

 무료 전자 책

배우기

firebase

Free unaffiliated eBook created from
Stack Overflow contributors.

#firebase

| | |
|---------------------------------------|-----------|
| | 1 |
| 1: firebase | 2 |
| | 2 |
| | 2 |
| Examples..... | 2 |
| Android Firebase | 2 |
| Firestore | 2 |
| SDK | 3 |
| IOS Firebase | 4 |
| Hello World Firestore | 11 |
| | 11 |
| 2: Firestore Firestore | 17 |
| Examples..... | 17 |
| | 17 |
| ?..... | 17 |
| | 17 |
| 3: Firestore ? | 19 |
| | 19 |
| Examples..... | 19 |
| Android | 19 |
| 4: Firestore | 20 |
| | 20 |
| | 20 |
| Examples..... | 20 |
| (Android / iOS)..... | 20 |
| (Android / iOS)..... | 21 |
| 5: Firestore | 22 |
| | 22 |
| Examples..... | 22 |
| | 22 |
| Firestore | 22 |

| | |
|-------------------------------|-----------|
| Firestore CLI | 22 |
| Google Cloud | 22 |
| | 23 |
| 6: Firebase | 24 |
| | 24 |
| | 24 |
| | 24 |
| Examples..... | 24 |
| Firestore All In One..... | 24 |
| 7: FirebaseUI | 25 |
| | 25 |
| Examples..... | 25 |
| FirebaseUI | 25 |
| 8: FirebaseUI () | 27 |
| Examples..... | 27 |
| | 27 |
| ListView | 27 |
| 9: | 29 |
| | 29 |
| | 29 |
| | 29 |
| Examples..... | 29 |
| - - AngularFire..... | 29 |
| 10: Firebase | 31 |
| Examples..... | 31 |
| Hello World Firebase | 31 |
| Firebase | 32 |
| 11: | 35 |
| | 35 |
| Examples..... | 35 |
| | 35 |
| | |

| | |
|---|-----------|
| 12: | 38 |
| | 38 |
| | 38 |
| | 38 |
| | 38 |
| Examples..... | 38 |
| | 38 |
| | 38 |
| | 38 |
| | 38 |
| | 39 |
| | 39 |
| 13: ? | 40 |
| | 40 |
| Examples..... | 40 |
| Android | 40 |
| Android | 40 |
| iOS | 41 |
| JavaScript | 41 |
| JavaScript | 41 |
| iOS | 41 |
| 14: | 43 |
| | 43 |
| Examples..... | 43 |
| Firebase HTTP | 43 |
| Admin SDK (js) | 43 |
| 15: Firbase | 45 |
| Examples..... | 45 |
| | 45 |
| 16: RecyclerView FirebaseRecyclerViewAdapter ? | 46 |
| Examples..... | 46 |
| FirebaseUi FirebaseRecyclerViewAdapter..... | 46 |

| | |
|------------------|-----------|
| 17: | 52 |
| | 52 |
| | 52 |
| Examples..... | 52 |
| Android | 52 |
| Android | 52 |
| 18: | 53 |
| | 53 |
| Examples..... | 53 |
| iOS | 53 |
| | 53 |
| Firebase | 53 |
| Firebase | 53 |
| 19: | 55 |
| Examples..... | 55 |
| firebase | 55 |
| | 55 |
| | 55 |
| | 58 |

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [firebase](#)

It is an unofficial and free firebase ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official firebase.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: firebase

Firestore Backend as a Service (Baas) .

, , , AdMod

Android, iOS, Web, NodeJS, C ++ Java Server SDK .

| SDK | | |
|-----------------------------|--------|------------|
| Firestore JavaScript SDK | 3.7.0 | 2017-03-01 |
| Firestore C ++ SDK | 3.0.0 | 2017-02-27 |
| Firestore Unity SDK | 3.0.0 | 2017-02-27 |
| Firestore iOS SDK | 3.14.0 | 2017-02-23 |
| Firestore Android SDK | 10.2 | 2017-02-15 |
| Firestore Admin Node.js SDK | 4.1.1 | 2017-02-14 |
| Firestore Admin Java SDK | 4.1.2 | 2017-02-14 |

Examples

Android Firestore

Firestore Android .

Firestore

1. [Firestore](#) Firestore **Create New Project** .
2. **Android Firestore** .
3. .
Firestore .
4. **Auth** , **Google** SHA1 Android Studio Gradle Refresh project (root) -> Tasks -> android -> signingReport . Run **MD5 SHA1** . SHA1 firebase .
5. google-services.json . .
6. (app /) .



1. build.gradle **google-services** .

```
buildscript {
    // ...
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:3.0.0'
    }
}
```

Gradle (app/build.gradle) Gradle .

```
apply plugin: 'com.android.application'

android {
    // ...
}

dependencies {
    // ...
    compile 'com.google.firebase:firebase-core:9.4.0'
}

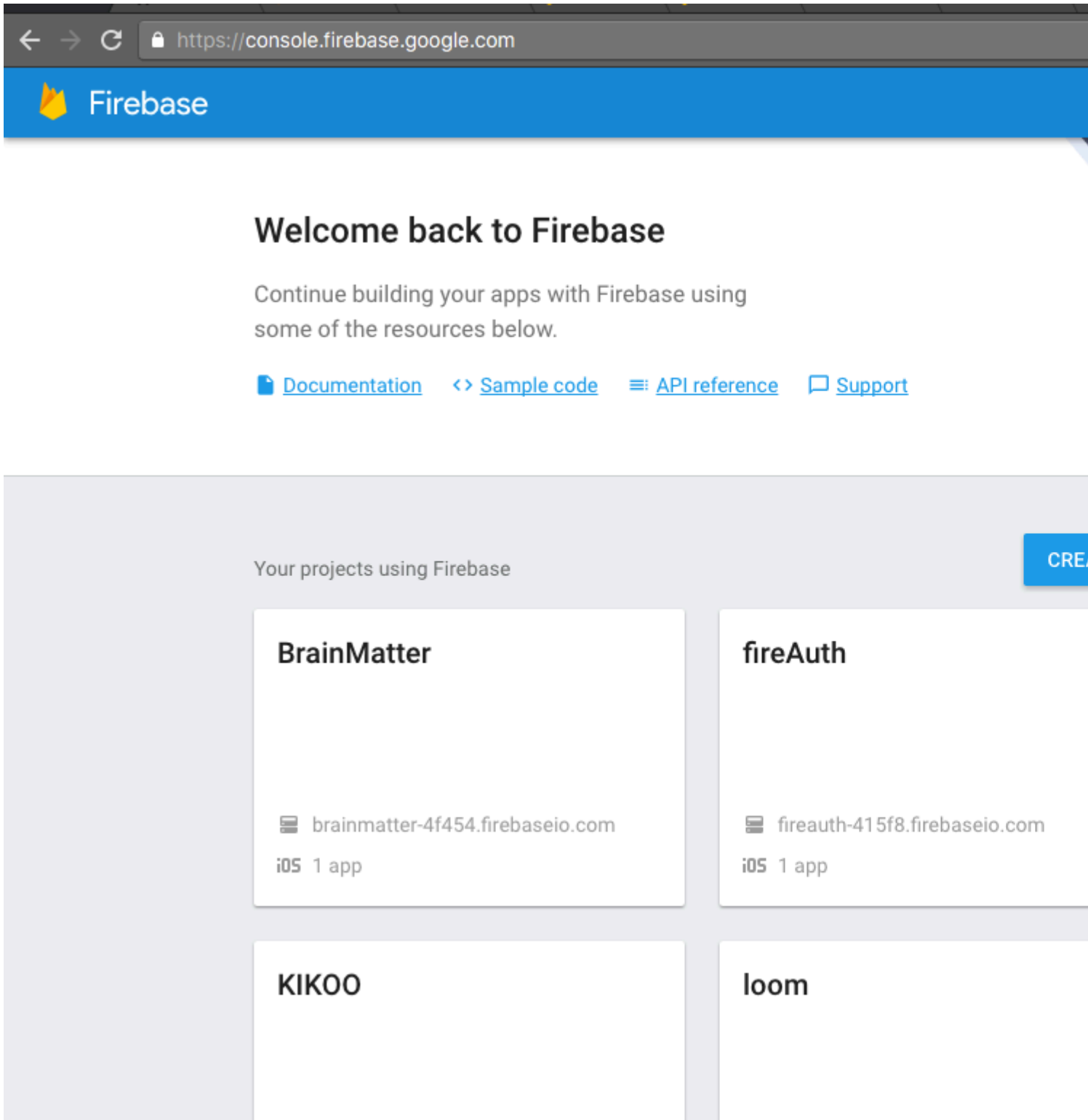
// ADD THIS AT THE BOTTOM
apply plugin: 'com.google.gms.google-services'
```

| Gradle Dependency Line | |
|--|---|
| com.google.firebase : firebase-core : 9.4.0 | |
| com.google.firebase : firebase-database : 9.4.0 | |
| com.google.firebase : firebase-storage : 9.4.0 | |
| com.google.firebase : firebase-crash : 9.4.0 | |
| com.google.firebase : firebase-auth : 9.4.0 | |
| com.google.firebase : firebase-messaging : 9.4.0 | / |
| com.google.firebase : firebase-config : 9.4.0 | |
| com.google.firebase : firebase-invites : 9.4.0 | / |

| Gradle Dependency Line | |
|--|-------|
| com.google.firebase : firebase-ads : 9.4.0 | AdMob |
| com.google.android.gms : play-services-appindexing : 9.4.0 | |

IOS Firebase

1. , firebase 'Create New Project' .



