View online

Download PDF

How to breed a good OSS community

Paolo Insogna

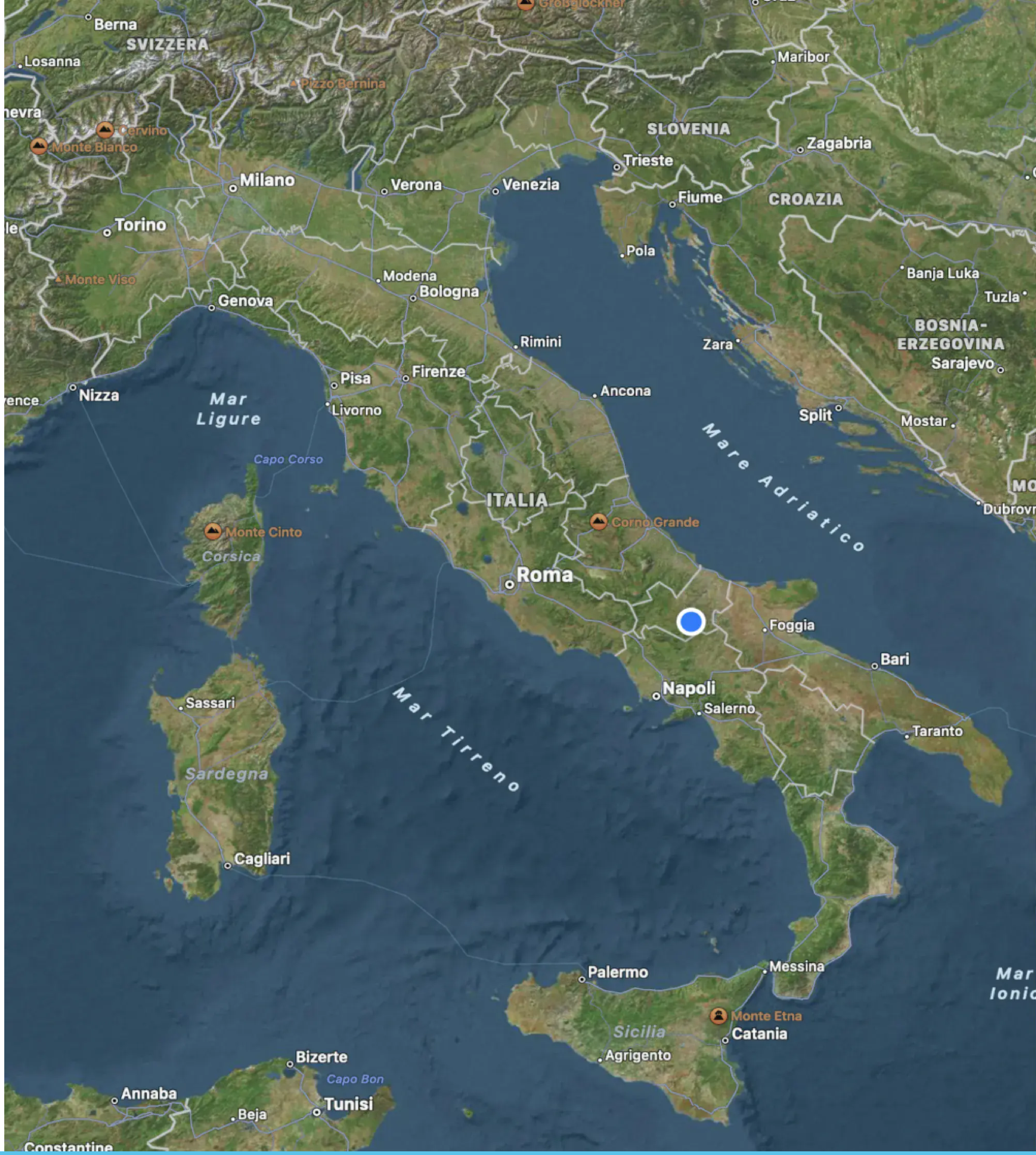
Node.js TSC, Principal Engineer @ **Platformatic**

Don't ask if you can't handle the answers!

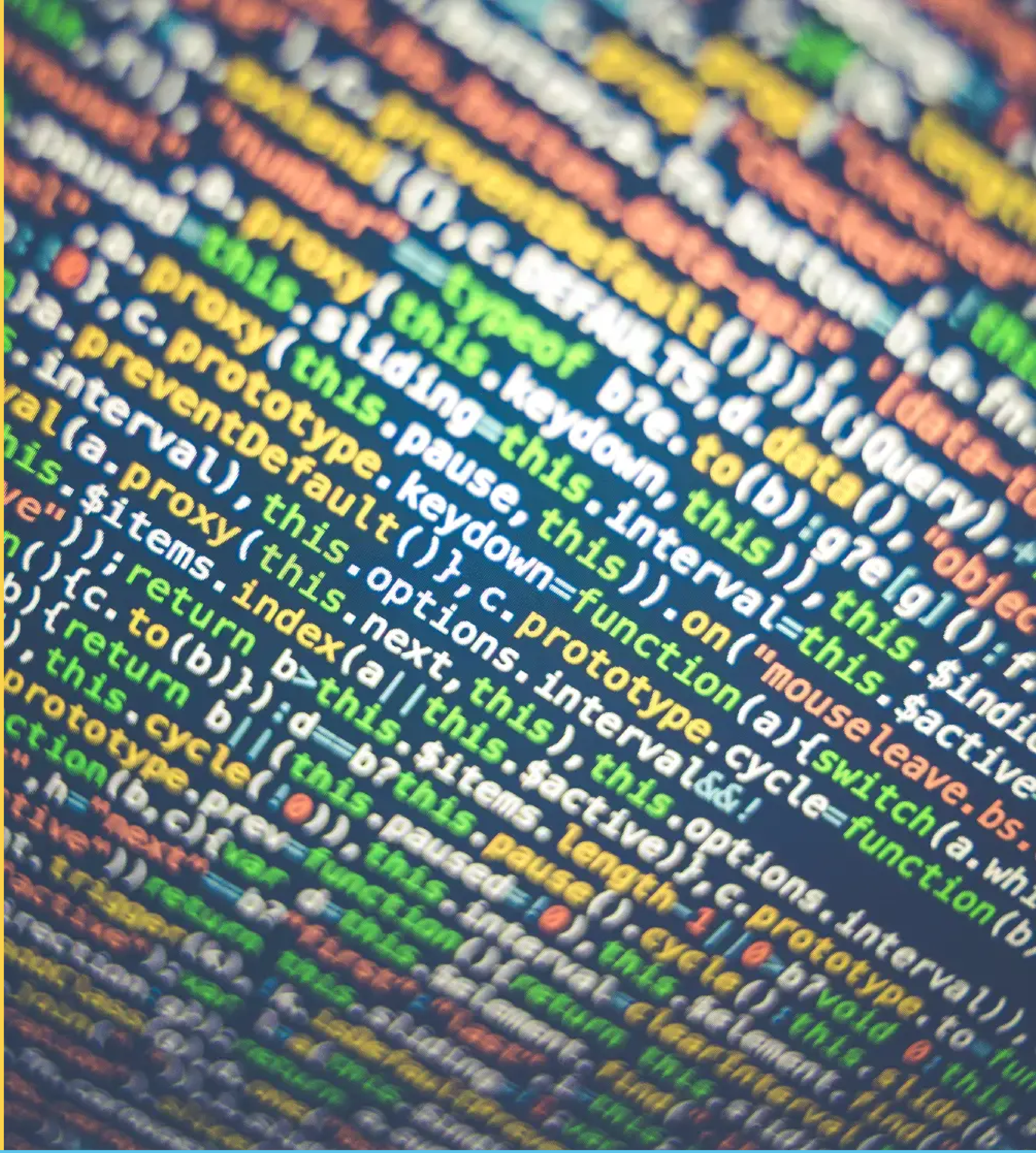
Hello, I'm **Paolo**!

