tuleap

WHAT IS TULEAP SOLUTIONS GET STARTED COMMUNITY RESOURCES BLOG

Search docs

HOW-TO GUIDES

⊟ USER GUIDE
Introduction
User
Writing in Tuleap
Project Management
Backlog
Kanban
Program management
Baseline
Trackers
⊟ Source Code Management
⊟ Code versioning
Git
Subversion
GitLab
Code review
Continuous Integration
Test Management
Documents and files
Integration
OAuth2 and OpenIDConnect
Tuleap Query Language (TQL)
Miscellaneous
Legal Notice

INSTALLATION GUIDE

ADMINISTRATION GUIDE

DEPLOYMENT GUIDE

DEVELOPER GUIDE

Docs » User guide » Source Code Management » Code versioning » GitLab

GitLab

Overview

The purpose of this integration is to link Tuleap projects and GitLab repositories.

If you are using GitLab and want to keep a trace of your commits and merge requests in Tuleap, this plugin will allow you to reference Tuleap artifacts in your commit messages or merge requests title/description/branch source name and conversely.

References

A GitLab commit, tag or merge request can reference several Tuleap artifacts, in different projects. A GitLab branch name can only reference one Tuleap artifact.

A Tuleap artifact can reference several commits, tags, branches or merge requests, in different GitLab repositories.

REFERENCE A TULEAP ARTIFACT

To be able to create GitLab cross-references, you need to:

- Register your GitLab repository in the Git service of your Tuleap project
- Reference Tuleap artifacts in GitLab commit messages
- Reference Tuleap artifacts in GitLab merge request title, description or source branch name
- Reference Tuleap artifacts in GitLab tag message
- Reference Tuleap artifacts in GitLab branch name

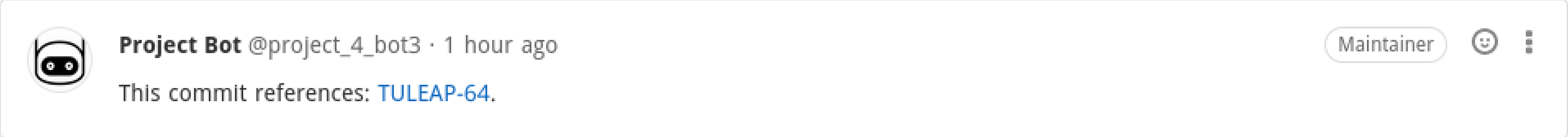
To link your commit (or merge request or tag or branch) to the Tuleap artifact of your choice, you must add the keyword `TULEAP-<artifact_id>` (case-sensitive) to your commit message (or merge request title, description or source branch name). You can reference as many artifacts as you want in the same commit message (or merge request title/description or tag message). You can reference only one artifact in a branch name or in the source branch name of a merge request.

REFERENCE A TULEAP ARTIFACT IN GITLAB COMMIT

When the committer email is matching a Tuleap account, then its avatar and username will be displayed in the reference. Otherwise, the committer name is displayed as received from the GitLab API.

On GitLab side, when a commit message contains some references to Tuleap, then a comment is automatically added to the commit.

A comment is composed of a list of Tuleap references included in the commit message, with links to Tuleap.


Project Bot @project_4_bot3 · 1 hour ago

This commit references: TULEAP-64.

REFERENCE A TULEAP ARTIFACT IN GITLAB MERGE REQUEST

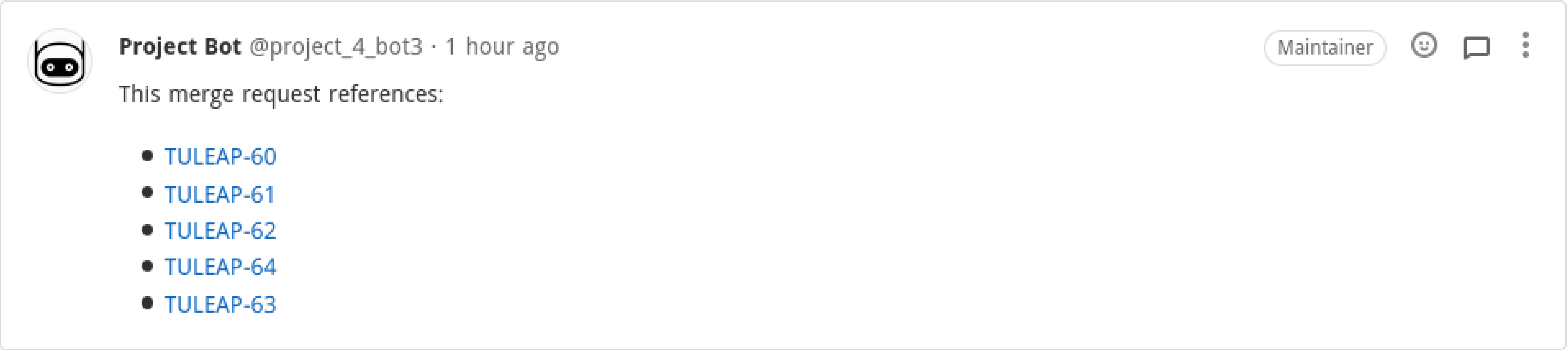
When the public email of the author of the Merge Request is matching a Tuleap account, then its avatar and username will be displayed in the reference. Otherwise, the author name is displayed as received from the GitLab API.

> Note

> If the author of the Merge Requests changes its public email after creating the Merge Request, then no refresh will be performed to update data shown in the reference on Tuleap.

On GitLab side, when a merge request contains some references to Tuleap, then a comment is automatically added.

A comment is composed of a list of Tuleap references included in the merge request title, description or source branch name, with links to Tuleap.



Project Bot @project_4_bot3 · 1 hour ago

This merge request references:

- TULEAP-60
- TULEAP-61
- TULEAP-62
- TULEAP-64
- TULEAP-63

REFERENCE A TULEAP ARTIFACT IN GITLAB TAG MESSAGE

The Tuleap reference will display the tag name, the tag message and the associated commit SHA-1.

> Note
>
> If the tag is moved, all previous information will be deleted on Tuleap side and recreated with new ones.

On GitLab side, no backlink can be added because it’s not possible to add comments on tags.

REFERENCE A TULEAP ARTIFACT IN GITLAB BRANCH NAME

The Tuleap reference will display the branch name, the associated commit SHA-1 and the last push date received for this branch.

> Note
>
> If the branch is removed, all previous information will be deleted on Tuleap side. Only one artifact can be referenced in the branch name. If multiple references are found, only the first one will be taken into account.

On GitLab side, no backlink can be added because it’s not possible to add comments on branch.

REFERENCE A GITLAB COMMIT OR MERGE REQUEST

Please refer to Tuleap references for more details on references.

You can reference a commit or merge request of one of the GitLab repositories registered in your Tuleap project.

