

## Firestore Workshop Session#3 March-2020

*Firestore workshop by Cristian Arce & Alfredo Bonilla*

---

# Session #3

## Angular Project

Clone the base project for **Session#3** from this repo into a local folder

```
$ git clone git@github.com:cristianarceGL/firebase-workshop.git
```

```
$ cd firebase-workshop
```

```
$ git checkout Session3-Base
```

```
$ git pull
```

After pulling the repo, it is mandatory to install all dependencies and run the app

```
$ npm install && npm start
```

After running this, open the browser using the url <http://localhost:4200/>, a login page must be displayed

## Firebase Functions

**Step 1:** Move to **functions** folder and execute

```
$ npm install
```

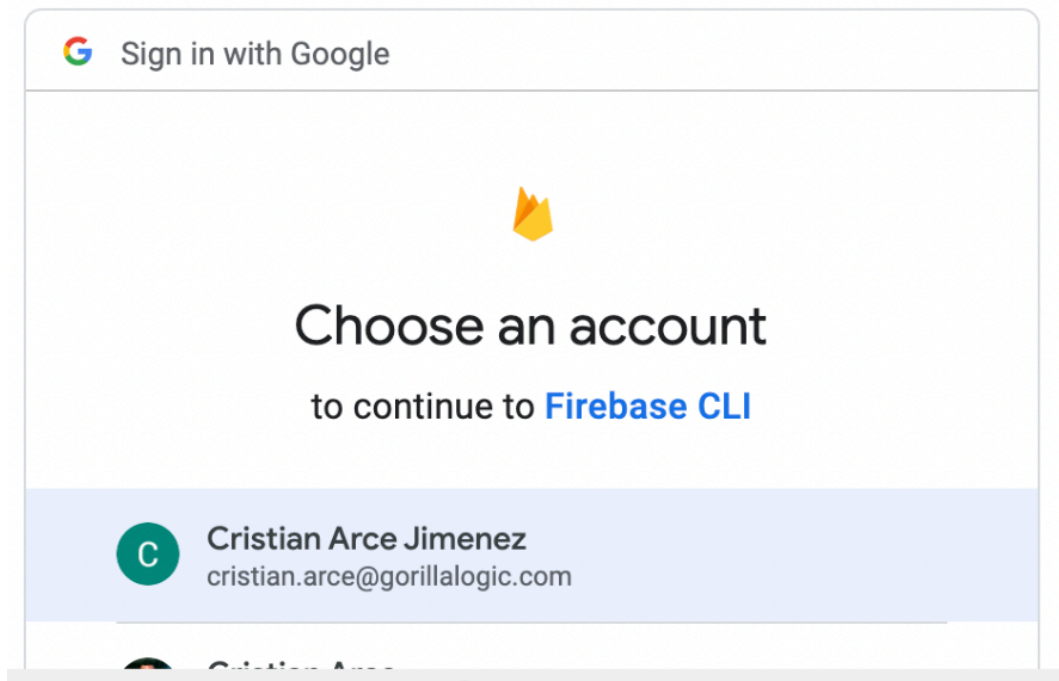
**Step 2:** It is required to have the latest version of Firebase tools, for install it globally execute

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

**Step 3:** It is also required to log into your Firebase profile; for doing this and create a reference in the current project, we execute

```
$ firebase login
```

After executing this command, a prompt window will be displayed for choosing a valid **Google Account**



The **Firebase CLI** will show the tools and features it will need to access in the profile







 Sign in with Google



## Firestore CLI wants to access your Google Account

 cristian.arce@gorillalogic.com

This will allow **Firestore CLI** to:

-  View and manage your data across Google Cloud Platform services 
-  View and administer all your Firestore data and settings 
-  View your Cloud Platform projects 

### Make sure you trust Firestore CLI

You may be sharing sensitive info with this site or app. Learn about how Firestore CLI will handle your data by reviewing its [terms of service](#) and [privacy policies](#). You can always see or remove access in your [Google Account](#).