Node.js	Technical Steering Committee Member
Platformatic	Principal Engineer

paoloinsogna.dev **ShogunPanda** **p_insogna** **pinsogna**



Do you want to mess with Open Source?



Let's clarify things

What is it?
A decentralized software development model that **encourages open collaboration**.

How is it possible?
The secret is in the **passion** of the people.

Is it working?
Open Source has proven to be really effective in improving software quality.

How it started?

Initially, it was free, not open

In the 70s the **free software movement** challenged the for-profit based development.

Free means too much

In English, the word free is ambiguous. **Not paid** was the most understood meaning.

Open Source was branched off the Free Software Movement

OSS software can be used for commercial purposes. **The freedom is retained.**

It might not be OpenSource

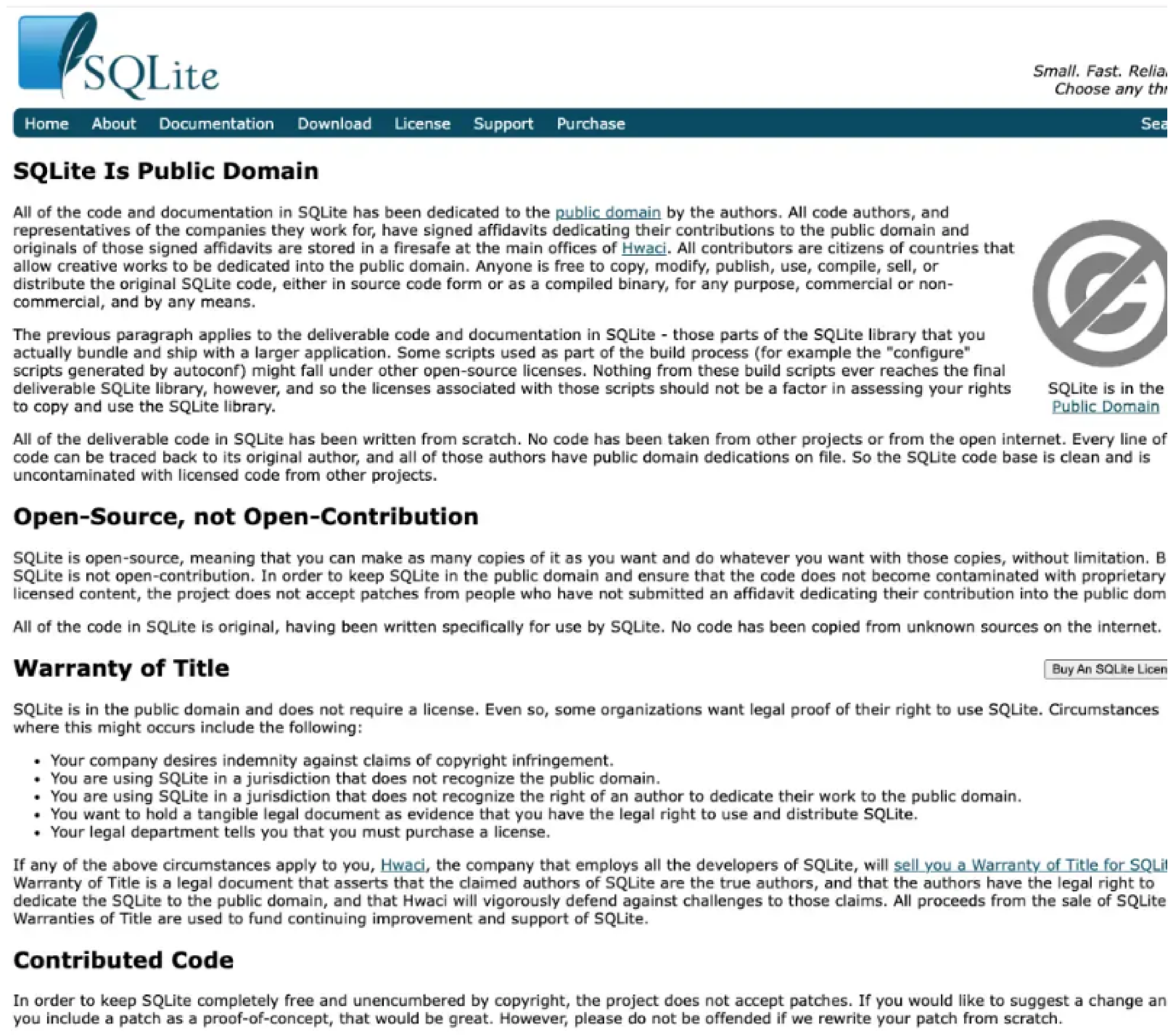
To be considered Open Source, a software must use a OSS compatible license.
The **Open Source Initiative** maintains a list of such licenses.

Beware!

The devil is in the details

Even if a software has a OSS license, it might not strictly be OSS.

SQLite is **Open Source but not Open Contribution**.


SQLite
Small. Fast. Relia
Choose any thr
Home About Documentation Download License Support Purchase Sea
Buy An SQLite Licen

SQLite Is Public Domain

All of the code and documentation in SQLite has been dedicated to the public domain by the authors. All code authors, and representatives of the companies they work for, have signed affidavits dedicating their contributions to the public domain and originals of those signed affidavits are stored in a firesafe at the main offices of Hwaci. All contributors are citizens of countries that allow creative works to be dedicated into the public domain. Anyone is free to copy, modify, publish, use, compile, sell, or distribute the original SQLite code, either in source code form or as a compiled binary, for any purpose, commercial or non-commercial, and by any means.

The previous paragraph applies to the deliverable code and documentation in SQLite - those parts of the SQLite library that you actually bundle and ship with a larger application. Some scripts used as part of the build process (for example the "configure" scripts generated by autoconf) might fall under other open-source licenses. Nothing from these build scripts ever reaches the final deliverable SQLite library, however, and so the licenses associated with those scripts should not be a factor in assessing your rights to copy and use the SQLite library.

SQLite is in the Public Domain

All of the deliverable code in SQLite has been written from scratch. No code has been taken from other projects or from the open internet. Every line of code can be traced back to its original author, and all of those authors have public domain dedications on file. So the SQLite code base is clean and is uncontaminated with licensed code from other projects.

Open-Source, not Open-Contribution

SQLite is open-source, meaning that you can make as many copies of it as you want and do whatever you want with those copies, without limitation. B SQLite is not open-contribution. In order to keep SQLite in the public domain and ensure that the code does not become contaminated with proprietary licensed content, the project does not accept patches from people who have not submitted an affidavit dedicating their contribution into the public dom

All of the code in SQLite is original, having been written specifically for use by SQLite. No code has been copied from unknown sources on the internet.