2.

Welcome back to Firebase

Continue building your apps with Firebase using some of the resources below.

[Documentation](#) <> [Sample code](#)

Your projects using Firebase

BrainMatter

brainmatter-4f454.firebaseio.com

iOS 1 app

KIKOO

loom

Create a project

Project name

Country/region ?

United States ▼

By default, your Firebase Analytics data will enhance other features and Google products. You can control how your data is shared in your settings at anytime. [Learn more](#)

CANCEL

CREATE

3. . firebase . IOS .



📊 Analytics

DEVELOP

👤 Auth

🗄 Database

📁 Storage

🌐 Hosting

🔗 Remote Config

📄 Test Lab

🐛 Crash

GROW

💬 Notifications

🔗 Dynamic Links

EARN

🎮 AdMob

Spark
Free

UPGRADE



Welcome to Firebase



Add Firebase to
your iOS app



Add
your A

Discover Firebase



Analytics

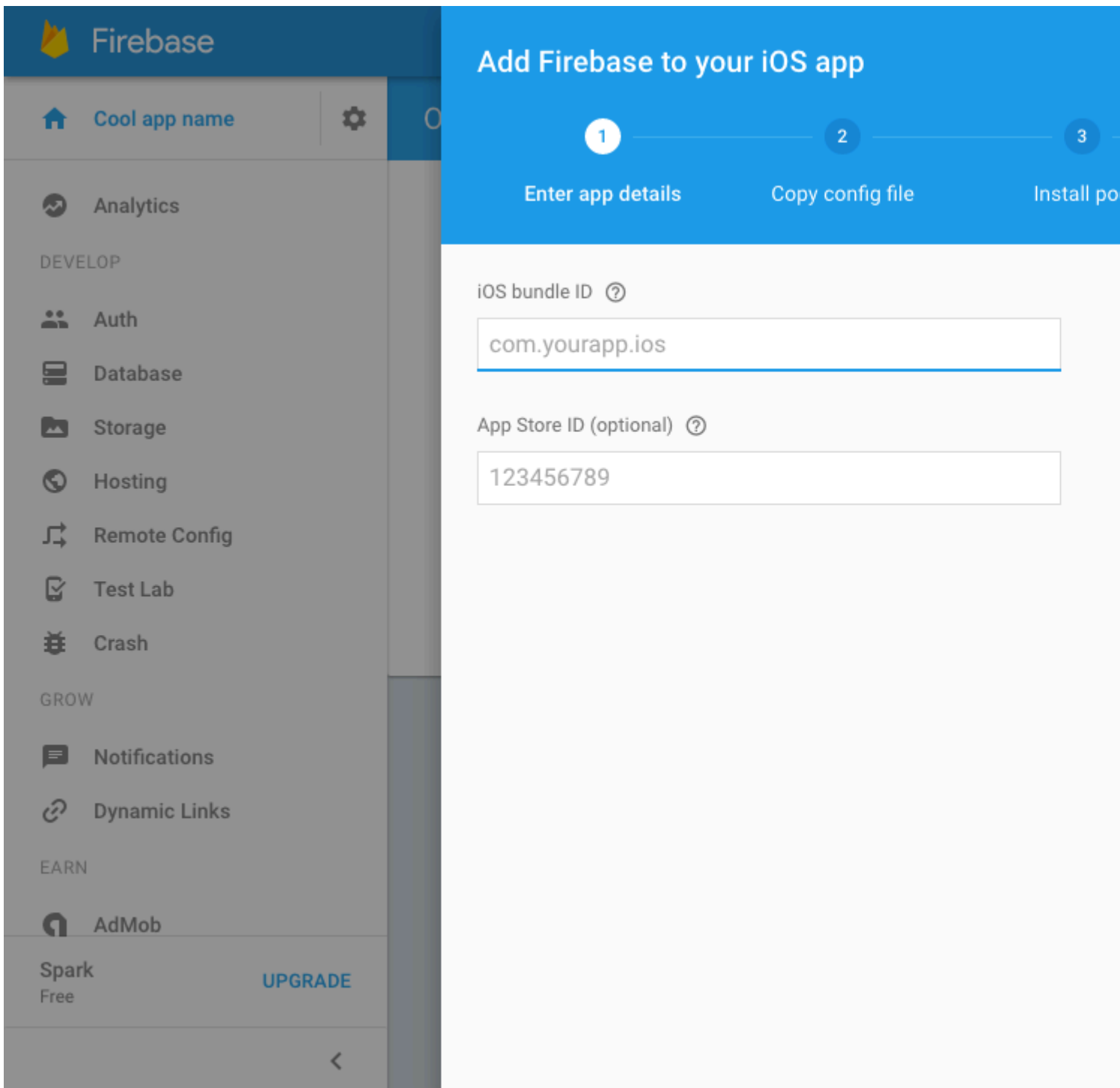
Get detailed analytics to measure and



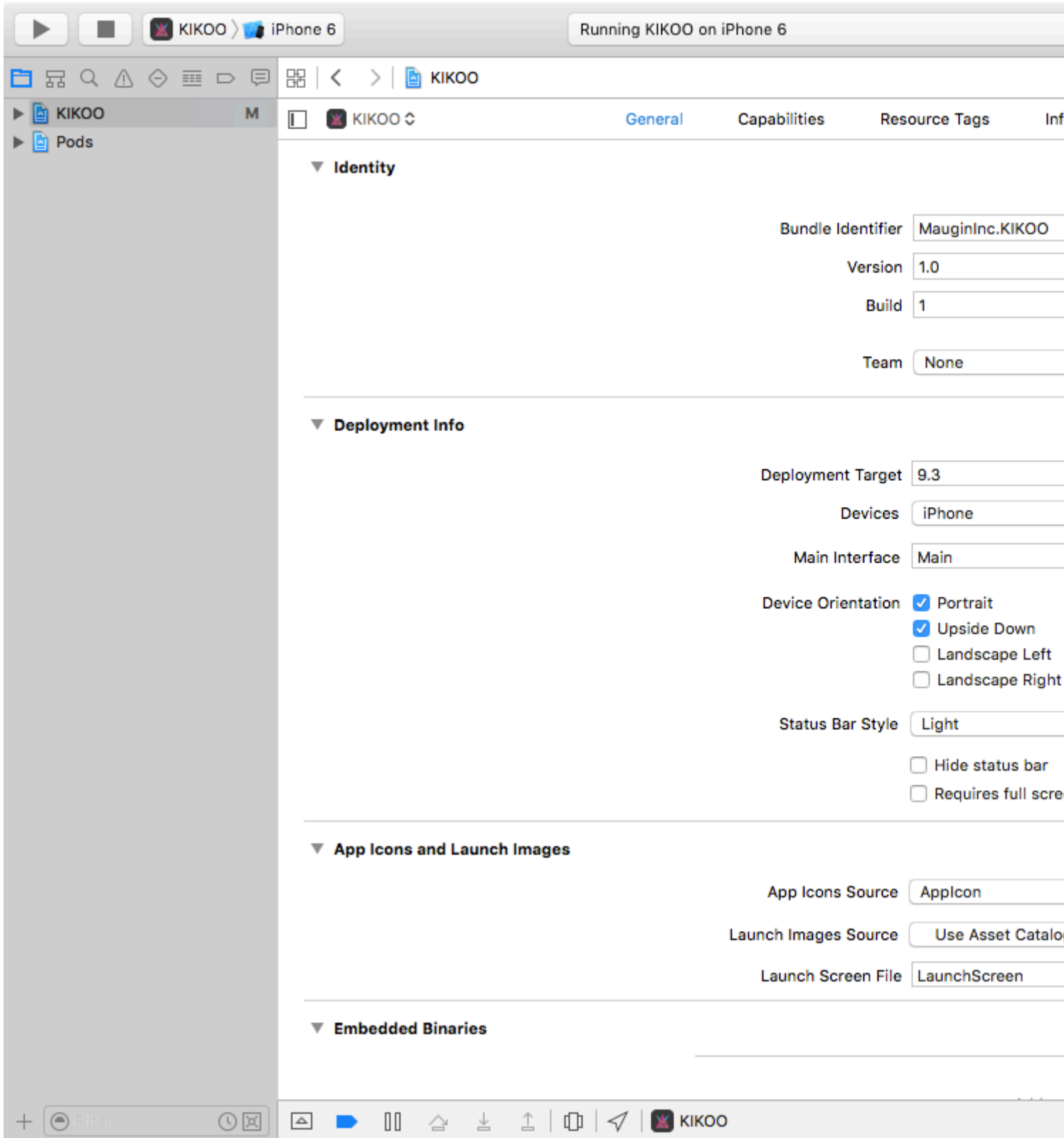
Auth

Authenticate and m

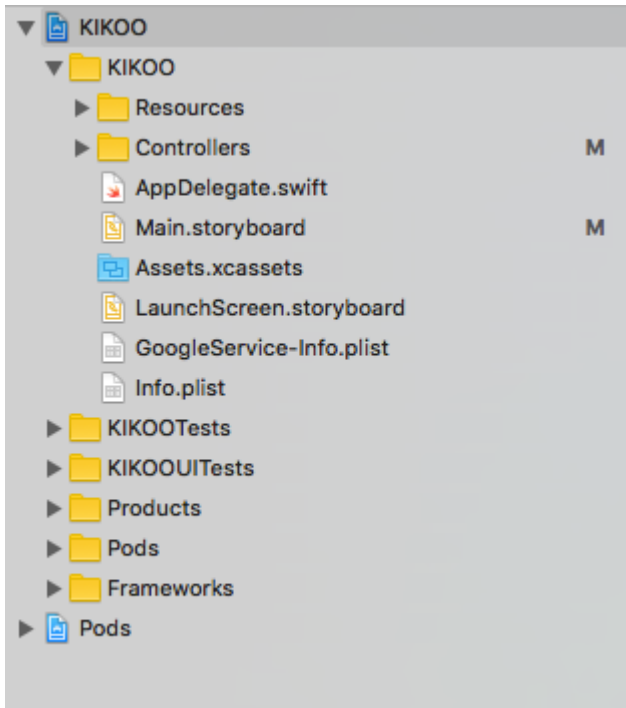
4. IOS IOS Bundle ID . IOS .



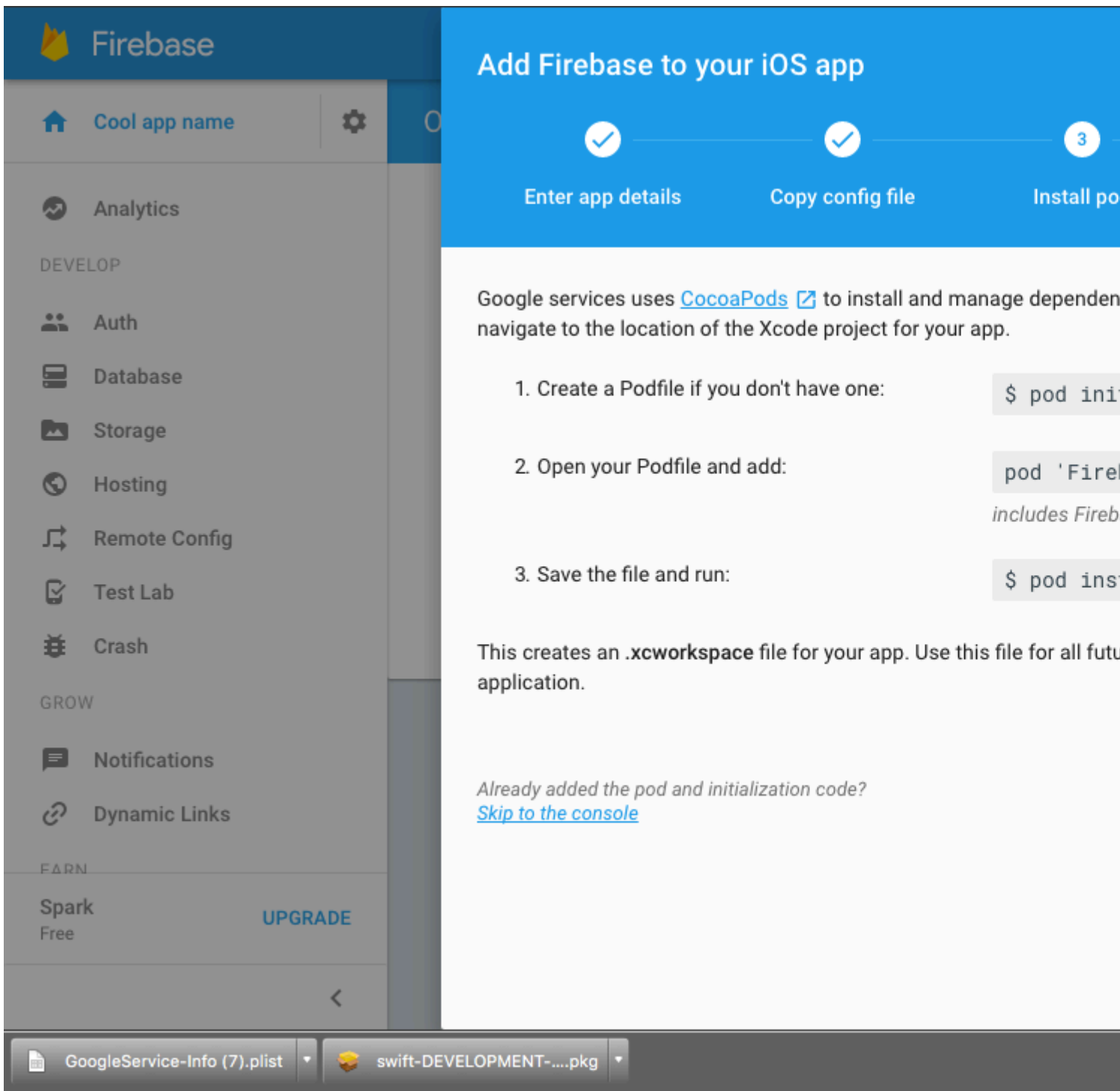
5. xcode xcode ID . ID . xcode Genral . firebase Bundle 'MauginInc.KIKOO' .



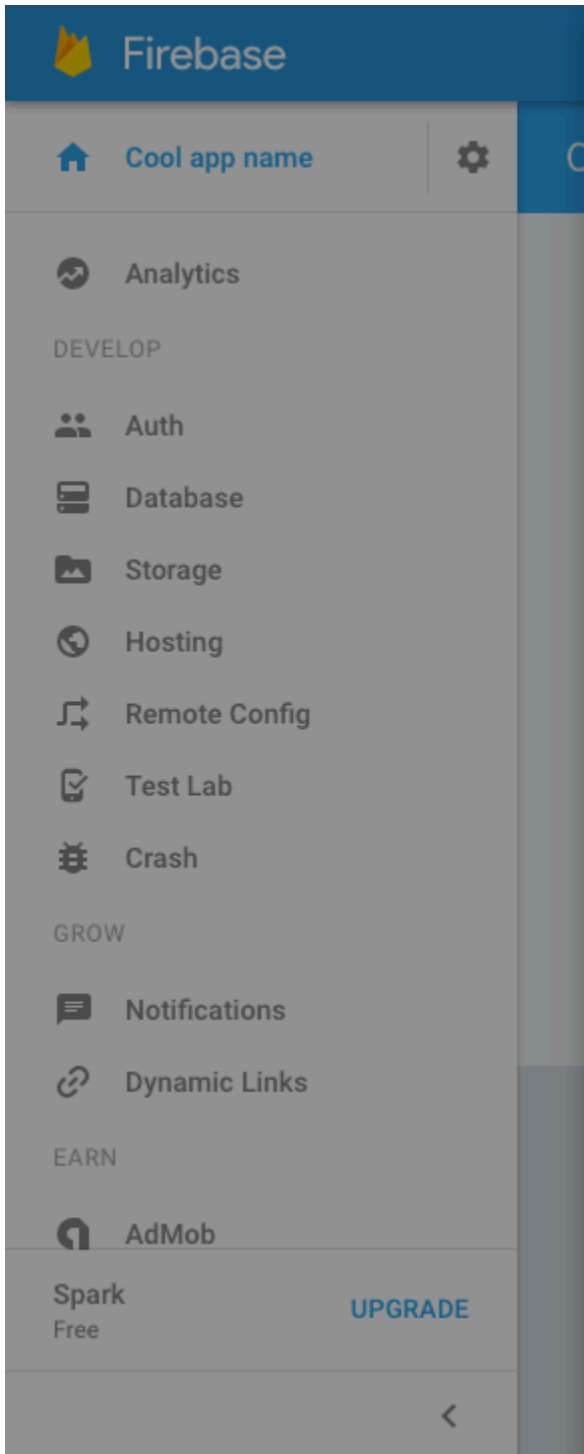
6. 'Next' 'GoogleService-Info.plist' xcode .



