# Firebase

firebase.google.com

https://github.com/kunzleigh/kla-firebase-poker

KL&A

KUNZ, LEIGH AND ASSOCIATES

# What is Firebase?

# What is Firebase?

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

In a nutshell, Firebase is a cost-effective platform provided by Google to allow rapid development with many features that can scale without worry.

# Services

# Services

- Analytics
- Authentication
- Database
- Storage
- Hosting
- Cloud Functions
- Test Lab
- Crash Reporting
- Notification
- Remote Config
- Dynamic Links
- AdMob
- Cloud Messaging
- AdWords
- App Indexing

# Authentication

- Provides backend services
- Easy-to-use SDKs
- Ready-made UI libraries

# Authentication - Support

- Passwords
- Phone numbers
- Popular federated identity providers like Google, Facebook and Twitter, and more.
- Leverages industry standards like OAuth 2.0 and OpenID Connect, allowing for easy integration.

# Database

- Cloud-hosted database
- Stored as JSON
- Synchronized in real-time to every connected client
- Works offline
- All clients on all platforms share the same real-time database
- Secured through existing authentication (no extra setup needed)

# Storage

- Robust
  - Automatically managed uploads / downloads regardless of network quality.
  - Supports download / upload resume.
- Strong security
- Highly scalable
  - Easily go from prototype to production without worry of capacity or infrastructure

# Hosting

- Fast and secure static hosting
- Global content-delivery network (CDN)
- Zero-configuration SSL
- Single command deploy, up and running in seconds
- One-click rollbacks

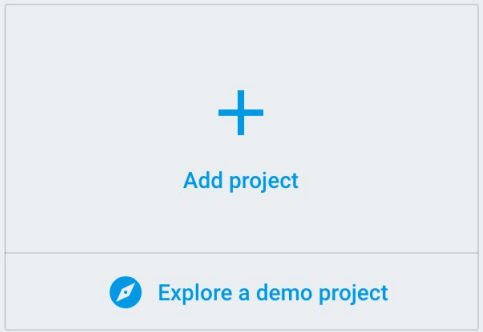
# Cloud Functions

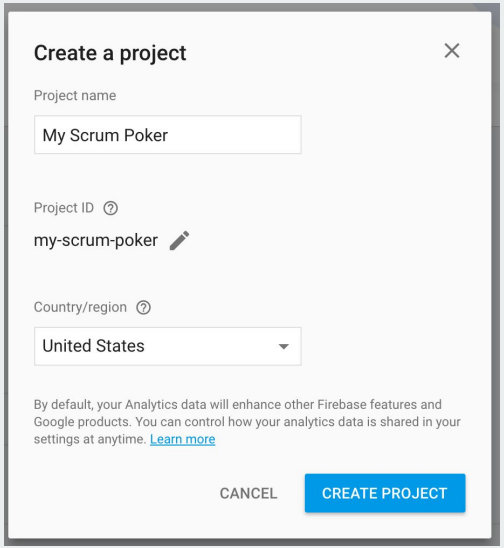
- Real-time Database Triggers
- Firebase Authentication Triggers
- Google Analytics for Firebase Triggers
- Cloud Storage Triggers
- Cloud Pub/Sub Triggers
- HTTP Triggers

