



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Study of Google Firebase API for Android

Navdeep Singh

Masters of Technology Student, Department of Computer Engineering, Indo Global College of Engineering and Technology, Mohali, Punjab, India

ABSTRACT: The purpose of this study is to introduce everyone with Google firebase API and its features. Firebase is a google provided API for database storage and syncing into your android, iOS or web application. A real-time database is one which stores data to database and fetches data from it very quickly but Firebase is not just a real-time database, it is much more than that. This article will cover how to use firebase as a database backend for your android application development. Google firebase have many features like Authentication, database, storage, hosting, notification etc. This study includes using many of these features of firebase into your android application.

KEYWORDS: Database, API (application programming interface), backend, upload, fetch, SDK (software development kit), Authentication

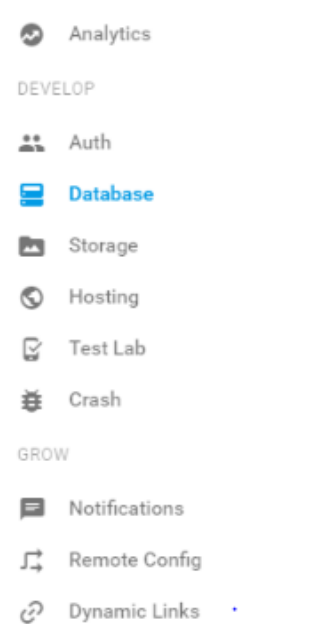
I. INTRODUCTION

A Database is an organised collection of data. Databases can be stored locally on your computer or can be stored in cloud storages. Every application whether android, iOS or web application, it has its own database. In the android application, we can create databases using SQLite, shared preferences, websites or some cloud-based storage sites. The basic idea behind creating database is to store data systematically and fetch data when required. Firebase is also a database backend for android, iOS and web applications. Firebase is google provided API to create a database and fetch from it in real time with only a few lines of code. Data is stored as JSON and is accessible from all the platforms. Firebase is a paid service and you get 200 MB of storage space for free.

II. FIREBASE FEATURES

Once firebase API included into android or iOS application, one can use firebase features with simple lines of code. How to add firebase to the android application is covered in next section. Firebase provided features are listed below:

1. **Analytics:** this feature enables the application developer to understand how users are using his application. The SDK capture events and properties on its own and also allows you to get custom data. The dashboard provides details like your most active user or what feature of your application is used most. It also provides you with summarised data.
2. **Authentication:** Auth feature in firebase let you let only authorised users access your application. Firebase provides login through Gmail, Github, twitter facebook and also let the developer create custom authentication.
3. **Messaging:** firebase cloud messaging lets you to deliver messages to different platforms at no cost. Messaging is also used for notifications purposes.
4. **Real-time Database:** Database in firebase is a cloud-based database and does not need SQL-based queries to store and fetch data. Database is highly reliable and superfast means that data is updated and synchronised in no time and data is maintained even user lose internet connection
5. **Storage:** firebase also provides storage facility. It can store and retrieve content like images, videos and audio directly from client SDK. Uploading





International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

and downloading is done in the background. Data stores are safe and the only authorised user can access it.

6. **Hosting:** firebase is also used for hosting purposes. Firebase delivers web content very fast and content is always delivered securely
7. **Crash reporting:** crash reporting feature on firebase creates reports of error in your app after its release. Errors are grouped into different groups according to how severe error is. You can also create custom events to catch steps leading to the crashing of the application.

III. ADDING FIREBASE TO YOUR PROJECT

You can add firebase API to any project intended for Android 2.3 (Gingerbread) or newer and have Google Play services 9.6.1 or newer.

Following steps are followed next:

1. You have to create a project in firebase console. Enter project name and your location. Project name under firebase console can be anything and can also be different from your application name.
2. Click on **Add Firebase to your Android app** and follow steps mentioned there.
3. You will be prompted with a page asking your package name and optional SHA-1 certificate. You can get your SHA-1 code from android studio easily.
4. After completion of above steps a google-services.json file will be downloaded. You can also download this file again at any time.
5. Copy this downloaded file into your project's app/ module folder. After these steps, firebase is added to your project.

