

# Set Up Buzztouch Android app with Google Cloud Services Dashboard

Created by Alan MacGechan 28<sup>th</sup> January 2013

## Overview

- 1) Log into my Google Account on the internet and create a Project ID and an Android API Key.
- 2) Set up the Buzztouch Control Panel.
- 3) Make changes in Eclipse

## Anatomy of an Android Push Notification

User → Backend → Google → App

User sends push from the backend console



Backend sends push to Google GCM (Google Cloud Messaging)



Google sends push to phone with app



### Establish app "Credentials"

- 1) Register for a GCM API key with a Project ID
- 2) Update Backend server with API Key and Project Number
- 3) Update App in Eclipse & build

### Register for Push Notifications

Device registers with GCM & obtains registration token when the app first starts

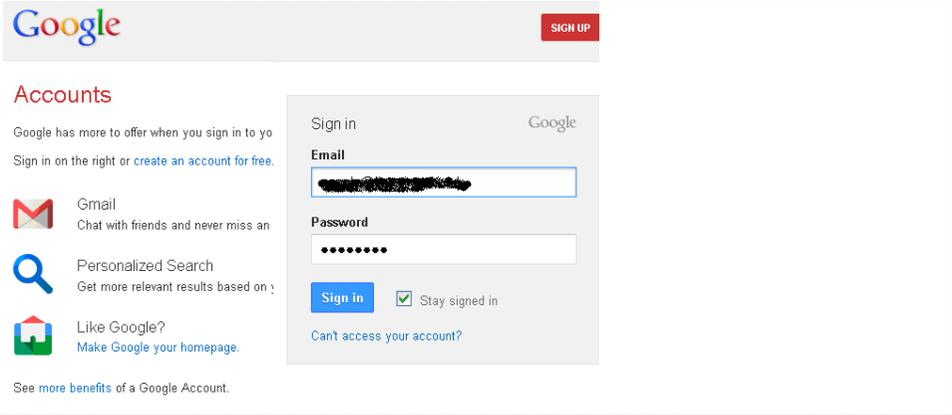
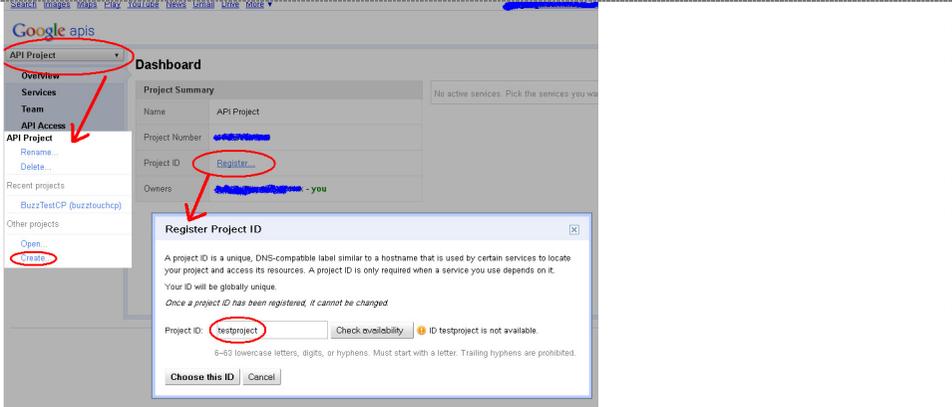
Device sends registration token to Backend for later use if user confirms 'OK'

