



Debugging

허가받지 않은 복제(복사), 전송, 수정 및 배포를 금합니다.

Debugging Flutter apps


DevTools

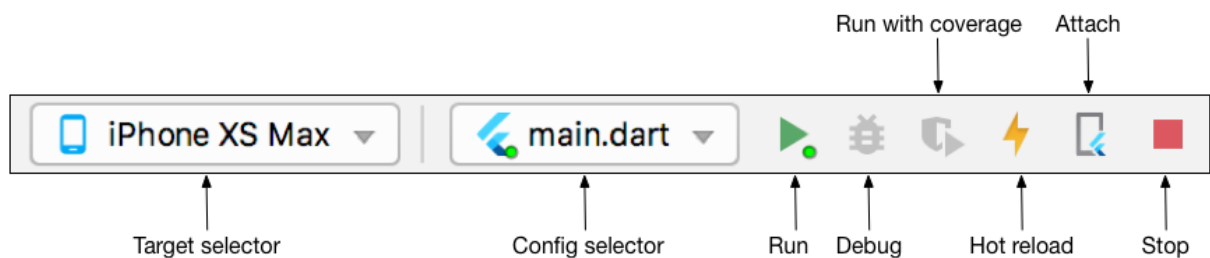
Chapter 10.

Setting breakpoints.

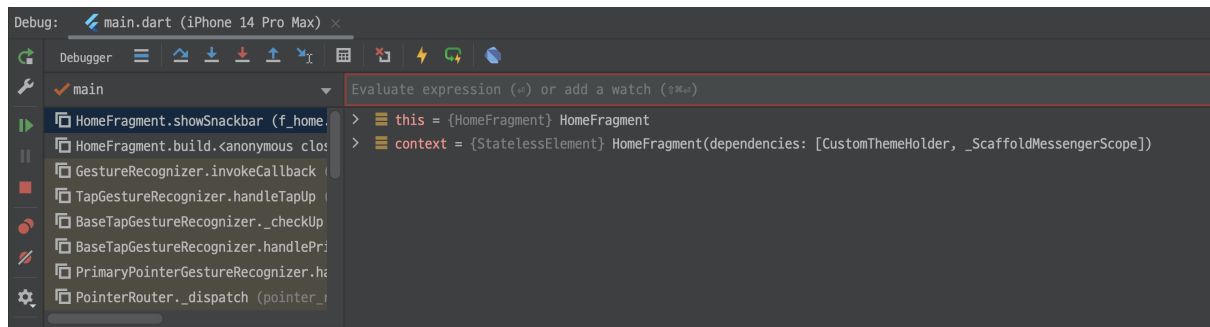
Android Studio and IntelliJ

How to develop Flutter apps in Android Studio or other IntelliJ products.

 <https://docs.flutter.dev/tools/android-studio#running-and-debugging>



- Breakpoint 를 활용하기 위해서는 **Debug** 로 실행시켜야함.



