Building APIs with Firebase

By: Andrew Evans

Agenda

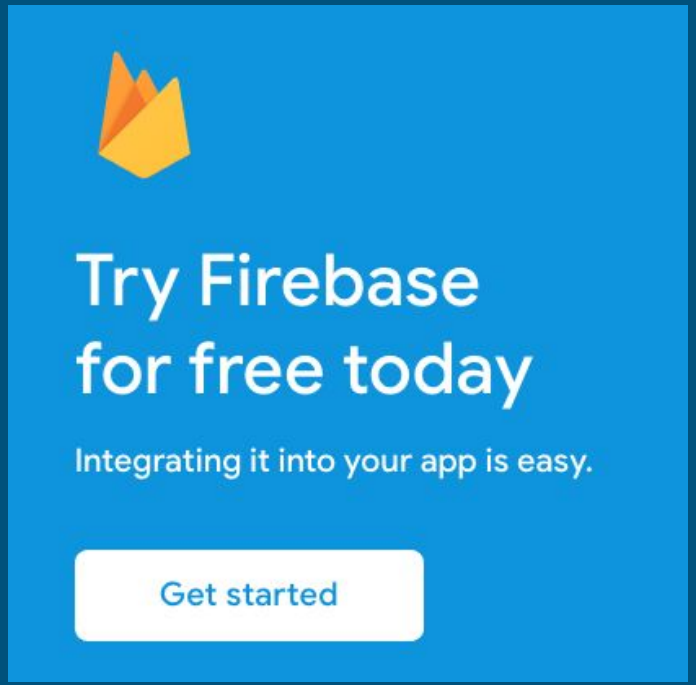
- Intro
- What is Firebase?
- Firebase Functions
- Firebase Functions to APIs
- Testing
- Deploying
- LIVE CODING
- Next Steps
- Questions

Intro

- Manager at CapTech Consulting
- 10+ years industry experience
- Full Stack
  - Frontend React, Angular, VueJS, more
  - Backend JavaScript, Java, learning C# and more
- Masters in CS from CNU
- MBA from W&M

What is Firebase?

- Platform that provides tools and services to quickly and easily build software
- Realtime Database
- Hosting
- Authentication
- Functions (HTTP)
- and more!



Firebase Functions

- Serverless Framework, exposes backend code via HTTP endpoints
- Sits on top of Google Cloud Functions

```
// when a new user is registered
exports.createUser = functions.firestore
  .document("users/{userId}")
  .onCreate((snap, context) => {
    const newValue = snap.data();
    const firstName = newValue.firstName;
    const lastName = newValue.lastName;
    const slackWebhook = process.env.CREATE_USER;
    const message = "user " + firstName + " " + lastName + " just registered!";

    request
      .post(slackWebhook, { json: { text: message } })
      .then(() => {
        return res.status(200).send("slack message sent successfully");
      })
      .catch(() => {
        return res.status(500).semd("error occured whens ending slack message");
      });
  });

```

```
app.get('/api/select-all', (req, res) => {
    (async () => {
        try {
            const allRecords = await firestore.selectAll(db);
            res.status(200).send(allRecords);
        } catch (error) {
            console.log(error);
            return res.status(500).send(error);
        }
    })();
});

```

Firebase Functions to APIs

- Created instance of ExpressJS app
- Wrap in Firebase Function

```
1   const functions = require('firebase-functions');
2   const express = require('express');
3   const cors = require('cors');
4   const app = express();
5   app.use(cors({ origin: true }));
6
7   app.get('/api/hello-world', (req, res) => {
8       res.status(200).send('hello world');
9   });
10
11  exports.app = functions.https.onRequest(app);
12
13  // when deployed you can access this
14  // https://<application_id>.cloudfunctions.net/app/hello-world
```