# Let’s Build

# Project Setup

- https://console.firebase.google.com/
- Add Project
- Create Project



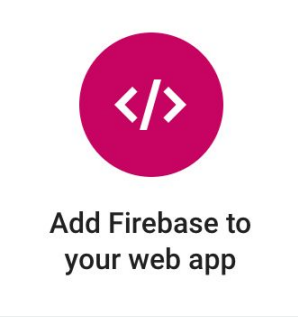

# Project Setup

git clone https://github.com/kunzleigh/kla-firebase-poker.git

cd kla-firebase-poker

npm install

# Project Setup

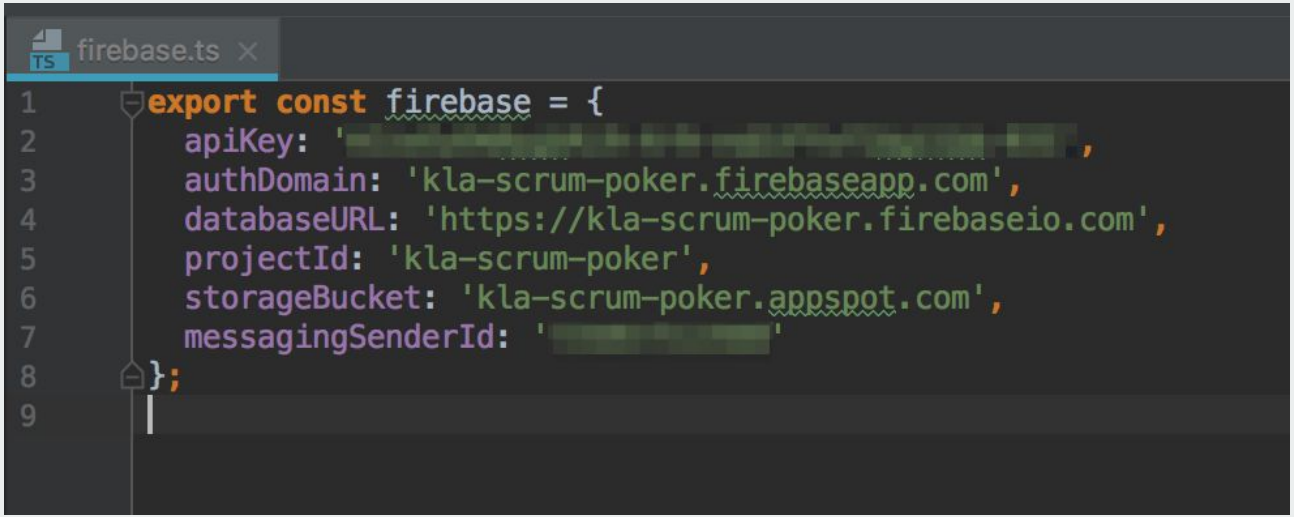




- src/configs/firebase.ts

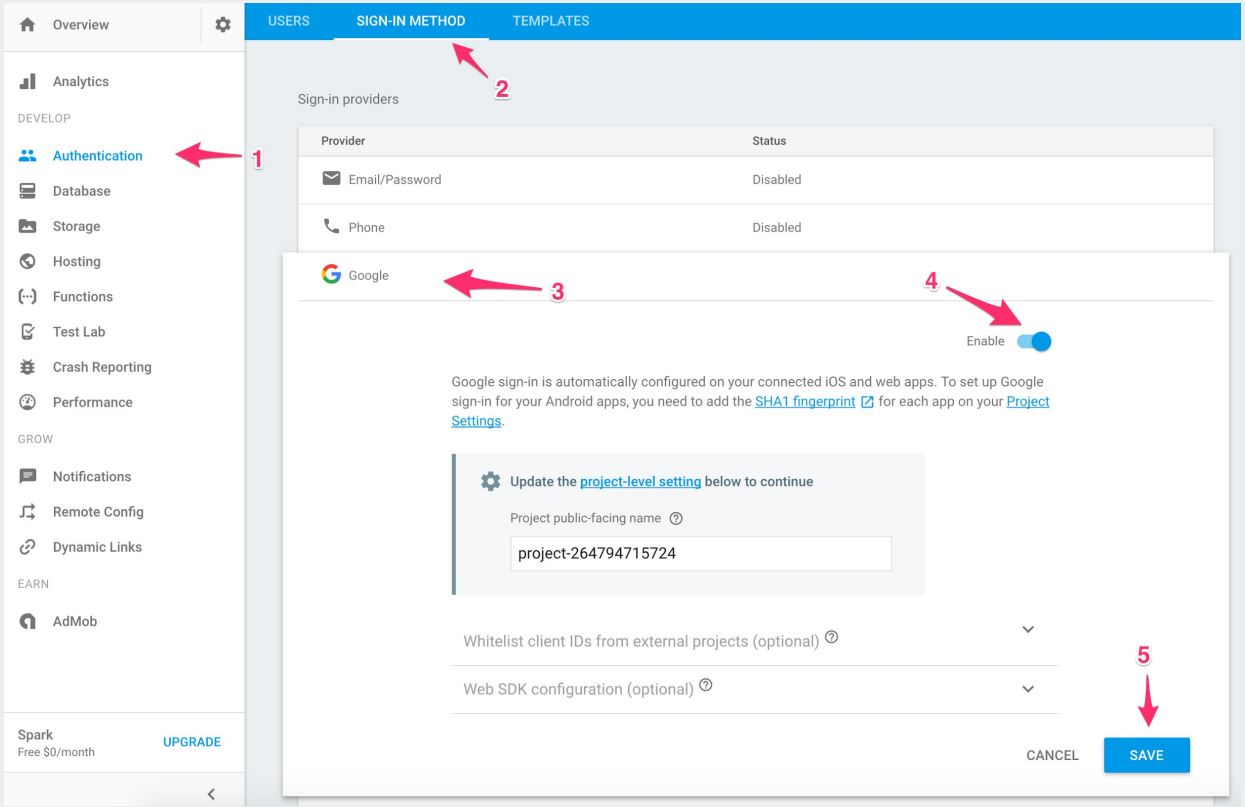
# Project Setup

- src/configs/firebase.ts

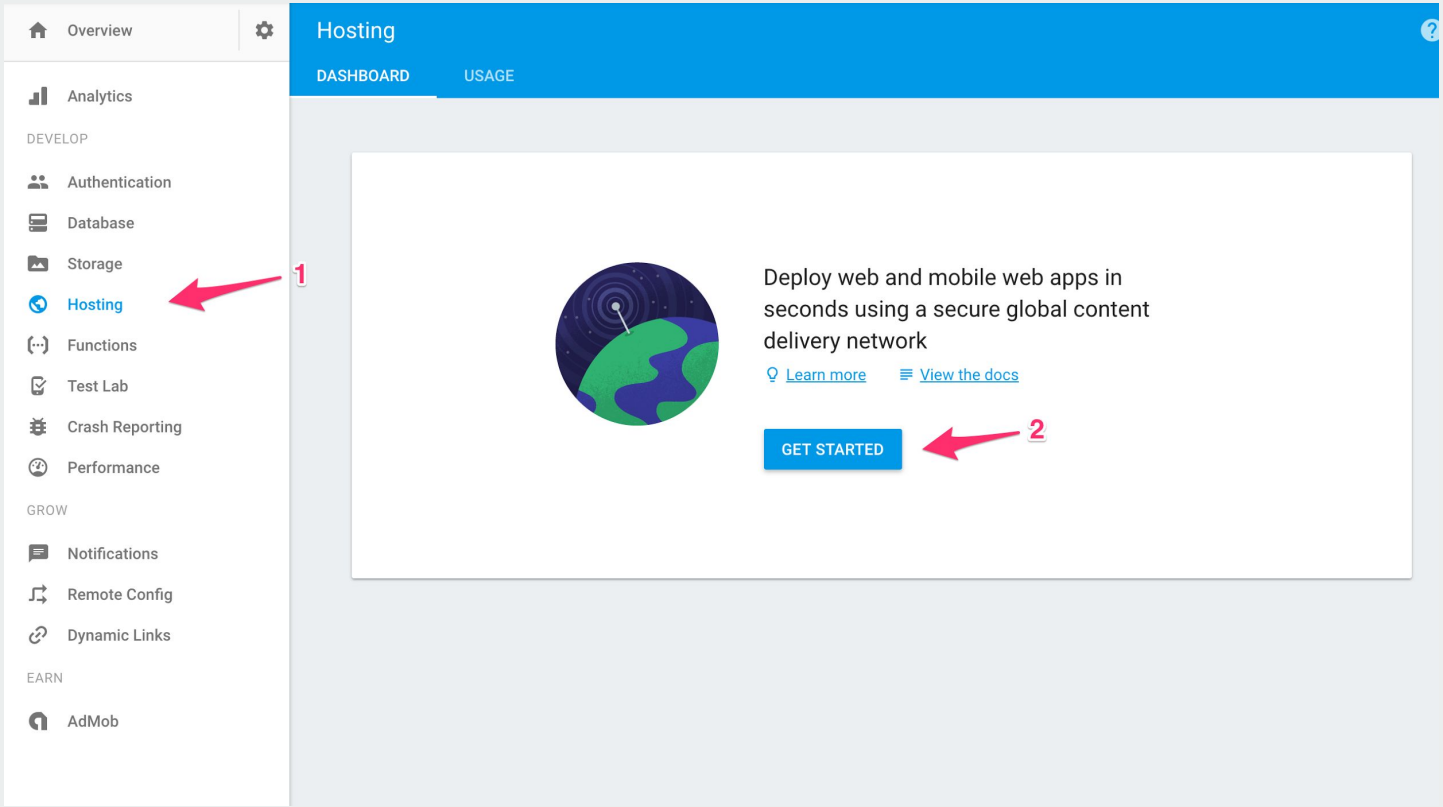



- git update-index --assume-unchanged src/configs/firebase.ts

# Project Setup




# Project Setup



# Project Setup

- ng build

```
Date: 2017-09-24T01:31:55.754Z
Hash: fc0f75f34788030e407a
Time: 42137ms
chunk {inline} inline.bundle.js, inline.bundle.js.map (inline) 5.83 kB [entry] [rendered]
chunk {main} main.bundle.js, main.bundle.js.map (main) 81.7 kB {vendor} [initial] [rendered]
chunk {polyfills} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 209 kB {inline} [initial] [rendered]
chunk {styles} styles.bundle.js, styles.bundle.js.map (styles) 57.3 kB {inline} [initial] [rendered]
chunk {vendor} vendor.bundle.js, vendor.bundle.js.map (vendor) 5.73 MB [initial] [rendered]
$
```

# Project Setup

- firebase login

```
$ firebase login
? Allow Firebase to collect anonymous CLI usage information? Yes

Visit this URL on any device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=[redacted]

Waiting for authentication...

✔  Success! Logged in as [redacted]
$
```

# Project Setup

- firebase init

```
$ firebase init

     ######## #### ########  ######## ########     ###     ######  ########
     ##        ##  ##     ## ##       ##     ##  ##   ##  ##       ##
     ######    ##  ########  ######   ########  #########  ######  ######
     ##        ##  ##    ##  ##       ##     ## ##     ##       ## ##