Warranty of Title

SQLite is in the public domain and does not require a license. Even so, some organizations want legal proof of their right to use SQLite. Circumstances where this might occurs include the following:

- Your company desires indemnity against claims of copyright infringement.
- You are using SQLite in a jurisdiction that does not recognize the public domain.
- You are using SQLite in a jurisdiction that does not recognize the right of an author to dedicate their work to the public domain.
- You want to hold a tangible legal document as evidence that you have the legal right to use and distribute SQLite.
- Your legal department tells you that you must purchase a license.

If any of the above circumstances apply to you, Hwaci, the company that employs all the developers of SQLite, will sell you a Warranty of Title for SQLi Warranty of Title is a legal document that asserts that the claimed authors of SQLite are the true authors, and that the authors have the legal right to dedicate the SQLite to the public domain, and that Hwaci will vigorously defend against challenges to those claims. All proceeds from the sale of SQLite Warranties of Title are used to fund continuing improvement and support of SQLite.

Contributed Code

In order to keep SQLite completely free and unencumbered by copyright, the project does not accept patches. If you would like to suggest a change an you include a patch as a proof-of-concept, that would be great. However, please do not be offended if we rewrite your patch from scratch.

Why would somebody want to go Open Source?

The best of the best of the best
You can access a developer base not physically close to you.

Different cultures, backgrounds and capabilities
You get different point of views, which increases quality.

Low maintenance cost
You don't need an office, just a version control server.

Wait a sec...

Ring a bell?

It's similar, but it's not!

The Open Source development model is similar to Remote Working.
The huge difference is that people are driven by **passion**, not money.

You are not their boss!

How to make it similar?

You can create bounty programs to prioritize collaboration on bug fixing or feature development.

BOUNTY HUNTERS ATTENTION!
WANTED
$5,000 REWARD!
NOTIFY NEAREST LAW ENFORCEMENT AGENCY

BAD!

DON'T!

Diversity and freedom are a huge treasure

Let them follow their path
The Open Source model works because people can freely follow their paths.

It's a hobby
Most collaborators send code in their free time, after a day of boring coding.

Keep the difference
If you make it look like a job, you will lose them.

How can you avoid losing control?

1

Propose, don't impose
You can share your vision and see if people are interested.

2

Follow your path
You can still deliver the features you care about.

3

Be open
Other people's path might be useful for yours.

Trust the community!

The world is nice ...

People generally want to help.

The have interest in what you are doing for their own needs.

... but don't be afraid to be harsh ...

Quick and effective intervention about uncomfortable situation are the key.

Avoiding creating a toxic community is always your **first** priority.

… and don't forget to be kind!

You will get people of all knowledges and expertises. Be patient, especially with newcomers.

Basically, it's like **parenting**: endless patience, close to no recognition.

But here we're talking about breeding...

The job NEVER ends

Too big to fail
Once you start a community, you are responsible for maintaining it. Potentially forever.

Don't take it personally
The best way to manage such communities is to avoid a **one-person show**.

Everybody is replaceable
All people are important, but prepare the community to survive if people suddenly disappear.

Nobody wants to be sued!

Why do we need legal licenses

Since Open Source Software is used in every sector, it has legally implication and liability. You need to avoid consequences.

Do not reinvent the wheel, choose an existing license!

Remember, it's permanent ...

You can change...
With every release, you can pick a new compatible license.

... but not retroactively.
Only new releases use a different licenses. Past is unchangeable.

Forking is unavoidable
If people don't like the new license, they can fork the software.

...but you can still profit from it!

Remember how it started?
Open Source doesn't mean free.

Think outside of the box
You don't have to think about profiting directly from the software.

Be around and everywhere
You can profit by offering premium services or a commercial derived product.

Don't try to mess the system

People don't like if you try to take away their beloved Open Source software.

Node.js	Terraform	Rust

Being kind never hurts!

Questions?

Coffee Break

Let's get to the action!

First of all, choose the hosting

GitHub

GitLab

BitBucket

Three reason not to go self-hosted or exotic

1

Self hosting is hard

Ownership cost will increase.

2

Self hosting is hidden

You will be hard to be found.

3

GIT is well known

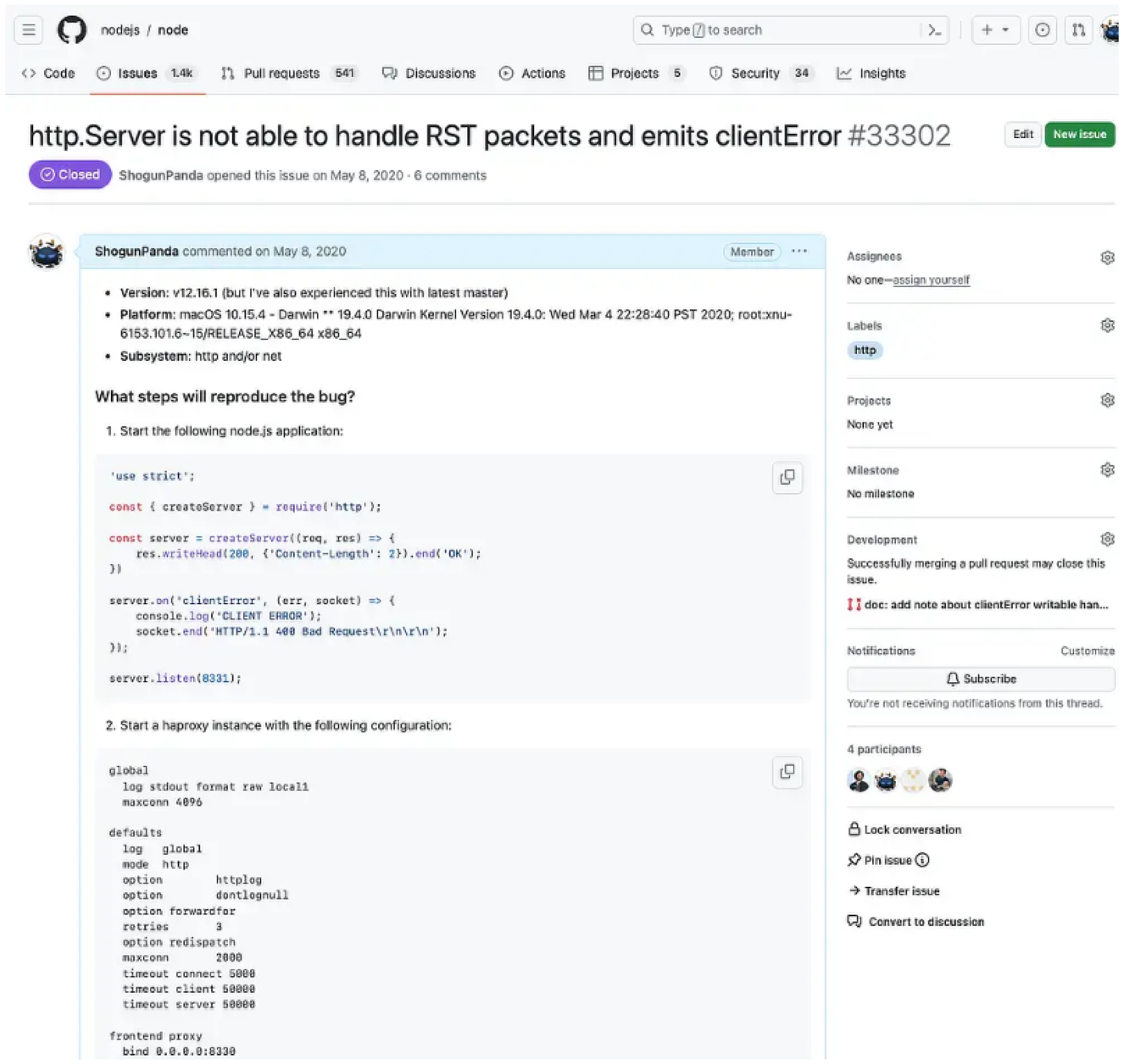
Do not create entry barriers.

