Application of Firebase in Android App Development-A Study

Chunnu Khawas
Department of Computer Applications
Sikkim University
Gangtok, India

Pritam Shah
Department of Computer Applications
Sikkim University
Gangtok, India

ABSTRACT

The web application has become more and more reliant upon large amount of database and unorganized data such as videos, images, audio, text, files and other arbitrary types. It is difficult for Relational Database Management System (RDBMS) to handle the unstructured data. Firebase is a relatively new technology for handling large amount of unstructured data. It is very fast as compared to RDBMS. This paper focuses on the application of Firebase with Android and aims at familiarizing its concepts, related terminologies, advantages and limitations. The paper also tries to demonstrate some of the features of Firebase by developing an Android app.

Keywords

Firebase, Android, RDBMS

1. INTRODUCTION

The server used for Android apps are Oracle SQL, Microsoft SQL Server, and MySQL which are connected to the server with PHP files. Then Firebase came into existence for Android apps which uses JSON for storing data. The other servers use a table (rows and columns) format for storing data.

Firebase is NoSQL based. There are very few cloud based server available which are similar to firebase, like:

AWS Mobile Hub- It is integrated console that helps to create, build, test, and monitor the mobile apps that leverages AWS services.

CloudKit- It is an Apple framework which helps to save data and store assets but similar to iOS only.

Parse Server- It was released by Facebook to replicate functionality of Parse which is an open source server. This is no longer in existence as Facebook shutdown this project.

1.1 Firebase

Firebase is considered as web application platform. It helps developers' builds high-quality apps. It stores the data in JavaScript Object Notation (JSON) format which doesn't use query for inserting, updating, deleting or adding data to it. It is the backend of a system that is used as a database for storing data.

The services available are:

1.1.1 Firebase Analytics

It provides insight into app usage. It is a paid app measurement solution that also provides user engagement. This unique feature enables the application developer to understand how users are using the application. The SDK has the feature of capturing events and properties on its own and also allows getting custom data.

1.1.2 Firebase Cloud Messaging (FCM)

It is formerly known as Google Clouds Messaging (GCM), FCM is a paid service which is a cross-platform solution for messages and notifications for Android, Web Applications, and IOS.

1.1.3 Firebase Auth

Firebase Auth supports social login provider like Facebook, Google GitHub, and Twitter. It is a service that can authenticate users using only client-side code and it is a paid service. It also includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

1.1.4 Real-time Database

Firebase provides services like a real-time database and backend. An API is provided to the application developer which allows application data to be synchronized across clients and stored on Firebase's cloud. The client libraries are provided by the company which enables integration with Android, IOS, and JavaScript applications.

1.1.5 Firebase Storage

It facilitates easy and secure file transfer regardless of network quality for the Firebase apps. It is backed by Google Cloud Storage which is cost-effective object storage service. The developer can use it to store images, audio, video, or other user-generated content.

1.1.6 Firebase Test Lab for Android

It provides cloud-based infrastructure for testing Android apps. With one operation, developers can initiate testing of their apps across a wide variety of devices and device configurations. The various test results like screenshots, videos and logs are available in the Firebase console.. Even if a developer hasn't written any test code for their app, Test Lab can exercise the app automatically, looking for crashes.

1.1.7 Firebase Crash Reporting

The detailed reports of the errors are created in the app. The errors are grouped into clusters of similar stack traces and triaged by the severity. The other features are: the developer can log custom events to help capture the steps leading up to a crash.

1.1.8 Firebase Notifications

It enables targeted user notifications for mobile app developers and the services are freely available.

2. LITERATURE REVIEW

Walter Kriha [1] in his article has mention the systematic overview of NoSQL database and common concepts, techniques and patterns as well as several classes of NoSQL databases (key/value-stores, document databases, column-oriented databases) and individual products. Various advantage and disadvantage of using NoSQL database has been discussed.