[Learn about the risks](#)

[Cancel](#)

[Allow](#)

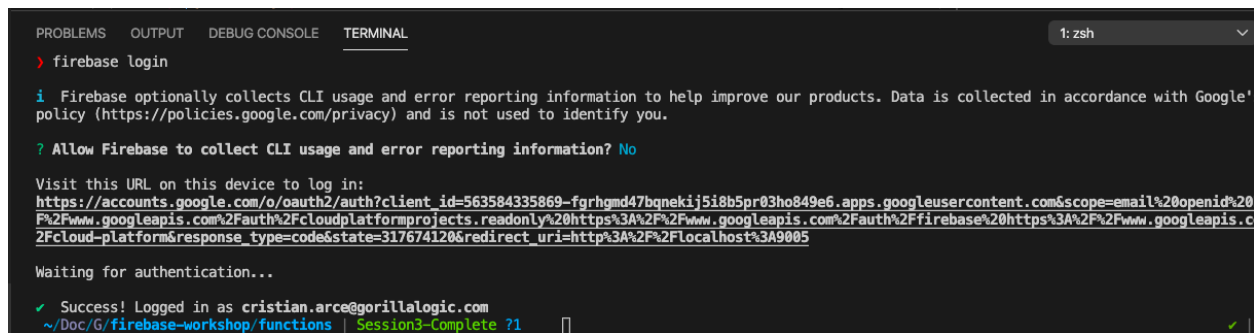
After allowing the access a successful message is displayed

Woohoo!

## Firebase CLI Login Successful

You are logged in to the Firebase Command-Line interface. You can immediately close this window and continue using the CLI.

A summary of the process result can be found in terminal



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: zsh
> firebase login

i Firebase optionally collects CLI usage and error reporting information to help improve our products. Data is collected in accordance with Google's policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI usage and error reporting information? No

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhmd47bqneki5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fcloud-platform&response_type=code&state=317674120&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...

✓ Success! Logged in as cristian.arce@gorillalogic.com
~/Doc/G/firebase-workshop/functions | Session3-Complete ?1
```

**Step 4:** Now, we need to initialize **Firebase Cloud Functions**, for that, in the terminal (while being in the new **functions** folder) type

```
$ firebase init functions
```

**Note:** Select in the 1st question = “**Typescript**”, in other questions select “**No**”

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
> firebase init functions
```

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

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

```
/Users/cristian.arce/Documents/Gorilla-Folders/firebase-workshop
```

Before we get started, keep in mind:

- \* You are initializing in an existing Firebase project directory

### === 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.

**i** Using project fir-workshop-b4991 (firebase-workshop)

### === Functions Setup

A `functions` directory will be created in your project with a Node.js package pre-configured. Functions can be deployed with `firebase deploy`.

```
? What language would you like to use to write Cloud Functions? TypeScript  
? Do you want to use TSLint to catch probable bugs and enforce style? No  
? File functions/package.json already exists. Overwrite? No  
i Skipping write of functions/package.json  
? File functions/tsconfig.json already exists. Overwrite? No  
i Skipping write of functions/tsconfig.json  
? File functions/src/index.ts already exists. Overwrite? No  
i Skipping write of functions/src/index.ts  
? File functions/.gitignore already exists. Overwrite? No  
i Skipping write of functions/.gitignore  
? Do you want to install dependencies with npm now? No
```

**i** Writing configuration info to `firebase.json`...

**i** Writing project information to `.firebaserc`...

✓ Firebase initialization complete!

~/Doc/G/firebase-workshop/functions | Session3-Complete !1 ?2

**Step 5:** In `functions/src/index.ts`, the content is override using this code below

```
import * as admin from 'firebase-admin';
import * as functions from 'firebase-functions';
import { UserRecord } from 'firebase-
functions/lib/providers/auth';

admin.initializeApp(functions.config().firebase);

// tslint:disable-next-line: no-implicit-
dependencies
const cors = require('cors')({ origin: true });
const moment = require('moment');
const firestoreInstance = admin.firestore();

exports.getDate = functions.https.onRequest((req,
res) => {
    if (req.method !== 'GET') {
        return res.status(403).send('Forbidden!');
    }
    return cors(req, res, () => {
        let format = req.query.format;
        if (!format) {
            format = req.body.format;
        }
        const formattedDate =
moment().format(format);
        res.status(200).send(formattedDate);
    });
});

exports.createUser =
```

```

functions.auth.user().onCreate((user: UserRecord,
context) => {
    return firestoreInstance
        .collection('user')
        .doc(`${user.uid}`)
        .set({
            id: user.uid,
            uid: user.uid,
            email: user.email,
            address: '**NotDefined**',
            cellphone: '**NotDefined**',
            displayName: '**NotDefined**',
            phoneNumber: '**NotDefined**',
            photoURL: '**NotDefined**',
        });
});

exports.deleteUser =
functions.auth.user().onDelete((user: UserRecord) =>
{
    return firestoreInstance
        .collection('user')
        .doc(`${user.uid}`)
        .delete();
});

```

**Step 6:** It is time to deploy the functions in Firebase, for doing that execute the following command



```
$ npm run deploy
```

This is the result in terminal

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

> npm run deploy

> fws-functions@ deploy /Users/cristian.arce/Documents/Gorilla-Folders/firebase-workshop/functions
> firebase deploy --only functions