### Send a Push

Backend sends push to GCM with device token

GCM sends push to device identified by token

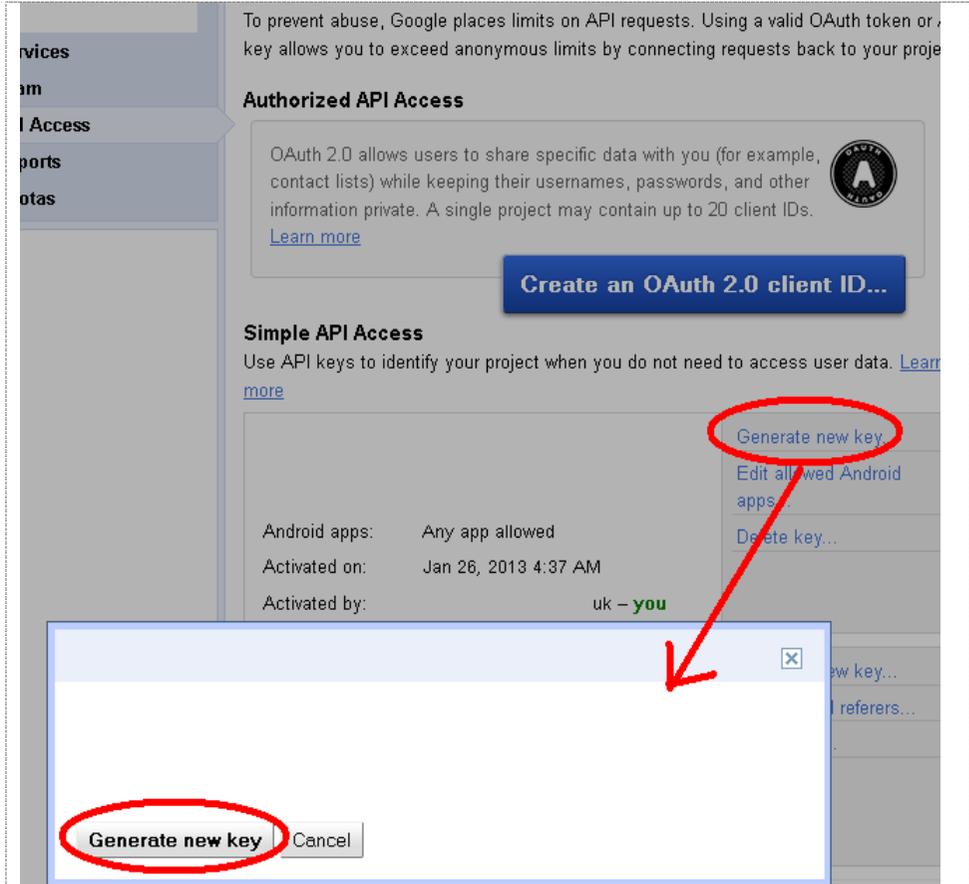
## Detailed Step-by-Step Guide

	Step	Comment	Example
1	<b>Create a Google Project ID and an Android API Key for GCM</b>	<b>Google Cloud Messaging is the bearer for the messaging service</b>	
1a	Sign in to your Google Developers account.	Navigate to <a href="http://code.google.com/">http://code.google.com/</a> Register if you do not have an account.	
1b	Create an API project	<a href="https://code.google.com/apis/console/">https://code.google.com/apis/console/</a>  Navigate to Google apis, click on the top menu  The Project ID name that you register must be all lower case.	

1c Generate a new key for Android apps

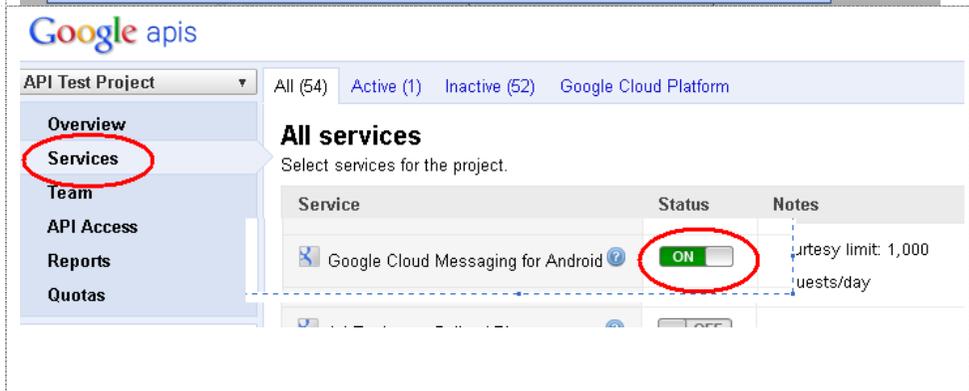
Do this once you have a Project ID name,.

