Reading / Writing to your Firebase Realtime Database Instance using its Rest API (2024)

This is an optional resource for your 1D Project and is not tested.

It is possible for you to CRUD with your Firebase Realtime Database using a REST API. Here's the documentation. https://firebase.google.com/docs/reference/rest/database

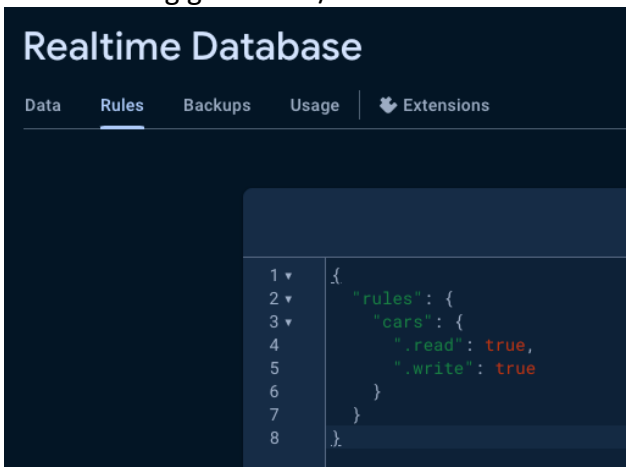
This is a brief guide to point you to the resources that you need to pick it up on your own.

Step 1. Create your Firebase Realtime Database

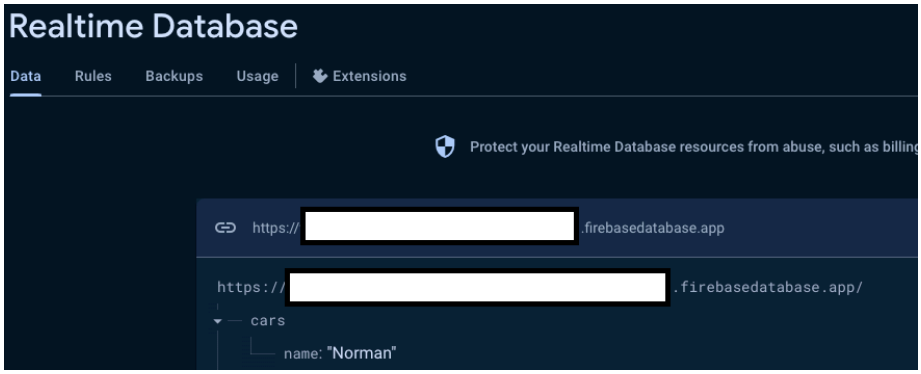
If you don't have a firebase realtime database instance created, watch the 10.014 CTD 2022 Firebase Guide until 1:54.

Step 2. Edit the rules so that you can test your access freely.

The following gives read/write access to ANYONE to your node called **cars**.

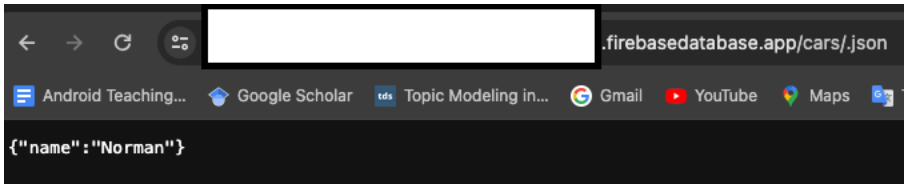


Step 3. Populate the cars node, and get the URL.



Step 4. Go to your browser to see the Json response.

If the URL above is **https://apple.firebasedatabase.app** then enter **https://apple.firebasedatabase.app/cars/.json** in your browser to get the JSON response.



Step 5. You can now write a Java method to get this string.

You can use the HttpURLConnection library to achieve this. By default, a HttpURLConnection is a GET request. You can refer to this article for details.

```
String myFirebaseUrl = ; // put your URL here
// 1. CREATE A HttpURLConnection
URL url = new URL(myFirebaseUrl);
HttpURLConnection con = (HttpURLConnection) url.openConnection();
con.setRequestMethod("GET"); //
System.out.println("HTTP Response Status Code:"+con.getResponseCode());

// 2. Pass the InputStream to BufferedReader object to download the data
BufferedReader in = new BufferedReader(
        new InputStreamReader(con.getInputStream()));


// 3. Read the data from the BufferedReader object and create the String
String inputLine;
StringBuffer content = new StringBuffer();
while ((inputLine = in.readLine()) != null) {
    content.append(inputLine);
}

// 4. Close and Disconnect and display the JSON result
in.close();
con.disconnect();
System.out.println("Downloaded Content:" + content);
```

Step 6. You can use the Gson library to parse the JSON string

Go to the **build.gradle** file for your app or java module and add this in the dependencies section.

```
dependencies {
'com.google.code.gson:gson:2.8.6'
//there may be other dependencies present
}
```

Then learn how to parse JSON with it.

Step 7. Read the Firebase Database Rest API to see the operations possible

https://firebase.google.com/docs/reference/rest/database

For example, to add data to a new node, you will need to do a POST Request.

To update data in an existing node, you will need to do a PATCH Request.

Step 8. Look at Android Lesson 3 on how to handle concurrency

You may have a lot of data to download from Firebase.
In such a situation, code that downloads the data should be executed concurrently with the code that runs the user interface.
How to do this is illustrated in Lesson 3.