=== Deploying to 'fir-workshop-b4991'...

i deploying functions
Running command: npm --prefix "$RESOURCE_DIR" run build

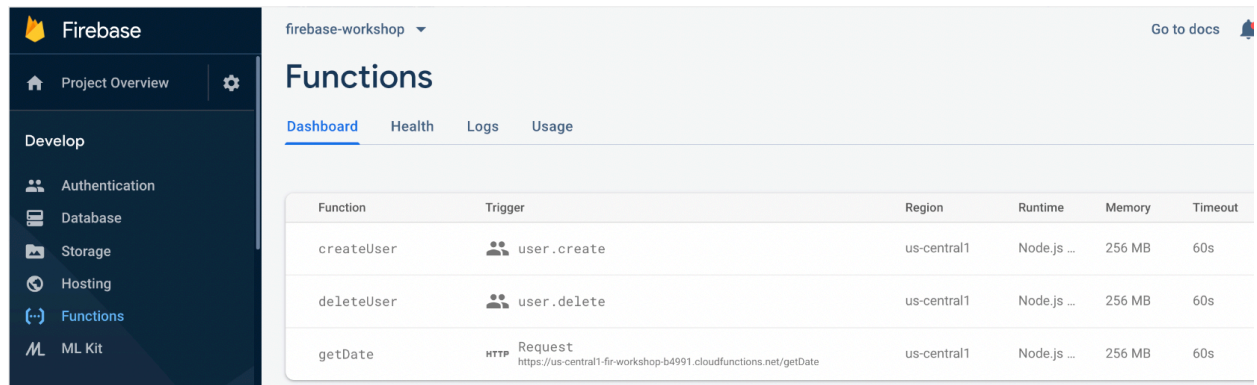
> fws-functions@ build /Users/cristian.arce/Documents/Gorilla-Folders/firebase-workshop/functions
> tsc

✓ functions: Finished running predeploy script.
i functions: ensuring necessary APIs are enabled...
✓ functions: all necessary APIs are enabled
i functions: preparing functions directory for uploading...
i functions: packaged functions (36.25 KB) for uploading
✓ functions: functions folder uploaded successfully
i functions: updating Node.js 10 (Beta) function getDate(us-central1)...
✓ functions[getDate(us-central1)]: Successful update operation.

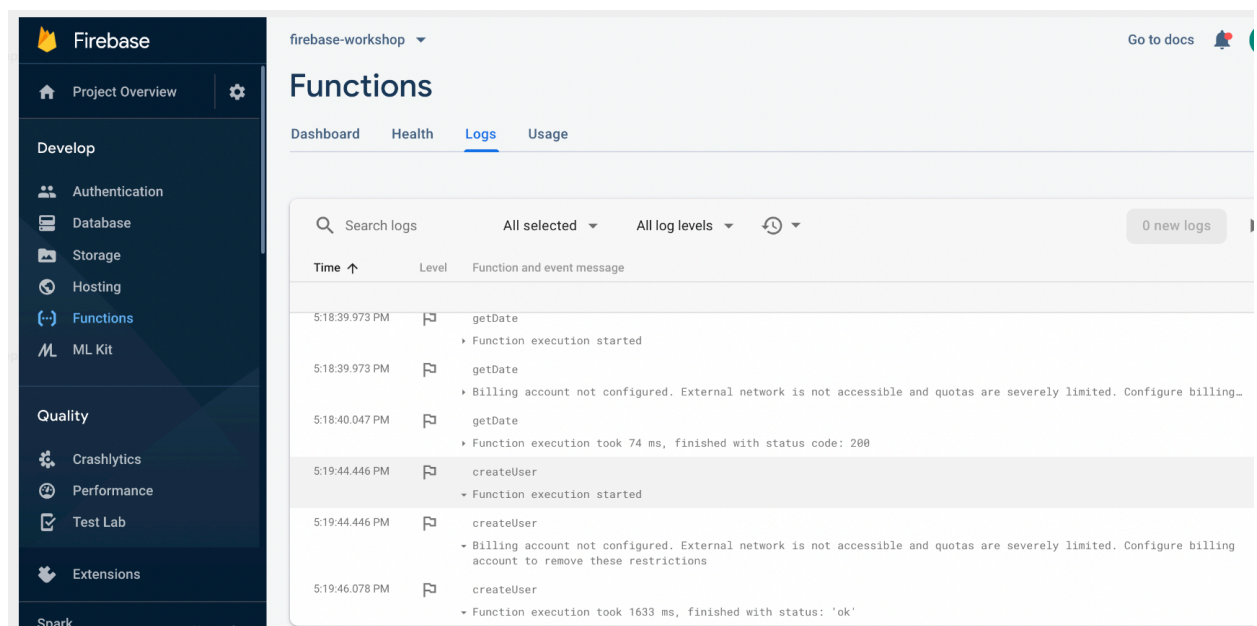
✓ Deploy complete!

