Get started

Samples & tutorials

Development

- User interface
- Data & backend
- Accessibility & internationalization
- Platform integration
- Packages & plugins
- Add Flutter to existing app
  - Introduction
  - Adding to an Android app
  - Adding to an iOS app
  - Running, debugging & hot reload
  - Loading sequence and performance
- Tools & techniques
- Migration notes

Testing & debugging

Performance & optimization

Deployment

Resources

Reference

- Widget index
- API reference
- Package site

Running, debugging, and hot reload

Docs > Development > Add Flutter to existing app > Debugging

Contents

Running, debugging, and hot reload

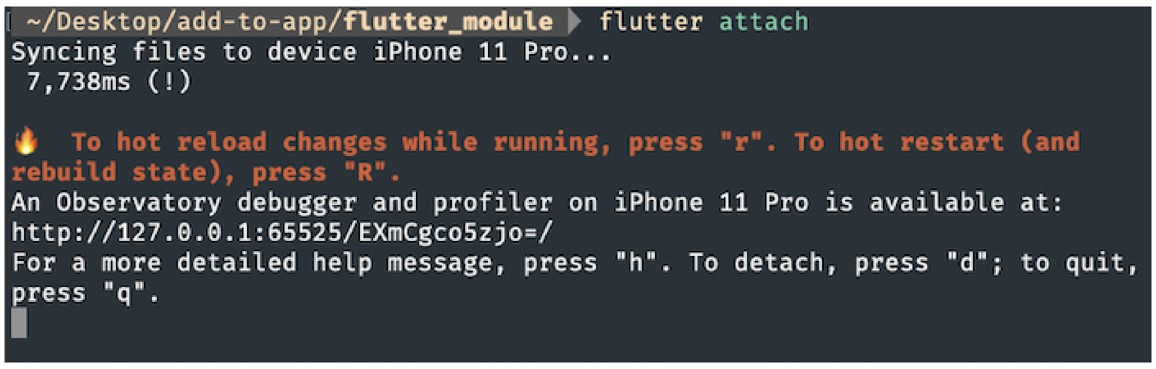
Once you've integrated the Flutter module to your project and used Flutter's platform APIs to run the Flutter engine and/or UI, you then build and run your Android or iOS app the same way you run normal Android or iOS apps.

However, Flutter is now powering the UI in places where you're showing a `FlutterActivity` or `FlutterViewController`.

Debugging

You may be used to having your suite of favorite Flutter debugging tools available to you automatically when running `flutter ru` an equivalent command from an IDE. But you can also use all your Flutter debugging functionalities such as hot reload, performa overlays, DevTools, and setting breakpoints in add-to-app scenarios.

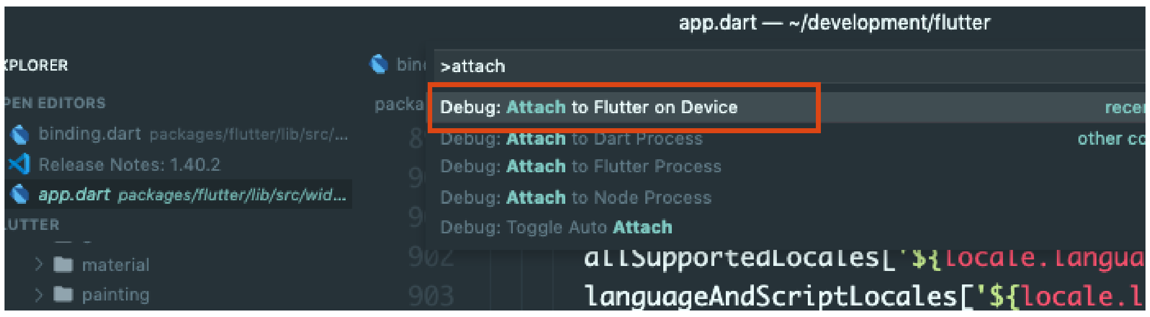
These functionalities are provided by the `flutter attach` mechanism. `flutter attach` can be initiated through different pathwa such as through the SDK's CLI tools, through VSCode or IntelliJ/Android Studio.



