Firebase Cloud Messaging Notification

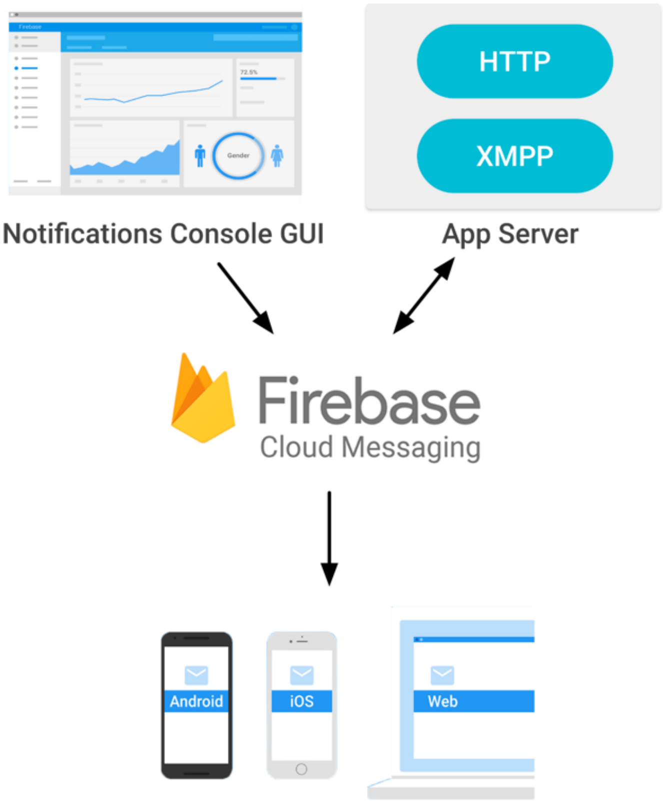
Introduction

Cloud Messaging: Send data from your server to your users' devices, and receive messages from devices on the same connection.

Google has introduced Firebase Cloud Messaging (FCM) instead of Google Cloud Messaging (GCM)

> FCM is the new version of GCM. It inherits the reliable and scalable GCM infrastructure, plus new features! See the FAQ to learn more. If you are integrating messaging in a new app, start with FCM. GCM users are strongly recommended to upgrade to FCM, in order to benefit from new FCM features today and in the future.

FCM is a cross-platform messaging solution that lets you reliably deliver messages at **no cost (really?)**. Using FCM, you can notify a client app that new email or other data is available to sync. You can send notification messages to drive user reengagement and retention. For use cases such as instant messaging, a message can transfer a payload of up to 4KB to a client app.



Key capabilities

1	Send notification messages or data messages	Send notification messages that are displayed to your user. Or send data messages and determine completely what happens in your application code. See Message types.

2	Versatile message targeting	Distribute messages to your client app in any of three ways — to single devices, to groups of devices, or to devices subscribed to topics.
3	Send messages from client apps	Send acknowledgments, chats, and other messages from devices back to your server over FCM’s reliable and battery-efficient connection channel.

FCM Message Types

- notification messages (2KB)
- data messages (4KB)