Project Console: https://console.firebase.google.com/project/fir-workshop-b4991/overview
~/Doc/G/firebase-workshop/functions | Session3-Complete !1 ?1
```

**Step 7:** After deploying, in the web profile, navigate to **Functions/Dashboard**, there it can be found all the deployed functions for the project



**Step 8:** Also in **Functions/Logs**, it is possible to track the steps followed in the execution plan of every single function

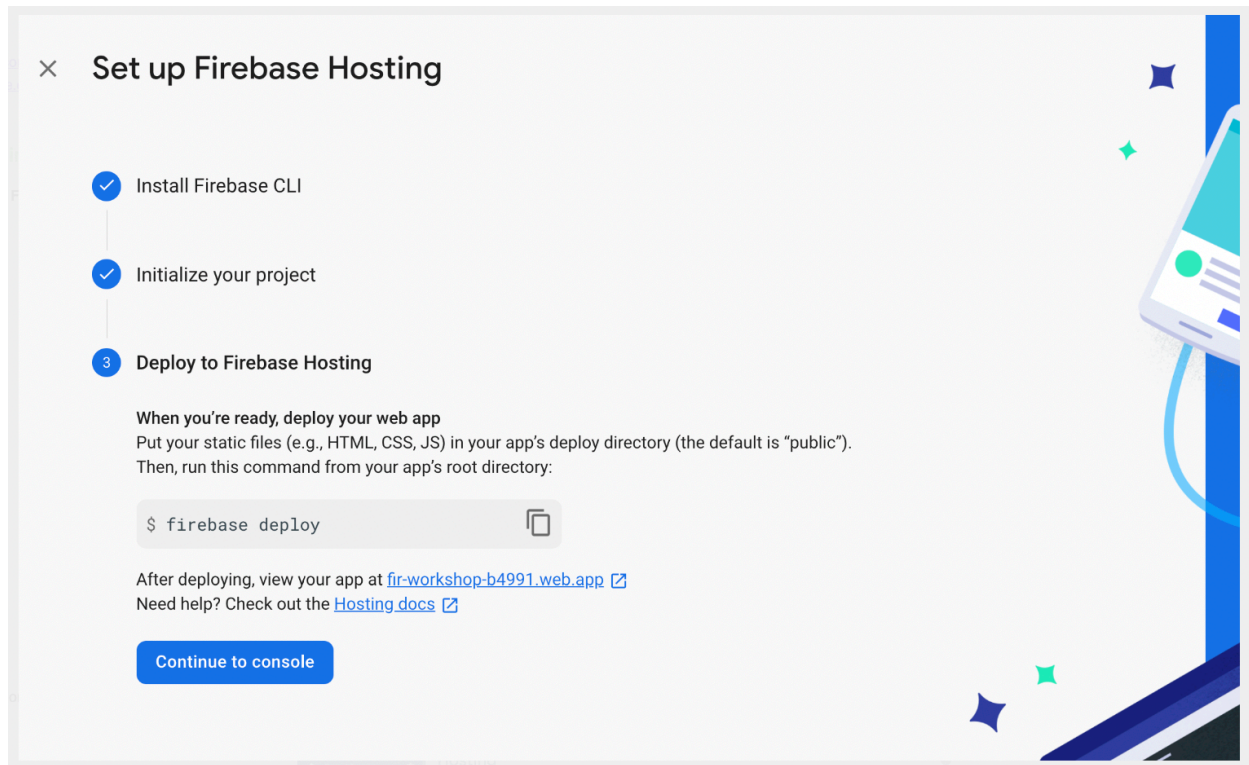


For more information about functions, samples and auth events

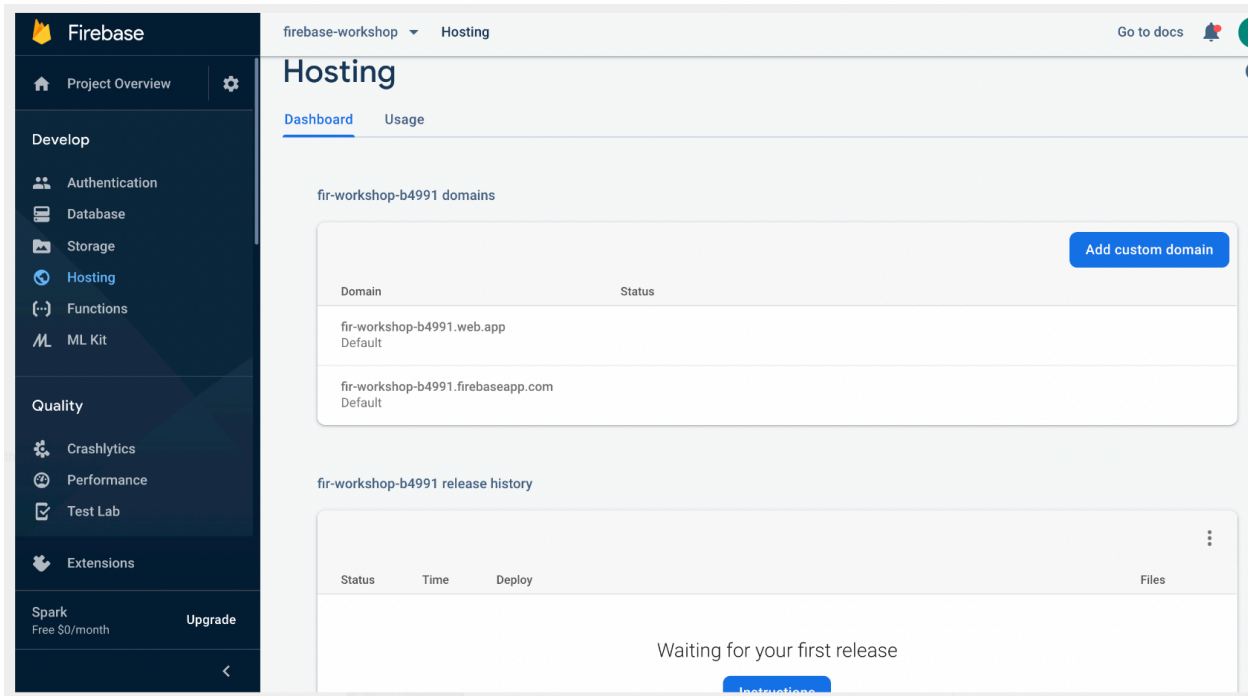
- <https://firebase.google.com/docs/functions/use-cases>
- <https://github.com/firebase/functions-samples>
- <https://firebase.google.com/docs/functions/auth-events>

## Firestore Hosting

**Step 1:** Navigate to **Firestore/Hosting** option, and start the set up using the default values



When the setup is done, the **Hosting** is ready to get deploys



**Step 2:** Move to **root** folder and execute

