



LẬP TRÌNH ĐA NỀN TẢNG 2

BÀI 8: GIỚI THIỆU VỀ FCM TRONG
FIREBASE VÀ TỰY CHỈNH NOTIFICATION

PHẦN 1: GIỚI THIỆU VỀ FCM TRONG
FIREBASE

- ☐ Giới thiệu về FCM
- ☐ Cài đặt thư viện

□ Firebase Cloud Messaging

Firestore Cloud Messaging (FCM) là một giải pháp nhắn tin đa nền tảng cho phép bạn gửi tin nhắn miễn phí một cách đáng tin cậy.

Sử dụng FCM, bạn có thể thông báo cho ứng dụng khách rằng email mới hoặc dữ liệu khác có sẵn để đồng bộ hóa. Bạn có thể gửi tin nhắn thông báo để thúc đẩy tương tác lại và giữ chân người dùng. Đối với các trường hợp sử dụng như nhắn tin tức thời, một tin nhắn có thể chuyển tải trọng lên đến 4000 byte sang ứng dụng khách.

Tính năng chính:

- ❖ Gửi tin nhắn thông báo được hiển thị cho người dùng của bạn. Hoặc gửi thông điệp dữ liệu và xác định hoàn toàn những gì xảy ra trong mã ứng dụng của bạn.
- ❖ Phân phối tin nhắn đến ứng dụng khách của bạn theo bất kỳ cách nào trong 3 cách—đến các thiết bị đơn lẻ, đến các nhóm thiết bị hoặc đến các thiết bị đã đăng ký chủ đề.
- ❖ Gửi xác nhận, trò chuyện và các tin nhắn khác từ thiết bị tới máy chủ của bạn qua kênh kết nối đáng tin cậy và tiết kiệm pin của FCM.

☐ FCM hoạt động như thế nào?

Việc triển khai FCM bao gồm hai thành phần chính để gửi và nhận:

1. Một môi trường đáng tin cậy như Cloud Functions cho Firebase hoặc một máy chủ ứng dụng để xây dựng, nhắm mục tiêu và gửi tin nhắn.
2. Ứng dụng khách Apple, Android hoặc web (JavaScript) nhận tin nhắn qua dịch vụ truyền tải dành riêng cho nền tảng tương ứng.

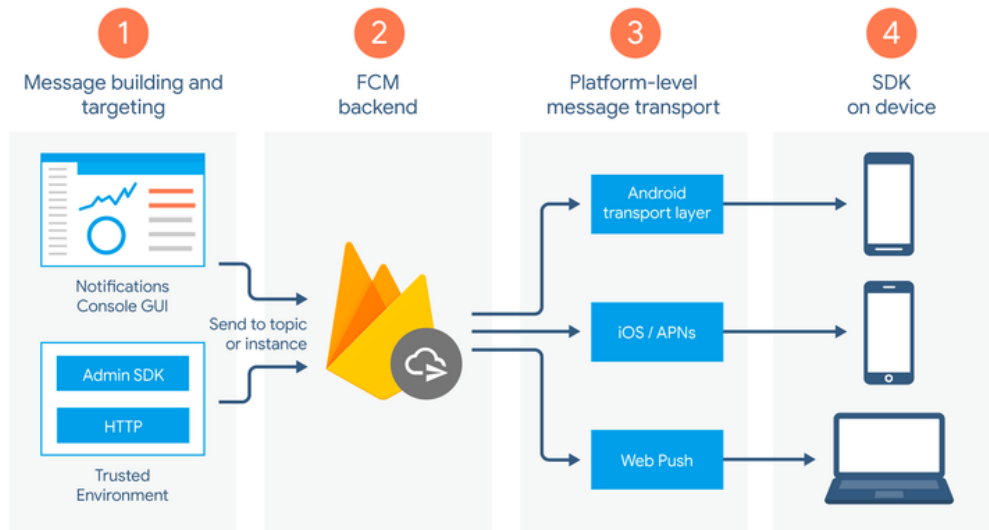
Bạn có thể gửi tin nhắn qua Firebase Admin SDK hoặc FCM server protocol. Bạn có thể gửi message trong Firebase Console để thử nghiệm và gửi tin nhắn tiếp thị hoặc tương tác bằng cách sử dụng nhắm mục tiêu và phân tích tích hợp mạnh mẽ hoặc phân khúc được nhập tùy chỉnh.

