Code Clapper

A Software Developed By Dakota Moore

Build & Deployment Guides

Last Revision: October 6th, 2022

Table of Contents

First-Time Deployment

1. Downloading Necessary Software and Files

1.1. Download/Clone the newest version of Code Clapper

This would be the stable-build folder located in projects Git Repo here:
https://github.com/DAMooreSE/codeClapper

1.2. Download Visual Studio Code

This will be the IDE we use to edit the Firebase configuration, run the commands that will install the necessary dependencies, and to deploy and run Code Clapper. Visual Studio Code can be downloaded here:
https://code.visualstudio.com/download

1.3. Download NodeJS and NPM

These are necessary for Code Clapper to successfully build and run.
These can be downloaded here:
https://nodejs.org/en/download

Next, if you haven't gone through the process of creating a Firebase account, configuring a new application, and linking it to your version of CodeClapper, the next steps will walk you through the process of doing so and swapping the firebase configuration code in a couple of your copied project files with new code provided to you after setting up your firebase application. If you have already configured your firebase application and implemented the changes to the code, please move onto the next step, Installing Dependencies.

2. Configuring Firebase

This project uses Google Firebase to manage multiple components of the project but requires new users to create and configure a new Firebase project that will be tied to that user's build. After following the steps to create a new project, it will supply the user with a block of code containing their unique configuration that needs to be copied and then used to replace specific parts of the code in order for the build to be tied to your Firebase project.

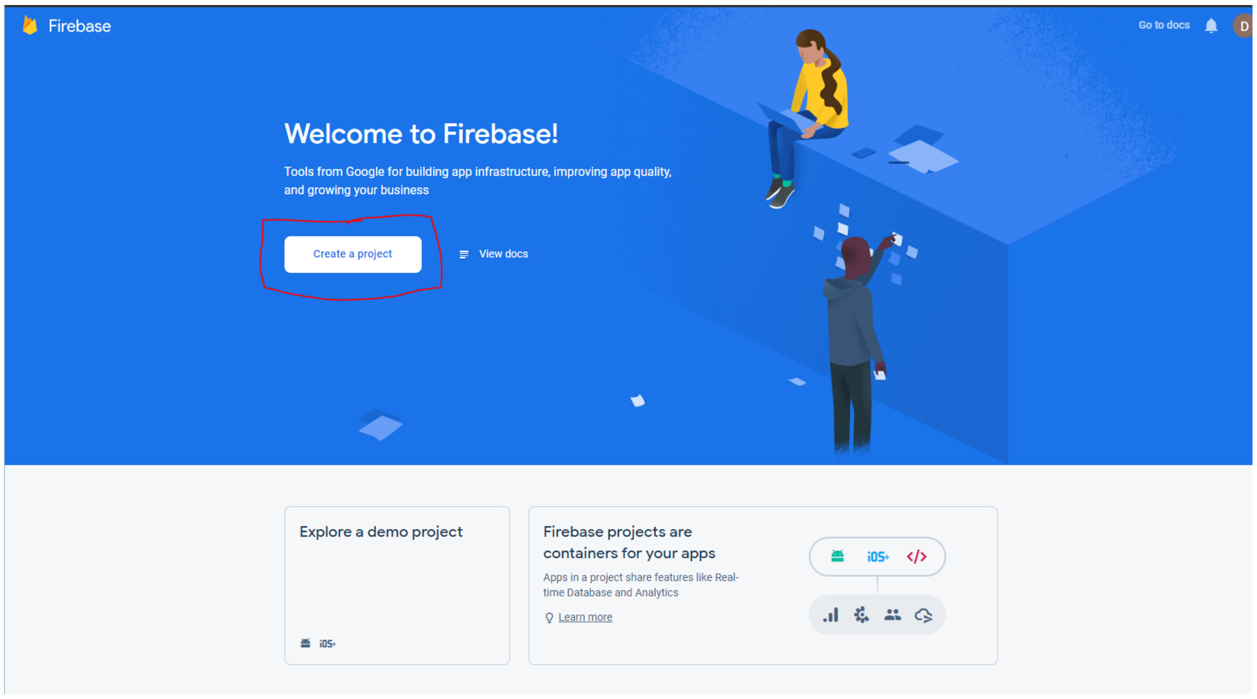
A video guide for creating a Firebase Account and properly configuring the project can be found here:

https://drive.google.com/file/d/1o7aLTdsmUv14VL0WJFEKHKoFuaW0Zzhh/view?usp=sharing

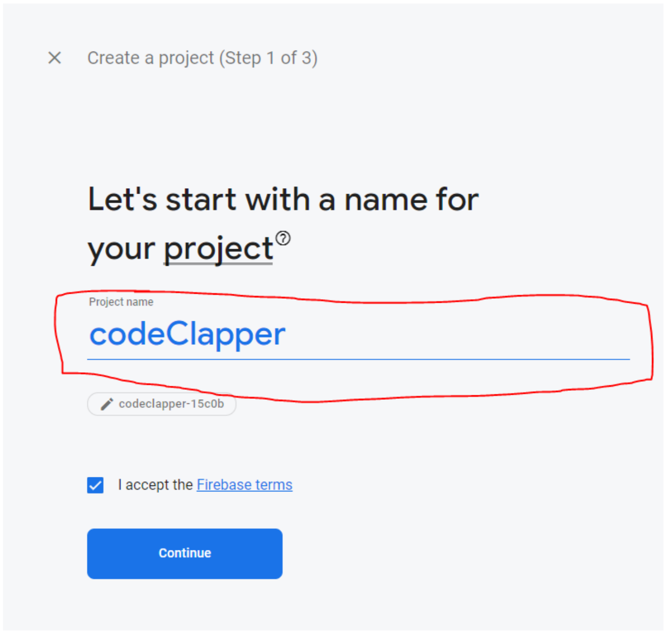
(Use Default values/options if not instructed otherwise)

2.1. Open a web browser and go to https://firebase.google.com/

2.2. Select "Create a project"


Firebase
Go to docs
D
Welcome to Firebase!
Tools from Google for building app infrastructure, improving app quality, and growing your business
Create a project
View docs
Explore a demo project
iOS+
Firebase projects are containers for your apps
Apps in a project share features like Real-time Database and Analytics
Learn more
iOS+
</>

2.3. Name your project whatever you want, I use "codeClapper", and accept the terms and click continue


× Create a project (Step 1 of 3)

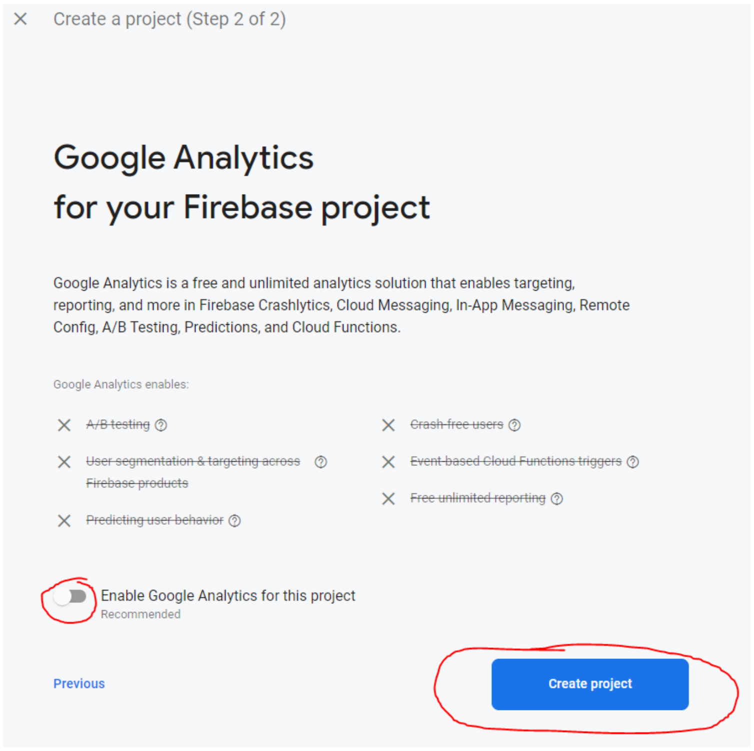
Let's start with a name for your project

Project name
codeClapper

codeclapper-15c0b

☑ I accept the Firebase terms

2.4. Make sure Google Analytics is NOT enabled for the project and then select "create project"


× Create a project (Step 2 of 2)

Google Analytics for your Firebase project

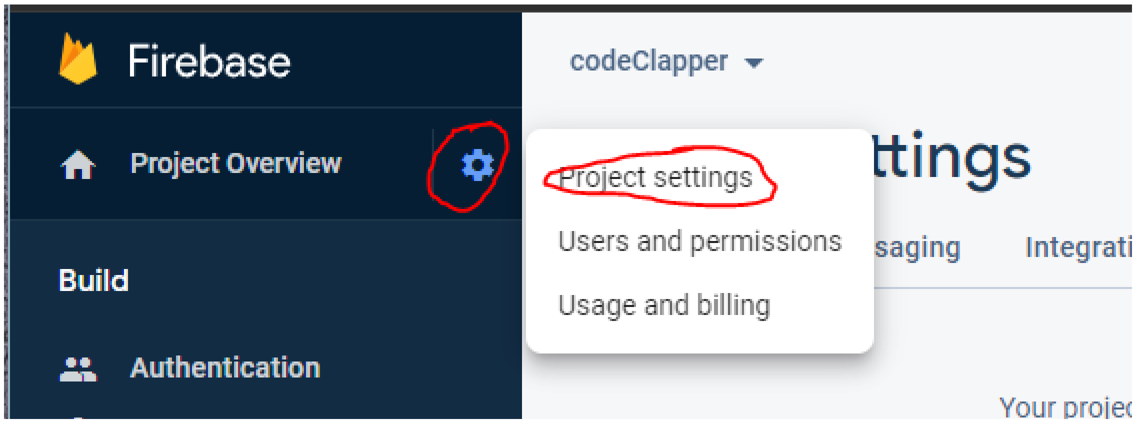
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, Predictions, and Cloud Functions.