```
$ firebase deploy --only hosting:target-name
```

**Note:** the **target-name** is the unique alphanumeric identifier of the project, it can be found in **Firebase/Hosting/Dashboard** next to **domain**

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

=== Deploying to 'fir-workshop-b4991'...

i  deploying hosting
i  hosting[fir-workshop-b4991]: beginning deploy...
i  hosting[fir-workshop-b4991]: found 30 files in dist/firebaseSessions
✓  hosting[fir-workshop-b4991]: file upload complete
i  hosting[fir-workshop-b4991]: finalizing version...
✓  hosting[fir-workshop-b4991]: version finalized
i  hosting[fir-workshop-b4991]: releasing new version...
✓  hosting[fir-workshop-b4991]: release complete


✓  Deploy complete!

Project Console: https://console.firebase.google.com/project/fir-workshop-b4991/overview
Hosting URL: https://fir-workshop-b4991.firebaseio.com
~/Doc/G/firebase-workshop | Session3-Base !1  □
```

**PS:** in case of wanting to avoid possible reference error, there is a npm script for installing dependencies, build the app and deploy to Firebase host

```
$ npm run deploy
```

**Step 3:** After a succeeded deploy, in **Firestore/Hosting/Dashboard** the hosting already give access to a couple domains for public access, there is even an option for adding a custom domain, for this workshop purposes the default ones are good to go

 **Firebase**

Project Overview

Develop

Quality

Spark

Authentication

Database

Storage

Hosting

Functions

ML Kit

Crashlytics

Performance

Test Lab

Extensions

Free \$0/month

Upgrade

firebase-workshop

Go to docs

Hosting

DashboardUsage

fir-workshop-b4991 domains

Add custom domain

Domain	Status
<a href="#">fir-workshop-b4991.web.app</a> Default	
fir-workshop-b4991.firebaseio.com Default	

fir-workshop-b4991 release history

Status	Time	Deploy	Files
<b>Current</b>	Feb 27, 2020 11:45 AM	<a href="#">cristian.arce@gorillalogic.com</a> aa3110	32