Supriya S. Pore, Swalaya B. Pawari [2] conducted a comparative study of SQL and NoSQL. The study highlights on the types of databases like SQL and NoSQL, it also differentiates among them. The Axiomatics of SQL and NoSQL databases has been described in this paper. The study says that due to data consistency, ACID property is not used in the NoSQL databases

Vatika Sharma, Meenu Dave [3] have given an overview of NoSQL databases focusing on how it has declined the dominance of SQL with its background and characteristics.

Daniel Pan [4] in his article has shown how to connect firebase to an Android app and basics of designing the structure of database in Firebase.

Landon Cox [5] study highlights the comparison between SQLite and firebase. It also focuses on organizing data in the form of JSON tree in order to store in Firebase.

3. COMPARISON OF VARIOUS DBMS

The given Table 1 below highlights the comparison of Firebase and SQL base on parameters such as data storage,

flexibility in terms of Schema etc. On the other hand, the comparison of Firebase is done with MS SQL Server based on

parameters such as developer; data model etc and has been tabulated in Table 2.

Table 1: Comparison between Firebase and SQL

Basis of comparison	**Firebase**	**SQL(RDBMS)**
Data Storage	Stored as JSON Tree	Stored in a Relational Model as Rows and Columns (Tables)
Schema flexibility	Dynamic Schema, data can be added, updated or deleted anytime	Fixed schema. Altering will result in going offline temporarily
Specialty	Data which has no definite type or Structure	Data whose type is known in advance
Technique	Synchronize data	Fire Query

Table 2: Comparison between Firebase and MS SQL Server

Basis of Comparison	**Firebase**	**MS SQL Server**
Data Model	Stored as JSON Tree	Table
DB Engine Ranking	Score-2.79 Rank-79 Overall, 13 Key-Value Pairs.	Score-94 Rank-3 Overall, 3 RDBMS
Website	Firebase.google.com	Microsoft.com./en-us/SQL-server
Developer	Google	Microsoft
Database Schema	Schema Free	Yes
SQL	No	Yes
Supported Languages	Java JavaScript Swift	C++, Delphi, Go Java, JavaScript PHP, Python Ruby, Visual Basic
Initial release	2012	1989

4. STEPS TO ADD FIREBASE TO ANDROID

Firebase can be added to any project intended for Android 2.3 (Gingerbread) or newer version and have Google Play services 9.6.1 or newer. Following steps are to be followed next:

A project has to be created in firebase console. Enter project name and the location. Project name under firebase console can be anything and can also be different from the application name given.

Click on Add Firebase to the Android app and follow steps mentioned there.

The user will be prompted with a page asking for the package name and optional SHA-1 certificate. SHA-1 code is available at android studio.

At the end google-services.json file will be downloaded once the above steps are completed. The file can be downloaded again at anytime.

The downloaded file has to be copied into the project's app or module folder. Firebase is added to the project.

5. USING FIREBASE FEATURES IN ANDROID APPLICATION

Using all features of firebase in the Android application is very easy and is just a few lines of code. The features like authentication, database, and storage have been discussed in section. The other more detail about the features are available on Google Firebase guide link listed in the reference section. The methodology to use some features are as follow:

5.1.1 Authentication:

After adding firebase and authentication dependency to the Android application, the user can create login id by the following code [6]:

```
FirebaseAuth auth=FirebaseAuth.getInstance();

auth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(new OnCompleteListener())

{

 @Override

  public void onComplete (Task task)

     {

        if(task.isSuccessful())

          {

            FirebaseUser  user=task.getResult().getUser();
            String email=user.getEmail();
              //…
            }
        }
});
```

User login can also be used from Google, Facebook, Twitter, GitHub.