REFERENCE A GITLAB COMMIT IN TULEAP

To reference GitLab commit, you have to use the keyword `gitlab_commit` followed by a `#`, the repository name, and the commit sha1:

`gitlab_commit #<repository_name>/<sha1>`.

`<repository_name>` must be a registered GitLab repository. If not, no reference will be created.

When you click on this reference, you will be redirected to your GitLab instance, on the page displaying the commit details.

REFERENCE A GITLAB MERGE REQUEST IN TULEAP

To reference GitLab merge request, you have to use the keyword `gitlab_mr` followed by a `#`, the repository name, and the merge request id:

`gitlab_mr #<repository_name>/<merge_request_id>`

`<repository_name>` must be a registered GitLab repository. If not, no reference will be created.

When you click on this reference, you will be redirected to your GitLab instance, on the page displaying the merge request details.

REFERENCE A GITLAB TAG IN TULEAP

To reference GitLab tag, you have to use the keyword `gitlab_tag` followed by a `#`, the repository name, and the tag name:

`gitlab_tag #<repository_name>/<tag_name>`

`<repository_name>` must be a registered GitLab repository. If not, no reference will be created.

When you click on this reference, you will be redirected to your GitLab instance, on the page displaying the commit tagged with `<tag_name>`.

REFERENCE A GITLAB BRANCH IN TULEAP

To reference a GitLab branch, you have to use the keyword `gitlab_branch` followed by a `#`, the repository name, and the branch name:

`gitlab_branch #<repository_name>/<branch_name>`

`<repository_name>` must be a registered GitLab repository. If not, no reference will be created.

When you click on this reference, you will be redirected to your GitLab instance, on the page displaying the commit details of the last commit in the branch `<branch_name>`.

CLOSE TULEAP ARTIFACT WITH GITLAB COMMIT MESSAGES

When integrating a GitLab repository in a Tuleap project, there is an option (disabled by default) that allows to close artifacts in this project based on GitLab commit messages. If a commit message containing a closing pattern is pushed in a GitLab repository integrated in a Tuleap project and this integration has activated this feature, then the artifact will be closed :

- If the artifact is in the same project of the integration
- If a done semantic or a status semantic is defined for this artifact
- If the artifact is not yet closed
- If the push occurs in the GitLab repository’s default branch

The following keywords (case insensitive) can be used to close an artifact:

- `Closes` TULEAP-1234
- `Resolves` TULEAP-1234
- `Fixes` TULEAP-1234
- `Implements` TULEAP-1234

Some variations of these keywords are handled:

- `Close` / `Fix` / `Resolve` / `Implement`
- `Closes` / `Fixes` / `Resolves` / `Implements`
- `Closed` / `Fixed` / `Resolved` / `Implemented`
- `Closing` / `Fixing` / `Resolving` / `Implementing`

The artifact will be closed by a Tuleap bot named `Tracker Workflow Manager` and a comment will be added to explain why it has been closed. The first done value retrieved by Tuleap will be used to close the artifact. If there is no done semantic defined, then the first closed value will be used.

CREATE GITLAB BRANCHES FROM TULEAP ARTIFACT

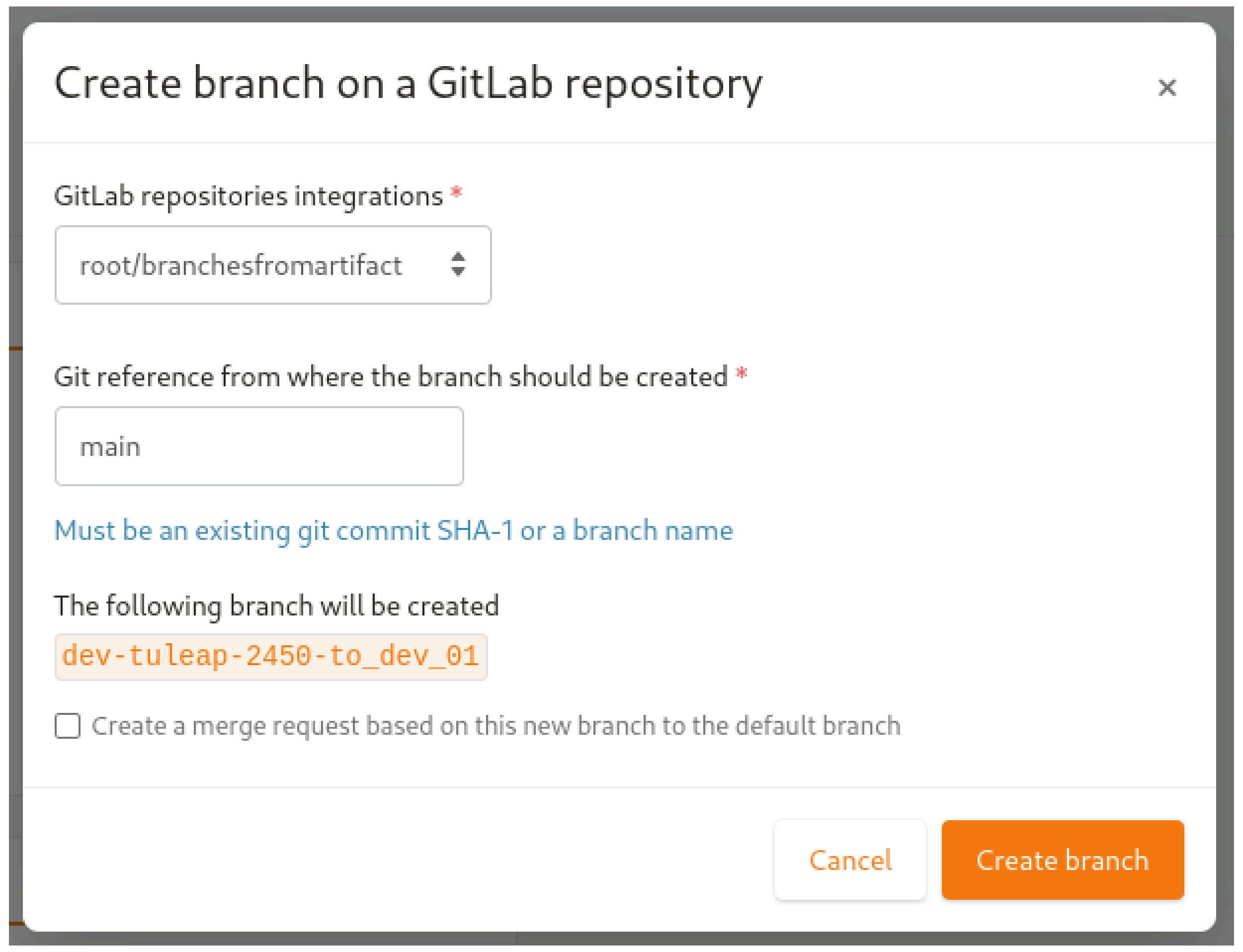
If some preconditions are met, user can be able to create a GitLab branch and the corresponding merge request directly from a Tuleap artifact. Those preconditions are:

- At least one GitLab repository is integrated in the artifact’s project with token well configured
- User is a project member
- User can view the artifact

The branch name cannot be chosen. It follows the pattern `(prefix)tuleap-{artifact_id}(-slugified artifact title)` with:

- `prefix` that can be defined for each integration
- `tuleap-{artifact_id}` to create a cross-reference between the branch and the artifact
- `slugified artifact title` if artifact has a semantic title defined and an associated value

This action can be performed through the artifact action button.



