



TRUSTING THE ROBOT

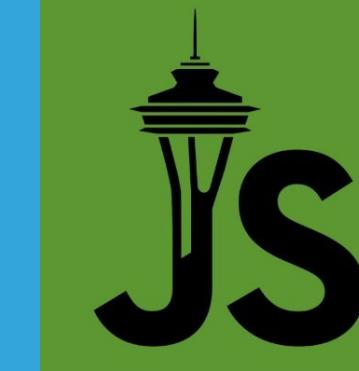
USING THE POWER OF GIT

TO AUTOMATE YOUR RELEASES



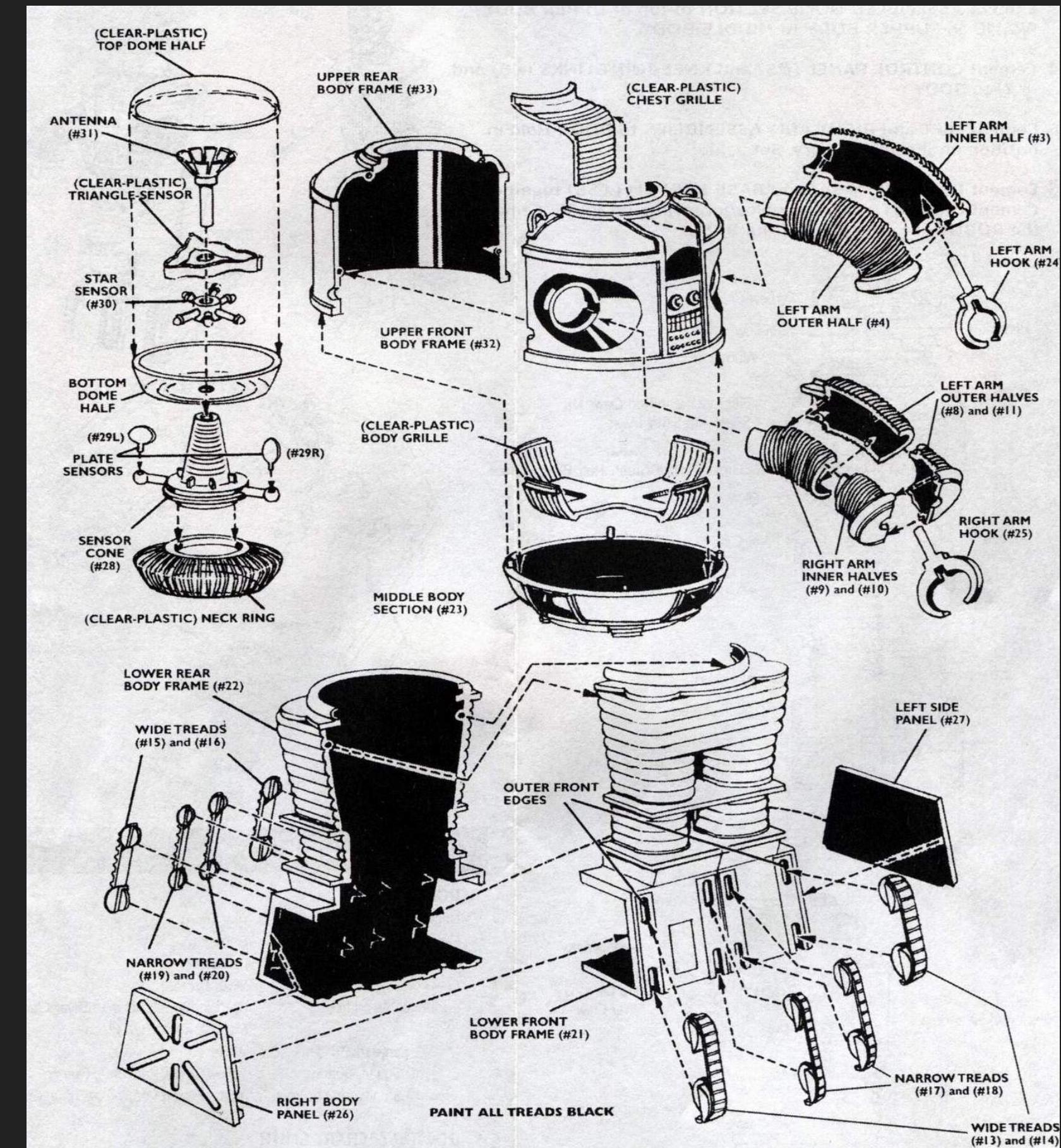
BRENDEN NIEDERMEYER

**Lead Software Engineer
EF Education First**



AGENDA

COMMUNICATING CHANGE
STANDARDIZING CHANGE
AUTOMATING CHANGE



STORY TIME

STORY TIME

COMMUNICATING CHANGE

COMMUNICATING CHANGE

New Features

General Maintenance

Bug Fixes

Refactor

Dev Quality of Life

Breaking Changes

Documentation Updates

COMMUNICATING CHANGE

New Features

General Maintenance

Bug Fixes

Refactor

Dev Quality of Life

Breaking Changes

Documentation Updates

**YOUR CHANGES ARE
DOCUMENTATION**

**YOUR COMMITS ARE
DOCUMENTATION**

**COMITS PROVIDE MEANING
AND HISTORICAL CONTEXT**



git log --oneline --decorate

```
51402cc (HEAD -> main) latest changes
aa0d463 chagnes
0d341af updates
41e9365 please let this one work
e5ba2b7 more fixes
7eaf6e1 fix
320feff fix bug in new feature
d5798c3 add new feature
f48ce80 updates
7a4c26b latest changes
