### **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 logic of adding to database is same as

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.

* + 1. Connecting the website to firebase cloud

To enable reading and writing from and to Firebase database, the following steps were executed:

Configuration steps:

1. Create an account and project on firebase as mentioned in (section 5.1.2).
2. After accomplishing steps above, starting with configuration steps by obtaining the config object. This config object is unique identifiers for our Firebase project. This required doing the following steps:
3. Opening the overview page and selecting “Add app”
4. Choosing “Add Firebase to your web app”.
5. Copying the snippet shown below and adding it a new HTML page.

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. Configue Nodejs app by installing the package.json and firebase npm packages 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 (5.1.2)

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, inserting and deleting data in Firebase Real-time Database:

Retrieving, inserting and deleting data in the website have the same logic as application however website uses different type of listeners and functions.

The following table illustrate the type of listener used to retrieve data:

Table ‎5.2: Type of event used

|  |  |  |
| --- | --- | --- |
| Event | Event callback | Typical usage |
| Value | On() and once() | Read and listen for changes |

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.