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](#)

After adding firebase to your android project you have to integrate Firebase libraries into your project by following steps:

1. Add rules to your **build.gradle** file, to include google-services plugin:

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

2. Then in app/build.gradle file add:
apply plugin: 'com.google.gms.google-services'
3. Then you can add dependencies according to your use from table below:

Required for Dynamic Links, Invites, and Google Sign-In support in Auth. Edit SHA-1s in Settings.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Gradle Dependency Line	Service
com.google.firebase:firebase-core:9.6.1	Analytics
com.google.firebase:firebase-database:9.6.1	Realtime
com.google.firebase:firebase-storage:9.6.1	Storage
com.google.firebase:firebase-crash:9.6.1	Crash Re
com.google.firebase:firebase-auth:9.6.1	Authentic
com.google.firebase:firebase-messaging:9.6.1	Cloud Me

IV. USING FIREBASE FEATURES IN ANDROID APPLICATION

Using all features of firebase in your android application is very easy and is just a few lines of code. We will talk about features like authentication, database, storage in section. You can find more details about these features and more on google firebase guide link listed in thereference section of this article. Way to use some features is as follow:

1. **Authentication:** after adding firebase and authentication dependency to your android application, you can let your user create login id by thefollowing code:

- First, add FirebaseAuth object in your onCreate method

```
private FirebaseAuth mAuth;  
mAuth = FirebaseAuth.getInstance();
```

- You can create a new account by asking for an email and setup password using createUserWithEmailAndPassword function:

```
mAuth.createUserWithEmailAndPassword(email, password).  
addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
    @Override  
    public void onComplete(@NonNull Task<AuthResult> task) {  
        if (task.isSuccessful()) {  
  
            Toast.makeText(EmailPasswordActivity.this, "successful", Toast.LENGTH  
                _SHORT).show();  
        }  
    }  
});
```

- You can setup an AuthStateListener to check login status

```
private FirebaseAuth.AuthStateListener mAuthListener;  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    mAuthListener = new FirebaseAuth.AuthStateListener() {  
        @Override  
        public void onAuthStateChanged(@NonNull FirebaseAuth  
            firebaseAuth) {  
            FirebaseUser user = firebaseAuth.getCurrentUser();  
            if (user != null) {  
                // User is signed in  
            } else {  
                // User is signed out  
            }  
        }  
    };  
}
```

You can also let user login from google, facebook, twitter, GitHub also. You can read about it on google firebase guide.

2. **Database:** firebase real-time database feature is very easy to use. One you have added firebase and database dependency to your app, you can add unstructured data to database by thefollowing code:

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

```
FirebaseDatabase database = FirebaseDatabase.getInstance();  
DatabaseReference myRef = database.getReference("message");  
myRef.setValue("Hello, World!");
```

To add structured data there are four methods:

- setValue() - this functions write or replace data to current DatabaseReference.
- push() - this function add list entries to current DatabaseReference with random unique ID.
- updateChildren() – Updates some keys of reference without replacing all of them.
- runTransaction() – Updates data carefully so it is not corrupted by concurrent updates.

Following is how data is stored in firebase console:



You can fetch data by adding a listener named ValueEventListener to DatabaseReference by following code:

```
myRef.addValueEventListener(new ValueEventListener() {  
    @Override  
    public void onDataChange(DataSnapshot dataSnapshot) {  
        String value = dataSnapshot.getValue(String.class);  
        // data is stored in string value and can be used according to need  
    }  
    @Override  
    public void onCancelled(DatabaseError error) {  
        // if data is not fetched this function is used  
    }  
});
```

there exists another listener named ChildEventListener and have four functions:

```
ChildEventListener childEventListener = new ChildEventListener() {  
    @Override  
    public void onChildAdded(DataSnapshot dataSnapshot, String previousChildName) { //  
        retrieve list of children and listens to addition of items into list  
    }  
    @Override  
    public void onChildChanged(DataSnapshot dataSnapshot, String previousChildName) {  
        // listens for a change to items of list  
    }  
};
```



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

```
}  
    @Override  
    public void onChildRemoved(DataSnapshot dataSnapshot) {  
        // listens to removal of any child from the list  
    }  
    @Override  
    public void onChildMoved(DataSnapshot dataSnapshot, String previousChildName) {  
        // this function listens to change in order list  
    }  
    @Override  
    public void onCancelled(DatabaseError error) {  
        // if data is not fetched this function is used  
    }  
};
```

Sort data: you can also retrieve data in sorted order by using one of the following functions:

orderByChild() - orders results by the value of child key.

orderByKey() - orders results by child keys.

orderByValue() - Orders results by child values.

