



EBook Gratis

APRENDIZAJE firebase-cloud- messaging

Free unaffiliated eBook created from
Stack Overflow contributors.

#firebase-

cloud-

messaging

Tabla de contenido

Acerca de.....	1
Capítulo 1: Comenzando con la mensajería en la nube de base de fuego.....	2
Observaciones.....	2
Examples.....	3
Instalación o configuración.....	3
Capítulo 2: Firebase Cloud Messaging.....	5
Introducción.....	5
Examples.....	5
Envío de mensajes en sentido descendente a través de cURL.....	5
Enviando mensajes en sentido descendente usando el cartero.....	5
Capítulo 3: Manejo de notificaciones de mensajes.....	9
Examples.....	9
Manejo de mensajes en Android.....	9
Manejo de mensajes en iOS.....	9
Manejo de mensajes con aplicación en segundo plano o muerto.....	9
Creditos.....	11

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [firebase-cloud-messaging](#)

It is an unofficial and free firebase-cloud-messaging ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official firebase-cloud-messaging.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Comenzando con la mensajería en la nube de base de fuego

Observaciones

Una pregunta común notoria es "cómo enviar notificaciones de dispositivo a dispositivo", lamentablemente la respuesta es: no se puede. FCM debe activarse para enviar notificaciones push. Eso se puede hacer de 3 maneras diferentes:

1. Directamente en la consola web de Firebase.
2. Configuración de un escucha de funciones de Firebase y luego activación de FCM
3. Un servidor solicita a FCM que envíe una notificación de inserción

Una notificación de inserción es una carga útil de información que se envía desde FCM. Hay 3 tipos de notificaciones push: `notification`, `data`, `notification and data`. Esta información puede ser representada como un JSON:

```
{
  "to" : "APA91bHun4MxP5egoKMwt2KZFBaFUH-1RYqx...",
  "notification" : {
    "body" : "great match!",
    "title" : "Portugal vs. Denmark",
    "icon" : "myicon"
  },
  "data" : {
    "Nick" : "Mario",
    "Room" : "PortugalVSDenmark"
  }
}
```

El ejemplo anterior es para el tercer tipo, `notification` y `data` combinados. Eso es lo que se le pedirá a FCM que envíe.

1. La consola puede enviar `notification` y `notification` con `data` pero nunca solo `data`
2. Funciones y cualquier servidor puede enviar los 3 tipos.

La importancia del tipo de `notification` es que permite que las aplicaciones reciban *empujes* predeterminados, lo que permite a otros equipos, como el marketing, aumentar el crecimiento de las aplicaciones simplemente usando la consola web sin necesidad de codificación adicional además de agregar la biblioteca al proyecto.

No confunda la notificación de inserción, el tipo de `notification` y la notificación visual, esto último corresponde a una clase de Android (comúnmente `NotificationCompat`).

El comportamiento del empuje es diferente según el tipo y si la aplicación está en primer plano o en segundo plano. No en primer plano significa, minimizado o cerrado.

1. `notification` activará una notificación visual predeterminada **si la aplicación no está en**

primer plano , esta notificación se puede personalizar en el manifiesto, consulte la [documentación](#) . Si la aplicación está en primer plano, tenemos que personalizar el comportamiento dentro del método `onMessageReceived` .

2. `data` comportamiento del tipo de `data` siempre debe ser personalizado.
3. La `notification` y los `data` combinados si la aplicación **no está en primer plano** activarán la notificación visual predeterminada y `data` carga útil de `data` estará disponible cuando el usuario haga clic. Dado que la actividad del iniciador se activa cuando se hace clic en la notificación visual, debe literalmente `getIntent().getStringExtra("yourKey");` en esa actividad para obtener los datos. Si la aplicación está activa (en primer plano), debe personalizar el comportamiento dentro del método `onMessageReceived` y acceder a la carga útil de `data` inmediatamente.

Para obtener la carga útil de la información, debe hacerlo dentro del método `onMessageReceived` , el único argumento disponible es el mensaje:

1. Para obtener la `notification` que tiene a `remoteMessage.getNotification()` , puede obtener el cuerpo o el título con los métodos correspondientes
2. Para obtener los `data` que tiene a `remoteMessage.getData().get("yourKey")` .

Es una buena idea agregar cada verificación que *no sea nula* , habrá varios tipos de notificaciones que llegarán a las aplicaciones avanzadas. Una buena estrategia es verificar si cada uno, la `notification` y los `data` no son nulos. Una estrategia útil consecuente será utilizar siempre una clave de `type` en las notificaciones de `data` para realizar un control de flujo.

