<https://developers.google.com/cloud-messaging/>

GCM（Google Cloud Messaging）升级为FCM（Firebase Cloud Messaging）

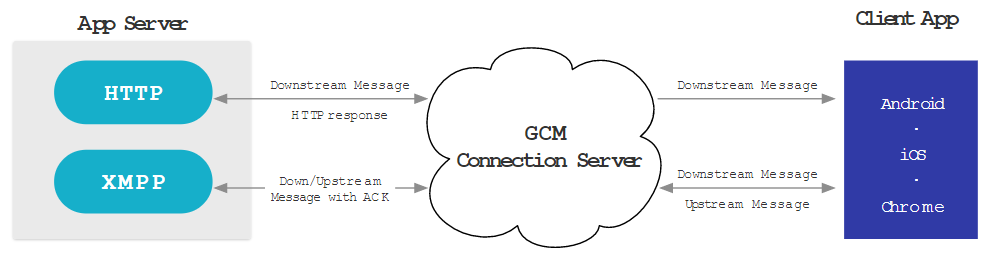
推送消息到客户端APP有3种方式：

单个设备

一组设备

订阅了主题的设备

GCM架构：



GCM Connection Server:

Google的服务器，参与App Server和Client App之间发送消息。

App Server：

程序员开发的，作为GCM实现的一部分，App Server通过GCM Connection Server向Client App发送消息。如果App Server实现了XMPP协议，它能接收来自Client App的消息。

Client App：

能与App Server通信的，支持GCM的App。

Sender ID:

在配置API项目时创建的唯一数值。用于在注册过程中识别是否允许App Server向Client App发送消息。

Server Key:

保存在App Server上的key，用于授权App Server访问Google Services。

HTTP协议中，Server Key包含在发送消息的POST请求的header中。

XMPP协议中，Server Key用于SASL PLAIN 身份验证请求中作为验证密码。

不要存放在客户端代码的任何地方。

Application ID:

对Android来讲，是清单文件中注册的包名。

Registration Token：

GCM Connection Server发送给Client App的ID，允许Client App接收消息。必须保密。

GCM工作流程：

1. Client App进行注册后，支持GCM，才能接收消息。
2. App Server给Client App发送消息
3. App Server发送消息到GCM Connection Server
4. 如果设备离线，GCM Connection Server将消息插入队列并保存
5. 当设备在线，GCM Connection Server发送消息给设备
6. 根据设备运行的特定平台，Client App根据实际情况接收消息
7. 如果使用XMPP协议，Client App能够给App Server发送消息。
8. 在设备端，Client App发送消息到XMPP connection server（需要查看设备是否支持）
9. 如果App Server未连接，XMPP connection server将消息插入队列并保存
10. 当App Server连接后，XMPP connection server发送消息给App Server
11. App Server收到消息后，解析消息头header进行验证，发送ack给XMPP connection server确认收到消息，可选地解析消息有效信息。

GCM的消息组成和API选项：

Massage的组成：GCM HTTP and XMPP connection server protocols通用

1. Target。必须有。当App Server发送消息的时候，必须指定一个标识消息目的地的target。
2. Options。当App Server发送消息给Client App的时候，能够设置很多选项（例如是否被后来者替换）。
3. Payload（有效载荷）。对于下发消息，GCM提供了两种类型的payload：notification和data。notification更轻量级，具有2KB的限制和预定义的一组用户可见的key，自动在设备通知栏上显示消息。data允许开发人员发送高达4KB的自定义 key/value对，必须由Client App处理消息，只包含自定义key/value。

notification payload：

{

"to" : "bk3RNwTe3H0:CI2k\_HHwgIpoDKCIZvvDMExUdFQ3P1...",

"notification" : {

"body" : "great match!",

"title" : "Portugal vs. Denmark",

"icon" : "myicon"

}

}

设备的通知栏会显示一条通知（即使App处于非活跃状态），body，title，icon是预定义的key。如果App处于活跃状态，notification payload会被传递到onMessageReceived()方法。

data payload：

{

"to" : "bk3RNwTe3H0:CI2k\_HHwgIpoDKCIZvvDMExUdFQ3P1...",

"data" : {

"Nick" : "Mario",

"body" : "great match!",

"Room" : "PortugalVSDenmark"

},

}

不论App处于前台还是后台，data payload会被传递到onMessageReceived()方法，利用其中的数据可以launch activity。

App Server能够发送消息同时包含notification payload和data payload，那么GCM处理显示在通知栏上的notification payload，Client App内部处理data payload。Client App接收到混合消息后，怎样处理与App处于什么状态有关：

{

"to" : "APA91bHun4MxP5egoKMwt2KZFBaFUH-1RYqx...",

"notification" : {

"body" : "great match!",

"title" : "Portugal vs. Denmark",

"icon" : "myicon"

},

"data" : {

"Nick" : "Mario",

"Room" : "PortugalVSDenmark"

}

}

如果App处于后台，App收到notification payload并显示在通知栏上，仅当用户点击通知时data notification才被传递到onMessageReceived()。

如果App处于前台，App会收到notification payload和data payload。

常用的消息选项：