GitLab

Improving Contribution Guidelines for First Time Contributors Report

FY24-Q1 OKR

Contributor Success & Technical Writing

Demo Video

Expected Benefits

Achieve a higher success rate of first time contributor onboarding

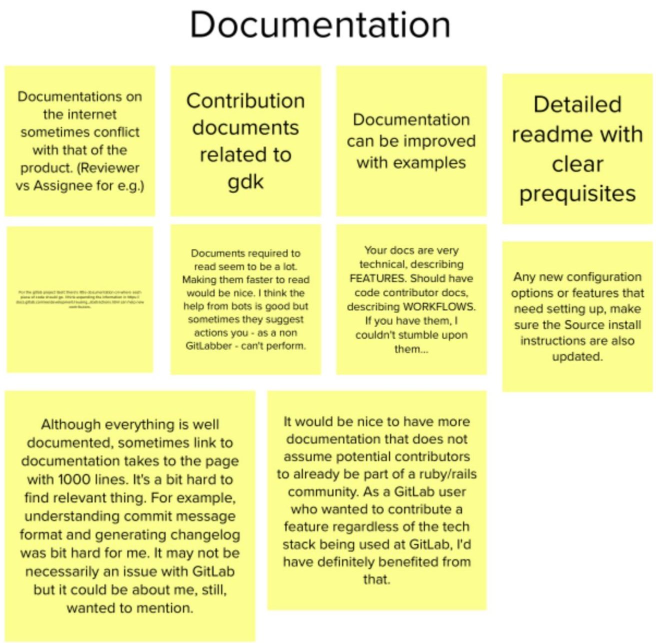
- Expect tutorial will help first time contributors make a merge request
- Expect GDK setup guides to reduce frustration in getting started
- Expect community forks documentation to remove blockers and improve collaboration
- Expect workflow guidelines to teach good practices and lead to faster team member reviews
- Expect non-developers to find contribution opportunities besides coding

Improve Contributor Success KPIs

- Potential UWCC with open merge requests
    - Expect reducing development blockers and frustrations to lead to more submitted MRs
- UWCC (Unique wider community contributors)
    - Expect more MRs using community forks to reach the finish line
- Returning contributors ratio
    - Expect smoother first time contributing experience to increase likelihood of returning

Old Contributor Journey

Contributor Survey Feedback | 2022



Documentation

Documentations on the internet sometimes conflict with that of the product. (Reviewer vs Assignee for e.g.)

Contribution documents related to gdk

Documentation can be improved with examples

Detailed readme with clear prequisites

For the gitlab project itself there's little documentation on where each piece of code should go. I think expanding the information in https://docs.gitlab.com/ee/development/reusing_abstractions.html can help new contributors.

Documents required to read seem to be a lot. Making them faster to read would be nice. I think the help from bots is good but sometimes they suggest actions you - as a non GitLabber - can't perform.

Your docs are very technical, describing FEATURES. Should have code contributor docs, describing WORKFLOWS. If you have them, I couldn't stumble upon them...

Any new configuration options or features that need setting up, make sure the Source install instructions are also updated.

Although everything is well documented, sometimes link to documentation takes to the page with 1000 lines. It's a bit hard to find relevant thing. For example, understanding commit message format and generating changelog was bit hard for me. It may not be necessarily an issue with GitLab but it could be about me, still, wanted to mention.

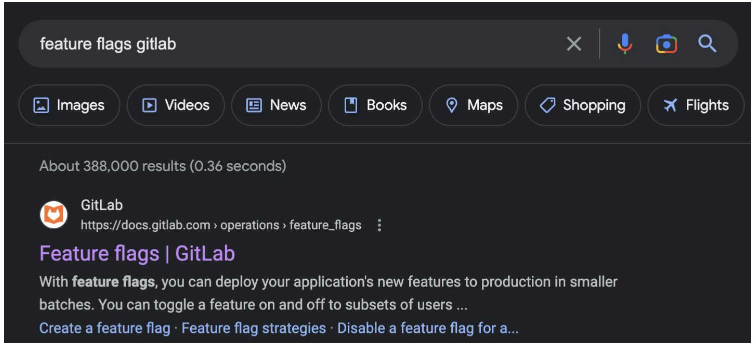
It would be nice to have more documentation that does not assume potential contributors to already be part of a ruby/rails community. As a GitLab user who wanted to contribute a feature regardless of the tech stack being used at GitLab, I'd have definitely benefited from that.

Contributors given multiple links for "contributing"

- about.gitlab.com/community/contribute/
- docs.gitlab.com/ee/development/
- docs.gitlab.com/
- developer.gitlab.com/

Google search

- **Google linking to user docs instead of contributing docs**
- **No way to tell the difference**



About 388,000 results (0.36 seconds)

GitLab
https://docs.gitlab.com › operations › feature_flags

Feature flags | GitLab

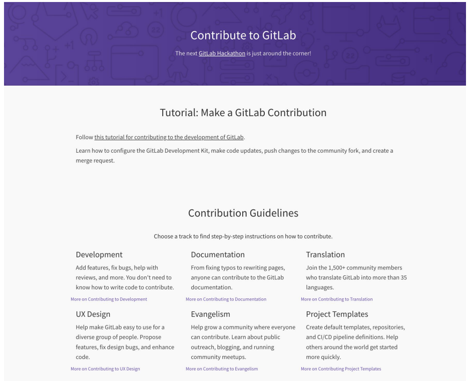
With **feature flags**, you can deploy your application's new features to production in smaller batches. You can toggle a feature on and off to subsets of users ...