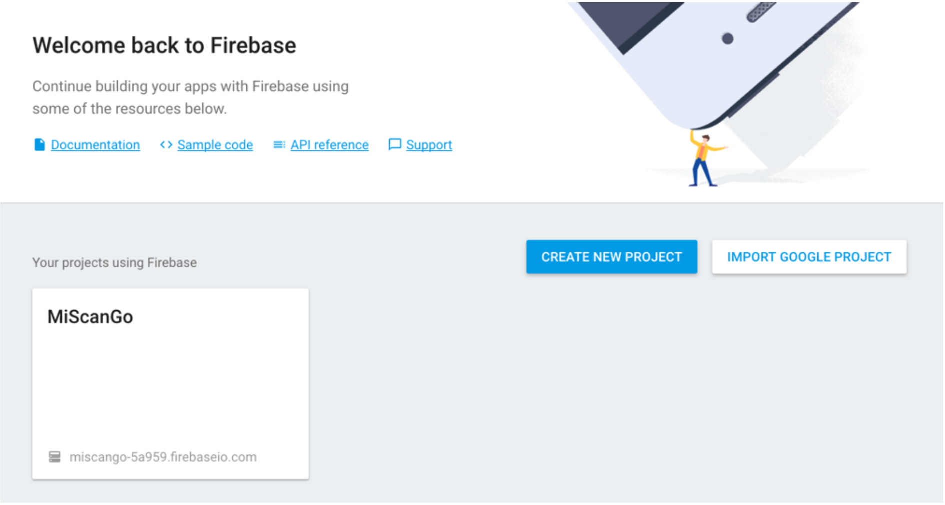
FCM Sever can send two types of message together.

Implementation path

1	Set up the FCM SDK	Set up Firebase and FCM on your app according the setup instructions for your platform.
2	Develop your client app	Add message handling, topic subscription logic, or other optional features to your client app. During the development, you can easily send test messages from the Notifications console.
3	Develop your app server	Decide which server protocol(s) you want to use to interact with FCM, and add logic to authenticate, build send requests, handle response, and so on. Note that if you want to use upstream messaging from your client applications, you must use XMPP.

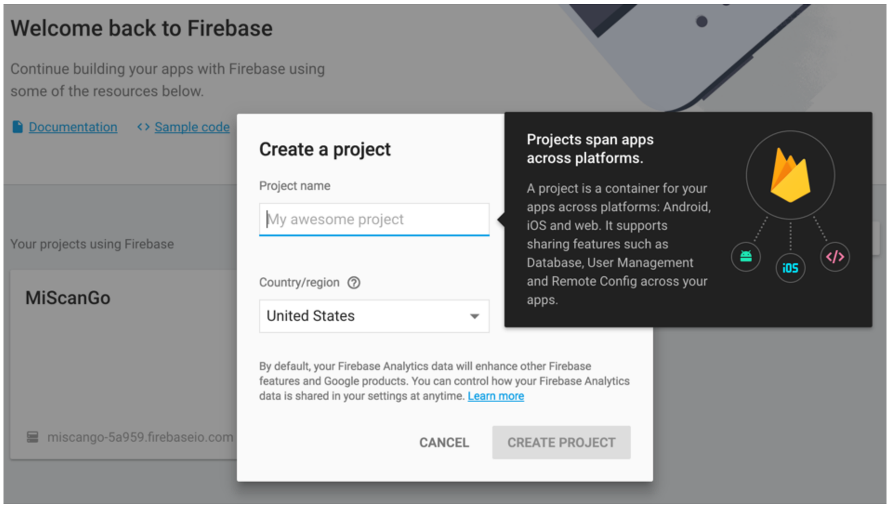
Create FCM Project

1. Register a Google account
2. Create a new project on firebase console



Welcome back to Firebase

Continue building your apps with Firebase using some of the resources below.



Welcome back to Firebase

Continue building your apps with Firebase using some of the resources below.