Google Analytics enables:

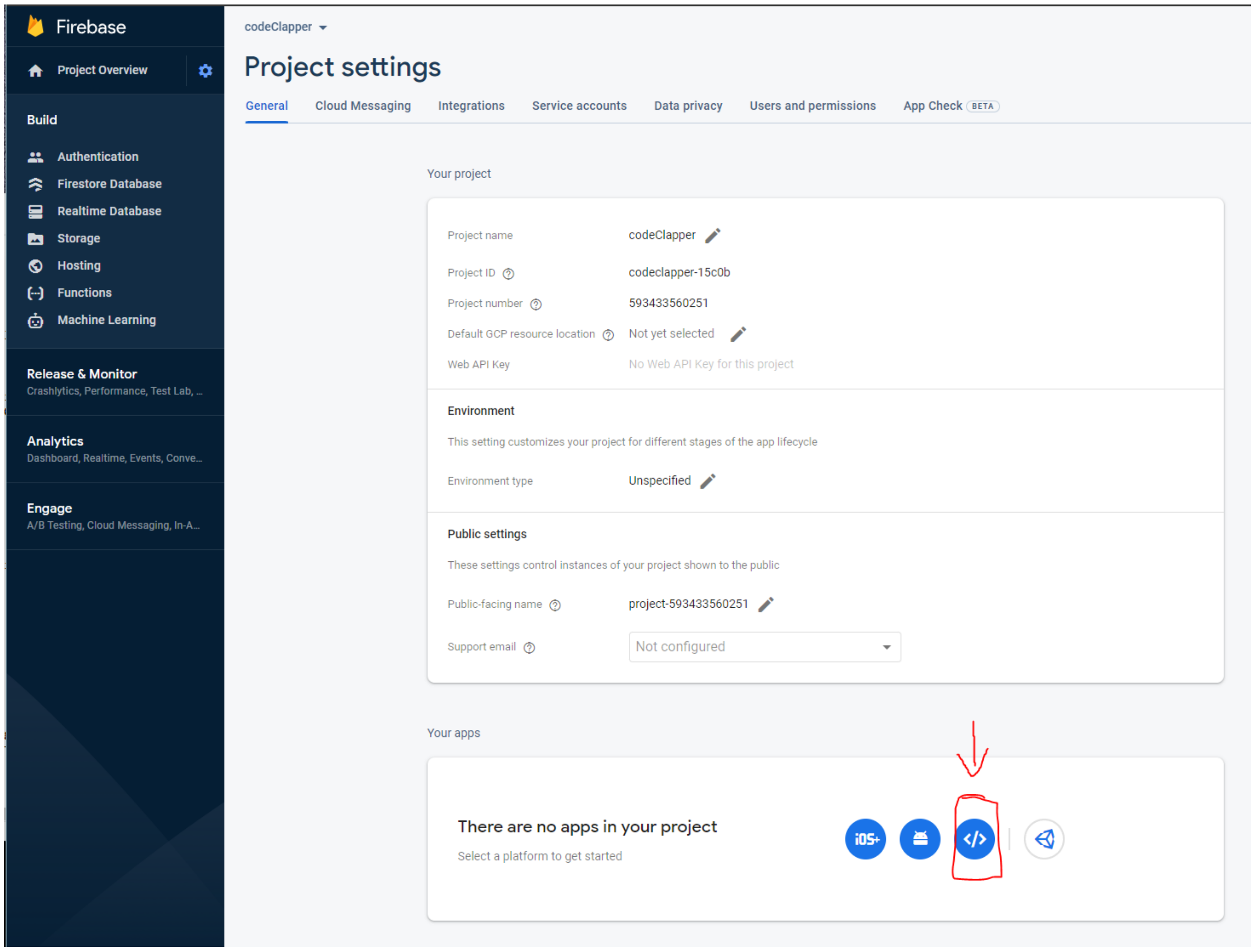
× ~~A/B testing~~
× ~~User segmentation & targeting across Firebase products~~
× ~~Predicting user behavior~~
× ~~Crash-free users~~
× ~~Event-based Cloud Functions triggers~~
× ~~Free unlimited reporting~~

Enable Google Analytics for this project
Recommended

2.5. Select the gear icon next to "project overview" and then "project settings"



2.6. Under "your apps", select the option to add a new web application.



Project settings

Your project

Project name	codeClapper
Project ID	codeclapper-15c0b
Project number	593433560251
Default GCP resource location	Not yet selected
Web API Key	No Web API Key for this project

Environment

This setting customizes your project for different stages of the app lifecycle

Environment type	Unspecified

Public settings

These settings control instances of your project shown to the public

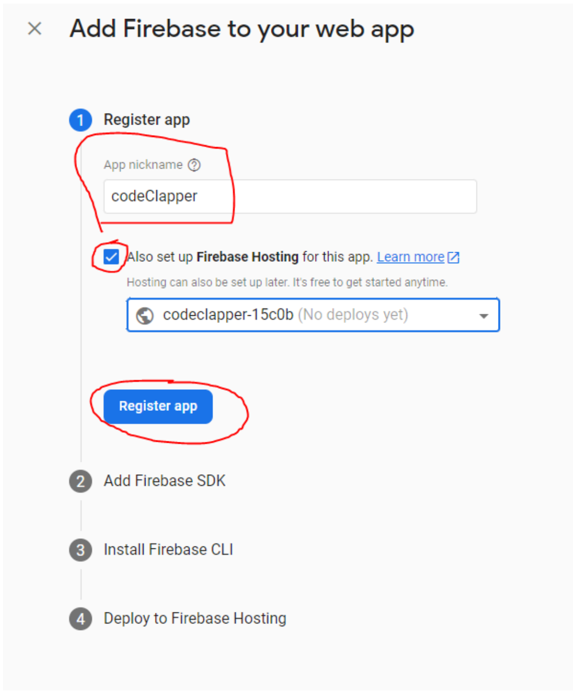
Public-facing name	project-593433560251
Support email	Not configured

Your apps

There are no apps in your project

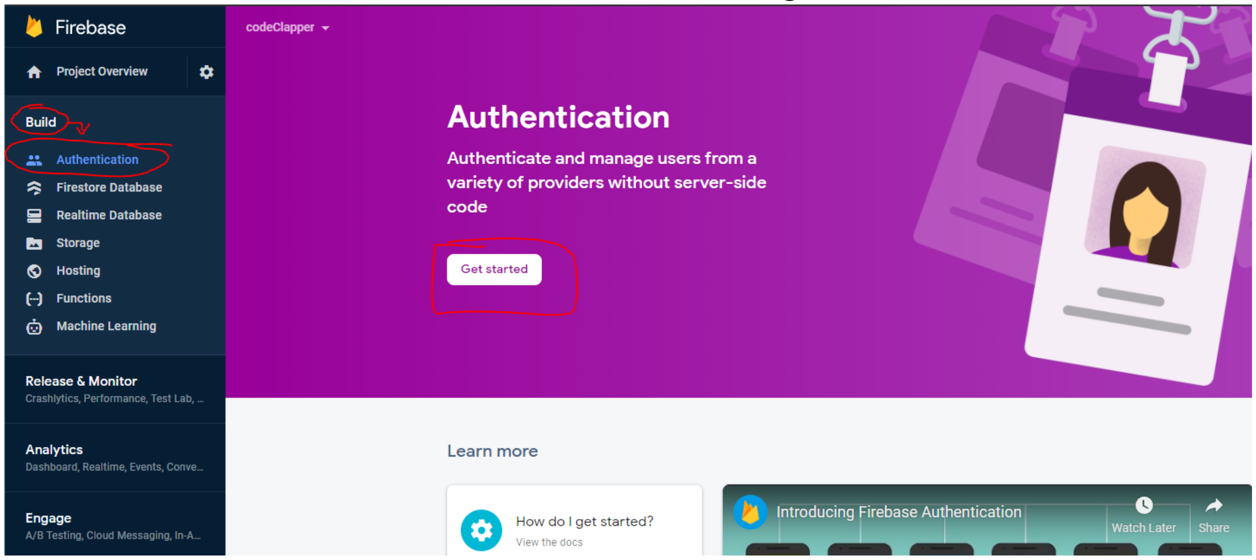
Select a platform to get started

2.7. Again, name your app whatever you want, I stick with codeClapper, and check the box to set up firebase hosting before selecting "register app"

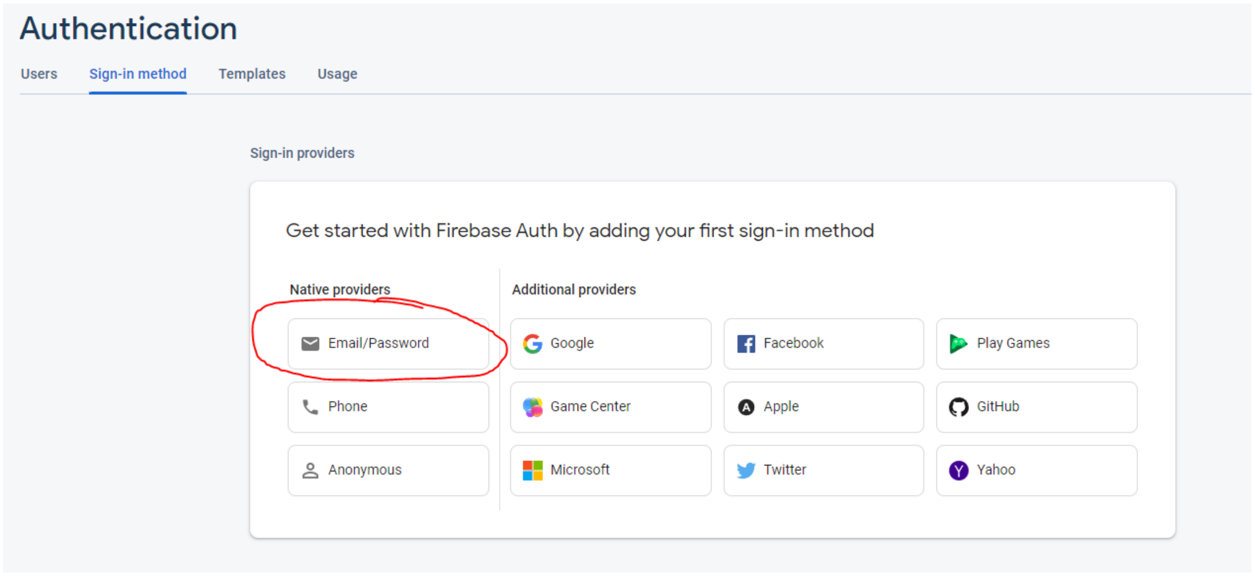

Add Firebase to your web app
1 Register app
App nickname
codeClapper
Also set up Firebase Hosting for this app. Learn more
Hosting can also be set up later. It's free to get started anytime.
codeclapper-15c0b (No deploys yet)
Register app
2 Add Firebase SDK
3 Install Firebase CLI
4 Deploy to Firebase Hosting

2.8. You can skip the instructions for adding the Firebase SDK and CLI for now, we'll do that afterwards, keep selecting next and then finally "continue to console".