7. xcode firebase .



8. , AppDelegate.swift .



IOS xcode firebase .

Hello World Firebase

Firebase .

.

■

- Firebase (<https://console.firebase.google.com>) . , / .

Create a project ×

Project name

Country/region ?

By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. [Learn more](#)

CANCEL CREATE PROJECT

- **index.html** . .

```
<body>
  <p>Getting started with Firebase</p>
  <h1 id="bigOne"></h1>
  <script>
    // your firebase JavaScript code here
  </script>
</body>
```

- **Firestore Console** .

Welcome to Firebase! Get started here.



Add Firebase to
your iOS app



Add Firebase to
your Android app



- **Firestore**


```
</body>
```

- **index.html** (`firebase` ,).

```
<script>

  // Initialize Firebase
  var config = {
    apiKey: "apiKey",
    authDomain: "authDomain",
    databaseURL: "databaseURL",
    storageBucket: "storageBucket",
    messagingSenderId: "messagingSenderId"
  };
  firebase.initializeApp(config);

  // getting the text value from the database
  var bigOne = document.getElementById('bigOne');
  var dbRef = firebase.database().ref().child('text');
  dbRef.on('value', snap => bigOne.innerText = snap.val());

</script>
```

- **index.html** . `firebase` .

• . . .

Key: null

Name: text Value: Hello Firebase

CANCEL ADD

- **ADD** .

Realtime Database

DATA

RULES

USAGE

BACKUPS

- **RULES** .

• .

```
{
  "rules": {
    ".read": "true",
    ".write": "true"
  }
}
```

```
1  {
2    "rules": {
3      ".read": "true",
4      ".write": "true"
5    }
6  }
```

- index.html .

- .

Getting started with Firebase

Hello Firebase

- . Firebase .

firebase : <https://riptutorial.com/ko/firebase/topic/816/firebase->

2: Firebase Firebase

Examples

Firestore uid .

-

```
{
  "users": {
    "uid1": {
      "name": "Steve",
      "surname": "Jobs"
    },
    "uid2": {
      "name": "Bill",
      "surname": "Gates"
    }
  }
}
```

-

```
{
  "rules": {
    "users": {
      "$uid": {
        // If node's key matches the id of the auth user
        ".write": "$uid == auth.uid"
      }
    }
  }
}
```

Firestore "\$uid", users . \$.

?

Firestore + . Firestore ?

? Firestore API .

. [Firestore Realtime Database](#) .

Firestore uid, displayName, photoURL, email . .updateProfile . , rDB, users . . .
.onAuthStateChange .onAuthStateChange .

.onAuthStateChange

- getItem('displayName') getItem('photoURL')
- user.displayName user.photoURL user.photoURL
- ◦ setItem('displayName')

```
setItem('photoURL')
◦ db.ref.child('users').update displayName / photoURL db.ref.child('users').update
  db.ref.child('users').update
.onAuthStateChange .onAuthStateChange . (: ). .
```

Firestore Firestore : <https://riptutorial.com/ko/firebase/topic/1729/firebase---firebase----->

3: Firebase ?

Firebase : . Firebase Database . . .

Examples

Android

Dogs Dog .

```
DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child("dogs");
```

```
String key = reference.push().getKey();  
Dog dog = new Dog("Spike");  
dog.setKey(key);  
reference.child(key).setValue(dog);
```

```
reference.child(key).setValue(dog); reference.push().setValue(dog);  
reference.push().setValue(dog); Dog .
```

Firebase ? : <https://riptutorial.com/ko/firebase/topic/10839/firebase----->

4: Firebase

Firestore Android .

? keepSynced ?

Firestore DiskPersistence .

Firestore DiskPersistence ., .

Firestore DiskPersistence keepSynced .

?

Firestore ListenerValueEvents . :

1. A "Hello World" . B .
2. B A "Hello World" .
3. "Firestore is cool" .
4. B Firestore "Hello World" .

?

Firestore diskPersistence keepSync .

Examples

(Android / iOS)

Firestore persistenceEnabled .

```
Firestore.getInstance().setPersistenceEnabled(true);
```

iOS

```
Database.database().isPersistenceEnabled = true //Swift  
[FIRDatabase database].persistenceEnabled = YES; //Objective-C
```

```
Firestore.getInstance().setPersistenceEnabled(false);
```

iOS

```
Database.database().isPersistenceEnabled = false //Swift  
[FIRDatabase database].persistenceEnabled = NO; //Objective-C
```

(Android / iOS)

Firestore . . .

Android :

```
DatabaseReference workoutsRef = FirebaseDatabase.getInstance().getReference("workouts");  
scoresRef.keepSynced(true);
```

:

```
//Objective-c  
FIRDatabaseReference *scoresRef = [[FIRDatabase database] referenceWithPath:@"scores"];  
[scoresRef keepSynced:YES];  
//Swift  
let scoresRef = Database.database().reference(withPath: "scores")  
scoresRef.keepSynced(true)
```

Firestore . . .

Android :

```
scoresRef.keepSynced(false);
```

iOS :

```
[scoresRef keepSynced:NO]; //Objective-C  
scoresRef.keepSynced(false) //Swift
```

Firestore : <https://riptutorial.com/ko/firebase/topic/10777/firebase-->

5: Firebase

Firestore [Firestore](#) [Google Cloud Platform](#) [Cloud](#) [Firebase](#) [Cloud Functions](#) .

[JavaScript](#) [Node.js](#) . [Firebase SDK for Cloud](#) [Firebase](#) [Firebase](#) .

Examples

GitHub . <https://github.com/firebase/functions-samples/blob/master/quickstarts/email-users>

- .

Firestore .

- [Firestore](#) [Firestore](#) [Project](#) .
- **Auth** [Google](#) .
- **>> TODO** / [index.html](#) .

```
* TODO(DEVELOPER): Paste the initialization snippet from: Firebase Console > Overview > Add
Firestore to your web app. *
```

```
*****
-->
<script src="https://www.gstatic.com/firebasejs/3.7.3/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "your apiKey",
    authDomain: "authDomain.firebaseio.com",
    databaseURL: "https://databaseURL.firebaseio.com",
    storageBucket: "storageBucket.appspot.com",
    messagingSenderId: "messagingID"
  };
  firebase.initializeApp(config);
</script>
```

Firestore CLI

- **NodeJS** <https://nodejs.org/ko/> ([NodeJS](#)).
- / **npm install -g firebase-tools** **firebase login** .
- **==> firebase** CLI **--add** .
- **. cd functions; npm ; cd -**

Google Cloud

- `gmail.email` `gmail.password` Google Cloud Gmail . Firebase CLI :

```
firebase : config : set gmail.email = "myusername@gmail.com" gmail.password = "secretpassword"
```

- `cmd / firebase deploy` .

```
=== Deploying to '[REDACTED]'...

i  deploying database, functions, hosting
+  database: rules ready to deploy.
i  functions: ensuring necessary APIs are enabled...
i  runtimeconfig: ensuring necessary APIs are enabled...
+  functions: all necessary APIs are enabled
+  runtimeconfig: all necessary APIs are enabled
i  functions: preparing functions directory for uploading...
i  functions: packaged functions (1.85 KB) for uploading
+  functions: functions folder uploaded successfully
i  hosting: preparing public directory for upload...
Uploading: [=====] 46%+  hosting: public folder
+  hosting: 82 files uploaded successfully
i  starting release process (may take several minutes)...
i  functions: updating function sendEmailConfirmation...
+  functions[sendEmailConfirmation]: Successful update operation.
+  functions: all functions deployed successfully!

+  Deploy complete!

Project Console: https://console.firebase.google.com/project/[REDACTED]/overview
Hosting URL: https://[REDACTED].firebaseapp.com ← Use this URL to open
```

- `firebase open hosting : site` URL .

Firebase : <https://riptutorial.com/ko/firebase/topic/9580/firebase--->

6: Firebase

1. Firebase .
2. Firebase .

| Firestore | Analytics | Auth |
|-----------|-----------|------|
| Firestore | ? | ? |

firebase . firebase .

Examples

Firestore All In One

[Firestore Console](#)

[Android : Firestore](#)

Android :

-
- FirebaseAuthActivity
-

Firestore : <https://riptutorial.com/ko/firebase/topic/6660/firebase->

7: FirebaseUI

Firebase , , . , , , .

Firebase . , Firebase . , .:)

FirebaseUI Firebase Firebase Google .

Lego , FirebaseUI . FirebaseUI Firebase FirebaseUI . FirebaseUI . . .
FirebaseUI .

?

FirebaseUI . . () - . FirebaseUI .

Examples

FirebaseUI

FirebaseUI [Android](#) , [iOS](#) . . .