Create branch on a GitLab repository

GitLab repositories integrations *

root/branchesfromartifact

Git reference from where the branch should be created *

main

Must be an existing git commit SHA-1 or a branch name

The following branch will be created

`dev-tuleap-2450-to_dev_01`

□ Create a merge request based on this new branch to the default branch

Register your GitLab repository

PREREQUISITES

To be able to register a GitLab repository in your project, please ensure that:

- both Git and GitLab plugins are installed and activated.
- you have administrator privileges in the Git service of your Tuleap project.
- you have a GitLab access token that authorizes querying the GitLab API (see GitLab access Token).

GITLAB ACCESS TOKEN

You can use a personal, a group or a project access token. The token will be used to manage integration of GitLab repository in Tuleap, and to write comments automatically on GitLab commit or merge requests.

With a group access token, you can only integrate the GitLab repositories of the corresponding group. With a project access token, you can only integrate the GitLab repository which provides the project access token. With a personal access token, you can integrate all repositories that you maintain. Don’t forget that comments will be added automatically on GitLab commits and merge requests. These comments will be written with the access token, so if you provide a personal access token, the user providing this token will be displayed next to comments.

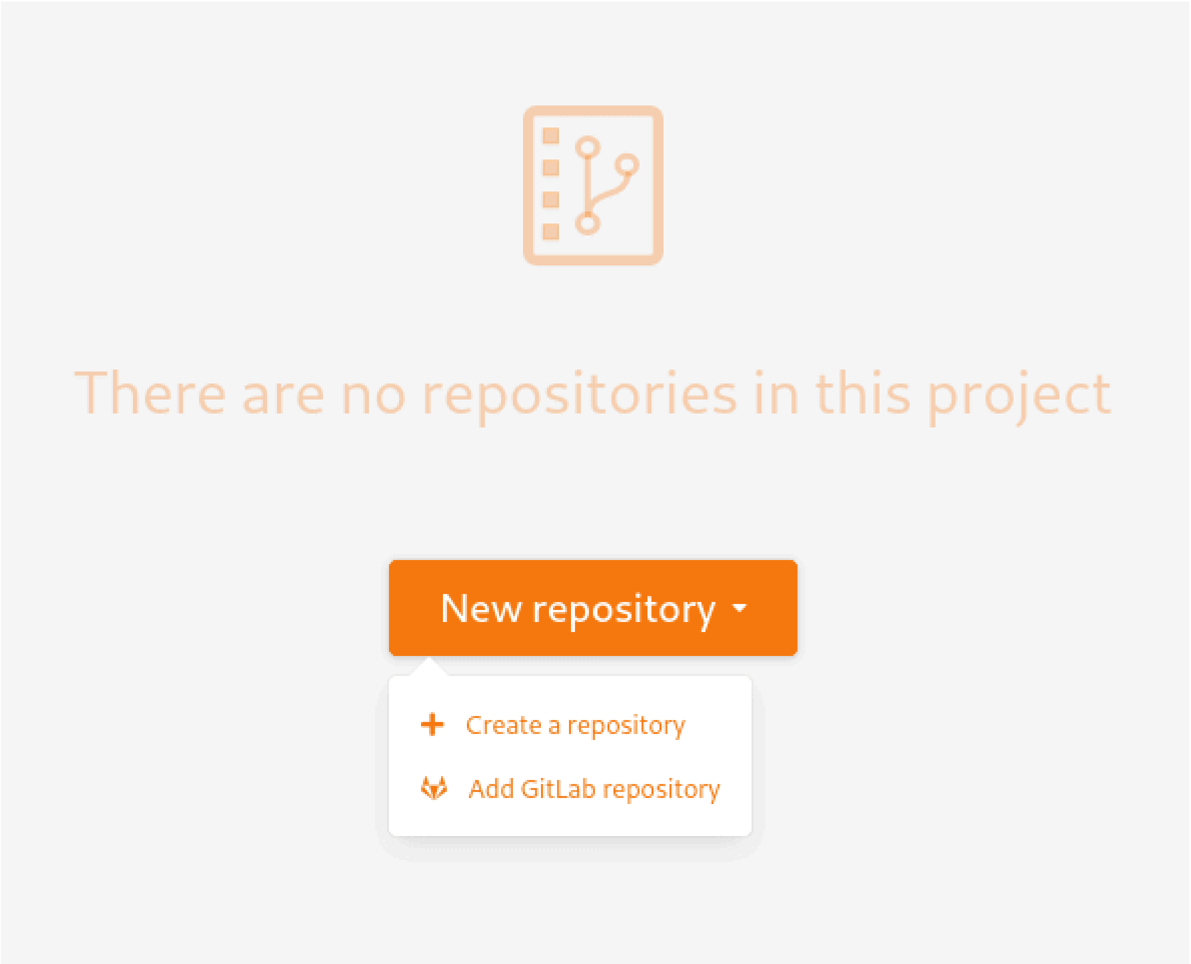
> Note
>
> If you use a personal access token, you need to be identified by GitLab as the maintainer of the repository that you want to integrate.

> Note
>
> The name of the token is not important, but you need at least to check `api` in the scopes list.

Once your GitLab access token is created, copy it and save it for later.

