Firebase

Documentation on using Firebase service for web development , includes functions

Written by :
Derradji Amine Abdelbasset

I am Derradji Amine Abdelbasset , a cloud service certified associate , web developer and a ICT Student .

I wrote this documentations during my development learning to strenght what I learnt and help my community to master Firebase .

Firebase is a Cloud Platform as a Service (PaaS) provided by Google Cloud to help developers build and manage their applications and projects.

This documentation explains how to use Firebase services and integrate them into web applications.

The main focus of this document is on Functions manipulation, allowing you to understand how all the functions work without needing to memorize them or watch long playlists or complex documentation.

It is designed specifically for beginners and intermediate developers.

Initialization:

After starting a new project in the Firebase platform, you need to connect your Firebase project with your web project. To do that, you must create a configuration file.

Firebase provides something called an SDK (Software Development Kit). First of all, you need to import a function called **initializeApp()**. This function is responsible for loading all the necessary tools.
You also need to import other functions such as **getDatabase()** and **getAuth()**, which are used to access the Firebase Realtime Database and Authentication system, respectively.

As second step , you have to provide some private informations wich are the keys of your project form the console , after you get those keys , you will put them in a object and declare a variable holds all those informations or what we calles "configuration" .

The last step will be about initialization of firebase , here we will use the functions that we import in the top of file , we have to declare variable called app equal the " initializeApp(configVariable) , this variable will holds all what firebase provides for us like db , auths .....We can declare another variable holds a specific service like db or auth by using the same functions that we import from SDK and put app inside them .

after we finished , we will export app and the authers variable to use them in our project file

Saving Data into Realtime Database

1. After preparing the configuration file, don't forget to create a variable using the getDatabase(app) function and export it.
We do this to gain access to our database.

2. In our project file, we will import both the app and the database, along with some specific functions:

ref() function:
The role of this function is to specify (or choose) a path in our database where the data will be saved.
This path should be created in the Firebase console before this step.
The function takes two parameters: the selected database and the path name as a string.
Example: ref(database,).

push( ) function : the real role of this function is add new data in the selected path when it's called up . this function take as a parameter the variable that holds the path of database . It's looks like : push(ref) .

set( ) function : this function put (becarefull put not add) new data in the path you give it to her , but becarefull when you give her the refrence as a parameter , she will deplaced all the saved data by the data that you give it to her , because of that we give her the push function to avoide this problem , it's structure looks like : set(pushVariable, {data}) .

just note: the data that you give as parameter must be a json object because database use this syntax of data + the data is useState data in react .

{
Key: useStateData
}

Project steps :

- import the functions .
- create a refrence and declare it as variable .
- create a ID generator "push" function .
- use the Set function by giving it the location and the data

Those steps must puted in a function excuted when forms submited .

Read data from realtime Database :

Initialization :

- when we use firebase project twice in the same project/browser/serveur we get a problem relys on initialization , and to avoide this prblm we use a specific function called **getApps( )** : it return an array contains all the apps that was initializated before in the same project . we import it from firebase/app , when we declare the app variable we have to use **getApps( )** like that :

```
getApps( ).length ? getApps( ) [0] : initializationApp(firebaseConfig) ;
```

Which mean that if there is a previous

initialization for this app so the length of the array that the **getApps( )** return is bigger than 0 , so the first condition will be excuted - using the first initialization - if there is no any previous initialization the second condition will excuted -initializate new app - .

Project :

We have to import some functions like database , onValue , off, ref and all those functions have specific role :
OnValue( ) : it is a function detected any change in the refrence and excute a some instructions when the change happends , it take tow value as parameter , the refrence and callback function .

Snapshot( ) : it is an object and not a function , this object holds all the data from the refrence and we can bring data from it for **snapshot.val( )** , we can verify if there is a data in the refrence by writting : **snapshot.exists( )** , we use this object as parameter for the callback function of onValue to bring data from database .

Off( ) : is a function famillar with onValue , the role of off function is stopping the detection of onValue , it take the refrence as parameter and we must use it after onValue .

Object.key( ) : is function take the refrence as a parameter and bring all the keys and put them in an array .

We have to mention the data in the refrence by this notation :

user [id].yourKey

Steps:

1. Import the functions ref, onValue, and off from Firebase.
2. Create a new useEffect that includes the following steps:
(a) Create a reference to the desired path in your Realtime Database.
(b) Use the onValue() function with the reference and a callback function that takes a snapshot as a parameter. Inside this callback, add logic to check whether there is data available in the snapshot or not.
(c) After that, use the off() function by passing the same reference as a parameter. This should be done inside the return function of useEffect to properly clean up the listener when the component unmounts.

(d) Finally, use Object.keys() together with a useState variable to retrieve the keys of the data and check whether any data exists. If there is data, display it in a table using an if conditional statement.

Firebase authentification :

After getting the data from Form element , u will use useState to manage data .

SignInWithEmailAndPassword( ): function that take auth, email, and password as parameteres , it verifys the data . You have to use it with try and catch function .

SignOut( ) : function give it to a button to sign out , it never take anything as parameters , just signOut( ) .