

[Get started](#)

[Samples & tutorials](#)

[Development](#)

▶ [User interface](#)

▶ [Data & backend](#)

▶ [Accessibility & internationalization](#)

▶ [Platform integration](#)

▶ [Packages & plugins](#)

▶ [Add Flutter to existing app](#)

▼ [Tools & techniques](#)

[Android Studio & IntelliJ](#)

[Visual Studio Code](#)

▼ [DevTools](#)

[Overview](#)

[Install from Android Studio & IntelliJ](#)

[Install from VS Code](#)

[Install from command line](#)

[Flutter inspector](#)

[Timeline view](#)

[Memory view](#)

[Performance view](#)

[Debugger](#)

[Logging view](#)

▶ [Flutter SDK](#)

[Hot reload](#)

[Code formatting](#)

▶ [Migration notes](#)

[Testing & debugging](#)

[Performance & optimization](#)

[Deployment](#)

[Resources](#)

[Reference](#)

[Widget index](#)

[API reference](#)

[Package site](#)

# Install and run DevTools from the command line

[Docs](#) > [Development](#) > [Tools](#) > [DevTools](#) > [Install and run DevTools from the command line](#)

## Contents

- [Install DevTools](#)
- [Launch the DevTools application server](#)
- [Start an application to debug](#)
- [Open DevTools and connect to the target app](#)

## Install DevTools

If you have `pub` on your path, you can run:

```
pub global activate devtools
```

If you have `flutter` on your path, you can run:

```
flutter pub global activate devtools
```

That command installs (or updates) DevTools on your machine.

## Launch the DevTools application server

Next, run the local web server, which serves the DevTools application itself. To do that, run one of the following two commands:

```
pub global run devtools    # If you have `pub` on your path.
```

OR

```
flutter pub global run devtools    # If you have `flutter` on your path.
```

On the command line, you should see output that looks something like:

```
Serving DevTools at http://127.0.0.1:9100
```

## Start an application to debug

Next, start an app to connect to. This can be either a Flutter application or a Dart command-line application. The command below uses a Flutter app:

```
cd path/to/flutter/app
flutter run
```

You need to have a device connected, or a simulator open, for `flutter run` to work. Once the app starts, you'll see a message in terminal that looks like:

```
An Observatory debugger and profiler on iPhone X is available
at: http://127.0.0.1:50976/Swm0bjIe0ks=/
```

Keep note this URL, as you will use it to connect your app to DevTools.

## Open DevTools and connect to the target app

Once it's set up, using DevTools is as simple as opening a Chrome browser window and navigating to `http://localhost:9100`.

Once DevTools opens, you should see a connect dialog:

## Connect

## Connect to a running app

Enter a port or URL to a running Dart or Flutter application.

Port or URL

Connect

Paste the URL you got from running your app (in this example, <http://127.0.0.1:50976/Swm0bjIe0ks/>) into the connect dialog, and then click **Connect** to connect your app to DevTools.

This URL contains a security token, so it will be different for each run of your app. This means that if you stop your application and run it, you need to connect to DevTools with the new URL.

[flutter-dev@](#) • [terms](#) • [security](#) • [privacy](#) • [español](#) • [社区中文资源](#)

Except as otherwise noted, this work is licensed under a Creative Commons Attribution 4.0 International License, and code samples are licensed under the BSD License.

## Overview

## Install from Android Studio & IntelliJ

## Install from VS Code

### Install from command line

## Flutter inspector

### Timeline view

Memory view

### Performance view

## Debugger

Logging view

- Flutter SDK

### Hot reload

## Code formatting

► Migration notes

## Testing & debugging

## Performance & optimization

## Deployment

## Resources

## Reference

## Widget index

[API reference](#) 

[Package site](#) 