Click on API Access and just generate a key, don't link it to anything and it is allowed for all apps. Do not create an OAuth 2.0 client ID



1d Enable Google Cloud Messaging for Android

Navigate to Services to do this



1e Make a note of the Android API Key

You will need this later

The screenshot shows the 'API Access' page for a project named 'API Test Project'. On the left, a navigation menu lists 'Overview', 'Services', 'Team', 'API Access' (highlighted with a red circle), 'Reports', and 'Quotas'. The main content area is titled 'API Access' and includes a warning about Google's API request limits. Below this, there are sections for 'Authorized API Access' (with a 'Create an OAuth 2.0 client ID...' button) and 'Simple API Access'. The 'Simple API Access' section contains a red circle around the 'Key for Android apps (with certificates)' information, which includes: API key: AIzaSyA-1YJxBuosQzrQbGyukCkRtAoY6sX2yT0, Android apps: Any app allowed, Activated on: Jan 26, 2013 4:37 AM, and Activated by: you.

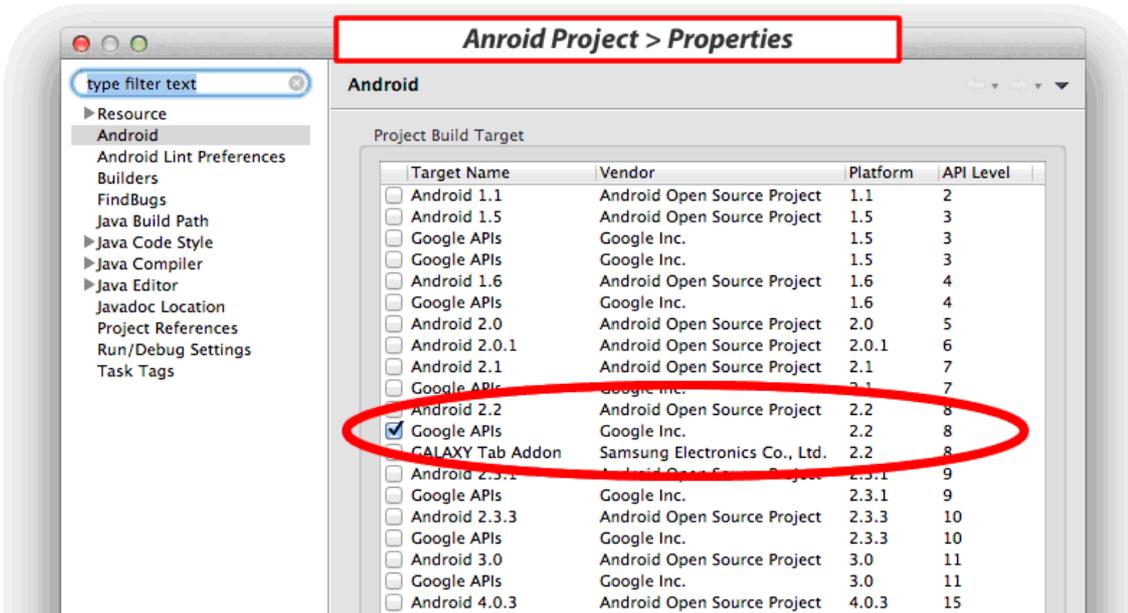
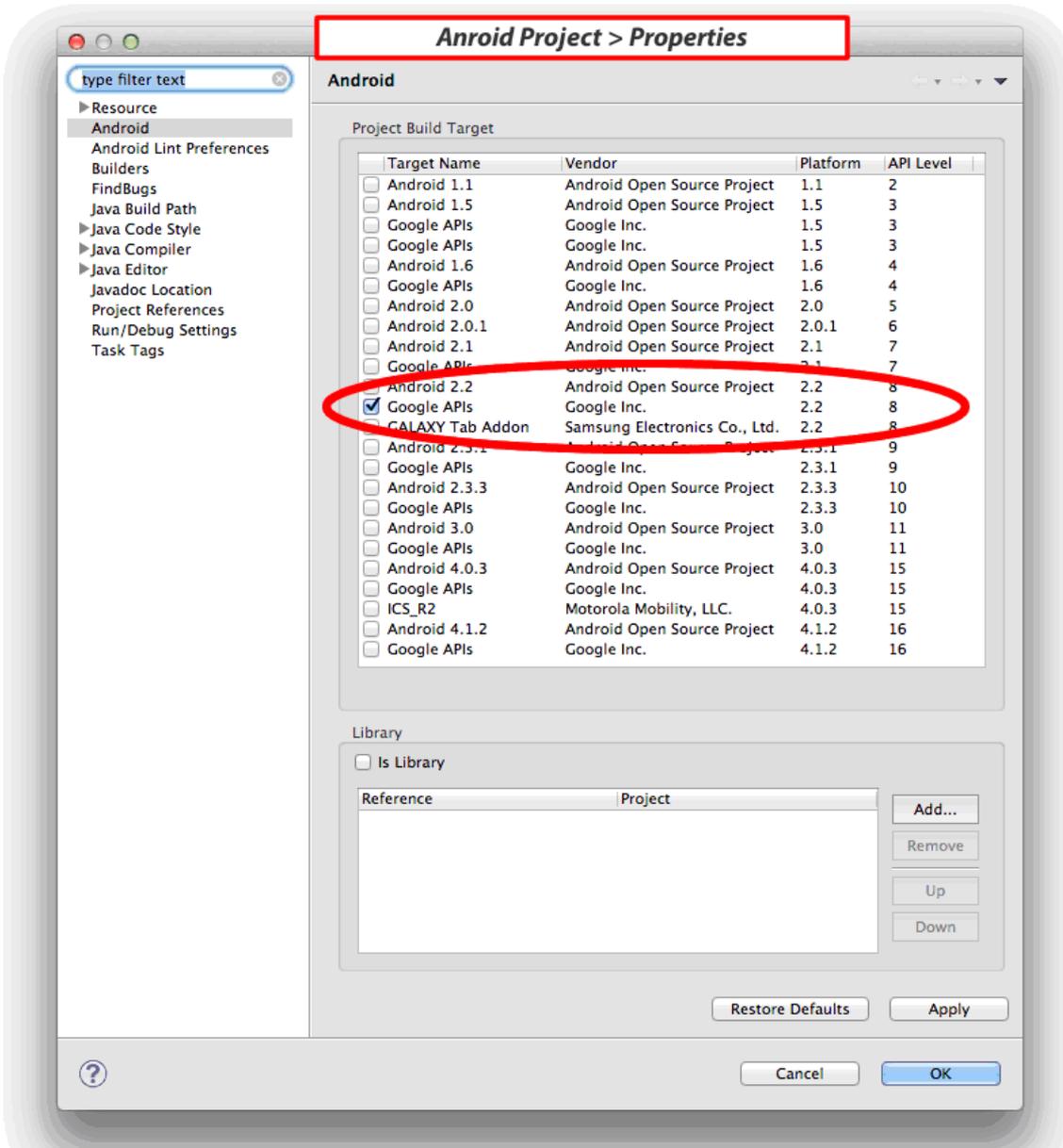
1e Navigate to Overview for your Google project and record the Project Number required for your app