- Để xây dựng một ứng dụng nhận được notification từ Firebase cần thực hiện 3 bước chính sau:
- ❖ Bước 1: Thiết lập Firebase và FCM trên ứng dụng của bạn theo hướng dẫn thiết lập cho nền tảng của bạn.
 - ❖ Bước 2: Thêm xử lý tin nhắn, logic đăng ký chủ đề hoặc các tính năng tùy chọn khác vào ứng dụng của bạn. Trong quá trình phát triển, bạn có thể dễ dàng gửi tin nhắn thử nghiệm từ Firebase Console.
 - ❖ Bước 3: Quyết định xem bạn muốn sử dụng SDK quản trị Firebase hay giao thức máy chủ để tạo logic gửi—logic để xác thực, xây dựng yêu cầu gửi, xử lý phản hồi, v.v. Sau đó xây dựng logic trong môi trường đăng tin cậy của bạn.

□ Tổng quan kiến trúc FCM

FCM dựa trên tập hợp các thành phần sau đây để xây dựng, vận chuyển và nhận thông điệp:

1. Công cụ để soạn hoặc xây dựng yêu cầu tin nhắn. Trình soạn thảo thông báo cung cấp tùy chọn dựa trên GUI để tạo yêu cầu thông báo. Để tự động hóa hoàn toàn và hỗ trợ cho tất cả các loại tin nhắn, bạn phải tạo yêu cầu tin nhắn trong môi trường máy chủ đáng tin cậy hỗ trợ SDK quản trị Firebase hoặc giao thức máy chủ FCM. Môi trường này có thể là Cloud Functions cho Firebase, App Engine hoặc máy chủ ứng dụng của riêng bạn.



2. FCM backend, (trong số các chức năng khác) chấp nhận các yêu cầu tin nhắn, thực hiện phân xuất tin nhắn thông qua các chủ đề và tạo siêu dữ liệu tin nhắn như ID tin nhắn.

3. Lớp truyền tải cấp nền tảng, định tuyến tin nhắn đến thiết bị được nhắm mục tiêu, xử lý việc gửi tin nhắn và áp dụng cấu hình dành riêng cho nền tảng khi thích hợp. Lớp vận chuyển này bao gồm:

- Lớp truyền tải Android (ATL) dành cho thiết bị Android có dịch vụ của Google Play.
- Dịch vụ Thông báo đẩy của Apple (APN) cho các thiết bị Apple

- Giao thức push web cho các ứng dụng web

4. FCM SDK trên thiết bị của người dùng, nơi thông báo được hiển thị hoặc tin nhắn được xử lý theo trạng thái foreground/background của ứng dụng và bất kỳ logic ứng dụng có liên quan nào.

Lifecycle flow

- Đăng ký thiết bị để nhận tin nhắn từ FCM. Một phiên bản của ứng dụng khách đăng ký nhận tin nhắn, nhận token thông báo đăng ký xác định duy nhất phiên bản ứng dụng.

- Gửi và nhận tin nhắn xuôi dòng
 - Tin nhắn được soạn, trong Notifications composer hoặc môi trường đăng tin cây và yêu cầu tin nhắn được gửi đến FCM backend.
 - FCM backend nhận yêu cầu tin nhắn, tạo ID tin nhắn và siêu dữ liệu khác và gửi nó đến lớp truyền tải cụ thể của nền tảng.
 - Khi thiết bị trực tuyến, tin nhắn được gửi qua lớp truyền tải dành riêng cho nền tảng đến thiết bị.
 - Trên thiết bị, ứng dụng khách sẽ nhận được tin nhắn hoặc thông báo.