How do we get started?

Everybody has issues

The core idea to attract people is to show you care.

The issue system is where you usually take feedback and bug reports from your user base.



http.Server is not able to handle RST packets and emits clientError #33302

ShogunPanda opened this issue on May 8, 2020 · 6 comments

ShogunPanda commented on May 8, 2020

- **Version:** v12.16.1 (but I've also experienced this with latest master)
- **Platform:** macOS 10.15.4 - Darwin ** 19.4.0 Darwin Kernel Version 19.4.0: Wed Mar 4 22:28:40 PST 2020; root:xnu-6153.101.6~15/RELEASE_X86_64 x86_64
- **Subsystem:** http and/or net

What steps will reproduce the bug?

1. Start the following node.js application:

```
'use strict';

const { createServer } = require('http');

const server = createServer((req, res) => {
    res.writeHead(200, {'Content-Length': 2}).end('OK');
})

server.on('clientError', (err, socket) => {
    console.log('CLIENT ERROR');
    socket.end('HTTP/1.1 400 Bad Request\r\n\r\n');
});

server.listen(8331);
```

2. Start a haproxy instance with the following configuration:

```
global
  log stdout format raw local1
  maxconn 4096

defaults
  log   global
  mode  http
  option        httplog
  option        dontlognull
  option forwardfor
  retries       3
  option redispatch
  maxconn       2000
  timeout connect 5000
  timeout client 50000
  timeout server 50000

frontend proxy
  bind 0.0.0.0:8330
```

Assignees
No one—assign yourself

Labels
http

Projects
None yet

Milestone
No milestone

Development
Successfully merging a pull request may close this issue.
doc: add note about clientError writable han...

Notifications
Customize
Subscribe
You're not receiving notifications from this thread.

4 participants

Lock conversation
Pin issue
Transfer issue
Convert to discussion

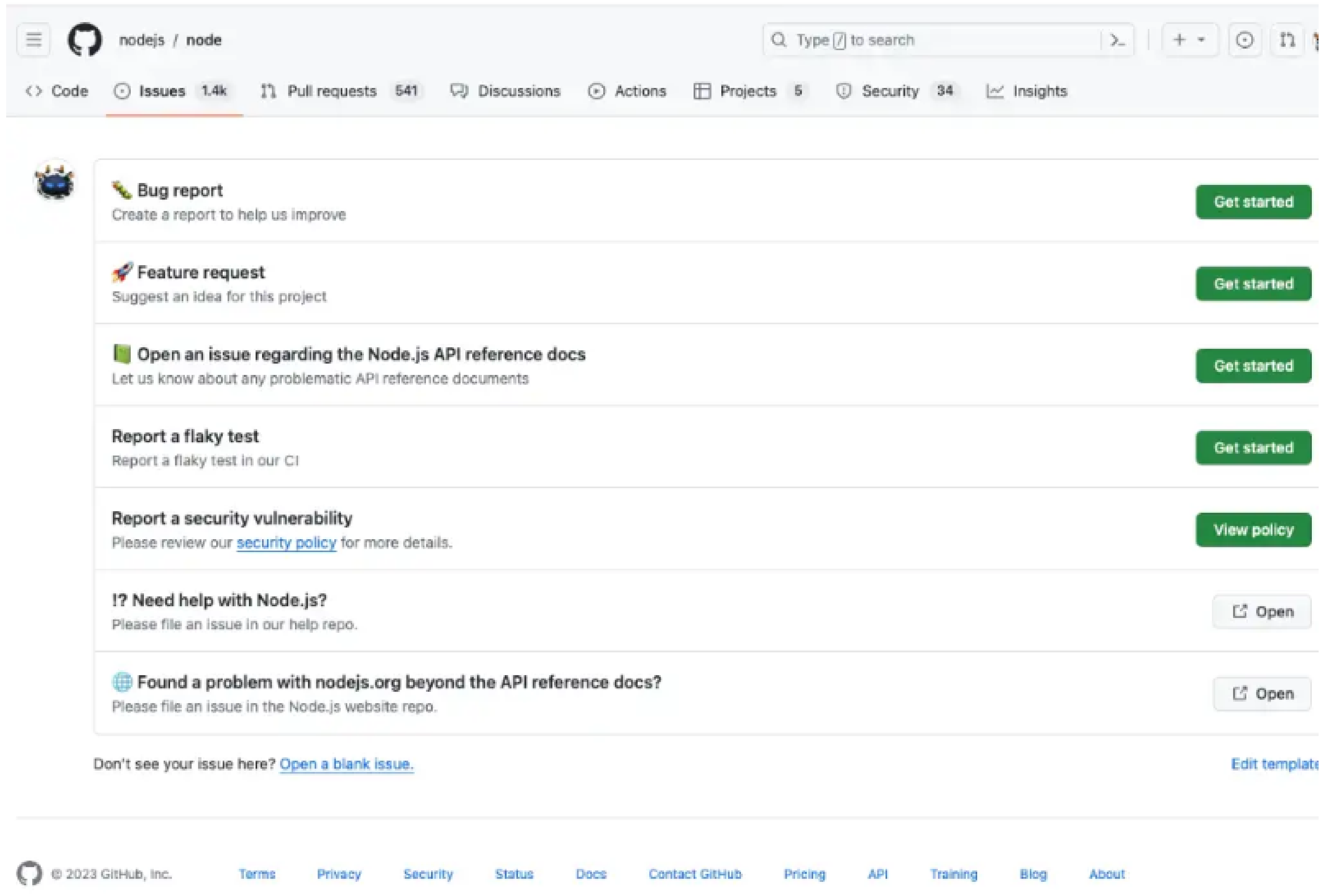
Templates help

It will help you

You can explicitly ask for the info you care the most.

It will help them

Having a templates speeds up their reporting time.



nodejs / node

Code · Issues 1.4k · Pull requests 541 · Discussions · Actions · Projects 5 · Security 34 · Insights

🐛 Bug report
Create a report to help us improve
Get started

🚀 Feature request
Suggest an idea for this project
Get started

📗 Open an issue regarding the Node.js API reference docs
Let us know about any problematic API reference documents
Get started

Report a flaky test
Report a flaky test in our CI
Get started

Report a security vulnerability
Please review our security policy for more details.
View policy

⁉️ Need help with Node.js?
Please file an issue in our help repo.
Open

🌐 Found a problem with nodejs.org beyond the API reference docs?
Please file an issue in the Node.js website repo.
Open

Don't see your issue here? Open a blank issue.

