

Gitee Plugin

Table of Contents

- [Introduction](#)
- [User support](#)
- [Global plugin configuration](#)
 - [Gitee-to-Jenkins auth](#)
 - [Jenkins-to-Gitee auth](#)
- [Jenkins Job Configuration](#)
 - [Git configuration](#)
 - [Freestyle jobs](#)
 - [Job trigger configuration](#)
 - [Webhook URL](#)
 - [Freestyle jobs](#)
 - [Build status configuration](#)
 - [Freestyle jobs](#)
- [Advanced features](#)
 - [Branch filtering](#)
 - [Add a note to merge requests](#)
- [Release Workflow](#)

Introduction

This plugin allows Gitee to trigger builds in Jenkins when code is committed or merge requests are opened/updated. It can also send build status back to Gitee.

Seeking maintainers

This plugin was developed base on [GitLab Plugin](#) by [Gitee.com](#). [Gitee.com](#) will continue to maintain this plugin.

User support

If you have a problem or question about using the plugin, please make sure you are using the latest version. Then create an issue in the Gitee project if necessary. New issues should include the following:

- Jenkins version (e.g. 2.111)
- Relevant log output from the plugin (see below for instructions on capturing this)

Gitee Jenkins plugin introduced improved logging for debugging purposes. To enable it:

1. Go to Jenkins -> Manage Jenkins -> System Log
2. Add new log recorder
3. Enter 'Gitee plugin' or whatever you want for the name
4. On the next page, enter 'com.gitee.jenkins' for Logger, set log level to FINEST, and save
5. Then click on your Gitee plugin log, click 'Clear this log' if necessary, and then use Gitee to trigger some actions
6. Refresh the log page and you should see output

Global plugin configuration

Gitee-to-Jenkins authentication

By default the plugin will require authentication to be set up for the connection from Gitee to Jenkins, in order to prevent unauthorized persons from being able to trigger jobs.

Configuring global authentication

1. Create a user in Jenkins which has, at a minimum, Job/Build permissions
2. Log in as that user (this is required even if you are a Jenkins admin user), then click on the user's name in the top right corner of the page
3. Click 'Configure,' then 'Show API Token...,' and note/copy the User ID and API Token
4. In Gitee, when you create webhooks to trigger Jenkins jobs, use this format for the URL and do not enter anything for 'Secret Token': http://USERID:APITOKEN@JENKINS_URL/project/YOUR_JOB
5. After you add the webhook, click the 'Test' button, and it should succeed

Configuring per-project authentication

If you want to create separate authentication credentials for each Jenkins job:

1. In the configuration of your Jenkins job, in the Gitee configuration section, click 'Advanced'
2. Click the 'Generate' button under the 'Secret Token' field
3. Copy the resulting token, and save the job configuration
4. In Gitee, create a webhook for your project, enter the trigger URL (e.g. http://JENKINS_URL/project/YOUR_JOB) and paste the token in the Secret Token field
5. After you add the webhook, click the 'Test' button, and it should succeed

Jenkins-to-Gitee authentication

PLEASE NOTE: This auth configuration is only used for accessing the Gitee API for sending build status to Gitee. It is not used for cloning git repos. The credentials for cloning (usually SSH credentials) should be configured separately, in the git plugin.

This plugin can be configured to send build status messages to Gitee, which show up in the Gitee Pull Request UI. To enable this functionality:

1. Create a new user in Gitee
2. Give this user 'Developer' permissions on each repo you want Jenkins to send build status to
3. Log in or 'Impersonate' that user in Gitee, click the user's icon/avatar and choose Settings
4. Click on 'Access Tokens'
5. Create a token named e.g. 'jenkins' with 'api' scope; expiration is optional
6. Copy the token immediately, it cannot be accessed after you leave this page
7. On the Global Configuration page in Jenkins, in the Gitee configuration section, supply the Gitee host URL, e.g. <http://your.gitee.server>
8. Click the 'Add' button to add a credential, choose 'Gitee API token' as the kind of credential, and paste your Gitee user's API key into the 'API token' field
9. Click the 'Test Connection' button; it should succeed

Jenkins Job Configuration

There are two aspects of your Jenkins job that you may want to modify when using Gitee to trigger jobs. The first is the Git configuration, where Jenkins clones your git repo. The Gitee Jenkins Plugin will set some environment variables when Gitee triggers a build, and you can use those to control what code is cloned from Git. The second is the configuration for sending the build status back to Gitee, where it will be visible in the commit and/or pull request UI.

