Use Firebase to host your Flutter app on the web

Anthony Robledo
Nov 12, 2019 · 5 min read

So you have a Flutter app, and you want to show it to the world? Look no further! This tutorial shows how to host your Flutter app on the web through a Firebase-hosted URL in just 4 easy steps (2 steps if you already have firebase-tools installed and an existing Firebase project). It is assumed that you have Flutter installed and are able to run your Flutter app locally. If you are new to Flutter, first check out the getting started guide.

Prerequisite

If your project was created before flutter web was enabled, your project needs a copy of the web folder with an `index.html` file. If there is not a web folder in your root directory, run the following command to make one:

```
$ flutter create .
```

If you create a project after enabling the web, it automatically creates the `web/index.html` file for you. Good, now on to the Firebase Hosting setup.

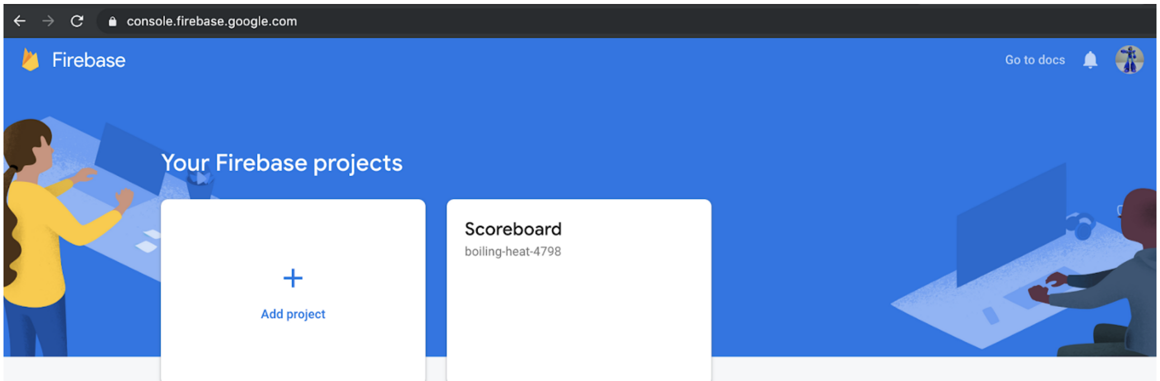
☝️Note: Step 1 and 2 can be skipped if you already have a Firebase project and firebase-tools installed.

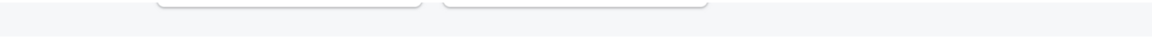
Step 1: Create the Firebase project

🏃 Skip this step if you already have a Firebase project to use.