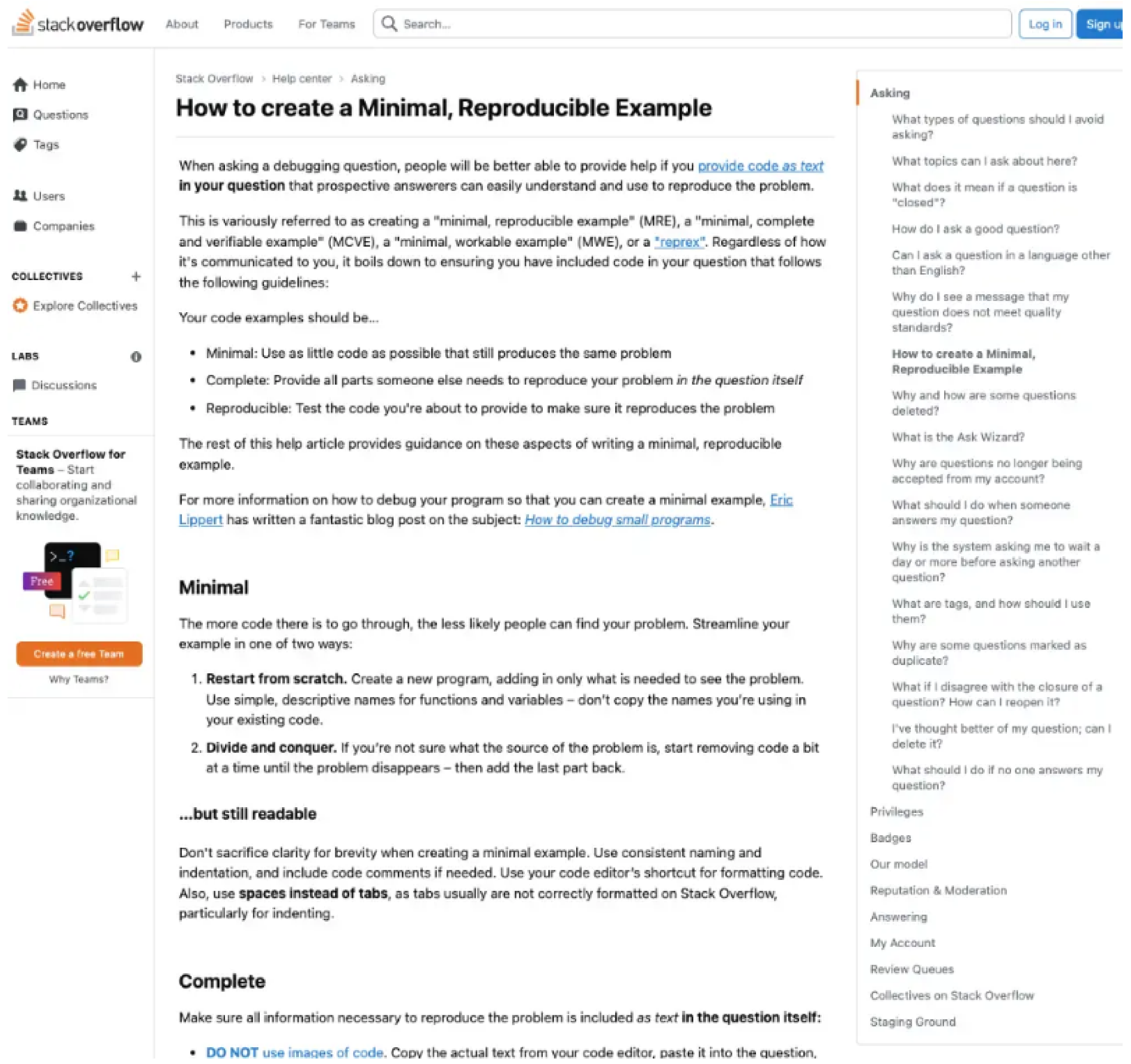
© 2023 GitHub, Inc. Terms Privacy Security Status Docs Contact GitHub Pricing API Training Blog About

Reproducible examples are crucial for bugs

Reduce to the minimum
You want to rule out all unrelevant details.

Remove sensitive information
Nobody wants to get in trouble.

Ask for a public repository
It will facilitate reporters to help triagers.


stackoverflow
About
Products
For Teams
Search...
Log in
Sign up
Home
Questions
Tags
Users
Companies
COLLECTIVES
Explore Collectives
LABS
Discussions
TEAMS
Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.
Free
Create a free Team
Why Teams?
Asking
What types of questions should I avoid asking?
What topics can I ask about here?
What does it mean if a question is "closed"?
How do I ask a good question?
Can I ask a question in a language other than English?
Why do I see a message that my question does not meet quality standards?
How to create a Minimal, Reproducible Example
Why and how are some questions deleted?
What is the Ask Wizard?
Why are questions no longer being accepted from my account?
What should I do when someone answers my question?
Why is the system asking me to wait a day or more before asking another question?
What are tags, and how should I use them?
Why are some questions marked as duplicate?
What if I disagree with the closure of a question? How can I reopen it?
I've thought better of my question; can I delete it?
What should I do if no one answers my question?
Privileges
Badges
Our model
Reputation & Moderation
Answering
My Account
Review Queues
Collectives on Stack Overflow
Staging Ground

Stack Overflow › Help center › Asking

How to create a Minimal, Reproducible Example

When asking a debugging question, people will be better able to provide help if you provide code as text **in your question** that prospective answerers can easily understand and use to reproduce the problem.

This is variously referred to as creating a "minimal, reproducible example" (MRE), a "minimal, complete and verifiable example" (MCVE), a "minimal, workable example" (MWE), or a "reprex". Regardless of how it's communicated to you, it boils down to ensuring you have included code in your question that follows the following guidelines:

Your code examples should be...

- Minimal: Use as little code as possible that still produces the same problem
- Complete: Provide all parts someone else needs to reproduce your problem *in the question itself*
- Reproducible: Test the code you're about to provide to make sure it reproduces the problem

The rest of this help article provides guidance on these aspects of writing a minimal, reproducible example.

For more information on how to debug your program so that you can create a minimal example, Eric Lippert has written a fantastic blog post on the subject: *How to debug small programs*.

Minimal

The more code there is to go through, the less likely people can find your problem. Streamline your example in one of two ways:

1. **Restart from scratch.** Create a new program, adding in only what is needed to see the problem. Use simple, descriptive names for functions and variables – don't copy the names you're using in your existing code.
2. **Divide and conquer.** If you're not sure what the source of the problem is, start removing code a bit at a time until the problem disappears – then add the last part back.

...but still readable

Don't sacrifice clarity for brevity when creating a minimal example. Use consistent naming and indentation, and include code comments if needed. Use your code editor's shortcut for formatting code. Also, use **spaces instead of tabs**, as tabs usually are not correctly formatted on Stack Overflow, particularly for indenting.

Complete

Make sure all information necessary to reproduce the problem is included *as text* **in the question itself:**

- **DO NOT** use images of code. Copy the actual text from your code editor, paste it into the question,

You don't owe anybody anything ...

It is in your rights to close bugs or features without solving them.

… but don't forget kindess!

1

Be respectful
Provide an exhaustive explanation.