3a17261 latest changes
dc09eba initial commit
(END)
```



Jeff Lembeck

@jefflembeck

...

The body of your commit message is for the "why".

INCLUDE THE WHY.

PLEASE ALWAYS INCLUDE THE WHY.

I CAN READ YOUR CODE I NEED TO KNOW WHY YOU
DID THAT.

I NEED TO KNOW WHY I DID THAT.

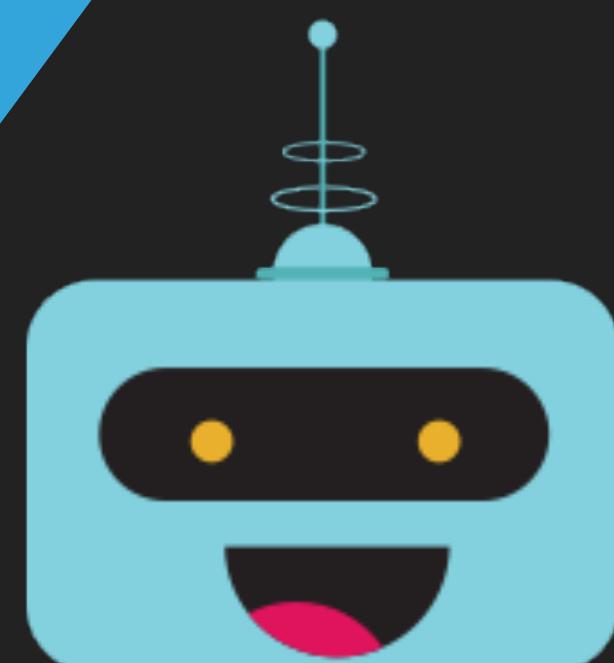
PLEASE TAKE MERCY ON ME, PAST SELF. I DON'T
UNDERSTAND WHY I MADE THESE CHOICES.

3:01 PM · Apr 18, 2019 · Twitter Web Client

1,086 Retweets 84 Quote Tweets 4,345 Likes



OK, ADDING MORE DETAILS IS
IMPORTANT. CAN WE STANDARDIZE
WHAT DETAILS WE INCLUDE?



STANDARDIZING CHANGE

Conventional Commits

A specification for adding human and machine readable meaning to commit messages

[Quick Summary](#)

[Full Specification](#)

[Contribute](#)



<type>[optional scope]: <description>

[optional body]

[optional footer(s)]

<type>[optional scope]: <description>

[optional body]

[optional footer(s)]

<type>[optional scope]: <description>

[optional body]

[optional footer(s)]

