

GitHub Actionsで ブログの校正を自動化してみた

自己紹介

Name: ユータ

Occupation: インフラエンジニア

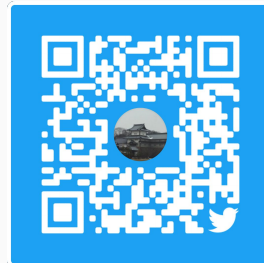
Twitter: @Y0u281 (オーでなくゼロです)

ブログ: <https://zenn.dev/yuta28> →

普段触るもの: AWS Ansible Linux Mint



Twitter



目次

- 背景
- 技術スタック紹介
 - textlint
 - reviewdog
- 実装
- 工夫ポイント
- まとめ

背景

- ZennブログをGitHub連携することができるようになった☆
 - [Zenn個人開発の限界に挑んだ話 - devsumi 2021](#) 詳しくは開発者のプレゼン 👁👁
- CIツールを導入する事で自動化が可能に 🙌
 - リポジトリを更新する度にドキュメントの校正
 - イイ感じのブログ執筆



個人開発の限界に挑んだ話

2021.02.19

textlint

- mdファイルやプレーンテキスト向けのLinterツール
- `npm install textlint `textlint-rule`` → ルールもインストール
- デフォルトルールが存在せず、自分でルールを決めなければならない
- `.textlintrc` にルールを記述

<https://textlint.github.io/>



textlint

textlint

```
# .textlintrc
{
  "rules": {
    "preset-smarthr": {
      "sentence-length": false
    },
    "prh": {
      "rulePaths": [
        "dict/smarthr-prh-basic.yml",
        "dict/smarthr-prh-tech-word.yml"
      ]
    }
  }
}
```

ルール

- textlint-rule-preset-smarthr
- textlint-rule-prh

参照

[よりよい文書を書くための校正ツール「textlint」のSmartHR用ルールプリセットを公開しました！](#)
[SmartHRオープン社内報](#)

reviewdog

- GitHubにlinterツールで検知した指摘事項をレビューコメントに投稿
- Goで書かれたOSS
- Circle CIやGitHub Actionsなど複数のCIサービスと連携可能

