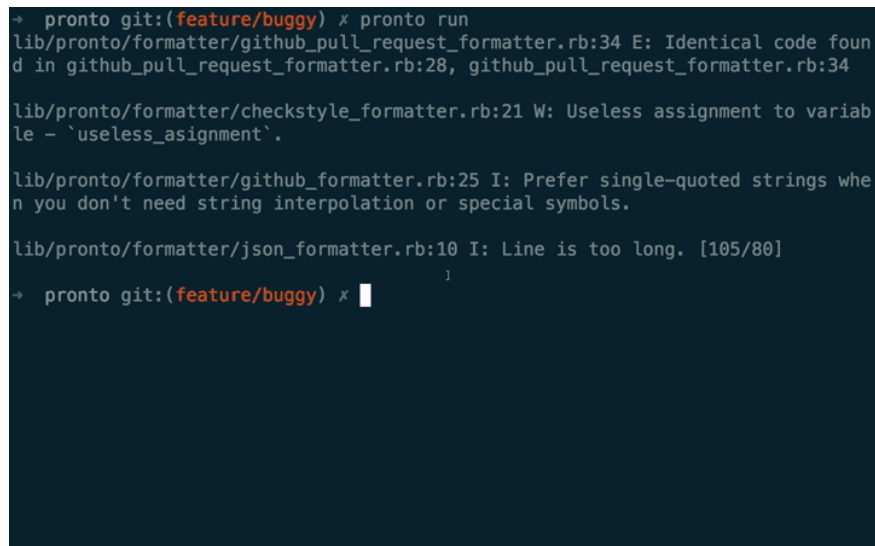


Saddler - better pronto



```
→ pronto git:(feature/buggy) x pronto run
lib/pronto/formatter/github_pull_request_formatter.rb:34 E: Identical code found in github_pull_request_formatter.rb:28, github_pull_request_formatter.rb:34

lib/pronto/formatter/checkstyle_formatter.rb:21 W: Useless assignment to variable - `useless_assignment`.

lib/pronto/formatter/github_formatter.rb:25 I: Prefer single-quoted strings when you don't need string interpolation or special symbols.

lib/pronto/formatter/json_formatter.rb:10 I: Line is too long. [105/80]
}
→ pronto git:(feature/buggy) x
```

Figure 1: Pronto demo

これは pronto のデモ画面です!!

Saddler というライブラリ群を作成中。こんなのあればいいのに、とかいろいろ募集。

Rubocop

デフォルト設定があるが、`.rubocop.yml` で全部設定変更できる

HoundCI

e.g. [Use pathname](#)

pull request にコメントしてくる。内容は, rubocop, coffeelint, jshint, scss-lint
`.rubocop.yml` は尊重、ただし、rubocop デフォルト設定を thoughtbot ルールで上書きしてる

- [HoundCI](#)
- [thoughtbot/hound](#)

