

Contributing to GitLab: Protected Packages

Gerardo Navarro (B310 Digital)



CODE

I'LL PROTECT YOU

Contributing on behalf of Siemens



**code
siemens**

Bug Reports
Feature Requests



**GitLab
Contributor**

Release
Community Edition



Upstream
Contribution



GitLab



Step #1: Issue

- Use Google and GitLab to find existing issues or epics
- Understand and research the scope of the issue
- Contribute your opinion, thoughts and questions to the discussions

Identify packages as protected to prevent accidental deletion or updates

Open Epic created 3 years ago by Tim Rizzi

Context

You can use the GitLab Package Registry to publish and store packages right alongside your source code and pipelines. A Developer in the respective project may publish new versions of a package using GitLab CI or the command line. However, similar to protected branches, there are many cases where only a Project Admin should be allowed to update or delete a given package.

The following features are already available to protect the release process:

- [Protected Branches](#)
- [Protected Tags](#)
- [Protected Environments](#)
- [Protecting Pipeline Settings by hosting .gitlab-ci.yml in a separate project](#)

However, there is no way to do the same for packages. This means that we are protecting all of the components responsible for creating a package but not the output.

Proposal

For each package manager format add configuration options for restricting all actions except read (create, update and delete) similar to the existing [Protected Branches](#), [Protected Tags](#) and [Protected Environments](#).



Step #2: Collaborate

- Express your desire to contribute
- Discuss scope of issue, feature or epic
- Propose implementation plan and next steps

Gerardo Navarro @gerardo-navarro · 11 months ago

Hi @trizzi


I would like to work on this epic (&5574) or issue (gitlab#18984 (closed)). Given all the comment threads related to this, I think it makes sense to discuss and define an implementation plan in order to sync and streamline our work. What do you think?

- Add specific feature flag => [gitlab#413641](#) => MR
- Add section in project setting to select and protected packages
- Add db migration for protected packages (possibly background db migration of existing data)
- Possibly adjust policies to handle protected packages
- Implement the handling of protected packages the [different package types \(e.g. Composer, Conan, Maven, ...\)](#), see existing child issues
- Documentation
- Rollout feature flag
- Clean up feature flag

Edited 11 months ago by Gerardo Navarro

2 ❤️ 😊

▼ Collapse replies

 **Tim Rizzi** @trizzi · 11 months ago Author

[@gerardo-navarro](#) That's great to hear. Do you think it would be helpful to have a short video call with one of the engineers at GitLab to review your implementation plan and talk through any expected pitfalls?



Step #3: Proof of Concept

- Understand necessary effort, expected feature scope and upcoming design decisions
- Focus on essentials functions
- Ask for feedback early on and start a discussion

Draft: Protected npm package in package registry [POC] Edit Code ▾ ⋮

🔒 Closed Gerardo Navarro requested to merge [gitlab-community/gitlab:...](#) into [master](#)
11 months ago

[Overview](#) 6 [Commits](#) 2 [Pipelines](#) 6 [Changes](#) 8

What does this MR do and why?

This MR will not be merged. The intention is to discuss and document design decision and implementation details.

This MR wants to provide a POC implementation for the [Identify packages as protected to prevent accid... \(&5574\)](#).

What do we want to achieve with this POC?

- Better understanding of implementation approach and complexity
- Better estimate of implementation plan
- Discuss the data model





What do we **not** want to achieve with this POC?

- Discuss necessary ux / ui changes in regards to the project settings
- Investigate the implementation of dependency types other than `npm`

🔧 with ❤️ at Siemens



Step #4: Implement

Protected packages				Add protection rule
Name pattern	Type	Push protected up to access level	Actions	
@siemens/open-source-at-siemens-prod-*	npm	Owner 		
@siemens/open-source-at-siemens	npm	Owner 		

```
npm notice Publishing to http://gdk.test:3000/api/v4/projects/34/packages/npm/ with tag latest and default access
npm ERR! code E403
npm ERR! 403 403 Forbidden - PUT http://gdk.test:3000/api/v4/projects/34/packages/npm/@siemens%2fopen-source-at-siemens-prod-variant-1 - Package protected.
npm ERR! 403 In most cases, you or one of your dependencies are requesting
npm ERR! 403 a package version that is forbidden by your security policy, or
npm ERR! 403 on a server you do not have access to.
```



Hint #1: Communication

- Use screenshot, screencasts or other visuals
- Apply text formatting for visual structure
- Include “Call-To-Action”, e.g. question