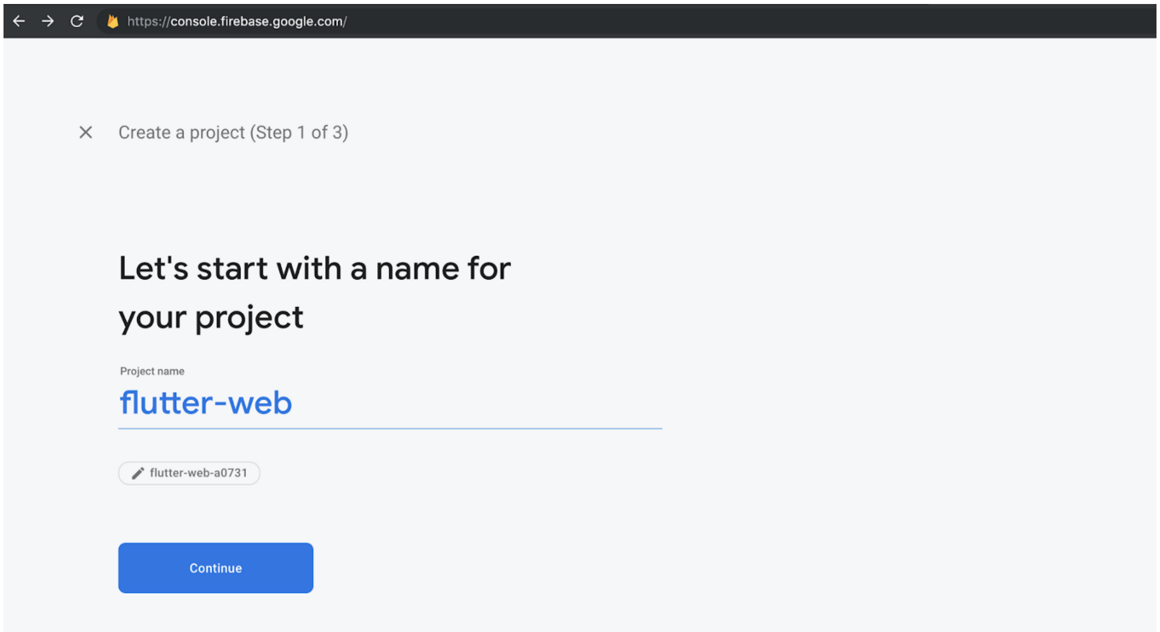
☝️Note: A new project can also be created on the command line from within step 2, but this tutorial creates it in the Firebase console.

In a browser, go to the Firebase console.





Click **Add project** and give your project a name:



https://console.firebase.google.com/

Create a project (Step 1 of 3)

Let's start with a name for your project

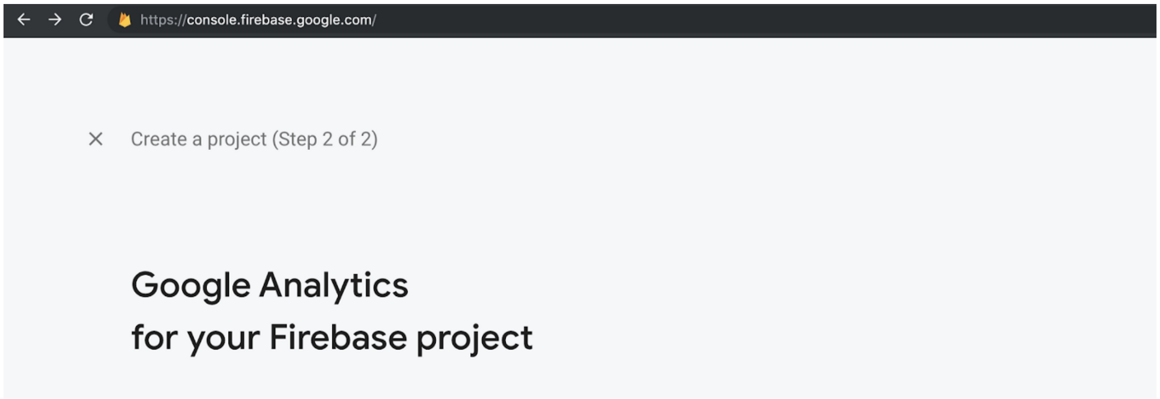
Project name

flutter-web

flutter-web-a0731

☝Note: The actual project name might generate some extra characters at the end for uniqueness.

