# 🔔 **Firebase Push Notifications Setup (React Native)**

## **1. Install Required Packages**

yarn add @react-native-firebase/app @react-native-firebase/messaging

or

npm install @react-native-firebase/app @react-native-firebase/messaging

## **2. Android Setup**

### **a) Update android/build.gradle**

buildscript {

  dependencies {

    classpath 'com.google.gms:google-services:4.3.15' // latest

  }

}

### **b) Update android/app/build.gradle**

At the bottom:

apply plugin: 'com.google.gms.google-services'

Also ensure:

minSdkVersion = 21

### **c) Add Firebase Config**

Place google-services.json into:

android/app/google-services.json

## **3. iOS Setup**

### **a) Add Firebase Config**

Put GoogleService-Info.plist into:

ios/YourApp/GoogleService-Info.plist

### **b) Pod Install**

cd ios

pod install

cd ..

### **c) Enable Push Capability**

1. Open Xcode → select project → Signing & Capabilities.
2. Add **Push Notifications** + **Background Modes → Remote notifications**.

## **4. Request Notification Permission**

Create a helper in notificationService.js:

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

import { PermissionsAndroid, Platform } from 'react-native';

// Request permissions (iOS) or handle runtime permission (Android 13+)

export 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);

  }

  if (Platform.OS === 'android') {

    const granted = await PermissionsAndroid.request(

      PermissionsAndroid.PERMISSIONS.POST\_NOTIFICATIONS

    );

    console.log('Android notification permission:', granted);

  }

}

Call it inside App.js:

import React, { useEffect } from 'react';

import { requestUserPermission } from './notificationService';

export default function App() {

  useEffect(() => {

    requestUserPermission();

  }, []);

  return <></>;

}

## **5. Get FCM Token**

This token uniquely identifies the device → used to send push messages.

async function getFCMToken() {

  const token = await messaging().getToken();

  console.log('FCM Token:', token);

  return token;

}

## **6. Handle Incoming Messages**

### **a) Foreground Notifications**

useEffect(() => {

  const unsubscribe = messaging().onMessage(async remoteMessage => {

    console.log('FCM Message in foreground:', remoteMessage);

    // Show a local notification if you want

  });

  return unsubscribe;

}, []);

## **7. Send Test Notification**

1. Go to **Firebase Console → Cloud Messaging**.
2. Click **Send your first message**.
3. Add a title + body.
4. Target → your app.
5. Click **Send** → your device should receive it if token is registered.

For more query refer to Firebase official documentation:

<https://rnfirebase.io/messaging/usage>