



**EBook Gratis**

# APRENDIZAJE google-cloud- messaging

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#google-  
cloud-

messaging

# Tabla de contenido

<b>Acerca de</b> .....	<b>1</b>
<b>Capítulo 1: Empezando con google-cloud-messaging</b> .....	<b>2</b>
Observaciones.....	2
Examples.....	4
Envía mensajes desde la nube.....	4
Manejo de mensajes downstream en Android.....	7
Manejo de mensajes posteriores en iOS.....	8
<b>Capítulo 2: Diferencias entre el envío a dispositivos Android y iOS</b> .....	<b>9</b>
Examples.....	9
Hacer que el dispositivo reciba notificación incluso cuando está durmiendo.....	9
<b>Creditos</b> .....	<b>10</b>

---

## Acerca de

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

It is an unofficial and free google-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 google-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](mailto:info@zzzprojects.com)

---

# Capítulo 1: Empezando con google-cloud-messaging

## Observaciones

### Google Cloud Messaging: descripción general

Google Cloud Messaging (GCM) es un servicio gratuito que permite a los desarrolladores enviar mensajes entre servidores y aplicaciones cliente. Esto incluye mensajes en sentido descendente de servidores a aplicaciones cliente, y mensajes en sentido ascendente de aplicaciones cliente a servidores.

Por ejemplo, un mensaje descendente ligero podría informar a una aplicación cliente que hay nuevos datos que se pueden recuperar del servidor, como en el caso de una notificación de "nuevo correo electrónico". Para casos de uso como la mensajería instantánea, un mensaje GCM puede transferir hasta 4kb de carga útil a la aplicación cliente. El servicio GCM maneja todos los aspectos de la cola de mensajes y la entrega hacia y desde la aplicación cliente de destino.

---

### Descripción arquitectónica

Una implementación de GCM incluye un servidor de conexión de Google, un servidor de aplicaciones en su entorno que interactúa con el servidor de conexión a través del protocolo HTTP o XMPP y una aplicación cliente.

Aquí es cómo interactúan estos componentes:

- **Los servidores de conexión de Google GCM** aceptan mensajes de su servidor de aplicaciones y los envían a una aplicación cliente. El servidor de conexión [XMPP](#) también puede aceptar mensajes enviados en sentido ascendente desde la aplicación cliente y reenviarlos a su servidor de aplicaciones. Para obtener más información, consulte [Acerca del servidor de conexión GCM](#).
- En su **Servidor de aplicaciones**, usted implementa el [protocolo HTTP](#) y / o [XMPP](#) para comunicarse con los servidores de conexión GCM. Los servidores de aplicaciones envían mensajes posteriores a un servidor de conexión GCM; el servidor de conexión pone en cola y almacena el mensaje, y luego lo envía a la aplicación cliente. Si implementa XMPP, su servidor de aplicaciones puede recibir mensajes enviados desde la aplicación cliente.
- La **aplicación de cliente** es una aplicación de cliente habilitada para GCM. Para recibir y enviar mensajes de GCM, esta aplicación debe registrarse con GCM y obtener un identificador único llamado token de registro. Para obtener más información sobre cómo implementar la aplicación cliente, consulte la documentación de su plataforma.

---

### Conceptos clave

A continuación se resumen los términos y conceptos clave involucrados en GCM. Se divide en

estas categorías:

- **Componentes** : las entidades que desempeñan un papel principal en GCM.
- **Credenciales** : los identificadores y tokens que se utilizan en GCM para garantizar que todas las partes se hayan autenticado y que el mensaje se envíe al lugar correcto.

Componentes y credenciales de GCM.

## Componentes

- **Servidores de conexión GCM: servidores de Google** que participan en el envío de mensajes entre el servidor de aplicaciones y la aplicación cliente.
- **Aplicación de cliente** : una aplicación de cliente habilitada para GCM que se comunica con su servidor de aplicaciones.
- **Servidor de aplicaciones** : un servidor de aplicaciones que escribe como parte de la implementación de GCM. El servidor de aplicaciones envía datos a una aplicación cliente a través del servidor de conexión GCM. Si su servidor de aplicaciones implementa el protocolo XMPP, también puede recibir mensajes enviados en sentido ascendente desde las aplicaciones cliente.

