View online

Download PDF



Love your maintainers

Paolo Insogna

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

The secret is in the manners!



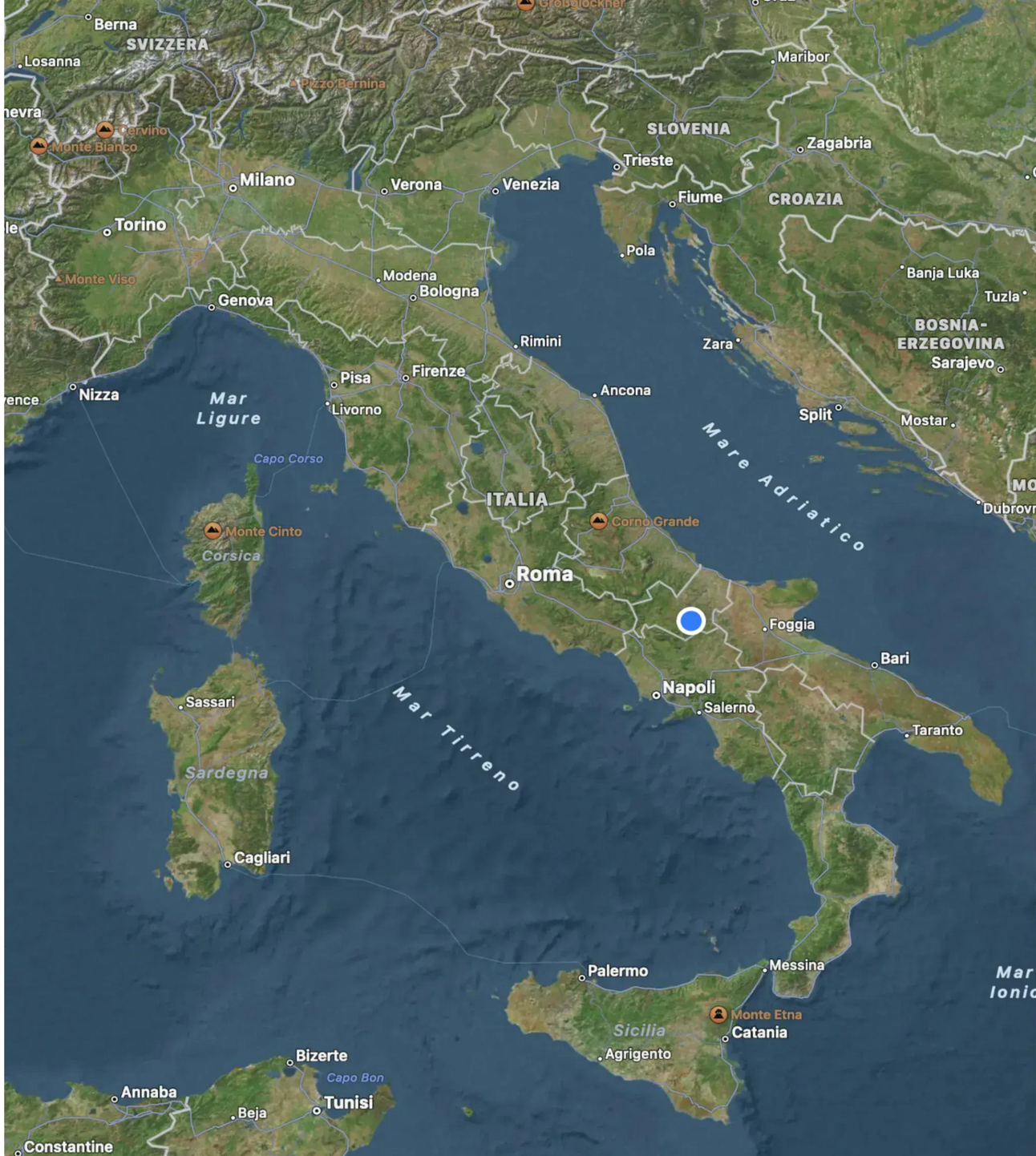
Hello, I'm **Paolo**!