Create a feature flag · Feature flag strategies · Disable a feature flag for a...

New Contributor Journey

Give one link

- about.gitlab.com/community/contribute
- First time contributor tutorial front and center
- All contributing areas easy to find
- Connected back to docs.gitlab.com/ee/development/



Contribute to GitLab

The next GitLab Hackathon is just around the corner!

Tutorial: Make a GitLab Contribution

Follow this tutorial for contributing to the development of GitLab.

Learn how to configure the GitLab Development Kit, make code updates, push changes to the community fork, and create a merge request.

Contribution Guidelines

Choose a track to find step-by-step instructions on how to contribute.

Development

Add features, fix bugs, help with reviews, and more. You don't need to know how to write code to contribute.

More on Contributing to Development

Documentation

From fixing typos to rewriting pages, anyone can contribute to the GitLab documentation.

More on Contributing to Documentation

Translation

Join the 1,500+ community members who translate GitLab into more than 35 languages.

More on Contributing to Translation

UX Design

Help make GitLab easy to use for a diverse group of people. Propose features, fix design bugs, and enhance code.

More on Contributing to UX Design

Evangelism

Help grow a community where everyone can contribute. Learn about public outreach, blogging, and running community meetups.

More on Contributing to Evangelism

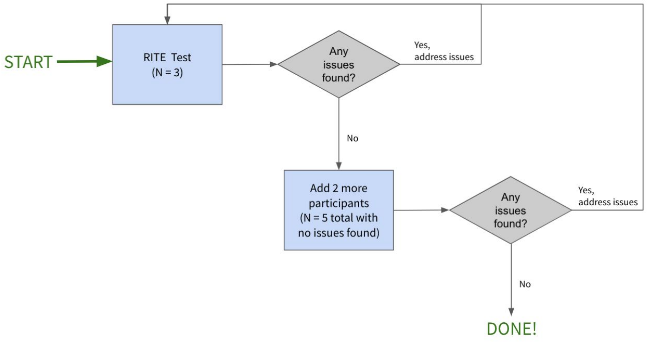
Project Templates

Create default templates, repositories, and CI/CD pipeline definitions. Help others around the world get started more quickly.

More on Contributing Project Templates

How will contributors navigate or stumble?

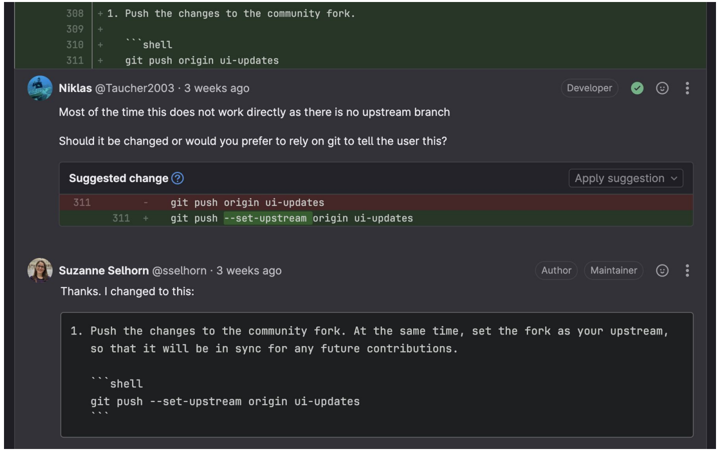
- Let’s do the research to find out
- Stop guessing what contributors are doing
- Work with UX Research for real data from contributor journeys
- Iterate improvements along the way



Learnings

What worked…

- Aligning Contributor Success & Technical Writing teams and OKRs
- Fast and quick iterations
- Feedback/MR reviews from community (see tutorial MR)



````
+1. Push the changes to the community fork.
+
+   ```shell
+   git push origin ui-updates
````

Niklas @Taucher2003 · 3 weeks ago
Developer

Most of the time this does not work directly as there is no upstream branch

Should it be changed or would you prefer to rely on git to tell the user this?

Suggested change
Apply suggestion

```
-   git push origin ui-updates
+   git push --set-upstream origin ui-updates
```

Suzanne Selhorn @sselhorn · 3 weeks ago
Author
Maintainer

Thanks. I changed to this:

~~~
1. Push the changes to the community fork. At the same time, set the fork as your upstream,
   so that it will be in sync for any future contributions.

