Build and release a web app

Contents

- [Minification](#)
- [Building the app for release](#)
- [Embedding a Flutter app into an HTML page](#)
- [Deploying to the web](#)

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

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

- [Minification](#)
- [Building the app for release](#)
- [Deploying to the web](#)

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. This generates the app, including the assets, and places t 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
    LICENSE
    fonts
      MaterialIcons-Regular.ttf
      <other font files>
    <image files>
  index.html
  main.dart.js
  main.dart.js.map
```

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

```
<iframe src="URL"></iframe>
```

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](#)

In future, we plan to generate PWA configuration files to support Progressive Web Apps.

▶ User interface

▶ Data & backend

▶ Accessibility & internationalization

▶ Platform integration

▶ Packages & plugins

▶ Add Flutter to existing app

▶ Tools & techniques

▶ Migration notes

Testing & debugging

Performance & optimization

Deployment

  Obfuscating Dart code

  Creating flavors for Flutter

  Build and release an Android app

  Build and release an iOS app

  Build and release a web app

  Continuous deployment

Resources

Reference

  Widget index

  API reference ⧉

  Package site ⧉