Para enviar `data` desde la consola web de Firebase, se deben abrir las opciones avanzadas.

Las claves de `notification` son limitadas, y se indican en la documentación. **Los valores en cualquier tipo pueden ser solo String** .

Si tiene problemas para encontrar documentación en Firebase, vaya a la parte inferior de la página y cambie el idioma a "Inglés"; las documentaciones son *más delgadas* en otros idiomas.

Examples

Instalación o configuración

Firebase Cloud Messaging es el servicio Firebase que maneja las notificaciones push. Puedes agregar este servicio en cualquier cliente: web, Android o IOS. El funcionamiento específico de cada uno debe leerse en la [documentación](#) .

Para agregar FCM en cualquier tipo de proyecto, siempre está agregando una biblioteca .

Teniendo en cuenta el soporte especial para Android es digno de tomar algunas líneas para ello. Cree un nuevo proyecto utilizando Android Studio, en el menú vaya a Herramientas / Firebase, se activará el asistente de Firebase. Seleccione "Cloud Messaging" y siga los pasos uno y dos.

1. Si su proyecto agrega previamente otro servicio Firebase, entonces el paso uno se marcará como completado, de lo contrario, debe hacerlo. El primer paso le permite crear un proyecto

en Firebase o crear uno nuevo. Este paso descargará un archivo `google-service.json` que tiene la configuración para conectarse con el proyecto Firebase. Este archivo está dentro de la carpeta "app".

2. Este paso agrega la biblioteca de servicios de Google y la biblioteca de Firebase al gradle, y también hará alguna configuración adicional en esos archivos.

Esta es la base para agregar FCM en un proyecto. A partir de este momento, el cliente ya puede recibir notificaciones push de FCM que contienen una carga útil de "notificación" siempre que la aplicación no esté en primer plano (más detalles en las observaciones).

Para personalizar aún más el comportamiento de FCM en el cliente, necesitamos agregar 2 servicios, esto está bien [documentado](#) en el sitio oficial. Una vez más vamos a tener alguna consideración para Android:

1. Cree una clase que amplíe `FirebaseMessagingService` y anule el método `onMessageReceived`
2. Cree una clase que amplíe `FirebaseInstanceIdService` y anule el método `onTokenRefresh`
3. Registre ambas clases en el manifiesto, haga esto dentro de la etiqueta de la `application`
`</intent-filter>` `</intent-filter>`

Puede obtener la carga útil de `notification` y la carga útil de `data` dentro del método `onMessageReceived` utilizando el único argumento allí. El método `onTokenRefresh` se llama cuando el token FCM es asignado por FCM. Un token de FCM es una identificación única para la instalación de la aplicación y el dispositivo, y se puede utilizar como una dirección del dispositivo para enviar notificaciones push directamente.

Lea las observaciones para obtener más información sobre los tipos de notificación y el comportamiento asociado.

Lea [Comenzando con la mensajería en la nube de base de fuego en línea](#):

<https://riptutorial.com/es/firebase-cloud-messaging/topic/8197/comenzando-con-la-mensajeria-en-la-nube-de-base-de-fuego>

Capítulo 2: Firebase Cloud Messaging

Introducción

Firebase Cloud Messaging (FCM) es una solución de mensajería multiplataforma que le permite entregar mensajes de manera confiable sin costo alguno. Con FCM, puede notificar a una aplicación cliente que hay un nuevo correo electrónico u otros datos disponibles para sincronizar. Puede enviar mensajes de notificación para impulsar la re inserción y retención de usuarios. Para casos de uso como la mensajería instantánea, un mensaje puede transferir una carga útil de hasta 4 KB a una aplicación cliente. Documentación oficial: <https://firebase.google.com/docs/cloud-messaging/>

Examples

Envío de mensajes en sentido descendente a través de cURL

Puede probar el envío de mensajes utilizando la API REST de FCM enviando una solicitud a través de cURL.

```
curl --header "Authorization: key=<API_KEY>" \
  --header Content-Type:"application/json" \
  https://fcm.googleapis.com/fcm/send \
  -d "{\"registration_ids\": [\"ABC\"]}"
```

Sintaxis recuperada desde [aquí](#) .

El `API_KEY` indicado anteriormente se refiere a la clave del servidor que se puede ver en la [pestaña de mensajería en la nube de su consola Firebase](#) .

