### **Connecting android application to firebase cloud**

Configuration steps:

1. To have a project in Firebase cloud we have first created a Gmail account and then logged in to Firebase cloud by that account.
2. Then we created a Firebase project and registered our app with it.
3. After that, a google-service.json file which is Firebase configuration file was given to be added to the Android project. To enable Firebase products in our app, we added the following google-services plugin to build.gradle file:

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

1. The dependency for Realtime Database was also added to the build.gradle file:

* dependencies {  
  implementation **'com.google.firebase:firebase-database:16.0.4'** }

1. Also, we configured Firebase Database Rules. To be able to send and receive data to the database we defined the database rules of read and write as true which allows to read and write to it.
2. After doing all the above configuration, we can start reading or writing to the database.

Retrieving data from Firebase Realtime Database:

Data stored in a Firebase Realtime Database is retrieved by attaching an asynchronous listener to a database reference. The listener is triggered once for the initial state of the data and again anytime the data changes. The following table illustrate witch listener was used:

|  |  |  |
| --- | --- | --- |
| Listener | Event callback | Typical usage |
| ValueEventListener | onDataChange() | Read and listen for changes |

Inserting data to Firebase Realtime Database:

The Realtime Database accepts multiple data types **String, Long, Double, Boolean, List<Object>** to store the data. It also allows us to use **custom java objects** to store the data which is very helpful when storing model class directly in database.

For example: to store user information,

* Step1: we created User model with all properties that we decided to store
* Step2: we got the reference to **‘**users’ node.
* Step3: we used the reference to generate a unique Id by calling push() method which creates an empty node with unique key.
* Step4: we created user object
* Step5: we used the generated unique Id in step3 to push user to 'users' node

Deleting data from Firebase Realtime Database:

To delete data, we simply called removeValue() method on to database reference.

To read the data, you need to attach the **ValueEventListener()** to the database reference. This event will be triggered whenever there is a change in data in realtime. In **onDataChange()** you can perform the desired operations onto new data.

Firebase realtime database is a schemaless database in which the data is stored in JSON format. Basically the entire database is a big JSON tree with multiple nodes.