2.9. Select "build", then "authentication", then "get started"

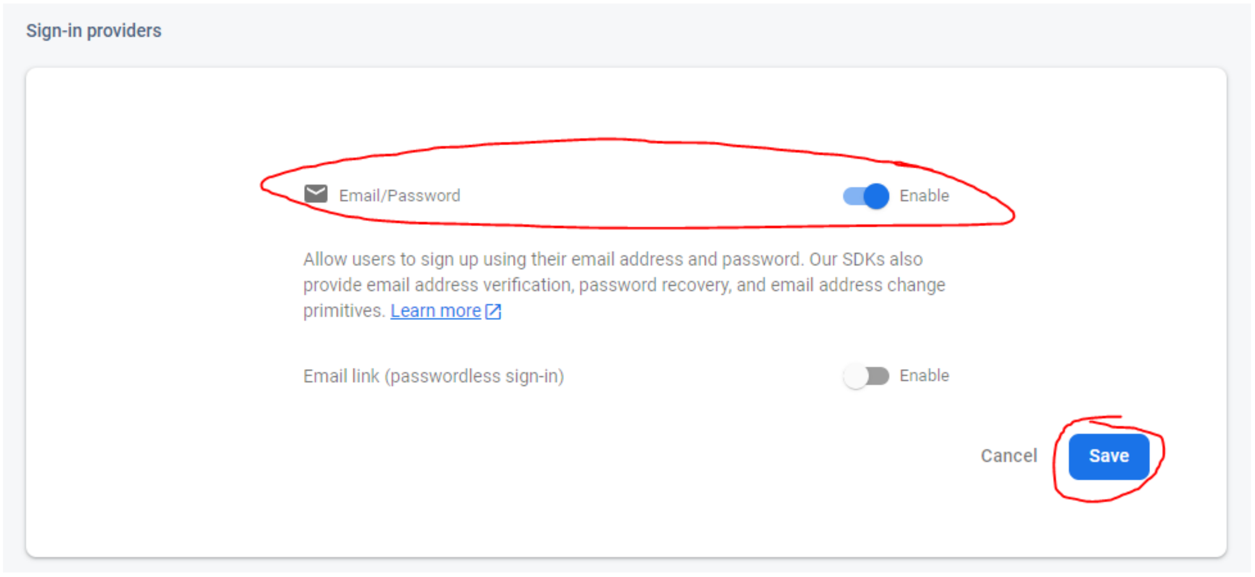


2.10. Select "Email/Password" from "Native providers" under "sign-in providers"



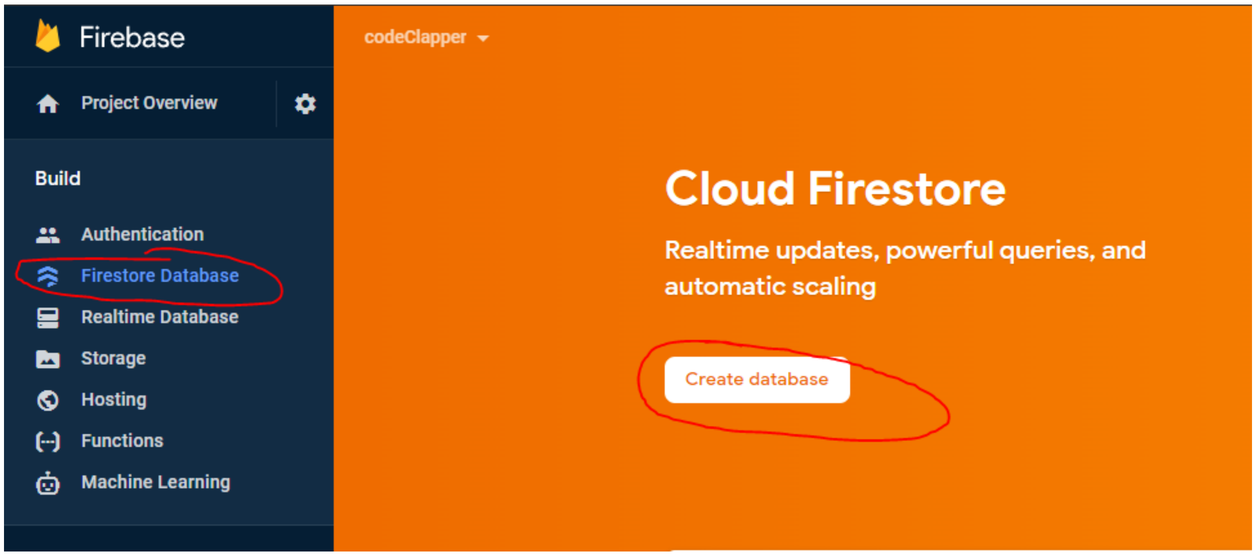
Get started with Firebase Auth by adding your first sign-in method

2.11. Enable Email/Password and click "save"

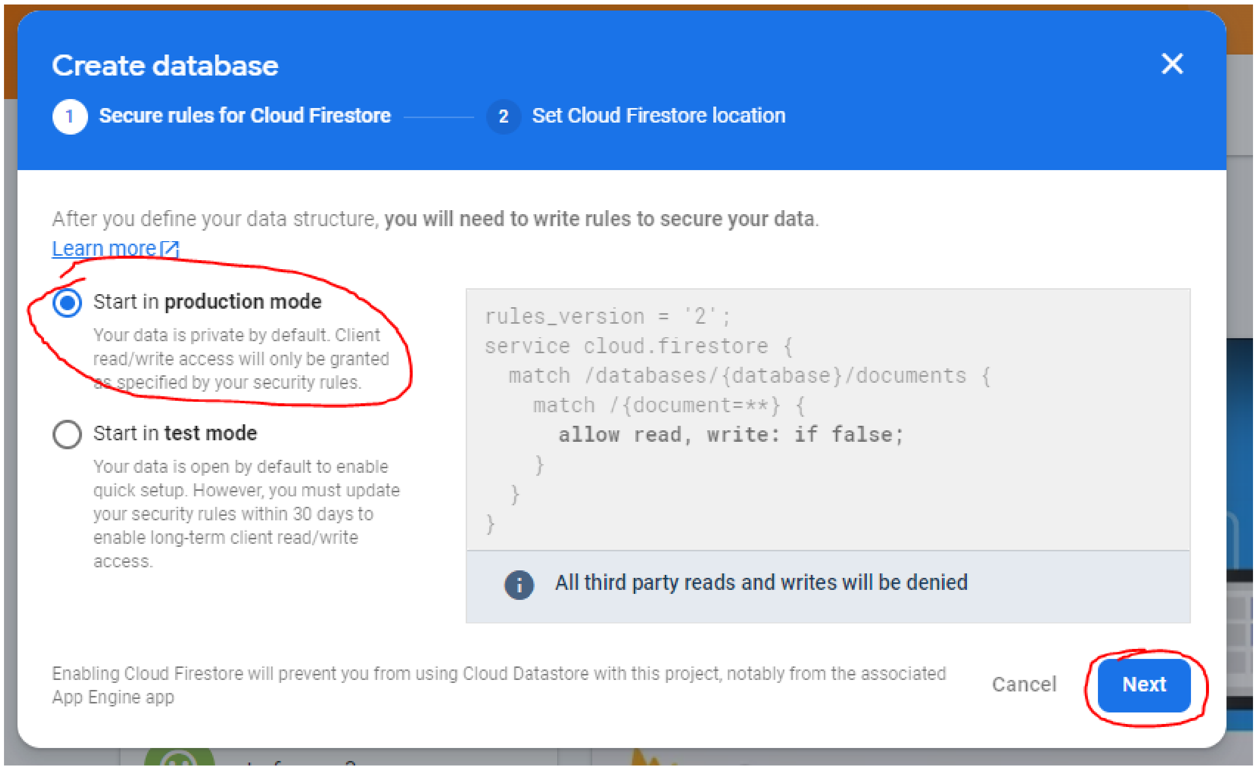


Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. Learn more

2.12. Select "Firestore Database" under "Build" then "create database"



2.13. Select "Start in production mode" and then select "next"


Create database
1 Secure rules for Cloud Firestore
2 Set Cloud Firestore location
Learn more
Cancel
Next

After you define your data structure, **you will need to write rules to secure your data.**

[✓ Start in **production mode**]
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

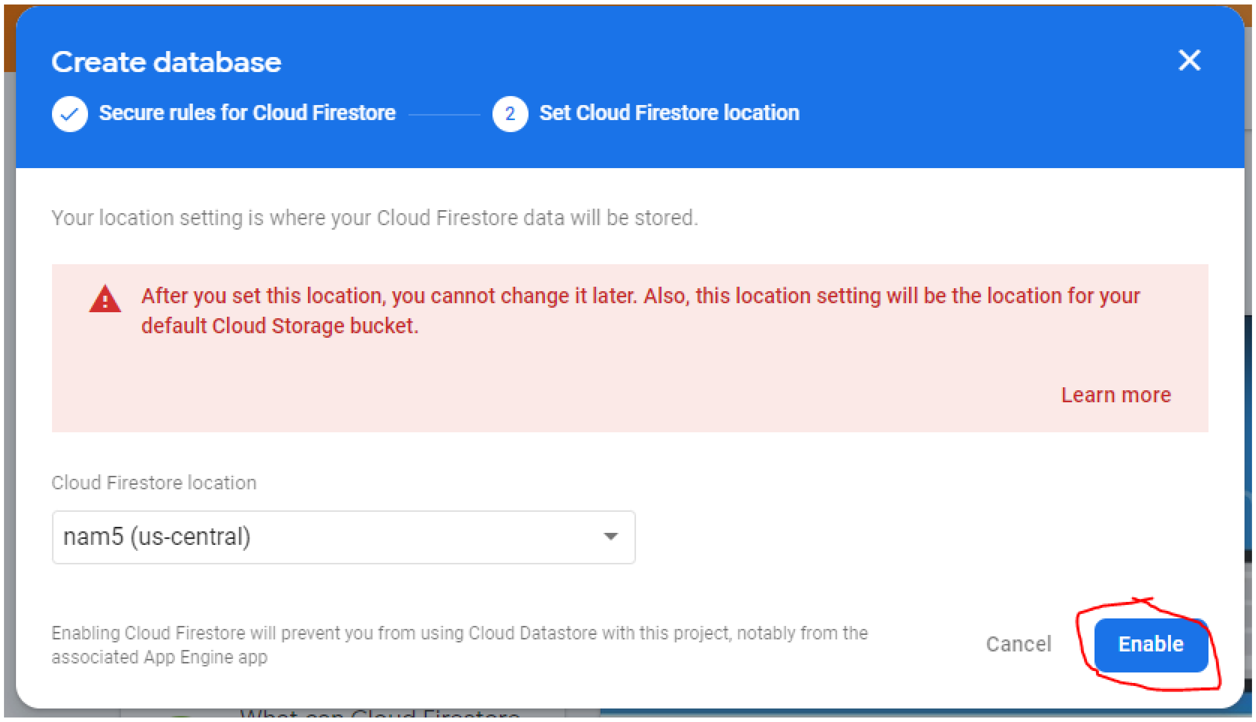
Start in **test mode**
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if false;
    }
  }
}
```

All third party reads and writes will be denied

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

2.14. Select whatever datacenter you'd like the database to be hosted at and select "enable"


Create database
Secure rules for Cloud Firestore
2 Set Cloud Firestore location

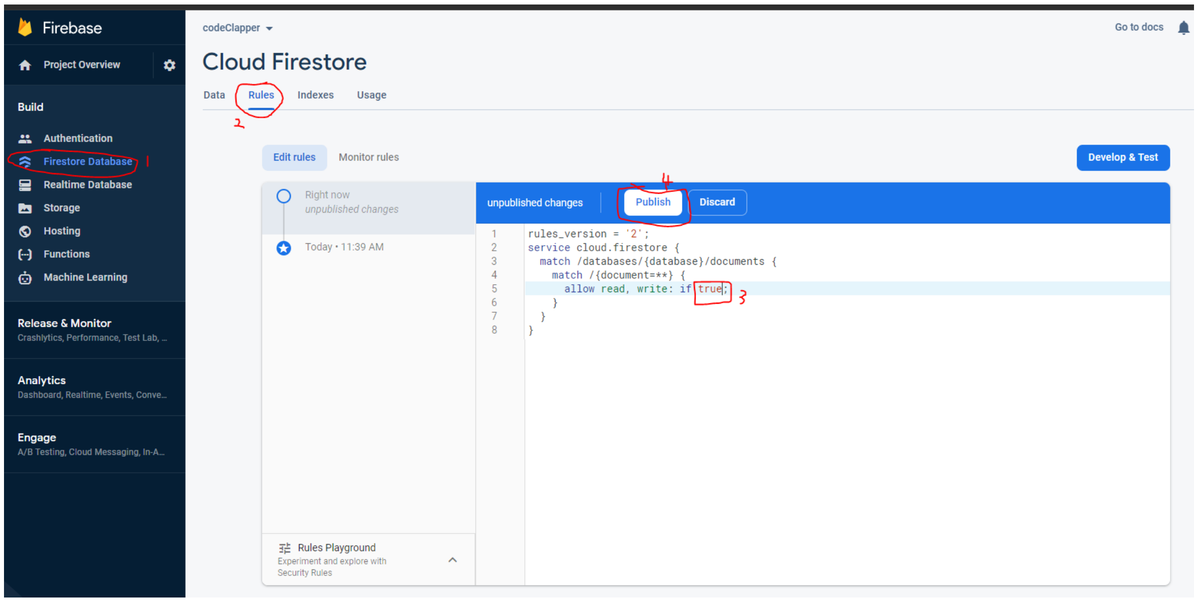
Your location setting is where your Cloud Firestore data will be stored.

After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

Cloud Firestore location: nam5 (us-central)

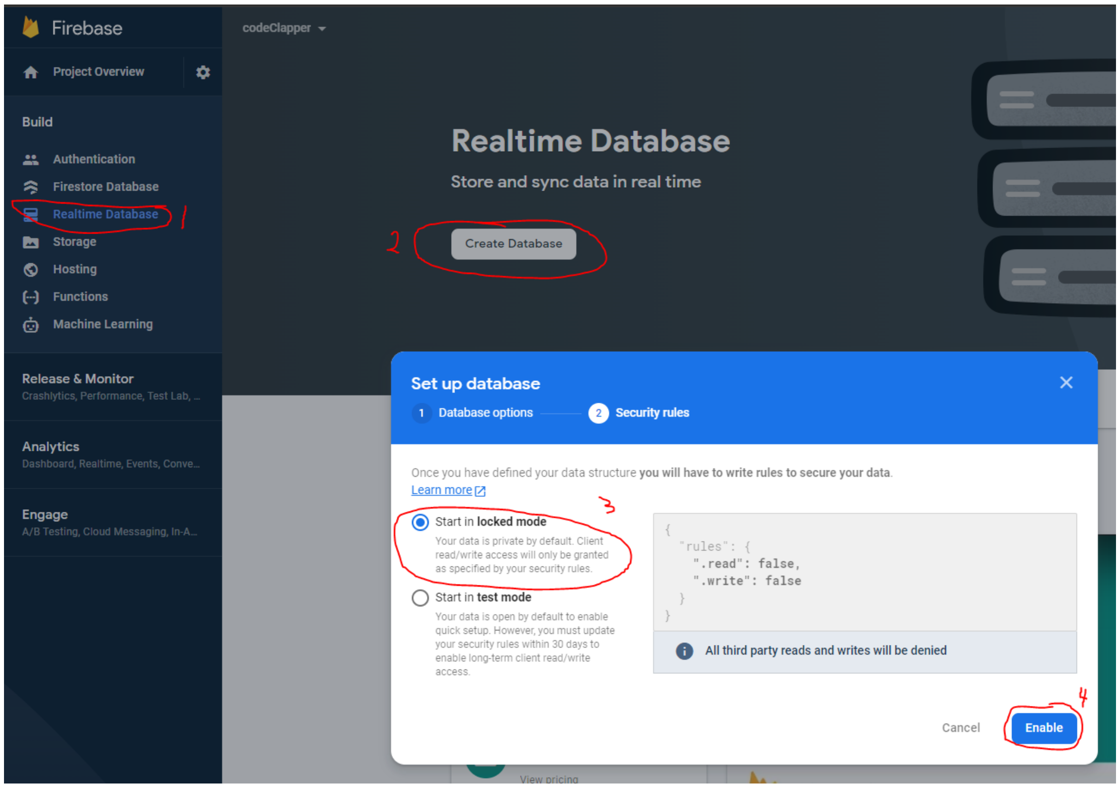
Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

2.15. Staying in "firestore database", select the "Rules" tab and change line 5 to "if true;" instead of "if false;" and select publish