:

```
// app/build.gradle

dependencies {
    // Single target that includes all FirebaseUI libraries
    compile 'com.firebaseui:firebase-ui:0.5.2'

    // FirebaseUI Database only
    compile 'com.firebaseui:firebase-ui-database:0.5.2'

    // FirebaseUI Auth only
    compile 'com.firebaseui:firebase-ui-auth:0.5.2'
}
```

iOS :

```
# Podfile

# Pull in all Firebase UI features
pod 'FirebaseUI', '~> 0.5'

# Only pull in the "Database" FirebaseUI features
pod 'FirebaseUI/Database', '~> 0.5'

# Only pull in the "Auth" FirebaseUI features (including Facebook and Google)
pod 'FirebaseUI/Auth', '~> 0.5'

# Only pull in the "Facebook" login features
pod 'FirebaseUI/Facebook', '~> 0.5'
```

```
# Only pull in the "Google" login features
pod 'FirebaseUI/Google', '~> 0.5'
```

:

```
<!--Include FirebaseUI sources in HTML-->

<script src="https://www.gstatic.com/firebasejs/ui/live/0.5/firebase-ui-auth.js"></script>
<link type="text/css" rel="stylesheet"
href="https://www.gstatic.com/firebasejs/ui/live/0.5/firebase-ui-auth.css" />
```

FirebaseUI : <https://riptutorial.com/ko/firebase/topic/6418/firebaseui>

8: FirebaseAuth ()

Examples

[FirebaseUI](#) [Firebase Auth](#) [Firebase Database](#) [UI](#) [Google](#) .

FirebaseUI `build.gradle` .

```
android {
    // ...
}

dependencies {
    // Required for FirebaseAuth Database
    compile 'com.google.firebase:firebase-database:9.4.0'
    compile 'com.firebaseui:firebase-ui-database:0.5.1'

    // FirebaseAuth only
    compile 'com.google.firebase:firebase-auth:9.4.0'
    compile 'com.firebaseui:firebase-ui-auth:0.5.1'

    // Single dependency if you're using both
    compile 'com.firebaseui:firebase-ui:0.5.1'
}

apply plugin: 'com.google.gms.google-services'
```

ListView

Android Studio , `ListView` ().

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- Your toolbar, etc -->

    <ListView
        android:id="@+id/list_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.design.widget.CoordinatorLayout>
```

`ListView` .

```
public class Person {

    private String name
```



```

public Person() {
    // Constructor required for Firebase Database
}

public String getName() {
    return name;
}
}

```

ListView **id** Activity FirebaseListAdapter .

```

public class MainActivity extends AppCompatActivity {

    // ...

    private ListView mListView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Find the ListView
        mListView = (ListView) findViewById(R.id.list_view);

        /*
         * Create a DatabaseReference to the data; works with standard DatabaseReference
methods
         * like limitToLast() and etc.
         */
        DatabaseReference peopleReference = FirebaseDatabase.getInstance().getReference()
            .child("people");

        // Now set the adapter with a given layout
        mListView.setAdapter(new FirebaseListAdapter<Person>(this, Person.class,
            android.R.layout.one_line_list_item, peopleReference) {

            // Populate view as needed
            @Override
            protected void populateView(View view, Person person, int position) {
                ((TextView) view.findViewById(android.R.id.text1)).setText(person.getName());
            }
        });
    }
}

```

ListView .

FirestoreUI () : <https://riptutorial.com/ko/firebase/topic/6610/firebaseui--->

9:

- . Firebase .
-
- angle-ui-router URL .
- Firebase applyCode params applyCode .
- .

| | |
|--------------------------|-------------------------------|
| ... | |
| sendEmailVerification () | . |
| applyActionCode () | emailVerified false true true |

Firestore . .

[Firestore 3.0 SDK Email Verification](#) .

Examples

- - AngularJS

```
// thecontroller.js
$scope.sendVerifyEmail = function() {
  console.log('Email sent, whaaaaam!');
  currentAuth.sendEmailVerification();
}

// where currentAuth came from something like this:
// routerconfig

....
templateUrl: 'bla.html',
resolve: {
  currentAuth:['Auth', function(Auth) {
    return Auth.$requireSignIn() // this throws an AUTH_REQUIRED broadcast
  }]
}
...

// intercept the broadcast like so if you want:

....

$rootScope.$on("$stateChangeError", function(event, toState, toParams, fromState, fromParams,
error) {
  if (error === "AUTH_REQUIRED") {
    $state.go('login', { toWhere: toState });
  }
});
....
```

```

// So user receives the email. How do you process the `oobCode` that returns?
// You may do something like this:

// catch the url with its mode and oobCode
.state('emailVerify', {
  url: '/verify-email?mode&oobCode',
  templateUrl: 'auth/verify-email.html',
  controller: 'emailVerifyController',
  resolve: {
    currentAuth:['Auth', function(Auth) {
      return Auth.$requireSignIn()
    }]
  }
})

// Then digest like so where each term is what they sound like:

.controller('emailVerifyController', ['$scope', '$stateParams', 'currentAuth', 'DatabaseRef',
function($scope, $stateParams, currentAuth, DatabaseRef) {
  console.log(currentAuth);
  $scope.doVerify = function() {
    firebase.auth()
      .applyActionCode($stateParams.oobCode)
      .then(function(data) {
        // change emailVerified for logged in User
        toastr.success('Verification happened', 'Success!');
      })
      .catch(function(error) {
        $scope.error = error.message;
        toastr.error(error.message, error.reason, { timeOut: 0 });
      })
  };
}
])

```

[: https://riptutorial.com/ko/firebase/topic/3380/---](https://riptutorial.com/ko/firebase/topic/3380/---)

10: Firebase

Examples

Hello World Firebase

:

- [JS](#)

1. [Firebase Console](#) .
2. .
3. .
4. [JSON](#) (Google Apps) .
5. [JSON](#) .
6. [Firebase](#) [Firebase](#) . [user_data](#) .
7. [Firebase](#) .

```
//Loading Firebase Package
var firebase = require("firebase");

/**
 * Update your Firebase Project
 * Credentials and Firebase Database
 * URL
 */
firebase.initializeApp({
  serviceAccount: "<path to Firebase Credentials Json File>",
  databaseURL: "<Firebase Database URL>"
}); //by adding your credentials, you get authorized to read and write from the database

/**
 * Loading Firebase Database and refering
 * to user_data Object from the Database
 */
var db = firebase.database();
var ref = db.ref("/user_data"); //Set the current directory you are working in

/**
 * Setting Data Object Value
 */
ref.set([
  {
    id:20,
    name:"Jane Doe",
    email:"jane@doe.com",
    website:"https://jane.foo.bar"
  },
  {
    id:21,
    name:"John doe",
    email:"john@doe.com",
    website:"https://foo.bar"
  }
]);
```

```

}
]);

/**
 * Pushing New Value
 * in the Database Object
 */
ref.push({
  id:22,
  name:"Jane Doe",
  email:"jane@doe.com",
  website:"https://jane.foo.bar"
});

/**
 * Reading Value from
 * Firebase Data Object
 */
ref.once("value", function(snapshot) {
  var data = snapshot.val(); //Data is in JSON format.
  console.log(data);
});

```

8. JSON URL (index.js).

9. index.js Firebase URL . Firebase URL . URL *https://firebaseio.com/* .

10.

```
npm install firebase
```

11. NPM Firebase .

```
node index.js
```

?

Firebase . Firebase .

Database . .

Firestore

firebase firebase

firebase

1. Firebase Firestore . Google Google . Create New Project .

2. .

3. .

4. .

.

ID ID .

> .

JSON .

Google Apps .

.

5. JSON . SDK .

nodejs npm firebase-queue .

