

PHPプロジェクトに 静的解析を新規導入する

Introduce static analyzers to PHP project

！ お前誰よ



- うさみけんた (@tadsan) / Zonu.EXE
- GitHub/Packagistでは id: zonuexe
- ピクシブ株式会社 pixiv事業本部
- Emacs Lisper, PHPer
 - Emacs PHP Modeのメンテナ引き継ぎました
 - 好きなリスプはEmacs Lispです
- Qiitaに記事を書いたり変なコメントしてるよ

pixiv



お絵描きがもっと楽しくなる場所を創る

pixiv 2018年度新卒
エントリー受付中



Friends of Emacs-PHP development

Join us!

<?php

-  **Repositories** 25
-  **People** 5
-  **Teams** 1
-  **Projects** 0
-  **Settings**

Find a repository...

Type: All ▾Language: All ▾

Customize pinned repositories

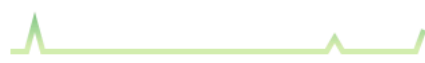
New

php-runtime.el

PHP-Emacs bridge, call PHP function from Emacs

- php
- emacs
- melpa

Emacs Lisp ★ 2 1 GPL-3.0 Updated 2 days ago



Top languages

Emacs Lisp PHP

phpactor.el

Interface to Phpactor (an intelligent code-completion and refactoring tool for PHP)

Emacs Lisp ★ 8 4 1 issue needs help Updated 3 days ago



Most used topics

Manage

emacs melpa php

major-mode

php-mode

A PHP mode for GNU Emacs



People 5 >



Invite someone



Search or jump to...



[Pull requests](#)

[Issues](#)

[Marketplace](#)

[Explore](#)



emacs-jp

📍 Japan

🔗 <http://emacs-jp.github.com/>

Repositories 18

People 37

Teams 8

Projects 0

Settings

Find a repository...

Type: All ▾

Language: All ▾

[Customize pinned repositories](#)

New

reading-init.el

init.el読書会のページ

[html](#)

[dotfiles](#)

[emacs](#)

[emacs-lisp](#)

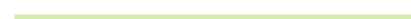
● Ruby ★ 6 🍴 1 Updated 6 days ago



helm-c-yasnippet

Helm source for yasnippet

● Emacs Lisp ★ 14 🍴 5 Updated on 21 Mar



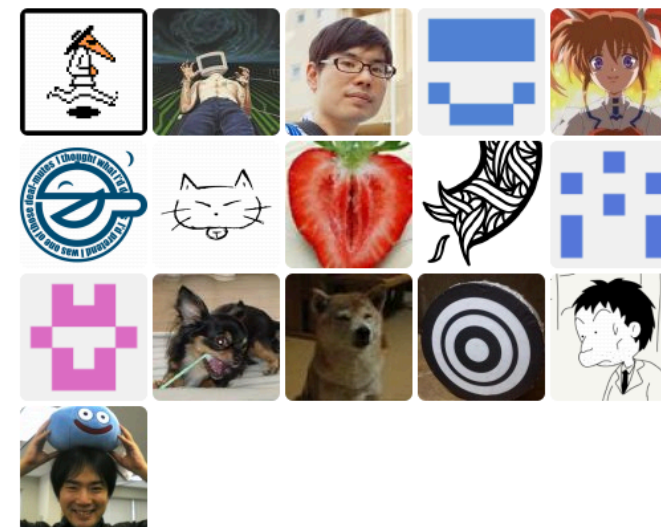
replace-colorthemes

Top languages

● Emacs Lisp ● Ruby ● TeX

People

37 >



本日のお題



11月
30

大改修！PHPレガシーコードビフォーアフター

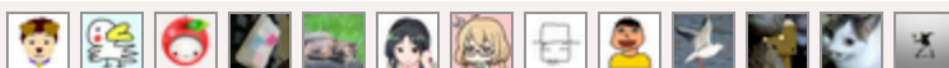


5人の匠が問題設計大改修！

主催：PHPerKaigi / @tomzoh

\$PHP_legacy_code ->
before()->after();

