PSR-2 で Lint した結果を見えるようにして、コード品質の最低限を上げる

TL;DR

- Lint の結果をもっと活用しよう。
- Lint の結果をもっと見えるようにしよう。
- 重要なのはコードを介した対話 (総花的だ)

コーディング規約の課題

- このフレームワークのルールはそうかも知れないが自分はこっちの方がいい問題
  - 自転車置場の議論はヤメロ
- Lint の結果が出てくるのがCI ではタイミング遅い
  - エディタや IDE で直せるように入れよう 機械がやれ
- リポジトリに設定は入っているのだが、従っていないコードがpush される
  - ココの話! 機械がやれ

Example - PHP Code Sniffer

例えばこのコード、`if` のあとのカッコの前に space がない、`if () {} else {}`の brace がない。

```
# test.php
<?php
if($a==2)
    echo $a;
else
    echo '3';
```

これを PHP Code Snifferでチェックすると、エラーが表示される。

```
$ php phpcs.phar --standard=PSR2 test.php

FILE: C:\Users\ishida\src\test.php
----------------------------------------------------------------------

FOUND 7 ERRORS AFFECTING 4 LINES
----------------------------------------------------------------------

 1 | ERROR | [x] Expected 1 space after closing brace; 0 found
 2 | ERROR | [x] Expected 1 space after IF keyword; 0 found
 2 | ERROR | [x] Inline control structures are not allowed
 2 | ERROR | [x] Whitespace found at end of line
 4 | ERROR | [x] Expected 1 space after ELSE keyword; newline found
 4 | ERROR | [x] Inline control structures are not allowed
 5 | ERROR | [x] Expected 1 blank line at end of file; 2 found
----------------------------------------------------------------------

PHPCBF CAN FIX THE 7 MARKED SNIFF VIOLATIONS AUTOMATICALLY
----------------------------------------------------------------------


Time: 90ms; Memory: 2.5Mb
```

PHP のコーディングスタイルを PHP_CodeSniffer で修正する- Qiita

コード規約あると何がウレシイかというと、どうでもいいことでのストレスが減らせる。

コードレビューの時間をどうでもいい指摘ばっかりしたくない。どっちでもいいこととか。 4space, 2space, tab とか。

どうでもいいことは、指摘もしたくない。機械的に勝手に直されてほしい。

`php-cs-fixer` を使おう。

コーディング規約の課題 (一部済)

- ~~フレームワークのルールはそうかも知れないが自分はこっちの方がいい問題~~
  - ~~自転車置場の議論はヤメロ~~
- Lint の結果が出てくるのがCI ではタイミング遅い

    - エディタや IDE で直せるように入れよう
- リポジトリに設定は入っているのだが、従っていないコードがpush される
    - ココの話!

正論だがしかし

正論だし、メリットもわかる。ではなんで徹底されない? 単に、ツールやテクノロジーが足りてない。知ってればわかるだけの話。やろう。

仕事の生産性上げるのは、仕事だ!

入力支援

エディタなり IDE なりの手元でガンガン支援受けて直すのが一番いい。

コマンド

`php-cs-fixer` や php code sniffer 付属の `phpcbf`。

- FriendsOfPHP/PHP-CS-Fixer
- PHP CS Fixer で快適 PHP ライフ - Fivestar's blog
- コーディング規約自動調整ツールCodeSniffer2 と php-cs-fixer - Qiita
- PHP コードをコマンドで自動整形! Condig Standards Fixer と PHP_CodeSniffer - Qiita

エディタ

- stephpy/vim-php-cs-fixer (vim)
- PHP CS fixer を emacs から使うための関数 (emacs)
- benmatselby/sublime-phpcs (sublime)
- benmatselby/atom-php-checkstyle (atom)
- Using PHP Code Sniffer Tool (phpstorm)
- PHP features The PDT Extension Group eclipse p2 repository (eclipse)

良いタイトルが合ったので、持ってきた。

- PHP コードの整形はプログラマがやるべきことじゃない - Shin x blog(phpstorm)

コーディング規約の課題 (一部済)

- ~~フレームワークのルールはそうかも知れないが自分はこっちの方がいい問題~~
  - ~~自転車置場の議論はヤメロ~~
- ~~Lint の結果が出てくるのがCI ではタイミング遅い~~
  - ~~エディタや IDE で直せるように入れよう~~
- リポジトリに設定は入っているのだが、従っていないコードがpush される
  - ココの話!

