

Firebase Integration for React Native Project

Step 1: Install Firebase Dependencies

```
# Install Firebase core library
npm install @react-native-firebase/app
```

```
# Install Firebase services (choose the ones you need)
npm install @react-native-firebase/auth @react-native-firebase/firestore @react-native-firebase/messaging
```

Step 2. Configure Firebase for Android

1. Register Your App in Firebase Console

- Go to the Firebase Console.
- Create a new project or use an existing one.
- Register your Android app by providing your app's package name (e.g., `com.example.myapp`).

2. Download `google-services.json`

- After registering, download the `google-services.json` file.
- Place the file in the `android/app/` directory of your project.

3. Update `android/build.gradle` Add the Google Services plugin to the buildscript

```
4     dependencies {
5         classpath("com.android.tools.build:gradle")
6         classpath("com.facebook.react:react-native-gradle-plugin")
7         classpath("org.jetbrains.kotlin:kotlin-gradle-plugin")
8         classpath 'com.google.gms:google-services:4.4.2' // Latest version
9     }
```

4. Update `android/app/build.gradle` Apply the Google Services plugin and dependencies:

```
apply plugin: "com.android.application"
apply plugin: "org.jetbrains.kotlin.android"
apply plugin: "com.facebook.react"
apply plugin: 'com.google.gms.google-services'
```

3. Configure Firebase for iOS

1.Register Your App in Firebase Console

- Go to the Firebase Console and select **Add App** → **iOS**.
- Provide your iOS app's **Bundle Identifier** (e.g., `com.example.myapp`).

2.Download GoogleService-Info.plist

- Download the `GoogleService-Info.plist` file.
- Add the file to your Xcode project by dragging it into the `ios/` directory in Xcode.

3.Install CocoaPods Dependencies

Navigate to the `ios/` folder and install dependencies:

```
cd ios
pod install
```

4.Update AppDelegate.m

Configure Firebase in the `AppDelegate.m` file:

```
#import "AppDelegate.h"
#import <Firebase.h>

#import <React/RCTBundleURLProvider.h>

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.moduleName = @"EventPlannerApp";
    // You can add your custom initial props in the dictionary below.
    // They will be passed down to the ViewController used by React Native.
    self.initialProps = @{};

    [FIRApp configure];

    return [super application:application didFinishLaunchingWithOptions:launchOptions];
}
```

Firestore Cloud Messaging Integration for push notification

Step 1: Configure Firestore in Your React Native App

Ensure that your Firestore configuration is complete and that you have integrated Firestore Cloud Messaging into your app. This includes:

1. Adding the Firestore configuration files (google-services.json for Android, GoogleService-Info.plist for iOS).
2. Installing the Firestore SDKs, including @react-native-firebase/app and @react-native-firebase/messaging.

Step 2: Enable Background Notifications

Update your code to handle background and foreground push notifications:

In App.js:

```

useEffect(() => {
  const requestFCMPermissions = async () => {
    const granted = await PermissionsAndroid.request(
      PermissionsAndroid.PERMISSIONS.POST_NOTIFICATIONS
    );

    if (granted !== PermissionsAndroid.RESULTS.GRANTED) {
      Alert.alert('Permission Denied', 'You need to enable notifications.');
```

Step 3: Create a Cloud Function to Schedule Notifications

Firebase Cloud Functions allow you to send notifications programmatically.

Install Firebase CLI and Set Up Functions:

```

npm install -g firebase-tools
firebase login
firebase init functions
```

Create a Scheduled Function:

1. Navigate to the `functions` directory.
2. Install the required dependencies:

```
npm install firebase-admin
```

npm install node-schedule

3 Update index.js or index.ts in the functions directory with the following code:

```
ex.js > ...
const functions = require('firebase-functions');
const admin = require('firebase-admin');
const schedule = require('node-schedule');

admin.initializeApp(); // Initialize Firebase Admin SDK

const sendNotification = async (title, body) => {
  const payload = {
    notification: {
      title,
      body,
    },
    topic: 'all', // Send to all users subscribed to the 'all' topic
  };

  try {
    await admin.messaging().send(payload);
    console.log(`Notification sent: ${title}`);
  } catch (error) {
    console.error('Error sending notification:', error);
  }
};

exports.scheduleNotifications = functions.pubsub.schedule('0 8,12,17 * * *').onRun(async () => {
  const times = [
    { hour: 8, title: 'Good Morning!', body: 'Start your day with some positivity!' },
    { hour: 12, title: 'Lunch Time!', body: 'Take a break and enjoy your lunch!' },
    { hour: 17, title: 'Evening Reminder!', body: 'Wrap up your day with a smile!' },
  ];

  const currentHour = new Date().getHours();
  const notification = times.find((t) => t.hour === currentHour);

  if (notification) {
    await sendNotification(notification.title, notification.body);
  }
});
```

Deploy the Function:

firebase deploy --only functions

Step 4: Test the Notifications

1. Run your app on a real device (push notifications don't work on iOS simulators).
2. Deploy the function.
3. Wait for the scheduled times to verify notifications.