## Cartas credenciales

- **identificación del remitente**  
Un valor numérico único creado al configurar su proyecto API. La identificación del remitente se usa en el [proceso de registro](#) para identificar un servidor de aplicaciones que tiene permitido enviar mensajes a la aplicación cliente.
- **Clave del servidor**  
Una clave guardada en el servidor de aplicaciones que le otorga acceso autorizado a los servicios de Google. En HTTP, la clave del servidor se incluye en el encabezado de las solicitudes POST que envían mensajes. En XMPP, la clave del servidor se usa en la solicitud de autenticación SASL PLAIN como contraseña para autenticar la conexión. No incluya la clave del servidor en ningún lugar de su código de cliente. Obtienes la clave del servidor cuando creas tu proyecto API.
- **ID de aplicación**  
La aplicación cliente que se está registrando para recibir mensajes. Cómo se implementa esto depende de la plataforma:
  - **Android** : use el nombre del paquete desde el manifiesto de la aplicación.
  - **iOS** : utilizar el identificador de paquete de la aplicación.
  - **Chrome** : usa el nombre de la extensión de Chrome.
- **Token de Registro**  
Una ID emitida por los servidores de conexión GCM a la aplicación cliente que le permite recibir mensajes. Tenga en cuenta que los tokens de registro deben mantenerse en secreto.

---

## Flujo del ciclo de vida

- **Regístrese para habilitar GCM** . Una instancia de una aplicación cliente se registra para recibir mensajes. Para obtener más información, consulte [Registro de aplicaciones cliente](#) .
- **Enviar y recibir mensajes de bajada** .
  - Enviar un mensaje. El servidor de aplicaciones envía mensajes a la aplicación cliente:
    1. El servidor de aplicaciones [envía un mensaje](#) a los servidores de conexión GCM.
    2. El servidor de conexión GCM pone en cola y almacena el mensaje si el dispositivo está fuera de línea.
    3. Cuando el dispositivo está en línea, el servidor de conexión GCM envía el mensaje al dispositivo.
    4. En el dispositivo, la aplicación cliente recibe el mensaje de acuerdo con la implementación específica de la plataforma. Consulte la documentación específica de su plataforma para obtener más información.
  - Recibe un mensaje. Una aplicación cliente recibe un mensaje de un servidor de conexión GCM. Consulte la documentación específica de su plataforma para obtener detalles sobre cómo una aplicación cliente en ese entorno procesa los mensajes que recibe.
- **Enviar y recibir mensajes en sentido ascendente** . Esta función solo está disponible si está utilizando el [servidor de conexión XMPP](#) .
  - Enviar un mensaje. Una aplicación cliente envía mensajes al servidor de aplicaciones:
    1. En el dispositivo, la aplicación cliente envía mensajes al servidor de conexión XMPP. Consulte la documentación específica de su plataforma para obtener detalles sobre cómo una aplicación cliente puede enviar un mensaje a través de XMPP.
    2. El servidor de conexión XMPP pone en cola y almacena el mensaje si el servidor está desconectado.
    3. Cuando se vuelve a conectar el servidor de aplicaciones, el servidor de conexión XMPP envía el mensaje al servidor de aplicaciones.
  - Recibe un mensaje. Un servidor de aplicaciones recibe un mensaje del servidor de conexión XMPP y luego hace lo siguiente:
    1. Analiza el encabezado del mensaje para verificar la información del remitente de la aplicación cliente.
    2. Envía "ack" al servidor de conexión XMPP para acusar recibo del mensaje.
    3. Opcionalmente analiza la carga útil del mensaje, tal como lo define la aplicación cliente.

---

La documentación oficial de referencia se puede encontrar [aquí](#) .

