Road to Zero Lint Failures: Tackling Code Quality Challenges at Scale

Chris Ng

Sr. Staff Software Engineer at LinkedIn

@chrisrng

LinkedIn

Overview

1 Why lint rules?

2 Code quality at scale

3 How to fix it:
Road to zero lint failures

4 Key results and next steps

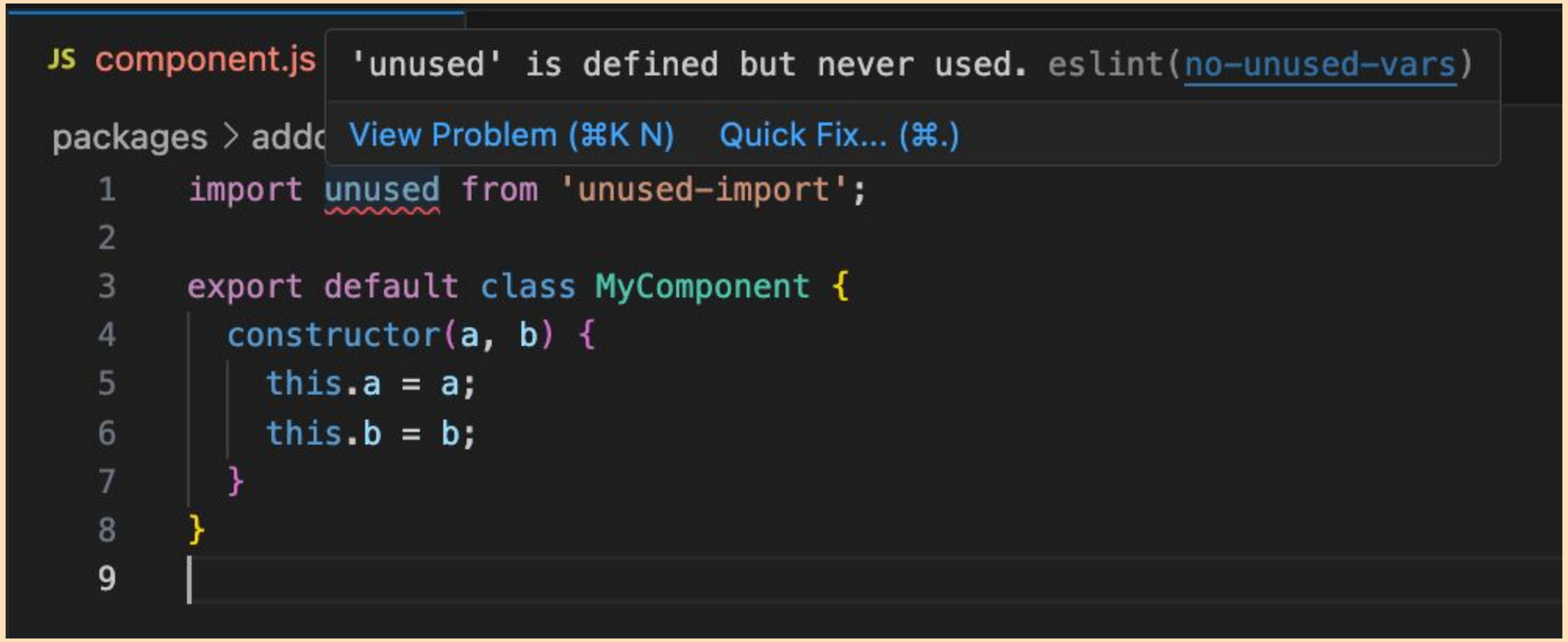
Why **lint rules**?

“If it's not in the docs, it ain't a real thing”

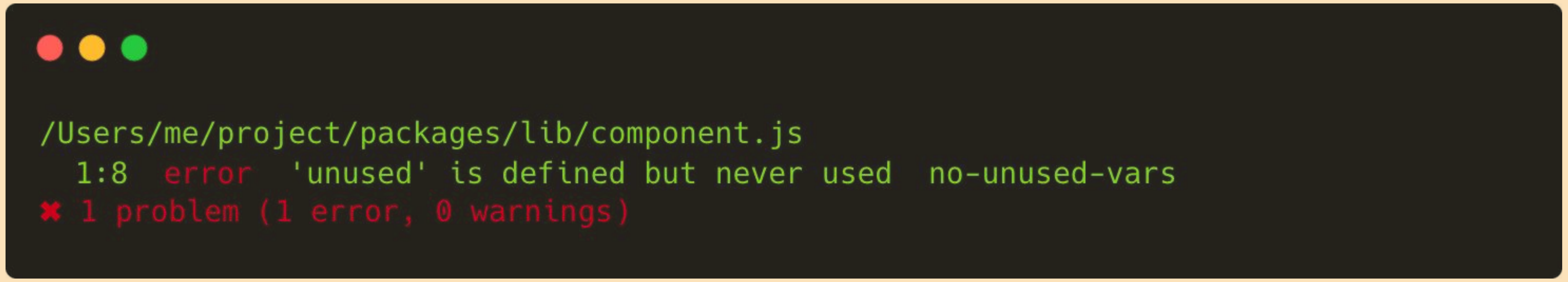
2017

Lint rules

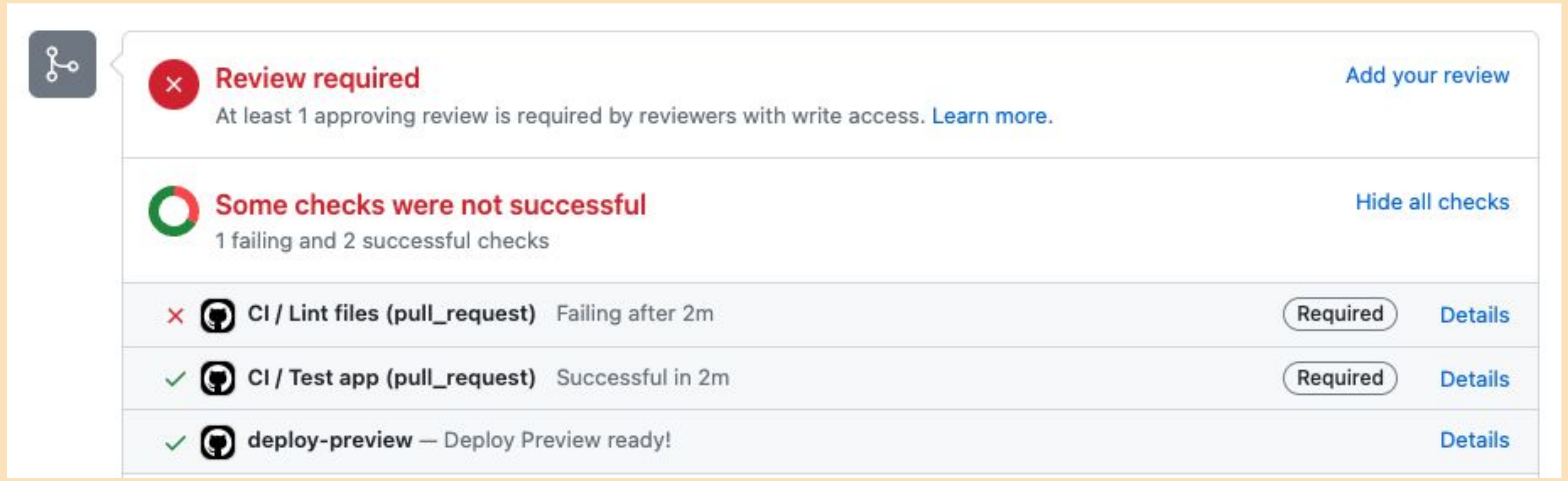
Automated code analysis tools to catch and prevent **errors** and **inconsistencies**.



```
JS component.js   'unused' is defined but never used. eslint(no-unused-vars)
packages > addc   View Problem (⌘K N)   Quick Fix... (⌘.)
import unused from 'unused-import';

export default class MyComponent {
  constructor(a, b) {
    this.a = a;
    this.b = b;
  }
}

```



```
/Users/me/project/packages/lib/component.js
  1:8  error  'unused' is defined but never used  no-unused-vars
✖ 1 problem (1 error, 0 warnings)
```



Review required — Add your review

At least 1 approving review is required by reviewers with write access. Learn more.

Some checks were not successful — Hide all checks

1 failing and 2 successful checks

✕	CI / Lint files (pull_request)	Failing after 2m	Required	Details
✓	CI / Test app (pull_request)	Successful in 2m	Required	Details
✓	deploy-preview — Deploy Preview ready!			Details

Automated

- During development
- Pre-commit
- Pre-merge

Prettier

No more arguments about formatting

Benefits: Errors and Inconsistencies

It’s more than just for style issues!