```
1  rules_version = '2';
2  service cloud.firestore {
3    match /databases/{database}/documents {
4      match /{document=**} {
5        allow read, write: if true;
6      }
7    }
8  }
```

2.16. Select "realtime database" under "Build", then "create database", select "start in locked mode" and then "enable"



Once you have defined your data structure **you will have to write rules to secure your data.**

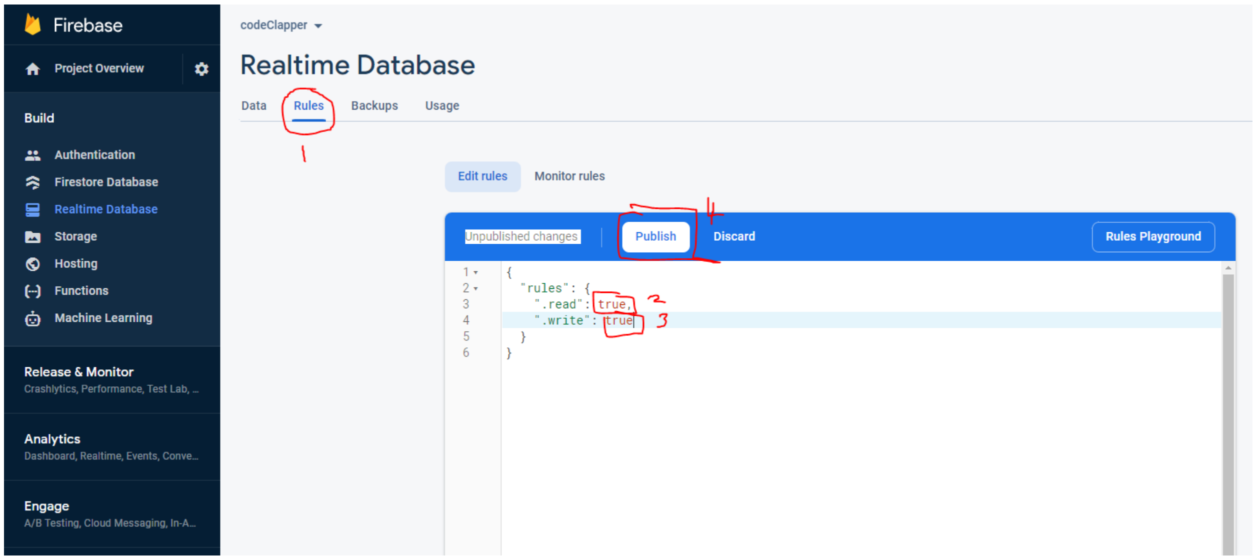
[✓ Start in **locked mode**] Your data is private by default. Client read/write access will only be granted as specified by your security rules.

Start in **test mode** Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": false,
    ".write": false
  }
}
```

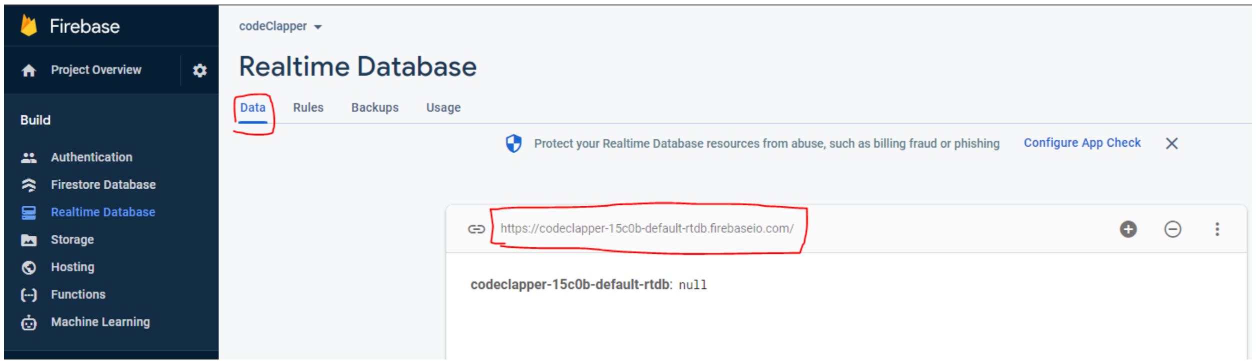
All third party reads and writes will be denied

2.17. Staying in "realtime database", select the "rules" tab and then change "false" to "true" on lines 3 and 4 before selecting "publish"



```
1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }
```

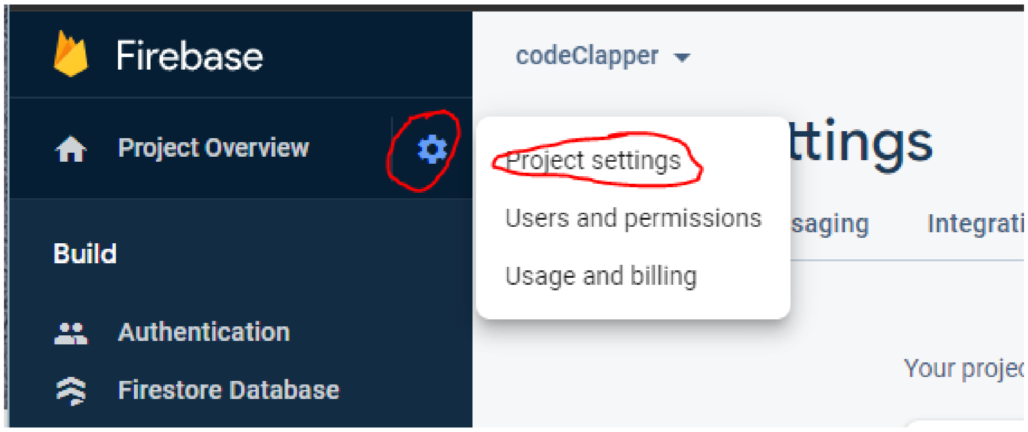
2.18. Staying in "realtime database", select the "Data" tab and then copy the url of the database to some place where you'll be able copy it again for use later.



