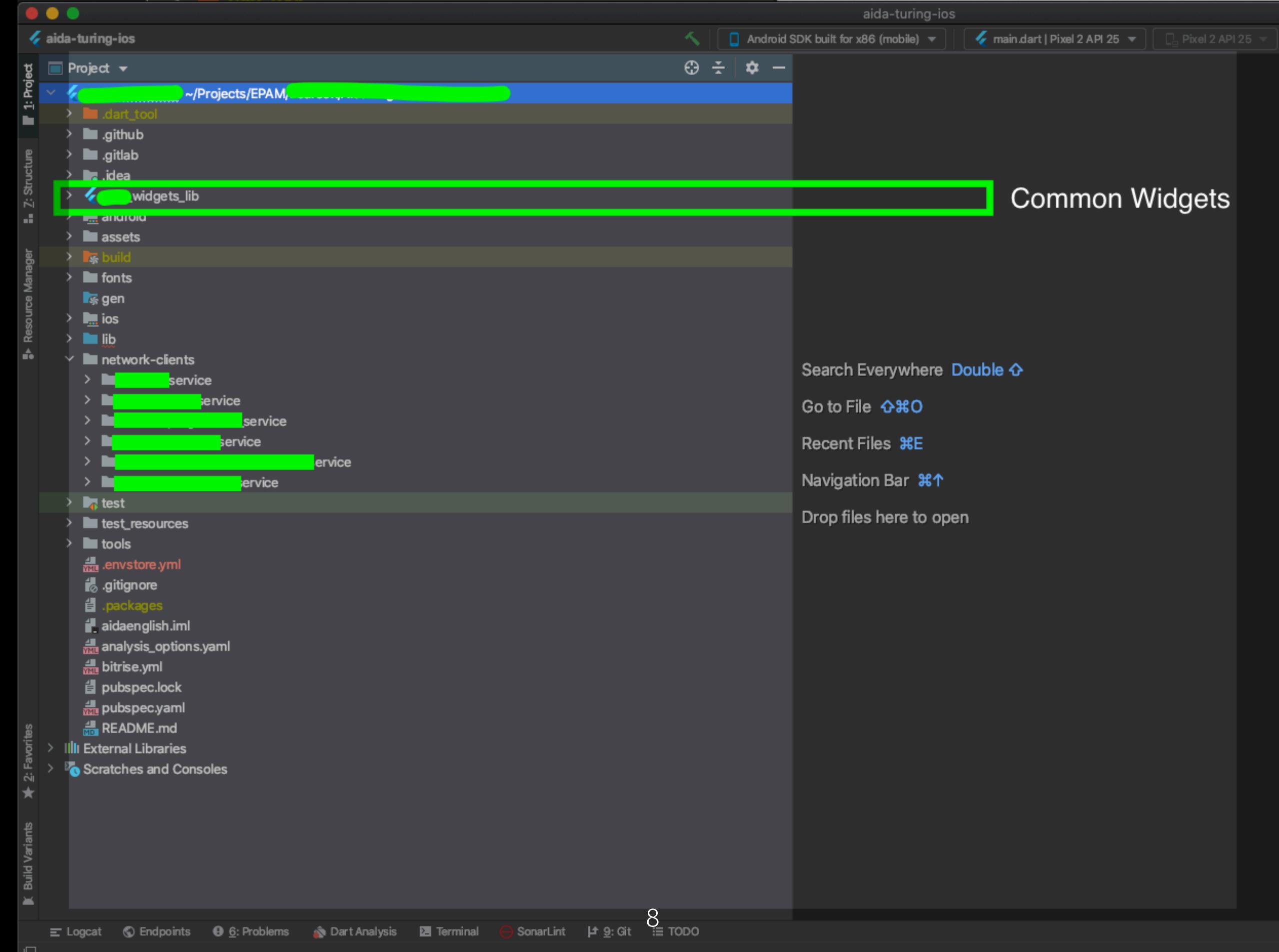
Project Specifics

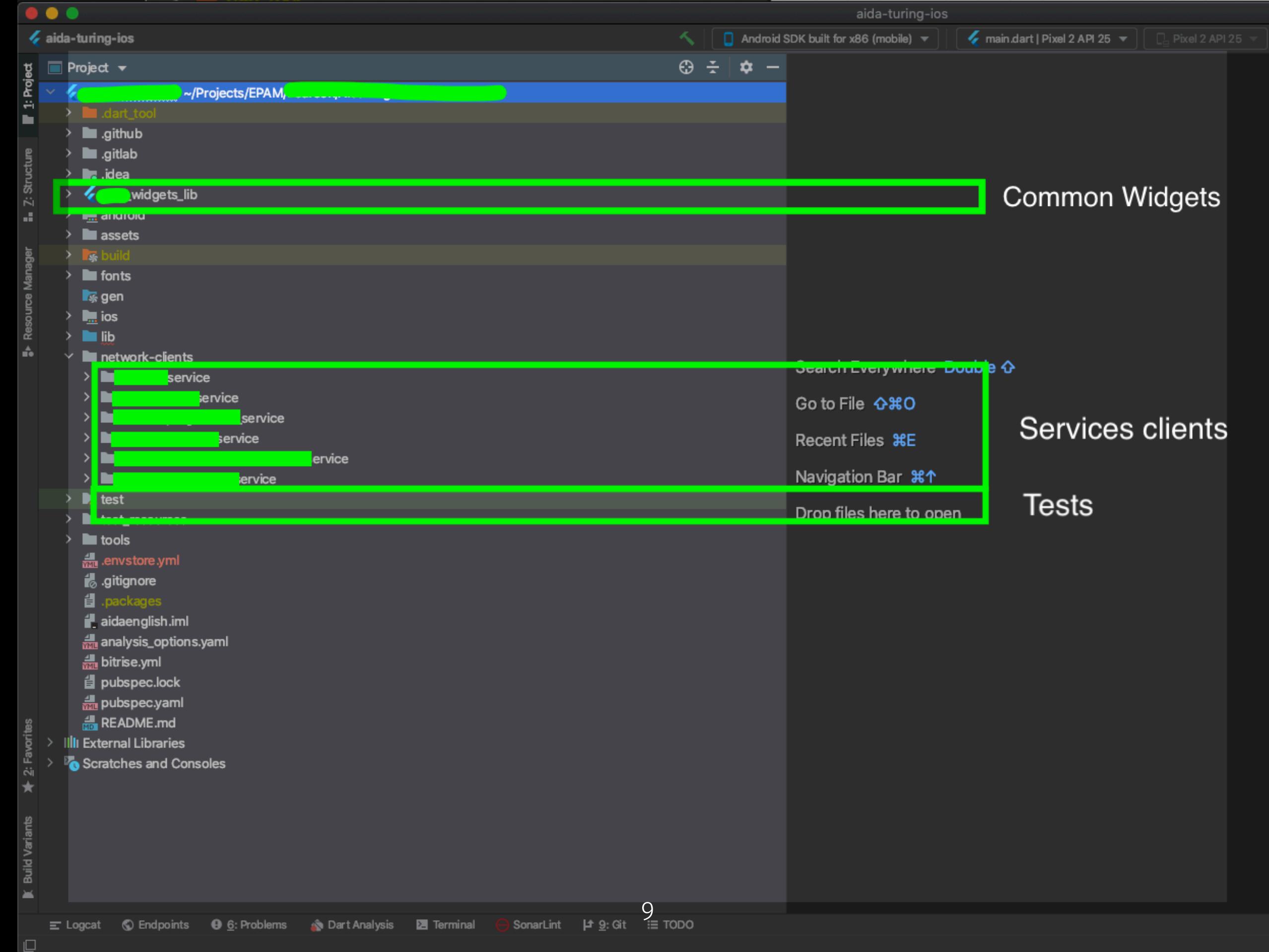
- High number of integrations

Project Specifics

- High number of integrations
- The initial team was not familiar with Flutter, only React-Native