   ```shell
   git push --set-upstream origin ui-updates
   ```
~~~

What didn’t work…

- Contributor Success DRI not having maintainer rights at start of OKR
  - Slowed down MRs
- Deduplicating content across different projects
  - Required multiple MRs and review processes
- Attempting too much criteria for one quarter
  - Documentation was messier and more unorganized than planned for

Next Steps

Immediate

- Redesign about.gitlab.com Contribute page with buttons / navigation elements
- Reorganize docs.gitlab.com contribute process navigation
- Rework issue/merge request workflows

Future Quarter

- Align with UX Research on RITE study of contributor documentation journey
- Measure analytics on pages and iterate to improve
- Rework Code Review Guidelines
- Add more tutorials/video walkthroughs
- Reach out for community support, even specific trusted contributors

GitLab

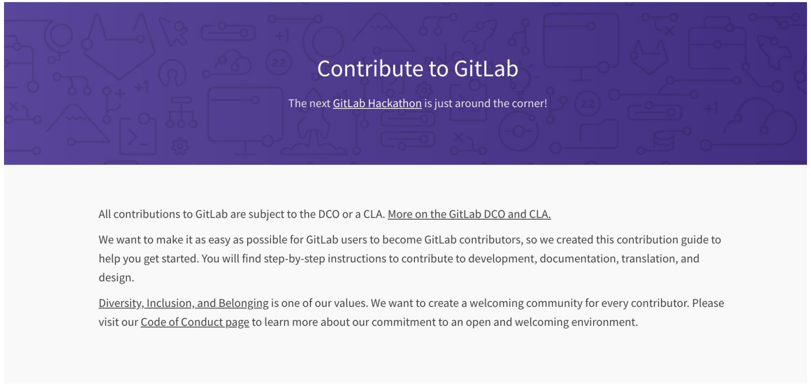
Examples

GitLab

Contribute Landing Page
Before & After

Before Contribute Landing Page After

Intro pulls away from contributing



Contribute to GitLab

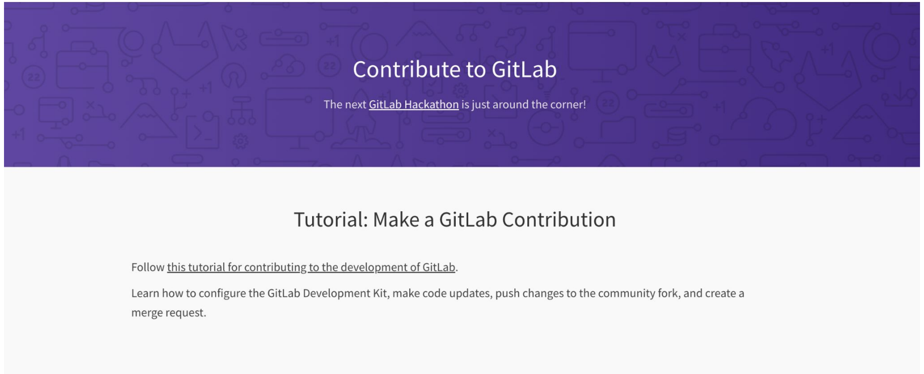
The next GitLab Hackathon is just around the corner!

All contributions to GitLab are subject to the DCO or a CLA. More on the GitLab DCO and CLA.

We want to make it as easy as possible for GitLab users to become GitLab contributors, so we created this contribution guide to help you get started. You will find step-by-step instructions to contribute to development, documentation, translation, and design.

Diversity, Inclusion, and Belonging is one of our values. We want to create a welcoming community for every contributor. Please visit our Code of Conduct page to learn more about our commitment to an open and welcoming environment.

Top page links to tutorial



Contribute to GitLab

The next GitLab Hackathon is just around the corner!

Tutorial: Make a GitLab Contribution

Follow this tutorial for contributing to the development of GitLab.

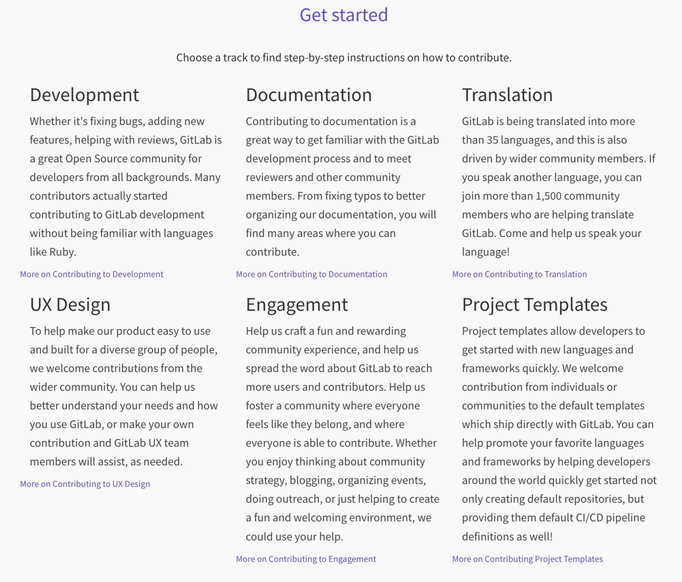
Learn how to configure the GitLab Development Kit, make code updates, push changes to the community fork, and create a merge request.

Before

Contribute Landing Page

After

Wordy and distracting descriptions



Get started

Choose a track to find step-by-step instructions on how to contribute.

Development

Whether it's fixing bugs, adding new features, helping with reviews, GitLab is a great Open Source community for developers from all backgrounds. Many contributors actually started contributing to GitLab development without being familiar with languages like Ruby.

More on Contributing to Development

Documentation

