Push Notification Application

|  |  |
| --- | --- |
| **Canada Office** | **Global Development Centre** |
| 120 -  1412 7th Ave W   Vancouver,  BC, V6H1C1, Canada.  🕾:+1.604.601.2185 | H 110, Sector 63,  NOIDA, U.P. 201 301 India  🕾 +91-9810286872 |

**Contents**

**Manual for using Push Notification sample application**

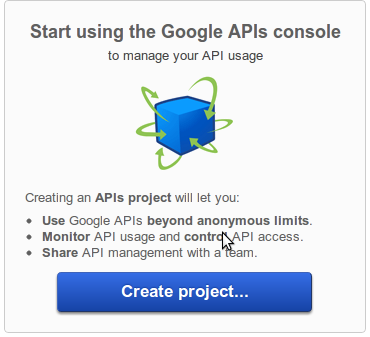
**PRE-REQUISITES :** To deploy this application, your phone should have Android market / playstore installed and at least one signed in Google account. If your Google account is signed out before you run the application or while the application is running, you will get an error and the push notification application will not work.

Abbreviations used :

**GCM** : Google Cloud Messaging

1. Open the [Google APIs Console page](https://code.google.com/apis/console).

2. If you haven't created an API project yet, this page will prompt you to do so:



**Note:** If you already have existing projects, the first page you see will be the **Dashboard** page. From there you can create a new project by opening the project drop-down menu (upper left corner) and choosing **Other projects > Create**.

3. Click **Create project**. Your browser URL will change to something like:

https://code.google.com/apis/console/#project:**4815162342**

Take note of the value after **#project:** (4815162342 in this example). This is your project ID, and it will be used later on as the GCM sender ID.

4. Enable the GCM service by following the below steps :

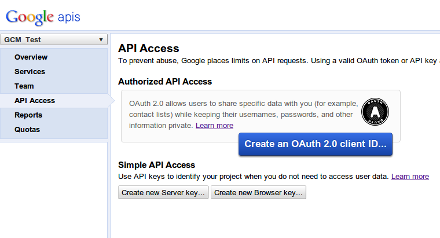
a. In the main Google APIs Console page, select **Services**.

b. Turn the **Google Cloud Messaging** toggle to ON.

c. In the Terms of Service page, accept the terms.

5. Obtain the API key :

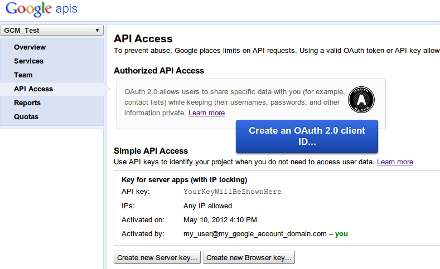
a) In the main Google APIs Console page, select **API Access**. You will see a screen that resembles the following:



b) Click **Create new Server key**. Either a server key or a browser key should work. The advantage to using a server key is that it allows you to white list IP addresses. The following screen appears:



c) Click **Create**:



Take note of the **API key** value (**YourKeyWillBeShownHere**) in this example, as it will be used later on.

**Requirements**

For the web server: (**Here we are using Apache Tomcat as our web server**)

* [Ant 1.8](http://ant.apache.org/) (it might work with earlier versions, but it's not guaranteed).
* One of the following:
  + A running web server compatible with Servlets API version 2.5, such as [Tomcat 6](http://tomcat.apache.org/) or [Jetty](http://jetty.codehaus.org/), or
  + [Java App Engine SDK](http://code.google.com/appengine/) version 1.6 or later.
* A Google account registered to use GCM.
* The API key for that account.

For the Android application:

* The Google API project ID of the account registered to use GCM.

## Setting Up the Server

1. In a text editor, edit the **android-sdk-windows\extras\google\gcm\samples\gcm-demo-server\WebContent\WEB-INF\classes\api.key** and replace the existing text with the API key obtained above.

2. In a command prompt, go to **android-sdk-windows\extras\google\gcm\samples\gcm-demo-server** directory and type **Ant**.

3 "Build successful" message will appear.

4. Also you will see a “dist” folder created with a “gcm-demo.war” file under the **android-sdk-windows\extras\google\gcm\samples\gcm-demo-server** folder**.**

5. Copy this file to your webapps folder inside your Apache Tomcat folder.

6. In the command prompt, go to the directory : \apache-tomcat\bin and type startup.bat to start your server.

7. Open a browser and type http:///<your\_host\_ip>:8080/gcm-demo/

8. It will show “No Device Registered”, so your server side is ready.

**Running the application**

1. Open the Eclipse and import this sample application.

2. Go to the "**CommonUtilities.java**" file under the source folder.

3. Replace the IP Address in the *SERVER\_URL* variable by your machine's IP Address.

4. Replace the *SENDER\_ID* variable by your project ID that you received from the google console page.

5. Run the application on your phone.

6. If your device is registered on the server, your device will display a message that registration successful.

7. Refresh the browser page http:///<your\_host\_ip>:8080/gcm-demo/

8. The Web page will show that 1 device is registered and it will show a button to send a message.

9. Click on the button and your device will show a notification that a message has been received.

**This was how the sample application of push notification works.**