1

# Firebase with react native

## Firebase Realtime Database

The Firebase Realtime Database is a cloud-hosted database.
Data is stored as JSON and synchronized in realtime to every connected client.
When you build cross-platform apps with Apple platforms, Android, and JavaScript SDKs
Pupporting both realtime data sync and offline capabilities.

2

# Firebase with react native

## Installation

This module requires that the @react-native-firebase/app

# Install & setup the app module
*yarn add @react-native-firebase/app*
*Npm install @react-native-firebase/app*

# Install the database module
*yarn add @react-native-firebase/database*
*Npm install @react-native-firebase/database*



3

# Firebase with react native

A core concept to understanding Realtime Database are references - a reference to a specific node within your database. A node can be a specific property or sub-nodes.

To create a Reference, call the ref method
*import database from '@react-native-firebase/database';*

*const reference = database().ref('/users/123');*

NOTE: To get a reference to a database other than a 'us-central1' default database, you must pass the database URL. You can find your Realtime Database URL in the Realtime Database section of the Firebase console
*const reference = firebase*
*.app()*
*.database('https://<databaseName>.<region>.firebasedatabase.app/')*
*.ref('/users/123');*



4

# Firebase with react native

## Reading data

**One-time read**

```
import database from '@react-native-firebase/database';

database()
  .ref('/users/123')
  .once('value')
  .then(snapshot => {
    console.log('User data: ', snapshot.val());
  });
```



5

# Firebase with react native

## Reading data

**Realtime changes**

```
import database from '@react-native-firebase/database';

database()
  .ref('/users/123')
  .on('value', snapshot => {
    console.log('User data: ', snapshot.val());
  });
```



6

# Firebase with react native

## Querying

**Order**
By default, results are ordered based on the node keys. If however you are using custom keys you can use one of the orderByX methods to order your data.
*const scores = database().ref('scores').orderByValue().once('value');*

**Limiting**
*const users = database().ref('users').limitToFirst(10).once('value');*





# Firebase with react native

## Writing data

**Setting data**
The set method on a Reference overwrites all of the existing data at that reference node. The value can be anything; a string, number, object etc:
f you set the value to null, Firebase will automatically class the node as removed, and delete it from the database.

```
import database from '@react-native-firebase/database';

database()
  .ref('/users/123')
  .set({
    name: 'Ada Lovelace',
    age: 31,
  })
  .then(() => console.log('Data set.'));
```

# Firebase with react native

## Writing data

**Updating data**
Firebase will automatically merge the data depending on what currently exists

```
import database from '@react-native-firebase/database';

database()
  .ref('/users/123')
  .update({
    age: 32,
  })
  .then(() => console.log('Data updated.'));
```





# Firebase with react native

## Removing data

To remove data, you can call the remove method on a reference:

```
await database().ref('/users/123').remove();
```

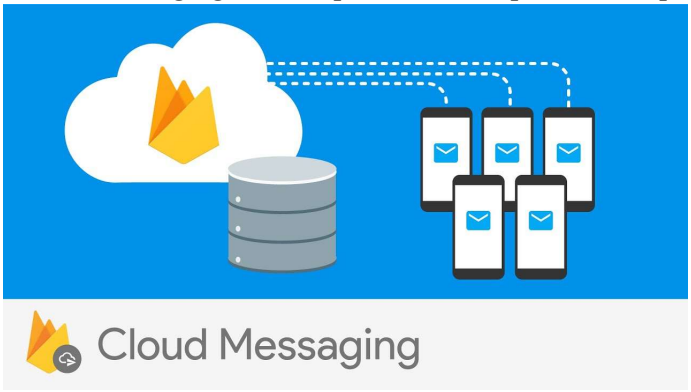
Optionally, you can also set the value of a reference node to null to remove it from the database:

```
await database().ref('/users/123').set(null);
```

# Firebase with react native

## Firebase Cloud Messaging

React Native Firebase provides native integration of Firebase Cloud Messaging (FCM) for both Android & iOS.
FCM is a cost free service, allowing for server-device and device-device communication.
The React Native Firebase Messaging module provides a simple JavaScript API to interact with FCM.






11

# Firebase with react native

## Installation

This module requires that the @react-native-firebase/app module is already setup and installed
# Install & setup the app module
*yarn add @react-native-firebase/app*
*npm install @react-native-firebase/app*

# Install the messaging module
yarn add @react-native-firebase/messaging
*npm install* @react-native-firebase/messaging



12

# Firebase with react native

## Android - Requesting permissions

On Android API level 32 and below, you do not need to request user permission. This method can still be called on Android devices; however, and will always resolve successfully.
For API level 33+ you will need to request the permission manually using either the built-in react-native PermissionsAndroid APIs or a related module such as react-native-permissions

```
import {PermissionsAndroid} from 'react-native';
 PermissionsAndroid.request(PermissionsAndroid.PERMISSIONS.POST_NOTIFICATIONS);
```





# Firebase with react native

## iOS - Requesting permissions

iOS prevents messages containing notification (or 'alert') payloads from being displayed unless you have received explicit permission from the user.

```
import messaging from '@react-native-firebase/messaging';

async function requestUserPermission() {
  const authStatus = await messaging().requestPermission();
  const enabled =
    authStatus === messaging.AuthorizationStatus.AUTHORIZED ||
    authStatus === messaging.AuthorizationStatus.PROVISIONAL;

  if (enabled) {
    console.log('Authorization status:', authStatus);
  }
}
```

# Firebase with react native

## Receiving messages

| State | Description |
|---|---|
| **Foreground** | When the application is open and in view. |
| **Background** | When the application is open, however in the background (minimised). This typically occurs when the user has pressed the "home" button on the device or has switched to another app via the app switcher. |
| **Quit** | When the device is locked or application is not active or running. The user can quit an app by "swiping it away" via the app switcher UI on the device. |