結果の可視化

git のコミットフックとか、テスト/CI に混ぜてるとか、悪くない。

でもカジュアルにpush はしてほしい。 push 蹴られるのは個人的にはあまり好みではない。

Pull Request Review Comment

全員が lint 済みのものをpush してくれるか? というと疑問。

こんな感じに、pull request に review comment がつくツールやサービスがある。おもに ruby プロジェクトの場合 Hound(Web Service), Hound(OSS), Pronto というのを使う。

言語ごとに実装してるのマヌケっぽい。

- packsaddle/ruby-saddler

言語中立なのを作った。 **Saddler**

今は github だけですが、reporter 差し替えて使えるようにしているので、gitlab, bitbucket, なども対応するつもり。 svn とか mercurial でも使いたい人がいれば reporter つくればよい。

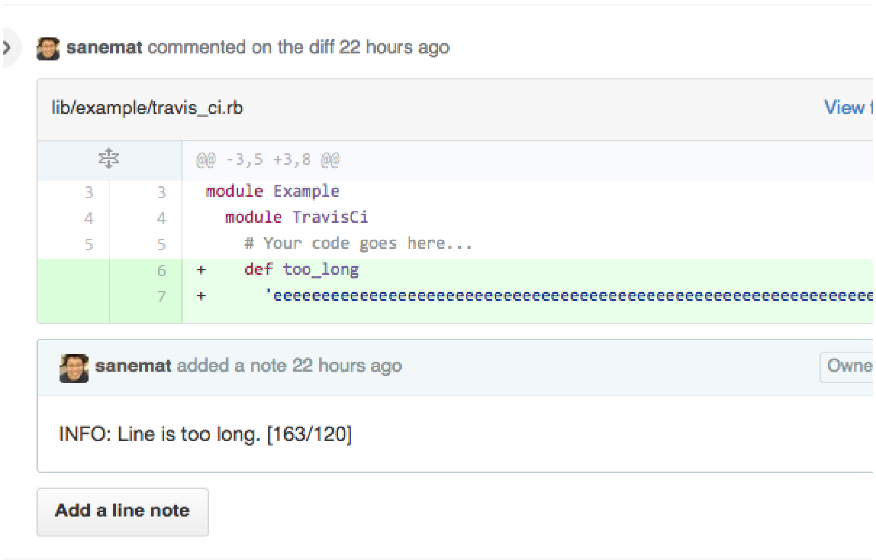

Figure 1: review comment

example - Saddler　プルリクエストに対してテストが走る、その後処理の中で、実行する。

```
# TravisCI や CircleCI の after test の中で (travisCI 用語で after_script, CircleCI 用語で post
git diff --name-only origin/master \
  | grep ".*.php$" \
  | xargs phpcs --report=checkstyle \
  | checkstyle_filter-git diff origin/master \
  | saddler report \
      --require saddler/reporter/github \
      --reporter Saddler::Reporter::Github::PullRequestReviewComment
```

途中出力

```
git diff --name-only origin/master \
  | grep ".*.php$" \
  | xargs phpcs --report=checkstyle
```

```
<?xml version="1.0" encoding="UTF-8"?>
<checkstyle version="1.0.0">
 <file name="/path/to/code/myfile.php">
  <error line="2" column="1" severity="error" message="Missing file doc comment" source=
```

```
  <error line="20" column="43" severity="error" message="PHP keywords must be lowercase;
  <error line="47" column="1" severity="error" message="Line not indented correctly; exp
  <error line="47" column="20" severity="warning" message="Equals sign not aligned with
  <error line="51" column="4" severity="error" message="Missing function doc comment" so
 </file>
</checkstyle>
```

と、たぶんなるはずなんだけど、手元にいい感じに setup した php プロジェクト無いので、誰か手伝ってほしい。

メリット

- github 上に可視化できること。
- コード増分だけに適用出来ること。
- コメントを無視もできるところ。身も蓋もないけど。

build phases

- TravisCI (after_script)
  - Travis CI: Configuring your build
- CircleCI (post test)
  - Configuring CircleCI - CircleCI
- Jenkins (post build task)
  - Post build task - Jenkins - Jenkins Wiki
  - Jenkins でジョブが失敗した時にだけ実行したい処理があった場合の対応パターン - Thanks Driven Life

コーディング規約の課題 (一部済)

- ~~フレームワークのルールはそうかも知れないが自分はこっちの方がいい問題~~
  - ~~自転車置場の議論はヤメロ~~
- ~~Lint の結果が出てくるのがCI ではタイミング遅い~~
  - ~~エディタや IDE で直せるように入れよう~~
- ~~リポジトリに設定は入っているのだが、従っていないコードがpush される~~
  - ~~ココの話!~~

設定したがしかし

設定したら終わりか?

ノー。そこから始まり。

無視できなくする

あまりうまくない。 直さないと緊急デプロイできない とかはつらい。

重要なのは

重要なのはコードを介した対話だ!

参照

- PHP のコーディングスタイルを PHP_CodeSniffer で修正する- Qiita
- PSR-2 — Coding Style Guide
- 新標準 PSR に学ぶきれいな PHP
- More PHP Formatting Options and Bundled Code Styles for PSR-1/PSR-2 and Symfony2 | WebStorm & PhpStorm Blog
- PSR-0 はなぜ 0（≒ 最重要）なのか- 泥のように
- PHP CS Fixer で快適 PHP ライフ - Fivestar's blog
- CakePHP のソースコードのレビュー結果を共有してみる - Qiita
- Saddler

sanemat {AT} tachikoma.io