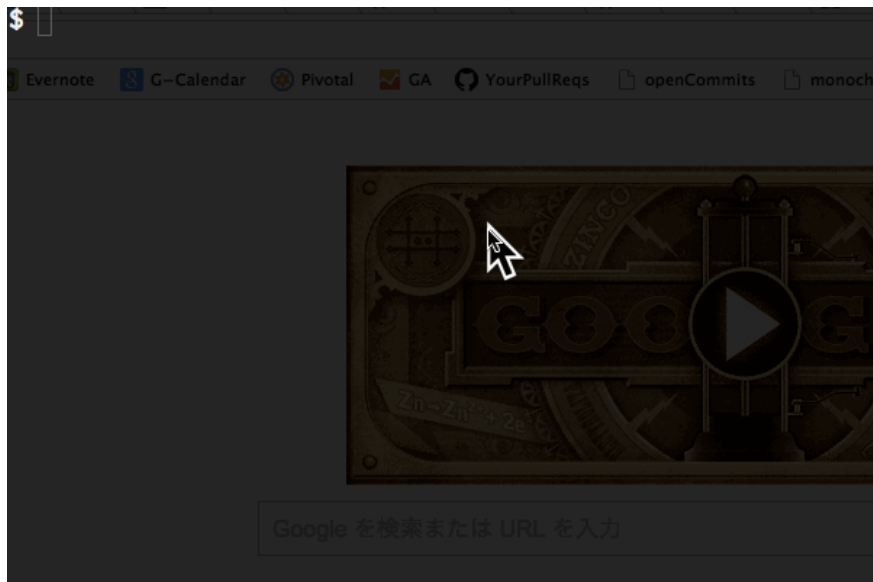


Figure 2: Rubocop demo

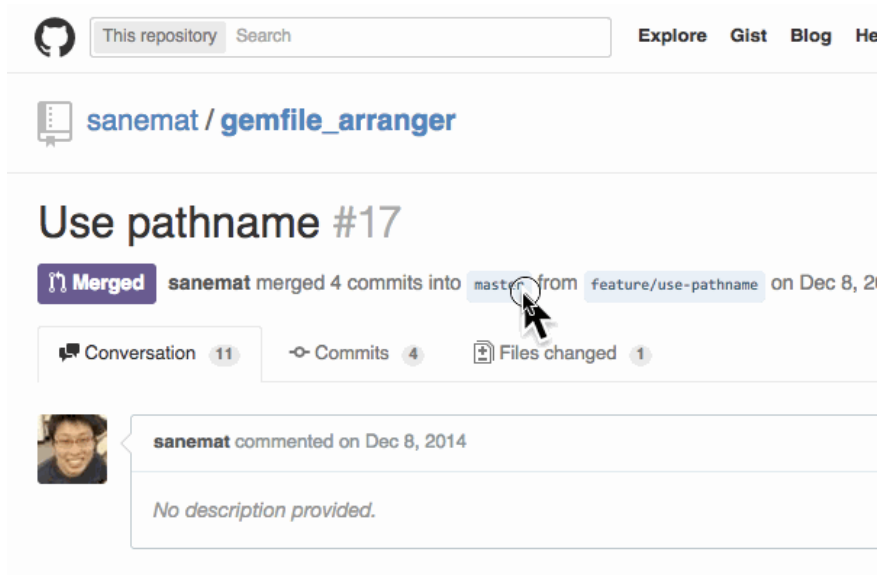
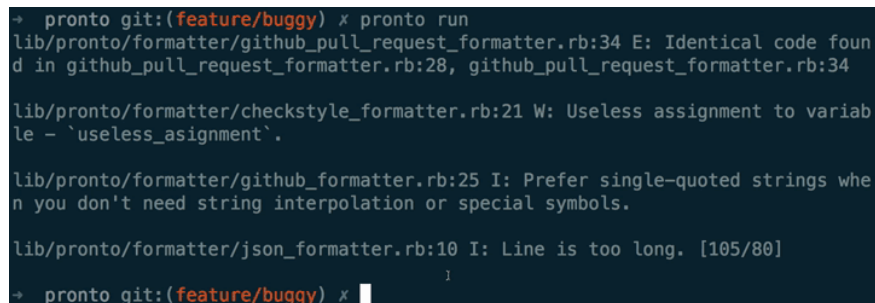


Figure 3: HoundCI demo

Pronto



```
→ pronto git:(feature/buggy) x pronto run
lib/pronto/formatter/github_pull_request_formatter.rb:34 E: Identical code found in github_pull_request_formatter.rb:28, github_pull_request_formatter.rb:34

lib/pronto/formatter/checkstyle_formatter.rb:21 W: Useless assignment to variable - `useless_assignment`.

lib/pronto/formatter/github_formatter.rb:25 I: Prefer single-quoted strings when you don't need string interpolation or special symbols.

lib/pronto/formatter/json_formatter.rb:10 I: Line is too long. [105/80]
}
→ pronto git:(feature/buggy) x
```

Figure 4: Pronto demo

```
$ pronto run -f github -c origin/master
```

```
$ GITHUB_ACCESS_TOKEN=token PULL_REQUEST_ID=id pronto run -f github_pr -c origin/master
```

対応 runner

- pronto-brakeman
- pronto-coffeescript
- pronto-flay
- pronto-foodcritic
- pronto-jshint
- pronto-haml
- pronto-poper
- pronto-rails_best_practices
- pronto-reek
- pronto-rubocop
- pronto-scss
- pronto-spell

対応 reporter

- checkstyle
- github
- github pull request
- gitlab
- json
- text

[mmozuras/pronto](#)

