### **Connecting the website to firebase cloud**

Configuration steps:

1. Creating an account and project on firebase as mentioned in (XXX).

* After accomplishing steps above, starting with configuration steps by obtaining the config object. This required opening the project and selecting “Add Firebase to the web app”, and finally copying the snippet shown below and adding it the front-end.

var config = {

apiKey: "AIzaSyCYgApFzAtXIVgdI4J09RWseIxlqfAS8hA",

authDomain: "fir-auth-45665.firebaseapp.com",

databaseURL: "https://fir-auth-45665.firebaseio.com",

projectId: " fir-auth-45665",

storageBucket: " fir-auth-45665.appspot.com",

messagingSenderId: "777762792693",

};

1. Configuring Nodejs app by firstly installing package.json and firebase npm package after running the commands below:

npm init

npm install --save firebase

1. Using Firebase module after requiring them from JavaScript.

var firebase = require("firebase/app");

require("firebase/auth");

require("firebase/database");

1. Initializing the Realtime database and configuring the rules as mentioned in (XXX)

firebase.initializeApp(config);

// Get a reference to the database service

var database = firebase.database();

1. After doing all the above configuration, we can start reading or writing to the database.

Retrieving data from Firebase Realtime Database:

Refer to XXX

|  |  |  |
| --- | --- | --- |
| Event | Event callback | Typical usage |
| Value | On() and once() | 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:

In order to delete data from Firebase, we call **remove()**on reference to the data we want to delete.

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.