Contributing to documentation is a great way to get familiar with the GitLab development process and to meet reviewers and other community members. From fixing typos to better organizing our documentation, you will find many areas where you can contribute.

More on Contributing to Documentation

Translation

GitLab is being translated into more than 35 languages, and this is also driven by wider community members. If you speak another language, you can join more than 1,500 community members who are helping translate GitLab. Come and help us speak your language!

More on Contributing to Translation

UX Design

To help make our product easy to use and built for a diverse group of people, we welcome contributions from the wider community. You can help us better understand your needs and how you use GitLab, or make your own contribution and GitLab UX team members will assist, as needed.

More on Contributing to UX Design

Engagement

Help us craft a fun and rewarding community experience, and help us spread the word about GitLab to reach more users and contributors. Help us foster a community where everyone feels like they belong, and where everyone is able to contribute. Whether you enjoy thinking about community strategy, blogging, organizing events, doing outreach, or just helping to create a fun and welcoming environment, we could use your help.

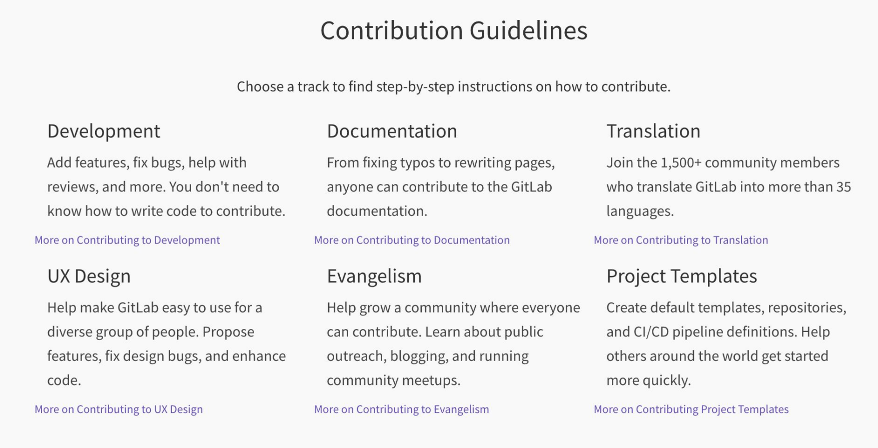
More on Contributing to Engagement

Project Templates

Project templates allow developers to get started with new languages and frameworks quickly. We welcome contribution from individuals or communities to the default templates which ship directly with GitLab. You can help promote your favorite languages and frameworks by helping developers around the world quickly get started not only creating default repositories, but providing them default CI/CD pipeline definitions as well!

More on Contributing Project Templates

Scannable tracks



Contribution Guidelines

Choose a track to find step-by-step instructions on how to contribute.

Development

Add features, fix bugs, help with reviews, and more. You don't need to know how to write code to contribute.

More on Contributing to Development

Documentation

From fixing typos to rewriting pages, anyone can contribute to the GitLab documentation.

More on Contributing to Documentation

Translation

Join the 1,500+ community members who translate GitLab into more than 35 languages.

More on Contributing to Translation

UX Design

Help make GitLab easy to use for a diverse group of people. Propose features, fix design bugs, and enhance code.

More on Contributing to UX Design

Evangelism

Help grow a community where everyone can contribute. Learn about public outreach, blogging, and running community meetups.

More on Contributing to Evangelism