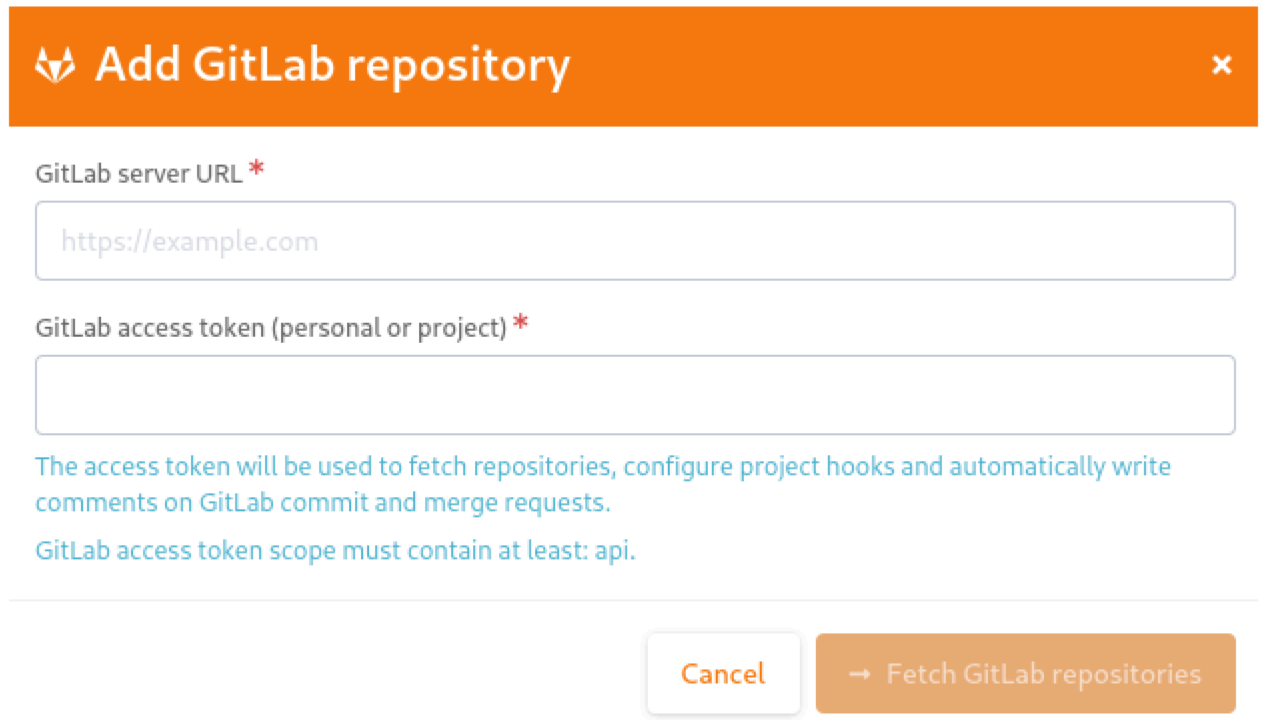
GITLAB REPOSITORY REGISTRATION

Go to the Git service of your Tuleap project, click on [New repository], then click on [Add GitLab repository].



There are no repositories in this project

In the modal, provide the URL of your GitLab instance and the GitLab access token.



GitLab server URL *

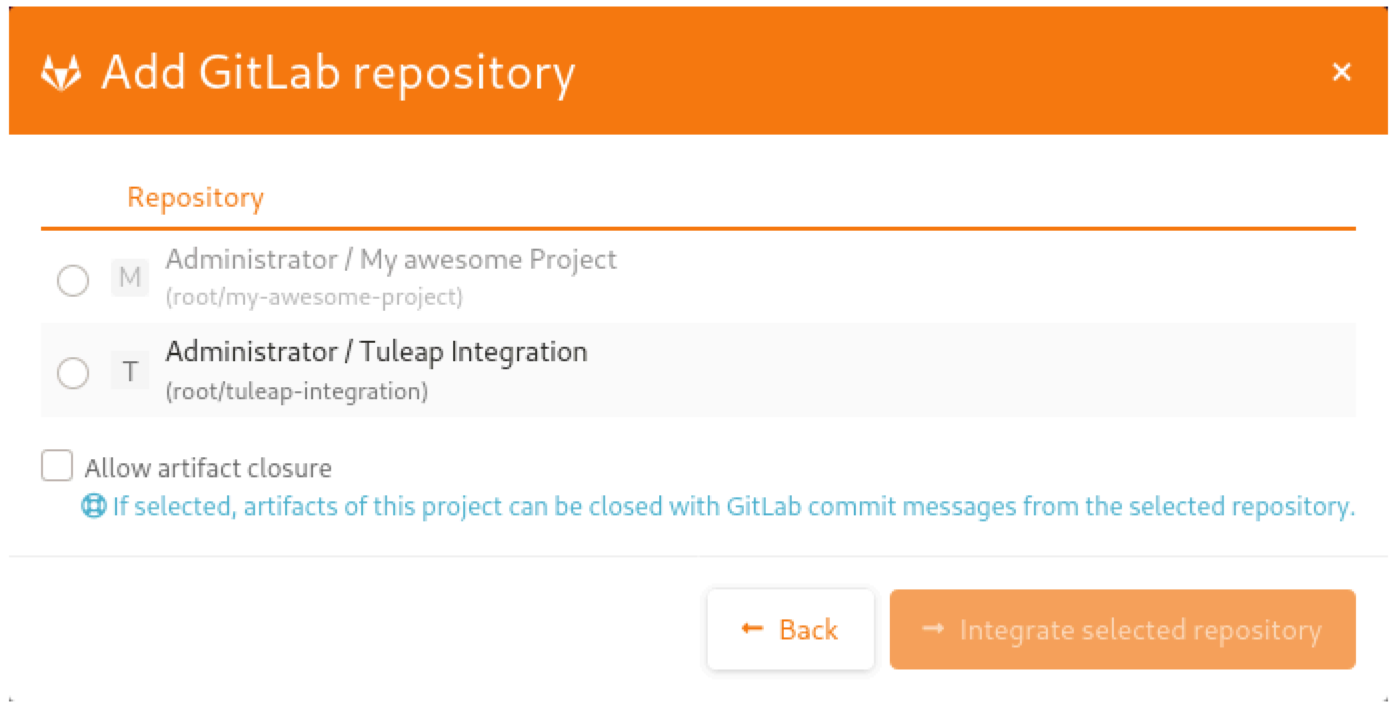
https://example.com

GitLab access token (personal or project) *

The access token will be used to fetch repositories, configure project hooks and automatically write comments on GitLab commit and merge requests.

GitLab access token scope must contain at least: api.

The list of the repositories that you can integrate is displayed. Select the repository to link.



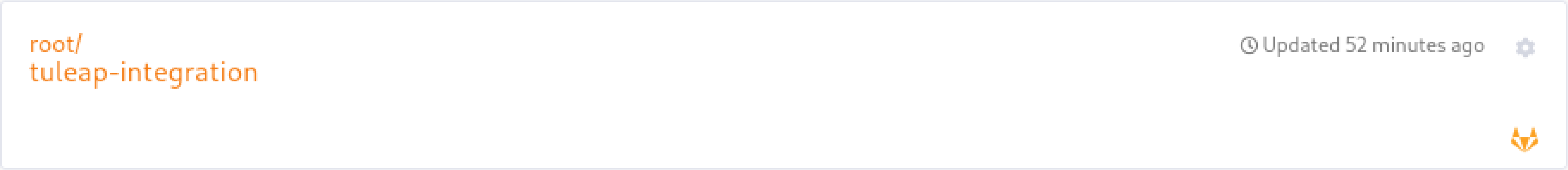
○ M Administrator / My awesome Project (root/my-awesome-project)

○ T Administrator / Tuleap Integration (root/tuleap-integration)

□ Allow artifact closure

If selected, artifacts of this project can be closed with GitLab commit messages from the selected repository.