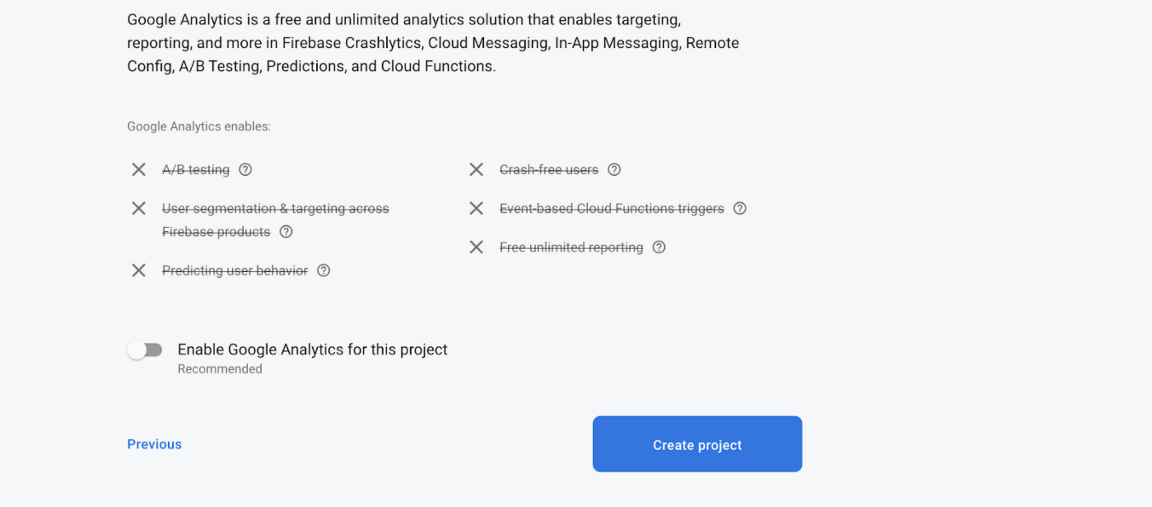
Click **Continue.** For simplicity, don’t enable Google Analytics for this project, so turn off the switch and click **Create project**. If you want to use Google Analytics, leave the switch on, and there will be one extra step of choosing or creating a Google Analytics project.



https://console.firebase.google.com/

Create a project (Step 2 of 2)

Google Analytics for your Firebase project



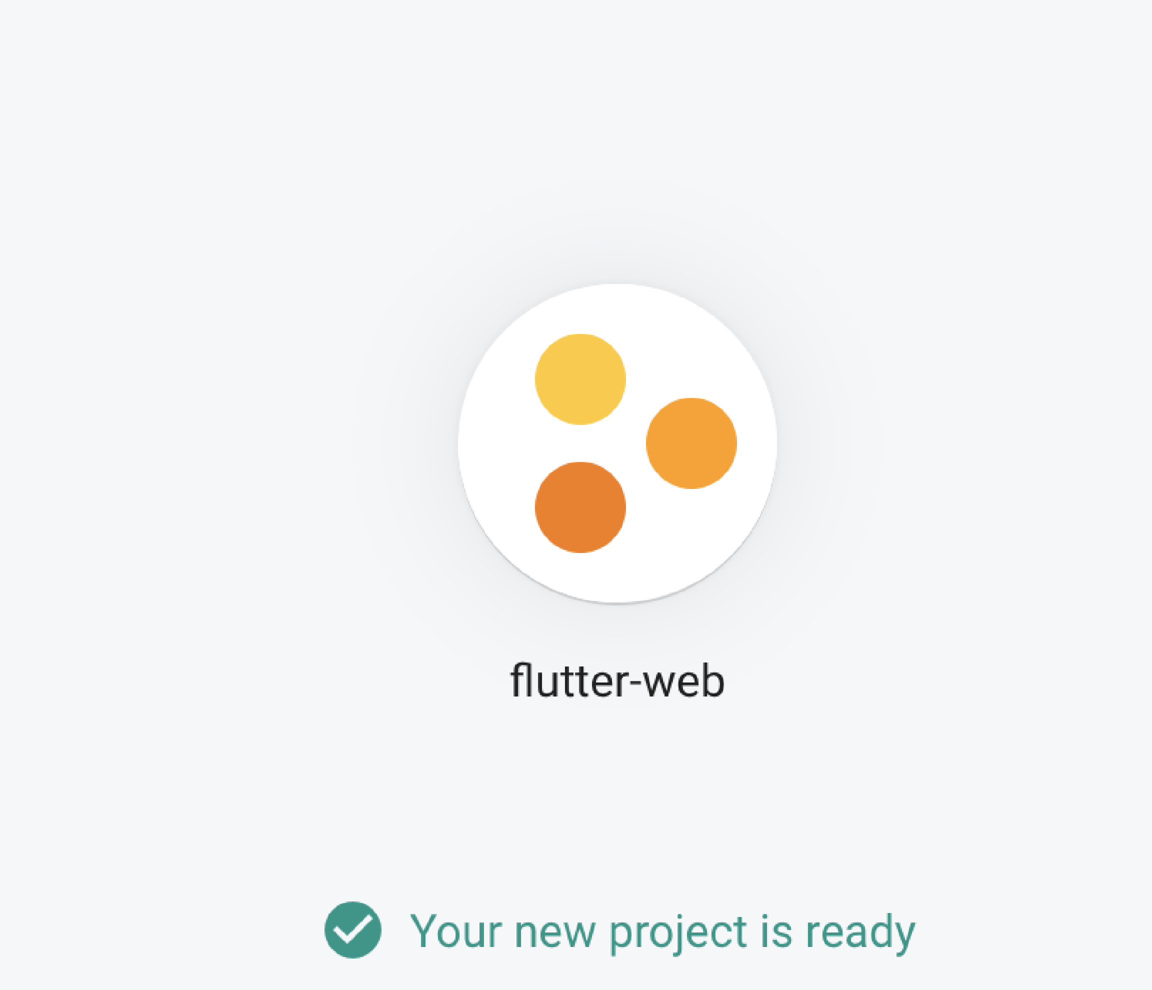
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

