To Create a Server to Send Push Notifications with GCM to Android Devices

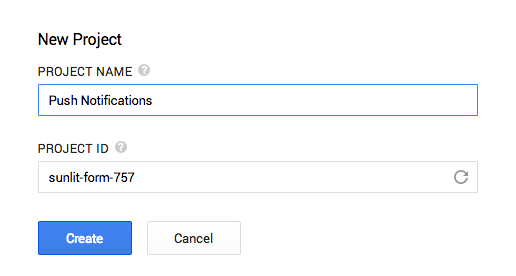
### About Push Notifications

Google-provided GCM Connection Servers take messages from a third-party application server, such as your Droplet, and send these messages to a GCM-enabled Android application (the client app) running on a device. Currently, Google provides connection servers for HTTP and XMPP.

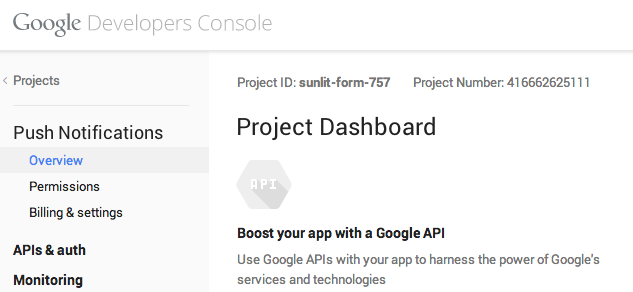


In other words, you need your own server to communicate with Google's server in order to send the notifications. Your server sends a message to a GCM (Google Cloud Messaging) Connection Server, then the connection server queues and stores the message, and then sends it to the Android device when the device is online.

1. We need to create a Google API project to enable GCM for our app.
2. Visit the [Google Developers Console](https://console.developers.google.com).
3. Click **Create Project**.
4. Enter a project name, then click **Create**.



1. Then, view your **Project ID** and **Project Number** on the upper left of the project page.



1. select **APIs & auth**.
2. Choose **APIs**.
3. In the displayed list of APIs, turn the Google Cloud Messaging for Android toggle to **ON**. Accept the terms of service.
4. **Google Cloud Messaging for Android** should now be in the list of enabled APIs for this project.
5. Choose **Credentials**.
6. Under **Public API access**, click **Create new Key**.
7. Choose **Server key**.
8. Click **Create**.
9. Copy the **API KEY**. You'll need to enter this on your server later.
10. In your phonegap application include the push notification source files from <https://github.com/guruSathsih/TentInternalProject-Camfox>
11. Include thes lines in AndroidManifest.xml

|  |
| --- |
| <!-- For push Notification GCM Access --> |
|  | <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" /> |
|  | <uses-permission android:name="android.permission.GET\_ACCOUNTS" /> |
|  | <uses-permission android:name="android.permission.WAKE\_LOCK" /> |
|  | <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" /> |
|  | <permission android:name="$PACKAGE\_NAME.permission.C2D\_MESSAGE" android:protectionLevel="signature" /> |
|  | <uses-permission android:name="$PACKAGE\_NAME.permission.C2D\_MESSAGE" /> |
|  |  |
|  | <!-- activity, receiver and service tags to the application section. --> |
|  | <activity android:name="com.plugin.gcm.PushHandlerActivity"/> |
|  | <receiver android:name="com.plugin.gcm.CordovaGCMBroadcastReceiver" android:permission="com.google.android.c2dm.permission.SEND" > |
|  | <intent-filter> |
|  | <action android:name="com.google.android.c2dm.intent.RECEIVE" /> |
|  | <action android:name="com.google.android.c2dm.intent.REGISTRATION" /> |
|  | <category android:name="$PACKAGE\_NAME" /> |
|  | </intent-filter> |
|  | </receiver> |
|  | <service android:name="com.plugin.gcm.GCMIntentService" /> |
|  |  |
|  | <activity ... android:launchMode="singleTop"> |

1. Include these lines in config.xml

|  |
| --- |
| <preference name="android-build-tool" value="gradle" /> |
|  | <gap:plugin name="phonegap-plugin-push" source="npm" /> | |
| <feature name="Notification"> | |
|  | | <param name="android-package" value="org.apache.cordova.dialogs.Notification" /> | |
|  | | </feature> | |
|  | | <feature name="Vibration"> | |
|  | | <param name="android-package" value="org.apache.cordova.vibration.Vibration" /> | |
|  | | </feature> | |

1. Add push.js file inside www folder.
2. Using javascript function get registeration id from your device and store it in database for future reference.

var push = PushNotification.init({

android: {

senderID: "127422360909"

},

ios: {

alert: "true",

badge: "true",

sound: "true"

},

windows: {}

});

push.on('registration', function(data) {

// data.registrationId

alert(data.registrationId);

$.ajax({

type:'POST',

url: getBaseURL()+"process/api.php?rquest=store\_registrationId",

data: {reg : data.registrationId},

success:function(responseText){

alert(responseText);

if(responseText != 0)

{

}

else {

alert("In else part");

}

}

});

});

push.on('notification', function(data) {

// data.message,

// data.title,

// data.count,

// data.sound,

// data.image,

// data.additionalData

});

push.on('error', function(e) {

alert(e.message);

});

1. Then send a push notification using php code

function sendPushNotification($to,$title,$message){

// API access key from Google API's Console

// replace API

//define( 'API\_ACCESS\_KEY', 'AIzaSyCtDCehTVdcYJQVQa\_tJ6cuUnYI\_6uWXKc');

if (!defined('API\_ACCESS\_KEY')) define('API\_ACCESS\_KEY', 'AIzaSyCtDCehTVdcYJQVQa\_tJ6cuUnYI\_6uWXKc');

$registrationIds = array($to);

$msg = array

(

'message' => $message,

'title' => $title,

'vibrate' => 1,

'sound' => 'beep.wav'

// you can also add images, additionalData

);

$fields = array

(

'registration\_ids' => $registrationIds,

'data' => $msg

);

$headers = array

(

'Authorization: key=' . API\_ACCESS\_KEY,

'Content-Type: application/json'

);

$ch = curl\_init();

curl\_setopt( $ch,CURLOPT\_URL, 'https://gcm-http.googleapis.com/gcm/send' );

curl\_setopt( $ch,CURLOPT\_POST, true );

curl\_setopt( $ch,CURLOPT\_HTTPHEADER, $headers );

curl\_setopt( $ch,CURLOPT\_RETURNTRANSFER, true );

curl\_setopt( $ch,CURLOPT\_SSL\_VERIFYPEER, false );

curl\_setopt( $ch,CURLOPT\_POSTFIELDS, json\_encode( $fields ) );

$result = curl\_exec($ch );

curl\_close( $ch );

echo $result;

}