Once the GitLab repository is registered, it is displayed in the repositories list and is visually identifiable thanks to the GitLab icon.



From now on, each time you reference an artifact in a commit or merge request, a cross-reference will be created in the target artifact.

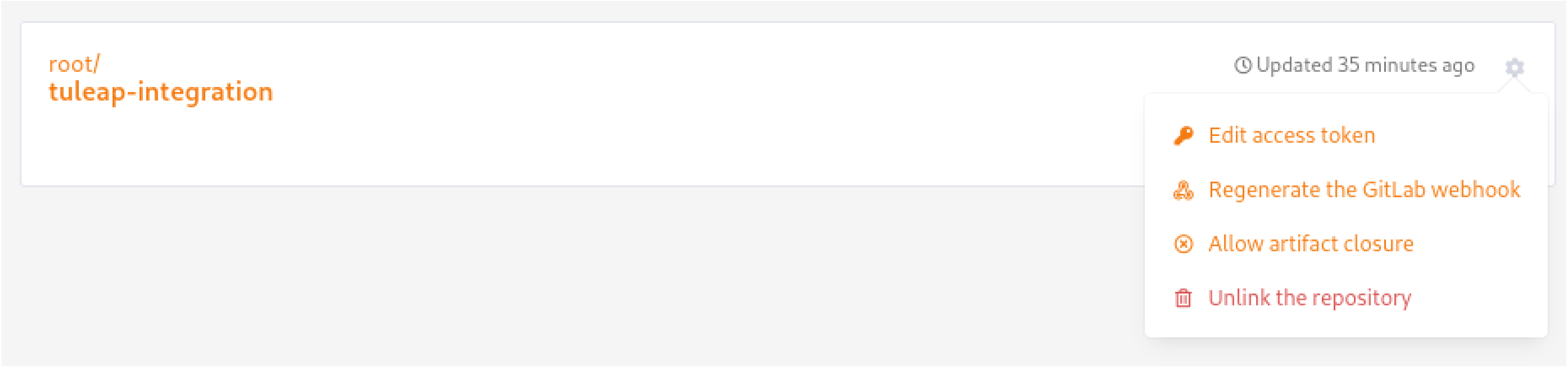
> Note
>
> During the registration, a webhook is created in the GitLab repository. If the parameters of this webhook change (URL, events, or anything else), we cannot ensure that cross-references will continue to be created. See Regenerate the GitLab webhook to have more details.

POSSIBLE ACTIONS ON GITLAB REPOSITORY

As a Git administrator, go to the Git service of your project and find the repository that you want to apply action.

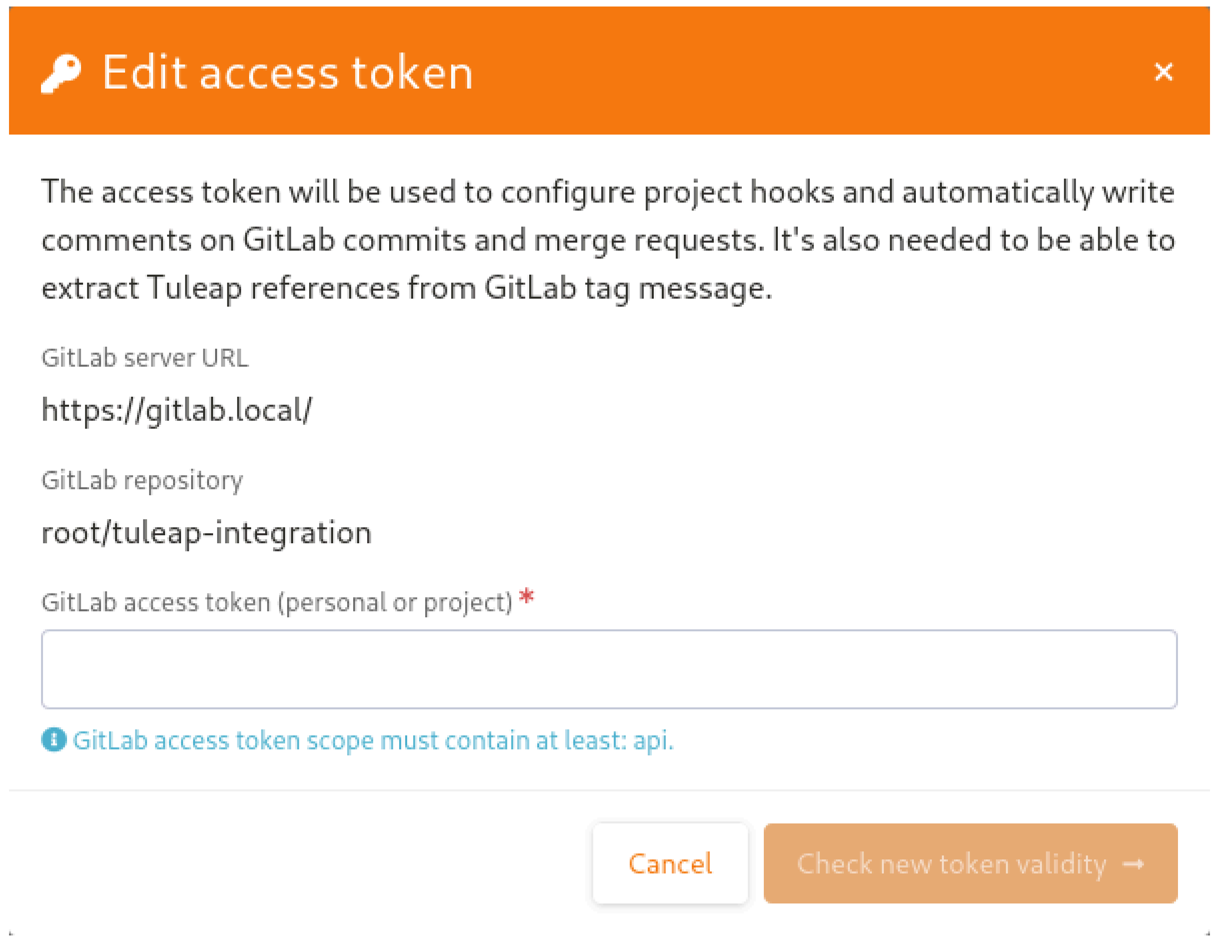
When you click on cog icon in GitLab tile, you can:

- Edit access token
- Regenerate the GitLab webhook
- Allow artifacts closure option
- Edit create branch prefix
- Unlink the repository



EDIT ACCESS TOKEN

If the token used during the integration has been revoked, you can change it by clicking on [Edit access token]. See GitLab access Token to have more details.