5.1.2 Database

Firebase real-time database feature is very easy to use. Once the Firebase and database dependency is added to the app, unstructured data can be added to database by the following code [7]:

```
//Write a message to the database
FirebaseDatabase database= FirebaseDatabase.getInstance;
DatabaseReference myref=database.getReference(“node”);
myRef.setValue (“Hello, World”);
```

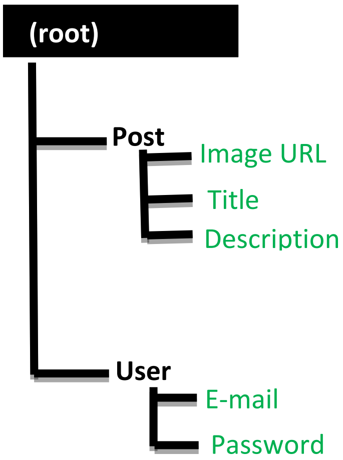

Figure 1: Firebase Data Structure.

In Figure 1: the data structure of Firebase is shown where handshake is the root of the app and there is two child of the root that is post and user. Again, the post and user has their child nodes accordingly.

5.1.3 Storage

The files like images, audio, video etc can be stored in the app. The data stored is highly secured and is robust in nature means it resumes from the last point if any network error occurs. The steps below are to be followed to use storage feature in Android application:

- Once the Firebase and storage dependency are added to the application, create instance of FirebaseStorage storageobject =FirebaseStorage.getInstance();
- Second, create the reference to location by: StorageReference FileRef = storageRef.child("filePath");
- The file can be uploaded by using one of putBytes(), putFile(), putData() or putStream() method which returns to UploadTask.

5.1.4 Inserting New Data

An object class or a map can be used to insert new data.

Once the object is created, navigate the Firebase reference to the position where a child can be added.

If a list is created and does not have a specific names for each child, the push() method can be used before the setValue() is called.

Call: ref.push().setValue(object) or ref.setValue(object).

5.1.5 Updating Data

Navigate the Firebase reference to the parent of the item that one wants to update

Then a map containing the update values can be created.

Call ref.updateChildren(map).

5.1.6 Removing Data

Navigate the Firebase reference to the item that one wants to remove

Call ref.removeValue(object).

6. FIREBASE AND ANDROID APP

An Android application has been developed for the demonstration of Firebase. In this app images along with strings are loaded to Firebase and retrieved from Firebase similar to Instagram. For the development of an Android app to demonstrate the use of Firebase, prototyping model has been followed.

Steps for connecting App to Firebase:

Step1: An account in the Firebase Login has to be created at https://www.firebase.com/login/ using the Google account.

Step2. Creating a new application on Firebase. Firebase creates a new application when one logs in for the first time. Also, at the bottom left corner, one can find an option to create a new application on the Firebase server. The app url has to be unique among all applications deployed on Firebase.

Step3. Next step is to add Firebase as a project dependency. Make changes to the following lines to the build.gradle file, which is located in the app’s project folder, and not the root folder.

```
dependencies
{    compile ‘com.firebase:firebase-client-android:2.5.0+'
}
```

After adding any dependency one has to make sure to sync the application.

If there is any build error complaint about duplicate files then one can choose to exclude those files by adding the packagingOptions directive to the build.gradle file:

```
android
{    ...    packagingOptions
{        exclude 'META-INF/LICENSE'        exclude 'META-INF/LICENSE-FIREBASE.txt'        exclude 'META-INF/NOTICE'
}}
```

4. Next, add permissions to Android application, Add network permission to the app, the same way it has been done for parse earlier. Now add the following line to the AndroidManifest.xml file:

```
<uses-permission
```

android:name="android.permission.INTERNET" />

The User-Interface is designed for the android application in Figure 2 .The name of the app is given handshake. It has a view of image along with title and description.

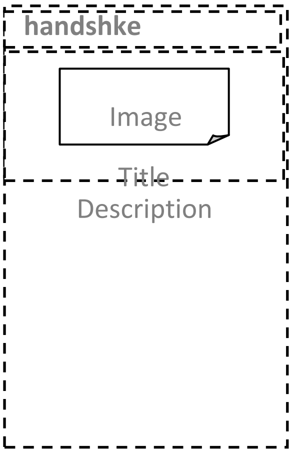

Figure 2: User Interface Design

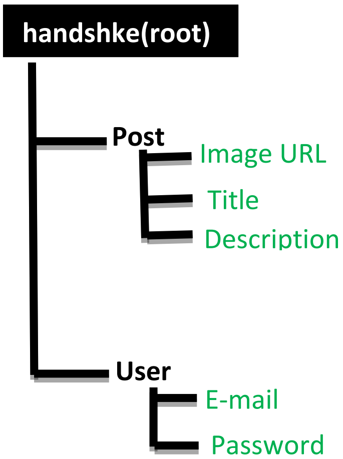

Figure 3: Database Structure of app.