- ☐ Firebase Cloud Messaging (FCM) cung cấp một loạt các tùy chọn và khả năng nhắn tin. Thông tin trong trang này nhằm giúp bạn hiểu các loại thông điệp FCM khác nhau và những gì bạn có thể làm với chúng.
- ☐ Message types

Với FCM, bạn có thể gửi hai loại tin nhắn cho user:

- Notification messages, đôi khi được coi là 'tin nhắn hiển thị'. Chúng được xử lý bởi FCM SDK tự động.
- Data messages, được xử lý bởi ứng dụng khách.

Notification messages chứa một tập hợp các key hiển thị cho người dùng được xác định trước. Ngược lại, data messages chỉ chứa các cặp key-value tùy chỉnh do người dùng xác định. Notification messages có thể chứa tải trọng dữ liệu tùy chọn. Tải trọng tối đa cho cả hai loại tin nhắn là 4000 byte, ngoại trừ khi gửi tin nhắn từ bảng điều khiển Firebase, thực thi giới hạn 1000 ký tự.

Sử dụng notification messages khi bạn muốn SDK FCM xử lý việc tự động hiển thị thông báo khi ứng dụng của bạn đang chạy background. Sử dụng data messages khi bạn muốn xử lý tin nhắn bằng mã ứng dụng khách của riêng mình.

Sử dụng notification messages khi bạn muốn SDK FCM xử lý việc tự động hiển thị thông báo khi ứng dụng của bạn đang chạy background. Sử dụng data messages khi bạn muốn xử lý tin nhắn bằng mã ứng dụng khách của riêng mình.

Notification messages type

- Khi một notification messages được gửi đi, user sẽ nhận được một object data như dưới đây:

```
{
  "message":{
    "token":"bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
    "notification":{
      "title":"Portugal vs. Denmark",
      "body":"great match!"
    }
  }
}
```

Notification messages được gửi đến khay thông báo khi ứng dụng ở chế độ background. Đối với các ứng dụng ở foreground, tin nhắn được xử lý bằng callback function.

- Đặt khóa thích hợp với các cặp key-value trị tùy chỉnh của bạn để gửi data payload đến ứng dụng khách.

Ví dụ: đây là thông báo có định dạng JSON trong cùng một ứng dụng IM như trên, trong đó thông tin được đóng gói trong key dữ liệu chung và ứng dụng khách dự kiến sẽ diễn giải nội dung:

```
{
  "message":{
    "token":"bk3RNwTe3H0:CI2k_HHwgIpoDKCIZvvDMExUdFQ3P1...",
    "data":{
      "Nick" : "Mario",
      "body" : "great match!",
      "Room" : "PortugalVSDenmark"
    }
  }
}
```

Ví dụ trên cho thấy việc sử dụng trường dữ liệu top-level hoặc phổ biến, được user diễn giải trên tất cả các nền tảng nhận được tin nhắn. Trên mỗi nền tảng, client app nhận data payload trong hàm callback.

- ☐ Bạn cũng có thể vừa nhận được notification messages và data messages cùng lúc

- ☐ Cài đặt thư viện @react-native-firebase/messaging, đảm bảo rằng bạn đã cài đặt thư viện @react-native-firebase/app và setup firebase trước đó rồi

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

- ☐ Ở bài này các bạn đã học được các khái niệm về FCM, cách đó hoạt động và cách sử dụng như thế nào cho hợp lý.
- ☐ Ở bài sau, các bạn sẽ được hướng dẫn cách nhận notification khi app ở foreground và background, cách custom notification cho ứng dụng.

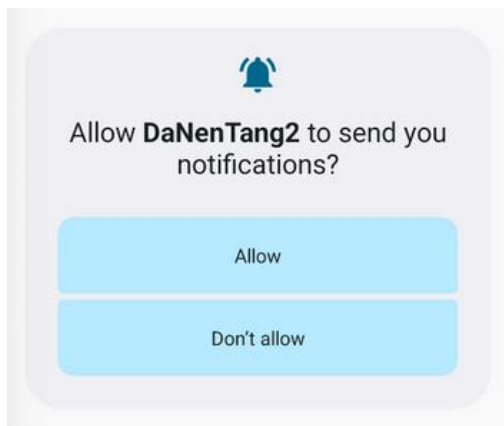


LẬP TRÌNH ĐA NỀN TẢNG 2

BÀI 8: GIỚI THIỆU VỀ FCM TRONG
FIREBASE VÀ TÙY CHỈNH NOTIFICATION
PHẦN 2: GỬI NOTIFICATION, TÙY CHỈNH
NOTIFICATION CHO ỨNG DỤNG

