

Test drive

[Docs](#) [Get started](#) Test drive

This page describes how to create a new Flutter app from templates, run it, and experience “hot reload” after you make changes to the app.

Select your development tool of choice for writing, building, and running Flutter apps.

Android Studio and IntelliJ

Visual Studio Code

Terminal & editor

Create the app

Do you want to run your Flutter app on the web? The web version of Flutter is available on the beta channel. To try it out, check out the [Write your first Flutter app for the web](#) codelab.

1. Open the IDE and select **Start a new Flutter project**.
2. Select **Flutter Application** as the project type. Then click **Next**.
3. Verify the Flutter SDK path specifies the SDK's location (select **Install SDK...** if the text field is blank).
4. Enter a project name (for example, `myapp`). Then click **Next**.
5. Click **Finish**.
6. Wait for Android Studio to install the SDK and create the project.

Note: When creating a new Flutter app, some Flutter IDE plugins ask for a company domain name in reverse order, something like `com.example`. The company domain name and project name are used together as the package name for Android (the Bundle ID for iOS) when the app is released. If you think that the app might be released, it's better to

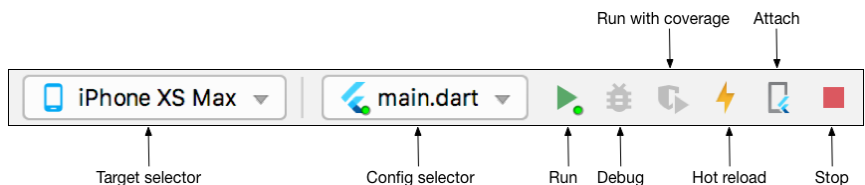
specify the package name now. The package name can't be changed once the app is released, so make the name unique.

The above commands create a Flutter project directory called `myapp` that contains a simple demo app that uses [Material Components](#).

Tip: The code for your app is in `lib/main.dart`. For a high-level description of what each code block does, see the comments at the top of that file.

Run the app

1. Locate the main Android Studio toolbar:



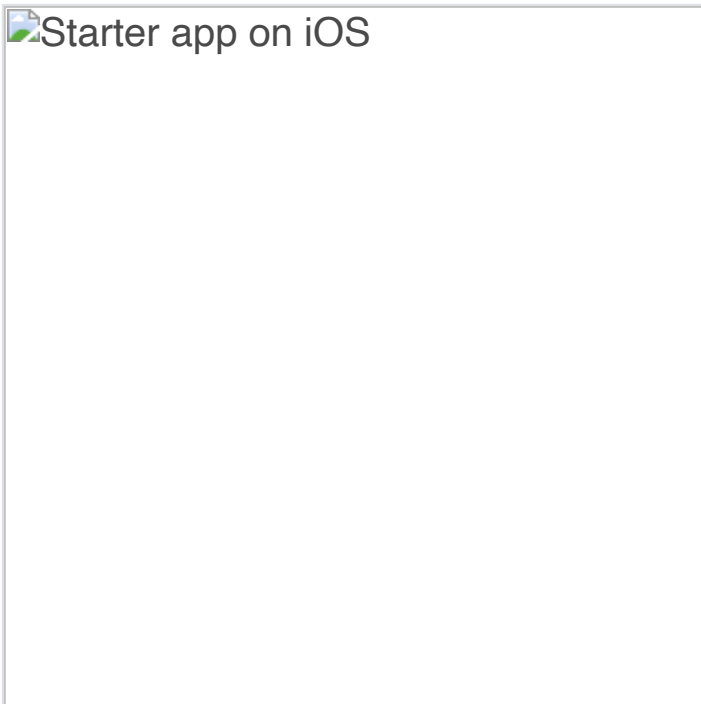
2. In the **target selector**, select an Android device for running the app. If none are listed as available, select **Tools > Android > AVD**

Manager and create one there. For details, see [Managing AVDs](#).

3. Click the run icon in the toolbar, or invoke the menu item **Run > Run**.

Warning: When launching your app from a Mac, if you see `ERROR: Could not connect to lockdownd, error code -17`, make sure that you have [trusted your computer](#).

After the app build completes, you'll see the starter app on your device.



Try hot reload

Flutter offers a fast development cycle with *Stateful Hot Reload*, the ability to reload the code of a live running app without restarting or losing app state. Make a change to app source, tell your IDE or command-line tool that you want to hot reload, and see the change in your simulator, emulator, or device.

1. Open `lib/main.dart`.
2. Change the string

```
'You have pushed the button this  
many times'
```

to

```
'You have clicked the button this  
many times'
```

Important: Do *not* stop your app. Let your app run.

3. Save your changes: invoke **Save All**, or click **Hot Reload** `offline_bolt`.

You'll see the updated string in the running app almost immediately.

Profile or release runs

Important: Do *not* test the performance of your app with debug and hot reload enabled.

So far you've been running your app in *debug* mode. Debug mode trades performance for useful developer features such as hot reload and step debugging. It's not unexpected to see slow performance and janky animations in debug mode.

Once you are ready to analyze performance or release your app, you'll want to use Flutter's "profile" or "release" build modes. For more details, see [Flutter's build modes](#).

Important: If you're concerned about the package size of your app, see [Measuring your app's size](#).