Create a project

Project name: My awesome project

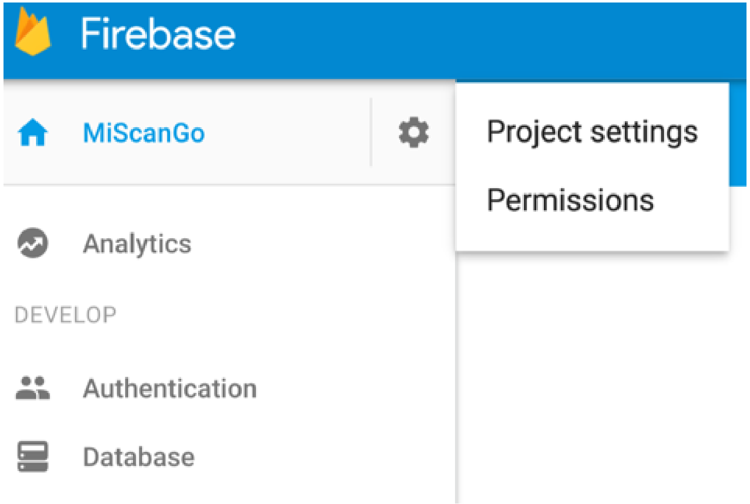
Country/region: United States

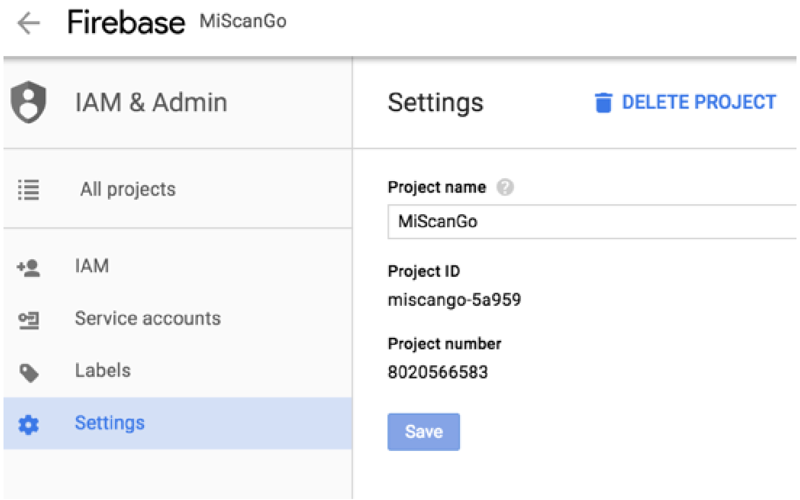
By default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your settings at anytime. Learn more

Projects span apps across platforms.

A project is a container for your apps across platforms: Android, iOS and web. It supports sharing features such as Database, User Management and Remote Config across your apps.