Consistency

- Enforce coding standards
- Streamline development
- Reduce bikeshedding

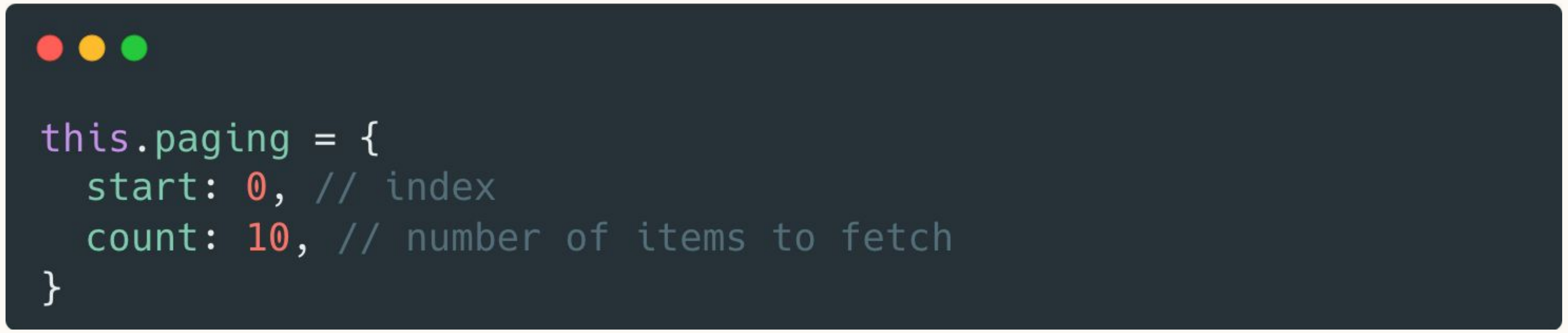
Bug Prevention

- Catch common mistakes
- Spread knowledge
- Identify issues earlier

Maintenance

- Deterministic patterns
- Easier migrations
- Easier refactors

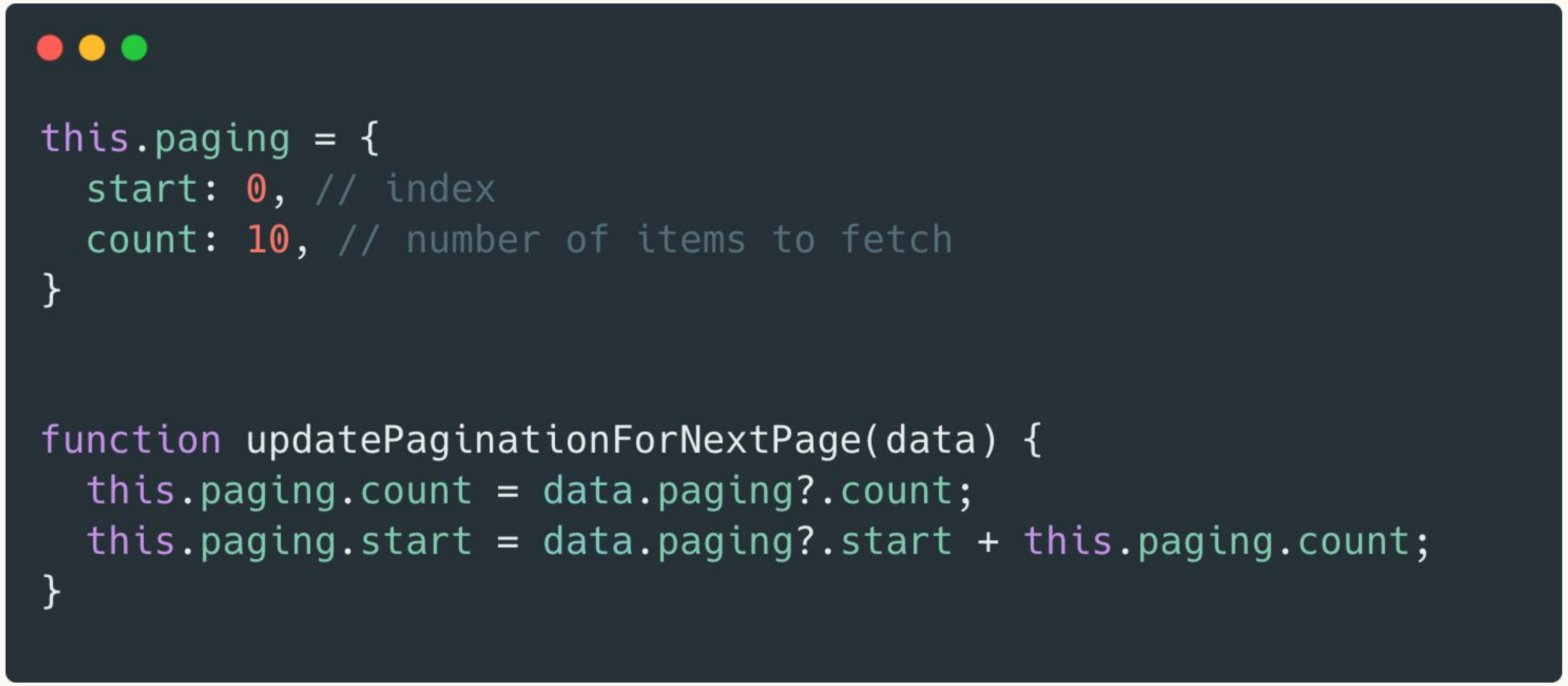
Example: no-unsafe-optional-chaining



```
this.paging = {
  start: 0, // index
  count: 10, // number of items to fetch
}
```

Sample Code Only

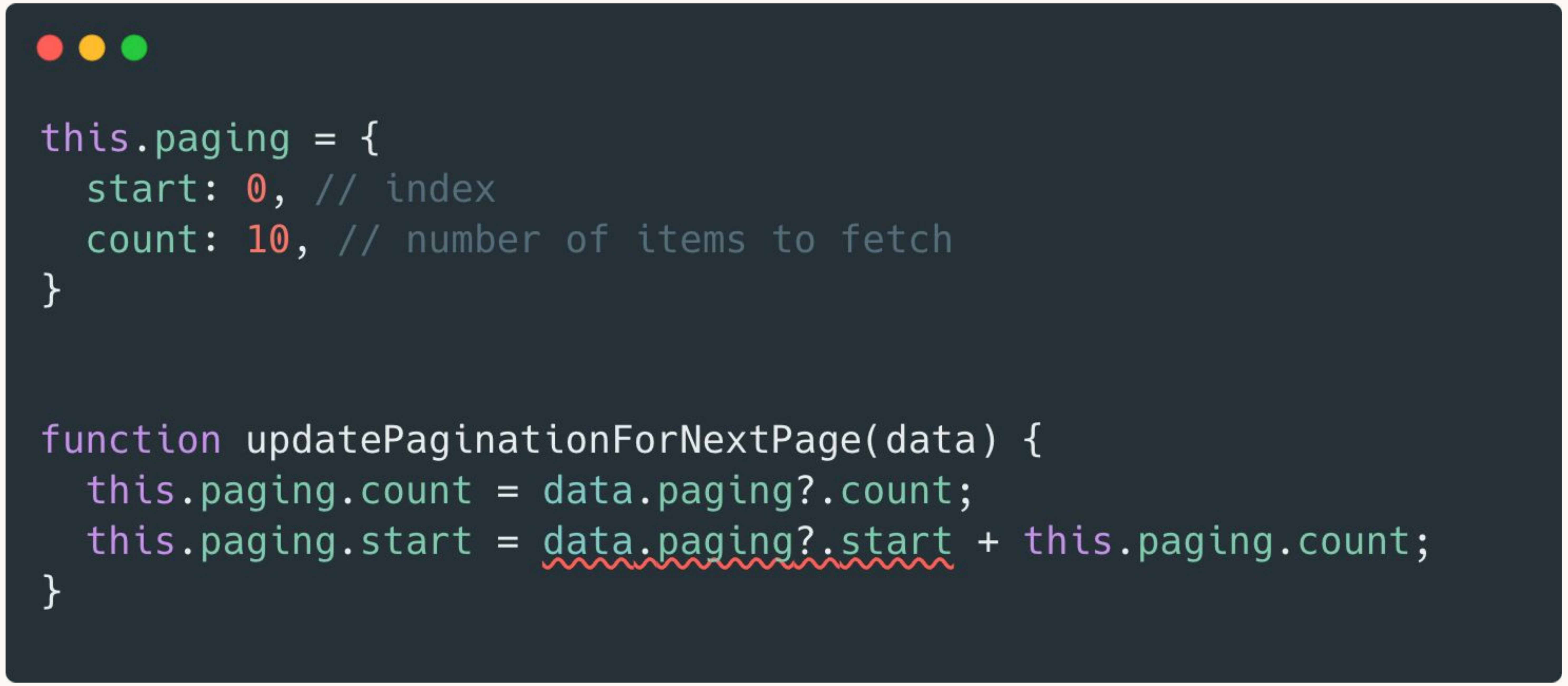
Example: no-unsafe-optional-chaining



```
this.paging = {
  start: 0, // index
  count: 10, // number of items to fetch
}


function updatePaginationForNextPage(data) {
  this.paging.count = data.paging?.count;
  this.paging.start = data.paging?.start + this.paging.count;
}
```