🟢 フォロー参加者



★ フォローブックマーク



募集内容

一般
無料

先着順
159/180人

運営お手伝い
無料

先着順
6/8人

グループ

メンバーです

PHPerKaigi



イベント数 1回
メンバー数 216人

開催前

2019/11/30(土)

13:00 ~ 18:40

Googleカレンダー icsファイル

このイベントに申し込む

開催日時が重複しているイベントに申し込んでいる場合、このイベントには申し込むことができません

募集期間

2019/11/12(火) 08:28 ~
2019/11/30(土) 18:40

✉ イベントへのお問い合わせ

人気

人気



イベント概要

PHPの匠によるビフォーアフター、見てみたいですね？

このイベント主催の長谷川は、PHPerKaigi や iOSDC Japan といったカンファレンスを主催し、その運営をサポートするために fortee というシステムを開発・運営しています。

<https://forte.jp>

forte は2017年12月から開発を開始し、それ以降長谷川が主催した4つのカンファレンスと、深く関わったいくつかのカンファレンスのために追加開発を続けてきました。

その結果、長年にわたって運用しつつ継ぎ足し継ぎ足し開発されたシステムには技術的負債がたまり、たいへんツライ状態になってきています。

「マイクロサービスとはほど遠いモノリシックなコード」

「テストがごく一部しか存在しない」

「突然現れる巨大なコントローラ」

「テンプレートに書かれた if 文」

「バージョンアップに追従していないフレームワーク」

きっとみなさんもこんなシステムを見たことがあると思いますし、現在進行形でお困りの方もいるかと思います。
今回は5名のPHPの匠に fortee のソースコード一式をお渡しして見て頂き、匠ならではの改善を発表して頂きます。



2019/09/05 ~ 09/07
iOSDC Japan 2019

[公式サイト](#) [公式Twitter](#) [#iosdc](#)

[プロポーザル一覧](#) [タイムテーブル](#)



2019/03/29 ~ 03/31
PHPerKaigi 2019

[公式サイト](#) [公式Twitter](#) [#phperkaigi](#)

[プロポーザル一覧](#) [タイムテーブル](#)

Laravel JP Conference

2019 / 2 / 16 (Sat)

グランパークカンファレンス

[#laraveljpcon](#)

2019/02/16
Laravel JP Conference 2019

[公式サイト](#) [公式Twitter](#) [#laraveljpcon](#)

[プロポーザル一覧](#) [タイムテーブル](#)



2018/11/10
Hi-Con 2018

[公式サイト](#) [公式Twitter](#) [#hicon2018](#)

[プロポーザル一覧](#)



2018/08/30 ~ 09/02
iOSDC Japan 2018

[公式サイト](#) [公式Twitter](#) [#iosdc](#)

[プロポーザル一覧](#) [タイムテーブル](#)



2018/03/09 ~ 03/10
PHPerKaigi 2018

[公式サイト](#) [公式Twitter](#) [#phperkaigi](#)

[プロポーザル一覧](#)

さて

PHPカンファレンス 2016

pixiv inside [archive]



2016-11-11

PHP

Phan静的解析がもたらす大PHP型検査時代

こんにちは、pixivでPHPをやってるうさみです。健全なコードベースは黙っても降ってこないの
で、チーム全体で開発効率を高めるような改善をするのがお仕事です。

テキストエディタはmicro推しです ヲ(//><)ノ ☆

さる11月3日に大田区産業プラザ PiOで開催されたPHPカンファレンス 2016にて大怪獣に蹂躪さ
れながらPhanについて30分のセッション発表をいたしましたので、その内容を紹介します！

まだユニットテストと同様に「あって当然」のものとまでは普及できてない

*php*ゆるふわCI入門

A casually guide for PHP Continuous Integration

pixiv

2019-03-30 PHPerKaigi 2019 Day 1
Nerima Coconeri Hall #phperkaigi

既存のプロジェクトに
導入する話は
あまりできてなかった

PHP

型検査・夢と理想と現実

PHP typing: the gap between ideal and reality.

