How To Build and Run Code Clapper V1.0

# Step 1: Downloading Necessary Software and Files

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. 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. 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/

# Step 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

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

2.2. Select "Create a project"

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

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

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.

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"

2.8. You can skip the instructions for installing firebase and its packages for now, we'll do that afterwards, keep selecting next and then "continue to console".

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

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

2.11. Enable Email/Password and click "save"

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

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

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

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

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

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

2.18. Staying in "realtime database", select the "Data" tab and then copy the url of the database to use later.

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

2.20. Under "SDK setup and configuration", 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);"

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

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

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

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

3.2. Go back to the firebase webpage and go to "realtime database" again, copy the URL for your previously created database

3.3. Create a new line under "authDomain:" and add the line "databaseURL: "URLFROMPREVIOUSSTEP", "

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

5.Copy lines 17 through 29.

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

# Step 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

3.4. Type the command “npm install -g firebase-tools”

# Step 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.

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

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 caching page and login information.

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

# Step 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.

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 already have an account created and 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.

5.4. 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.