

# GCM Notification Plugin

## Plugin Information

Distribution of this plugin has been suspended due to unresolved security vulnerabilities, see below.



The current version of this plugin may not be safe to use. Please review the following warnings before use:

- [Credentials stored in plain text](#)

Sends build notifications to Android devices using the Google Cloud Messaging (GCM) service.

## Setup

Once fully set up, Jenkins will be able to send notifications to the Google Cloud Messaging (GCM) service — formerly known as Cloud to Device Messaging (C2DM) — which will forward those notifications to the relevant devices which have been paired with your Jenkins server.

## Prerequisites

### Admin

- Jenkins 1.466+
- A Google Account

### Users

- A device running Android 2.2 or newer
  - Must be signed into a Google Account, unless the phone is running Android 4.0.4+
  - Must have an app installed which is compatible with this plugin (e.g. [BuildWatch](#)).
- Preferably your Jenkins server should be accessible from your Android device (e.g. both on the same wifi network, or your Jenkins instance is publicly available)

## Admin: One-time Jenkins setup

Once you have the plugin installed, you must use your Google Account to create a new API Project, which gives you access to use the GCM service.

1. Sign in to the [Google APIs Console](#)
2. Create a new project
3. Enable the "Google Cloud Messaging for Android" service
4. Under "API Access", create a new Server Key and note the "API key" value
5. From the project "Overview" page, note the "Project Number" value
6. Paste these two values into the respective fields under the Jenkins system config (Manage Jenkins > Configure System)

This process is demonstrated in this video: <https://www.youtube.com/watch?v=3QFVUg0nfRM>

## Users: One-time device setup

Each user who wishes to be able to receive notifications must pair their device with the particular Jenkins instance.

The simplest way to set up an Android device and receive notifications is to use the [BuildWatch app](#).

Ways in which other apps can integrate with the GCM plugin are described below, but for now we'll assume the use of BuildWatch.

1. You must be logged in to an account on your Jenkins instance, as each device must be associated with a user account.
2. From your user's configuration page (i.e. click your user ID in the top right corner of any page, then click "Configure" on the left), you'll see a "Google Cloud Messaging Token" section.
  - Here you should see a "Show QR code" button. If not, you're either not logged in, or looking at the configuration page for a different user.
3. Clicking the button will display a QR code which you can scan with the BuildWatch app by clicking "Auto setup" in the app.
4. Scanning the code will request a GCM token from Google and then register this with your Jenkins server.

Now your device is ready to receive notifications.

This process is demonstrated in this video: <https://www.youtube.com/watch?v=mt0ctjBTiZw>

## Configure jobs

Once you have one or more users with devices paired to your Jenkins server, you can enable build notifications.

In any job, enable the "Notify Android devices" post-build action.

Here you can enter the Jenkins user IDs you wish to be notified, separated by spaces.

The advanced options available are specified by the [Instant Messaging Plugin](#).

## Troubleshooting

### I don't see a "Show QR code" button on my user page

Make sure you're logged into Jenkins, and that you're looking at the correct user configuration page. e.g. If you're logged in as an admin user, you may have access to other users pages. Here the QR code won't be available.

### I scanned the QR code, but I can't see any token on my user configuration page

If you scanned the QR code and directly afterwards pressed "Enter token manually", you won't see the token — you must reload the page first.

Alternatively, your Android device failed to upload the GCM token to Jenkins. This can be for several reasons:

- Your Android device doesn't currently have network connectivity
- Your Jenkins server doesn't currently have network connectivity
- Your Jenkins server is not accessible from your Android device (e.g. Android device is not on the same wifi network)
- The hostname of the Jenkins server is something that the Android device cannot resolve (e.g. Jenkins hostname "our-ci-server.local" may not be a resolvable by the DNS server the Android device is using)

Check that you have the correct connectivity and try again. You can also confirm from your Android browser that you can reach the Jenkins URL listed under Manage Jenkins > Configure System > Jenkins URL.

## Implementing compatible Android apps

If you want to integrate Jenkins GCM notifications into your own app, you can use the following info.

### Registering

#### Recognising QR code

The QR code shown by this plugin simply contains a JSON object with the following key/value string pairs:

Key	Value
username	Jenkins user ID
token	Jenkins API token for the user
senderId	Google API project number (from Configure System > Google Cloud Messaging Notifications)
url	Jenkins base URL (from Configure System > Jenkins URL)

For example:

```
{ "username": "bob", "token": "8feb49ac4902ccd17e5a0", "senderId": "987654321", "url": "http://ci.example.com/" }
```

The `senderId` is used to request a GCM device token via the Android APIs. Once received, the `username` and `token` are used to send the value to the Jenkins instance at `url`.

#### Uploading token

Once you have a GCM device token, it should be sent as an HTTP POST request to `JENKINS_URL/gcm/register` with the parameter `token` containing the GCM device token.

The HTTP request should be made as the Jenkins user in question — i.e. you should use the user ID and API token from the QR code to do [HTTP Basic authentication](#).

Jenkins will reply with HTTP 200 OK if the token was registered, and with the relevant error code otherwise, depending on whether the token was missing or the user was not authenticated.

Once you've uploaded the user's GCM device token, you can safely throw away the info gained from the QR code. If you do want to hold onto the user ID and their API token, you should give the user a method of deleting these from your app, as these values are sensitive — i.e. they can be used to perform any action as that Jenkins user.

## Receiving notifications

When this plugin sends a notification to a device, it currently simply sends a single string value "m", containing the plain text of the notification, as you would normally see from the Instant Messaging plugin.

e.g. "Project foobar build #123: SUCCESS in 2 min: <http://ci.example.com/job/foobar/123/>"

Your app will receive this directly as an `Android Intent`, with the `String` value of "m" in the intent extras.

Currently the Instant Messaging plugin doesn't deliver more structured data to its users, but this is something we'd like to improve on in future...

## More info

You can find various demonstrations, setup instructions and further info at [buildwatch.org](http://buildwatch.org).

If you have any questions, get in touch via the links here or on the BuildWatch site.

## Version history

### Version 1.0 (November 20, 2012)

- Initial release