```
npm install firebase firebase-queue --save
```

firebase firebase-queue , Queue Firebase .

firebase worker firebase-queue .

* server.js

```
var app=express();
var Queue = require('firebase-queue'),
    Firebase = require('firebase');
```

Firebase Firebase URL

```
var firebase = Firebase.initializeApp({
  serviceAccount: "path/to/serviceAccountCredentials.json",
  databaseURL: "https://databaseName.firebaseio.com"
});
```

Firebase

```
var firebase = Firebase.initializeApp({
  serviceAccount: {
    projectId: "projectId",
    clientEmail: "foo@projectId.iam.gserviceaccount.com",
    privateKey: "-----BEGIN PRIVATE KEY-----\nkey\n-----END PRIVATE KEY-----\n"
  },
  databaseURL: "https://databaseName.firebaseio.com"
});

var refQueue = firebase.database().ref("queue/tasks");

createUser = function(email, password){
  var user = {
    username: email,
    password: password
  };
  user = new db.users(user);
  user.save(function(err, user){
```

```

        if(!err){
            refQueue.push({case: "NEW_USER", data: user});
        }
    })
}

createUser("abc@abc.com", "password");

```

* worker.js

```

var Queue = require('firebase-queue'),
    Firebase = require('firebase');

//Update your Firebase Project Credentials and Firebase Database URL by one of the way
specified in server.js
var firebase = Firebase.initializeApp({
    serviceAccount: "path/to/serviceAccountCredentials.json",
    databaseURL: "https://databaseName.firebaseio.com"
});

var refQueue = firebase.database().ref("queue");

var queue = new Queue(refQueue, function(data, progress, resolve, reject) {
    switch(data.case){
        case "NEW_USER":
            sendMail(data.data.email);
            console.log("user created");
            //sendMail function is not an inbuilt function and will not work unless you define
            and implement the function
            break;

            // Finish the task asynchronously
            setTimeout(function() {
                resolve();
            }, 1000);
    });
}

```

firebase

```
node server.js
```

```
node worker.js
```

Firestore : <https://riptutorial.com/ko/firebase/topic/6443/--firebase->

11:

Firebase JSON NoSQL .SQL - .

. .JSON .

Examples

.

```
{
  "users": {
    // Uniquely generated IDs for children is common practice,
    // it's actually really useful for automating child creation.
    // Auto-incrementing an integer for a key can be problematic when a child is removed.

    "-KH3Cx0KFvSQELIYZezv": {
      "name": "Jon Snow",
      "aboutMe": "I know nothing...",
      "posts": {
        "post1": {
          "body": "Different roads sometimes leads to the same castle",
          "isHidden": false
        },
        "post2": { ... },
        // Possibly more posts
      }
    },
    "-KH3Dx2KFdSLerIYZcgk": { ... }, // Another user
    // A lot more users here
  }
}
```

.

Firebase () . () . . 4 .

.

```
{
  // "users" should not contain any of the posts' data
  "users": {
    "-KH3Cx0KFvSQELIYZezv": {
      "name": "Jon Snow",
      "aboutMe": "I know nothing..."
    },
    "-KH3Dx2KFdSLerIYZcgk": { ... },
    // More users
  },

  // Posts can be accessed provided a user key
  "posts": {
    "-KH3Cx0KFvSQELIYZezv": { // Jon Snow's posts
```



```

    "post1": {
      "body": "Different roads sometimes leads to the same castle",
      "isHidden": false
    },
    "post2": { ... },
    // Possibly more posts
  },
  "-KH3Dx2KFdSLERiYZcgk": { ... },
  // other users' posts
}
}

```

. *posts* .

```

{
  "students": {
    "-SL3Cs0KFvDMQLIYZEzv": {
      "name": "Godric Gryffindor",
      "id": "900130309",
      "courses": {
        "potions": true,
        "charms": true,
        "transfiguration": true,
      }
    },
    "-SL3ws2KvZQLTYMqzSas": {
      "name": "Salazar Slytherin",
      "id": "900132319",
      "courses": {
        "potions": true,
        "herbs": true,
        "muggleStudies": true,
      }
    },
    "-SL3ns2OtARSTUMyqwWt": { ... },
    // More students here
  },

  "courses": {
    "potions": {
      "code": "CHEM305",
      "enrolledStudents": {
        "-SL3Cs0KFvDMQLIYZEzv": true,      // Godric Gryffindor
        "-SL3ws2KvZQLTYMqzSas": true,     // Salazar Slytherin
        // More students
      }
    },
    "muggleStuddies": {
      "code": "SOC215",
      "enrolledStudents": {
        "-SL3ws2KvZQLTYMqzSas": true,     // Salazar Slytherin
        "-SL3ns2OtARSTUMyqwWt": true,     // Some other student
        // More students
      }
    },
    // More courses
  }
}

```

.
/ .

SQL .

- 1. ...
- 2. .

. .

: [https://riptutorial.com/ko/firebase/topic/8912/-](https://riptutorial.com/ko/firebase/topic/8912/)

12:

Firestore Database

Firestore Database

Firestore Database , . Google .

Firestore

<https://firebase.google.com/docs/database/security/quickstart>

Examples

1. Firestore .
- 2.
3. .

.

.

.

```
// These rules require authentication
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

:

```
// These rules give anyone, even people who are not users of your app,
// read and write access to your database
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

.

.

(Firestore).

```
// These rules don't allow anyone read or write access to your database
{
```

```

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

```

`/users/$user_id` . `$ user_id` ID.

```

// These rules grant access to a node matching the authenticated
// user's ID from the Firebase auth token
{
  "rules": {
    "users": {
      "$user_id": {
        ".read": "$user_id === auth.uid",
        ".write": "$user_id === auth.uid"
      }
    }
  }
}

```

ID . , `/administrators/$user_id` true "administrators" . . .

```

{
  "rules": {
    "administrators": {
      // No one can list administrators
      ".read": "false",
      "$uid": {
        // Authenticated user can check if they are in this group
        ".read": "$uid === auth.uid",
        // Administrators can write
        ".write": "data.parent().child(auth.uid).val() === true",
        // Allow only add or delete, no duplicates
        ".validate": "!data.exists() || !newData.exists() || newData.isBoolean()",
      }
    }
  }
}

```

: <https://riptutorial.com/ko/firebase/topic/3352/>-

13: ?

. () / .

Examples

Android .

. () .

logcat . . . (JSON,).

Android `setValue` .

```
ref.setValue("My new value", new DatabaseReference.CompletionListener() {  
    public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference) {  
        throw databaseError.toException();  
    }  
});
```

throw .

Android .

().

logcat . . . (JSON,).

Android `ChildEventListener onCancelled` .

```
databaseRef.addChildEventListener(new ChildEventListener() {  
    public void onChildAdded(DataSnapshot dataSnapshot, String s) { ... }  
    public void onChildChanged(DataSnapshot dataSnapshot, String s) { ... }  
    public void onChildRemoved(DataSnapshot dataSnapshot) { ... }  
    public void onChildMoved(DataSnapshot dataSnapshot, String s) { ... }  
    public void onCancelled(DatabaseError databaseError) {  
        throw databaseError.toException();  
    }  
});
```

ValueEventListener :

```
databaseRef.addValueEventListener(new ValueEventListener() {  
    public void onDataChange(DataSnapshot dataSnapshot, String s) { ... }  
    public void onCancelled(DatabaseError databaseError) {  
        throw databaseError.toException();  
    }  
});
```

Android .

iOS .

. () .

. . . (JSON,).

iOS setValue .

```
let message = [{"name": "puf", "text": "Hello from iOS"}]
ref!.childByAutoId().setValue(message) { (error) in
    print("Error while writing message \(error)")
}
```

throw .

JavaScript

().

JavaScript . . . (JSON,).

JavaScript on() on() .

```
ref.on('value', function(snapshot) {
    console.log(snapshot.key, snapshot.val());
}, function(error) {
    alert(error);
})
```

JavaScript .

JavaScript .

. () .

. . . (JSON,).

set .

```
ref.set("My new value").catch(function(error)
    console.error(error);
    alert(error);
});
```

iOS .

().

. . . (JSON,).

iOS withCancel .

```
ref!.child("notAllowed").observe(.value, with: { (snapshot) in
    print("Got non-existing value: \(snapshot.key)")
}, withCancel: { (error) in
    print(error)
})
```

? : <https://riptutorial.com/ko/firebase/topic/5548/----->

14:

Firestore admin SDK , HTTP POST

Examples

Firestore HTTP

.

```
https://fcm.googleapis.com/fcm/send
```

```
Authorization    key=<Your_key_from_the_console>
Content-Type     application/json
```

.

```
{
  "to" : <tokens or the topic>,
  "notification" : {
    "title":"This is a test title",
    "body":"This is the body"
  },
  "data": {
    //whatever key value payer you need to send
  }
}
```

to .

```
["token1","token2",.....]
```

```
"token"
```

/ topic / like

```
"/topic_name/"
```

|| &&

```
"/topic_name/ && /topic2/"
```

Admin SDK (js)

firebase sdk admin SDK .


```

const functions = require('firebase-functions');
const admin = require('firebase-admin');

admin.initializeApp({
  credential: admin.credential.cert({
    //your admin credential certificate generated from the console. Follow this [link][1].
  }),
  databaseURL: "https://<PROJECT_NAME>.firebaseio.com"
});

```

JSON .

```

var payload = {
  notification: {
    title: "Title of the notification",
    body: "Body of the notification",
  },
  data: {
    //required key value pair
  }
};

```

send .