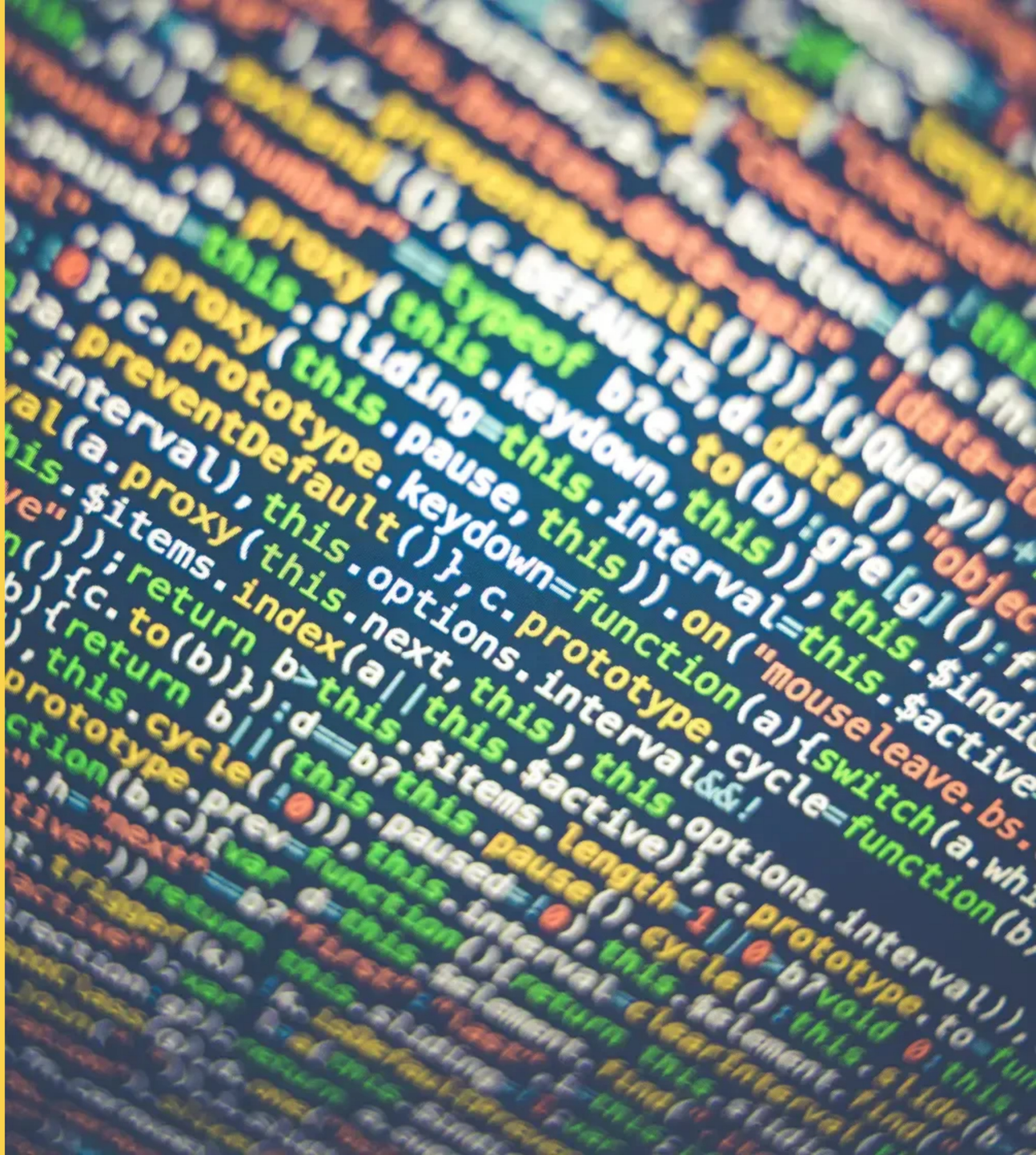
Node.js Technical Steering Committee Member

Platformatic Principal Engineer

`paoloinsogna.dev` `ShogunPanda` `p_insogna` `pinsogna`



What do you ~~expect~~ pretend from 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 did it start?



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.**

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, not the same!

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!



Everybody is nice if people are kind.

Passion
Maintainers love that you use their software.

Respect
They don't owe you anything. They are giving you their **free** time.