Sample Code Only

Example: no-unsafe-optional-chaining

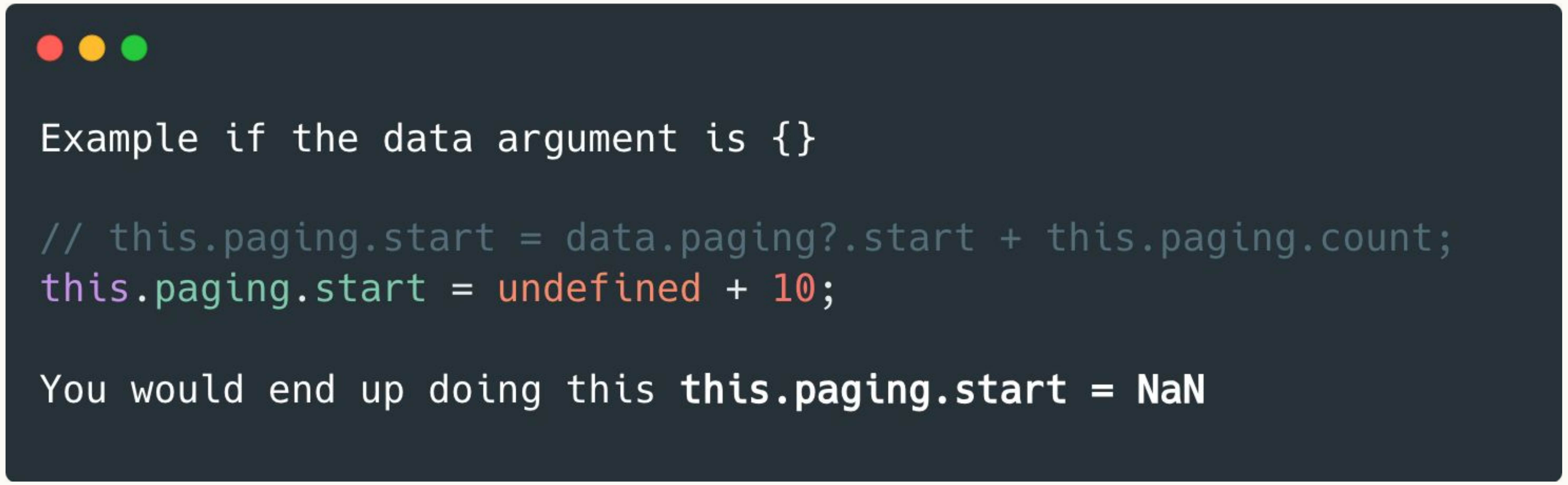


```
this.paging = {
  start: 0, // index
  count: 10, // number of items to fetch
}


function updatePaginationForNextPage(data) {
  this.paging.count = data.paging?.count;
  this.paging.start = data.paging?.start + this.paging.count;
}
```

Unsafe arithmetic operation on optional chaining. It can result in NaN.

Sample Code Only

Example: no-unsafe-optional-chaining



```
Example if the data argument is {}

// this.paging.start = data.paging?.start + this.paging.count;
this.paging.start = undefined + 10;

You would end up doing this this.paging.start = NaN
```

Edge cases always happen in a large enough application

Asking everyone to check for a condition isn't scalable

Sample Code Only

Expand the "WELL LIT PATH"

Make it easy to do the right thing.

LET'S ADD ALL THE LINT RULES!!!

Code quality **at scale**

What does “at scale” mean?

But it applies to projects of all sizes too!

8

Years old

For context, React was in v0.13 in 2015

24k

Files (app + test)

Over 100k total files in the repository

100k

Commits

Through the lifetime of the app

What does "at scale" mean?

But it applies to projects of all sizes too!

21%

YoY LOC Growth

Over 1M LOC!

~67

PRs per day

From humans not automated

800+

Unique contributors

With at least 5 commits

Problem

Code quality at scale

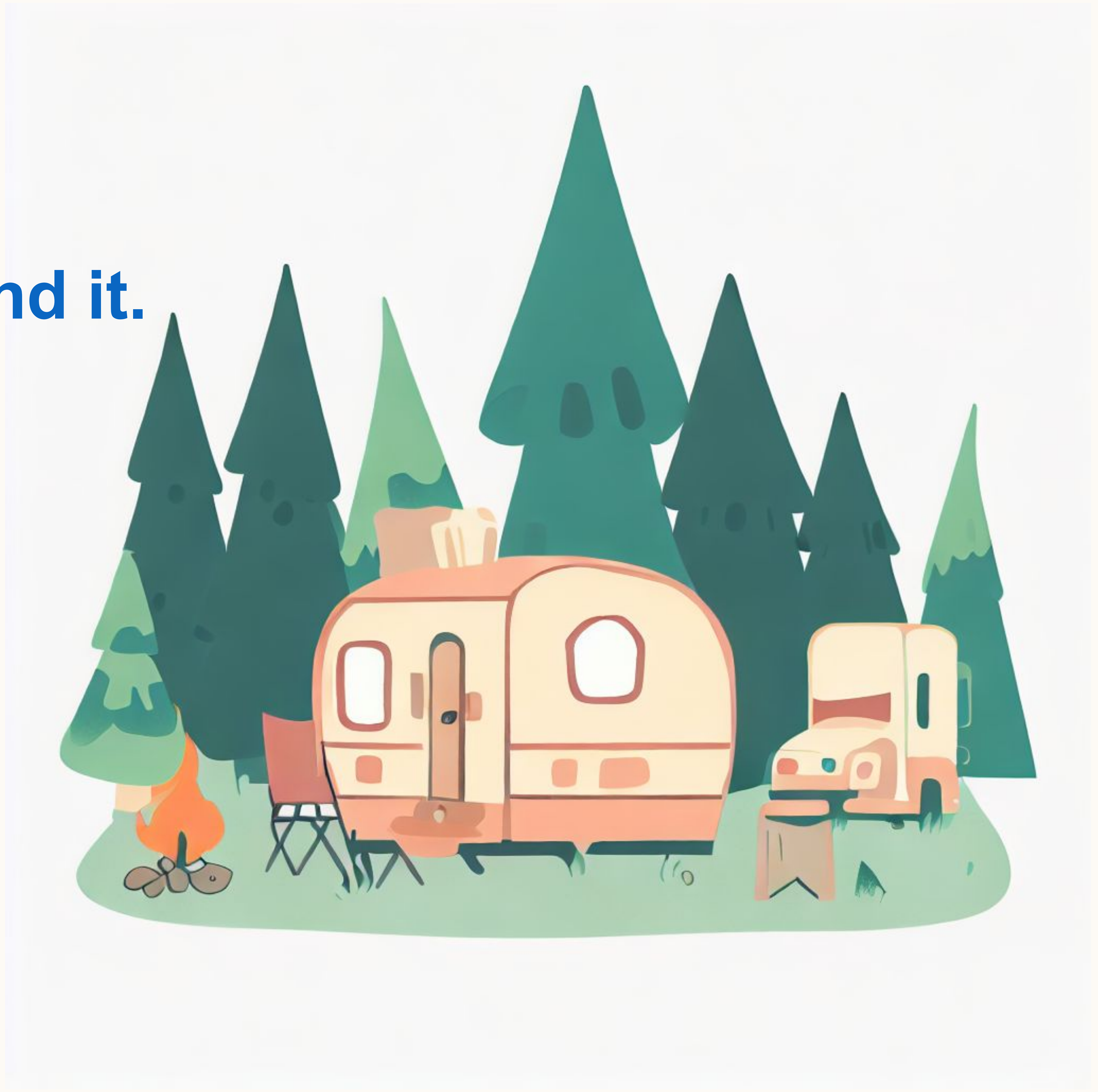
We had an **increasing number of lint failures** in the application that were not getting fixed.

When we started it was around **7000+ errors** already!

Campsite analogy

Ideally…

Leave the campsite better than you found it.