Project Structure





What we faced



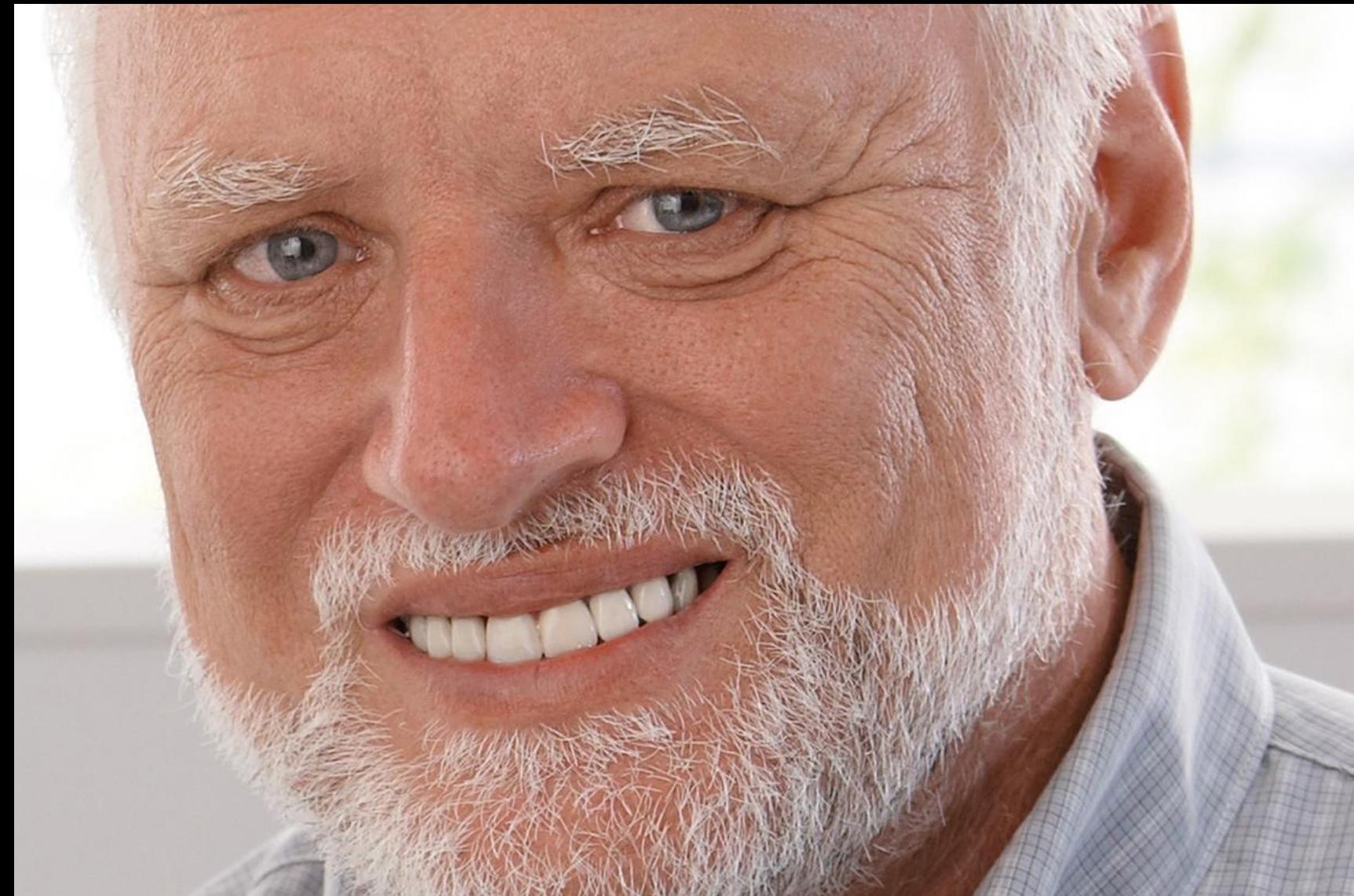
What we faced

- Relied on default code style / static analysis



What we faced

- Relied on default code style / static analysis
- Struggled with code generation



What we faced

- Relied on default code style / static analysis
- Struggled with code generation
- Struggled with test runs for different modules



Case #1

Type system and code analysis

Dart & Type System

The Dart language is type safe: it uses a combination of static type checking and runtime checks to ensure that a variable's value always matches the variable's static type, sometimes referred to as sound typing.

– <https://dart.dev/guides/language/type-system>

But nothing by default

```
void printInts(List<int> a) => print(a);
```

```
void main() {  
    var list = [];  
    list.add(1);  
    list.add('2');  
    printInts(list);  
}
```

NOPE

typecheck – check.dart

typecheck > lib > check.dart

Project 1: typecheck ~/Projects/EPAM/Pearson/Aida English/typecheck

.dart_tool .idea android ios lib check.dart main.dart test widget_test.dart .gitignore .metadata .packages pubspec.lock pubspec.yaml README.md

void printInts(List<int> a) => print(a);

void main() {
 var list = [];
 list.add(1);
 list.add('2');
 printInts(list);
}

Dart Analysis

Description Location ▲ Nothing to show

6: Problems 9: Git Terminal SonarLint Dart Analysis TODO Event Log

Frameworks Detected: Android framework is detected. // Configure (a minute ago)

16 7:19 LF UTF-8 2 spaces master

```
flutter analyze  
Analyzing typecheck...  
No issues found! (ran in 2.1s)
```