https://codeclapper-15c0b-default-rtdb.firebaseio.com/

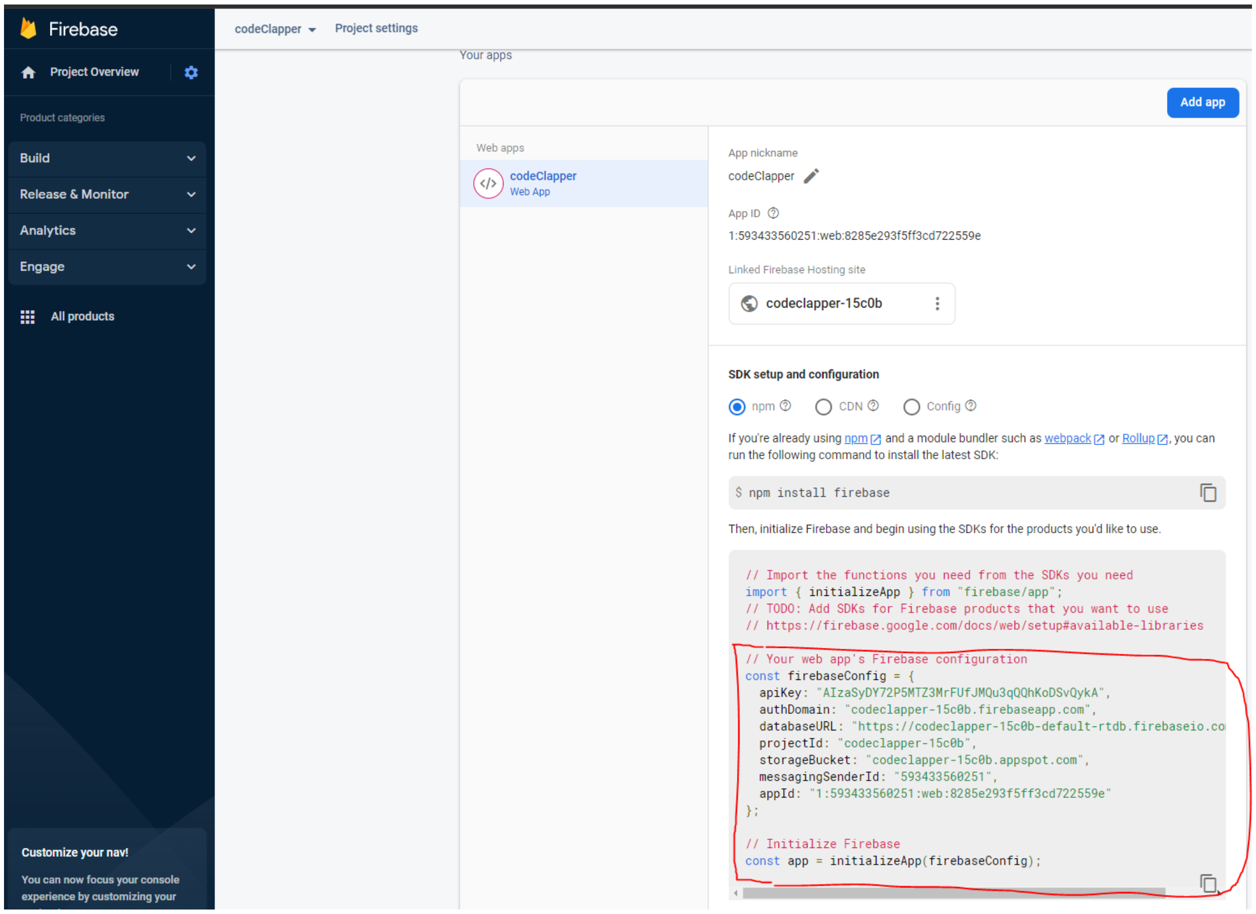
codeclapper-15c0b-default-rtdb: null

2.19. Select the gear icon next to "project overview" and then "project settings" then scroll down.



2.20. Under "SDK setup and configuration", scroll down and go to the second text box of code starting with "// Import the..."

2.21. You're going to copy everything from the line with "const firebaseConfig = {" down to "const app = initializeApp(firebaseConfig);"



App nickname
codeClapper

App ID
1:593433560251:web:8285e293f5ff3cd722559e

Linked Firebase Hosting site
codeclapper-15c0b

SDK setup and configuration

[✓ npm] CDN Config

If you're already using npm and a module bundler such as webpack or Rollup, you can run the following command to install the latest SDK:

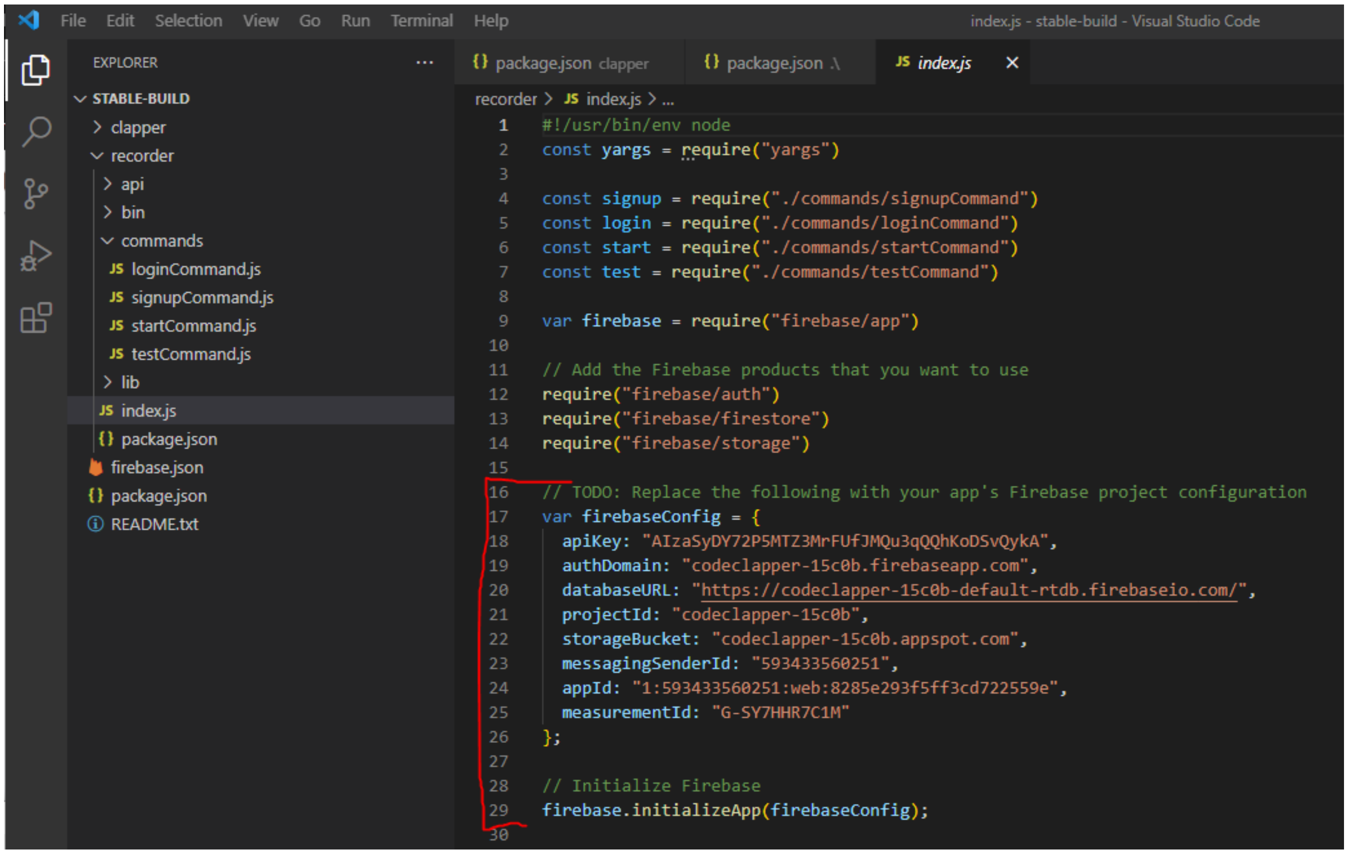
```
$ npm install firebase
```

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyDY72P5MTZ3MrFUfJMQu3qQQhKoDSvQykA",
  authDomain: "codeclapper-15c0b.firebaseapp.com",
  databaseURL: "https://codeclapper-15c0b-default-rtdb.firebaseio.co
  projectId: "codeclapper-15c0b",
  storageBucket: "codeclapper-15c0b.appspot.com",
  messagingSenderId: "593433560251",
  appId: "1:593433560251:web:8285e293f5ff3cd722559e"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

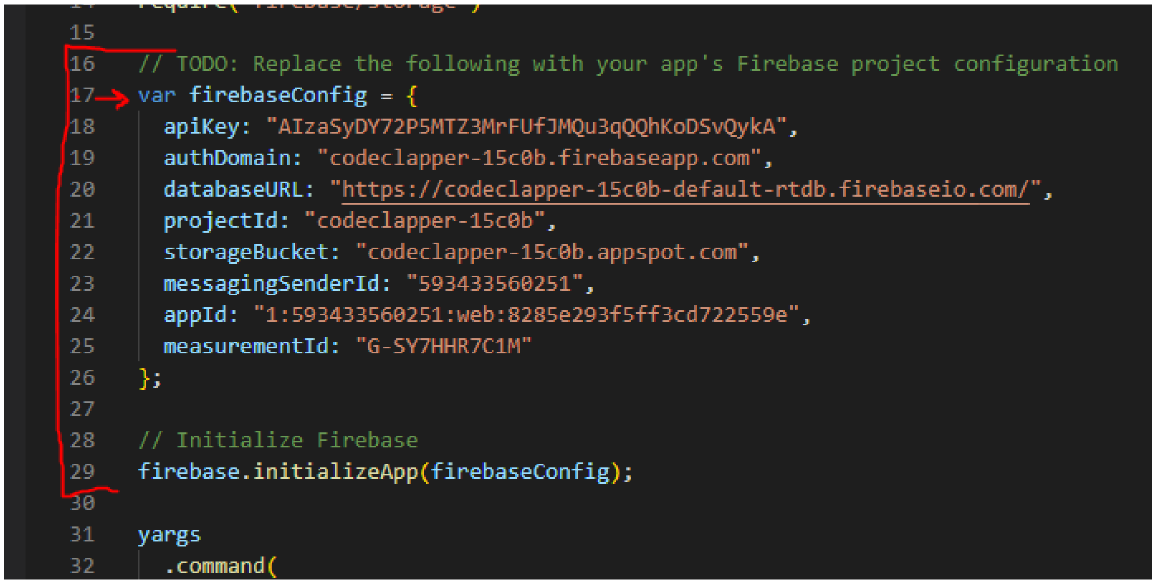
2.22. Open /recorder/index.js make the following changes:

2.22.1. Replace lines 17 through 29 with your copied code.



```
 1  #!/usr/bin/env node
 2  const yargs = require("yargs")
 3
 4  const signup = require("./commands/signupCommand")
 5  const login = require("./commands/loginCommand")
 6  const start = require("./commands/startCommand")
 7  const test = require("./commands/testCommand")
 8
 9  var firebase = require("firebase/app")
10
11  // Add the Firebase products that you want to use
12  require("firebase/auth")
13  require("firebase/firestore")
14  require("firebase/storage")
15
16  // TODO: Replace the following with your app's Firebase project configuration
17  var firebaseConfig = {
18    apiKey: "AIzaSyDY72P5MTZ3MrFUfJMQu3qQQhKoDSvQykA",
19    authDomain: "codeclapper-15c0b.firebaseapp.com",
20    databaseURL: "https://codeclapper-15c0b-default-rtdb.firebaseio.com/",
21    projectId: "codeclapper-15c0b",
22    storageBucket: "codeclapper-15c0b.appspot.com",
23    messagingSenderId: "593433560251",
24    appId: "1:593433560251:web:8285e293f5ff3cd722559e",
25    measurementId: "G-SY7HHR7C1M"
26  };
27
28  // Initialize Firebase
29  firebase.initializeApp(firebaseConfig);
30
```

