

Gitea/Jenkins: Setting up Webhooks

From Gcube Wiki

Contents

- 1 What is a webhook?
- 2 Gitea Plugin on Jenkins (only for Jenkins admins)
 - 2.1 Installation
 - 2.2 Configuration
- 3 Build project configuration
- 4 Webhook on the Gitea repository
- 5 Testing the Gitea/Jenkins round trip

What is a webhook?

A webhook is a mechanism to automatically trigger the build of a Jenkins project upon a commit pushed in a Git repository.

This guide details the steps to have Jenkins automatically create a build if it detects changes to a Gitea repository. This can be a very useful improvement to continuous integration setup with Jenkins because this method is only telling Jenkins to attempt a new build when a change is detected rather than polling on an interval, which can be a very inefficient.

Gitea Plugin on Jenkins (only for Jenkins admins)

This plugin allows to configure Jenkins to talk to Gitea.

Installation

In Jenkins: download and install the Gitea plugin in Jenkins.

Go in the page: Manage Jenkins -> Manage Plugins -> Available -> Gitea plugin

And install the plugin.

Configuration

Go in the page: Manage Jenkins -> Configure System -> Gitea Servers:

Gitea Servers

Gitea Server

Name:

Server URL:

Gitea Version: 73ce024

☒ Manage hooks

Credentials: (Git user that can access the gCube/D4Science private repositories) ➤ Add

Hook management will be performed as git.gcube

Alias URL:

Add Delete

Build project configuration

In Jenkins, under the project settings page "Source Code Management":

- set option to "Git",
- provide URL to your repo (e.g. https://code-repo.d4science.org/gCubeSystem/gxRest.git),
- set the credentials to *git.gcube* (only if the repository is NOT public),
- specify **/master* as branch to build,

and in "Build Triggers" section:

- check "Poll SCM" option with no schedule defined. This setup basically tells Jenkins to poll your Gitea repo only when requested via the webhook.

General **Source Code Management** Build Triggers Build Environment Pre Steps Build Post Steps Build Settings Post-build Actions

☐ None
☒ Git

Repositories

Repository URL

Credentials [Add](#)

[Advanced...](#)
[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any') [X](#) [?](#)

[Add Branch](#)

Repository browser [?](#)

Additional Behaviours [Add](#)

☐ Subversion [?](#)

Build Triggers

☒ Build whenever a SNAPSHOT dependency is built [?](#)

☐ Schedule build when some upstream has no successful builds [?](#)

☐ Trigger builds remotely (e.g., from scripts) [?](#)

☐ Build after other projects are built [?](#)

☐ Build periodically [?](#)

☐ GitHub hook trigger for GITScm polling [?](#)

☒ Poll SCM [?](#)

Schedule


No schedules so will only run due to SCM changes if triggered by a post-commit hook [?](#)

Ignore post-commit hooks ☐ [?](#)

Webhook on the Gitea repository

In Gitea, under repo -> Settings -> Webhooks:

- click on "Add Webhook",
- select "Gitea" in the drop down list

 **gCubeSystem** / **gxRest** [Unwatch](#) 13 [Unstar](#) 2 [Fork](#) 0

[Code](#) [Issues](#) [Pull Requests](#) 0 [Releases](#) 1 [Wiki](#) [Activity](#) [Settings](#)

Repository Collaborators Branches **Webhooks** Deploy Keys

Webhooks [Add Webhook](#)

Webhooks automatically make HTTP POST requests to a server when certain Gitea events trigger. Read more in the [webhooks guide](#).

[Gitea](#)
[Gogs](#)

In the Webhook form:

- set the URL to `https://jenkins.d4science.org/gitea-webhook/post?job=project`, where *project* is the name of the project in Jenkins to build,
- set post content type to "application/json",
- check "Push Events" for the *Trigger On* option,
- clear the secret (leave it blank).

Update Webhook

Gitea will send POST requests with a specified content type to the target URL. Read more in the [webhooks guide](#).