Unhandled exception:
type 'List<dynamic>' is not a subtype of type 'List<int>'

```
class HelloTest {  
  
    final interactions;  
  
    HelloTest(this.interactions);  
  
    void addInteraction(interaction) {  
        interactions.add(interaction);  
    }  
}
```

```
class HelloTest {  
  
    final interactions;  
  
    HelloTest(this.interactions);  
  
    void addInteraction(interaction) {  
        interactions.add(interaction);  
    }  
}
```

m **toString()** String
p **hashCode** int
m **noSuchMethod(Invocation invocation)** dynamic
p **runtimeType** Type
nn expr.nn -> if (expr != null) {}
notnull expr.notnull -> if (expr != null) {}
null expr.null -> if (expr == null) {}
par expr.par -> (expr)
return expr.return -> return expr
switch expr.switch -> switch (expr) {}
try stmt.try -> try {stmt} catch (e,s) {}
+newn stmt +new -> new Stmt<on Execution catch (e,s)>

Press ^ to choose the selected (or first) suggestion and insert a dot afterwards [Next Tip](#)

```
class HelloTest {  
  
    final interactions;  
  
    HelloTest(this.interactions);  
  
    void addInteraction(interaction) {  
        interactions.add(interaction);  
    }  
}
```

A code editor showing a Dart class named `HelloTest`. The `addInteraction` method has a warning icon next to it. A context menu is open over the line `interactions.add(interaction);`, displaying the following options:

- Suppress 'always_specify_types' warning
- Convert to async function body
- Add a not-null assertion
- Convert to expression body

At the bottom of the menu, it says "Press ⌘Space to open preview".

Location ▲

[typecheck] lib/check.dart:4
[typecheck] lib/check.dart:4
[typecheck] lib/check.dart:9
[typecheck] lib/check.dart:14
[typecheck] lib/check.dart:18

```
analyzer:  
  strong-mode:  
    implicit-dynamic: false
```

```
void printInts(List<int> a) => print(a);

void main() {
  var list = [];
  list.add(1);
  list.add('2');
  //printInts(list);

  HelloTest([]).addInteraction(123);
}

class HelloTest {

  final interactions;

  HelloTest(this.interactions);

  void addInteraction(int value) {
    interactions.add(value);
  }
}
```

① 4 ▲ 4 ⌂ ⌃

Missing field type for 'interactions'.
Try adding an explicit type like 'dynamic', or enable implicit-dynamic in your analysis options file.

package:typecheck/check.dart
dynamic interactions

Containing class: HelloTest

Type: dynamic

Case #1 - conclusions

Case #1 - conclusions

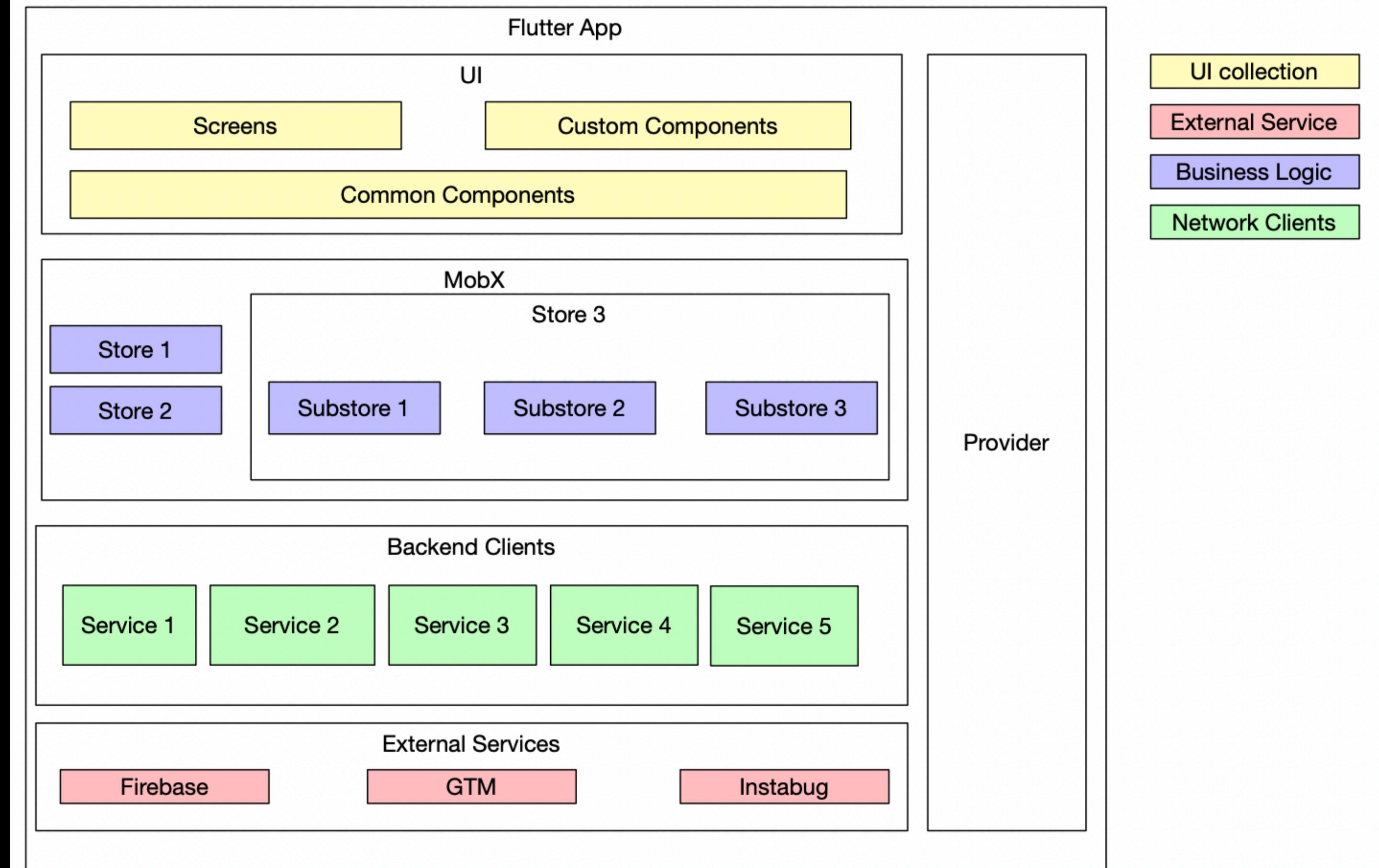
- Use analysis_options.yaml from the beginning