補訂公開版

pixiv

[2019-06-29]

PHPCon Fukuoka 2019
FFBHall #phpconfuk

なぜ静的解析を
するのか

関数がどのような引数
を期待しどんな値を
返すのか明示したい

プログラムが
間違ってることを
実行前に検証したい

それ以上に

オブジェクトのメソッド
やプロパティが適切に
補完されると気持ち良い

補完がきちんと機能
するようにPHPに
型をつけていこう

ただし

静的解析は、現に
「動いている」事実を
覆すものではない

エラーとして見えないが
「たまたま動いている」
現実を炙り出すことはある

変更によって動かなくなっ
てしまいうことを
「動かさずに」発見できる
かもしれない

静的解析基盤を整えた
瞬間に致命的な問題を
検出できることはまれ

ということで静的
解析をしましょう

静的解析 ツールの種類

| 現行の高度な静的解析ツール

- PhpStorm (IDE)
- Phan
- PHPStan
- Psalm

| PhpStorm

- よいところ
 - 導入が最も簡単
 - 十分な精度のエラー検出
- よくないところ
 - エラーの列挙ができないのでCIのフローに組み込めない

| Phan

- よいところ
 - 導入は比較的簡単
 - とても高い精度のエラー検出
- よくないところ
 - 大規模プロジェクトでは解析に時間がかかる

| PHPStan

- よいところ
 - 実行時情報を使用し高速
 - 高い精度のエラー検出
- よくないところ
 - Composer以外の依存関係があると導入にコツが要る

| Psalm

- よいところ
 - 独自の引数アサーション
 - 高い精度のエラー検出
- よくないところ
 - 大規模プロジェクトでは解析に時間がかかる

静的解析の選択軸

| | Phan | PHPStan |
|------|---|--------------------------|
| 解析方法 | 実行しない(静的) | 定義のみ実行が必要
(静的と動的のmix) |
| 依存関係 | php-ast (高速/C拡張)
Tolerant PHP Parser(低速) | PHP-Parser |
| 解析単位 | プロジェクトおよび
依存ライブラリ全体 | 対象ファイルを
個別指定 |
| | | |

| つまりどれがいいのか (CI)

- 現時点ではそれぞれのツールで検出項目に一長一短があるため、使い分けるとよい
- PHPStanは高速なのでGitでブランチにpushするごとに検査するようなCIと相性がよい

| PHPStanの制約

- クラス/関数/定数の定義と副作用のある処理は明確に分かれていないといけない
- つまり、PSR-1
- Phanなら定義と処理が混在したファイルも解析できる

| こういうコードはPHPStanで解析できない

```
<?php
include __DIR__ . '/config.php';

$connection = new PDO(DSN);
drop_table($connection);
exit;

function drop_table($connection)
{
    $connection->exec('DROP TABLE `users`');
}
```

静的解析を
導入しよう

CakeFest



THE **CAKEPHP**
CONFERENCE
TOKYO, JAPAN
NOV 7TH - 10TH 2019



Be the first to clip this slide

より安全で便利な
SAFER, MORE HELPFUL

CAKEPHP

1 of 31



Safer, More Helpful CakePHP

555 views



markstory

+ Follow

Recommended



経営課題、専門家を活用して
解決しませんか？
経営支援サービスi-
common (アイコモン)

Sponsored Content



Educational Technology for
Student Success

Online Course - LinkedIn Learning



PowerPoint for Teachers:
Creating Interactive Lessons

Online Course - LinkedIn Learning



Teaching Online:
Synchronous Classes

Online Course - LinkedIn Learning



CakePHP - The Road Ahead
markstory



Future of HTTP in CakePHP
markstory



CakePHP mistakes made
2015
markstory

May 31, 2019

CakePHP 4.0.0-alpha2 Released

The CakePHP core team is proud to announce the second alpha release of CakePHP 4.0.0. Since the previous alpha we've been hard at work finishing off the improvements to typehinting and decreasing the number of skipped warnings from both [psalm](#) and [phpstan](#). This may result in some breaking changes in your application or plugins, but in the long term we feel these changes will make CakePHP better.

