

Gitlab Integration

Table of Contents

1. What can we do currently?	1
2. And what can't we do?	1
3. Configure the integration module in Taiga	2
4. Configure Webhooks in your GitLab repository	2

GitLab is an incredibly powerful open source code collaboration platform, git repository manager, issue tracker and code reviewer. It integrates with issue trackers, continuous integration services and its Enterprise Edition has phenomenal LDAP and Active Directory support.

1. What can we do currently?

Right now you can connect a GitLab repository with a Taiga project and generate a one-way communication (from GitLab to Taiga) to:

- **Change the status of an epic, user story, issue, or task in Taiga with the commit message:** you may want to read more about this feature in the article ["Changing elements status via commit message"](#).
- **Attach commits in an epic, user story, issue, or task of Taiga with the commit message:** you may want to read more about this feature in the article ["Attach commits to elements via commit message"](#).
- **Create issues on Taiga when they are created on GitLab.**
- **Add comments to the connected issues on Taiga when they are created on GitLab.**

In Taiga an issue is connected with a GitLab issue if it was created by the GitLab integration. In the detail page of an issue on Taiga you can see a link under the subject to the original issue in GitLab.

2. And what can't we do?

- **Dual synchronization:** currently the integration functionality only allows receiving messages from GitLab. Taiga can't communicate with GitLab (the one-way communication is from GitLab to Taiga), so changes made in Taiga won't be reflected in GitLab.
- **Show commit links in Taiga issues:** If you name a story, task or issue in a commit message (by its reference number) a link to commit won't appear in Taiga.
- **Sync current GitLab issues and comment when the integration module in Taiga is enabled:** the integration only works with the future issues and comments added in GitLab.

Currently GitLab offers a limitation on its webhooks system that must be considered. It is not

hardcore but it's good to know them to be aware of the limitation that exist.

- GitLab doesn't sign the messages so Taiga only can confirm the origin checking the IPs in the request.

WARNING

Integration will always take time so please be patient. If you need a specific integration and you feel ready to get a machete and get into the Taiga, please review our [API Docs](#) and our section in [Taiga](#) resources about "[How you can contribute?](#)". If you need help with a specific feature, you can always get in touch with our community through our [community space](#).

3. Configure the integration module in Taiga

1. Go to **Admin > Integrations > GitLab**
2. Fill **secret key** or use the auto generated one
3. Copy the **payload URL** input
4. Optionally you can define the valid emitter IPs or IP ranges to validate the origin of the requests

The screenshot shows the Taiga Admin interface. On the left is a sidebar with a navigation menu including 'The Princess Bride', 'Epics', 'Scrum', 'Kanban', 'Issues', 'Search', 'Wiki', 'Team', and 'Settings'. The main content area is divided into two columns: 'PROJECT' and 'WEBHOOKS'. Under 'PROJECT', there are links for 'ATTRIBUTES', 'MEMBERS', 'PERMISSIONS', 'INTEGRATIONS' (highlighted in blue), and 'PLUGINS'. Under 'WEBHOOKS', there are links for 'GITHUB', 'GITLAB' (highlighted in blue), 'BITBUCKET', and 'GOGS'. On the right side, the 'Gitlab' configuration form is displayed. It includes a 'Secret key' field with a generated key '7br5476fhg7b641b2a0857bcdce28d114', a 'Payload URL' field with the URL 'https://api.taiga.io/api/v1/gitlab-hook?project=50493&key=7br5476ft', and a 'Valid source IPs (separated by ,)' field with the text 'Gitlab requests are not signed so the best way of verifying the origin is by IP. If the field'. There is a green 'SAVE' button and a link to the support page.

4. Configure Webhooks in your GitLab repository

Go to your repository

1. Click on **Webhooks**
2. On that screen set the url with the payload url of this screen.
3. Taiga currently listens for two different kind of events from GitLab:
 - a. Push events: changing element status via commit message

b. Comments: issue comments created in GitLab appear automatically in Taiga

c. Issues events: issues created in GitLab appear automatically in Taiga

4. Click on "Add Webhook"

The screenshot shows the GitLab interface for a project named 'the-princess-bride' by user 'Íñigo Montoya'. The left sidebar contains a navigation menu with options: Go to project, Project Settings, Groups, Deploy Keys, Webhooks (highlighted), Git Hooks, Services, Protected Branches, Mirror Repository, Pages, Audit Events, Runners, Variables, Triggers, and a user profile for 'imontoya'. The main content area is titled 'Webhooks' and includes a sub-header: 'Webhooks can be used for binding events when something is happening within the project.' Below this, there is a form to add a new webhook. The 'URL' field contains 'http://example.com/trigger-ci.json'. The 'Trigger' section has several options, each with a checkbox and a description:

- ☒ **Push events**: This url will be triggered by a push to the repository
- ☐ **Tag push events**: This url will be triggered when a new tag is pushed to the repository
- ☐ **Comments**: This url will be triggered when someone adds a comment
- ☐ **Issues events**: This url will be triggered when an issue is created/updated/merged
- ☐ **Merge Request events**: This url will be triggered when a merge request is created/updated/merged
- ☐ **Build events**: This url will be triggered when the build status changes

At the bottom of the form, there is an 'SSL verification' section with a checkbox for 'Enable SSL verification' which is checked. A green 'Add Webhook' button is located at the bottom left of the form area.