3. Project configuration in Project Setting





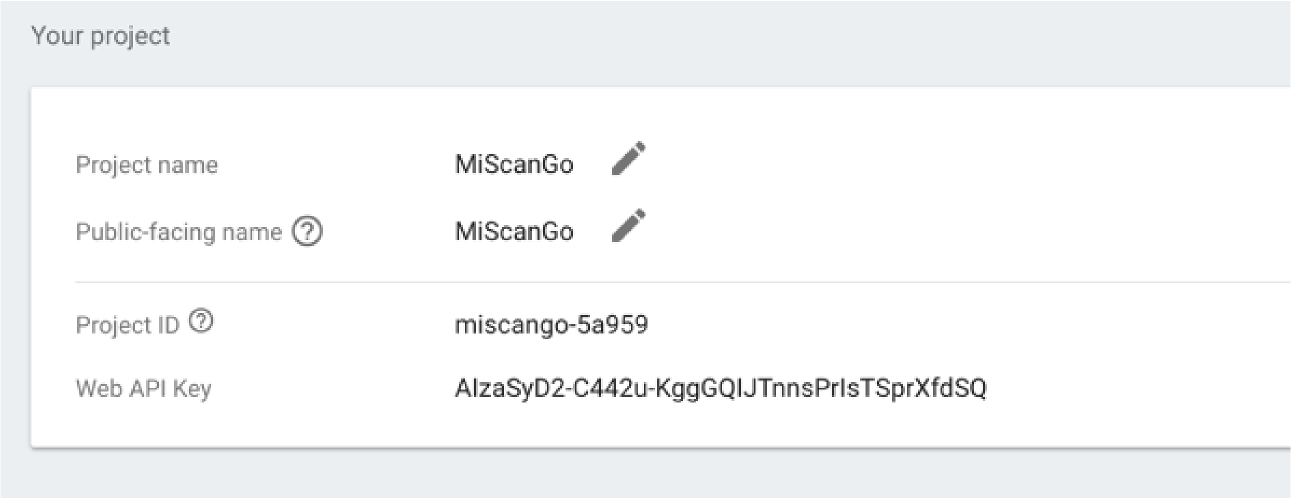
Settings

Project name: MiScanGo

Project ID: miscango-5a959

Project number: 8020566583

4. Require **Sender ID** and **Server Key** on server side



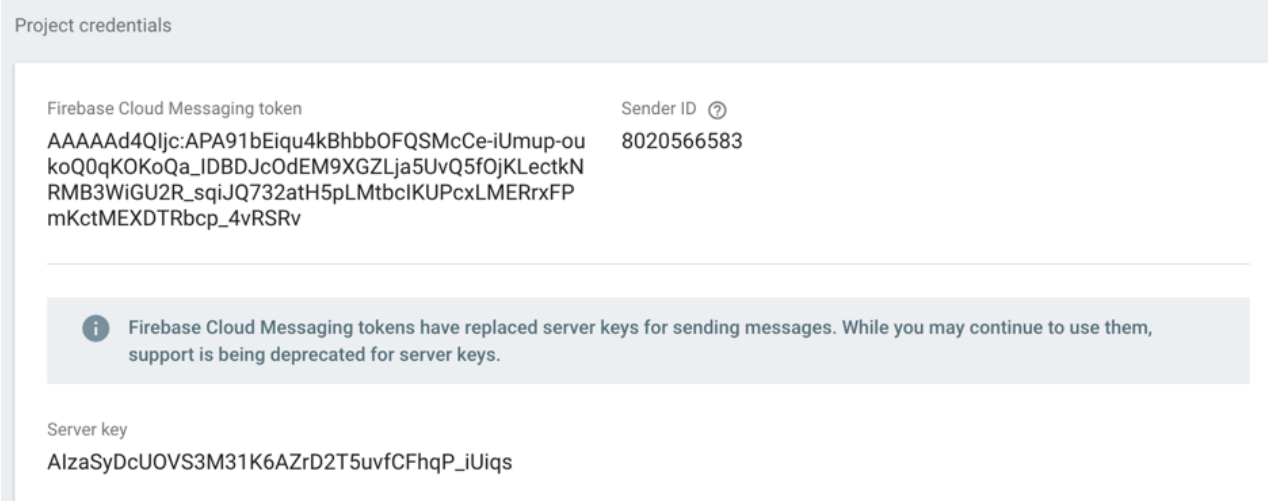
Your project

Project name MiScanGo

Public-facing name MiScanGo

Project ID miscango-5a959

Web API Key AIzaSyD2-C442u-KggGQIJTnnsPrIsTSprXfdSQ



Project credentials

Firebase Cloud Messaging token
AAAAd4QIjc:APA91bEiqu4kBhbbOFQSMcCe-iUmup-oukoQ0qKOKoQa_IDBDJcOdEM9XGZLja5UvQ5fOjKLectkNRMB3WiGU2R_sqiJQ732atH5pLMtbcIKUPcxLMERrxFPmKctMEXDTRbcp_4vRSRv

Sender ID
8020566583

Firebase Cloud Messaging tokens have replaced server keys for sending messages. While you may continue to use them, support is being deprecated for server keys.