```

admin.messaging().sendToTopic("/topic/", payload)
  .then(function(response) {
    console.log("Successfully sent message:", response);
  })
  .catch(function(error) {
    console.log("Error sending message:", error);
  });
});

```

```

admin.messaging().sendToDevice(tokens, payload).then(response=>{
  response.results.forEach((result, index) => {
    const error = result.error;
    if (error) {
      console.error('Failure sending notification to', tokens, error);
    } else{
      console.log('Sucessfully sent to '+tokens);
    }
  });
});

```

: <https://riptutorial.com/ko/firebase/topic/10548/---->

15: Firebase

Examples

FirebaseRealTime . .

1. firebase SDK . URL . [Firebase SDK](#)
2. thet firbase firbase URL [Firebase Console Url.](#)
3. successfully gradel . 'com.google.firebase : firebase-database : 9.2.1'
4. firebase . URL . [firebase](#)
5. . FirebaseDatabase .

```
Firebase = FirebaseDatabase.getInstance (); DatabaseReference myRef =  
database.getReference ( "message");
```

DatabaseReference .

6. DataBaseReference . :

```
myRef.setValue ( " ");
```

:

```
myRef.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        // This method is called once with the initial value and again  
        // whenever data at this location is updated.  
        String value = dataSnapshot.getValue(String.class);  
        Log.d(TAG, "Value is: " + value);  
    }  
  
    @Override  
    public void onCancelled(DatabaseError error) {  
        // Failed to read value  
        Log.w(TAG, "Failed to read value.", error.toException());  
    }  
});
```

: FirebaseRealtime ,

[Firebase](https://riptutorial.com/ko/firebase/topic/6482/-firebase--) : <https://riptutorial.com/ko/firebase/topic/6482/-firebase-->

16: RecyclerViewAdapter FirebaseRecyclerViewAdapter ?

Examples

Firestore FirebaseRecyclerViewAdapter.

firebase UI , gradle .

```
compile 'com.firebaseui:firebase-ui-database:0.4.0'
```

firebase

```
DatabaseReference databaseReference = database.getReference().child("users");  
Query query = databaseReference.limitToFirst(50);
```

FirestoreRecyclerViewAdapter

```
private void setUpFirestoreAdapter(Query query) {  
  
    mFirestoreAdapter = new FirestoreRecyclerViewAdapter<UserModel, FirestoreViewHolder>  
        (UserModel.class, R.layout.row_user_list, FirestoreViewHolder.class, query)  
    {  
        @Override  
        protected void populateViewHolder(FirestoreViewHolder viewHolder, UserModel  
model, int position) {  
            customLoaderDialog.hide();  
            viewHolder.bindUser(model);  
        }  
    };  
  
    my_recycler_view.setHasFixedSize(true);  
    my_recycler_view.setLayoutManager(new LinearLayoutManager(this));  
    my_recycler_view.setAdapter(mFirestoreAdapter);  
  
}
```

ChatUserModel.java ()

```
public class ChatUserModel {  
    private long badge;  
    private String chat_id;  
    private String isDelete;  
    private String latestactivity;  
    private double timestamp;  
    private String user_id;  
    private String profilePic;  
    private String displayName;  
    private boolean isGroup;  
    String groupId;  
    private String creatorId;
```

```

public String getGroupId() {
    return groupId;
}

public void setGroupId(String groupId) {
    this.groupId = groupId;
}

public String getCreatorId() {
    return creatorId;
}

public void setCreatorId(String creatorId) {
    this.creatorId = creatorId;
}

public boolean isGroup() {
    return isGroup;
}

public void setGroup(boolean group) {
    isGroup = group;
}

public ChatUserModel() {
}

public long getBadge() {
    return badge;
}

public void setBadge(long badge) {
    this.badge = badge;
}

public String getChat_id() {
    return chat_id;
}

public void setChat_id(String chat_id) {
    this.chat_id = chat_id;
}

public String getIsDelete() {
    return isDelete;
}

public void setIsDelete(String isDelete) {
    this.isDelete = isDelete;
}

public String getLatestactivity() {
    return latestactivity;
}

public void setLatestactivity(String latestactivity) {
    this.latestactivity = latestactivity;
}

```

```

public double getTimestamp() {
    return timestamp;
}

public void setTimestamp(double timestamp) {
    this.timestamp = timestamp;
}

public String getUser_id() {
    return user_id;
}

public void setUser_id(String user_id) {
    this.user_id = user_id;
}

public String getProfilePic() {
    return profilePic;
}

public void setProfilePic(String profilePic) {
    this.profilePic = profilePic;
}

public String getDisplayName() {
    return displayName;
}

public void setDisplayName(String displayName) {
    this.displayName = displayName;
}
}
}

```

FirestoreChatUserViewHolder.java (RecyclerView.ViewHolder)

```

public class FirestoreChatUserViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {

    private static final int MAX_WIDTH = 200;
    private static final int MAX_HEIGHT = 200;
    View mView;
    Context mContext;
    ChatUserModel userModel;

    public FirestoreChatUserViewHolder(View itemView) {
        super(itemView);
        mView = itemView;
        mContext = itemView.getContext();
        itemView.setOnClickListener(this);
    }

    public void bindUser(ChatUserModel userModel) {
        this.userModel = userModel;
        ImageView imgUser = (ImageView) mView.findViewById(R.id.imgUser);
        TextView tvName = (TextView) mView.findViewById(R.id.tvName);
        TextView tvStatus = (TextView) mView.findViewById(R.id.tvStatus);
        BadgeView badgeChat = (BadgeView) mView.findViewById(R.id.badgeChat);
        if (userModel.isGroup()) {
            //
            imgUser.setImageDrawable(mContext.getResources().getDrawable(R.drawable.create_group));
        }
    }
}

```

```

    } else {
        Picasso.with(mContext)
            .load(userModel.getProfilePic())
            .resize(MAX_WIDTH, MAX_HEIGHT)
            .centerCrop()
            .into(imgUser);
    }

    tvName.setText(userModel.getDisplayName());
    tvStatus.setText(userModel.getLatestactivity());
    if (userModel.getBadge() > 0) {
        badgeChat.setVisibility(View.VISIBLE);
        badgeChat.setText("" + userModel.getBadge());
    } else {
        badgeChat.setVisibility(View.GONE);
    }
}

@Override
public void onClick(View view) {
    if (!userModel.isGroup()) {
        Intent intent = new Intent(mContext, ChatConverstion.class);
        intent.putExtra("chat_id", "" + userModel.getChat_id());
        intent.putExtra("reciverUserName", "" + userModel.getDisplayName());
        intent.putExtra("reciverProfilePic", "" + userModel.getProfilePic());
        intent.putExtra("reciverUid", "" + userModel.getUser_id());
        mContext.startActivity(intent);
    }
}
}
}

```

row_user_list.xml ()

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:color/white"
    android:orientation="horizontal"
    >

    <LinearLayout
        android:gravity="center_vertical"
        android:layout_width="match_parent"
        android:id="@+id/llMainChat"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:paddingTop="@dimen/margin_small"
        android:paddingLeft="@dimen/margin_small"
        android:paddingBottom="@dimen/margin_small"
        android:paddingRight="@dimen/margin_small"
        >

        <com.tristate.firebasechat.custome_view.CircleImageView
            android:id="@+id/imgUser"
            android:layout_width="@dimen/tab_top_height"
            android:layout_height="@dimen/tab_top_height"

            app:civ_border_color="@color/dark_white"

```

```

        app:civ_border_width="2dp" />

<RelativeLayout
    android:layout_width="match_parent "
    android:layout_height="wrap_content "
    android:orientation="horizontal "
    android:layout_marginLeft="@dimen/margin_medium"
    >

    <LinearLayout
        android:layout_width="match_parent "
        android:layout_height="wrap_content "
        android:orientation="vertical "
        android:layout_toLeftOf="@+id/badgeChat "
        android:layout_toStartOf="@+id/badgeChat "
        android:id="@+id/linearLayout">

        <TextView
            android:id="@+id/tvName "
            android:layout_width="wrap_content "
            android:layout_height="wrap_content "
            android:ellipsize="marquee"
            android:singleLine="true"
            android:text="Dhaval Solanki"
            android:textSize="@dimen/textsize_midle" />

        <TextView
            android:id="@+id/tvStatus "
            android:ems="3"
            android:layout_width="wrap_content "
            android:layout_height="wrap_content "
            android:gravity="center_vertical "
            android:lines="1"
            android:text="Online"
            android:textColor="@color/greenStatusBar"
            android:textSize="@dimen/textsize_small" />
    </LinearLayout>
    <com.tristate.firebasechat.custome_view.BadgeView
        android:id="@+id/badgeChat "
        android:layout_width="@dimen/margin_very_big"
        android:layout_height="@dimen/margin_very_big"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:background="@drawable/badge_bg"
        android:gravity="center"
        android:padding="@dimen/corner_radius "
        android:text="999"
        android:textColor="@color/white"
        android:textSize="@dimen/textsize_verysmall"
        android:visibility="gone" />

</RelativeLayout>

</LinearLayout>
<View
    android:layout_alignBottom="@id/llMainChat "
    android:layout_marginTop="@dimen/margin_small "
    android:layout_marginLeft="@dimen/margin_small "
    android:layout_marginRight="@dimen/margin_small "

```

```
        android:layout_width="match_parent"  
        android:background="@color/avatar_back_color"  
        android:layout_height="1dp"  
    ></View>  
</RelativeLayout>
```

RecyclerView Adapter FirebaseRecyclerViewAdapter ? : <https://riptutorial.com/ko/firebase/topic/8982/-recycleradapter--firebaserecycleradapter-->

17:

Android iOS .