La parte donde:

```
"{\"registration_ids\": [\"ABC\"]}"
```

Es, puede ser reemplazado con su propia carga útil. Consulte la [documentación del protocolo HTTP de FCM](#) para obtener más detalles.

Enviando mensajes en sentido descendente usando el cartero

Para hacer esto en [Postman](#) , simplemente tienes que configurar lo siguiente:

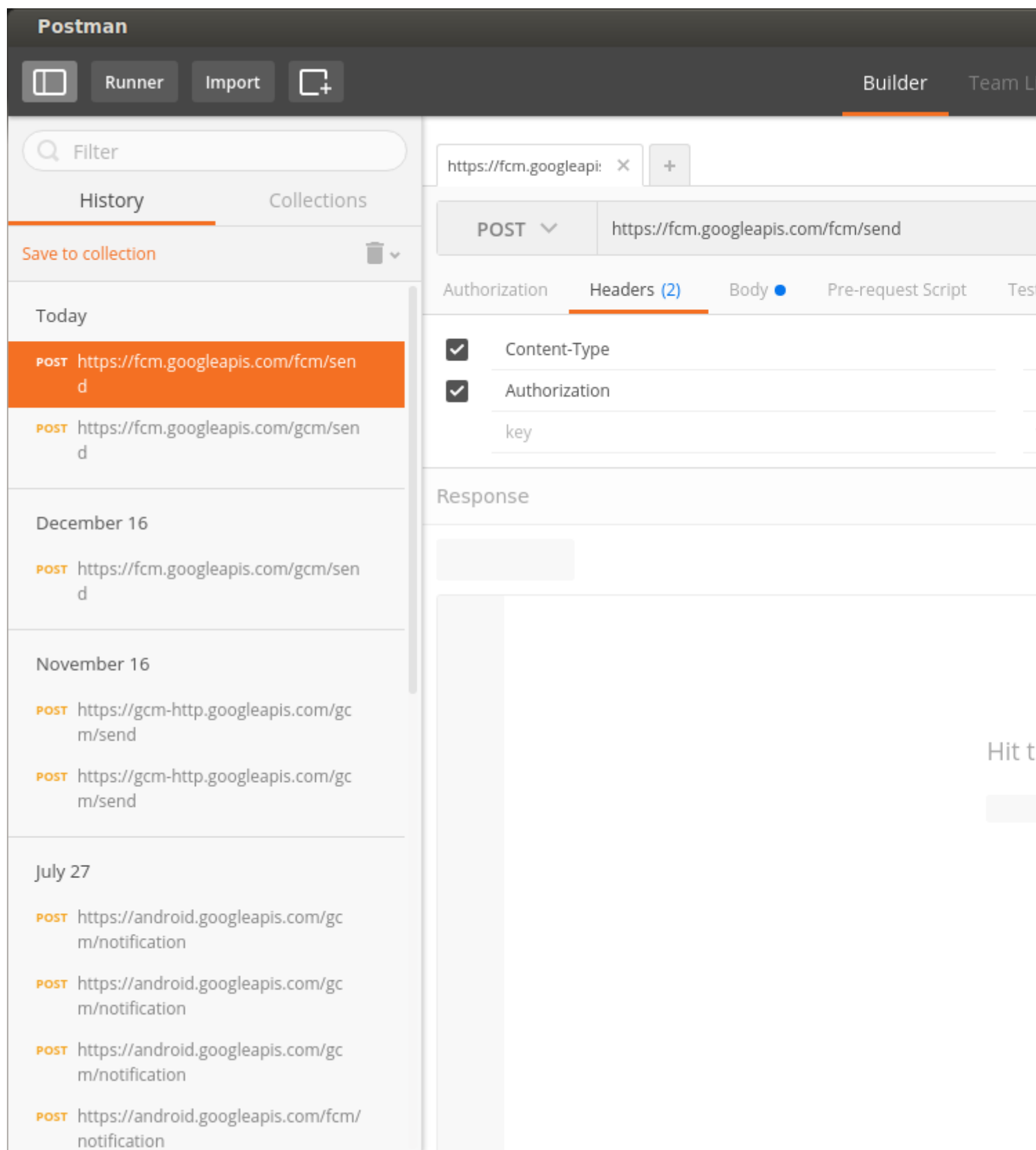
1. Establecer el tipo de solicitud en `POST`
2. En los *encabezados* , establece lo siguiente:
 - Tipo de contenido = aplicación / json
 - Autorización = <Su clave de servidor FCM> (Consulte la [pestaña de mensajería en la nube de su consola Firebase](#))
3. Establezca los parámetros de carga útil en el *Cuerpo* (*en este ejemplo, usamos la opción*

sin procesar , vea la captura de pantalla (2))