Testing Functions

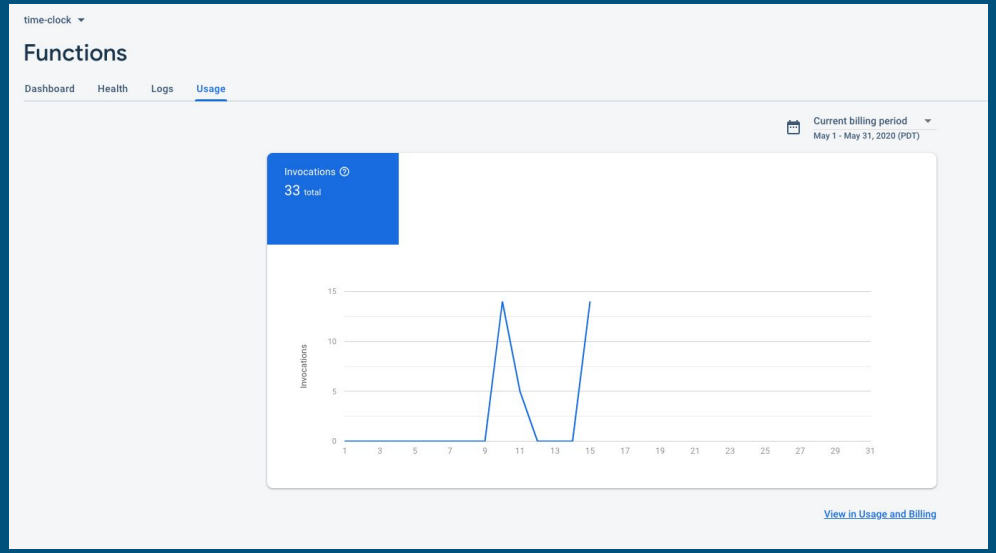
- Firebase Emulator
- Mocha Support

```
i  firestore: firestore emulator logging to firestore-debug.log
✔  firestore: firestore emulator started at http://localhost:8080
i  firestore: For testing set FIRESTORE_EMULATOR_HOST=localhost:8080
i  Running script: mocha --timeout=10000 test/firestore-tests.js


  The timeclock API should
    ✓ enable you to get length when 0 (203ms)
    ✓ enable you to get length when clock in exists (132ms)
    ✓ enable you to clock in and then call select all (61ms)
    ✓ enable you to clock out successfully (99ms)
    ✓ enable you to clock in and out successfully multiple times (162ms)
    ✓ enable you to delete all records after created (143ms)


  6 passing (1s)
```

```
1   const firebase = require('@firebase/testing');
2   const projectId = '1234';
3   const chai = require('chai');
4   const chaiHttp = require('chai-http');
5   chai.use(chaiHttp);
6   const expect = chai.expect;
7   const firestore = require('../firestore/firestore');
8
9   function setupDB() {
10      return firebase.initializeTestApp({ projectId }).firestore();
11  }
12
13  beforeEach(async () => {
14      // Clear the database between tests
15      await firebase.clearFirestoreData({ projectId });
16  });
17
18  describe('My app', () => {
19      it('should enable you to get length when 0', async () => {
20          const db = setupDB();
21          let length = await firestore.getLength(db);
22          expect(length).to.deep.equal(0);
23      });
24  });
```

Deploying Functions

- Firebase CLI does all the work for you!
- Packages app and deploys within Google Cloud




```
i  deploying functions
i  functions: ensuring required API cloudfunctions.googleapis.com is enabled...
✔  functions: required API cloudfunctions.googleapis.com is enabled
i  functions: preparing . directory for uploading...
i  functions: packaged . (65.7 KB) for uploading
✔  functions: . folder uploaded successfully
i  functions: updating Node.js 10 function app(us-central1)...
✔  functions[app(us-central1)]: Successful update operation.

✔  Deploy complete!
```

Sample Project

- Created an API that functions as a timeclock
  - GET “/hello-world”
  - POST “/clock-in”
  - PUT “/clock-out”
  - GET “/select-all”
  - DELETE “/delete-all”
- Includes tests with Firebase Emulator
- Project includes both an initial and completed setup
- Deploy using the Firebase CLI
- Will be using Postman to demonstrate both local and hosted application

LIVE CODING

Next Steps

- Create a free Firebase account
- Build out some basic functions
- Check out the MANY other services Firebase offers
- Have fun!

QUESTIONS