Server key
AIzaSyDcUOVS3M31K6AZrD2T5uvfCFhqP_iUiqs

Server Side Implementation

Server Coding

1. Maven Dependency

```
<dependency>
  <groupId>com.google.gcm</groupId>
  <artifactId>gcm-server</artifactId>
  <version>1.0.0</version>
</dependency>
```

2. Notification Service

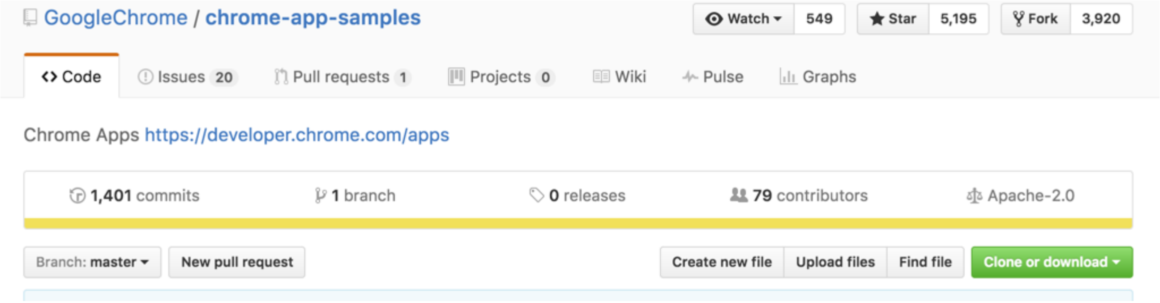
```
Result result = null; //single user
// MulticastResult multicastResult = null; //multiple users
Sender sender = new Sender(Constants.FCM_SERVER_KEY); // server key in
FCM project setting
Message msg = new Message.Builder()
 .collapseKey("collapseKey")
    .timeToLive(Constants.TIME_TO_LIVE)
    .delayWhileIdle(true)
    .addData("title", title)
    .addData("message", message)
 .build();
result = sender.send(msg, regIds.get(0), Constants.INT_RETRY);
// multicastResult = sender.send(msg, regIds, Constants.INT_RETRY);
```

For multicast, the upper limit is 1000.

Server API Test

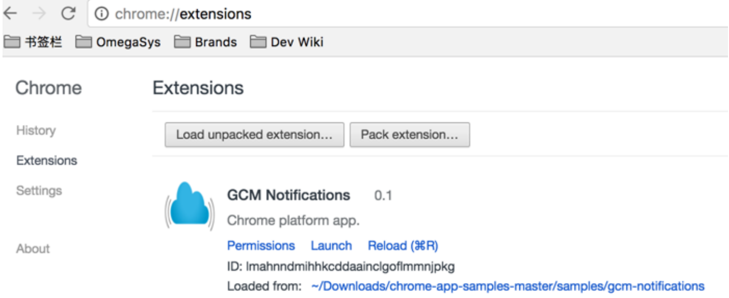
There might be other better ways, following is what I found for now.