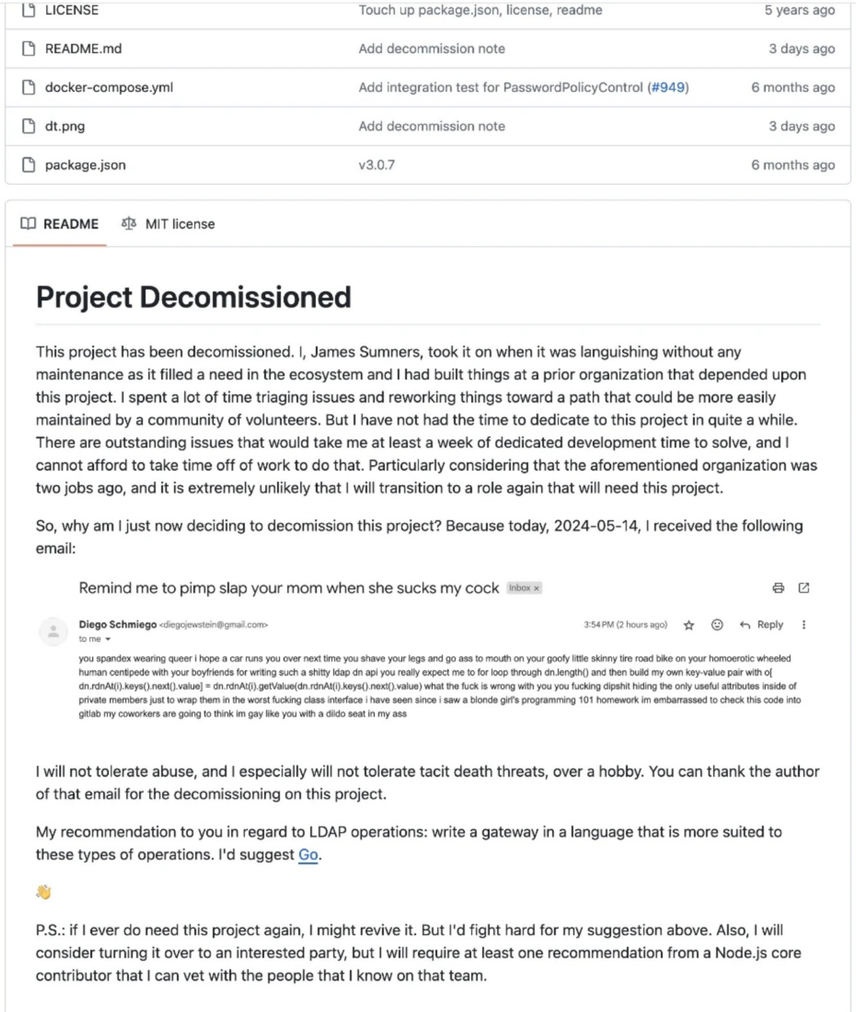
Actions have consequences
Demanding fixes, features or simply attention **will push** people away.



Avoid burnouts!

Needless to say, people have **sentiments** and a life in the real world.

Even a single small act of rudeness can **burn people out**.



LICENSE	Touch up package.json, license, readme	5 years ago
README.md	Add decommission note	3 days ago
docker-compose.yml	Add integration test for PasswordPolicyControl (#949)	6 months ago
dt.png	Add decommission note	3 days ago
package.json	v3.0.7	6 months ago

README MIT license

Project Decomissioned

This project has been decomissioned. I, James Sumners, took it on when it was languishing without any maintenance as it filled a need in the ecosystem and I had built things at a prior organization that depended upon this project. I spent a lot of time triaging issues and reworking things toward a path that could be more easily maintained by a community of volunteers. But I have not had the time to dedicate to this project in quite a while. There are outstanding issues that would take me at least a week of dedicated development time to solve, and I cannot afford to take time off of work to do that. Particularly considering that the aforementioned organization was two jobs ago, and it is extremely unlikely that I will transition to a role again that will need this project.

So, why am I just now deciding to decomission this project? Because today, 2024-05-14, I received the following email:

> Remind me to pimp slap your mom when she sucks my cock Inbox ×
>
> **Diego Schmiego** <diegojewstein@gmail.com> 3:54 PM (2 hours ago) Reply
> to me
>
> you spandex wearing queer i hope a car runs you over next time you shave your legs and go ass to mouth on your goofy little skinny tire road bike on your homoerotic wheeled human centipede with your boyfriends for writing such a shitty ldap dn api you really expect me to for loop through dn.length() and then build my own key-value pair with o[dn.rdnAt(i).keys().next().value] = dn.rdnAt(i).getValue(dn.rdnAt(i).keys().next().value) what the fuck is wrong with you you fucking dipshit hiding the only useful attributes inside of private members just to wrap them in the worst fucking class interface i have seen since i saw a blonde girl's programming 101 homework im embarrassed to check this code into gitlab my coworkers are going to think im gay like you with a dildo seat in my ass

I will not tolerate abuse, and I especially will not tolerate tacit death threats, over a hobby. You can thank the author of that email for the decomissioning on this project.

My recommendation to you in regard to LDAP operations: write a gateway in a language that is more suited to these types of operations. I'd suggest Go.

👋

P.S.: if I ever do need this project again, I might revive it. But I'd fight hard for my suggestion above. Also, I will consider turning it over to an interested party, but I will require at least one recommendation from a Node.js core contributor that I can vet with the people that I know on that team.

How do you ask for help?