15

# Firebase with react native

## Message handlers

The device state and message contents determines which handler will be called:

| | Foreground | Background | Quit |
|---|---|---|---|
| **Notification** | `onMessage` | `setBackgroundMessageHandler` | `setBackgroundMessageHandler` |
| **Notification + Data** | `onMessage` | `setBackgroundMessageHandler` | `setBackgroundMessageHandler` |
| **Data** | `onMessage` | `setBackgroundMessageHandler` (***see below***) | `setBackgroundMessageHandler` (***see below***) |



16

# Firebase with react native

## Notifications

The device state and message contents can also determine whether a Notification will be displayed:

| | Foreground | Background | Quit |
|---|---|---|---|
| **Notification** | Notification: ❌ | Notification: ✅ | Notification: ✅ |
| **Notification + Data** | Notification: ❌ | Notification: ✅ | Notification: ✅ |
| **Data** | Notification: ❌ | Notification: ❌ | Notification: ❌ |



17

# Firebase with react native

## Foreground state messages

The React Native Alert API could be used to display a new Alert each time a message is delivered

```
import React, { useEffect } from 'react';
import { Alert } from 'react-native';
import messaging from '@react-native-firebase/messaging';

function App() {
  useEffect(() => {
    const unsubscribe = messaging().onMessage(async remoteMessage => {
      Alert.alert('A new FCM message arrived!', JSON.stringify(remoteMessage));
    });

    return unsubscribe;
  }, []);
}
```



18

# Firebase with react native

## Background & Quit state messages

To setup a background handler, call the setBackgroundMessageHandler outside of your application logic as early as possible

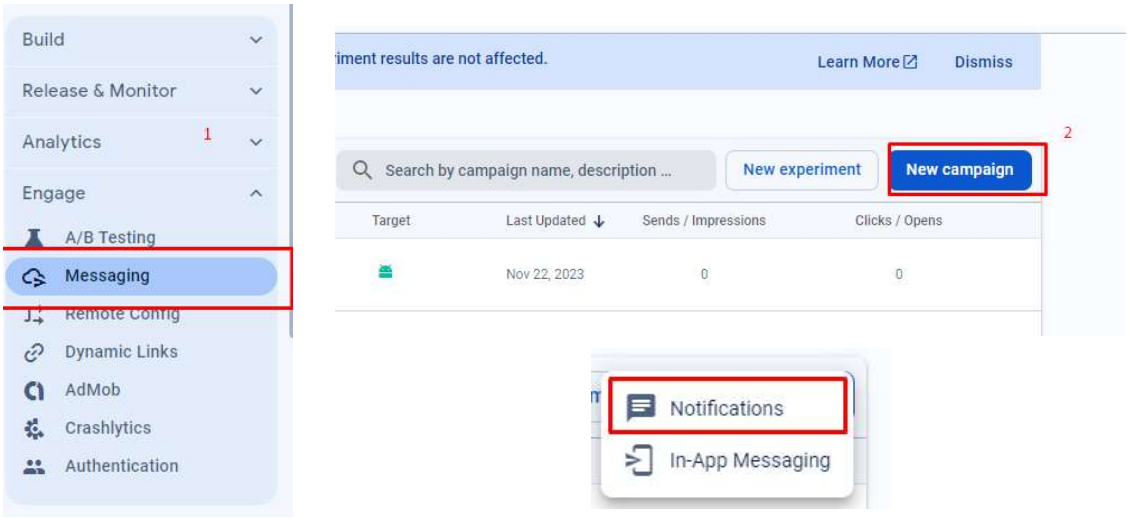
```
// index.js
import { AppRegistry } from 'react-native';
import messaging from '@react-native-firebase/messaging';
import App from './App';

// Register background handler
messaging().setBackgroundMessageHandler(async remoteMessage => {
  console.log('Message handled in the background!', remoteMessage);
});

AppRegistry.registerComponent('app', () => App);
```





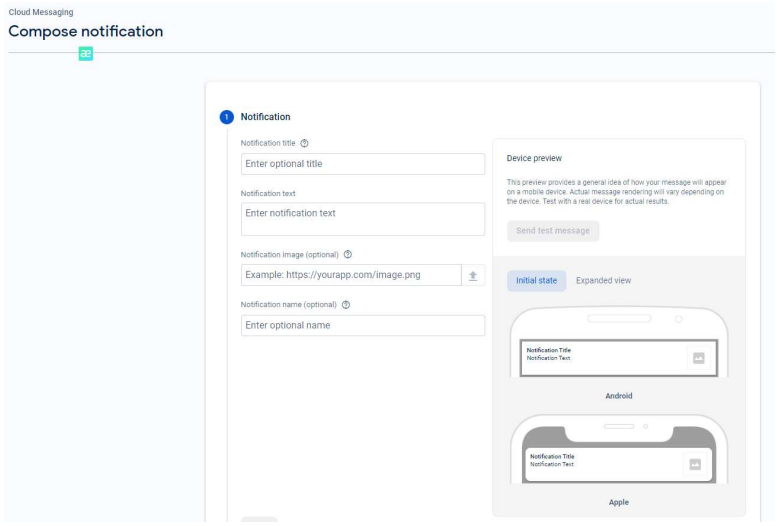
# Firebase with react native

## Send messages using the Firebase console



# Firebase with react native

**Send messages using the Firebase console**






21

# Exercise

Create an app screen to receive notifications sent from firebase cloud message



22

EIU
TRƯỜNG ĐẠI HỌC
QUỐC TẾ
MIỀN ĐÔNG
EASTERN
INTERNATIONAL
UNIVERSITY
Q&A
23

23