



The cloud functions shell can be super helpful in speeding up development. *It has hot reloading* :)

It's only one part of the entire Firebase Emulator Suite, but tbh its the only part I've bothered to set up.

More about the suite: <https://firebase.google.com/docs/emulator-suite>

More about the shell inspecific: <https://firebase.google.com/docs/functions/local-shell>

The only thing you need to do is make sure you have firebase-tools installed **globally** on your machine.

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

After that, you should be all set!

cd into the /functions directory, then run npm run shell. That should run bash_scripts/run-cloud-functions-shell.sh script, which is a wrapper script that sets up the shell for you and skips a lot of annoying manual steps you'd otherwise have to do.

One thing to note is that **you must make sure that you are using the dev build**. I am very serious about this! Even though the functions you are running are local, they are still going to be affecting the actual servers they are associated with. The shell script I made doesn't protect the prod server (yet), so you have to make sure that you're using dev. The script does tell you about what you're using.

When you initially run the script, it should output something like this:

📄 Applying dev environment configuration

Be sure to pay attention because that output is temporary. If it doesn't say this, ctrl + c to kill the shell (I think you might have to do this twice) and firebase switch dev.

As for the syntax for the actual shell, refer to https://firebase.google.com/docs/functions/local-shell#invoke_https_functions

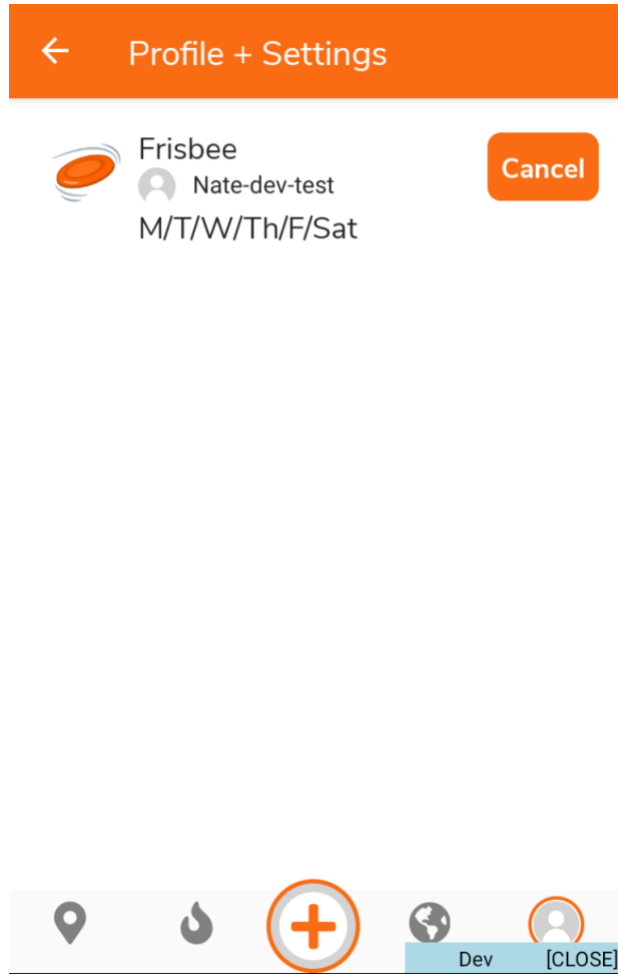
The only thing not in the docs that's worth mentioning is that if you have a callable function you're trying to run locally and it takes no arguments (ie it's something like this:)

```
export const test = functions.https.onCall(  
  async (_, __) => {  
    try{  
      ...  
      return successReport()  
    }catch(err) {  
      return handleError(err)  
    }  
  })  
);
```

You can't call it like "test()". Instead, you call it like "test({})"

Connecting the emulator to the local functions (for either QA or dev)

One of the main benefits of the shell is that apart from giving you the ability to run your functions on the terminal, it also hosts those functions on localhost (still with hot reloading!) for other processes on your computer to use.



All you gotta do is click the little Dev/QA banner at the button right and click on the “Use emulator functions” on the dialogue that comes. The banner will now show the 🧑 emoji