4. Envíe la solicitud a <https://fcm.googleapis.com/fcm/send>

Capturas de pantalla:

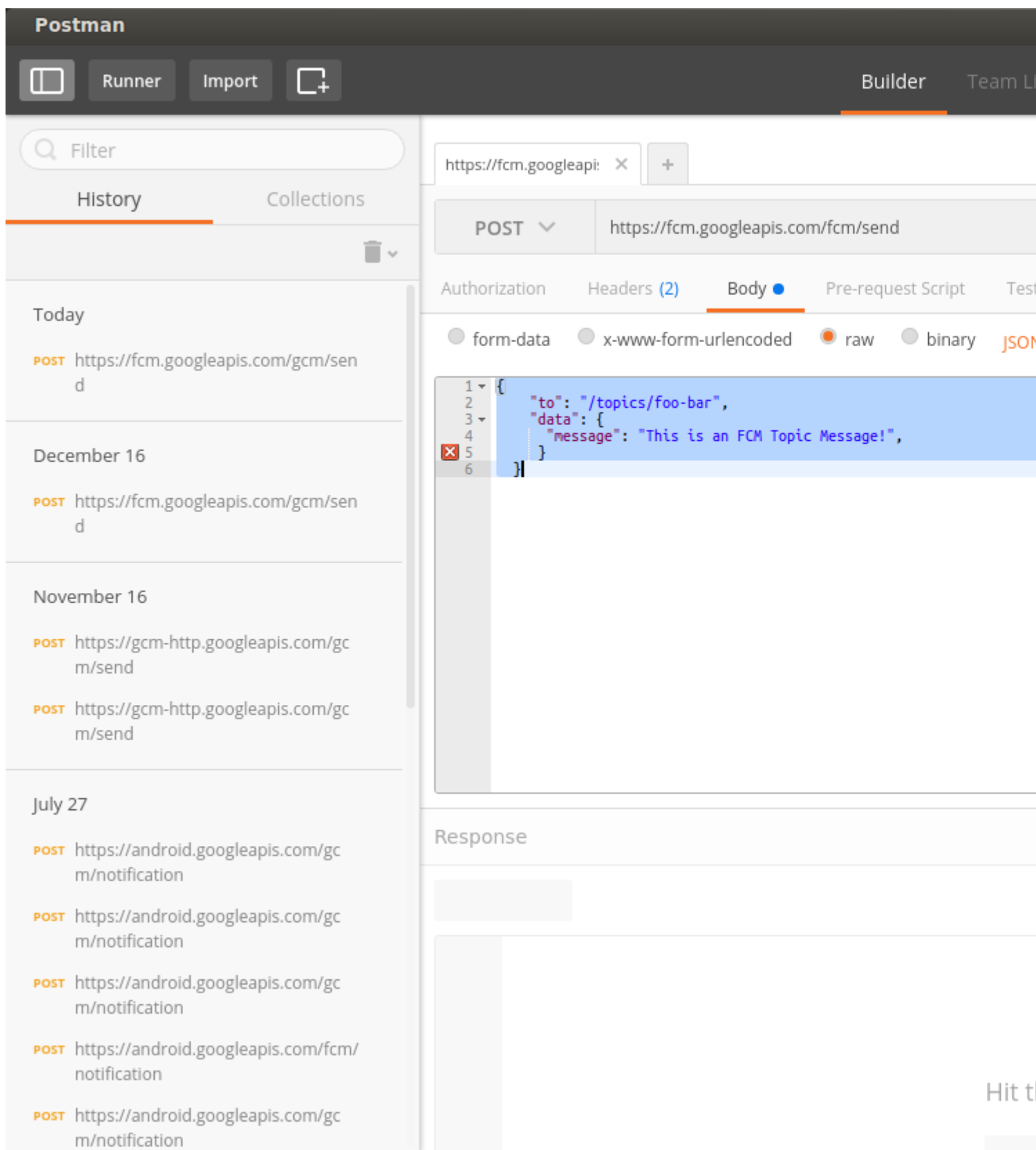
(1)



Nota : Mantenga siempre su clave de servidor en secreto. Aquí solo se ve una parte de mi llave,

así que debería estar bien.