Campsite analogy

What if it <u>already</u> looks like this when you get there?

How to limit even more lint failures?

Spoiler alert: some worked, some did not work.

- **Blocking commits** when there is a lint failure in the changed file (warnings are allowed)
- **Setting limits** such as existing failures will not block merge (e.g. you cannot increase the number of failures)
- **"Platform review" system** where there is a specific group of reviewers with write access to enforce, spread, and maintain best practices

Campsite analogy (revisited)

🤔

What if you really really needed to land something …fast?

Shipped it!

(by avoiding or overriding)

How to limit even more lint failures? (revisited)

Stop the bleeding.

Lint Proposal Process

- **Democratize** the process
- **Rule proposal template** with answers to FAQ about new lint rules being added (e.g. Who, What, When, Where, and Why)
- **All existing failures must be resolved** before turning on a lint rule as error

Linting Task Force

- **Ownership** so there is a clearly defined set of individuals when an engineer hits a roadblock
- **Accountability** to provide guidance for engineers looking for help and improve code quality at scale
- **Consensus building** across all stakeholders but with a more targeted number of people who can give approval vs input

Lint rule process: All existing failures must be resolved

Before turning on a lint rule as **error** to ensure sustainable growth

- What if there are 1000 errors?
- What if I don't know how to fix the existing errors?
- I don't have time to do this!
- What if I break something?
- ***There are existing failures to files I am cleaning up!***

Not all problems are solved through engineering

Being a HUMAN helps!

How to fix it:
Road to **zero** lint failures

INCENTIVES, INCENTIVES, INCENTIVES!

Why would someone do something?

Road to zero lint failures

How to fix the messy campsite

Goals:

- **Accountable** code owners
- Provide a business case for **prioritization**
- **Actionable insight** for engineers
- **Tooling** that is actually used by engineers

Why?

- **Iterative improvement** without large scale disruption
- **Best practices** are codified
- **Engineering velocity**
- **Unpredictable cost** of fixing lint failures

How we ran it

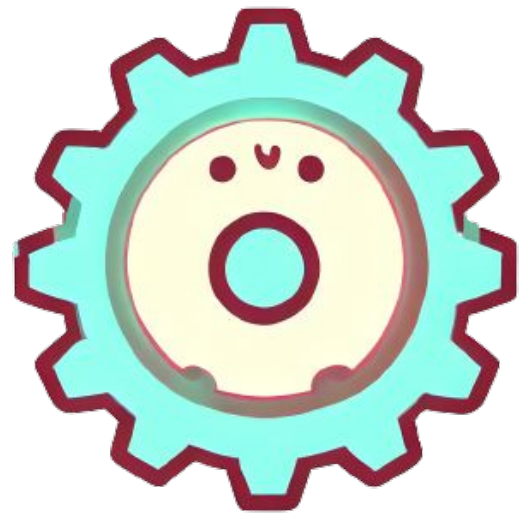
Shout outs

Technical Support

Developer Experience

Recognition

Visibility



Tooling for engineers

Road to zero lint failures

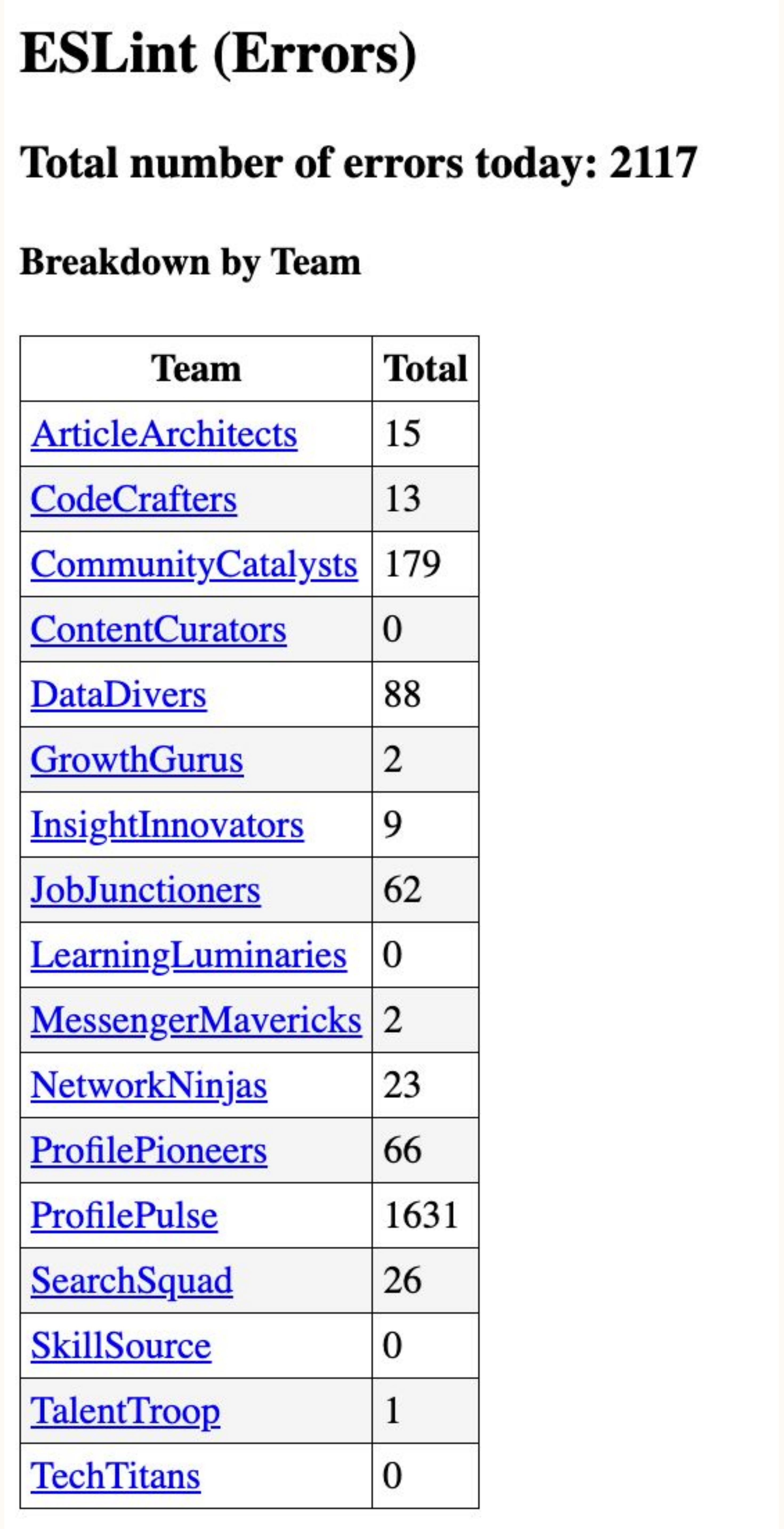
- Identify errors
- Identify owners
- Keep it up to date
- Make it simple

Single owner per file

Responsibility and accountability

Lint Failure Visualizer

MVP Version



ESLint (Errors)

Total number of errors today: 2117

Breakdown by Team

Team	Total
ArticleArchitects	15
CodeCrafters	13
CommunityCatalysts	179
ContentCurators	0
DataDivers	88
GrowthGurus	2
InsightInnovators	9
JobJunctioners	62
LearningLuminaries	0
MessengerMavericks	2
NetworkNinjas	23
ProfilePioneers	66
ProfilePulse	1631
SearchSquad	26
SkillSource	0
TalentTroop	1
TechTitans	0

Sample Data Only

Digestible chunks:

- Breakdown by team
- Breakdown by lint rule
- Breakdown by team and lint rule

MVP:

- Updated nightly via cron job
- Running off someone's VM
- Errors, warnings, disables

ESLint (Errors)