     ##       #### ##     ## ######## ########  ##     ##  ######  ########

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

  [redacted]

Before we get started, keep in mind:

  * You are initializing in an existing Firebase project directory

? Which Firebase CLI features do you want to setup for this folder? Press Space to select features, then Enter to confirm your c
hoices.
 ◉ Database: Deploy Firebase Realtime Database Rules
 ◉ Functions: Configure and deploy Cloud Functions
 ◉ Hosting: Configure and deploy Firebase Hosting sites
```

# Project Setup

- firebase init - continued...

```
    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.

? Select a default Firebase project for this directory:
  [don't setup a default project]
❯ My Scrum Poker (my-scrum-poker)
```

# Project Setup

- firebase init - continued...

```
? Select a default Firebase project for this directory: My Scrum Poker (my-scrum-poker)

    Database Setup

Firebase Realtime Database Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Database Rules? database.rules.json
? File database.rules.json already exists. Do you want to overwrite it with the Database Rules for my-scrum-poker from the Firebas
e Console? No
Skipping overwrite of Database Rules.
The rules defined in database.rules.json will be published when you do firebase deploy.

    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.

? File functions/package.json already exists. Overwrite? No
i  Skipping write of functions/package.json
? File functions/index.js already exists. Overwrite? No
i  Skipping write of functions/index.js
? Do you want to install dependencies with npm now? Yes
up to date in 2.421s

    Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? dist