- ✕ ~~A/B testing~~
- ✕ ~~User segmentation & targeting across Firebase products~~
- ✕ ~~Predicting user behavior~~
- ✕ ~~Crash-free users~~
- ✕ ~~Event-based Cloud Functions triggers~~
- ✕ ~~Free unlimited reporting~~

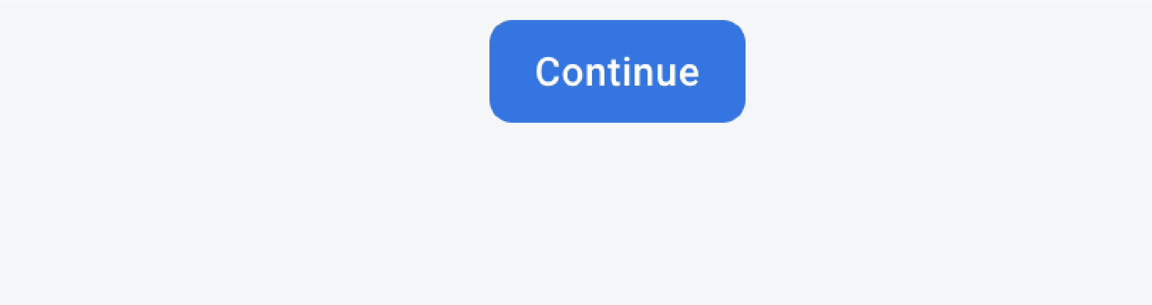
○ Enable Google Analytics for this project
Recommended

Watch for a few seconds as the project is created, until you see that it is ready and click **Continue.**



flutter-web

✓ Your new project is ready



That’s it for step 1!

Step 2: Install the firebase-tools CLI

🏃‍♀️Skip this step if you already have firebase-tools installed.

If you do not have `npm` , you must install that first.

☝️Note: If you are not sure whether `npm` is installed on your machine, run `$ npm -v` , and see if it lists a version number. If it does, then you already have `npm` . If it says “command not found”, you need to install it.

Installing npm on windows.

Installing npm on linux.

Installing npm on mac.

Or download it directly from Node.js.