Total number of errors today: 2117

Breakdown by Lint Rule

Lint Rule	Total
no-unused-vars	9

List of Results

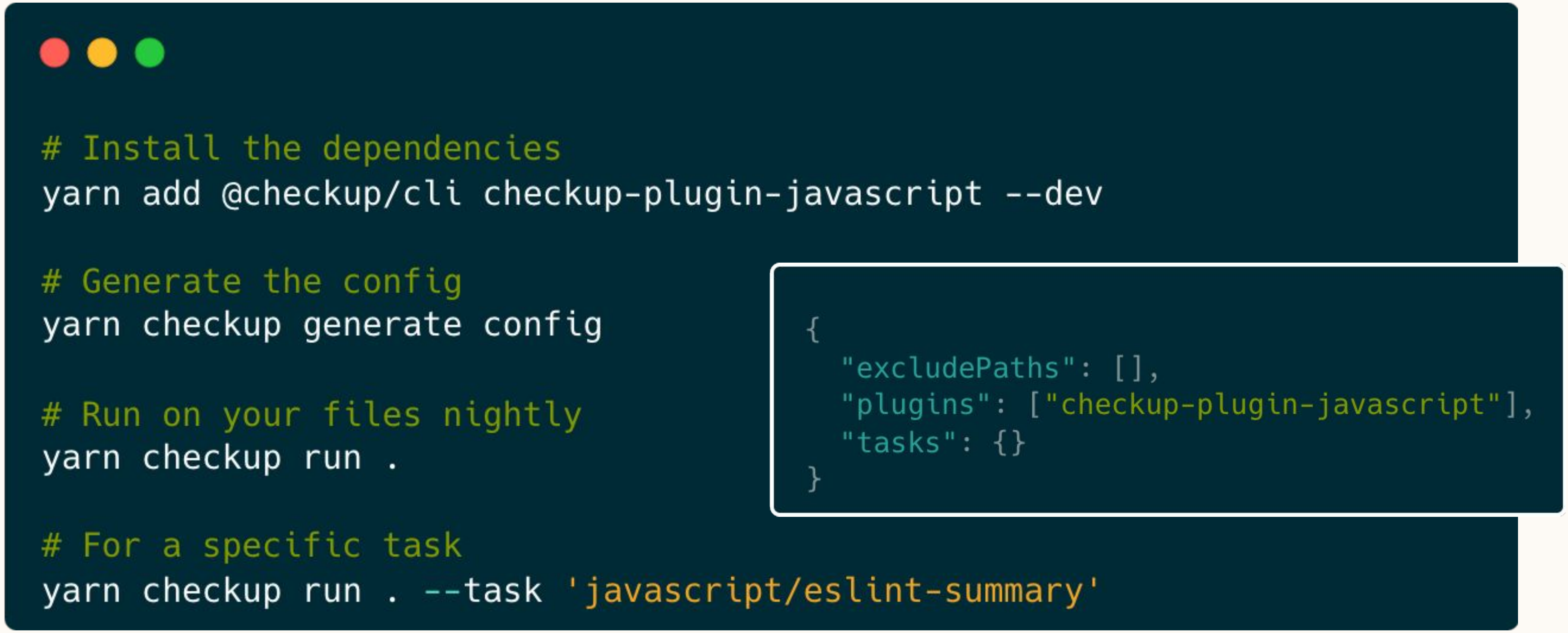
Number	Lint Rule	Level	Message	File Name	Line	Column
1	no-unused-vars	ERROR	'testDoSuperTest' is defined but never used.	packages/featured/tests/actions-list-test.js	104	10
2	no-unused-vars	ERROR	'test' is defined but never used.	packages/featured/tests/actions-list-test.js	7	18
3	no-unused-vars	ERROR	'get' is defined but never used.	packages/featured/components/actions-list.js	6	25
4	no-unused-vars	ERROR	'get' is defined but never used.	packages/featured/components/view-profile.js	3	10
5	no-unused-vars	ERROR	'get' is defined but never used.	packages/featured/components/identity-circle.js	5	10
6	no-unused-vars	ERROR	'test' is defined but never used.	packages/tests/unit/position-group-test.js	4	18
7	no-unused-vars	ERROR	'get' is defined but never used.	packages/article-reader/components/candidate.js	4	25
8	no-unused-vars	ERROR	'get' is defined but never used.	packages/articles/components/cards-list.js	3	10
9	no-unused-vars	ERROR	'get' is defined but never used.	packages/global/services/upload.js	6	10

Sample Data Only

github.com/checkupjs/checkup

Checkup

Setup nightly job



```
# Install the dependencies
yarn add @checkup/cli checkup-plugin-javascript --dev

# Generate the config
yarn checkup generate config

# Run on your files nightly
yarn checkup run .

# For a specific task
yarn checkup run . --task 'javascript/eslint-summary'
```

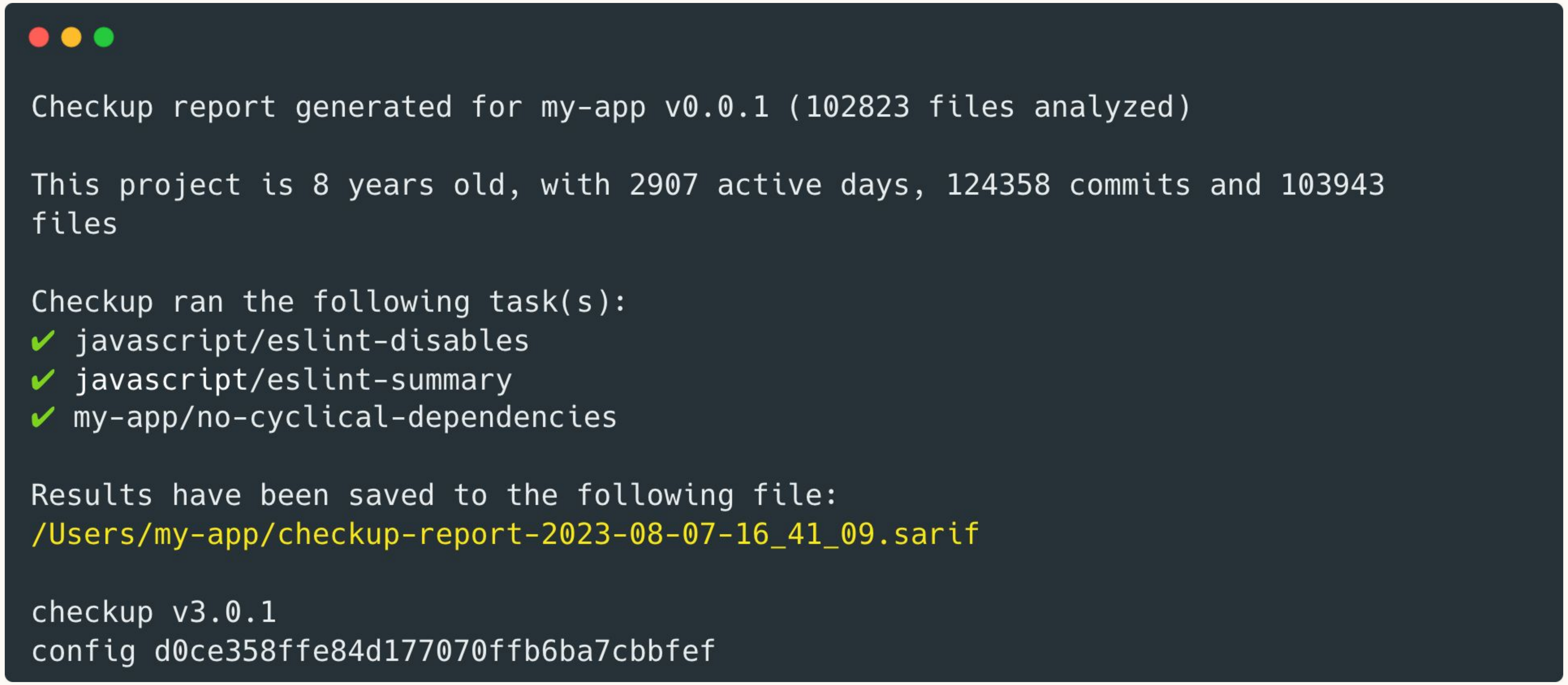
```
{
  "excludePaths": [],
  "plugins": ["checkup-plugin-javascript"],
  "tasks": {}
}
```