## Examples

### Envía mensajes desde la nube

Envía un mensaje usando el protocolo del servidor de conexión GCM HTTP:

```
https://gcm-http.googleapis.com/gcm/send
Content-Type:application/json
Authorization:key=AIzaSyZ-1u...0GBYzPu7Udno5aA
```

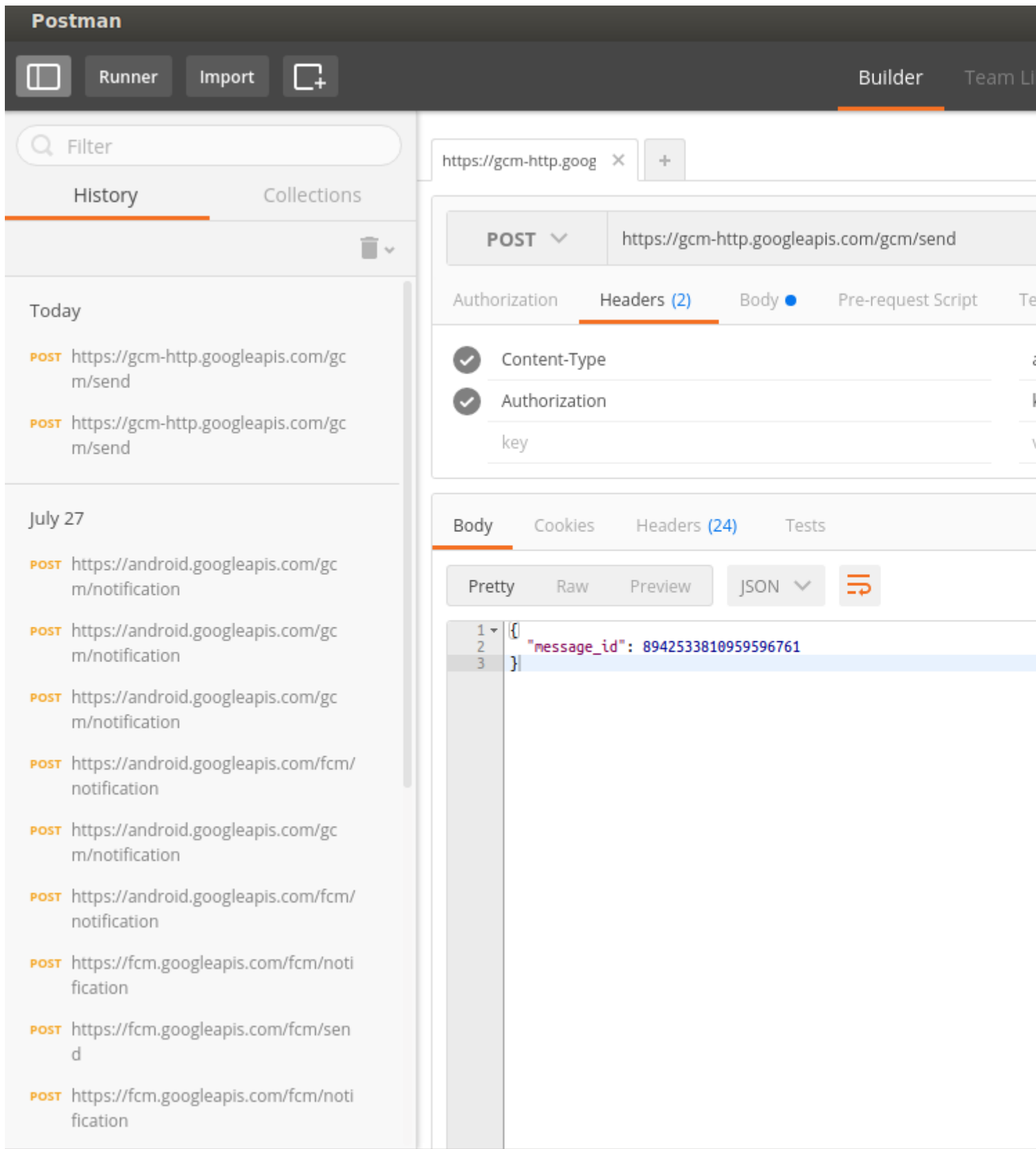
```
{
  "to": "/topics/foo-bar",
  "data": {
    "message": "This is a GCM Topic Message!",
  }
}
```