Edit access token ×

The access token will be used to configure project hooks and automatically write comments on GitLab commits and merge requests. It's also needed to be able to extract Tuleap references from GitLab tag message.

GitLab server URL

https://gitlab.local/

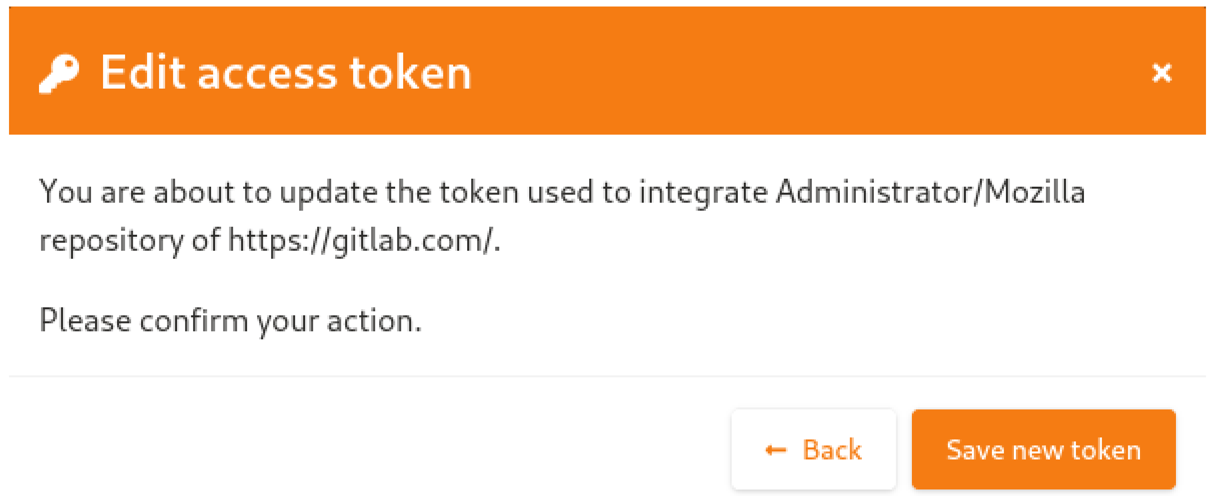
GitLab repository

root/tuleap-integration

GitLab access token (personal or project) *

GitLab access token scope must contain at least: api.

Confirm the action.



Edit access token

You are about to update the token used to integrate Administrator/Mozilla repository of https://gitlab.com/.

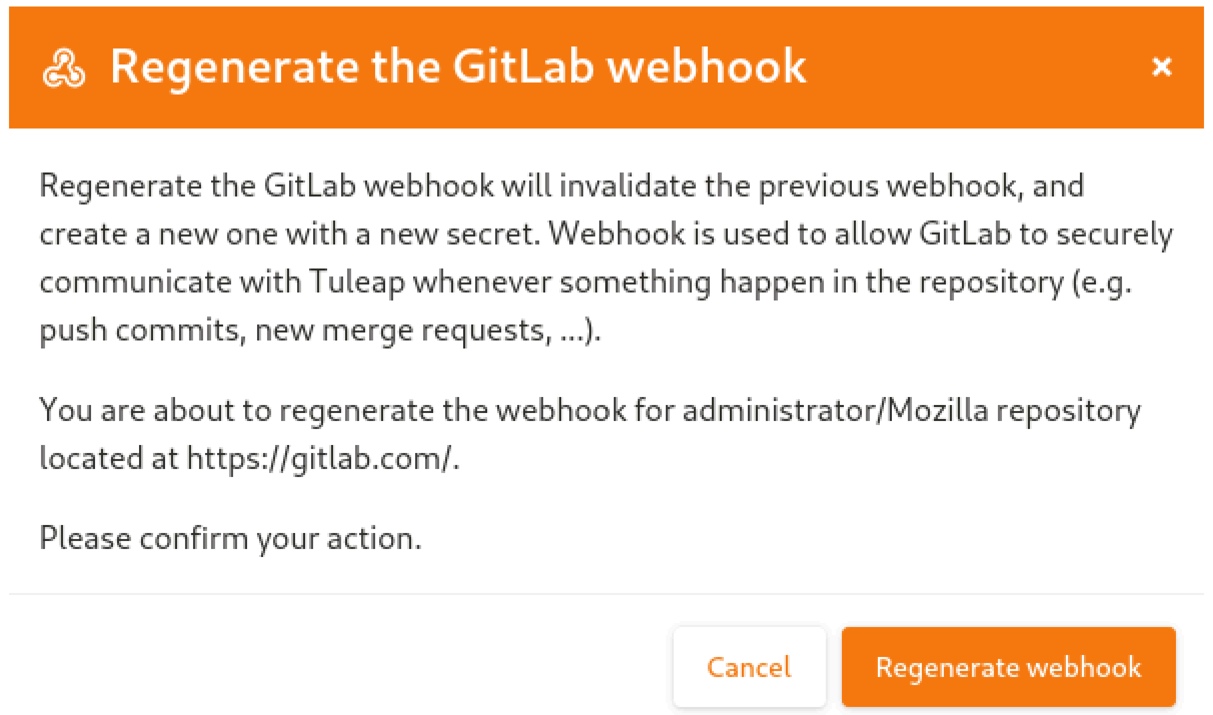
Please confirm your action.

Note

When you change access token, the webhook is also regenerated on GitLab side.

REGENERATE THE GITLAB WEBHOOK

A webhook allows GitLab to communicate with Tuleap. This webhook is composed of a secret generated automatically by Tuleap and some events (push and merge requests events). If the webhook has been changed and is not functional, you can regenerate it by clicking on [Regenerate GitLab webhook].



Regenerate the GitLab webhook

Regenerate the GitLab webhook will invalidate the previous webhook, and create a new one with a new secret. Webhook is used to allow GitLab to securely communicate with Tuleap whenever something happen in the repository (e.g. push commits, new merge requests, ...).