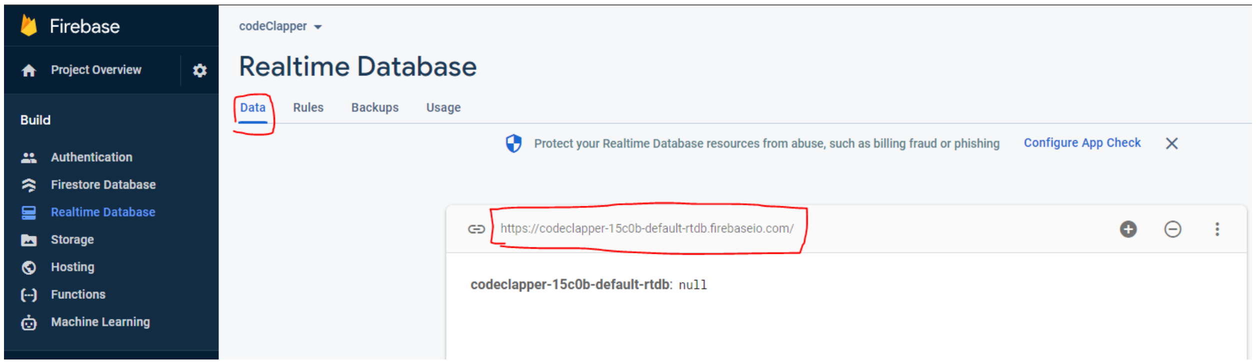
2.22.2. On line 17, change "const firebaseConfig = {" to "var firebaseConfig = {"



```
15
16  // TODO: Replace the following with your app's Firebase project configuration
17  var firebaseConfig = {
18    apiKey: "AIzaSyDY72P5MTZ3MrFUfJMQu3qQQhKoDSvQykA",
19    authDomain: "codeclapper-15c0b.firebaseapp.com",
20    databaseURL: "https://codeclapper-15c0b-default-rtdb.firebaseio.com/",
21    projectId: "codeclapper-15c0b",
22    storageBucket: "codeclapper-15c0b.appspot.com",
23    messagingSenderId: "593433560251",
24    appId: "1:593433560251:web:8285e293f5ff3cd722559e",
25    measurementId: "G-SY7HHR7C1M"
26  };
27
28  // Initialize Firebase
29  firebase.initializeApp(firebaseConfig);
30
31  yargs
32    .command(
```

2.22.3. If you're missing a line starting with "databaseURL: " under line 19 "authDomain: " then follow these steps to add it:

2.22.4. Go back to the firebase web page and go to "realtime database" again, copy the URL for your previously created database



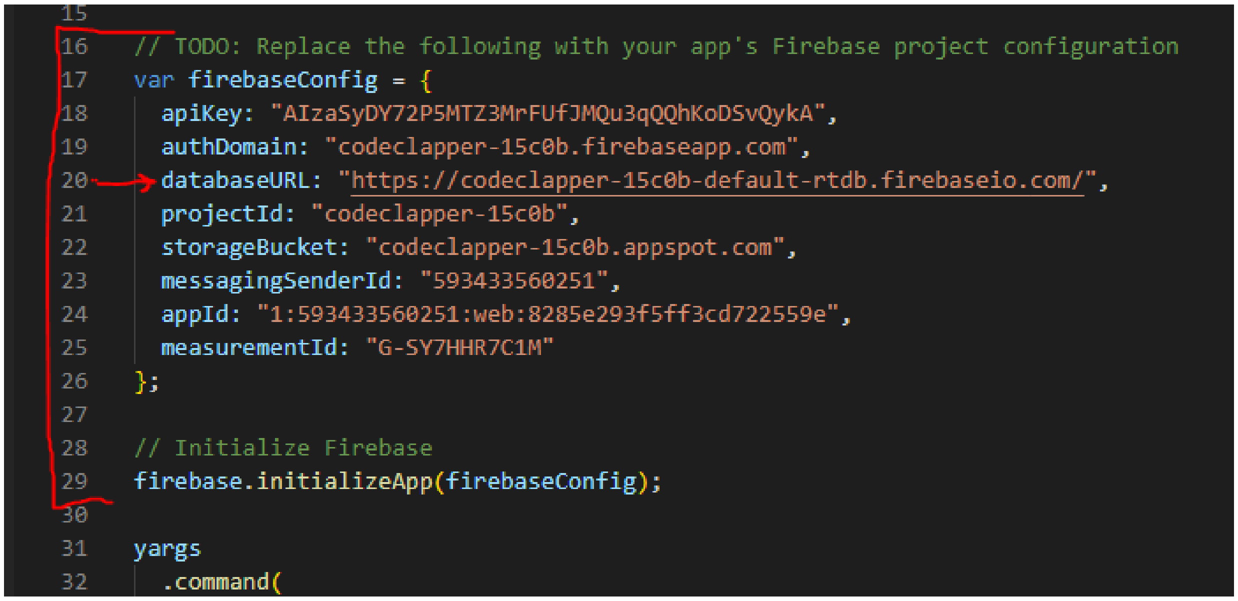
Realtime Database

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

https://codeclapper-15c0b-default-rtdb.firebaseio.com/

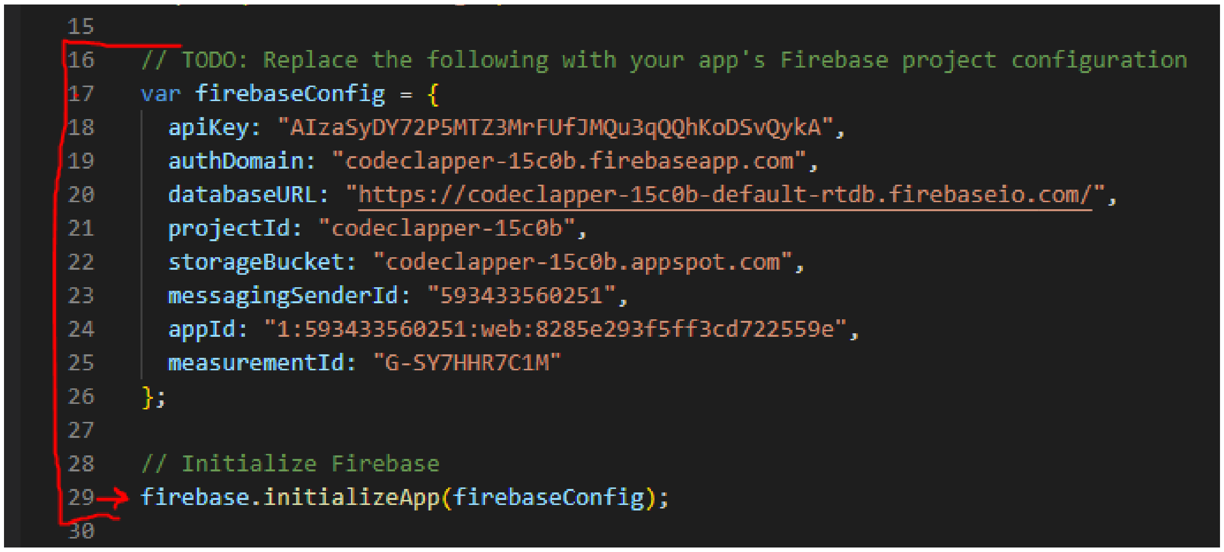
codeclapper-15c0b-default-rtdb: null

2.22.5. Create a new line under "authDomain:" and add the line "databaseURL: "URLFROMPREVIOUSSTEP", " making sure to include the comma at the end.



```
15
16   // TODO: Replace the following with your app's Firebase project configuration
17   var firebaseConfig = {
18     apiKey: "AIzaSyDY72P5MTZ3MrFUfJMQu3qQQhKoDSvQykA",
19     authDomain: "codeclapper-15c0b.firebaseapp.com",
20     databaseURL: "https://codeclapper-15c0b-default-rtdb.firebaseio.com/",
21     projectId: "codeclapper-15c0b",
22     storageBucket: "codeclapper-15c0b.appspot.com",
23     messagingSenderId: "593433560251",
24     appId: "1:593433560251:web:8285e293f5ff3cd722559e",
25     measurementId: "G-SY7HHR7C1M"
26   };
27
28   // Initialize Firebase
29   firebase.initializeApp(firebaseConfig);
30
31   yargs
32     .command(
```

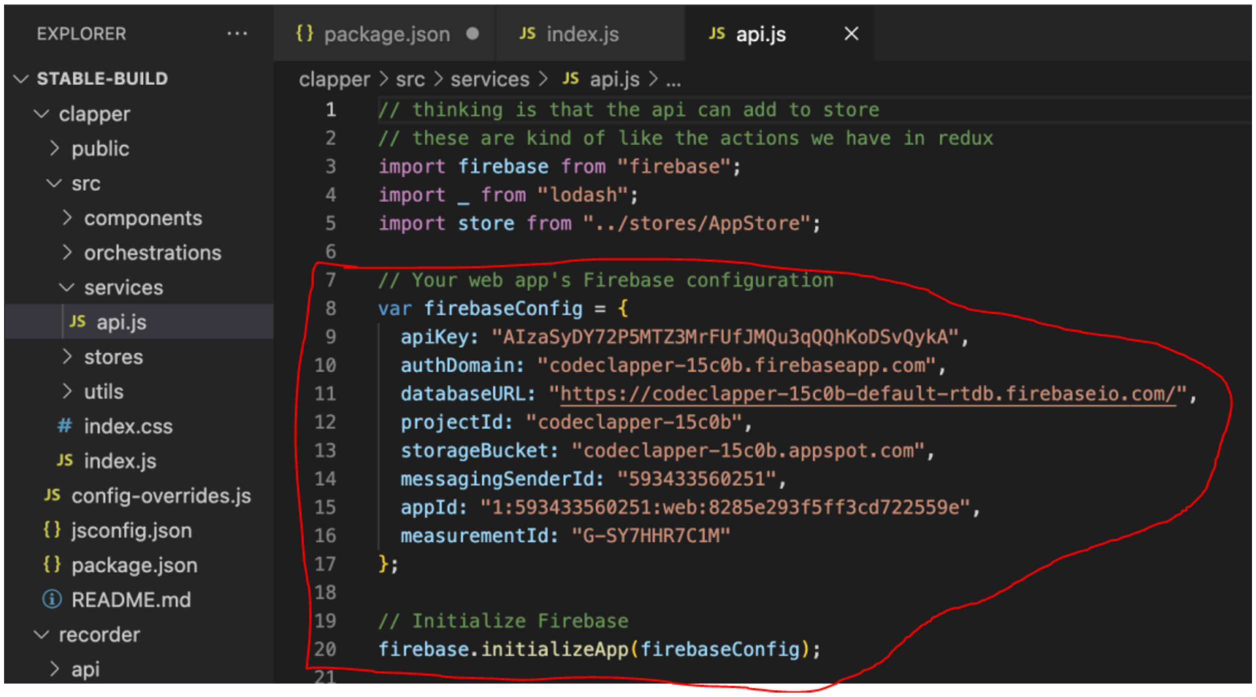
2.22.6. On line 29, change "const app = initializeApp(firebaseConfig);" to "firebase.initializeApp(firebaseConfig)"



```
15
16   // TODO: Replace the following with your app's Firebase project configuration
17   var firebaseConfig = {
18     apiKey: "AIzaSyDY72P5MTZ3MrFUfJMQu3qQQhKoDSvQykA",
19     authDomain: "codeclapper-15c0b.firebaseapp.com",
20     databaseURL: "https://codeclapper-15c0b-default-rtdb.firebaseio.com/",
21     projectId: "codeclapper-15c0b",
22     storageBucket: "codeclapper-15c0b.appspot.com",
23     messagingSenderId: "593433560251",
24     appId: "1:593433560251:web:8285e293f5ff3cd722559e",
25     measurementId: "G-SY7HHR7C1M"
26   };
27
28   // Initialize Firebase
29   firebase.initializeApp(firebaseConfig);
30
```