It all starts with a detailed search

1. **Issues**
   The easier is to reproduce a problem, the higher the chance somebody already reported it.

2. **PRs**
   Similarly to the above, somebody might already be working on it.

3. **Code**
   Examples, tests or just the plain source code might already have the solution.

4. **Documentation**
   Known workaround or out of scope problems are usually well documented.

Don't worry about overcommunicating



We are distributed

Working across the globe introduced delays.



We are different

Developers come from different cultures and expertises.



Time is finite

We all have our lives. We don't have time to request missing informations.

It is like journalism

1. **Who and Where**
   This is usually implicit, but it's always better to be explicit.

2. **What and When**
   Explain what happened, what did not happen and what were you expecting.

3. **How**
   What is the smallest way to reproduce the problem? **This will return**.

Dude, you forgot the **Why**!

No, I have not ...



Narrow things down



Avoid flooding

Even if you don't worry to overcommunicate, you shouldn't just paste tons of output.



Keep focus

Reports with lots of superfluous information will likely distract the maintainer.



It's for your own good

The more superfluous information you give, the less the chance to see you problem solved.

Reproducible repos

Reduce to the minimum

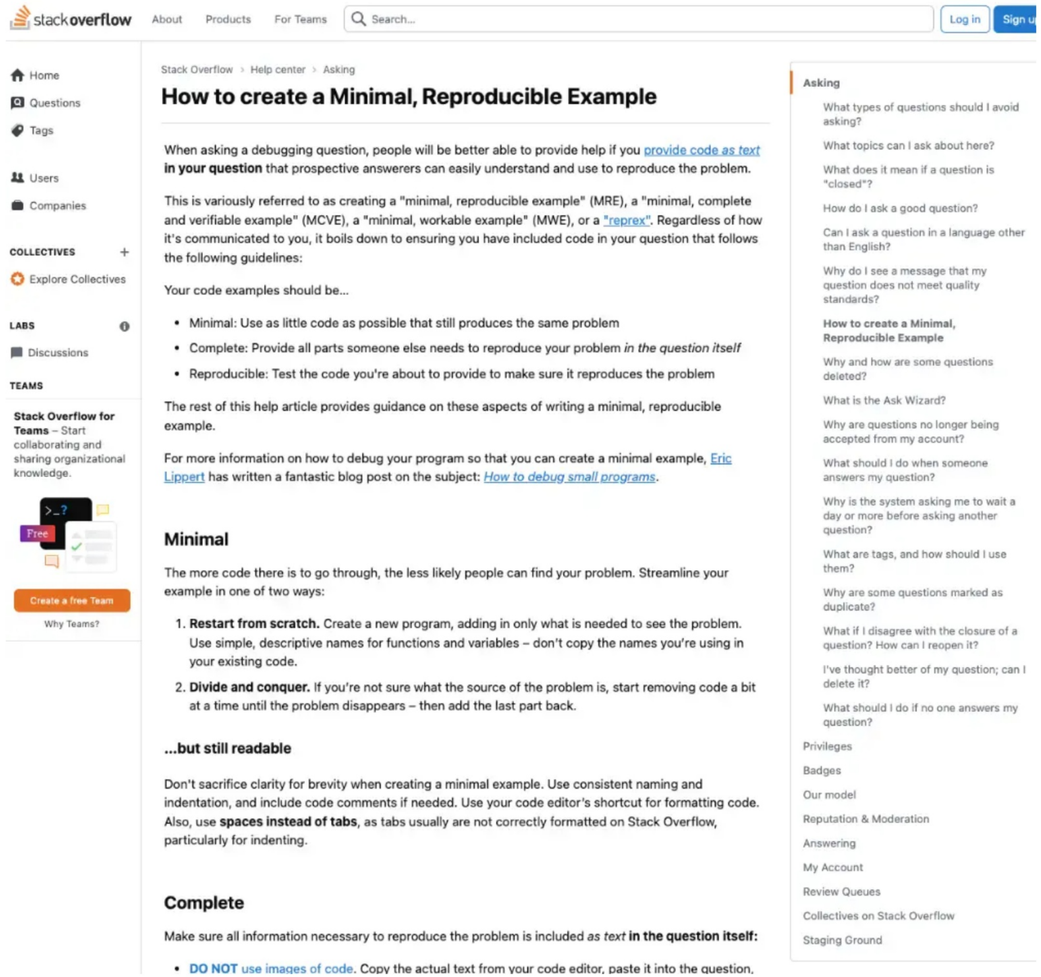
You want to rule out all irrelevant details.

Remove sensitive information

Nobody wants to get in trouble.

Provide a public repository

It will facilitate the triagers' work.