Sample Data Only

github.com/checkupjs/checkup

Checkup

Gather static analysis



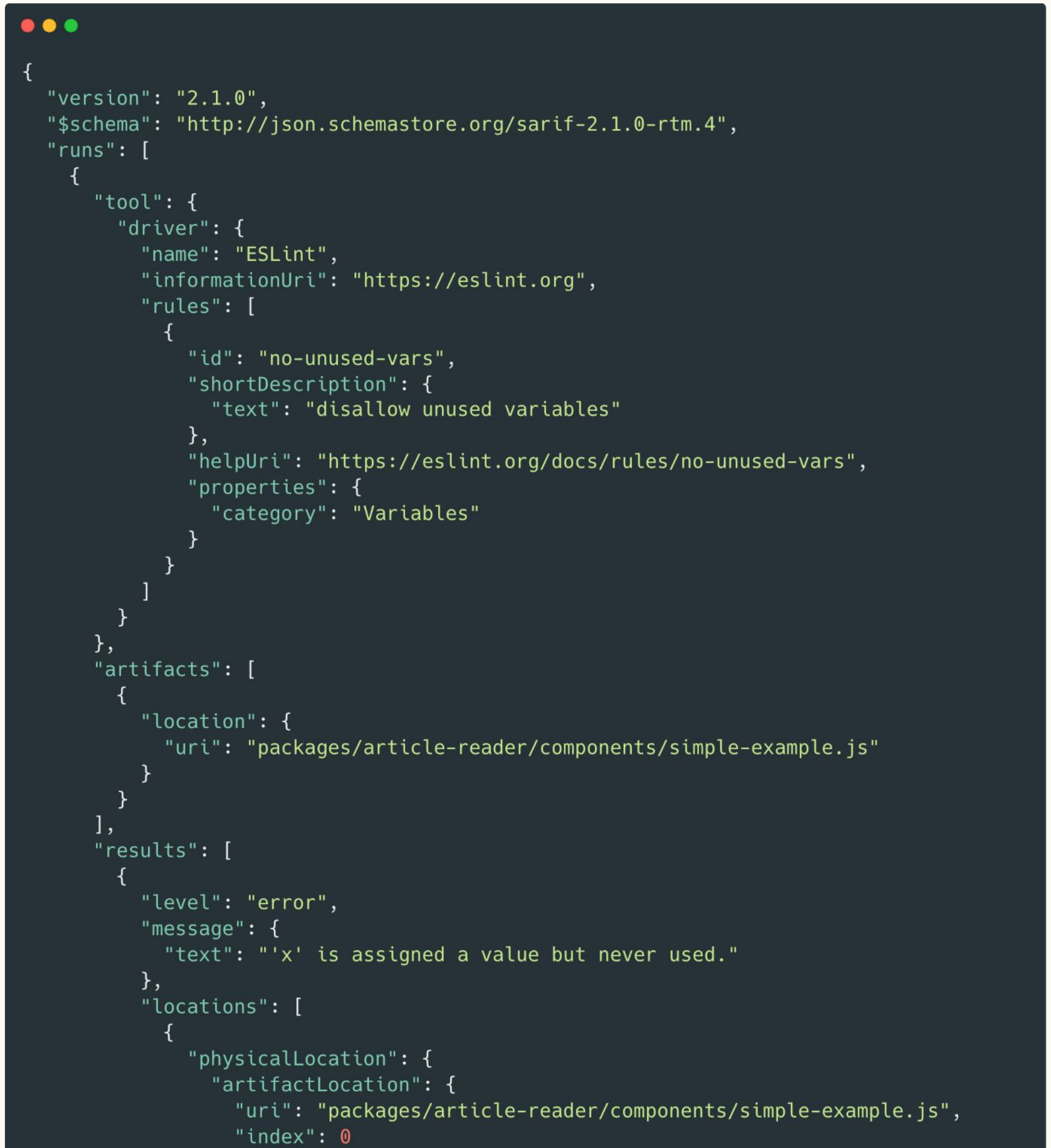
```
Checkup report generated for my-app v0.0.1 (102823 files analyzed)

This project is 8 years old, with 2907 active days, 124358 commits and 103943
files

Checkup ran the following task(s):
✔ javascript/eslint-disables
✔ javascript/eslint-summary
✔ my-app/no-cyclical-dependencies

Results have been saved to the following file:
/Users/my-app/checkup-report-2023-08-07-16_41_09.sarif

checkup v3.0.1
config d0ce358ffe84d177070ffb6ba7cbbfef
```

Sample Data Only

github.com/checkupjs/checkup

Checkup

Outputs a SARIF file

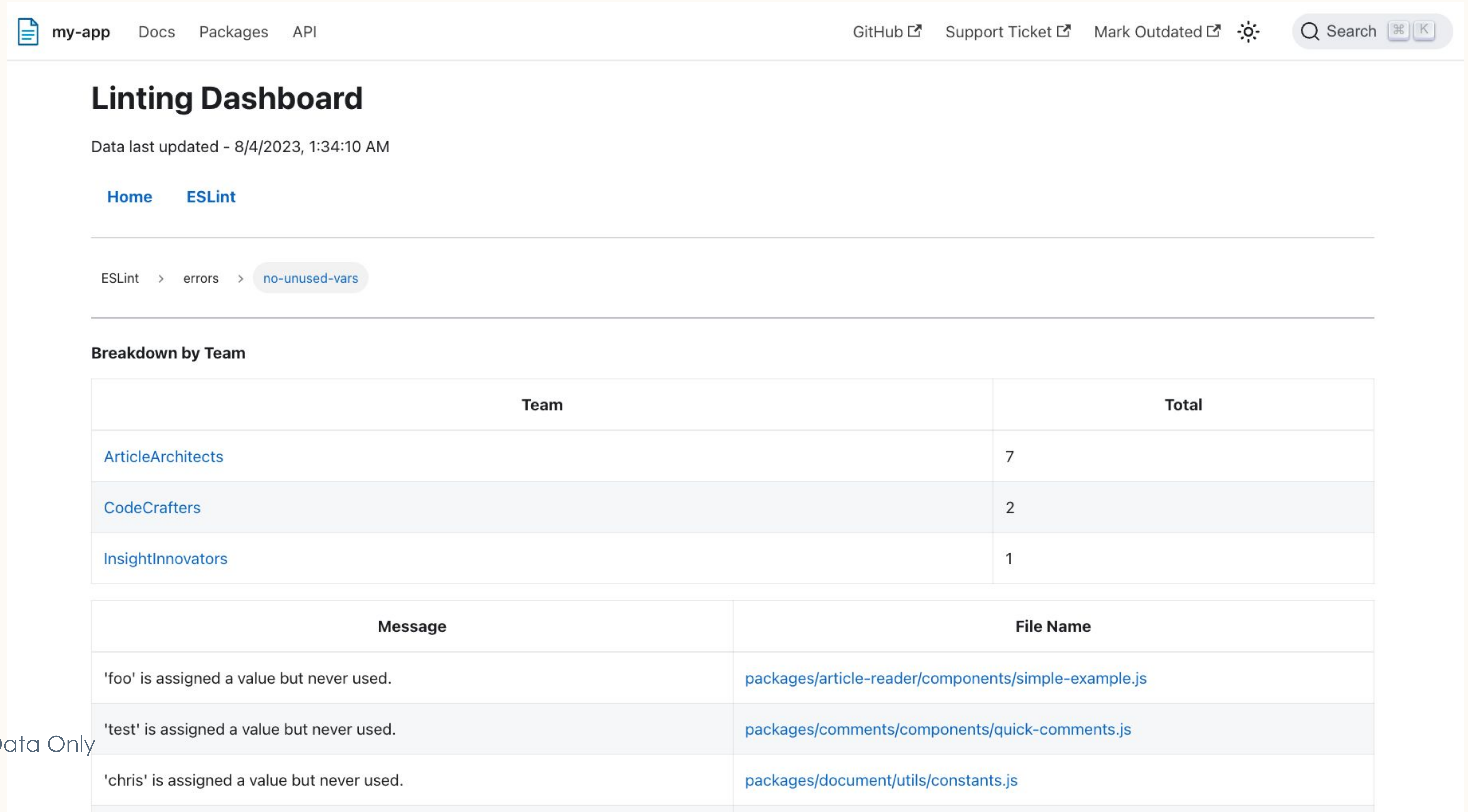


```
{
  "version": "2.1.0",
  "$schema": "http://json.schemastore.org/sarif-2.1.0-rtm.4",
  "runs": [
    {
      "tool": {
        "driver": {
          "name": "ESLint",
          "informationUri": "https://eslint.org",
          "rules": [
            {
              "id": "no-unused-vars",
              "shortDescription": {
                "text": "disallow unused variables"
              },
              "helpUri": "https://eslint.org/docs/rules/no-unused-vars",
              "properties": {
                "category": "Variables"
              }
            }
          ]
        }
      },
      "artifacts": [
        {
          "location": {
            "uri": "packages/article-reader/components/simple-example.js"
          }
        }
      ],
      "results": [
        {
          "level": "error",
          "message": {
            "text": "'x' is assigned a value but never used."
          },
          "locations": [
            {
              "physicalLocation": {
                "artifactLocation": {
                  "uri": "packages/article-reader/components/simple-example.js",
                  "index": 0
```

Sample Data Only

Collect data from SARIF and match by team

Oh yeah, new UI :D



Linting Dashboard

Data last updated - 8/4/2023, 1:34:10 AM

Home ESLint

ESLint › errors › no-unused-vars

Breakdown by Team

Team	Total
ArticleArchitects	7
CodeCrafters	2
InsightInnovators	1

Message	File Name
'foo' is assigned a value but never used.	packages/article-reader/components/simple-example.js
'test' is assigned a value but never used.	packages/comments/components/quick-comments.js
'chris' is assigned a value but never used.	packages/document/utils/constants.js

