

Build and release a web app

Contents

[Handling images on the web](#)

[Choosing a web renderer](#)

[Minification](#)

[Building the app for release](#)

[Embedding a Flutter app into an HTML page](#)

[Deploying to the web](#)

[PWA Support](#)

During a typical development cycle, you test an app using `flutter run -d chrome` (for example) at the command line. This builds a *debug* version of your app.

This page helps you prepare a *release* version of your app and covers the following topics:

- [Handling images on the web](#)
- [Choosing a web renderer](#)
- [Minification](#)
- [Building the app for release](#)
- [Deploying to the web](#)

Handling images on the web

The web supports the standard `Image` widget to display images. However, because web browsers are built to run untrusted code safely, there are certain limitations in what you can do with images compared to mobile and desktop platforms.

For more information, see [Displaying images on the web](#).

Choosing a web renderer

By default, the `flutter build` and `flutter run` commands use the `auto` choice for the web renderer. This means that your app runs with the HTML renderer on mobile browsers and CanvasKit on desktop browsers. This is our recommended combination to optimize for the characteristics of each platform.

For more information, see [Web renderers](#).

Minification

Minification is handled for you when you create a release build.

A debug build of a web app is not minified and tree shaking has not been performed.

A profile build is not minified and tree shaking has been performed.

A release build is both minified and tree shaking has been performed.

Building the app for release

Build the app for deployment using the `flutter build web` command. You can also choose which renderer to use by using the `--web-renderer` option (See [Web renderers](#)). This generates the app, including the assets, and places the files into the `/build/web` directory of the project.

The release build of a simple app has the following structure:

```
/build/web
  assets
    AssetManifest.json
    FontManifest.json
    NOTICES
  fonts
    MaterialIcons-Regular.ttf
    <other font files>
  <image files>
  index.html
  main.dart.js
  main.dart.js.map
```

content_copy

Launch a web server (for example, `python -m http.server 8000`, or by using the [dhttpd](#) package), and open the `/build/web` directory. Navigate to `localhost:8000` in your browser (given the python SimpleHTTPServer example) to view the release version of your app.

Embedding a Flutter app into an HTML page

You can embed a Flutter web app, as you would embed other content, in an [iframe](#) tag of an HTML file. In the following example, replace “URL” with the location of your HTML page:

Deploying to the web

When you are ready to deploy your app, upload the release bundle to Firebase, the cloud, or a similar service. Here are a few possibilities, but there are many others:

- [Firebase Hosting](#)
- [GitHub Pages](#)
- [Google Cloud Hosting](#)

PWA Support

As of release 1.20, the Flutter template for web apps includes support for the core features needed for an installable, offline-capable PWA app. Flutter-based PWAs can be installed in the same way as any other web-based PWA; the settings signaling that your Flutter app is a PWA are provided by `manifest.json`, which is produced by `flutter create` in the `web` directory.

PWA support remains a work in progress, so please [give us feedback](#) if you see something that doesn't look right.