- Add helpful (necessary) context or background information

The screenshot shows a GitHub discussion thread. The top comment is by Marcel van Remmerden (@mvanremmerden), a Developer, posted 6 days ago. The comment text is: "Resolved 6 days ago by Marcel van Remmerden @gerardo-navarro I'm generally trying to understand some of the previous design choices that were made, as it seems like it's partially inspired by the UI of the 'Protected branches' and 'Protected tags' table, but with some differences, and I'm not sure if each of these are intentional:". To the left of this comment are three green checkmark icons. The first checkmark is next to a list of three bullet points: "Different action text ('Unprotect' vs. 'Delete')", "Different permission level text ('Allowed to push' vs. 'Push protected up to access level')", and "Alignment of text (Vertically centered in 'Protected branches' vs. 'Top aligned' in 'Protected packages')". The second checkmark is next to a question: "Any chance you could give me some background on these aspects?". Below this is a "Collapse replies" button. The second comment is by Gerardo Navarro (@gerardo-navarro), the Author, posted 6 days ago. The comment text is: "@mvanremmerden These questions are totally valid." To the left of this comment is a green checkmark icon. Below this comment is a large green-bordered box containing the text: "Background information: The feature 'Protected containers' (and 'Protected packages') is currently under active discussion and development. Therefore, this feature is currently behind a feature flag that will not be activated for gitlab.com. So, things are still moving and changing to some extent." Below this box is a vertical line and the text: "Different action text ('Unprotect' vs. 'Delete')". At the bottom of the thread, there is a paragraph: "Our goal is to align the wording between 'Protected containers' and 'Protected branches'. This is an open todo for the existing CRUD feature 'Protected packages', see #413641 (comment 1770699822). I will adjust the wording accordingly for this specific MR, see https://gitlab.com/gitlab-org/gitlab/-/merge_requests/146436/diffs?commit_id=39b2845da9cfe4c2d836660dbd4d13657c3801c9&pin=7c21745545099a19bd5fa2a9f25d8c660425db32#7c21745545099a19bd5fa2a9f25d8c660425db32_44_43."



Hint #2: Communication

- Assume positive intent in every interaction
- Friendly, responsible and inclusive communication is key
- @Mention / Ping reviewers or collaborators directly to get their attention
- Quote previous questions in your answer
- Use links to directly reference resources



Marcel van Remmerden @mvanremmerden · 6 days ago Developer ✓ 😊 ↶ ⋮

Resolved 6 days ago by Marcel van Remmerden

✓ @gerardo-navarro I'm generally trying to understand some of the previous design choices that were made, as it seems like it's partially inspired by the UI of the "Protected branches" and "Protected tags" table, but with some differences, and I'm not sure if each of these are intentional:

- Different action text ("Unprotect" vs. "Delete")
- Different permission level text ("Allowed to push" vs. "Push protected up to access level")
- Alignment of text (Vertically centered in "Protected branches" vs. "Top aligned" in "Protected packages")

Any chance you could give me some background on these aspects?

✓ Collapse replies

Gerardo Navarro @gerardo-navarro · 6 days ago Author Contributor 😊 ✎ ⋮

✓ @mvanremmerden These questions are totally valid.

Background information: The feature "Protected containers" (and "Protected packages") is currently under active discussion and development. Therefore, this feature is currently behind a feature flag that will not be activated for `gitlab.com`. So, things are still moving and changing to some extent.

✓ Different action text ("Unprotect" vs. "Delete")

Our goal is to align the wording between "Protected containers" and "Protected branches". This is an open todo for the existing CRUD feature "Protected packages", see [#413641 \(comment 1770699822\)](#). I will adjust the wording accordingly for this specific MR. see https://gitlab.com/gitlab-org/gitlab/-/merge_requests/146436/diffs?commit_id=39b2845da9cfe4c2d836660dbd4d13657c3801c9&pin=7c21745545099a19bd5fa2a9f25d8c660425db32#7c21745545099a19bd5fa2a9f25d8c660425db32_44_43.

Hint #3: Work in Parallel

- Work on multiple MR to avoid idle time
- “Stacking” MRs with separate scopes when building on larger features
- Work in different areas of the code base to engage with different GitLab teams
- Keep track of follow-up todos

The screenshot displays a list of GitLab Merge Requests (MRs) for the project 'Protected packages'. Each MR is marked with a green checkmark in a box, indicating it is a draft or ready for review. The MRs are:

- Draft: Protected packages: Shorten GraphQL field `protectionRuleExists` Part 3** (3 of 9 checklist items completed, created 2 days ago by Gerardo Navarro). Tags: Community contribution, Leading Organization, devops, package, group, package registry, linked-issue, section ci, type maintenance, workflow in dev.
- Draft: Protected packages: Shorten GraphQL field `protectionRuleExists` Part 2** (0 of 11 checklist items completed, created 2 days ago by Gerardo Navarro). Tags: Community contribution, Leading Organization, devops, package, group, package registry, section ci, type maintenance, workflow in dev.
- Protected packages: Add help text for name pattern input** (0 of 10 checklist items completed, created 3 days ago by Gerardo Navarro). Tags: Community contribution, Leading Organization, Technical Writing, UI text, documentation, linked-issue, tw triaged, type feature, workflow in dev.
- Protected packages: Shorten GraphQL field `protectionRuleExists` Part 1** (9 of 9 checklist items completed, created 1 week ago by Gerardo Navarro). Tags: Community contribution, Leading Organization, Technical Writing, UX, backend, devops, package, docs improvement, documentation, frontend, group, package registry, linked-issue, maintenance, refactor, pipeline:mr-approved, section ci, tw triaged, type maintenance, workflow ready for review.
- Draft: Protected packages: REST API POST create package protection rules** (0 of 1 checklist item completed, created 1 week ago by Gerardo Navarro). Tags: Community contribution, Hackathon, Leading Organization, devops, package, group, package registry, section ci, type feature, workflow in dev.
- Protected containers: Use can_admin_all_resources? instead of user.admin?** (4 of 5 checklist items completed, created 2 weeks ago by Gerardo Navarro). Tags: Community contribution, Hackathon, Leading Organization, backend, devops, package, feature enhancement, group, container registry, pipeline:mr-approved, section ci, type feature, workflow ready for review.





Try out protected packages on code.siemens.com

Gerardo Navarro

Full-Stack / DevOps Software Engineer

gerardo@b310.de

<https://gitlab.com/gerardo-navarro>