Case #1 - conclusions

- Use analysis_options.yaml from the beginning
- Configure it according your needs

Case #2

Code generation



The problem

5+ services

50+ types for the service

OpenAPI

The OpenAPI Specification (OAS) defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of the service without access to source code, documentation, or through network traffic inspection

-- <https://swagger.io/specification/>

OpenAPI

```
{  
  "/pets": {  
    "get": {  
      "description": "Returns all pets from the system that the user has access to",  
      "responses": {  
        "200": {  
          "description": "A list of pets.",  
          "content": {  
            "application/json": {  
              "schema": {  
                "type": "array",  
                "items": {  
                  "$ref": "#/components/schemas/pet"  
                }  
              }  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

```
components:  
schemas:  
  GeneralError:  
    type: object  
    properties:  
      code:  
        type: integer  
        format: int32  
      message:  
        type: string
```

Code generation with OpenAPI

```
java -jar openapi-generator-cli-4.3.1.jar generate  
-i openapi.yaml  
-g dart-dio  
-o newversion
```

Code generation with OpenAPI

```
java -jar openapi-generator-cli-4.3.1.jar generate  
-i openapi.yaml  
-g dart-dio  
-o newversion
```

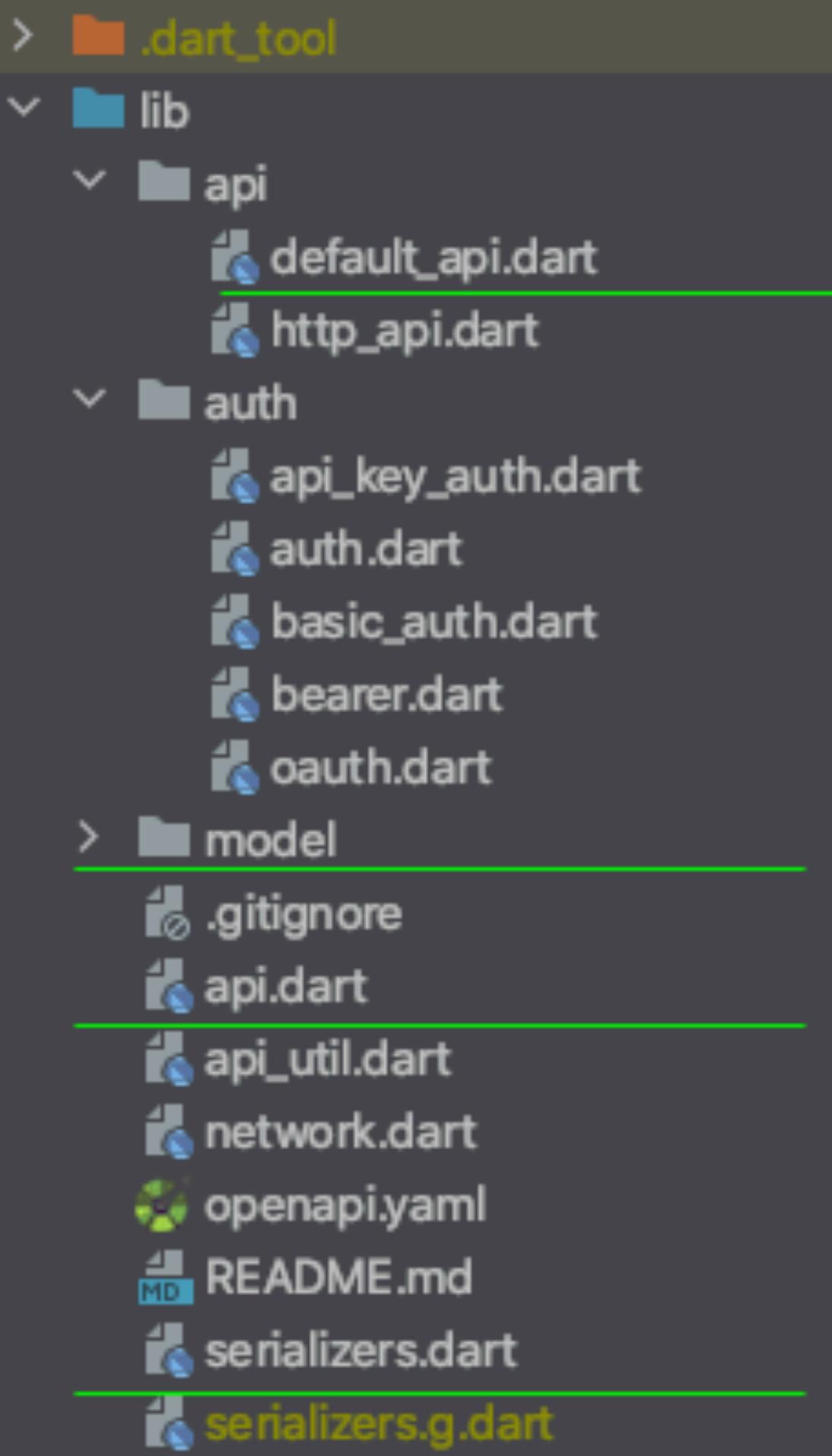
Code generation with OpenAPI

```
java -jar openapi-generator-cli-4.3.1.jar generate  
-i openapi.yaml  
-g dart-dio  
-o newversion
```

Code generation with OpenAPI

```