Sample Data Only

Weekly scorecard

Broken down per team

Green 🎉	Negative WoW increase in lint failures OR 0 total lint failures for the team
Yellow 😐	Zero (0) WoW increase/decrease in lint failures
Red 🤔	Positive Wow increase in lint failures

Weekly scorecard

- Scorecard updated **weekly**
- Identify 2-3 engineers who fixed big issues and give them a **shoutout** with a link to the PR
- **Manual email**

Team	Total Failure Count
ArticleArchitects	55
CodeCrafters	0
CommunityCatalysts	10
DataDivers	43
GrowthGurus	19
InsightInnovators	8
JobJunctioners	26
LearningLuminaries	15

Sample Data Only

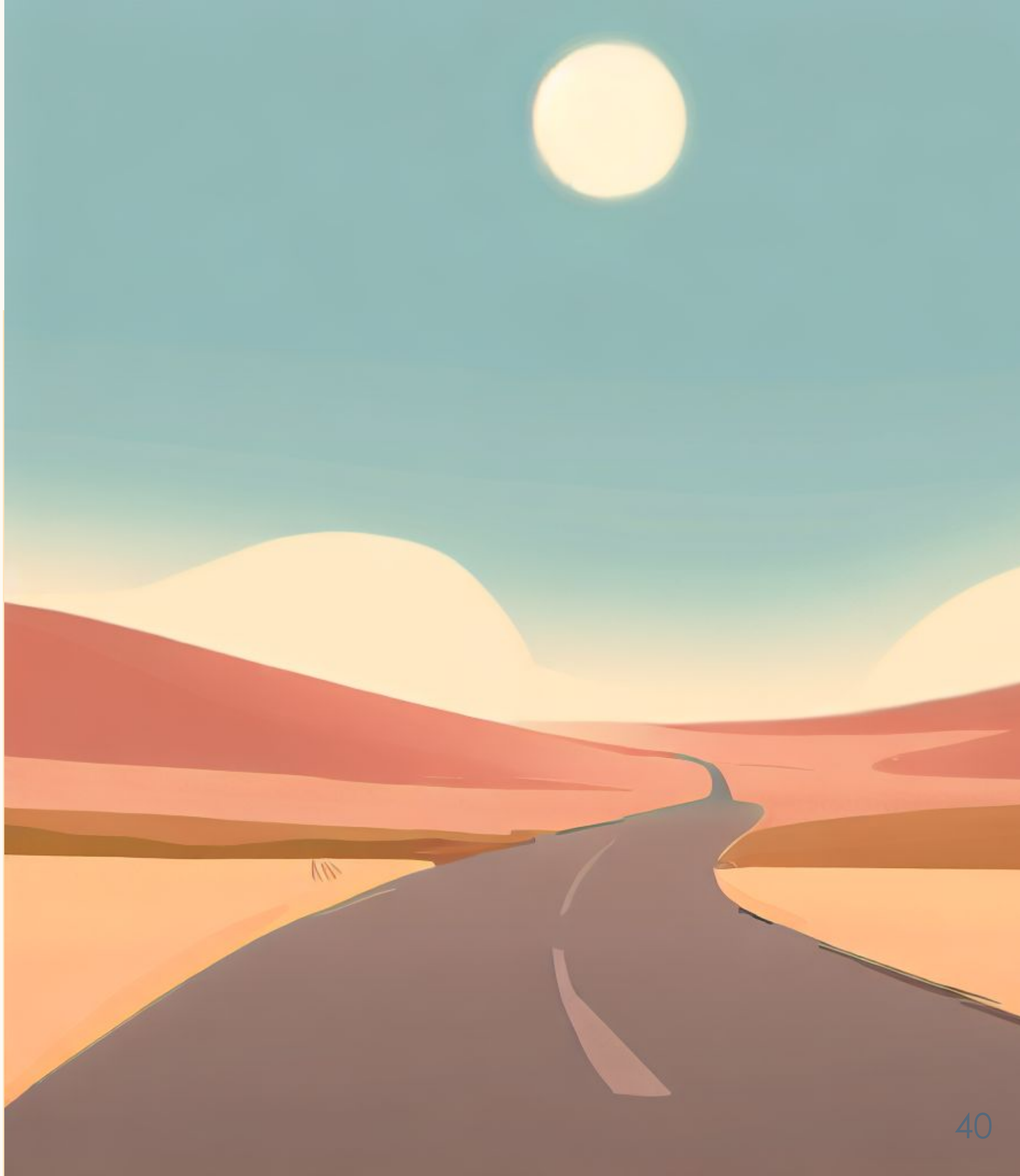
Dealing with the last mile

80-20 Rule

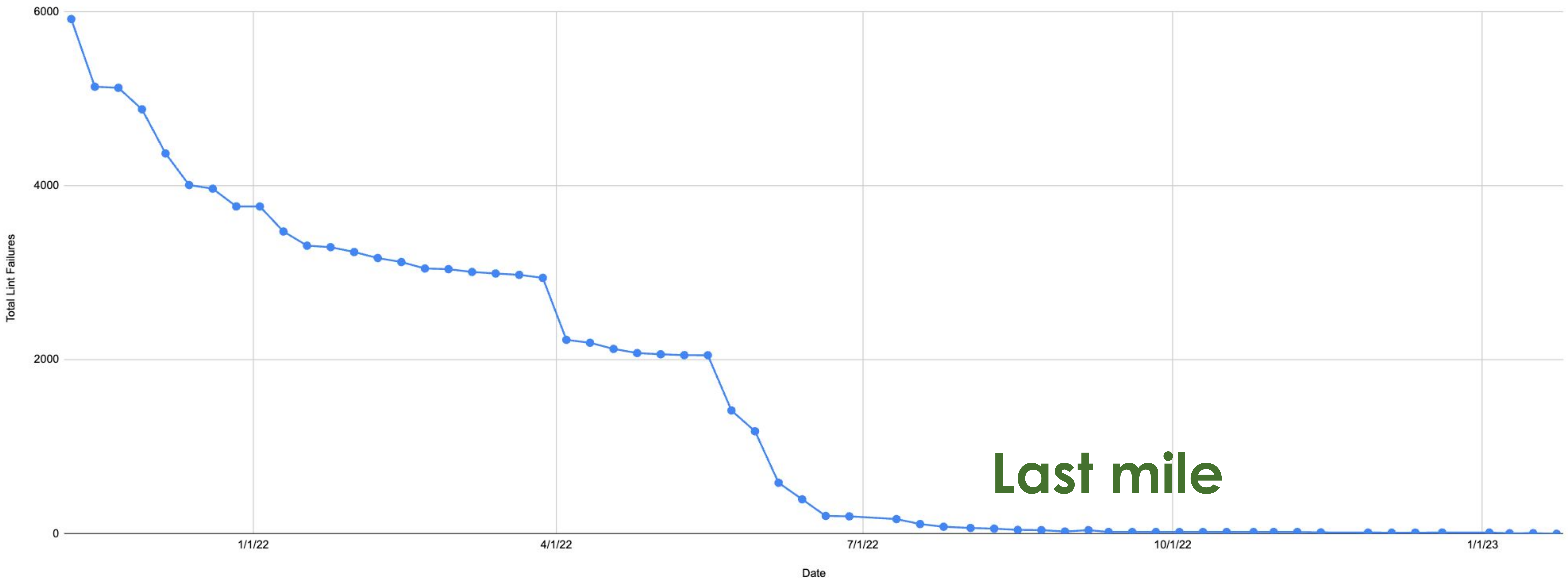
Getting to zero

- Identify stragglers
- Team effort
- Ask for help
- Leave no code left behind
- Get your hands dirty

→ “DoEs iT rEaLLY MaTTer?”