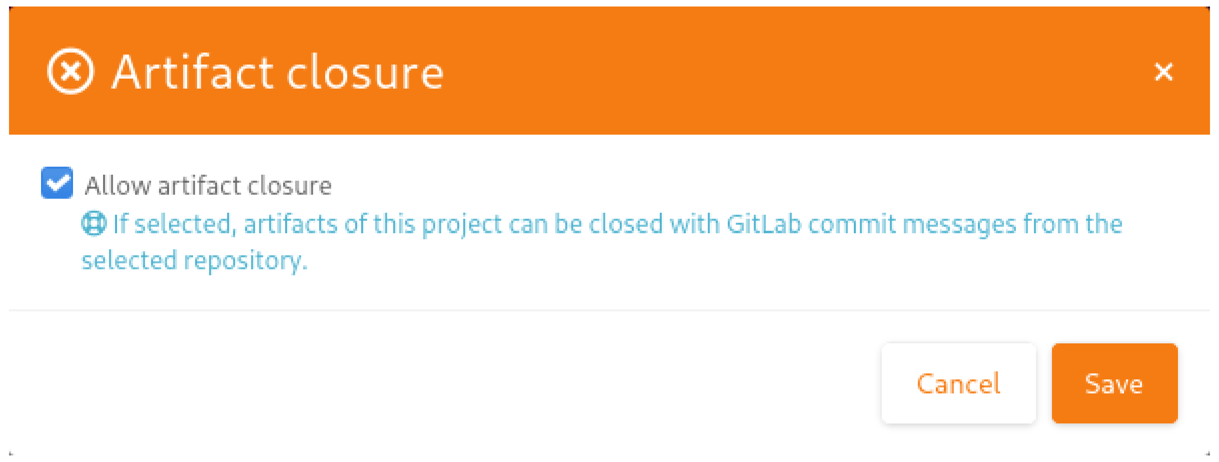
You are about to regenerate the webhook for administrator/Mozilla repository located at https://gitlab.com/.

Please confirm your action.

When the webhook is regenerated, the old one is deleted from GitLab server, and a new webhook with a new secret is created.

ALLOW ARTIFACTS CLOSURE OPTION

This option can be selected at repository integration. It can also be edited by any Git administrator.

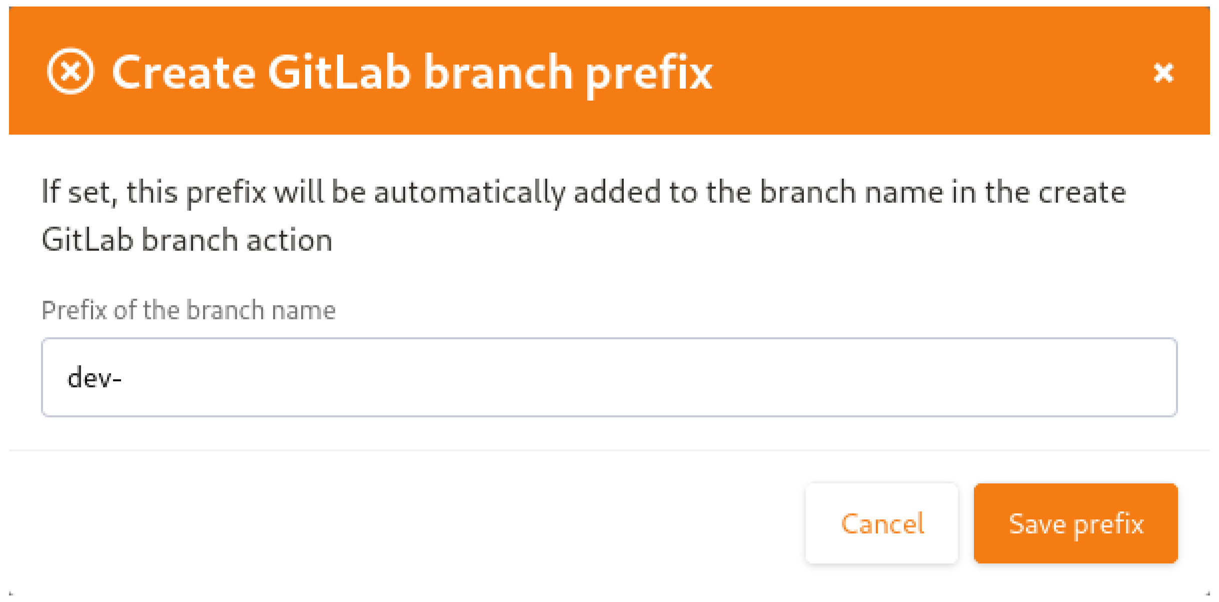


Artifact closure

☑ Allow artifact closure
If selected, artifacts of this project can be closed with GitLab commit messages from the selected repository.

EDIT CREATE BRANCH PREFIX

This option allows to edit the prefix used in the branch name while creating a GitLab branch from a Tuleap artifact. It's empty by default. It can be edited by any Git administrator.



Create GitLab branch prefix

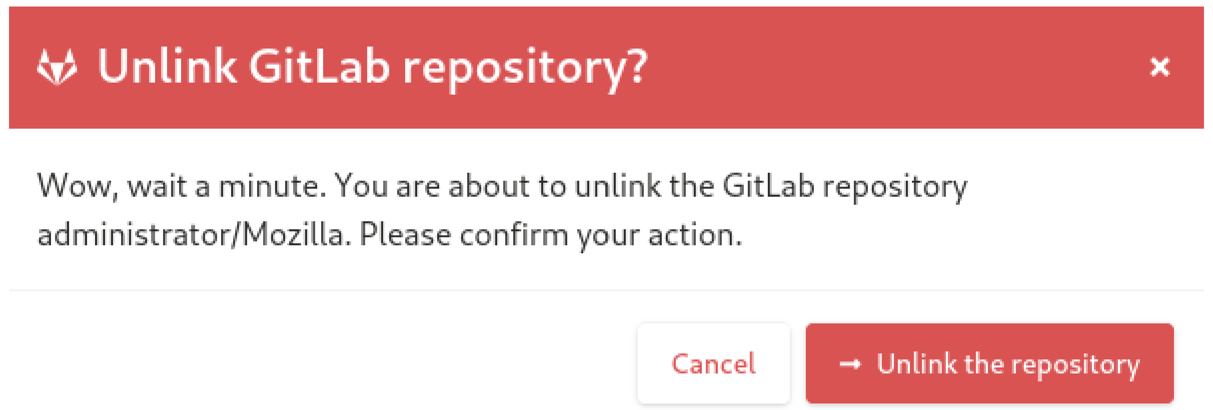
If set, this prefix will be automatically added to the branch name in the create GitLab branch action

Prefix of the branch name

dev-

UNREGISTER REPOSITORIES

If you want to unregister a repository, you need to select [Unlink the repository] in the list. Then a modal opens and you need to confirm the action.



Unlink GitLab repository?

Wow, wait a minute. You are about to unlink the GitLab repository administrator/Mozilla. Please confirm your action.

From now on, existing references will be removed and any new commit in this repository referencing a Tuleap artifact in this project will not create cross-references.

> Attention
>
> Known issues / limitations
>
> - If you already have a project reference named `gitlab_commit`, `gitlab_mr`, `gitlab_tag` or `gitlab_branch`, it will override the one used by the plugin.
> - GitLab provides two names for a repository:
>   - `name_with_namespace` is displayed in UI
>   - `path_with_namespace` is used to clone/checkout the repository
>   - Tuleap displays only `path_with_namespace` and references are created with it.
> - Two repositories with the same name and path from two different GitLab instances cannot be integrated in the same Tuleap project.
> - For the moment, the project name and namespace of your GitLab project must **not** contain a “-” or a “.”.