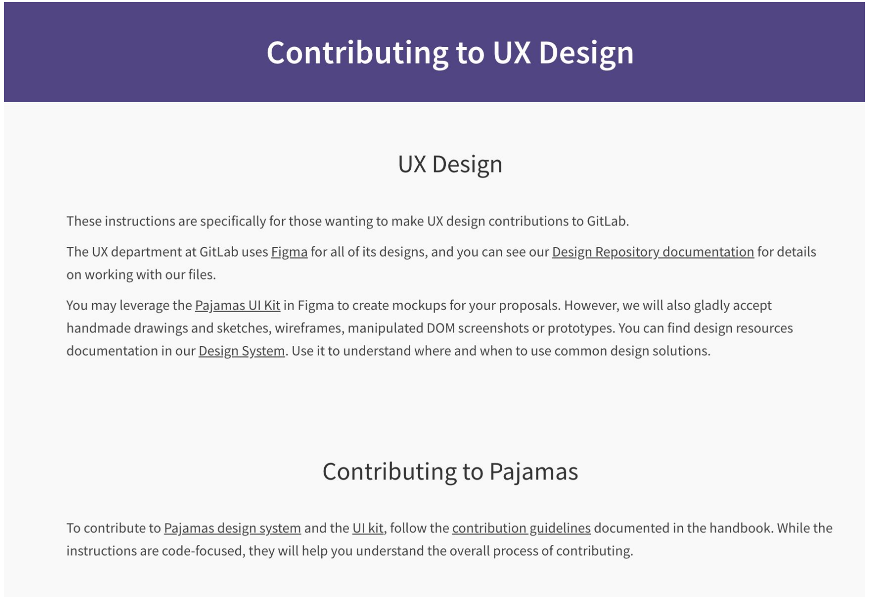
Project Templates

Create default templates, repositories, and CI/CD pipeline definitions. Help others around the world get started more quickly.

More on Contributing Project Templates

Before Contribute Landing Page After

Links go to subpage instead of docs



Contributing to UX Design

UX Design

These instructions are specifically for those wanting to make UX design contributions to GitLab.

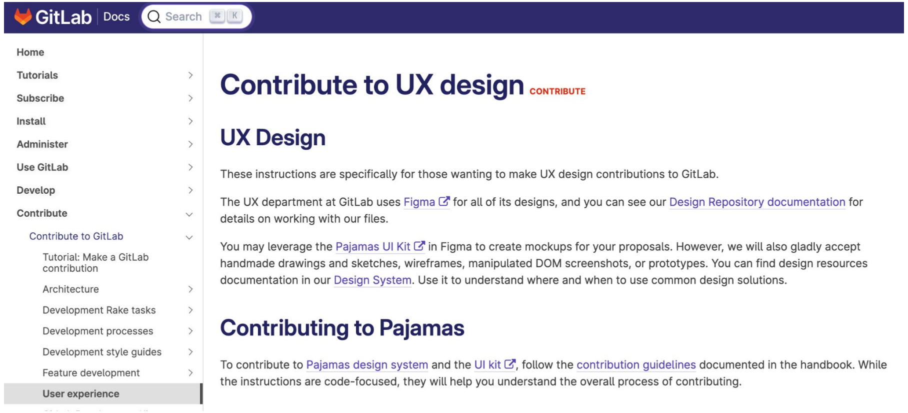
The UX department at GitLab uses Figma for all of its designs, and you can see our Design Repository documentation for details on working with our files.

You may leverage the Pajamas UI Kit in Figma to create mockups for your proposals. However, we will also gladly accept handmade drawings and sketches, wireframes, manipulated DOM screenshots or prototypes. You can find design resources documentation in our Design System. Use it to understand where and when to use common design solutions.

Contributing to Pajamas

To contribute to Pajamas design system and the UI kit, follow the contribution guidelines documented in the handbook. While the instructions are code-focused, they will help you understand the overall process of contributing.

Links go directly to docs



Contribute to UX design CONTRIBUTE

UX Design

These instructions are specifically for those wanting to make UX design contributions to GitLab.

The UX department at GitLab uses Figma for all of its designs, and you can see our Design Repository documentation for details on working with our files.

You may leverage the Pajamas UI Kit in Figma to create mockups for your proposals. However, we will also gladly accept handmade drawings and sketches, wireframes, manipulated DOM screenshots, or prototypes. You can find design resources documentation in our Design System. Use it to understand where and when to use common design solutions.

Contributing to Pajamas

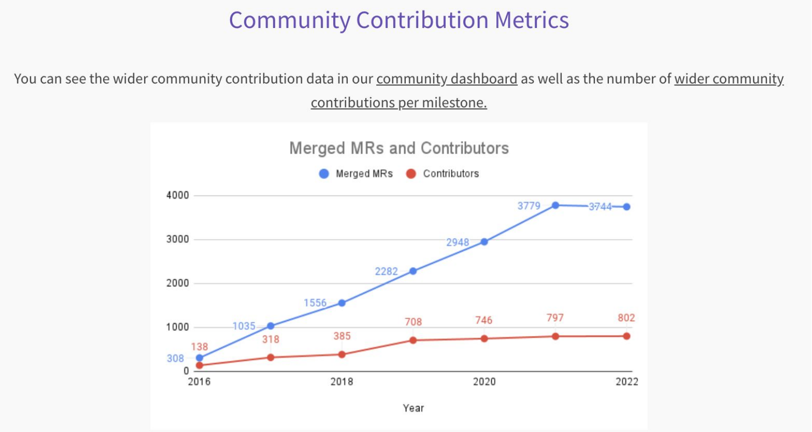
To contribute to Pajamas design system and the UI kit, follow the contribution guidelines documented in the handbook. While the instructions are code-focused, they will help you understand the overall process of contributing.

Before

Contribute Landing Page

After

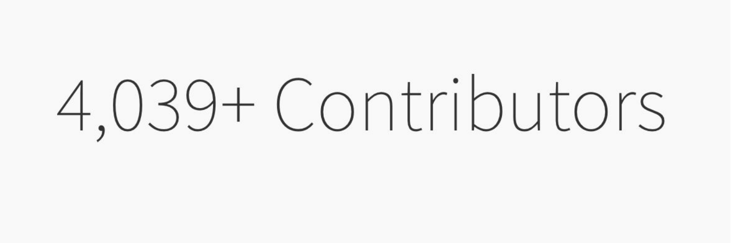
Graph misleads low contributor totals



Community Contribution Metrics

You can see the wider community contribution data in our community dashboard as well as the number of wider community contributions per milestone.

Single metric shows total contributors



4,039+ Contributors

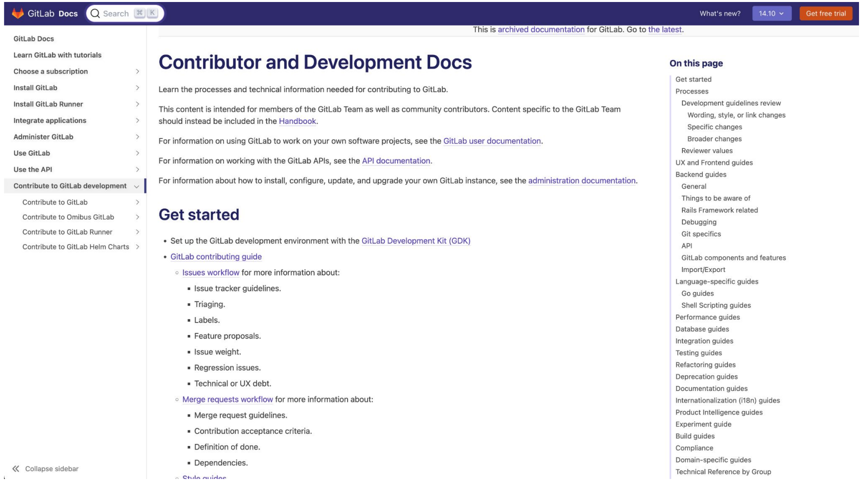
GitLab

Contribute Documentation

Before & After

Before Contribute Documentation After

Overwhelming main page



This is archived documentation for GitLab. Go to the latest.