2

Be gentle
Use a kind tone.

3

Be reasonable
Be open to change your mind.

In case of doubts

“A component of a system should behave in a way that most users will expect it to behave, and therefore not astonish or surprise users.”

Principle of least surprise

Label everything

Have a process
Following procedures will speed you up.

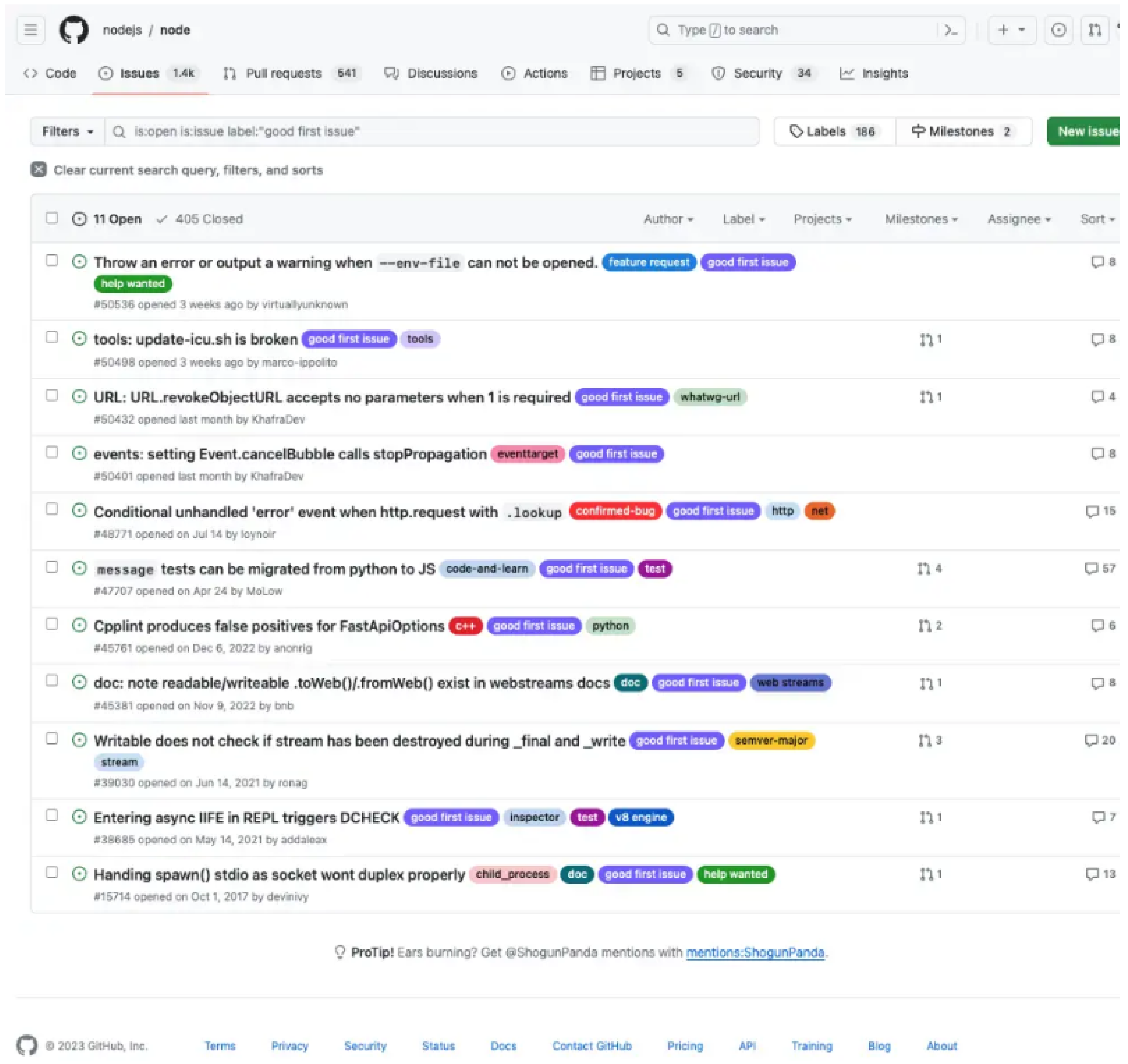
Divide et impera
Using labels helps categorizing and dividing tasks.

Don't overcommit
Limit the number of labels so that they are still useful.

Remember about good first time issues

When people come in your community and want to help, they don't know where to start.

By having **good first time issues** will help them to independently starting being involved.



nodejs / node

Type [/] to search

Code · Issues 1.4k · Pull requests 541 · Discussions · Actions · Projects 5 · Security 34 · Insights

Filters · is:open is:issue label:"good first issue" · Labels 186 · Milestones 2 · New issue

Clear current search query, filters, and sorts

11 Open ✓ 405 Closed · Author · Label · Projects · Milestones · Assignee · Sort

- Throw an error or output a warning when `--env-file` can not be opened. [feature request] [good first issue] [help wanted] — 8
  #50536 opened 3 weeks ago by virtuallyunknown
- tools: update-icu.sh is broken [good first issue] [tools] — 1 — 8
  #50498 opened 3 weeks ago by marco-ippolito
- URL: URL.revokeObjectURL accepts no parameters when 1 is required [good first issue] [whatwg-url] — 1 — 4
  #50432 opened last month by KhafraDev
- events: setting Event.cancelBubble calls stopPropagation [eventtarget] [good first issue] — 8
  #50401 opened last month by KhafraDev
- Conditional unhandled 'error' event when http.request with `.lookup` [confirmed-bug] [good first issue] [http] [net] — 15
  #48771 opened on Jul 14 by loynoir
- `message` tests can be migrated from python to JS [code-and-learn] [good first issue] [test] — 4 — 57
  #47707 opened on Apr 24 by MoLow
- Cpplint produces false positives for FastApiOptions [c++] [good first issue] [python] — 2 — 6
  #45761 opened on Dec 6, 2022 by anonrig
- doc: note readable/writeable .toWeb()/.fromWeb() exist in webstreams docs [doc] [good first issue] [web streams] — 1 — 8
  #45381 opened on Nov 9, 2022 by bnb
- Writable does not check if stream has been destroyed during _final and _write [good first issue] [semver-major] [stream] — 3 — 20
  #39030 opened on Jun 14, 2021 by ronag
- Entering async IIFE in REPL triggers DCHECK [good first issue] [inspector] [test] [v8 engine] — 1 — 7
  #38685 opened on May 14, 2021 by addaleax
- Handing spawn() stdio as socket wont duplex properly [child_process] [doc] [good first issue] [help wanted] — 1 — 13
  #15714 opened on Oct 1, 2017 by devinivy

ProTip! Ears burning? Get @ShogunPanda mentions with mentions:ShogunPanda.

© 2023 GitHub, Inc. · Terms · Privacy · Security · Status · Docs · Contact GitHub · Pricing · API · Training · Blog · About

The next big pillar

Pull Request

The core of the model
This is how collaborators send you code.