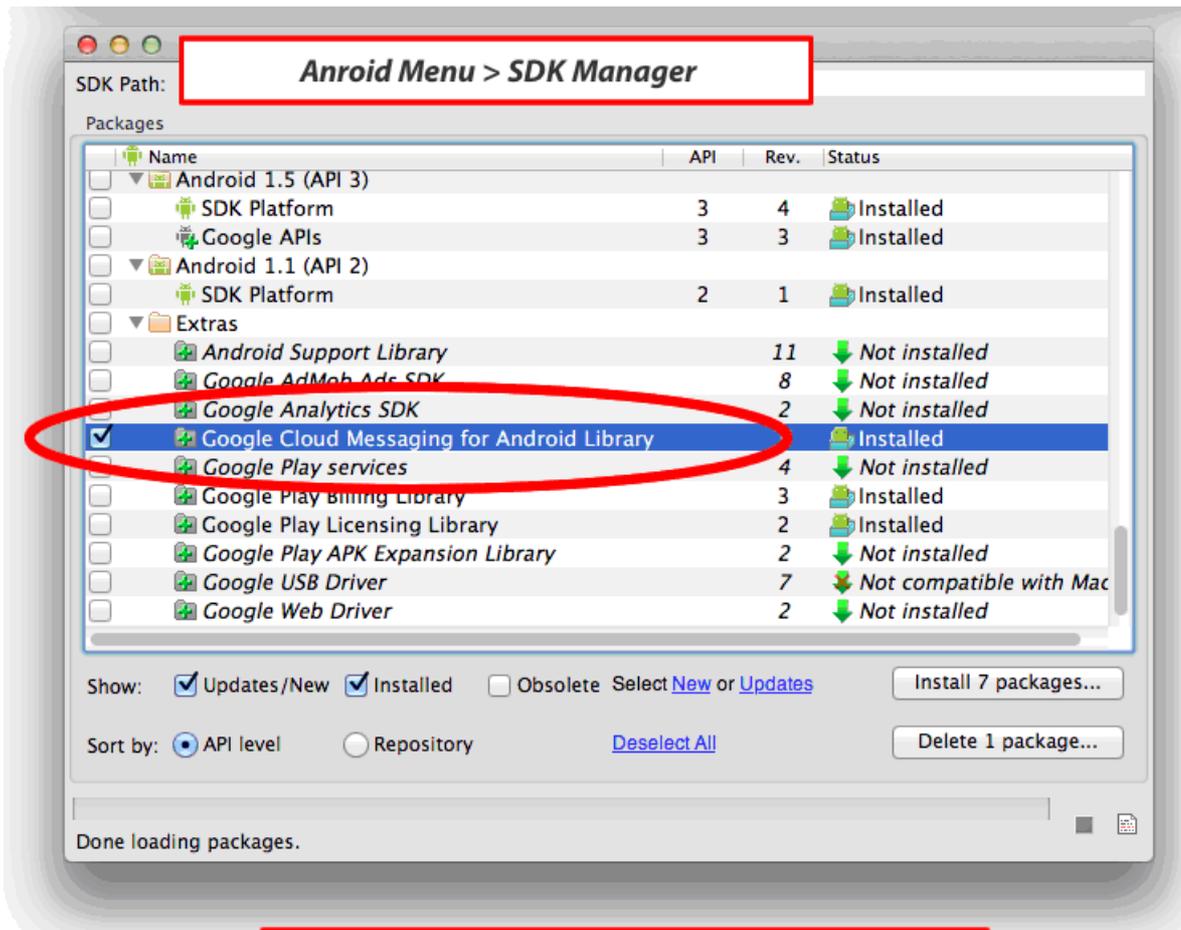
This panel holds the critical information required to implement GCM in your app

The screenshot shows the 'Dashboard' page for the 'API Test Project'. On the left, a navigation menu lists 'Overview' (highlighted with a red circle), 'Services', 'Team', 'API Access', 'Reports', and 'Quotas'. The main content area is titled 'Dashboard' and features a 'Project Summary' table and a 'Service' table. The 'Project Summary' table has a red circle around the 'Project Number' field, which contains the value '9999999999'. The 'Service' table shows 'Google Cloud Messaging for Android' with a status of 'No known issues'.