Target URL *

POST Content Type

Secret

Trigger On:

☒ Push Events
☐ All Events
☐ Custom Events...

☒ Active

Information about triggered events will be sent to this webhook URL.

Update Webhook

Remove Webhook

Recent Deliveries

Test Delivery

✓

17421524-ac8c-4708-b697-e9a2dccb7155

2019-05-09 22:14:17 CEST

Request

Response 200

Headers

```

Request URL: https://jenkins.d4science.org/gitea-webhook/post?job=gxRest
Request method: POST
Content-Type: application/json
X-GitHub-Delivery: 17421524-ac8c-4708-b697-e9a2dccb7155
X-GitHub-Event: push
X-Gitea-Delivery: 17421524-ac8c-4708-b697-e9a2dccb7155
X-Gitea-Event: push
X-Gogs-Delivery: 17421524-ac8c-4708-b697-e9a2dccb7155
X-Gogs-Event: push

```

Do note that

1. the value of the job option is case-sensitive (in the example, *gxrest* instead of *gxRest* would fail).
2. HTTPS is needed to send a POST request

At this point clicking on "Test Delivery" button should produce a successful delivery attempt (green checkmark, as shown in the figure).


If the test deliveries fail, try to see if you can POST to Jenkins webhook URL (<http://jenkins.d4science.org/gitea-webhook/post>). E.g. using Postman (<https://www.getpostman.com/>) or with curl:

```
curl -vvv -H "Content-Type: application/json" -H "X-Gitea-Event: push" -X POST http://jenkins.d4science.org/gitea-webhook/post -d "{}"
```

Correct response should be just plain "Processed" string. If you get something else, post it here.

Testing the Gitea/Jenkins round trip

- 1) We push a new commit with a message "Update README.md" in the Gitea repository:


 [gCubeSystem](#) / [gxRest](#)

[Code](#)
[Issues](#)
[Pull Requests](#) **0**
[Releases](#) **1**
[Wiki](#)
[Activity](#)

16 Commits

Branch: master [Commit Graph](#)

16 Commits (master)

Author	SHA1	Message
 Manuele Simi	29b6095d54	Update 'README.md'

2) If the webhook has been properly set up, on the Jenkins interface we should see that the build of the linked project has started:

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Workspace](#)
[Build Now](#)
[Delete Maven project](#)
[Configure](#)
[Modules](#)
[Dependency Graph](#)
[Git Polling Log](#)
[Rename](#)
[Embeddable Build Status](#)

Maven project gxRest

gCube eXtensions to REST

[Workspace](#)
[Recent Changes](#)
[Latest Test Result](#) (no failures)

Disk Usage

Job	454 KB
All builds	454 KB
Locked builds	-
All workspaces	3 MB
Slave workspaces	3 MB
Non-slave workspaces	-

[Latest Test Result](#) (no failures)

Build History

[trend =](#)

find

#2	May 9, 2019 10:16 PM	12 KB
#1	May 9, 2019 3:49 PM	454 KB

[RSS for all](#)
[RSS for failures](#)

Upstream Projects

- [authorization-client](#)
- [maven-parent](#)

3) Clicking on the ongoing build, we can appreciate that it has been triggered by the commit:

Build #2 (May 9, 2019 10:16:26 PM)



Changes

1. Update 'README.md' ([detail](#))



[Started by an SCM change](#)



Revision: 29b6095d54fa403840f95c6e50130b63e7cd5ba1

- refs/remotes/origin/master

Module Builds

[gCube eXtensions to REST with HTTP](#) 11 sec and counting

[gCube eXtensions to JAX-RS](#) Not started yet

[gCube eXtensions to REST](#) 2.5 sec

4) Projects configured downstream will be also built if the build completes successfully.

Back to the CI guide.

Retrieved from "https://gcube.wiki.gcube-system.org/index.php?title=Gitea/Jenkins:_Setting_up_Webhooks&oldid=33034"

Category: Continuous Integration

- This page was last modified on 30 November 2019, at 18:51.