2.22.7. You're now going to copy lines 17 through 29 and paste them in another file.

2.22.8. Open /clapper/src/services/api.js and replace lines 8 through 20 with your copied code and then save both files.



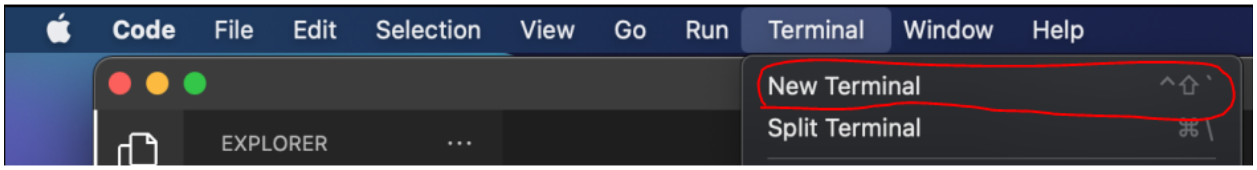
```
// thinking is that the api can add to store
// these are kind of like the actions we have in redux
import firebase from "firebase";
import _ from "lodash";
import store from "../stores/AppStore";

// Your web app's Firebase configuration
var firebaseConfig = {
  apiKey: "AIzaSyDY72P5MTZ3MrFUfJMQu3qQQhKoDSvQykA",
  authDomain: "codeclapper-15c0b.firebaseapp.com",
  databaseURL: "https://codeclapper-15c0b-default-rtdb.firebaseio.com/",
  projectId: "codeclapper-15c0b",
  storageBucket: "codeclapper-15c0b.appspot.com",
  messagingSenderId: "593433560251",
  appId: "1:593433560251:web:8285e293f5ff3cd722559e",
  measurementId: "G-SY7HHR7C1M"
};

// Initialize Firebase
firebase.initializeApp(firebaseConfig);
```

3. Installing Necessary Packages For First Deployment

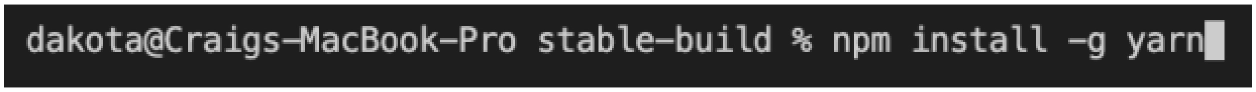
The following steps only need to be done once during the first deployment of Code Clapper for a user. After these packages are installed once, there's no need to redo any of these steps when redeploying Code Clapper in the future.

3.1. Open Visual Studio Code and select file then open folder in the menu bar and locate the stable-build folder containing the newest version of Code Clapper to open the project.

3.2. Open a new terminal by selecting Terminal, then New Terminal in the menu bar.

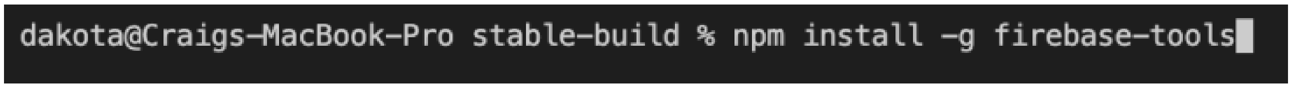


3.3. Type the command "npm install -g yarn" and hit enter



```
dakota@Craigs-MacBook-Pro stable-build % npm install -g yarn
```

3.4. Type the command "npm install -g firebase-tools"

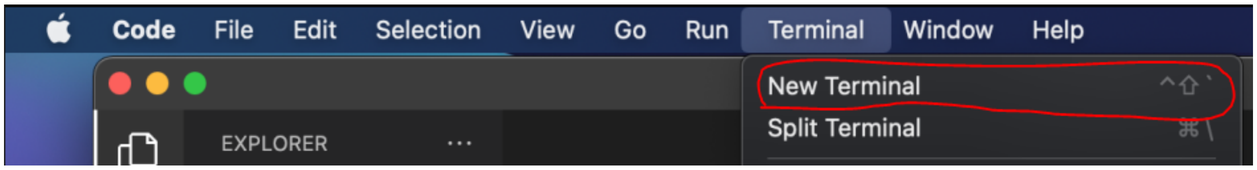


```
dakota@Craigs-MacBook-Pro stable-build % npm install -g firebase-tools
```

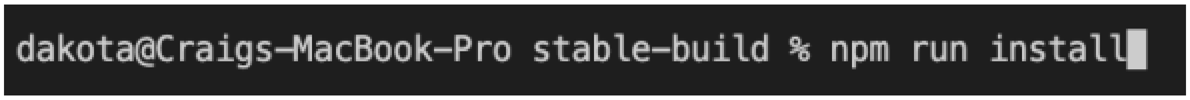
4. Starting the UI

4.1. Open the project in Visual Studio Code if it's not already open from the last step.

4.2. If one's not already open, open a new terminal by clicking Terminal, then New Terminal in the menu bar.

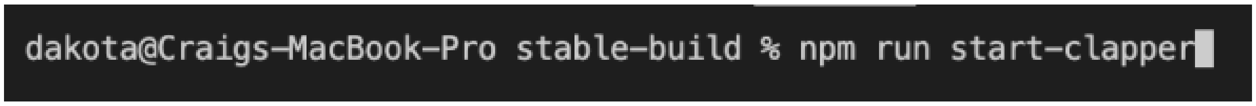


4.3. Type "npm run install" in the terminal and then hit enter and wait for all the packages to download and install.



```
dakota@Craigs-MacBook-Pro stable-build % npm run install
```

4.4. Type "npm run start-clapper" in the terminal and then enter.



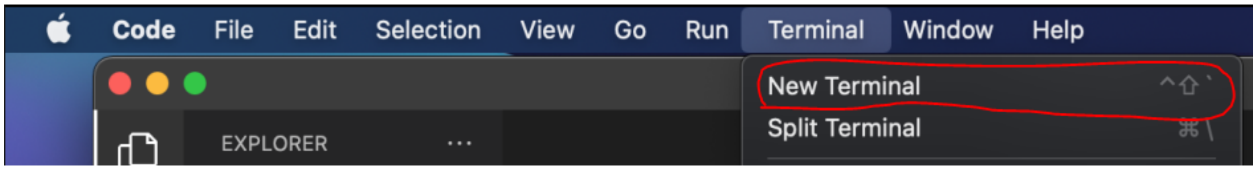
```
dakota@Craigs-MacBook-Pro stable-build % npm run start-clapper
```

This should start and then run the services required for the User Interface and after compiling will open the Code Clapper application in your default browser by going to the URL localhost:8080. The UI can be accessed by any device on the same network as the host machine by opening a browser and going to the address HOST_IP_ADDRESS:8080 where HOST_IP_ADDRESS is replaced with the IP address of the host machine. It is recommended to open the application in an "incognito" tab due to an issue with cookies caching previous sessions.

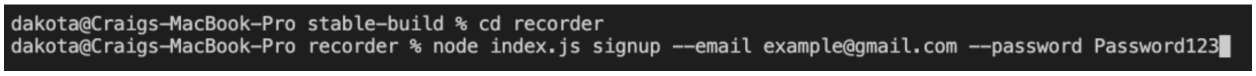
YOU MUST FOLLOW THE NEXT STEPS TO START THE RECORDING SERVICE BEFORE THE APPLICATION WILL PROPERLY FUNCTION!

5. Starting the Recorder

5.1. While leaving the other terminal open and running, open a new terminal by going to Terminal, then New Terminal in the menu bar.



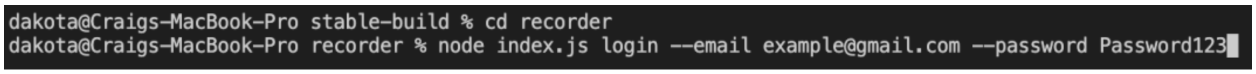
5.2. If this is the first time you're deploying Code Clapper, you must sign up and log into a new account. This can be done by typing "cd recorder" and hitting enter, then typing "node index.js signup --email YOUREMAIL --password YOURPASSWORD" (substituting whatever email and password you'd like to use) and hitting enter.



```
dakota@Craigs-MacBook-Pro stable-build % cd recorder
dakota@Craigs-MacBook-Pro recorder % node index.js signup --email example@gmail.com --password Password123
```

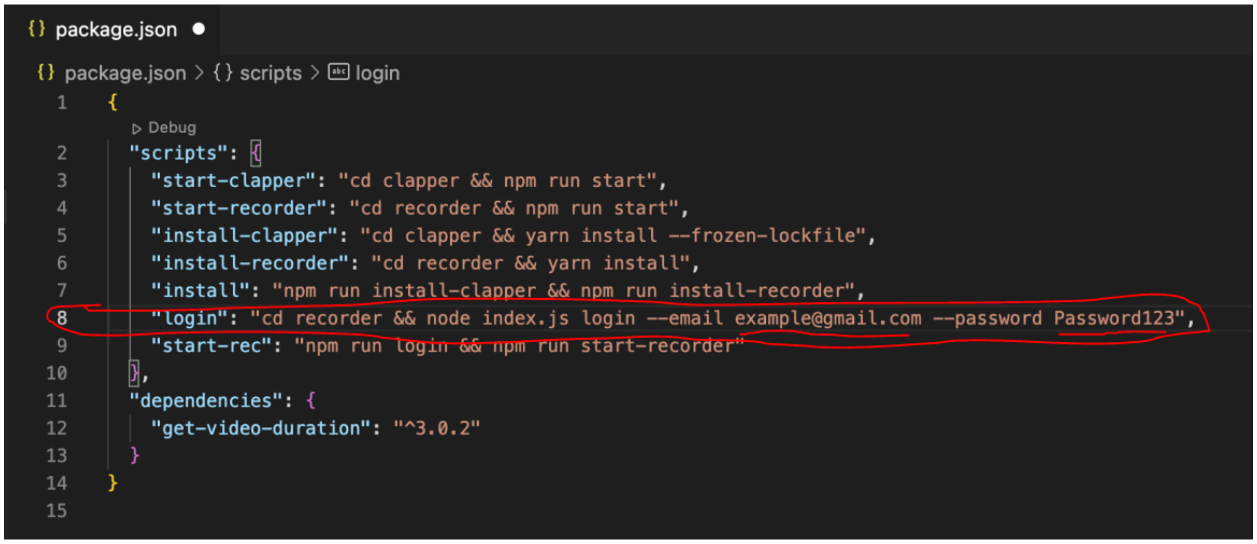
This login information will be saved in the file /stable-build/.creds.json and a user only needs to sign up once and the login will be saved in the Firebase project. You only need to log back in when deploying a new build, not when redeploying the same build.

