

Gogs 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 Gogs repository	2

Gogs is a self-hosted Git service written in Go.

1. What can we do currently?

Right now you can connect a Gogs repository with a Taiga project and generate a one-way communication (from Gogs 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"](#).

2. And what can't we do?

- **Create issues on Taiga when they are created on Gogs.**
- **Add comments to the connected issues on Taiga when they are created on Gogs.**
- **Dual synchronization:** currently the integration functionality only allows receiving messages from Gogs. Taiga can't communicate with Gogs (the one-way communication is from Gogs to Taiga), so changes made in Taiga won't be reflected in Gogs.
- **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 Gogs issues and comment when the integration module in Taiga is enabled:** the integration only works with the future issues and comments added in Gogs.

Currently Gogs 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.

- Gogs 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 > Gogs**
2. Fill **secret key** or use the auto generated one
3. Copy the **payload URL** input

The screenshot shows the Taiga Admin interface. On the left is a sidebar with navigation links: 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, BITBUCKET, and GOGS (highlighted in blue). On the right, the 'Gogs' configuration form is displayed. It includes a 'Secret key' field with a generated key '2f18d54ghj171407686bfeebd0f5bb25c', a 'Payload URL' field with the URL 'https://api.taiga.io/api/v1/gogs-hook?project=50493', and a 'SAVE' button. At the bottom of the form, there is a link: 'Do you need help? Check out our support page!'.

4. Configure Webhooks in your Gogs repository

Go to your repository

1. Click on **Settings > Webhooks > Add webhook**
2. On the form set the **Payload URL** and the **Secret** with the payload URL and Secret key from Taiga.
3. Taiga only listen for push events (changing element status via commit message) in the case of gogs.

Settings

Options

Collaboration

Webhooks

Add Webhook

Hook Type*

Gogs

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [Webhooks Guide](#).

Payload URL*

http://api.taiga.io/api/v1/gogs-hook?project=X

Content Type*

application/json

Secret

.....

Which events would you like to trigger this webhook?

Just the push event.

Active

☒

We will deliver event details when this hook is triggered.

Add Webhook

3