Contributor and Development Docs

Learn the processes and technical information needed for contributing to GitLab.

This content is intended for members of the GitLab Team as well as community contributors. Content specific to the GitLab Team should instead be included in the Handbook.

For information on using GitLab to work on your own software projects, see the GitLab user documentation.

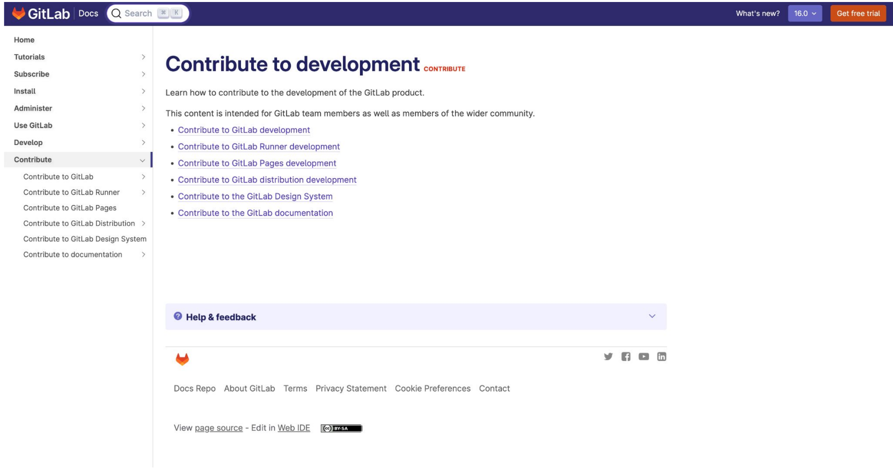
For information on working with the GitLab APIs, see the API documentation.

For information about how to install, configure, update, and upgrade your own GitLab instance, see the administration documentation.

Get started

- Set up the GitLab development environment with the GitLab Development Kit (GDK)
- GitLab contributing guide
  - Issues workflow for more information about:
    - Issue tracker guidelines.
    - Triaging.
    - Labels.
    - Feature proposals.
    - Issue weight.
    - Regression issues.
    - Technical or UX debt.
  - Merge requests workflow for more information about:
    - Merge request guidelines.
    - Contribution acceptance criteria.
    - Definition of done.
    - Dependencies.
  - Style guides

Clean main page with tracks


GitLab Docs
Search ⌘ K
What's new?
16.0
Get free trial
Home
Tutorials
Subscribe
Install
Administer
Use GitLab
Develop
Contribute
Contribute to GitLab
Contribute to GitLab Runner
Contribute to GitLab Pages
Contribute to GitLab Distribution
Contribute to GitLab Design System
Contribute to documentation

Contribute to development CONTRIBUTE

Learn how to contribute to the development of the GitLab product.

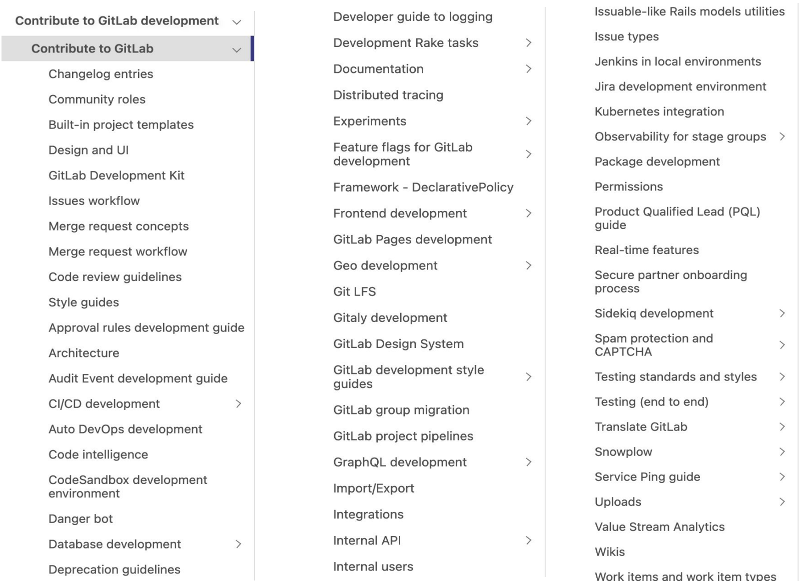
This content is intended for GitLab team members as well as members of the wider community.

- Contribute to GitLab development
- Contribute to GitLab Runner development
- Contribute to GitLab Pages development
- Contribute to GitLab distribution development
- Contribute to the GitLab Design System
- Contribute to the GitLab documentation

View page source - Edit in Web IDE

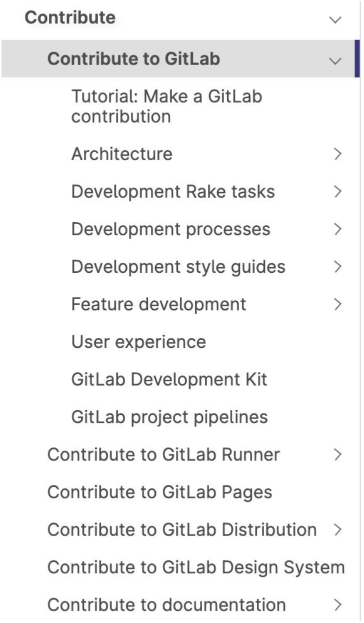
Before Contribute Documentation After

65+ subpages in navigation



- Contribute to GitLab development
  - Contribute to GitLab
    - Changelog entries
    - Community roles
    - Built-in project templates
    - Design and UI
    - GitLab Development Kit
    - Issues workflow
    - Merge request concepts
    - Merge request workflow
    - Code review guidelines
    - Style guides
    - Approval rules development guide
    - Architecture
    - Audit Event development guide
    - CI/CD development
    - Auto DevOps development
    - Code intelligence
    - CodeSandbox development environment
    - Danger bot
    - Database development