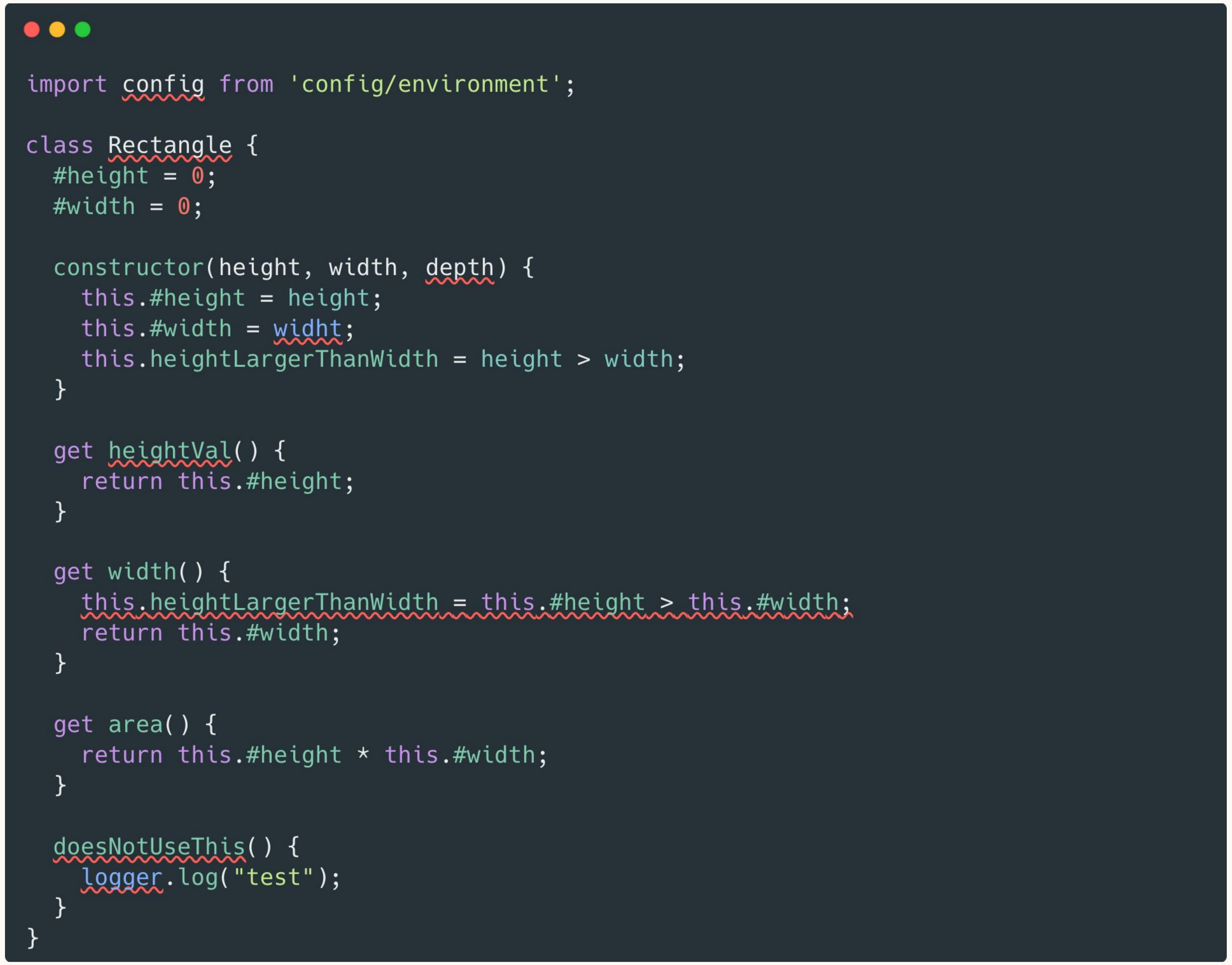


Road to zero journey



Sustainability is key

Yes the last mile matters!

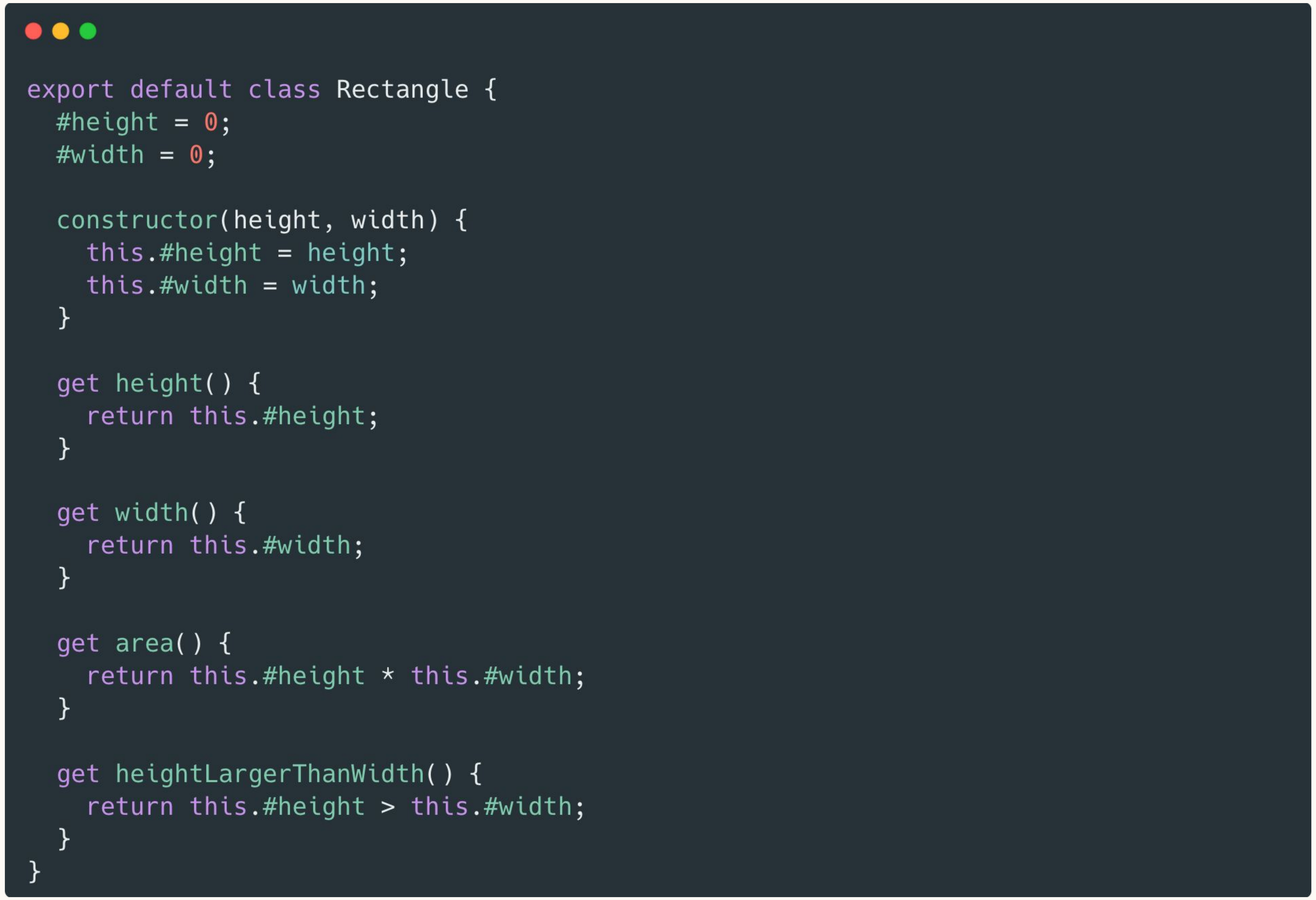


```
import config from 'config/environment';

class Rectangle {
  #height = 0;
  #width = 0;

  constructor(height, width, depth) {
    this.#height = height;
    this.#width = widht;
    this.heightLargerThanWidth = height > width;
  }

  get heightVal() {
    return this.#height;
  }

  get width() {
    this.heightLargerThanWidth = this.#height > this.#width;
    return this.#width;
  }

  get area() {
    return this.#height * this.#width;
  }

  doesNotUseThis() {
    logger.log("test");
  }
}
```

Sample Code Only

Sustainability is key

Accountable code owners



```
export default class Rectangle {
  #height = 0;
  #width = 0;

  constructor(height, width) {
    this.#height = height;
    this.#width = width;
  }

  get height() {
    return this.#height;
  }

  get width() {
    return this.#width;
  }

  get area() {
    return this.#height * this.#width;
  }

  get heightLargerThanWidth() {
    return this.#height > this.#width;
  }
}
```

Sample Code Only

Summary

Road to zero lint failures

1. No new errors
2. Single owner per file
3. Cron job using Checkup
4. Dashboard identifying issues
5. Weekly scorecard
6. Recognition for fixes
7. Get your hands dirty

Key results and next steps

Lessons learned

Road to zero lint failures

Identify who is doing the work?

- Focus on individual DevEx
- Personal shout outs
- No regressions

Road to zero lint failures stats

Between November 2021 and January 2023

5915

Lint failures removed

437

Days to complete

55

Unique contributors

+30%

Code quality NSAT score
(NSAT = net satisfaction)

Impact of the linting initiatives including Lint Proposal Process, Linting Task Force, Road to Zero Lint Failures, etc.

80+

Lint rule changes (or lint dep)

45+

Unique lint rule contributors

“If it's not in the docs, it ain't a real thing”

2017

“If it's not a **lint rule**, it ain't a real thing”

2023

Thank you

@chrisrng