The application skeleton and bake have been refreshed with a new coat of paint. We've opted to not continue using foundation for the basic CSS, and adopted [milligram](#) instead. The smaller footprint of milligram makes it easier for you to remove or extend if you'd prefer.

CakePHP 4.0.0では
PHPStanとPsalmで
型定義が大幅に改善
(現在RC1)

じゃあPhanを
入れるか

基本はcomposer requireか
Pharファイルを
ダウンロードするだけ

forteeはレガシー
を自称しているが
Docker Compose

Docker Compose
使ったことない僕の方が
レガシーなんですけど...

Docker Phan [検索]



だめです

[Pull requests](#)[Issues](#)[Marketplace](#)[Explore](#)[phan / phan](#)[Used by](#)

388

[Unwatch](#)

117

[★ Star](#)

4.6k

[Fork](#)

293

[Code](#)[Issues](#) 483[Pull requests](#) 3[Actions](#)[Projects](#) 3[Wiki](#)[Security](#)[Insights](#)[Releases](#)[Tags](#)

Latest release

2.4.4

9236f30

Verified

Release 2.4.4 supporting PHP 7.0-7.4 syntax

 TysonAndre released this 5 days ago · [8 commits](#) to master since this release

New features(CLI, Configs):

- When stderr is redirected a file or to another program, show an append-only progress bar by default. ([#3514](#))
Phan would previously disable the progress bar entirely by default.

The new `--long-progress-bar` CLI flag can be used to choose this progress bar.

(The `--no-progress-bar` CLI flag or the environment variable `PHAN_DISABLE_PROGRESS_BAR=1` can be used to disable this)

- Treat `$var = $x['possibly undefined offset']` as creating a definitely defined variable, not a possibly undefined variable. ([#3534](#))

The config setting `convert_possibly_undefined_offset_to_nullable` controls whether the field type gets cast to the nullable equivalent after removing undefined.

New features(Analysis):



Search or jump to...



Pull requests

Issues

Marketplace

Explore



phan / phan

Used by ▾

388

Unwatch ▾

117

★ Star

4.6k

Fork

293

<> Code

Issues 483

Pull requests 3

Actions

Projects 3

Wiki

Security

Insights

Releases

Tags

Pre-release

0.9.4

d2f3fbd

Development release 0.9.4 supporting PHP 7.1 syntax



TysonAndre released this on 16 Aug 2017 · [3538 commits](#) to master since this release

The 0.9.x versions will be tracking syntax from PHP versions 7.1.x and are runnable on PHP 7.1+. Please use version 0.8.x if you're using a version of PHP < 7.1.

See the file [NEWS](#) for what has changed in this version.

▼ Assets 3



phan.phar

3.39 MB



Source code (zip)



Source code (tar.gz)

古すぎる！

Phanはほぼ毎週
アップデートされて
さまざまな改善がある

最新のPhanだけが
良いPhanだ

| Phanの要求環境

- 基本的に普通のPHPイメージ
 - ext-ast (PECL ast)は必須
 - Xdebugは無効化したい
 - パフォーマンスに影響する
- ext-pcntl, ext-iconv, ext-mbstringも推奨