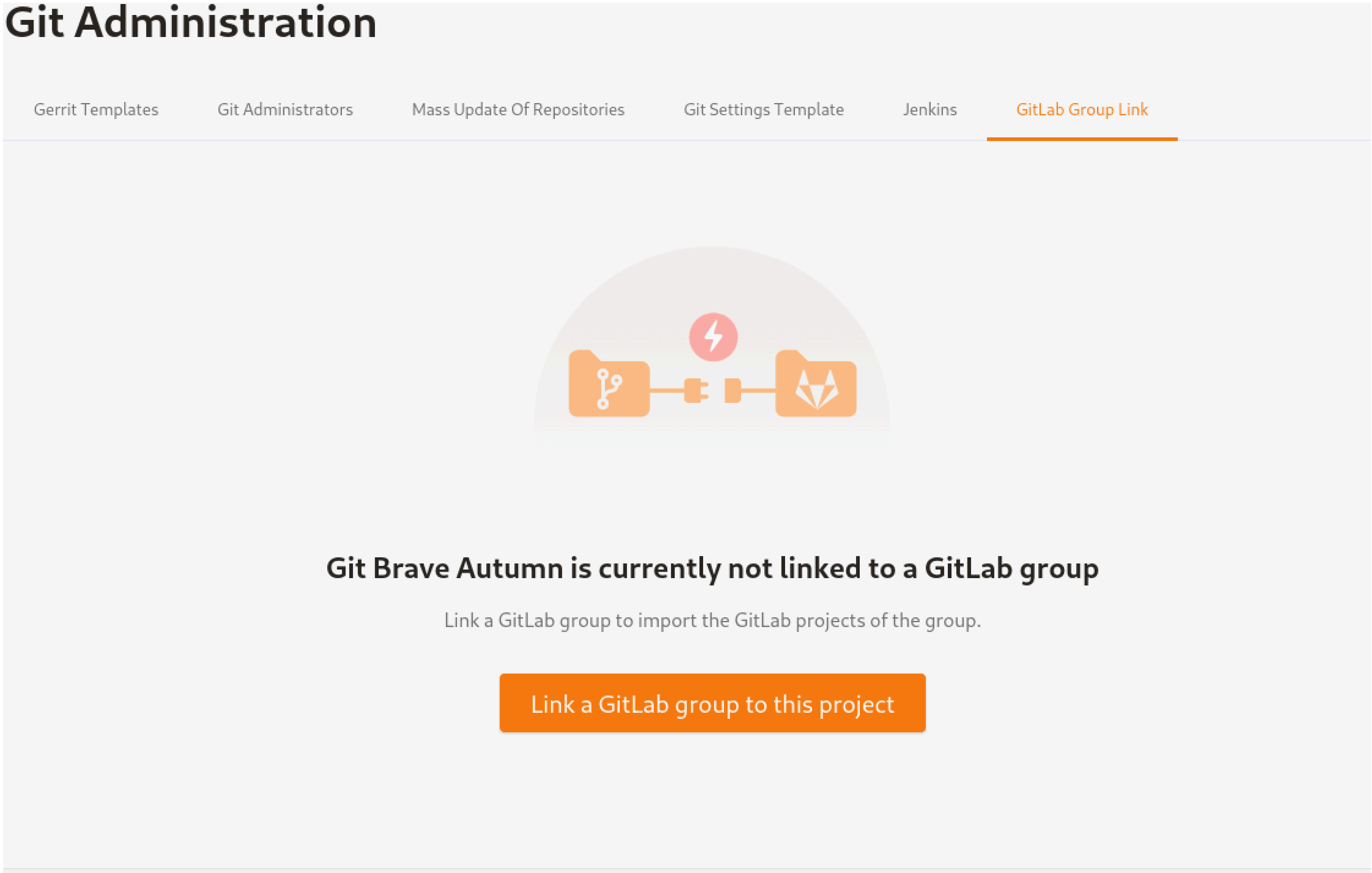
Link a Tuleap project to a GitLab group

PREREQUISITES

To be able to link a GitLab group to your Tuleap project, please make sure that:

- both Git and GitLab plugins are installed and activated.
- you have administrator privileges in the Git service of your Tuleap project.
- you have a GitLab access token that authorizes querying the GitLab API (see GitLab access Token).

CREATE THE LINK TO THE GITLAB GROUP



Git Administration

Git Brave Autumn is currently not linked to a GitLab group

Link a GitLab group to import the GitLab projects of the group.

Click on the button to start the wizard

Enter the URL of your GitLab instance and the GitLab access token. Then in the list of visible groups, select the group to link. A Tuleap project may only be linked to one GitLab group at a time.

Select a GitLab group

Then, configure the settings that will be applied for each GitLab repository in the chosen group: whether to allow closing Tuleap Artifacts and the branch prefix used when creating a GitLab branch from a Tuleap artifact. Both settings are optional.

> Note
>
> The settings here, only apply at the initial synchronization of each repository with Tuleap. They are not locked down and can be changed individually for each repository after their synchronization.

Choose settings that will be applied for each newly synchronized repository

When you link the group, all the repositories of that group will be integrated in the Tuleap project. They will be visible in the list of Git repositories of the project.

Once linked, you can do the following actions:

- Synchronize the GitLab group
- Update the access token
- Edit configuration
- Unlink the GitLab group

SYNCHRONIZE THE GITLAB GROUP

Synchronize the repositories of the GitLab group with the project

Tuleap will fetch the list of repositories in the group and will integrate all the new repositories since last synchronization.

> Note
>
> If a GitLab repository has been deleted or moved to another group since last synchronization, it will NOT be removed on Tuleap side. It will stay in the same project.

UPDATE THE ACCESS TOKEN

If the access token used to link the GitLab group has been revoked, you can change it by clicking on the [Update token] button. See GitLab access Token for more details.

EDIT CONFIGURATION

You can change the settings of the group: whether to allow closing Tuleap Artifacts and the branch prefix used when creating a GitLab branch from a Tuleap artifact.

> Note
>
> The settings here only apply when synchronizing new repositories. Changing them here will NOT change them for any repository previously synchronized. Furthermore, those settings are not locked down and can be changed individually for each repository after their synchronization.

UNLINK THE GITLAB GROUP

This will remove the link between the Tuleap project and the GitLab group.

> Note
>
> This will not delete the group on GitLab side. It only removes the link between the Tuleap project and the GitLab group. All GitLab projects part of the group will stay integrated with Tuleap, their configuration will not change. If you wish to remove the integrations of the GitLab projects part of this group, you must remove each one manually.

Previous

Next

Trademarks Security Legal