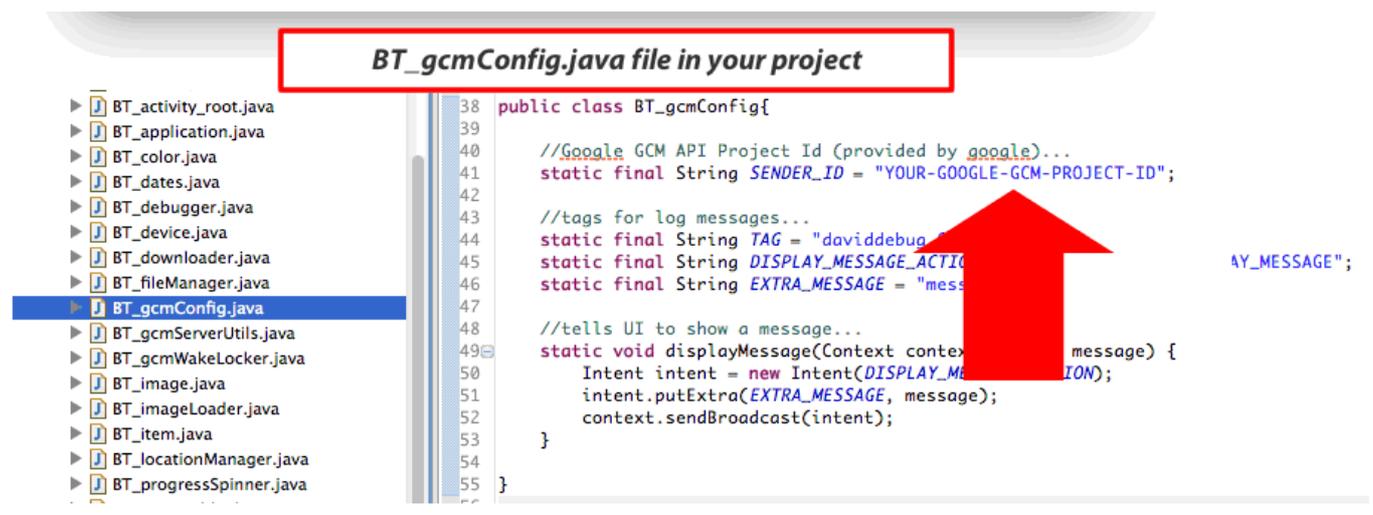
2a Enable the SDK in Eclipse

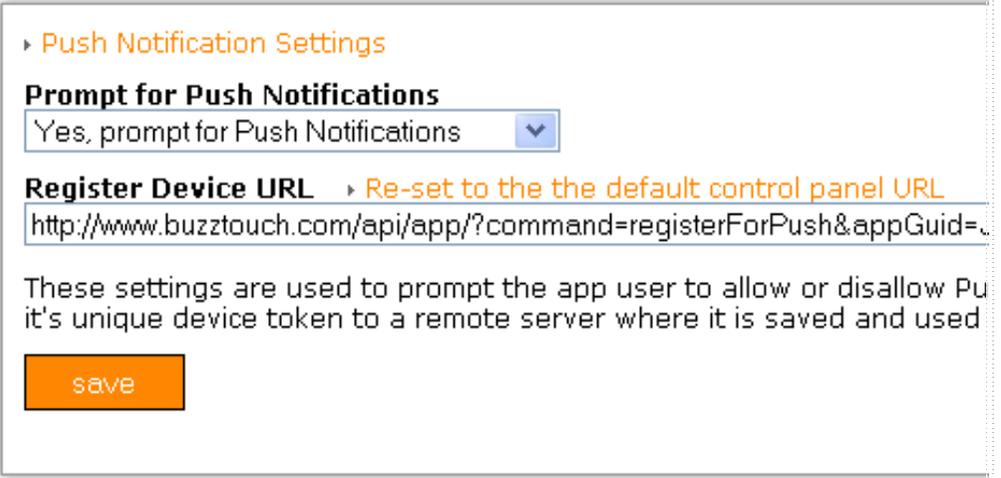
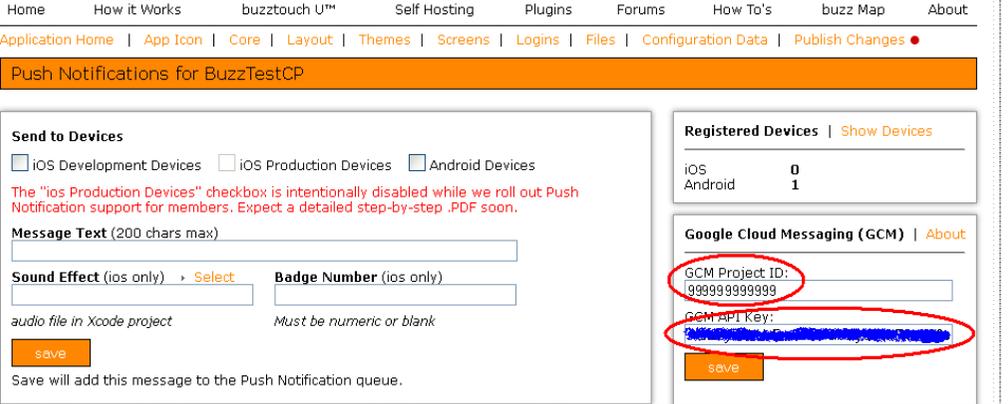
Screenshot graphic here is the original provided by David Book