- ☐ Ở bài này, các bạn sẽ học được cách gửi notification đến ứng dụng ở chế độ foreground và background
- ☐ Tùy chỉnh notification hiển thị trên ứng dụng

- ❑ Để giao tiếp với message FCM, chúng ta sẽ sử dụng khá nhiều hàm, thế nên chúng ta sẽ gom chúng lại và xử lý chúng trong hook của chúng ta có tên useFCM
- ❑ Đầu tiên, viết hàm để yêu cầu người dùng bật setting notification của ứng dụng ngay khi ứng dụng được mở



- ❑ Lưu ý, ở Android 13 giới thiệu quyền thời gian chạy mới để hiển thị thông báo. Điều này ảnh hưởng đến tất cả các ứng dụng chạy trên Android 13 trở lên sử dụng thông báo FCM.
- ❑ Bạn cần thêm permission mới này vào AndroidManifest.xml

```
<uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
```


- ❑ Để hiện popup yêu cầu permission thông báo ứng dụng như hình trên bạn cần cài đặt thêm thư viện react-native-permissions

npm i react-native-permissions

- Viết hàm `requestUserPermission` để hiển thị popup yêu cầu người dùng cho phép thông báo

```

DaNenTang2 - useFCM.ts
1  import {requestNotifications} from 'react-native-permissions'

```

```

DaNenTang2 - useFCM.ts
1  const requestUserPermission = async () => {
2    try {
3      requestNotifications(['alert', 'sound']).then(({status, settings}) => {
4        // ... Làm gì đó ở đây
5      });
6    } catch (error) {
7      console.log('requestUserPermission', error);
8    }
9  };

```

- Gọi hàm `requestUserPermission` trong hook `useFCM` tại `App.tsx`, mỗi khi người dùng được mở lên sẽ gọi hàm này

```
DaNenTang2 - App.tsx

1  function App(): JSX.Element {
2    const {requestUserPermission} = useFCM();
3
4    useEffect(() => {
5      requestUserPermission();
6    }, []);
```

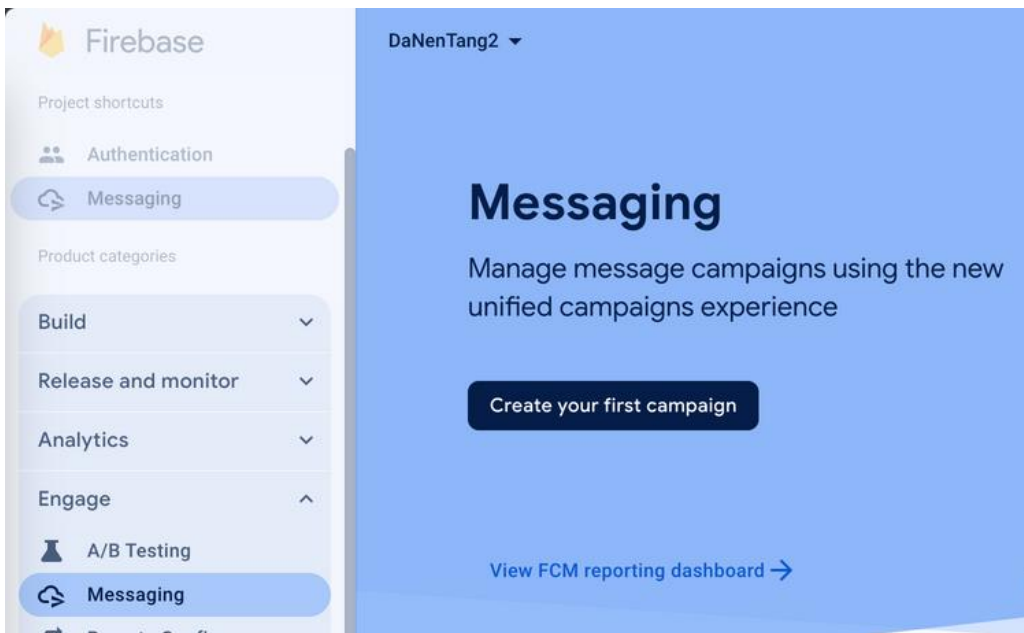
- ❑ Để nhận message từ Firebase sử dụng hàm `messaging().onMessage()`

```
DaNenTang2 - useFCM.ts
1  import messaging from '@react-native-firebase/messaging';
2
```

