

Danger

@vividmuimui


2018/09/05 社内LT



DANGER

<https://danger.systems/ruby/>


Danger とは




vividmuimui commented 9 days ago • edited ▾


Owner + 🧑


1 Error

[sample-project/README.md#L3](#) - Line 3 sentence length(103) exceeds the maximum sentence length of 100.
Over 3 characters.(preset-ja-technical-writing/sentence-length)


1 Warning

 Big PR

Generated by  Danger

 wip

✖ 4d2d1dd



👁 vividmuimui reviewed 9 days ago

View changes

sample-project/README.md

... ... @@ -22,3 +22,5 @@ Things you may want to cover:


22 22 * Deployment instructions

23 23

24 24 * ...


25 +


26 + インターフェース



vividmuimui 9 days ago

Owner

 インターフェース => インタフェース(spellcheck-tech-word)

 Reply...

3

Danger とは

レビューbot用のツール
jsとrubyで提供されている

サンプル

```
# PRタイトルにWIPついてたら、コメント
warn("PR is classed as Work in Progress") if github.pr_title.incl

# PRの変更行数が500以上だったら、コメント
warn("Big PR") if git.lines_of_code > 500

# 普通にrubyスクリプトなので、やりたい放題
fail("fdescribe left in tests") if `grep -r fdescribe specs/`.length > 0
fail("fit left in tests") if `grep -r fit specs/`.length > 1

# rubocopなどよく使うやつはpluginがある
rubocop.lint inline_comment: true
```

メリット

- 各種CIやgitのホスティングサービスをサポートしている
- rubocopなどのよく使うところはプラグインがある
 - プラグインの中見てもだいぶシンプルなので、カスタマイズや自作も簡単
- チーム独自のルールや簡単にかける
 - ruby(js)を書けばいいだけなので簡単&やりたい放題
 - たとえば xxx.ja.html を編集したけど xxx.zh-TW.html を編集してなければ警告など

使い方

注意: ruby, Github, CircleCIでの話です。雰囲気を読み取ってください。

コメント

```
# PRにコメント
message "foo bar"
# markdownとしてコメント
markdown ":alert:"
# CI を落とす
fail "This build didn't pass tests"
# :alert: マークつく
warn "Big PR"

# file changedにコメントもできる
warn("xxxxxx", file: "Dangerfile", line: 10)
```

1 Error



[sample-project/README.md#L3](#) - Line 3 sentence length(103) exceeds the maximum sentence length of 100.
Over 3 characters.(preset-ja-technical-writing/sentence-length)

1 Warning



Big PR

Git, Github

```
git.added_files  
git.modified_files  
git.deleted_files  
git.lines_of_code
```

```
# コミットメッセージ
```

```
warn("") if git.commits.any? { |c| c.message =~ /^Merge branch 'm
```

```
# diff
```

```
diff = git.diff_for_file("Gemfile.lock")  
message("") if diff && diff.patch =~ "nokogiri"
```

```
github.pr_title  
github.pr_body  
github.pr_labels  
github.branch_for_base
```

```
# review
```

```
github.review.start  
github.review.fail(message)  
github.review.warn(message)  
github.review.submit
```

コードの共有

gemとしてpluginを追加する以外にも色々な方法で追加できる

```
# githubのファイル直指定
device_grid = "https://raw.githubusercontent.com/fastlane/fastlane"
danger.import_plugin(device_grid)
# local file
danger.import_plugin("danger/plugins/watch_plugin.rb")
danger.import_plugin("danger/plugins/*.rb")
danger.import_dangerfile(path: "path/to/Dangerfile")
# gem
danger.import_dangerfile(gem: "ruby-grape-danger")
# github, gitlab
danger.import_dangerfile(gitlab: "ruby-grape/danger")
danger.import_dangerfile(
  github: "ruby-grape/danger",
  branch: "custom",
  path: "path/to/Dangerfile"
)
```

導入の仕方

- Dangerfile を用意
- danger gemをinstall
- DANGER_GITHUB_API_TOKEN を設定
- bundle exec danger --verboseとかでdangerを実行

だけ！

ローカルでも動かせる

```
danger pr --verbose
```

ただ、対象にするPRが、そのブランチで一番最近マージされたPRを対象にしているっぽい(たぶん)

GithubでオープンになっているPRを対象に実行したりはできなそう(未確認)

参考

- 公式
 - <https://danger.systems>
- 試して動かしたやつ
 - <https://github.com/vividmuimui/danger-sample>
 - <https://github.com/vividmuimui/danger-sample/pull/7>
- 参考プラグイン
 - https://github.com/ashfurrow/danger-rubocop/blob/master/lib/danger_plugin.rb
- Github検索
 - <https://github.com/search?q=warn+filename%3ADangerfile>