    - Deprecation guidelines
    - Developer guide to logging
    - Development Rake tasks
    - Documentation
    - Distributed tracing
    - Experiments
    - Feature flags for GitLab development
    - Framework - DeclarativePolicy
    - Frontend development
    - GitLab Pages development
    - Geo development
    - Git LFS
    - Gitaly development
    - GitLab Design System
    - GitLab development style guides
    - GitLab group migration
    - GitLab project pipelines
    - GraphQL development
    - Import/Export
    - Integrations
    - Internal API
    - Internal users
    - Issuable-like Rails models utilities
    - Issue types
    - Jenkins in local environments
    - Jira development environment
    - Kubernetes integration
    - Observability for stage groups
    - Package development
    - Permissions
    - Product Qualified Lead (PQL) guide
    - Real-time features
    - Secure partner onboarding process
    - Sidekiq development
    - Spam protection and CAPTCHA
    - Testing standards and styles
    - Testing (end to end)
    - Translate GitLab
    - Snowplow
    - Service Ping guide
    - Uploads
    - Value Stream Analytics
    - Wikis
    - Work items and work item types

9 direct subpages in navigation



- Contribute
  - Contribute to GitLab
    - Tutorial: Make a GitLab contribution
    - Architecture
    - Development Rake tasks
    - Development processes
    - Development style guides
    - Feature development
    - User experience
    - GitLab Development Kit
    - GitLab project pipelines
  - Contribute to GitLab Runner
  - Contribute to GitLab Pages
  - Contribute to GitLab Distribution
  - Contribute to GitLab Design System
  - Contribute to documentation

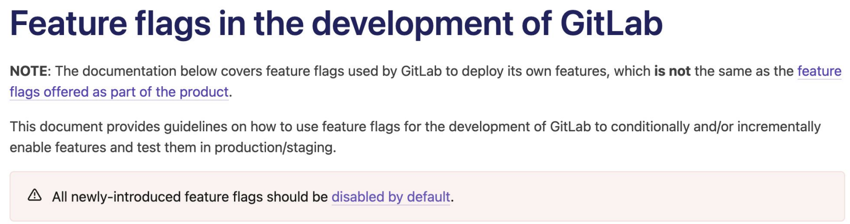
Before

Contribute Documentation

After

Contribute docs look like user docs

Added CONTRIBUTE tags and info boxes

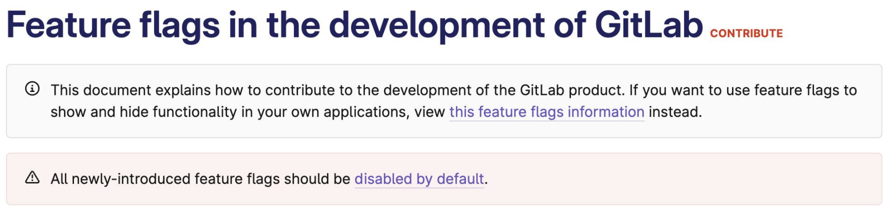


Feature flags in the development of GitLab

NOTE: The documentation below covers feature flags used by GitLab to deploy its own features, which **is not** the same as the feature flags offered as part of the product.

This document provides guidelines on how to use feature flags for the development of GitLab to conditionally and/or incrementally enable features and test them in production/staging.

⚠ All newly-introduced feature flags should be disabled by default.



Feature flags in the development of GitLab CONTRIBUTE

ⓘ This document explains how to contribute to the development of the GitLab product. If you want to use feature flags to show and hide functionality in your own applications, view this feature flags information instead.

⚠ All newly-introduced feature flags should be disabled by default.