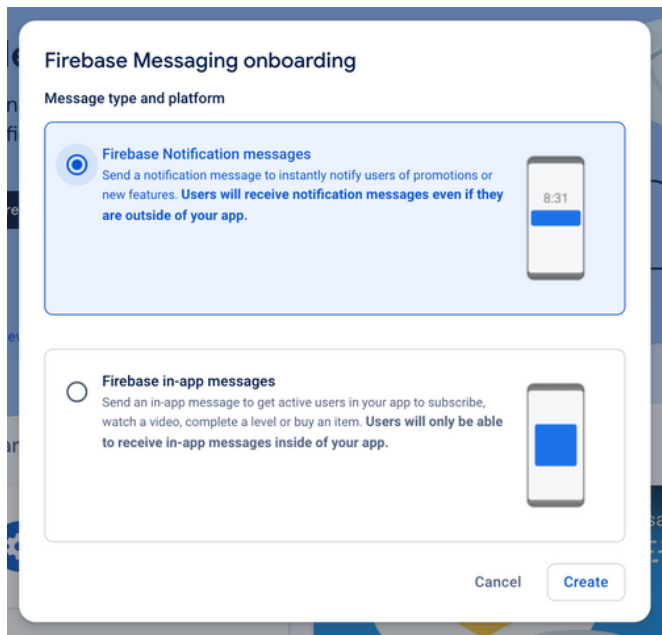
```
DaNenTang2 - useFCM.ts

1  useEffect(() => {
2    const unsubscribe = messaging().onMessage(
3      async (remoteMessage: unknown) => {
4        console.log('onMessage ', remoteMessage);
5      },
6    );
7
8    return () => {
9      unsubscribe();
10   };
11 }, []);
```

- ❑ Sau đó bạn trở lại Firebase console, đã tạo từ slide 7. Vào mục Message chọn Create your first campaign



- ☐ Popup hiện ra, chọn Firebase Notification messages. Sau đó nhấn Create



- ☐ Nhập nội dung Notification messages type theo ví dụ ở đây:

1 Notification

Notification title ?

Notification text

Notification image (optional) ?



Notification name (optional) ?

Device preview

This preview provides a general idea of how your message will appear on a mobile device. Actual message rendering will vary depending on the device. Test with a real device for actual results.

[Send test message](#)

Initial state

Expanded view



- ☐ Chọn app bạn muốn gửi thông báo tới, ở bản demo này tôi chọn com.danentang2. Sau đó nhấn Next


2 Target

User segment

Topic

Target user if...

App

 com.danentang2

▼

Target another app

Fewer than 100 users match this condition ?

Next

- Nhấn nút Review, sau đó có một Popup hiện lên để xác nhận gửi notification. Bạn nhấn nút Publish trên popup đó

The screenshot shows a multi-step notification configuration interface. A vertical progress bar on the left indicates the current step is 3, 'Scheduling', which is highlighted with a grey background. Below the header, the text 'Send to eligible users' is followed by a text input field containing the word 'Now'. A blue 'Next' button is positioned below the input field. The progress bar continues down to step 4, 'Conversion events (optional)', and then to step 5, 'Additional options (optional)'. At the bottom right of the interface, there are two buttons: 'Save as draft' and 'Review'.