1. Download google chrome app samples



Chrome Apps https://developer.chrome.com/apps

2. Load unpacked extension in chrom://extensions, find **GCM Notifications**



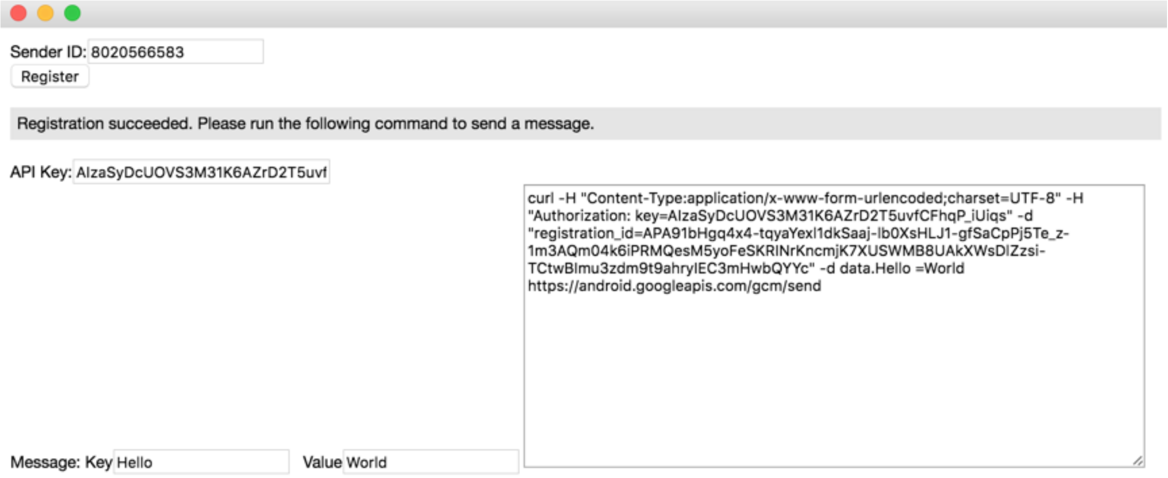
GCM Notifications 0.1

Chrome platform app.

ID: lmahnndmihhkcddaainclgoflmmnjpkg

Loaded from: ~/Downloads/chrome-app-samples-master/samples/gcm-notifications

3. Register the **Sender ID**. After registration succeeded. Input Server Key, message key and value for test.



Sender ID: 8020566583

Registration succeeded. Please run the following command to send a message.

API Key: AIzaSyDcUOVS3M31K6AZrD2T5uvf

```
curl -H "Content-Type:application/x-www-form-urlencoded;charset=UTF-8" -H
"Authorization: key=AIzaSyDcUOVS3M31K6AZrD2T5uvfCFhqP_iUiqs" -d
"registration_id=APA91bHgq4x4-tqyaYexl1dkSaaj-lb0XsHLJ1-gfSaCpPj5Te_z-
1m3AQm04k6iPRMQesM5yoFeSKRINrKncmjK7XUSWMB8UAkXWsDlZzsi-
TCtwBImu3zdm9t9ahryIEC3mHwbQYYc" -d data.Hello =World
https://android.googleapis.com/gcm/send
```

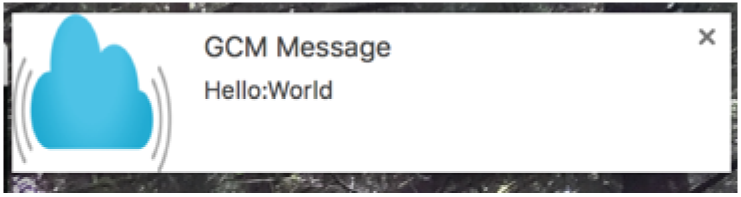
Message: Key Hello Value World

4. Then, we will see the curl information as

```
curl
-H "Content-Type:application/x-www-form-urlencoded;charset=UTF-8"
-H "Authorization: key=AIzaSyDcUOVS3M31K6AZrD2T5uvfCFhqP_iUiqs"
-d
"registration_id=APA91bHgq4x4-tqyaYexl1dkSaaj-lb0XsHLJ1-gfSaCpPj5Te_z
-1m3AQm04k6iPRMQesM5yoFeSKRINrKncmjK7XUSWMB8UAkXWsDlZzsi-TCtwBImu3zdm
9t9ahryIEC3mHwbQYYc"
-d data.Hello=World
https://android.googleapis.com/gcm/send
```

It shows two header blocks "Content-Type" and "Authorization", two data blocks "**registration_id**" and "data", with google api method "https://android.googleapis.com/gcm/send".