You will need to update this code anytime you add or remove parameters.

Git configuration

Freestyle jobs

In the *Source Code Management* section:

1. Click *Git*
2. Enter your *Repository URL*, such as `git@your.gitee.server:gitee_group/gitee_project.git`
 - a. In the *Advanced* settings, set *Name* to `origin` and *Refspec* to `+refs/heads/*:refs/remotes/origin/* +refs/pull/* /MERGE:refs/pull/* /MERGE`
3. In *Branch Specifier* enter:
 - a. For single-repository workflows: `origin/${giteeSourceBranch}`
 - b. For forked repository workflows: `merge-requests/${giteePullRequestId}`
4. In *Additional Behaviours*:
 - a. Click the *Add* drop-down button
 - b. Select *Merge before build* from the drop-down
 - c. Set *Name of repository* to `origin`
 - d. Set *Branch to merge* as `${giteeTargetBranch}`

Job trigger configuration

Webhook URL

When you configure the plugin to trigger your Jenkins job, by following the instructions below depending on job type, it will listen on a dedicated URL for JSON POSTs from Gitee's webhooks. That URL always takes the form http://JENKINS_URL/project/PROJECT_NAME, or http://JENKINS_URL/project/FOLDER/PROJECT_NAME if the project is inside a folder in Jenkins. You should not be using http://JENKINS_URL/job/PROJECT_NAME/build or http://JENKINS_URL/job/gitee-plugin/buildWithParameters, as this will bypass the plugin completely.

Freestyle jobs

1. In the *Build Triggers* section:
 - Select *Build when a change is pushed to Gitee*
 - Copy the *Gitee webhook URL* shown in the UI (see [here](#) for guidance)

- Use the check boxes to trigger builds on *Push Events* and/or *Created Pull Request Events* and/or *Accepted Pull Request Events* and/or *Closed Pull Request Events*
 - Optionally use *Rebuild open Pull Requests* to enable re-building open merge requests after a push to the source branch
 - If you selected *Rebuild open Pull Requests* other than *None*, check *Comments*, and specify the *Comment for triggering a build*. A new build will be triggered when this phrase appears in a commit comment. In addition to a literal phrase, you can also specify a Java regular expression
2. Configure any other pre build, build or post build actions as necessary
 3. Click *Save* to preserve your changes in Jenkins
 4. Create a webhook in the relevant Gitee projects (consult the Gitee documentation for instructions on this), and use the URL you copied from the Jenkins job configuration UI. It should look something like http://JENKINS_URL/project/yourbuildname

Build status configuration

You can optionally have your Jenkins jobs send their build status back to Gitee, where it will be displayed in the commit or merge request UI as appropriate.

Freestyle jobs

Freestyle jobs can only send build status after the build steps are complete. To do this, choose 'Publish build status to Gitee' from the available 'Post-build actions' in your Jenkins job config. Also make sure you have chosen the appropriate Gitee instance from the 'Gitee connection' dropdown menu, if you have more than one.

Advanced features

Branch filtering

Triggers may be filtered based on the branch name, i.e. the build will only be allowed for selected branches. On the project configuration page, when you configure the Gitee trigger, you can choose 'Filter branches by name' or 'Filter branches by regex.' Filter by name takes comma-separated lists of branch names to include and/or exclude from triggering a build. Filter by regex takes a Java regular expression to include and/or exclude.

Note: This functionality requires access to Gitee and a git repository url already saved in the project configuration. In other words, when creating a new project, the configuration needs to be saved *once* before being able to add branch filters. For Pipeline jobs, the configuration must be saved *and* the job must be run once before the list is populated.

Add a note to merge requests

To add a note to Gitee merge requests after the build completes, select 'Add note with build status on Gitee merge requests' from the optional Post-build actions. Optionally, click the 'Advanced' button to customize the content of the note depending on the build result.

Release Workflow

To perform a plugin hpi file, maintainers can run `mvn package` To release a snapshot, e.g. with a bug fix for users to test, just run `mvn hpi:run`