Delete data: the simplest way to delete data under current DatabaseReference is by calling removeValue() function.

3. **Storage:** storage feature is for those who want their apps to store files like images, audio, video etc. Data stored is highly secured and robust means they resume from last point if network error occur. Following steps are followed to use storage feature in android application:

- After adding firebase and storage dependency to your application create instance of FirebaseStorage:

```
FirebaseStorage storageobject =FirebaseStorage.getInstance();
```

- In the next step we create reference to location by:

```
StorageReference FileRef = storageRef.child("filePath");
```

- You can upload file by using one of putBytes(), putFile(), putData() or putStream() method which returns to UploadTask:

```
UploadTask uploadTask =FileRef.putBytes(data);
```

you can check status of the upload by adding listener to UploadTask object:

```
uploadTask.addOnFailureListener(newOnFailureListener() {
```

```
    @Override
```

```
    public void onFailure(@NonNullException exception) {
```

```
        // Handle unsuccessful uploads
```

```
    }
```

```
}).addOnSuccessListener(newOnSuccessListener<UploadTask.TaskSnapshot>() {
```

```
    @Override
```

```
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
```

```
        // handles successful uploads
```

```
    }
```

- Using above reference, we can download data by methods like getBytes() or getStream():

```
StorageReference FileRef = storageRef.child("images/island.jpg");
```

```
FileRef.getBytes().addOnSuccessListener(newOnSuccessListener<byte[]>() {
```

```
    @Override
```

```
    public void onSuccess(byte[] bytes) {
```

```
        // handles successfully downloaded data    }
```

```
}).addOnFailureListener(newOnFailureListener() {
```

```
    @Override
```

```
    public void onFailure(@NonNullException exception) {
```



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

```
// Handle any errors  
});
```

V. WHY USE FIREBASE

We have discussed many features of firebase that are not available in most of other database backend services. Even if you find all features somewhere else you can't find all of these features under one API. Firebase is a real-time database which makes it stand above all the traditional database services like SQL, SQLite, shared preference etc. firebase is added to your app very quickly and app using firebase requires least development time. Under firebase you do not miss any step, you have to follow all the steps to make your app run. Firebase is provided by google itself which means its most reliable and going to have a lot of new features in future.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we have studied about google provided firebase API and its unique features. We have studied how to add firebase to our android project and how to use different features in our android application according to our need. Google is adding new features to firebase continuously, AdSense for firebase is in beta phase. So in future, we can study those new features and how to implement them in our android application and why to keep using firebase than any other database service out there.

REFERENCES

1. Isuru Madusanka, Busy programmer's guide to Firebase with Android, 2016
2. Android guide- Firebase (<https://firebase.google.com/docs/>)
3. Connect to Firebase | Android studio (<https://www.developer.android.com/studio/write/firebase.html>)
4. Introduction to Firebase by Adam Moore on CodePen (<https://codepen.io/adamaoc/post/introduction-to-firebase>)
5. Using firebase and App Engine Standard Environment in an Android app (<https://cloud.google.com/solutions/mobile/firebase-app-engine-android-studio>)
6. Bill Stoneham, Google Android Firebase: Learning the Basics Paperback, 2016
7. Firebase Tutorial: Building a Realtime App with Firebase - AirPair (<https://www.airpair.com/firebase/posts/firebase-building-realtime-app>)
8. Firebase Tutorial: Getting Started - Ray Wenderlich (<https://www.raywenderlich.com/139322/firebase-tutorial-getting-started-2>)
9. Firebase Android Tutorial - Reading/Writing Firebase Data (<https://www.simplifiedcoding.net/firebase-android-tutorial-writing-firebase-data/>)
10. Firebase tutorials, tutorialspoint (<http://www.tutorialspoint.com/firebase/>)
11. University of California San Diego, computer science engineering class pdf (<https://cseweb.ucsd.edu/classes/wi16/cse110-a/applications/ln/cse110-discussion6.pdf>)
12. Firebase API | Codecademy (<https://www.codecademy.com/en/tracks/firebase>)

BIOGRAPHY

Navdeep Singh is a student pursuing Masters of Technology Student, Department of Computer Engineering, Indo Global College of Engineering and Technology, Mohali, Punjab, India.