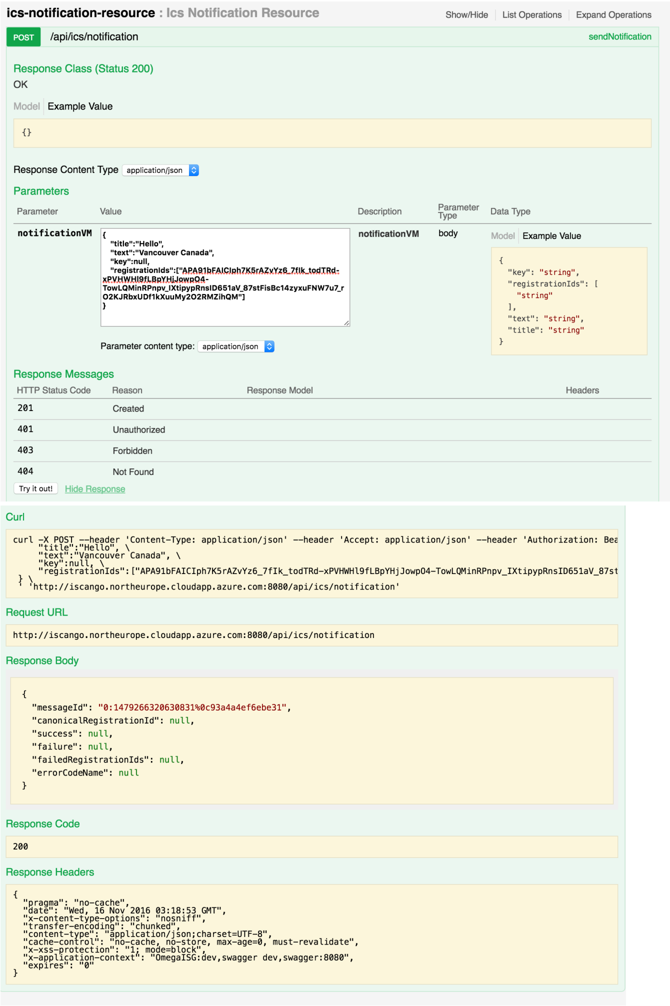
5. Run the above in command line. We will see the push notification pop-up.



GCM Message

Hello:World

6. We can receive notification by testing the API.



/api/ics/notification

Response Class (Status 200)

OK

Example Value

```
{}
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type

notificationVM	{
	"title":"Hello",
	"text":"Vancouver Canada",
	"key":null,
	"registrationIds":["APA91bFAICIph7K5rAZvYz6_7fIk_todTRd-xPVHWHl9fLBpYHjJowpO4-TowLQMinRPnpv_IXtipypRnsID651aV_87stFisBc14zyxuFNW7u7_rO2KJRbxUDf1kXuuMy2O2RMZihQM"]
	}
	Parameter content type: application/json
		notificationVM	body	Example Value
				{
				"key": "string",
				"registrationIds": [
				"string"
				],
				"text": "string",
				"title": "string"
				}

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Curl

```
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'Authorization: Bea
     "title":"Hello", \
     "text":"Vancouver Canada", \
     "key":null, \
     "registrationIds":["APA91bFAICIph7K5rAZvYz6_7fIk_todTRd-xPVHWHl9fLBpYHjJowpO4-TowLQMinRPnpv_IXtipypRnsID651aV_87st
 } \
 ' 'http://iscango.northeurope.cloudapp.azure.com:8080/api/ics/notification'
```

Request URL

```
http://iscango.northeurope.cloudapp.azure.com:8080/api/ics/notification
```

Response Body

```
{
  "messageId": "0:1479266320630831%0c93a4a4ef6ebe31",
  "canonicalRegistrationId": null,
  "success": null,
  "failure": null,
  "failedRegistrationIds": null,
  "errorCodeName": null
}
```

Response Code

```
200
```

Response Headers

```
{
  "pragma": "no-cache",
  "date": "Wed, 16 Nov 2016 03:18:53 GMT",
  "x-content-type-options": "nosniff",
  "transfer-encoding": "chunked",
  "content-type": "application/json;charset=UTF-8",
  "cache-control": "no-cache, no-store, max-age=0, must-revalidate",
  "x-xss-protection": "1; mode=block",
  "x-application-context": "OmegaISG:dev,swagger dev,swagger:8080",
  "expires": "0"
}
```

It will return "messageId" from FCM server.

References

1. Quick Start - Android
2. Firebase Documentation
3. Firebase Admin SDK to your Server [setup prerequisites]
4. Firebase Cloud Messaging notification from Java [stackoverflow]