? Configure as a single-page app (rewrite all urls to /index.html)? Yes
? File dist/index.html already exists. Overwrite? No
i  Skipping write of dist/index.html

i  Writing configuration info to firebase.json...
i  Writing project information to .firebaserc...

✔  Firebase initialization complete!
$
```

# Project Setup

- firebase use

```
$ firebase use
Active Project: my-scrum-poker-ee990

Project aliases for /Users/jason/Development/kla/temp/kla-scrum-poker:

* default (my-scrum-poker-ee990)

Run firebase use --add to define a new project alias.
$
```

# Project Setup

- firebase deploy

```
$ firebase deploy

    Deploying to 'my-scrum-poker-ee990'...

i  deploying database, functions, hosting
✔  database: rules ready to deploy.
i  functions: ensuring necessary APIs are enabled...
i  runtimeconfig: ensuring necessary APIs are enabled...
✔  runtimeconfig: all necessary APIs are enabled
✔  functions: all necessary APIs are enabled
i  functions: preparing functions directory for uploading...
i  functions: packaged functions (14.7 KB) for uploading
✔  functions: functions folder uploaded successfully
i  hosting: preparing dist directory for upload...
✔  hosting: 14 files uploaded successfully
i  starting release process (may take several minutes)...
i  functions: creating function setupNewUser...
✔  functions[setupNewUser]: Successful create operation.
✔  functions: all functions deployed successfully!

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/my-scrum-poker-ee990/overview
Hosting URL: https://my-scrum-poker-ee990.firebaseapp.com
$
```

# Build - Hosting

- That’s it. Your app is now hosted.

```
✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/my-scrum-poker-ee990/overview
Hosting URL: https://my-scrum-poker-ee990.firebaseapp.com
$
```