java -jar openapi-generator-cli-4.3.1.jar generate  
-i openapi.yaml  
-g dart-dio  
-o newversion
```

Generation output



Expectation

Expectation

- single "Make it perfect" button

Reality

Reality

- Buggy generators

Reality

- Buggy generators
- Developers have to patch buggy code

Reality

- Buggy generators
- Developers have to patch buggy code
- Developers have to redo patches after openapi.yaml update



Case #2 - conclusions

Case #2 - conclusions

- Try to avoid codegen

Case #2 - conclusions

- Try to avoid codegen
- Otherwise, automate it

Automating openapiⁱ²

ⁱ² <https://vvsevolodovich.dev/working-with-openapi-in-flutter-fully-automatically/>

Automating openapiⁱ²

- Download jar/install through npm

ⁱ² <https://vvsevolodovich.dev/working-with-openapi-in-flutter-fully-automatically/>

Automating openapiⁱ²

- Download jar/install through npm
- Download openapi.yml

ⁱ² <https://vvsevolodovich.dev/working-with-openapi-in-flutter-fully-automatically/>

Automating openapiⁱ²

- Download jar/install through npm
- Download openapi.yml
- Generate the files

ⁱ² <https://vvsevolodovich.dev/working-with-openapi-in-flutter-fully-automatically/>

Automating openapiⁱ²

- Download jar/install through npm
- Download openapi.yml
- Generate the files
- Patch imports and formatting

² <https://vvsevolodovich.dev/working-with-openapi-in-flutter-fully-automatically/>

Automating openapiⁱ²

- Download jar/install through npm
- Download openapi.yml
- Generate the files
- Patch imports and formatting
- Run build_runner build

² <https://vvsevolodovich.dev/working-with-openapi-in-flutter-fully-automatically/>

Case #3

Tests

Flutter