Type
Scope

feat(auth): add new auth interceptor ← Description

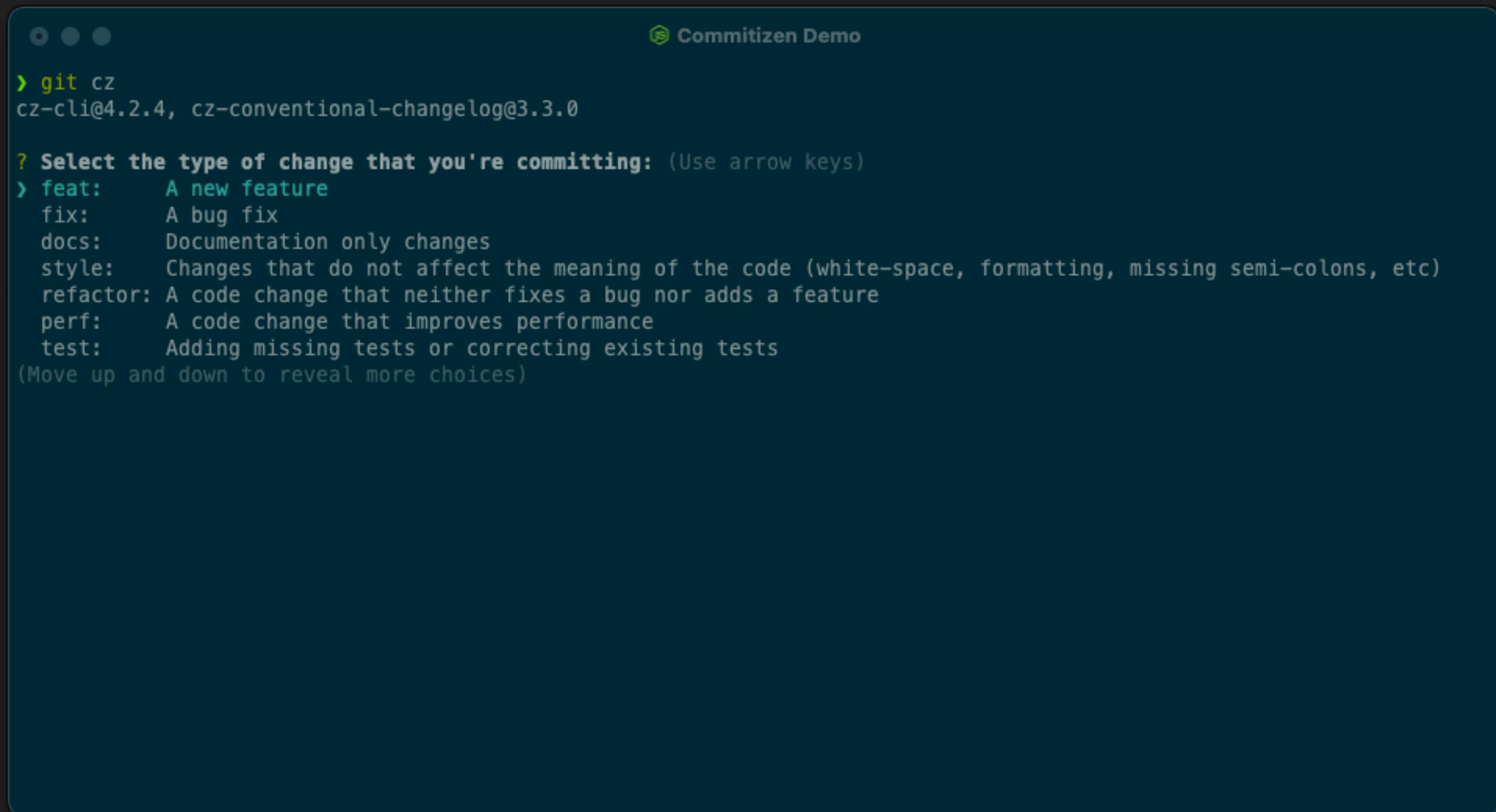
Body

Adds a new http interceptor to authenticate requests with our new auth system via the auth service added in #340. This is disabled behind a flag until we are ready to migrate from the legacy auth system to the new one.