2b Add the Project Number into the project in Eclipse File to edit is bt\_gcmConfig.java, one entry to change, around line 47



3	<b>Enable changes in the BuzzTouch Control Panel</b>	<b>Go to the project on the BuzzTouch Control Panel Eclipse and enter the Project Number</b>					
3a	Set the Push Notification Settings Core Properties	In Core Properties for your application, enable your app for push, reset the default URL so it is visible and save.	 <p>► <b>Push Notification Settings</b></p> <p><b>Prompt for Push Notifications</b> Yes, prompt for Push Notifications ▼</p> <p><b>Register Device URL</b> ► <b>Re-set to the the default control panel URL</b> <code>http://www.buzztouch.com/api/app/?command=registerForPush&amp;appGuid=...</code></p> <p>These settings are used to prompt the app user to allow or disallow Push Notifications. The app user's unique device token to a remote server where it is saved and used for push notifications.</p> <p><b>save</b></p>				
3b	Add the GCM values for your app under Push Notifications menu	Enter the Google Android API key and the Project Number. <b>Note</b> - Do not use Project ID, it is the Project Number that is needed.	 <p>Home   How it Works   buzztouch U™   Self Hosting   Plugins   Forums   How To's   buzz Map   About</p> <p>Application Home   App Icon   Core   Layout   Themes   Screens   Logins   Files   Configuration Data   Publish Changes ●</p> <p><b>Push Notifications for BuzzTestCP</b></p> <p><b>Send to Devices</b>  <input type="checkbox"/> iOS Development Devices   <input type="checkbox"/> iOS Production Devices   <input type="checkbox"/> Android Devices  <small>The "ios Production Devices" checkbox is intentionally disabled while we roll out Push Notification support for members. Expect a detailed step-by-step .PDF soon.</small></p> <p><b>Message Text</b> (200 chars max)  <input type="text"/></p> <p><b>Sound Effect</b> (ios only) ► <b>Select</b>   <b>Badge Number</b> (ios only)  <input type="text"/>   <input type="text"/>  <small>audio file in Xcode project   Must be numeric or blank</small></p> <p><b>save</b>  Save will add this message to the Push Notification queue.</p> <p><b>Registered Devices</b>   <a href="#">Show Devices</a></p> <table border="1"> <tr><td>iOS</td><td>0</td></tr> <tr><td>Android</td><td>1</td></tr> </table> <p><b>Google Cloud Messaging (GCM)</b>   <a href="#">About</a></p> <p>GCM Project ID:  <input type="text" value="999999999999"/></p> <p>GCM API Key:  <input type="text" value="AIzaSyA..."/></p> <p><b>save</b></p>	iOS	0	Android	1
iOS	0						
Android	1						

If it works, you devices will show up as a registered device in the Buzztouch control panel  
When you launch your app, it will ask you if you wish to receive push notifications  
When you send a push, and your app needs to be running, it will be received on your devices as a system notification