```
flutter test
```

Dart

```
flutter pub run test ./  
test
```

Tests

Tests

- flutter test doesn't run tests for submodules(in comparison with 'pub get') 

Tests

- flutter test doesn't run tests for submodules(in comparison with 'pub get') 
- Separate code coverage 

Tests

- flutter test doesn't run tests for submodules(in comparison with 'pub get') 
- Separate code coverage 
- Code coverage requires additional steps 

Coverage



Coverage

- Generated as lcov.info



Coverage

- Generated as lcov.info
- Accepted by Github/Gitlab, not the bitrise



Coverage

- Generated as lcov.info
- Accepted by Github/Gitlab, not the bitrise
- Html report shows the coverage



Coverage

- Generated as lcov.info
- Accepted by Github/Gitlab, not the bitrise
- Html report shows the coverage
- Code coverage should not drop(we have ~45% now)



Coverage

- Generated as lcov.info
- Accepted by Github/Gitlab, not the bitrise
- Html report shows the coverage
- Code coverage should not drop(we have ~45% now)
- Coverage doesn't work for branches and functions



LCOV - code coverage report

Current view: top level

Test: lcov.info

Date: 2020-10-08 17:28:04

	Hit	Total	Coverage
Lines:	504	1866	27.0 %
Functions:	0	0	-

Directory	Line Coverage ▾	Functions ▾
	100.0 % 2 / 2	- 0 / 0
	49.8 % 100 / 201	- 0 / 0
	36.4 % 47 / 129	- 0 / 0
	1.4 % 2 / 138	- 0 / 0
	26.1 % 40 / 153	- 0 / 0
	37.3 % 47 / 126	- 0 / 0
	44.8 % 43 / 96	- 0 / 0
	12.6 % 15 / 119	- 0 / 0
	18.4 % 29 / 158	- 0 / 0
	50.0 % 2 / 4	- 0 / 0
	18.0 % 79 / 440	- 0 / 0
	85.7 % 6 / 7	- 0 / 0
	2.4 % 1 / 41	- 0 / 0
	41.9 % 91 / 217	- 0 / 0
	0.0 % 0 / 1	- 0 / 0
	0.0 % 0 / 2	- 0 / 0
	0.0 % 0 / 2	- 0 / 0
	0.0 % 0 / 2	- 0 / 0
	0.0 % 0 / 2	- 0 / 0
	0.0 % 0 / 2	- 0 / 0
	0.0 % 0 / 20	- 0 / 0

How to generate coverage report and check it

```
// install junit
flutter "pub" "global" "activate" "junitreport"
// execute tests and convert the results
flutter "test" "--machine" | tojunit "--output" "./junit_results.xml"
// exclude the files
flutter pub run remove_from_coverage -f coverage/lcov.info -r \
'.g.dart,lib/widgets/*,lib/screens/*'
// Generate report
genhtml -o coverage/html coverage/lcov.info
// Check coverage
flutter pub run check_coverage -f coverage/lcov.info --min-line-coverage 45
```

How to generate coverage report and check it

```
// install junit
flutter "pub" "global" "activate" "junitreport"
// execute tests and convert the results
flutter "test" "--machine" | tojunit "--output" "./junit_results.xml"
// exclude the files
flutter pub run remove_from_coverage -f coverage/lcov.info -r \
  '.g.dart,lib/widgets/*,lib/screens/*'
// Generate report
genhtml -o coverage/html coverage/lcov.info
// Check coverage
flutter pub run check_coverage -f coverage/lcov.info --min-line-coverage 45
```

How to generate coverage report and check it

```
// install junit
flutter "pub" "global" "activate" "junitreport"
// execute tests and convert the results
flutter "test" "--machine" | tojunit "--output" "./junit_results.xml"
// exclude the files
flutter pub run remove_from_coverage -f coverage/lcov.info -r \
  '.g.dart,lib/widgets/*,lib/screens/*'
// Generate report
genhtml -o coverage/html coverage/lcov.info
// Check coverage
flutter pub run check_coverage -f coverage/lcov.info --min-line-coverage 45
```

How to generate coverage report and check it

```
// install junit
flutter "pub" "global" "activate" "junitreport"
// execute tests and convert the results
flutter "test" "--machine" | tojunit "--output" "./junit_results.xml"
// exclude the files
flutter pub run remove_from_coverage -f coverage/lcov.info -r \
  '.g.dart,lib/widgets/*,lib/screens/*'
// Generate report
genhtml -o coverage/html coverage/lcov.info
// Check coverage
flutter pub run check_coverage -f coverage/lcov.info --min-line-coverage 45
```

How to generate coverage report and check it