5.3. If you've previously created an account using your current Firebase Configuration, you can sign in by typing "cd /recorder" and hitting enter in the terminal, then typing "node index.js login --email YOUREMAIL --password YOURPASSWORD", substituting your email and password for YOUREMAIL and YOURPASSWORD, respectively.



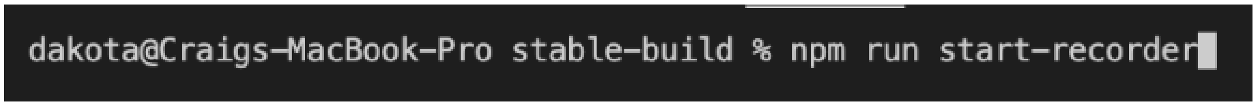
```
dakota@Craigs-MacBook-Pro stable-build % cd recorder
dakota@Craigs-MacBook-Pro recorder % node index.js login --email example@gmail.com --password Password123
```

5.4. If you already have an account created and it's saved in the /.creds.json file, but aren't logged in for this deployment, you can substitute your login information on line 8 in the file /stable-build/package.json and saving the file before typing "npm run start-rec" in the terminal and hitting enter.



```
{} package.json ●
{} package.json > {} scripts > login
{
  ▷ Debug
  "scripts": {
    "start-clapper": "cd clapper && npm run start",
    "start-recorder": "cd recorder && npm run start",
    "install-clapper": "cd clapper && yarn install --frozen-lockfile",
    "install-recorder": "cd recorder && yarn install",
    "install": "npm run install-clapper && npm run install-recorder",
    "login": "cd recorder && node index.js login --email example@gmail.com --password Password123",
    "start-rec": "npm run login && npm run start-recorder"
  },
  "dependencies": {
    "get-video-duration": "^3.0.2"
  }
}
```

5.5. If you've already logged in for this build of Code Clapper, you can just type "npm run start-recorder" in the new terminal and hit enter for the recording service to start. If you receive a message like "you need to be logged in to do that!" refer back to step 5.2 or 5.3.\



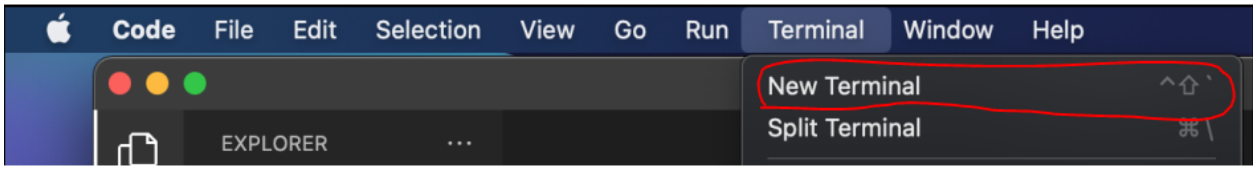
```
dakota@Craigs-MacBook-Pro stable-build % npm run start-recorder
```

If you've done everything successfully, the recording service is now running and your Code Clapper should have full functionality!

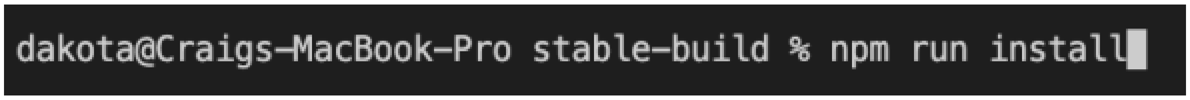
Redeploying Code Clapper

1. Installing Fresh Dependencies

1.1. Open your Code Clapper Project in Visual Studio Code and then open a new terminal with Terminal, then New Terminal in the menu bar.



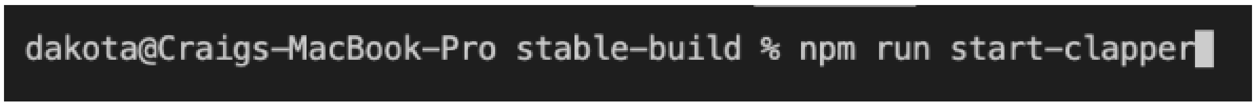
1.2. Type "npm run install" and wait for all the dependencies to download.



```
dakota@Craigs-MacBook-Pro stable-build % npm run install
```

2. Starting the UI

2.1. Inside the terminal, type "npm run start-clapper"

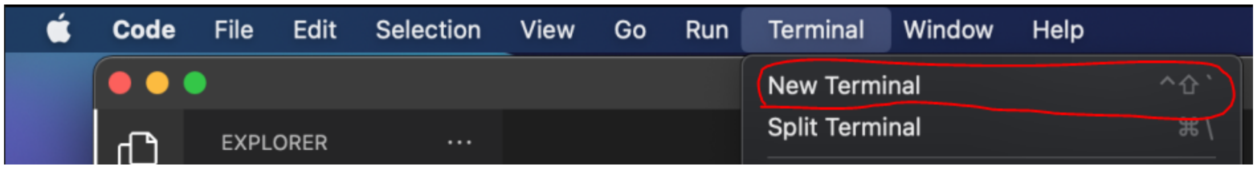


```
dakota@Craigs-MacBook-Pro stable-build % npm run start-clapper
```

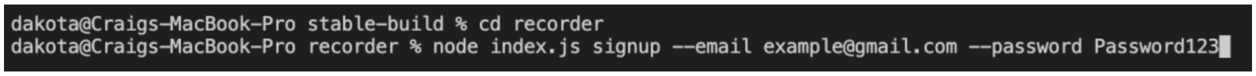
This will start the services that run the UI. After it's done compiling, it will open Code Clapper in a new browser at the address "localhost:8080". It is recommended that you open a new "private/incognito" tab and go to the address "localhost:8080" to avoid possible issues with cookies caching login sessions. You can also visit this page on another device that's connected to the same network by opening a new web browser and going to the address "HostIP:8080", substituting "HostIP" with the IP address for the machine that's running CodeClapper.

3. Starting the Recorder

3.1. While letting the other process run in the first terminal, open a new terminal by going to Terminal, then New Terminal in the menu bar.

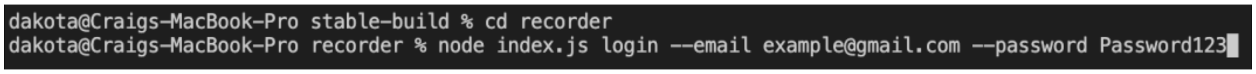


3.2. If you need to create a new account, type “cd recorder” and hit enter in the terminal, then type “node index.js signup --email YOUREMAIL --password YOURPASSWORD” (substituting whatever email and password you’d like to use) and hit enter.



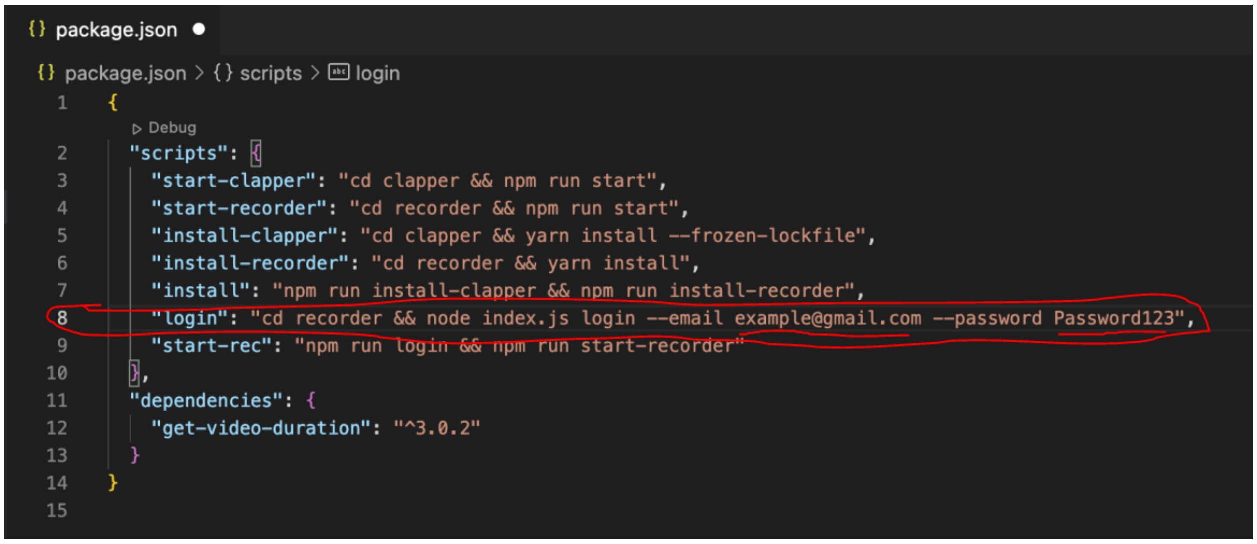
```
dakota@Craigs-MacBook-Pro stable-build % cd recorder
dakota@Craigs-MacBook-Pro recorder % node index.js signup --email example@gmail.com --password Password123
```

3.3. If you’ve previously created an account using your current Firebase Configuration, you can sign in by typing “cd /recorder” and hitting enter in the terminal, then typing “node index.js login --email YOUREMAIL --password YOURPASSWORD”, substituting your email and password for YOUREMAIL and YOURPASSWORD, respectively.



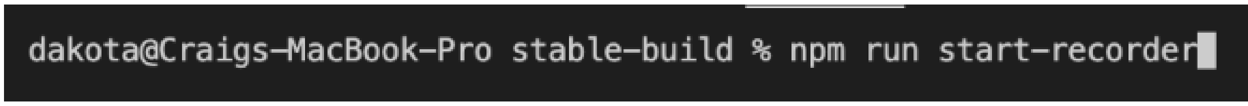
```
dakota@Craigs-MacBook-Pro stable-build % cd recorder
dakota@Craigs-MacBook-Pro recorder % node index.js login --email example@gmail.com --password Password123
```

3.4. If you already have an account created and it’s saved in the /.creds.json file, but aren’t logged in for this deployment, you can substitute your login information on line 8 in the file /stable-build/package.json and saving the file before typing “npm run start-rec” in the terminal and hitting enter.



```
{
  "scripts": {
    "start-clapper": "cd clapper && npm run start",
    "start-recorder": "cd recorder && npm run start",
    "install-clapper": "cd clapper && yarn install --frozen-lockfile",
    "install-recorder": "cd recorder && yarn install",
    "install": "npm run install-clapper && npm run install-recorder",
    "login": "cd recorder && node index.js login --email example@gmail.com --password Password123",
    "start-rec": "npm run login && npm run start-recorder"
  },
  "dependencies": {
    "get-video-duration": "^3.0.2"
  }
}
```

3.5. Once you sign in, or if you've already logged in for this build of Code Clapper, you can just type "npm run start-recorder" in the new terminal and hit enter for the recording service to start. If you receive a message like "you need to be logged in to do that!" refer back to step 3.2 or 3.3.



```
dakota@Craigs-MacBook-Pro stable-build % npm run start-recorder
```

If you've done everything successfully, the recording service is now running and your Code Clapper should have full functionality!