<https://github.com/reviewdog/reviewdog>



実装

```
jobs:
  reviewdog-github-check:
    name: reviewdog (github-check)
    runs-on: ubuntu-latest

    steps:
      #reviewdogのアクション
      - uses: reviewdog/action-setup@v1
        with:
          reviewdog_version: latest

      #textlintを動かすためのnodeアクション
      - uses: actions/setup-node@v2

      - name: Install textlint
        run: 'npm install --save-dev textlint-rule-preset-smarthr textlint-rule-prh'

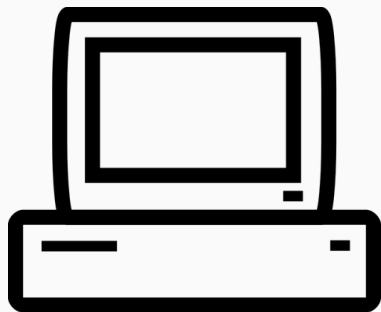
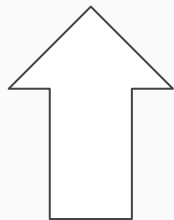
      - name: Install dependent module
        run: npm install

      - name: Execute textlint for ブログ記事
        run: |
          npx textlint -f checkstyle "articles/*.md" >> .textlint.log

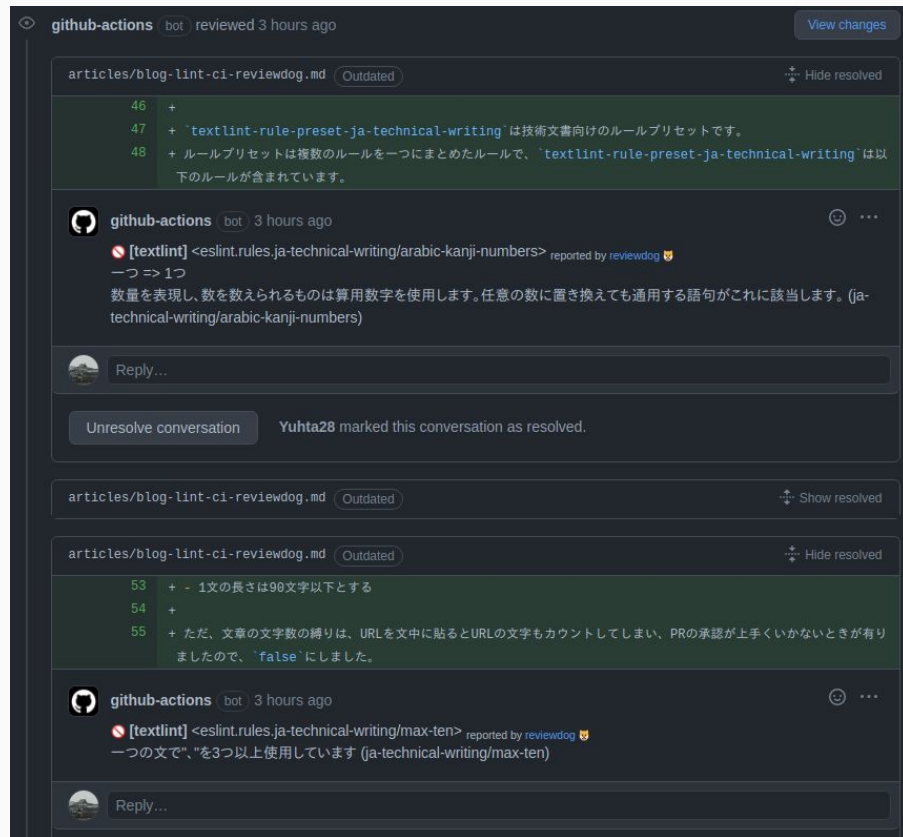
      - name: Run reviewdog
        # textlintで文章上のミスがあった場合のみ reviewdogを実行させるようにする
        if: failure()
        env:
          REVIEWDOG_GITHUB_API_TOKEN: ${ secrets.GITHUB_TOKEN }
        run: |
          cat .textlint.log | reviewdog -f=checkstyle -name="textlint" -reporter="github-pr-review"
```

- 1) reviewdog,nodeのアクションを起動
- 2) textlintのインストール
- 3) textlintを記事に対して実行
- 4) 何かしらの記載ミスがあった場合のみtextlintの出力結果を
reviewdogに渡してGitHubにレビューコメントを投稿

GitHub

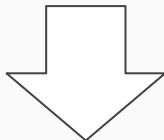


イメージ図



工夫ポイント

- キャッシュを使って高速化
 - GitHub Actionの公式にactions/cache



- stepが失敗するとキャッシュしてくれない

- stepが終了してもキャッシュしてくれるアクションを使う

<https://github.com/pat-s/always-upload-cache>

```
▼ ❌ Execute textlint for ブログ記事
1 ▶ Run npx textlint -f checkstyle "articles/*.md" >> .textlint.log
6 Error: Process completed with exit code 1.

> ✅ Run reviewdog
🕒 Post Cache node modules
```

```
#stepsが失敗(文章の乱れ)した場合でもcacheを取得するようにする
uses: pat-s/always-upload-cache@v2.1.3
env:
  cache-name: cache-node-modules
with:
  path: ~/.npm
  key: node-${{ hashFiles('**/package-lock.json') }}
  restore-keys: |
    node-
```

工夫ポイント

- キャッシュ前（初回実行） 50秒



- キャッシュ取得後（2回目以降） 29秒



まとめ



ありがとうございました

より詳しい内容は以下のブログで↓
[GitHub ActionsでZennブログの校正を自動化してみた](#)