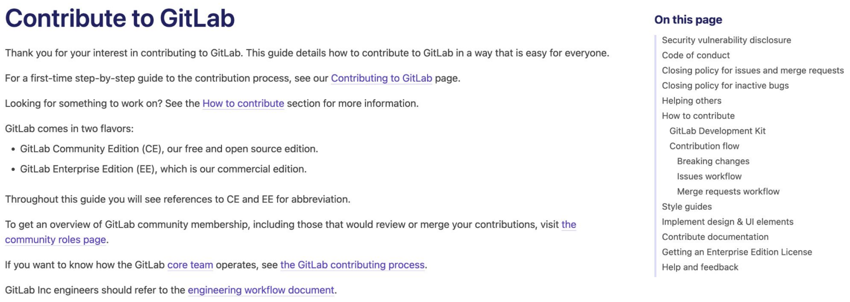
Before

Contribute Documentation

After

Messy development page

- Outdated workflows
- Step-by-step guide links to landing page



Contribute to GitLab

Thank you for your interest in contributing to GitLab. This guide details how to contribute to GitLab in a way that is easy for everyone.

For a first-time step-by-step guide to the contribution process, see our Contributing to GitLab page.

Looking for something to work on? See the How to contribute section for more information.

GitLab comes in two flavors:

- GitLab Community Edition (CE), our free and open source edition.
- GitLab Enterprise Edition (EE), which is our commercial edition.

Throughout this guide you will see references to CE and EE for abbreviation.

To get an overview of GitLab community membership, including those that would review or merge your contributions, visit the community roles page.

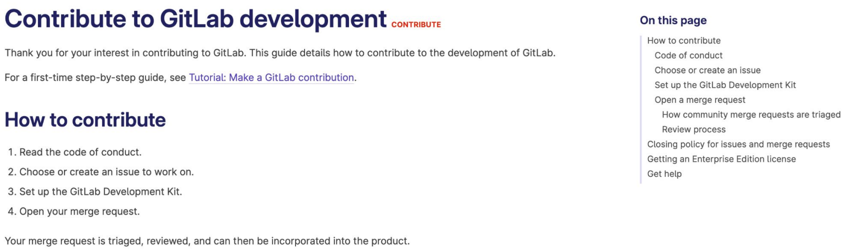
If you want to know how the GitLab core team operates, see the GitLab contributing process.

GitLab Inc engineers should refer to the engineering workflow document.

On this page

Streamlined development page

- Updated contribution workflow
- Step-by-step guide links to new tutorial



Contribute to GitLab development CONTRIBUTE

Thank you for your interest in contributing to GitLab. This guide details how to contribute to the development of GitLab.

For a first-time step-by-step guide, see Tutorial: Make a GitLab contribution.

How to contribute

1. Read the code of conduct.
2. Choose or create an issue to work on.
3. Set up the GitLab Development Kit.
4. Open your merge request.

Your merge request is triaged, reviewed, and can then be incorporated into the product.

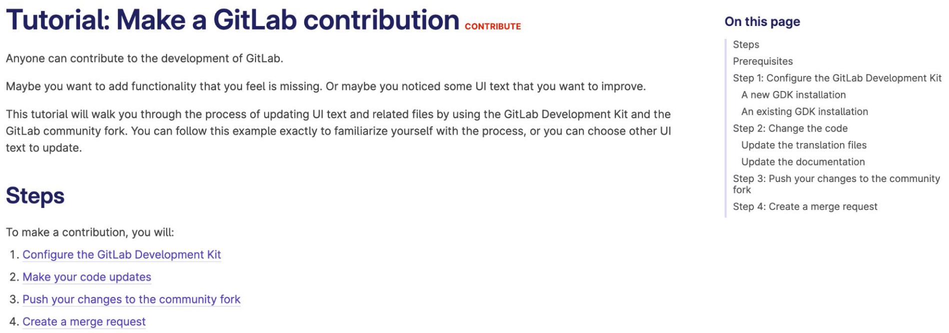
On this page

Before Contribute Documentation After

No guide for first time contributors

New tutorial for first time contributors

- GDK walkthrough with pictures
- How to make code changes
- How to update documentation
- Using the community forks
- Getting MR reviewed or finding help



Tutorial: Make a GitLab contribution CONTRIBUTE

Anyone can contribute to the development of GitLab.

Maybe you want to add functionality that you feel is missing. Or maybe you noticed some UI text that you want to improve.

This tutorial will walk you through the process of updating UI text and related files by using the GitLab Development Kit and the GitLab community fork. You can follow this example exactly to familiarize yourself with the process, or you can choose other UI text to update.

Steps

To make a contribution, you will:

1. Configure the GitLab Development Kit
2. Make your code updates
3. Push your changes to the community fork
4. Create a merge request

On this page