Para hacer esto en [Postman](#) , simplemente debe configurar lo siguiente ( *algunos detalles son como se mencionó anteriormente* ):

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 GCM>
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://gcm-http.googleapis.com/gcm/send>

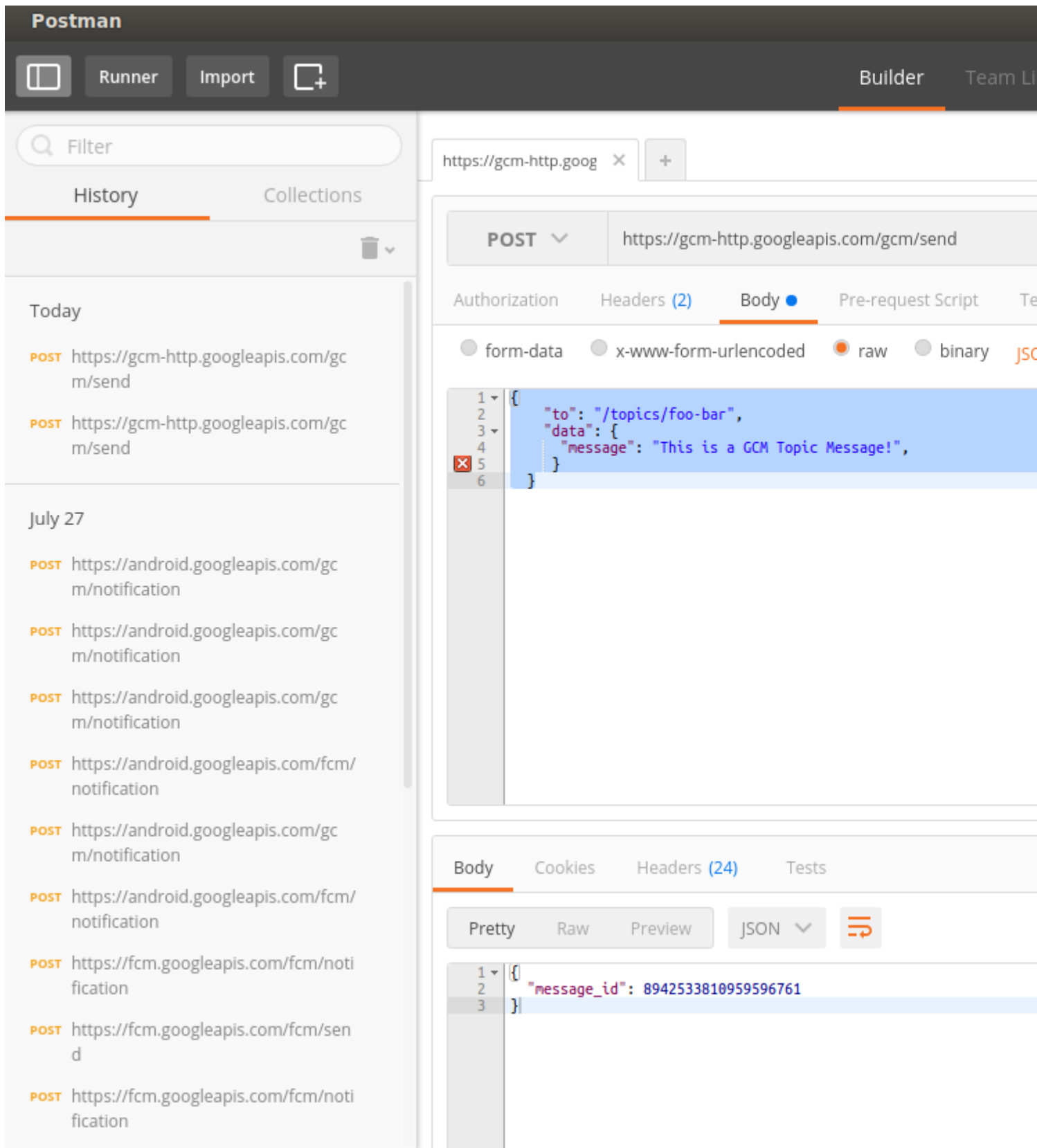
Capturas de pantalla:

(1)



(2)





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

PD: Mantengo visible la clave de servidor de muestra para que otros puedan probarla incluso si aún no han creado un proyecto. **PERO, tenga en cuenta que la clave del servidor debe mantenerse siempre en secreto.**

## Manejo de mensajes downstream en Android

Implementar `onMessageReceived` que detectará la notificación enviada desde el servidor GCM.

```
@Override
public void onMessageReceived(String from, Bundle data) {
    String message = data.getString("message");
    Log.d(TAG, "From: " + from);
    Log.d(TAG, "Message: " + message);
    // Handle received message here.
}
```

## Manejo de mensajes posteriores en iOS

Para recibir la notificación, implemente la

`application:didReceiveRemoteNotification:fetchCompletionHandler:` (o `application:didReceiveRemoteNotification:` para iOS <8.0), y llame a `GCMService:appDidReceiveMessage:message` para reconocer la recepción del mensaje a GCM.

```
- (void)application:(UIApplication *)application
    didReceiveRemoteNotification:(NSDictionary *)userInfo {
    NSLog(@"Notification received: %@", userInfo);
    // This works only if the app started the GCM service
    [[GCMService sharedInstance] appDidReceiveMessage:userInfo];
    // Handle the received message
    // ...
}

- (void)application:(UIApplication *)application
    didReceiveRemoteNotification:(NSDictionary *)userInfo
    fetchCompletionHandler:(void (^)(UIBackgroundFetchResult))handler {
    NSLog(@"Notification received: %@", userInfo);
    // This works only if the app started the GCM service
    [[GCMService sharedInstance] appDidReceiveMessage:userInfo];
    // Handle the received message
    // Invoke the completion handler passing the appropriate UIBackgroundFetchResult value
    // ...
}
```

Lea Empezando con google-cloud-messaging en línea: <https://riptutorial.com/es/google-cloud-messaging/topic/5811/empezando-con-google-cloud-messaging>

---

# Capítulo 2: Diferencias entre el envío a dispositivos Android y iOS.

## Examples

Hacer que el dispositivo reciba notificación incluso cuando está durmiendo.

Al enviar una notificación a un dispositivo iOS, debe establecer la `priority: "high"` para que se active. De lo contrario, la notificación no se recibirá mientras el teléfono esté en reposo.

Establece la prioridad del mensaje. Los valores válidos son "normal" y "alto". En iOS, estos corresponden a las prioridades 5 y 10 de APN.

Por defecto, los mensajes se envían con prioridad normal. La prioridad normal optimiza el consumo de la batería de la aplicación cliente y debe usarse a menos que se requiera una entrega inmediata. Para mensajes con prioridad normal, la aplicación puede recibir el mensaje con un retraso no especificado.

Cuando se envía un mensaje con prioridad alta, se envía inmediatamente y la aplicación puede activar un dispositivo inactivo y abrir una conexión de red a su servidor.

--- [Referencia del servidor FCM](#)

Lea [Diferencias entre el envío a dispositivos Android y iOS](#). en línea:

<https://riptutorial.com/es/google-cloud-messaging/topic/6492/diferencias-entre-el-envio-a-dispositivos-android-y-ios->

---

# Creditos

S. No	Capítulos	Contributors
1	Empezando con google-cloud-messaging	<a href="#">AL.</a> , <a href="#">Community</a>
2	Diferencias entre el envío a dispositivos Android y iOS.	<a href="#">Niels Abildgaard</a>