3 Scheduling

Send to eligible users

Now

Next

4 Conversion events (optional)

5 Additional options (optional)

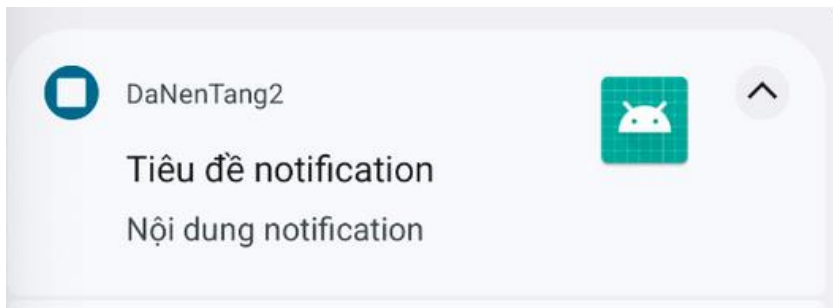
Save as draft Review

❖ Lưu ý: thời gian gửi notification này có thể bị delay lên đến 15 phút.

□ Đây là response hàm onMessage nhận được notification từ firebase console

```
onMessage {"collapseKey": "com.danentang2", "data": {}, "from": "182591633460", "messageId":  
"0:1701524662173735%a9cab6dfa9cab6df", "notification": {"android": {}, "body": "Nội dung notification",  
"title": "Tiêu đề notification"}, "sentTime": 1701524610802, "ttl": 2419200}
```

- Tiếp theo, bây giờ chúng ta sẽ muốn thông báo hiển thị nhận được từ FCM lên ứng dụng giống như hình dưới đây, bạn cần cài đặt thêm thư viện react-native-push-notification



Cài đặt thư viện

```
npm install --save react-native-push-notification
```

- ❑ Để push notification hoạt động, bạn cần tạo `createChannel` trên Android

```
DaNenTang2 - useFCM.ts

1  PushNotification?.createChannel(
2    {
3      channelId: 'notification-channel-id',
4      channelName: 'notification-channel',
5      soundName: 'default',
6    },
7    created => {
8      console.log('notification-channel-id: ', created);
9    },
10 );
```

- Tiếp theo bạn sử dụng hàm configure để bắt sự kiện từ prop onNotification người dùng nhấn lên notification trên thanh thông báo.

```
DaNenTang2 - useFCM.ts

1  PushNotification.configure({
2    onNotification(notification) {
3      console.log('On click notificatoin ', notification);
4    },
5    permissions: {
6      alert: true,
7      badge: true,
8      sound: true,
9    },
10  });
```

- Trở lại hàm `onMessage` khi nhận được thông báo, bạn gọi hàm `localNotification` để push thông dựa trên dữ liệu thông báo nhận được hiện lên cho người dùng thấy

```
DaNenTang2 - useFCM.ts
1  const unsubscribe = messaging().onMessage(async (remoteMessage: any) => {
2    console.log('onMessage ', remoteMessage);
3
4    PushNotification.localNotification({
5      channelId: 'notification-channel-id',
6      title:
7        remoteMessage.notification?.title || remoteMessage.data.title || '',
8      bigText:
9        remoteMessage.notification?.body || remoteMessage.data.body || '', //content for Android
10     message:
11       remoteMessage.notification?.body || remoteMessage.data.body || '', //content for Ios
12     ...remoteMessage,
13   });
14 }
```


- Ngoài ra, bạn có thể làm thêm các nội dung khác lên thông báo như thay đổi âm thanh thông báo, thay đổi icon, thêm nội dung,... để tìm hiểu thêm chi tiết, các bạn truy cập trang <https://www.npmjs.com/package/react-native-push-notification>

- ☐ Ở chương này, các bạn đã được học cách gửi notification trên Firebase console, cách sử dụng hàm onMessage để nhận thông báo từ firebase, hiển thị notification cho người dùng thấy và thao tác trên nó.

- ☐ Giới thiệu về FCM
- ☐ Cài đặt thư viện
- ☐ Ở bài này, các bạn sẽ học được cách gửi notification đến ứng dụng ở chế độ foreground và background
- ☐ Tùy chỉnh notification hiển thị trên ứng dụng

Kết thúc