Once `npm` is installed, run the following command to install the `firebase-tools` :

```
$ npm install -g firebase-tools
```

`firebase-tools` is now installed!

Step 3: Initialize Firebase hosting for your Flutter project

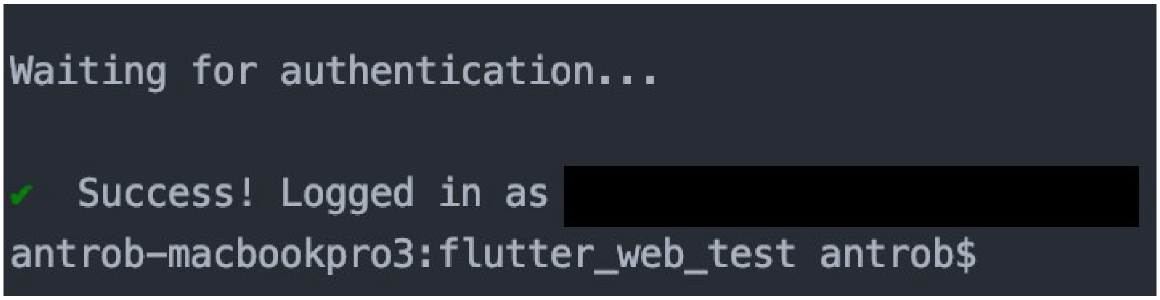
Open a terminal and navigate to the root directory for your Flutter app and run this command to login:

```
$ firebase login
```

☝️Note: If you are already logged in and want to re-login as different account, first logout with:

```
$ firebase logout
```

Follow the command prompt and the link in your browser. Accept the permissions by clicking **Ok**. When you return to your terminal you should see that you are now logged in:

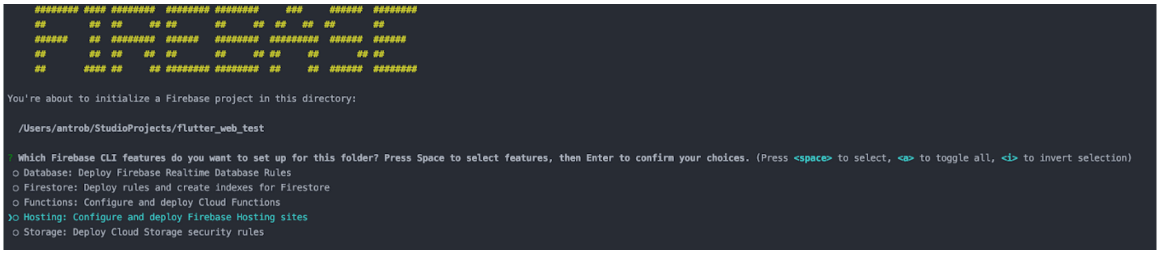


```
Waiting for authentication...

✔  Success! Logged in as [redacted]
antrob-macbookpro3:flutter_web_test antrob$
```

Next, run this command from the root directory of your Flutter app to initialize your Firebase project:

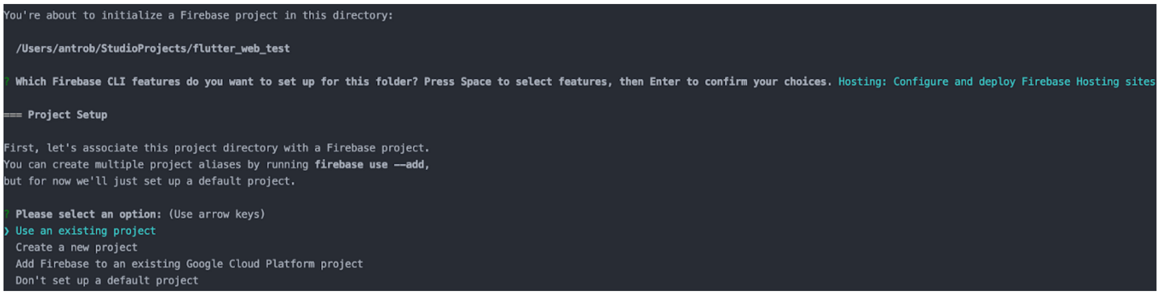
```
$ firebase init
```

```
antrob-macbookpro3:flutter_web_test antrob$ firebase init
```



```
     ######## #### ########  ######## ########     ###     ######  ########
     ##        ##  ##     ## ##       ##     ##  ##   ##  ##       ##
     ######    ##  ########  ######   ########  #########  ######  ######
     ##        ##  ##    ##  ##       ##     ## ##     ##       ## ##
     ##       #### ##     ## ######## ########  ##     ##  ######  ########

You're about to initialize a Firebase project in this directory:

  /Users/antrob/StudioProjects/flutter_web_test

? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection)
 ○ Database: Deploy Firebase Realtime Database Rules
 ○ Firestore: Deploy rules and create indexes for Firestore
 ○ Functions: Configure and deploy Cloud Functions
❯○ Hosting: Configure and deploy Firebase Hosting sites
 ○ Storage: Deploy Cloud Storage security rules
```

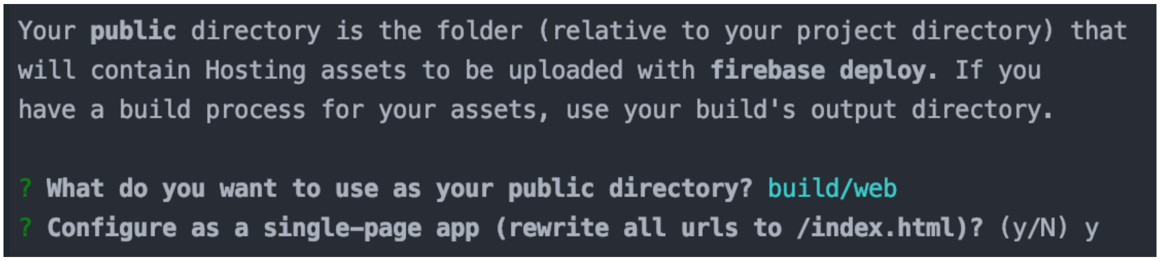
Use the arrow keys to navigate the cursor to **Hosting** and hit the spacebar to select it, then press enter. Now you will see this screen:



```
You're about to initialize a Firebase project in this directory:

  /Users/antrob/StudioProjects/flutter_web_test

? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices. Hosting: Configure and deploy Firebase Hosting sites

=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Please select an option: (Use arrow keys)
❯ Use an existing project
  Create a new project
  Add Firebase to an existing Google Cloud Platform project
  Don't set up a default project
```

Select **Use an existing project** by pressing Enter. Use the arrow keys to select the project you made in step 1.

Next, enter `build/web` as the public directory and press enter, then enter **y** (for yes) to select the single page app option:



```
Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? build/web
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N) y
```

☝️Note: After this step, 2 new files are created (`.firebaserc` and `firebase.json`) in your root directory. Check and make sure they are there. If not, check for errors in the `firebase init` step and retry.

Step 4: Build and deploy!

Build your app for web:

```
$ flutter build web
```

This builds the necessary files in `<root-directory>/build/web`.

Note: If you see a warning: “Experimental feature web is not supported on stable branches”, make sure you are on at least the dev channel and have web enabled. For example:

```
$ flutter channel
  beta
* dev
  master
  stable
$ ~ $ flutter devices
2 connected devices:

Chrome     • chrome     • web-javascript • Google Chrome
78.0.3904.87
Web Server • web-server • web-javascript • Flutter Tools
```

Finally, run:

```
$ firebase deploy
```

The hosting files are uploaded and hosted at your Firebase URL. You can follow the link in your terminal to your Flutter web project!

Flutter Firebase Flutter Web Firebasehosting