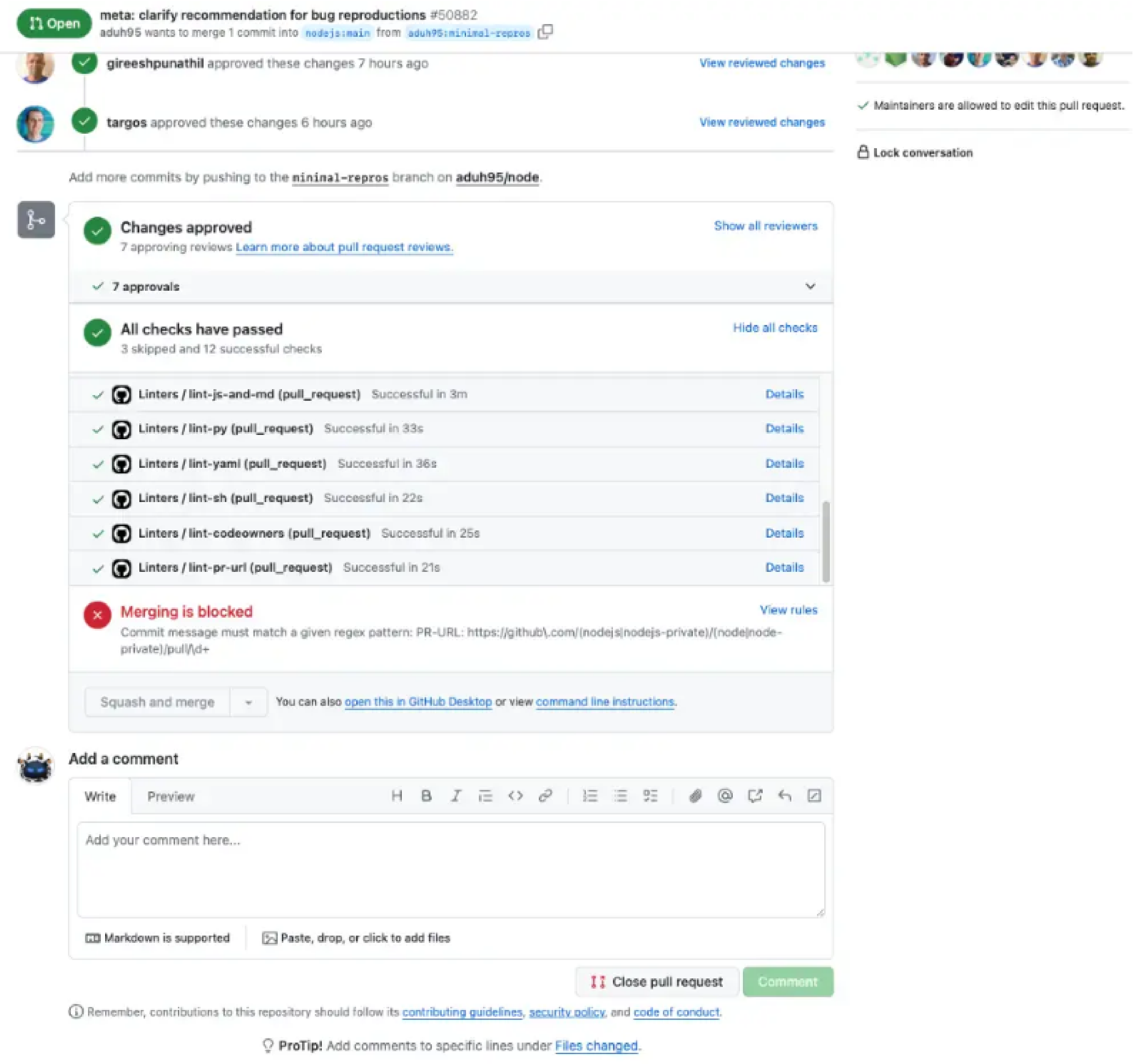
The quality will vary
It is your responsibility to maintain the level.

The scope will vary
Same as above, **don't be afraid of changes**.

Always ask for tests

While formal TDD is hard, Pull Request **must** contain tests when non trivial.

Even if a PR content is very appealing, you have to **resist the tentation to accept it without tests**.



Open meta: clarify recommendation for bug reproductions #50882
aduh95 wants to merge 1 commit into `nodejs:main` from `aduh95:minimal-repros`

gireeshpunathil approved these changes 7 hours ago — View reviewed changes

targos approved these changes 6 hours ago — View reviewed changes

✓ Maintainers are allowed to edit this pull request.

Lock conversation

Add more commits by pushing to the `minimal-repros` branch on aduh95/node.

Changes approved — Show all reviewers
7 approving reviews Learn more about pull request reviews.

✓ 7 approvals

All checks have passed — Hide all checks
3 skipped and 12 successful checks

✓ Linters / lint-js-and-md (pull_request)	Successful in 3m	Details
✓ Linters / lint-py (pull_request)	Successful in 33s	Details
✓ Linters / lint-yaml (pull_request)	Successful in 36s	Details
✓ Linters / lint-sh (pull_request)	Successful in 22s	Details
✓ Linters / lint-codeowners (pull_request)	Successful in 25s	Details
✓ Linters / lint-pr-url (pull_request)	Successful in 21s	Details

Merging is blocked — View rules
Commit message must match a given regex pattern: PR-URL: https://github\.com/(nodejs|nodejs-private)/(node|node-private)/pull/\d+

Squash and merge — You can also open this in GitHub Desktop or view command line instructions.

Add a comment

Remember, contributions to this repository should follow its contributing guidelines, security policy, and code of conduct.

ProTip! Add comments to specific lines under Files changed.

The review process

1. **Behavior and security**
   Make sure the PR does what it claims, and double check security implications.

2. **Performance**
   Nobody likes slow software, isn't it?

3. **Quality and clarity**
   A poorly (or obscurely) written performant code make more harm than good.

4. **Enforce the conventions**
   Same as above, **don't be afraid to ask for picky changes**.

Review must be thorough

Nobody rushes you to merge code. If possible, require for multiple approvals before committing.

Once you give enough time for objection, then you can merge.

CI and workflows

Available everywhere
In modern service you have a CI and workflow system.

It's so easy
Most of these tools are very easy to use.

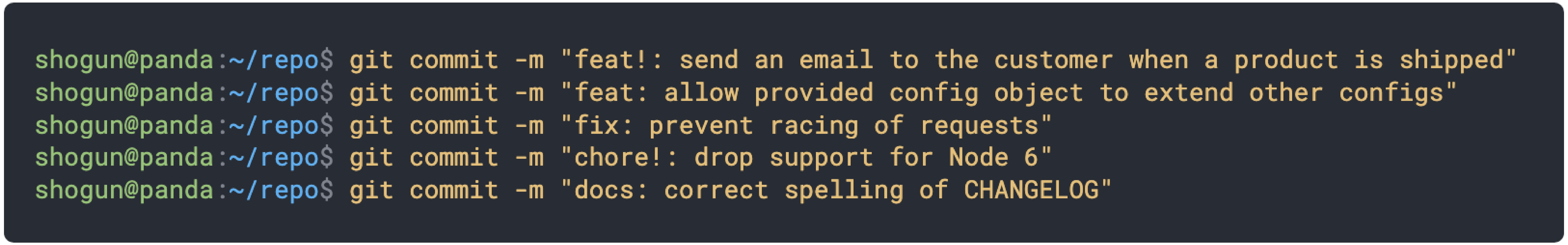
CI is a life saver
Don't merge any PR with a red CI.