Implements #342 ← Footer

TOOLS TO HELP EASE ADOPTION OF CONVENTIONAL COMMITS

Commitizen - CLI to help format commits



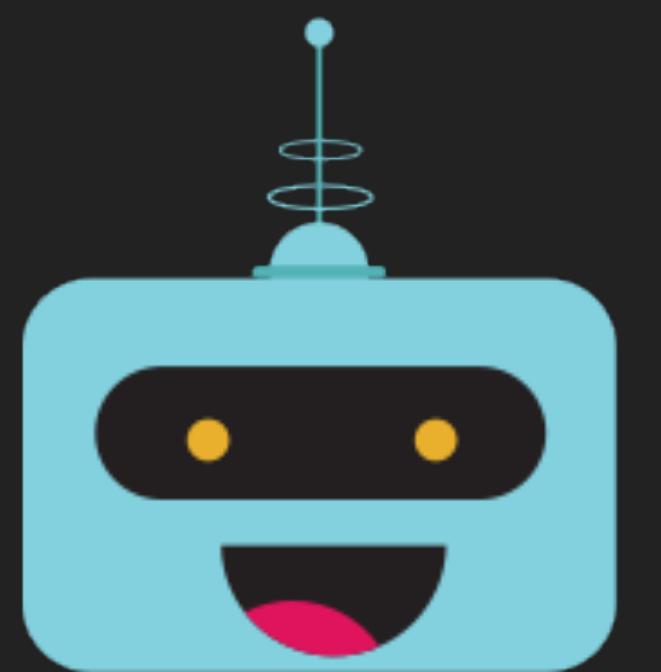
The screenshot shows a terminal window with the title "Commitizen Demo". The command "git cz" is run, followed by "cz-cli@4.2.4, cz-conventional-changelog@3.3.0". A prompt asks to "Select the type of change that you're committing: (Use arrow keys)". Below the prompt is a list of commit types with their descriptions:

- > feat: A new feature
- fix: A bug fix
- docs: Documentation only changes
- style: Changes that do not affect the meaning of the code (white-space, formatting, missing semi-colons, etc)
- refactor: A code change that neither fixes a bug nor adds a feature
- perf: A code change that improves performance
- test: Adding missing tests or correcting existing tests

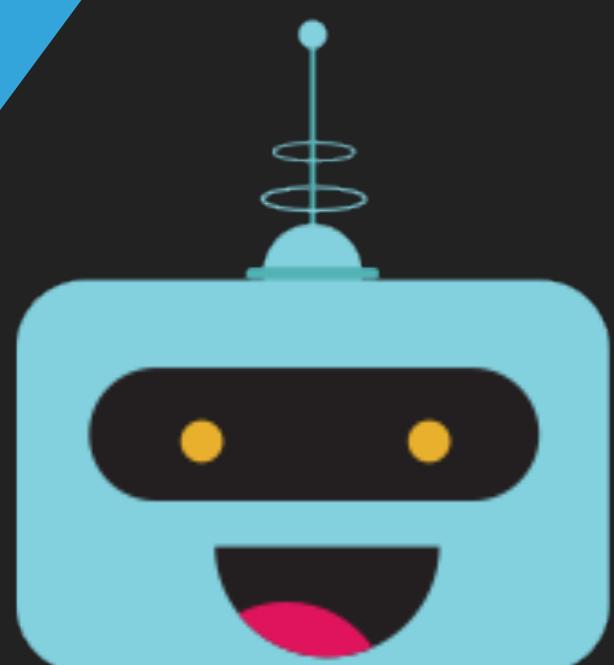
(Move up and down to reveal more choices)

Commitlint - Lint your commits

Husky - Git Hooks



WE HAVE A STANDARD WAY TO COMMUNICATE CHANGE, CAN WE TAKE ADVANTAGE OF THIS STANDARDIZED FORMAT IN OTHER WAYS? IT'S GREAT THAT THE CHANGES ARE MORE CLEAR, BUT WE DON'T HAVE AN OBJECTIVE WAY TO DETERMINE VERSION.



GIVEN A VERSION NUMBER MAJOR.MINOR.PATCH, INCREMENT THE:

1. MAJOR VERSION WHEN YOU MAKE INCOMPATIBLE API CHANGES.
2. MINOR VERSION WHEN YOU ADD FUNCTIONALITY IN A BACKWARDS COMPATIBLE MANNER, AND
3. PATCH VERSION WHEN YOU MAKE BACKWARDS COMPATIBLE BUG FIXES.

Semantic Versioning

<https://semver.org/>

GIVEN A VERSION NUMBER **MAJOR.MINOR.PATCH**, INCREMENT THE:

1. **MAJOR** VERSION WHEN YOU MAKE INCOMPATIBLE API CHANGES.
2. **MINOR** VERSION WHEN YOU ADD FUNCTIONALITY IN A BACKWARDS COMPATIBLE MANNER, AND
3. **PATCH** VERSION WHEN YOU MAKE BACKWARDS COMPATIBLE BUG FIXES.