```
// install junit
flutter "pub" "global" "activate" "junitreport"
// execute tests and convert the results
flutter "test" "--machine" | tojunit "--output" "./junit_results.xml"
// exclude the files
flutter pub run remove_from_coverage -f coverage/lcov.info -r \
  '.g.dart,lib/widgets/*,lib/screens/*'
// Generate report
genhtml -o coverage/html coverage/lcov.info
// Check coverage
flutter pub run check_coverage -f coverage/lcov.info --min-line-coverage 45
```

You need
check_coverage³

³ we wrote it and plan to share

Tightening the feedback loop

Tightening the feedback loop

- CI for tests, code coverage, static analysis and Android/iOS builds 

Tightening the feedback loop

- CI for tests, code coverage, static analysis and Android/iOS builds 
- It takes up to 30 

Git hook

```
./tools/flutter_analyze.sh
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter test
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter format
```

```
./tools/flutter_analyze.sh
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter test
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter format
```

```
./tools/flutter_analyze.sh
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter test
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter format
```

```
./tools/flutter_analyze.sh
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter test
if [ $? -ne 0 ]; then
  exit 1
fi
```

```
flutter format
```

Case #3 - conclusions

Case #3 - conclusions

- Check code coverage

Case #3 - conclusions

- Check code coverage
- Check the coverage is not dropping

Case #3 - conclusions

- Check code coverage
- Check the coverage is not dropping
- Use git hooks for faster feedback

Case #4

supporting different envs

Supporting different envs

Supporting different envs

- You have to test the app against different envs

Supporting different envs

- You have to test the app against different envs
- Require different configs for analytics/crash reporting

Expectations

Flavors will solve the problem

Reality

Configuring flavors is a pain for iOS

Android



Android

- Add flavor to build.gradle



Android

- Add flavor to build.gradle
- Add a folder to src



Android

- Add flavor to build.gradle
- Add a folder to src
- Done



iOS



iOS

- Clone the target



iOS

- Clone the target
- Clone the scheme



iOS

- Clone the target
- Clone the scheme
- Clone configurations



iOS

- Clone the target
- Clone the scheme
- Clone configurations
- Ensure, the scheme is named in Upper-case



iOS

- Clone the target
- Clone the scheme
- Clone configurations
- Ensure, the scheme is named in Upper-case
- Ensure, the scheme is pointing to a proper configuration



iOS

- Clone the target
- Clone the scheme
- Clone configurations
- Ensure, the scheme is named in Upper-case
- Ensure, the scheme is pointing to a proper configuration
- Ensure, the scheme is shared в Runner.xcproject(не воркспейс!)



iOS

- Clone the target
- Clone the scheme
- Clone configurations
- Ensure, the scheme is named in Upper-case
- Ensure, the scheme is pointing to a proper configuration
- Ensure, the scheme is shared в Runner.xcproject(не воркспейс!)
- Ensure, the configuration for all targets bundle id & provisioning profile are set



iOS

- Clone the target
- Clone the scheme
- Clone configurations
- Ensure, the scheme is named in Upper-case
- Ensure, the scheme is pointing to a proper configuration
- Ensure, the scheme is shared в Runner.xcproject(не воркспейс!)
- Ensure, the configuration for all targets bundle id & provisioning profile are set
- Check through flutter build ios -- flavor=<flavor-name>



Case #4 - conclusions

Case #4 - conclusions

- There is --dart-define, use it if you have to change only api url

Case #4 - conclusions

- There is --dart-define, use it if you have to change only api url
- Otherwise flavors, see the checklist

What we knew today

What we knew today

- How to configure code-style & typisation

What we knew today

- How to configure code-style & typisation
- How to work with code generation

What we knew today

- How to configure code-style & typisation
- How to work with code generation
- How to work with tests and code coverage

What we knew today

- How to configure code-style & typisation
- How to work with code generation
- How to work with tests and code coverage
- How to configure different environments

Conclusions

Conclusions

- Flutter is ready for commercial development

Conclusions

- Flutter is ready for commercial development
- It can be adjusted for high quality standards

Conclusions

- Flutter is ready for commercial development
- It can be adjusted for high quality standards
- More info in my twitter and my blog!

vladimir Ivanov



Vladimir Ivanov

- <https://vvsevolodovich.dev>



vladimir Ivanov

- <https://vvsevolodovich.dev> 
- <https://twitter.com/vvsevolodovich> 

