

Lesson 7: Becoming a React Native native

Working with Firebase

Karl Steven Velasco Orjalo

May 31, 2023

React Native Quickstart

Lesson Outline

1. What is Firebase?

2. Using Firebase

What is Firebase?

What is Firebase?

Firebase is a cloud services provider that Google runs.

The specific service we will be using in this lesson is Firestore, a no-SQL database. This means that data is classified into documents and collections.

Using Firebase

Taking a look at the Firestore console

Head to `console.firebase.google.com`

Taking a look at the Firestore console (cont.)

Create a new project, then call it `myNewProject`.

This name can be anything really, and it does not have to match the name of your working folder/repo.

Taking a look at the Firestore console (cont.)

Now head to the button that looks like `</>`, where we can 'Add Firebase to your web app'.

Then, we can register our app as `myNewProject`. Again, this doesn't matter too much.

Back in VSCode, in your Terminal enter the following command:

```
1 yarn add firebase
```


Taking a look at the Firestore console (cont.)

Then, in the next step 'Add Firebase SDK', copy the `const firebaseConfig = {}` and its contents.

This will be used in our project.

Setting up Firebase in React Native

Back in VSCode, create a folder in the root folder named `api` then create a file called `firebaseConfig.js`.

The hierarchy should look like `./api/firebaseConfig.js`.

Setting up Firebase in React Native(cont.)

In `./api/firebaseConfig.js`, enter the following code and replace the `fireabseConfig` with what you copied previously.

```
import { initializeApp } from 'firebase/app';
import { getFirestore } from "firebase/firestore";

const firebaseConfig = {
  // Place your firebaseConfig here
};

const app = initializeApp(firebaseConfig);
const db = getFirestore(app);

module.exports = db;
```

Setting up a Firestore database in the Firebase console

In the Firebase console, find the 'Firestore Database' page under the Build button on the left.

In the Firestore Database page, press 'Create database'.

Setting up a Firestore database in the Firebase console (cont.)

Next, select 'Start in production mode' and choose **asia-southeast1 (Singapore)** as the server location.

This will help Singapore users (in this case, us) access the database with lower latency.

Setting up a Firestore database in the Firebase console (cont.)

Press 'Start Collection' and call it `testCollection`. Then, give the first document an ID of `'testDocument'`.

Make two fields of type `string`, named `title` and `body`. Inside the values, give set the `title` to `testTitle` and `body` to `testBody`.

Getting our document into myNewProject

We'll now work with Firestore in our Account page. Import the `databaseConfig` into `./src/pages/Account.jsx`.

```
import { initializeApp } from 'firebase/app';
import { getFirestore } from "firebase/firestore";

const firebaseConfig = {
  // Place your firebaseConfig here
};

const app = initializeApp(firebaseConfig);
const db = getFirestore(app);

module.exports = db;
```

Getting our document into myNewProject (cont.)

Inside `./src/pages/Account.jsx`, we now need to import a couple things. Try typing it out, not copying.

```
import React, {useEffect, useState} from 'react'
import db from '../../api/firebaseConfig.js'
import {collection, onSnapshot} from "firebase/firestore"
import {View, FlatList, Text} from 'react-native'
```


Getting our document into myNewProject (cont.)

Now, let's change the displayed contents of the `<View />` into a `<FlatList />`.

```
return (  
  <View>  
    <FlatList  
      data={data}  
      renderItem={({ item }) => <Text>{item.title}</Text>}  
    />  
  </View>  
)
```

Getting our document into myNewProject (cont.)

Then, create a `data` state.

```
const [data, setData] = useState()
```

Getting our document into myNewProject (cont.)

To pre-load the data and receive realtime updates, we will be using the `useEffect` hook and Firestore's `onSnapshot()`.

```
useEffect(() => {  
  const getDocuments = () => {  
  
    const dbRef = collection(db, 'testCollection')  
  
    onSnapshot(dbRef, (docsSnap) => {  
      var docsArr = []  
  
      docsSnap.forEach(doc => {  
        docsArr.push(doc.data())  
        setData(docsArr)  
      })  
    })  
  }  
  
  return getDocuments()  
}, [])
```

End of Lesson

Questions? Reach out at: karlorjalo@gmail.com