(2)



(3)

The screenshot shows the Postman interface. The top bar includes 'Postman', 'Runner', 'Import', and 'Builder' tabs. The left sidebar shows a 'History' tab with a search filter and a list of requests categorized by date: Today, December 16, November 16, and July 27. The main area shows a selected POST request to 'https://fcm.googleapis.com/fcm/send'. The 'Body' tab is active, displaying the response in 'Pretty' JSON format:

```
{
  "message_id": 7401061585236820685
}
```

Observe que la solicitud fue un éxito con el `message_id` en la respuesta.

Lea [Firebase Cloud Messaging en línea](https://riptutorial.com/es/firebase-cloud-messaging/topic/8242/firebase-cloud-messaging): <https://riptutorial.com/es/firebase-cloud-messaging/topic/8242/firebase-cloud-messaging>

Capítulo 3: Manejo de notificaciones de mensajes

Examples

Manejo de mensajes en Android

```
@Override
public void onMessageReceived(RemoteMessage remoteMessage) {
    // ...

    // TODO(developer): Handle FCM messages here.
    Log.d(TAG, "From: " + remoteMessage.getFrom());

    // Check if message contains a data payload.
    if (remoteMessage.getData().size() > 0) {
        Log.d(TAG, "Message data payload: " + remoteMessage.getData());
    }

    // Check if message contains a notification payload.
    if (remoteMessage.getNotification() != null) {
        Log.d(TAG, "Message Notification Body: " + remoteMessage.getNotification().getBody());
    }

    // Also if you intend on generating your own notifications as a result of a received FCM
    // message, here is where that should be initiated. See sendNotification method below.
}
```

Manejo de mensajes en iOS

```
- (void)application:(UIApplication *)application didReceiveRemoteNotification:(NSDictionary *)userInfo
    fetchCompletionHandler:(void (^)(UIBackgroundFetchResult))completionHandler {
    // If you are receiving a notification message while your app is in the background,
    // this callback will not be fired till the user taps on the notification launching the
    application.
    // TODO: Handle data of notification

    // Print message ID.
    NSLog(@"Message ID: %@", userInfo[@"gcm.message_id"]);

    // Print full message.
    NSLog(@"%@", userInfo);
}
```

Manejo de mensajes con aplicación en segundo plano o muerto.

Firestore maneja las notificaciones de manera diferente cuando la aplicación está en segundo plano (proceso muerto) y cuando está en primer plano (activa).

Cuando la aplicación está en segundo plano, los mensajes de notificación se muestran

en la bandeja del sistema y no se llama a `onMessageReceived`. Para los mensajes de notificación con una carga útil de datos, el mensaje de notificación se muestra en la bandeja del sistema, y los datos que se incluyeron con el mensaje de notificación se pueden recuperar de la intención iniciada cuando el usuario toca la notificación. [1]

Si la aplicación está en segundo plano, el servicio activará la notificación de forma predeterminada con el `title` y el `body` de la notificación y, como se mencionó, el método `onMessageReceived` no se activará. En su lugar, el clic abrirá la `activity` del `Manifest.xml` marcado con:

```
<intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
```

Desde este punto en adelante, puede obtener sus `data` dentro de la `intent` esta actividad:

```
if (getIntent() != null && getIntent().getExtras() != null) {
    String customString = (String) getIntent().getExtras().get("myStringData");
    Integer customInteger = (Integer) getIntent().getExtras().get("myIntData");
}
```

El color de fondo de los íconos y los íconos en este caso se puede hacer desde `Manifest.xml` [2]:

```
<meta-data
    android:name="com.google.firebase.messaging.default_notification_icon"
    android:resource="@drawable/your_drawable_icon" />

<meta-data
    android:name="com.google.firebase.messaging.default_notification_color"
    android:resource="@color/your_color" />
```

fuelle 1: [manejo de fondo FCM](#)

fuelle 2: [repositorio github FCM](#)

Lea Manejo de notificaciones de mensajes en línea: <https://riptutorial.com/es/firebase-cloud-messaging/topic/8888/manejo-de-notificaciones-de-mensajes>

Creditos

S. No	Capítulos	Contributors
1	Comenzando con la mensajería en la nube de base de fuego	Community , cutiko
2	Firestore Cloud Messaging	AL. , Rowayda Khayri
3	Manejo de notificaciones de mensajes	Eric Gilmore , menilv , ThunderStruct