Step Into

- breakpoint 다음 라인으로 이동.
- 다음 라인이 함수일 경우, 함수 내부로 이동

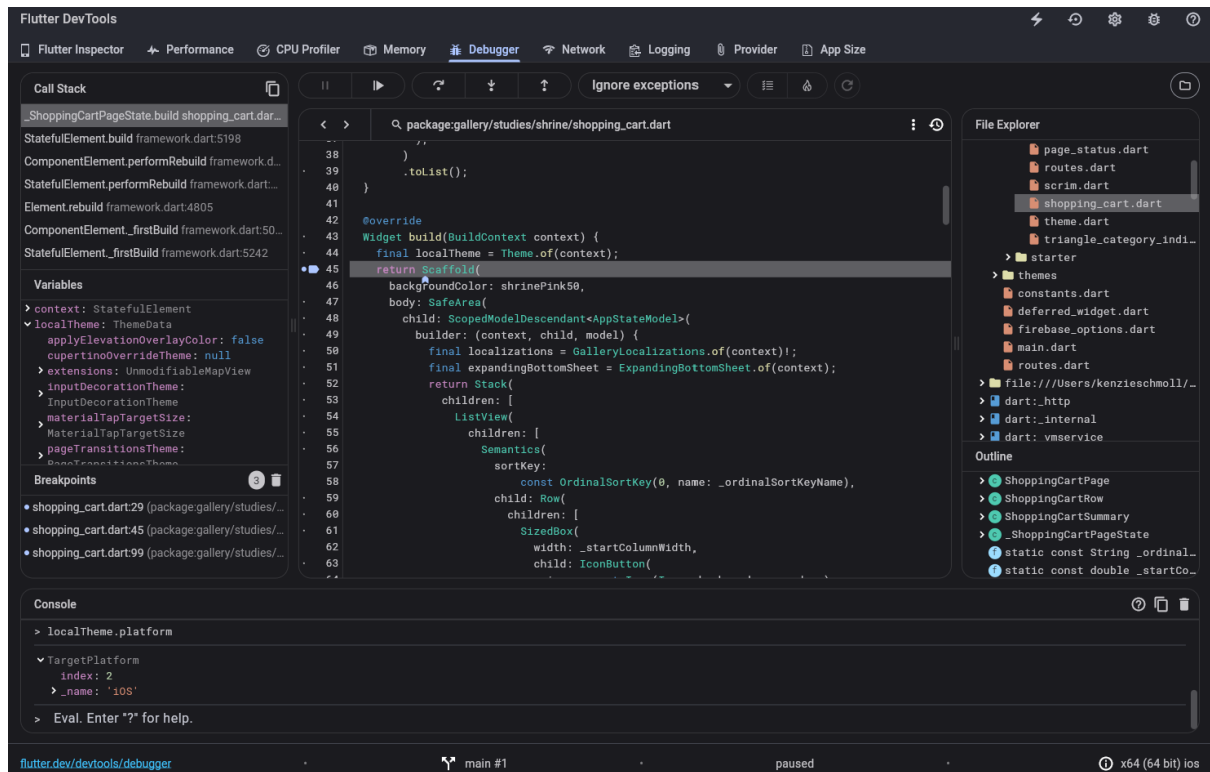
Step Over

- breakpoint 다음 라인으로 이동.
- 다음 라인이 함수일 경우, 실행은 하지만 **함수 내부로 이동하지 않음**

Step Out

- 현재 함수의 리턴으로 이동.

Debugger(DevTools).

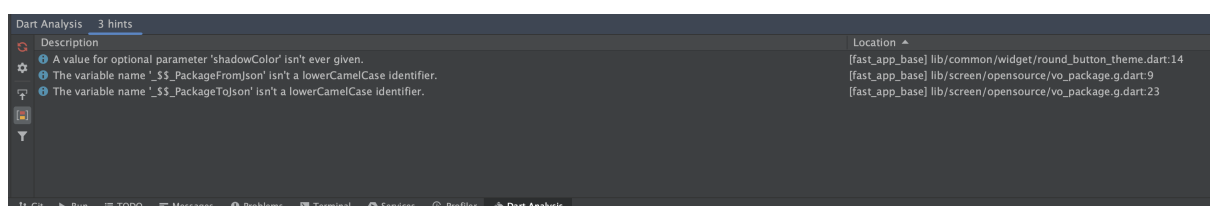


▼ 언제 활용하면 좋을까?

- 중간에 데이터 유실 또는 의도치 않은 데이터 값 변경
- 타이밍 이슈
 - 어떤 로직이 먼저 발생 하는지 정확하게 확인 할 수 있음

👉 간단한 디버깅 또는 빠른 개발을 위해서는 **Logging** 더 효율적일 수 있음.

The Dart analyzer.



```
flutter analyze
```

flutter_lints | Dart Package

Recommended lints for Flutter apps, packages, and plugins to encourage good coding practices.



https://pub.dev/packages/flutter_lints

recommended set of lints for Flutter apps, packages, and plugins to encourage **good coding practices**.

- lint 적용 권장

▼ Why?

- 정적 분석 도구로 실수 방지 및 보완.
- 도구를 활용하여 코딩 컨벤션 적용

```
dart fix --apply
```

Debug mode assertions.

일반적으로 개발 시, debug mode

- bug icon in Android Studio

- `flutter run` at the command line.

`--enable-asserts`

```
assert(color == null || decoration == null,  
  'Cannot provide both a color and a decoration\n'  
  'To provide both, use "decoration: BoxDecoration(color: color)".',  
),
```

Debugging animations.

Flutter inspector provides a **Slow Animations** button.

- devtools 주제에서 다룰 예정

Measuring app startup time.

```
flutter run --trace-startup --profile
```

- profile 모드로 실행 (debug / release / profile)

👉 `build/start_up_info.json`

```
{  
  "engineEnterTimestampMicros": 96025565262,  
  "timeToFirstFrameMicros": 2171978,  
  "timeToFrameworkInitMicros": 514585,  
  "timeAfterFrameworkInitMicros": 1657393  
}
```