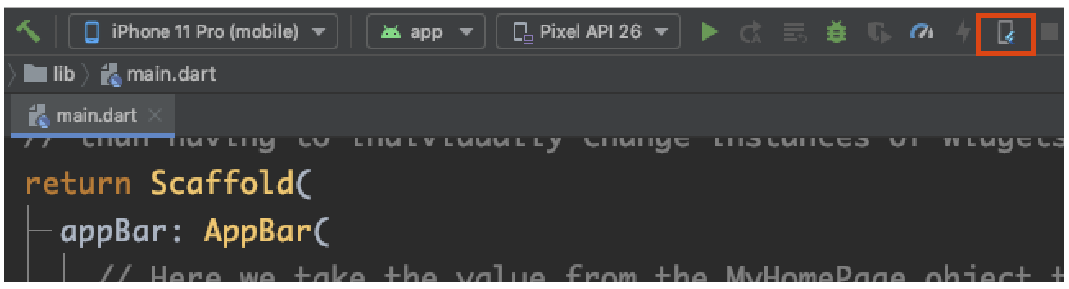
```
 ~/Desktop/add-to-app/flutter_module  flutter attach
Syncing files to device iPhone 11 Pro...
 7,738ms (!)

🔥  To hot reload changes while running, press "r". To hot restart (and
rebuild state), press "R".
An Observatory debugger and profiler on iPhone 11 Pro is available at:
http://127.0.0.1:65525/EXmCgco5zjo=/
For a more detailed help message, press "h". To detach, press "d"; to quit,
press "q".
```

flutter attach via terminal


app.dart — ~/development/flutter
XPLORER
PEN EDITORS
binding.dart packages/flutter/lib/src/...
Release Notes: 1.40.2
app.dart packages/flutter/lib/src/wid...
LUTTER
material
painting
>attach
Debug: Attach to Flutter on Device
recen
Debug: Attach to Dart Process
other co
Debug: Attach to Flutter Process
Debug: Attach to Node Process
Debug: Toggle Auto Attach
allSupportedLocales['${locale.langua
languageAndScriptLocales['${locale.l

flutter attach via VSCode



flutter attach via IntelliJ

`flutter attach` can connect as soon as you run your `FlutterEngine`, and remains attached until your `FlutterEngine` is dispose But you can invoke `flutter attach` before starting your engine. `flutter attach` waits for the next available Dart VM that is hos by your engine.

In IntelliJ or VSCode, you should select the device on which the Flutter module runs so `flutter attach` filters for the right start signals.

Debugging specific instances of Flutter

It's possible to add multiple instances of Flutter (`root isolates`) to an app. `flutter attach` connects to all of the available isola by default. Any commands sent from the attached CLI are then forwarded to each of the attached isolates.

You can list all the attached isolates by typing `l` from an attached `flutter` CLI tool. If unspecified, then the isolate names are automatically generated from the dart entrypoint file and function name.

Example `l` output for an application that is displaying two Flutter isolates simultaneously:

```
Connected views:
  main.dart$main-517591213 (isolates/517591213)
  main.dart$main-332962855 (isolates/332962855)
```

In order to attach to specific isolates instead, do the following:

1. Name the Flutter root isolate of interest in its Dart source.

```
// main.dart
import 'dart:ui' as ui;

void main() {
  ui.window.setIsolateDebugName("debug isolate");
  // ...
}
```

2. Run `flutter attach` with the `--isolate-filter` option.

```
$ flutter attach --isolate-filter='debug'
Waiting for a connection from Flutter...
Done.
Syncing files to device...      1.1s

🔥  To hot reload changes while running, press "r".
To hot restart (and rebuild state), press "R".
An Observatory debugger and profiler is available
at: http://127.0.0.1:43343/.
For a more detailed help message,
press "h". To detach, press "d"; to quit, press "q".

Connected view:
  debug isolate (isolates/642101161)
```

Get started
Samples & tutorials
Development
- User interface
- Data & backend
- Accessibility & internationalization
- Platform integration
- Packages & plugins
- Add Flutter to existing app
  - Introduction
  - Adding to an Android app
  - Adding to an iOS app
  - Running, debugging & hot reload
  - Loading sequence and performance
- Tools & techniques
- Migration notes

Testing & debugging
Performance & optimization
Deployment
Resources
Reference