Semantic Versioning

<https://semver.org/>

FIX → **PATCH**

FEAT → **MINOR**

BREAKING CHANGE → **MAJOR**

AUTOMATING CHANGE

AUTOMATING CHANGE

RELEASES

SEMANTIC RELEASE



SEMANTIC RELEASE

Commit message	Release type
<code>fix(pencil): stop graphite breaking when too much pressure applied</code>	Patch Fix Release
<code>feat(pencil): add 'graphiteWidth' option</code>	Minor Feature Release
<code>perf(pencil): remove graphiteWidth option</code> BREAKING CHANGE: The graphiteWidth option has been removed. The default graphite width of 10mm is always used for performance reasons.	Major Breaking Release (Note that the BREAKING CHANGE: token must be in the footer of the commit)

Configurable

Predictable

SEMANTIC-RELEASE IS MEANT TO BE EXECUTED ON THE CI ENVIRONMENT AFTER EVERY SUCCESSFUL BUILD ON THE RELEASE BRANCH. THIS WAY NO HUMAN IS DIRECTLY INVOLVED IN THE RELEASE PROCESS AND THE RELEASES ARE GUARANTEED TO BE UNROMANTIC AND UNSENTIMENTAL...

THIS REMOVES THE IMMEDIATE CONNECTION BETWEEN HUMAN EMOTIONS AND VERSION NUMBERS, STRICTLY FOLLOWING THE SPECIFICATION AND COMMUNICATING THE IMPACT OF CHANGES TO CONSUMERS.

Semantic Release Docs

<https://semantic-release.gitbook.io/semantic-release/>

SEMANTIC-RELEASE IS MEANT TO BE EXECUTED ON THE CI ENVIRONMENT AFTER EVERY SUCCESSFUL BUILD ON THE **RELEASE BRANCH**. THIS WAY NO HUMAN IS DIRECTLY INVOLVED IN THE RELEASE PROCESS AND THE RELEASES ARE GUARANTEED TO BE UNROMANTIC AND UNSENTIMENTAL...

THIS REMOVES THE IMMEDIATE CONNECTION **BETWEEN HUMAN EMOTIONS AND VERSION NUMBERS**, STRICTLY FOLLOWING THE SPECIFICATION AND **COMMUNICATING THE IMPACT OF CHANGES TO CONSUMERS**.

Semantic Release Docs

<https://semantic-release.gitbook.io/semantic-release/>

DEMO

TP 1: KEEP IT SIMPLE

**TIP 2: REVERT, DON'T
ROLLBACK**

**TIP 3: DON'T GET ATTACHED
TO MAJOR VERSIONS**

ANYTHING
GOES

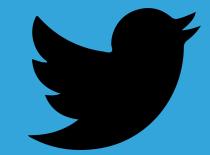


CONSISTENCY



LET THE ROBOT
WORRY ABOUT IT

BRENDEN NIEDERMEYER



@brndn_ndrmyr



<https://www.brenden.dev>



<https://careers.ef.com>

Further Reading & Resources

SLIDES & CODE: [HTTPS://GITHUB.COM/BNIEDERMEIER/TRUSTING-THE-ROBOT](https://github.com/bniedermeier/trusting-the-robot)

SEMANTIC RELEASE: [HTTPS://SEMANTIC-RELEASE.GITBOOK.IO/SEMANTIC-RELEASE/](https://semantic-release.gitbook.io/semantic-release/)

COMMITLINT: [HTTPS://COMMITLINT.JS.ORG/#/.GUIDES-LOCAL-SETUP?ID=GUIDES-LOCAL-SETUP](https://commitlint.js.org/#/.guides-local-setup?id=guides-local-setup)

COMMITZEN: [HTTPS://COMMITZEN.GITHUB.IO/CZ-CLI/](https://commitzen.github.io/cz-cli/)

NX RELEASE: [HTTPS://GITHUB.COM/GOVALTA/NX-TOOLS/TREE/MAIN/PACKAGES/NX-RELEASE](https://github.com/govalta/nx-tools/tree/main/packages/nx-release)

SEMANTIC VERSIONING: [HTTPS://SEMVER.ORG](https://semver.org)

Similar works

[HTTPS://DEV.TO/ANTONGOLUB/YOU-DON-T-NEED-SEMANTIC-RELEASE-SOMETIMES-3K6K](https://dev.to/antongolub/you-don-t-need-semantic-release-sometimes-3k6k)

[HTTPS://GITHUB.COM/CONVENTIONAL-CHANGELOG/STANDARD-VERSION](https://github.com/conventional-changelog/standard-version)

[HTTPS://GITHUB.COM/RELEASE-DRAFTER/RELEASE-DRAFTER](https://github.com/release-drafter/release-drafter)