logging.

```
import 'package:logging/logging.dart';

final log = Logger('ExampleLogger');

/// Example of configuring a logger to print to stdout.
///
/// This example will print:
///
/// INFO: 2021-09-13 15:35:10.703401: recursion: n = 4
/// INFO: 2021-09-13 15:35:10.707974: recursion: n = 3
/// Fibonacci(4) is: 3
/// Fibonacci(5) is: 5
/// SHOUT: 2021-09-13 15:35:10.708087: Unexpected negative n: -42
/// Fibonacci(-42) is: 1
void main() {
  Logger.root.level = Level.ALL; // defaults to Level.INFO
  Logger.root.onRecord.listen((record) {
    print('${record.level.name}: ${record.time}: ${record.message}');
  });

  print('Fibonacci(4) is: ${fibonacci(4)}');

  Logger.root.level = Level.SEVERE; // skip logs less than severe.
  print('Fibonacci(5) is: ${fibonacci(5)}');

  print('Fibonacci(-42) is: ${fibonacci(-42)}');
}

int fibonacci(int n) {
  if (n <= 2) {
    if (n < 0) log.shout('Unexpected negative n: $n');
    return 1;
  } else {
    log.info('recursion: n = $n');
    return fibonacci(n - 2) + fibonacci(n - 1);
  }
}
```


개발용, 디버깅용이 아닌 유지보수를 위한 로그 시스템

Firestore Crashlytics.

a lightweight, realtime **crash reporter** that helps you **track, prioritize, and fix** stability issues that erode your app quality.

firebase_crashlytics | Flutter Package

Flutter plugin for Firebase Crashlytics. It reports uncaught errors to the Firebase console.

 https://pub.dev/packages/firebase_crashlytics



Firebase Analytics.

an app **measurement solution** that provides **insight** on **app usage and user engagement** on Android and iOS.

firebase_analytics | Flutter Package

Flutter plugin for Google Analytics for Firebase, an app measurement solution that provides insight on app usage and user engagement on Android and iOS.

 https://pub.dev/packages/firebase_analytics



Sentry.

We support 100+ languages & frameworks, from **frontend to backend, mobile, gaming, & more.** Developer-first application **monitoring platform** built to give you answers - not clues.

sentry | Dart Package

A crash reporting library for Dart that sends crash reports to Sentry.io. This library supports Dart VM and Web. For Flutter consider sentry_flutter instead.

 <https://pub.dev/packages/sentry>



logger.

```
import 'package:logger/logger.dart';

var logger = Logger(
  printer: PrettyPrinter(),
);

var loggerNoStack = Logger(
  printer: PrettyPrinter(methodCount: 0),
);

void main() {
  print(
    'Run with either `dart example/main.dart` or `dart --enable-asserts example/main.dart`.');
  demo();
}

void demo() {
  logger.d('Log message with 2 methods');

  loggerNoStack.i('Info message');

  loggerNoStack.w('Just a warning!');

  logger.e('Error! Something bad happened', 'Test Error');

  loggerNoStack.v({'key': 5, 'value': 'something'});

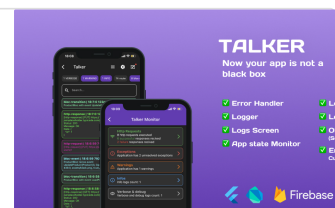
  Logger(printer: SimplePrinter(colors: true)).v('boom');
}
```

talker.

GitHub - Frezyx/talker: 📞 Advanced error handler and logger for dart and flutter apps

📞 Advanced error handler and logger for dart and flutter apps - GitHub - Frezyx/talker: 📞
Advanced error handler and logger for dart and flutter apps

🌐 <https://github.com/Frezyx/talker/tree/master>



Network Log

▼ Why?

개발을 하면서 응답값이 어떻게 나오는지 바로 확인하면서 개발.

pretty_dio_logger | Dart Package

Pretty Dio logger is a Dio interceptor that logs network calls in a pretty, easy to read format.

 https://pub.dev/packages/pretty_dio_logger



Network Log Console(Inspector).

▼ Why?

서버 개발자와의 커뮤니케이션 비용 절감

- “API 파라미터는 어떻게 호출하고 계시나요?”
- “홈 화면 접속 시, 어떤 API들 호출 하나요?”
- “API 실제 사용 체크 여부”
- ...

Alice.

alice | Flutter Package


Alice is an HTTP Inspector tool which helps debugging http requests. It catches and stores http requests and responses, which can be viewed via simple UI.

 <https://pub.dev/packages/alice>



Supported.

- Dio
- HttpClient from dart:io package
- Http from http/http package
- Chopper
- Generic HTTP client

 **주의!** 일반 유저한테 진입점을 노출하면 안돼요!

참고.

Debugging Flutter apps

How to debug your app using the DevTools suite.

 <https://docs.flutter.dev/testing/debugging>



Debugging Flutter apps programmatically

How to enable various debugging tools from your code and at the command line.

 <https://docs.flutter.dev/testing/code-debugging#logging>



Using an OEM debugger

How to connect an OEM debugger to your running Flutter app.

 <https://docs.flutter.dev/testing/oem-debuggers>