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.
Create a free Team
Why Teams?

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,

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

Follow up



Look out for clarification requests

The maintainer might need additional information in order to process the report.



Be heard

Following up on the report is your only way to give your feedback before it's too late.



Setup notifications

This is the easiest way to follow the discussion.

Contribute to the development



Engage
If you have the expertise you can propose to provide a PR.

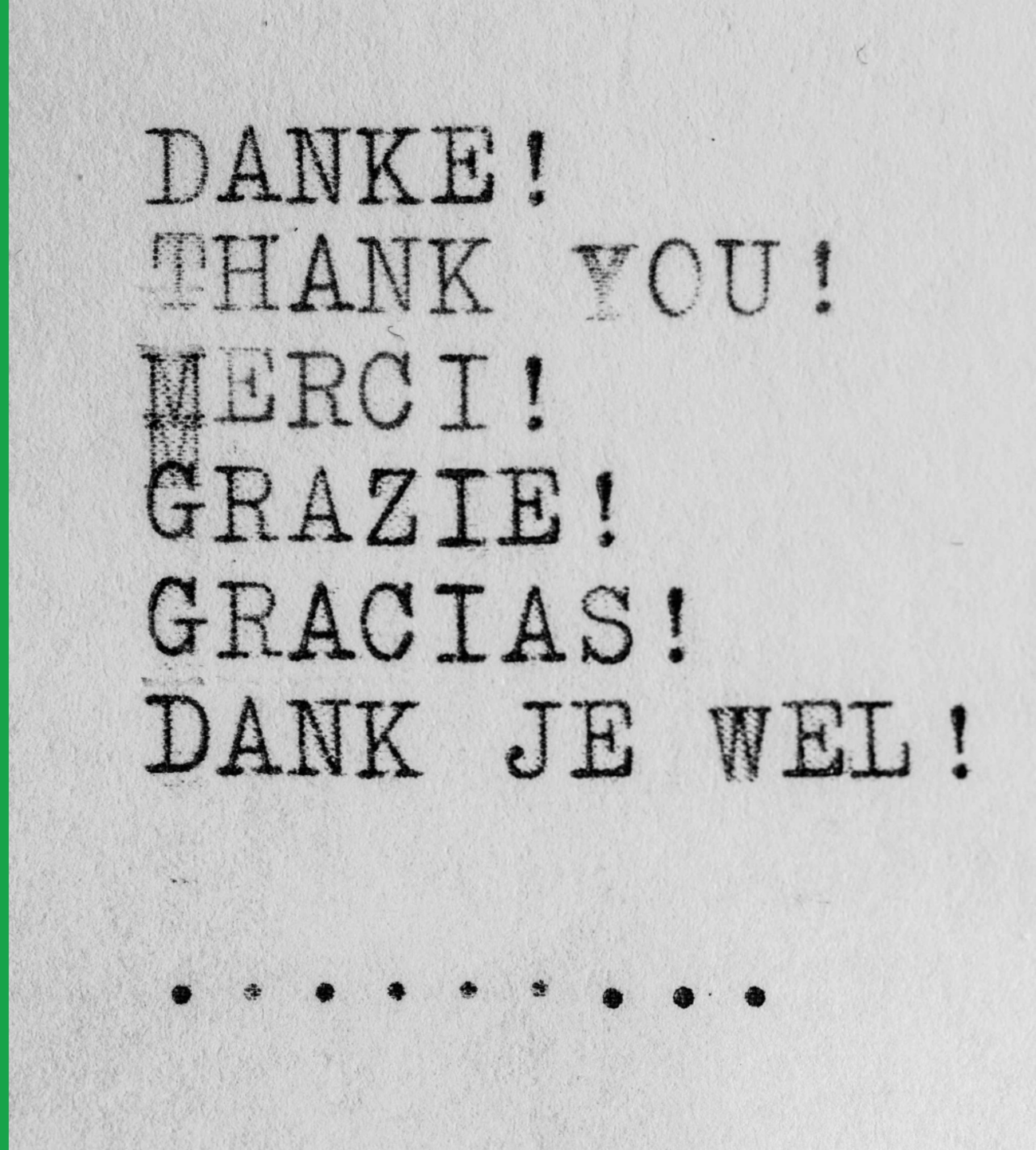


Test
Once the issue has been reported fixed, test it right away.



Give feedback
Let the maintainer know if the issue is solved.

And don't forget to thank them!



One last thing™

“No act of kindness, no matter how small, is ever wasted.”

Aesop





Thank you!

Paolo Insogna

Node.js TSC, Principal Engineer

@p_insogna

paolo.insogna@platformatic.dev

Platformatic