# Build - Authentication

- src/app/service/auth.service.ts

```
/**
 * Method for specifically logging in with Google
 */
loginWithGoogle() {
  const provider = (new auth.GoogleAuthProvider()).setCustomParameters({prompt: 'select_account'});

  this.angularFireAuth
    .auth
    .signInWithPopup(provider)
    .catch( onReject: e => {
      console.log('Ground control to major tom, your circuits dead there is something wrong', e.message);
    });
}
```

# Build - Database

- Data

https://kla-scrum-poker.firebaseio.com/

```
kla-scrum-poker
  tickets
    -KukUtru14ndPcQqXlAN
      description: "ticket 1 description"
      stats
      title: "ticket 1"
      votes
    -Kuk_BeRlDqSLWbaMgbB
    -KuvmwXmDTPsxOK85pTT
    -KuvszxvnGNKY4HDQ-X4
    -KuvtHi_DBplkAioYTuw
  users
```

# Build - Database

- Rules

SIMULATOR

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

    "users": {
      "$userId": {
        ".read": "auth != null",
        ".write": "auth != null && auth.uid == $userId",
        "votes": {
          "$voteId": {
            ".validate": "newData.hasChild('ticketId') && newData.child('ticketId').isString() && newData.hasChild('value') && newDa
          }
        },
```

# Build - Database

- get
  - Returns data below path
- push
  - Adds a new node to path
- set
  - Replaces node at path
- remove
  - Removes node at path
- update
  - Non-destructive replacement at path

# Build - Database

List

```
this.ticketList$ = this.angularFireDatabase.list( pathOrRef: '/tickets');
```

Object

```
getCurrentTicket(ticketId: string): Promise<boolean> {
  return new Promise<boolean>((resolve) => {
    this.currentTicket$ = this.angularFireDatabase.object( pathOrRef: '/tickets/' + ticketId);
    this.voteService.getCurrentVote(ticketId).then( onfulfilled: () => {
      resolve(true);
    });
  });
}
```

# Build - Cloud Functions

- Database Triggers

```
function voteSync(event, isCreate) {
  if (event.data.exists()) {
    const $uid = event.params.$uid;
    const data = event.data.val();
    const $ticketId = data.ticketId;

    if (isCreate) {
      data.created = new Date().toISOString();
      data.createdBy = $uid;
    } else {
      data.lastModified = new Date().toISOString();
      data.lastModifiedBy = $uid;
    }

    return admin.database().ref('/tickets/' + $ticketId + '/votes/' + $uid).set(data);
  }
}
```

# Build - Cloud Functions

- Authentication Triggers

```
exports.setupNewUser = functions.auth.user().onCreate(event => {
  const user = event.data;

  admin.database().ref('/users/' + user.uid).set({
    name: user.displayName,
    created: new Date().toISOString(),
    createdBy: user.uid,
    isAdmin: user.email.split('@')[1] === 'kunzleigh.com',
  });
});
```

# Build - Cloud Functions

- HTTP Triggers

```
exports.ticketStats = functions.https.onRequest((req, res) => {
  // Result object that will be returned
  const result = {
    totalTickets: 0,
  };

  // This will not scale well. A database trigger would be better.
  admin.database().ref('/tickets').once("value").then(function(snapshot) {
    const collection = snapshot.val();

    // Count of tickets
    result.totalTickets = Object.keys(collection).length;

    res.send(result);
  });

});
```

# Build - Storage

```
/**
 * Uploads the user image to the storage location
 * @param {Upload} upload
 */
upload(upload: Upload) {
  this.uploadTask = this.storageRef.child(upload.path + '/' + upload.name).put(upload.file);

  this.uploadTask.on(firebase.storage.TaskEvent.STATE_CHANGED,
    (snapshot: any) => {
      // upload in progress
      this.uploadProgress.next( value: (snapshot.bytesTransferred / snapshot.totalBytes) * 100);
    },
    (error) => {
      // upload failed
      console.log(error);
    },
    () => {
      // upload finished successfully
      this.uploadFinished.next(this.uploadTask.snapshot.downloadURL);
    }
  );
}
```

# Beta - Cloud Firestore

- Flexibility
  - Permission dependencies
- Expressive Querying
  - Full sorting support
  - Muli parameters
- Realtime Updates
  - Same, but enhanced
- Offline Support
  - Same, also enhanced
- Designed to Scale
  - Multi-region data replication
  - Transaction support

# Questions?

# Firebase

firebase.google.com

KL&A

KUNZ, LEIGH AND ASSOCIATES