Conventional commits

How to add human and machine readable meaning to commit messages.

Check the **official website**!

Show me some examples



```
shogun@panda:~/repo$ git commit -m "feat!: send an email to the customer when a product is shipped"
shogun@panda:~/repo$ git commit -m "feat: allow provided config object to extend other configs"
shogun@panda:~/repo$ git commit -m "fix: prevent racing of requests"
shogun@panda:~/repo$ git commit -m "chore!: drop support for Node 6"
shogun@panda:~/repo$ git commit -m "docs: correct spelling of CHANGELOG"
```

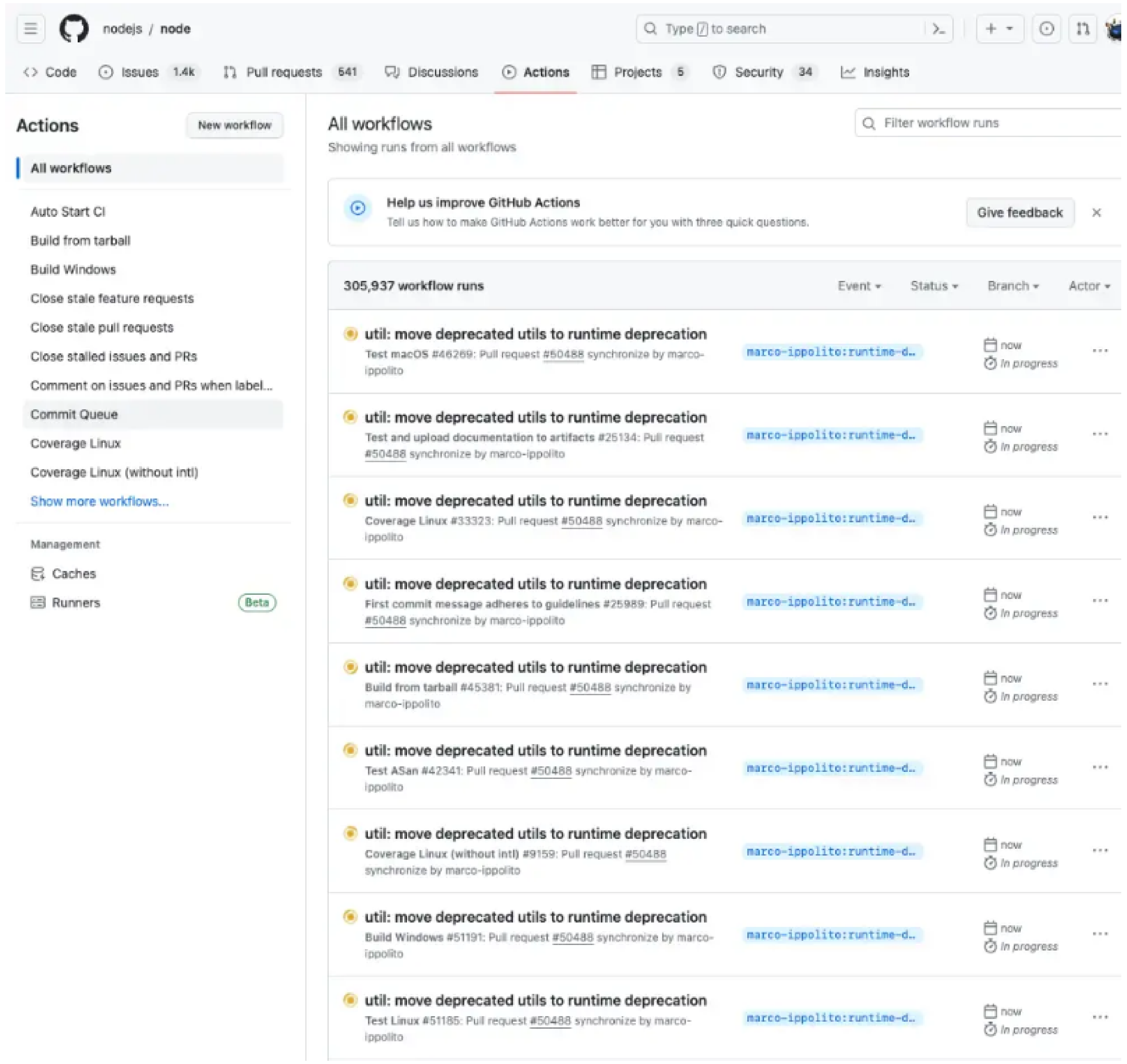
Semantic versioning

A specification for have meaningful software version.

Check the **official website**!

(Semi) Automated releasing

By leveraging existing tools and the workflow feature you can release new version with just a click.



All workflows

Showing runs from all workflows

Help us improve GitHub Actions
Tell us how to make GitHub Actions work better for you with three quick questions.

305,937 workflow runs

util: move deprecated utils to runtime deprecation Test macOS #46269: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation Test and upload documentation to artifacts #25134: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation Coverage Linux #33323: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation First commit message adheres to guidelines #25989: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation Build from tarball #45381: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation Test ASan #42341: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation Coverage Linux (without intl) #9159: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation Build Windows #51191: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress
util: move deprecated utils to runtime deprecation Test Linux #51185: Pull request #50488 synchronize by marco-ippolito	marco-ippolito:runtime-d..	now In progress

Questions?

Coffee Break

Humans are social beings

Let's talk about the community

Even if we're developing software, it's always about the people.

A line of of code can be replaced at any time with no effort, you can't do the same with people.

Where do you store the code?

It's better to keep things separate
The management will be simplified.

Public, separate organization are the best
They give you the visibility and ease of use you need.

Which accesses do you want to give?

Keep the accesses to the minimum
This is a general rule that should be follow as much as possible.

Make sure people cannot cut you out
It will be very hard to retake control.

Remember, you don't really own the code

It started with you ...
You own the software as whole.

... but it doesn't end with you ...
People are still owners and responsible for their contributions.

... so don't forget about others!
Contributors should have a say in relevant administrative tasks.

Diversity and Inclusion

Conscious and unconscious bias

We all suffer of these bias, there is simply no escape.

1

Conscious

We all have and use attitudes and heuristics.

2

Unconscious

More subtle because we don't control them.

Why do you want D&I?

Richness
Diversity is a pot of values.

New experiences
All people are having their lives ...

New perspectives
... and different point of views.

How do you enforce it? (1/2)

1

Be pro-active

You can't wait for things to happen to act.

2

Be receptive

Ask for suggestion from affected people.

How do you enforce it? (2/2)

3

Code of Conduct

This must be enforced from day zero.

4

Zero Tolerance

No exception allowed.

Everybody is responsible for moderation!

Last advices

1

Common sense

It might be surprising, but it is very underestimated and **never hurts**.

2

Yield control

You can't and neither want to control everything. **Don't burn out!**

3

Be genuine and honest

People will never fail to notice, appreciate and **love you**.

Questions?

Coffee Break

What about Node.js?

Questions?

One last thing™

*"Money is just a way to keep score.
The best people in any field are motivated
by passion. That becomes more true
the higher the skill level gets."*

Eric Raymond

Thank you!

Paolo Insogna

Node.js TSC, Principal Engineer

@p_insogna

paolo.insogna@platformatic.dev

Platformatic