Saddler

```
git diff -z --name-only origin/master.. \
| xargs -0 bundle exec rubocop-select \
| xargs rubocop \
    --require rubocop/formatter/checkstyle_formatter \
    --format RuboCop::Formatter::CheckstyleFormatter \
| checkstyle_filter-git diff origin/master.. \
| saddler run \
    --require github/pull-request-comment-formatter \
    --format Github::PullRequestCommentFormatter
```

```
$ git diff -z --name-only b5ee3a61...origin/master
.gitignore~@.rubocop.yml~@.travis.yml~@Gemfile~@README.md~@bin/console~@bin/setup~@check
```

```
$ git diff -z --name-only b5ee3a61...origin/master \
| xargs -0 rubocop-select
/Users/sane/work/ruby-study/checkstyle_filter-git/checkstyle_filter-git.gemspec
/Users/sane/work/ruby-study/checkstyle_filter-git/Gemfile
/Users/sane/work/ruby-study/checkstyle_filter-git/bin/console
/Users/sane/work/ruby-study/checkstyle_filter-git/example/invalid.rb
/Users/sane/work/ruby-study/checkstyle_filter-git/exe/checkstyle_filter-git
/Users/sane/work/ruby-study/checkstyle_filter-git/lib/checkstyle_filter/git/cli.rb
/Users/sane/work/ruby-study/checkstyle_filter-git/lib/checkstyle_filter/git/diff_parser..
/Users/sane/work/ruby-study/checkstyle_filter-git/lib/checkstyle_filter/git.rb
```

```
$ git diff -z --name-only b5ee3a61...origin/master \
| xargs -0 rubocop-select \
| xargs rubocop \
```

```

--require rubocop/formatter/checkstyle_formatter \
--format RuboCop::Formatter::CheckstyleFormatter
<?xml version='1.0'?>
<checkstyle>
  <file name='/Users/sane/work/ruby-study/checkstyle_filter-git/checkstyle_filter-git.ge
  <file name='/Users/sane/work/ruby-study/checkstyle_filter-git/Gemfile' />
  <file name='/Users/sane/work/ruby-study/checkstyle_filter-git/bin/console' />
  <file name='/Users/sane/work/ruby-study/checkstyle_filter-git/example/invalid.rb'>
    <error line='3' column='100' severity='info' message='Line is too long. [188/100]' s
  </file>
  <file name='/Users/sane/work/ruby-study/checkstyle_filter-git/exe/checkstyle_filter-gi
  <file name='/Users/sane/work/ruby-study/checkstyle_filter-git/lib/checkstyle_filter/gi
    <error line='14' column='6' severity='info' message='Assignment Branch Condition siz
(snip)

$ checkstyle_filter-git diff b5ee3a61...origin/master

$ saddler run \
  --require github/pull-request-comment-formatter \
  --format Github::PullRequestCommentFormatter

```

pronto dis

え、pronto でいいじゃん、足りないところあればコントリビューションすればいいじゃん

pronto 自体の出来はよい、but 確認しづらいパーツが多すぎる

rugged とか、gitlab とか、bitbucket とか git diff 相当で invalid byte sequence で落ちる

余談 rugged は libgit2 の ruby バインディングなんだけど、普通に wrapper た例えば、cli の git diff ってめっちゃめっちゃいろいろなことやるちょっとウソ言ってるかもだから補足お願い

```

git diff: HEAD..index git diff FILE_A FILE_B: FILE_A FILE_B の
HEAD..index git diff origin/master: origin/master..index git diff
origin/master..HEAD:

```

おれはいますぐ git diff の結果がほしいんだ! という場合には、難しい、でも他の選択肢ない checkstyle_filter-git では、git ライブラリいろいろ試し

た結果、`open3.capture3 libgit2` は当然中の `mapping` だから、`rugged` もだいたい、中の `mapping` と一緒

ラッパー頑張ってるけど、イマイチ

ラッパーないものとか、別言語とか

`diff` 部分だけを `lint` これは別の `lint` 作者 (`vimlint`) からの着想だけど、ファイルをやって、プラス分だけやれたほうがいいたとえば、`javascript` で `var` で変数宣言してる行を削った場合、など増分 `diff` の `lint` だけでは出てこない

増分 `diff` の範囲になければ無理だけど！ちょっと、これは後で考える、`github pull request comment` に引きずられ過ぎかもしれない

houndci dis

`ruby` で `wrapper` ライブラリあるものに今は絞ってる `js` のライブラリを `gem` にしたものってオワコン めっちゃはやい、`github pull request` の `diff files` から取得 ただし、反対にそれによる制約も出てくる

saddler

なんでもできる (はず) `jscs` とか `eslint` とか、`tslint` とか、行けるようになるんだ たぶん

今すぐためしてみたい！

やっと昨日 `saddler` コマンド作り始めたので、もうちょっとお待ちを。 `gem` と `npm` の `saddler` って名前だけ押さえた。

```
git diff -z --name-only origin/master.. \  
| xargs -0 bundle exec rubocop-select \  
| xargs rubocop \  
  --require rubocop/formatter/checkstyle_formatter \  
  --format RuboCop::Formatter::CheckstyleFormatter \  
| checkstyle_filter-git diff origin/master.. \  
| saddler run \  
  --require github/pull-request-comment-formatter \  
  --format Github::PullRequestCommentFormatter
```

ruby と nodejs は書くけど、本当は go lang でポン置き出来るようにしたい。
出来るのかは知らない。

[Tachikoma.io](https://tachikoma.io)

sanemat {AT} tachikoma.io