<https://firebase.google.com/docs/crash/>

Examples

Android

1. [Firebase](#) .
[Firebase](#) .
2. [Firebase CrashReporting](#) `build.gradle` `build.gradle` .

```
compile 'com.google.firebase:firebase-crash:9.4.0'
```

Android

[Firebase Crash Reporting](#) () .

```
FirebaseCrash.report(new Exception("My first Android non-fatal error"));
```

`FirebaseCrash` :

```
07-20 08 : 57 : 24.442 D / FirebaseCrashApiImpl : FirebaseCrash API 07-20 08 : 57 :
24.442 I / FirebaseCrash : FirebaseCrash d
com.google.firebase.crash.internal.zzg@3333d325 07-20 08 : 57 : 24.442 D /
FirebaseApp : com.google.firebase.crash.FirebaseCrash.
```

:

```
07-20 08 : 57 : 47.052 D / FirebaseCrashApiImpl : throwable java.lang.Exception :
Android 07-20 08 : 58 : 18.822 D / FirebaseCrashSenderServiceImpl : : 200 07-20
08 : 58 : 18.822 D / FirebaseCrashSenderServiceImpl :
```

```
FirebaseCrash.log("Activity created");
```

<https://riptutorial.com/ko/firebase/topic/4669/>

18:

Firestore . Firebase Storage . Google Cloud Storage .

Google App Engine . Google Cloud API . Google Cloud Platform . Firebase Storage . Google Cloud Storage .

gcloud . Firebase Storage . Google Cloud Storage XML JSON API .

Firebase Authentication .

API, [Firebase Storage](#) .

Examples

iOS

1. [Firebase](#) iOS .
2. `GoogleServices-Info.plist` .

Firestore

Podfile .

```
pod 'Firebase/Storage'
```

```
pod install .xcworkspace .
```

CocoaPod [Firestore](#)

Firestore

[Firestore](#) . [Firestore](#) .

Firestore :

```
// Obj-C
#import Firebase;
```

```
// Swift
import Firebase
```

```
application:didFinishLaunchingWithOptions: FIRApp application:didFinishLaunchingWithOptions:
method :
```

```
// Obj-C
[FIRApp configure];
```

```
// Swift
FIRApp.configure()
```

Firestore

```
// Obj-C
FIRFirestore *storage = [FIRFirestore storage];
```

```
// Swift
let storage = FIRFirestore.storage()
```

Firestore

```
// Obj-C
FIRFirestoreReference *reference = [[storage reference] child:@"path/to/file.txt"];
```

```
// Swift
let reference = storage.reference().child("path/to/file.txt")
```

Firestore

```
// Obj-C
NSData *data = ...
FIRFirestoreUploadTask *uploadTask = [riversRef putData:data metadata:nil
completion:^(FIRFirestoreMetadata *metadata, NSError *error) {
    if (error != nil) {
        // Uh-oh, an error occurred!
    } else {
        // Metadata contains file metadata such as size, content-type, and download URL.
        NSURL downloadURL = metadata.downloadURL;
    }
}];
```

```
// Swift
let data: NSData! = ...
let uploadTask = riversRef.putData(data, metadata: nil) { metadata, error in
    if (error != nil) {
        // Uh-oh, an error occurred!
    } else {
        // Metadata contains file metadata such as size, content-type, and download URL.
        let downloadURL = metadata!.downloadURL
    }
}
```

<https://riptutorial.com/ko/firebase/topic/4281/>

19:

Examples

firebase

Firestore . . .

firebase queue firebase . . .

. NodeJs

1. Google firebase

2. firebase .bower install firebase --save

3. firebase (-> -> -> -> " ") -> json .

4. NodeJs

5. queue/specs

"request_response":

```
{
  "error_state": "request_error_processing",
  "finished_state": "finished_state",
  "in_progress_state": "request_in_progress",
  "start_state": "request_started"
}
```

6. NodeJs firebase server side version npm install firebase --save npm install firebase --save 3 json .

firebase.initializeApp ({serviceAccount : './your_file.json', databaseURL : 'get_from_firebase_account'});

1. firebase web sdk ' / ' firebase .

2. , queue/tasks/{taskKey} . ,

1. ' / '

2. queue/tasks/response queue/tasks/response

3.

```

function createPromiseCallback() {
  var cb;
  var promise = new Promise(function (resolve, reject) {
    cb = function (err, data) {
      if (err) return reject(err);
      return resolve(data);
    };
  });
  cb.promise = promise;
  return cb;
}

```

```

function sendRequest(kind, params, cb) {

  cb = cb || createPromiseCallback();
  var requestObject = {
    kind: kind,
    params: params
  };
  var tasksRef = firebase.database().ref('queue/tasks');

  var requestKey = tasksRef.push().key;

  var requestRef = tasksRef.child(requestKey);

  function requestHandshake(snap) {
    if (snap && snap.exists() && (snap.val().response || snap.val()._state ===
config.firebase.task.finishState || snap.val()._error_details)) {
      var snapVal = snap.val();
      if (snapVal._error_details) {
        cb(snapVal._error_details.error);
      } else {
        cb(null, snapVal.response);
      }
    }
    requestRef.off('value', requestHandshake);
  }

  var bulkUpdate = {};
  bulkUpdate['queue/tasks/' + requestKey + '/request'] = requestObject;
  bulkUpdate['queue/tasks/' + requestKey + '/_state'] = config.firebase.task.startState;

  firebase.database().ref().update(bulkUpdate)
  .then(function (snap) {
    requestRef.on('value', requestHandshake);
  }).catch(function (err) {
    cb(err);
  });

  return cb.promise;
}

```

```
sendRequest('CreateHouseFacade', {houseName:'Test'})
```

```

const database = firebase.database();
const queueRef = database.ref('queue');

const queueOptions = {
  'specId': 'request_response',
  'sanitize': false,
  'suppressStack': false,
  'numWorkers': 3
};

function removeTask(task) {
  var taskRef = queueRef.child(`tasks/${task._id}`);
  return taskRef.remove();
}

function processTask(data, progress, resolve, reject) {
  try {
    requestHandler(data.request).then(response => {
      data.response = response || null;
      return resolve(data);
    }).catch(err => {
      return reject(err);
    }).then(snap => {
      removeTask(data);
    });
  } catch (err) {
    reject(err).then(snap => removeTask(data));
  }
}

function requestHandler(request) {
  if (!request || !request.kind) throw new Error('Absent Request or Kind');
  var deferredResponse = requestHandlerFactory(request.kind, request.params);
  return deferredResponse;
}

function requestHandlerFactory(kind, params) {
  // It includes mapping all your backend services
  switch (kind) {
    case 'CreateHouseFacade': return myService(params)
    default: throw new Error(`Invalid kind ${kind} was specified`);
  }
}

```

myService CreateHouseFacade .

: <https://riptutorial.com/ko/firebase/topic/7619/>

| S. No | | Contributors |
|-------|--|---|
| 1 | firebase | Ami Hollander , Community , Dan Levy , Devid Farinelli , ErstwhileIII , Gabriele Mariotti , RyanM , Sneh Pandya , TwiterZX , Vishal Vishwakarma |
| 2 | Firestore Firestore | Devid Farinelli , eikooc , Frank van Puffelen , Ron Royston |
| 3 | Firestore ? | cutiko |
| 4 | Firestore | Francisco Durdin Garcia |
| 5 | Firestore | Vishal Vishwakarma |
| 6 | Firestore | Priyank Bhojak |
| 7 | FirestoreUI | Mike McDonald |
| 8 | FirestoreUI () | Willie Chalmers III |
| 9 | | Rexford |
| 10 | Firestore | Akshay Khale , Laurel , Noushad PP , Shiven |
| 11 | | ThunderStruct |
| 12 | | Frank van Puffelen , Gabriele Mariotti , riggaroo , Sasxa , Velko Ivanov |
| 13 | ? | Frank van Puffelen , ThunderStruct |
| 14 | | Aawaz Gyawali |
| 15 | Firestore | Dhaval Solanki , Frank van Puffelen |
| 16 | FirestoreAdapter FirestoreFirestoreAdapter ? | Dhaval Solanki |
| 17 | | Gabriele Mariotti |
| 18 | | Mike McDonald |
| 19 | | Vladimir Gabrielyan |