Further in Figure 3, the data structure of the app is designed for storing data. The root of the app is handshke with two child nodes that is post and user. The post node has again three childs for storing images url, title and description of the image.

User has two child for storing email-id and passwords of user.

7. RESULT

The screenshots of the app developed using Firebase have been included:

Figure 4.a is the home screen of the app where user can select images for posting along with title and description of the image.

Figure 4.b is the post screen of the app where users can see the posts of other users.

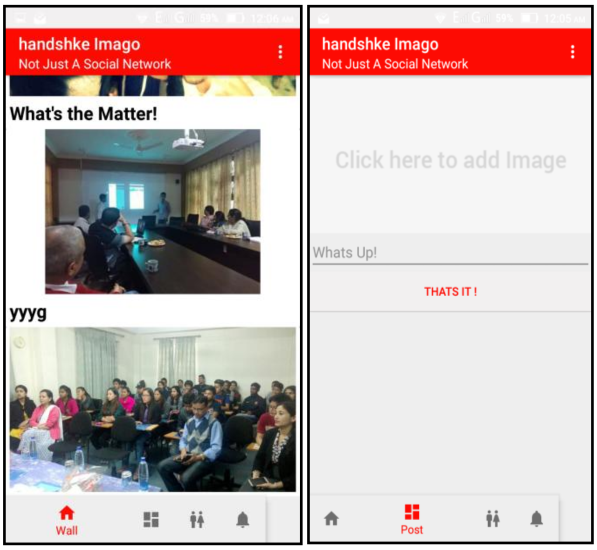

Figure 4: (a) (b)

a. **Main Activity of app which allows users to upload images along with caption.**

b. **Wall Activity of app where users can view posts of others users.**

8. CONCLUSION AND FUTURE SCOPE

This paper highlights on the study about google provided Firebase API and its unique features. This paper helps in studying how to use Firebase in the Android application according to the developer requirement. This also helps in making android apps faster and efficient as no PHP is required as a third party language to communicate with the database. It provides a secure channel to communicate with the database directly from JAVA. The study material is based on the data provided online and referring to the examples given. Google has been updating Firebase on regular basis, AdSense is the beta phase of Firebase. It can not only be used in Android but also to connect cross platform .The work can be further extended by adding new features and exploring new possibilities in Android applications.

9. REFERENCES

[1] Kriha Walter, 2009 NoSQL Databases Hochschule der Medien. Stuttgart Media University. Stuttgart.

[2] Pore Supriya S, Pawar Swalaya B, 2015. Comparative Study of SQL & NoSQL Databases. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 4 Issue 5, May 2015

[3] Sharma Vatika, Dave Meenu. 2012. SQL and NoSQL Databases. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 8, August 2012.

[4] Daniel Pan. 2016. Firebase Tutorial. October, 2016.

[5] Cox Landon. 2017. SQLite in Android. March 2017.

[6] Kalsov, 2012. Developer Meet Firebase dated 18/3/17.

[7] "Firebase Realtime Database". Firebase, Inc dated 18/3/17.

[8] Bill Stonehem, Google Android Firebase: Learning the Basics Paperback, 2016 dated 18/3/17.

[9] Isuru Madusanka, Busy programmer’s guide to Firebase

with Android, 2013.

[10] G. Harisson, “10 things you should know about NoSQL databases” dated 18/3/17.

[11] http://www.techrepublic.com/blog/10-things/10-things-you-should-know-about-nosql-databases/ dated 18/3/17

[12] http://developerfirebase/android.com.