| Dockerfileにphp-astを足す

```
RUN pecl install ast \  
&& docker-php-ext-enable ast
```

```
Staged changes (7)  
new file   .phan/.gitignore  
@@ -0,0 +1 @@  
+vendor/  
new file   .phan/composer.json  
@@ -0,0 +1,11 @@  
+{  
+  "require": {  
+    "phan/phan": "^2.4"  
+  },  
+  "config": {  
+    "optimize-autoloader": true,  
+    "classmap-authoritative": true,  
+    "discard-changes": true,  
+    "preferred-install": "dist"  
+  }  
+}  
modified   Makefile  
@@ -43,6 +43,12 @@ sync-files: vendor/autoload.php  
+    bin/sync-files.sh  
+    .PHONY: sync-files  
+.phan/vendor/bin/phan: .phan/composer.json  
+    docker-compose run --rm php-cli composer install --working-dir=.phan  
+    +  
+phan: .phan/vendor/bin/phan  
+    docker-compose run --rm php-cli php .phan/vendor/bin/phan  
+    +
```

astをデフォルト有効化したくなければ

RUN pecl install ast

```
✓Staged changes (2)
✓modified Makefile
@@ -44,10 +44,10 @@ sync-files: vendor/autoload.php
.PHONY: sync-files

.phan/vendor/bin/phan: .phan/composer.json .phan/composer.lock
- docker-compose run --rm php-cli composer install --working-dir=.phan
+ docker-compose run --rm php-cli php -d 'extension=ast.so' /usr/bin/composer install --workin

phan: .phan/vendor/bin/phan
- docker-compose run --rm php-cli php .phan/vendor/bin/phan
+ docker-compose run --rm php-cli php -d 'extension=ast.so' .phan/vendor/bin/phan

cypress:
./node_modules/.bin/cypress open
✓modified docker/web/Dockerfile
@@ -20,8 +20,7 @@ RUN docker-php-ext-configure gd --with-freetype-dir=/usr/include/ --with-jpeg-di
#\
#    && docker-php-ext-enable xdebug

-RUN pecl install ast \
-    && docker-php-ext-enable ast
+RUN pecl install ast
```

別のコンテナイメージ
に分割したい場合

| docker/phan/Dockerfile

```
FROM php:7.3-cli
RUN apt-get update -y \
    && apt-get install -y --no-install-recommends \
        libcu-dev \
        unzip
RUN docker-php-ext-install -j$(nproc) intl pcntl

RUN pecl install ast && docker-php-ext-enable ast

COPY --from=composer /usr/bin/composer \
    /usr/bin/composer

RUN composer global require phan/phan
ENTRYPOINT php /root/.composer/vendor/bin/phan
WORKDIR /var/www/html
```

| docker-compose.yml

```
version: "3.7"
services:
  phan:
    build: ./docker/phan
    volumes:
      - ./:/var/www/html:cached
    entrypoint: php /root/.composer/vendor/bin/phan
    working_dir: /var/www/html
```


独立したコンテナに分ける
メリットも大きくない気がする
ので既存のPHP用コンテナに
混ぜ込めばいいと思います

| composer require --devを使うべき？

- 好みでどうぞ
- アプリケーションと依存関係が混ざるのでメンテ難しいと思います
- 僕はサブディレクトリにcomposer.json作ります
- composer-bin-pluginを使ってもいいかも

以後./phan という
ラッパースクリプト
を使うことにします

| ./phan

```
#!/usr/bin
```

```
docker-compose run --rm php-cli \  
    .phan/vendor/bin/phan "$@"
```

- Windowsならphan.batってファイルを作って %* あたりにパラメータ展開すれば動くと思うけど、詳しいことはWindowsに詳しいひとに任せた！

| phan --initで初期化できる

- .phan/config.phpが生成される
- 自動追加されなかった
directory_listを追加していく
- composer requireで直接
依存するパッケージだけが
自動追加される

| 基本的な使いかた

- ./phan
- -o phan.log
結果をphan.logに書き出す
- -j 4
並列実行数。CPUのコア数を指定するのが良い (諸説あり)

| baseline

- `--save-baseline=.phan/baseline.php` を付けて実行するとファイルごとに検出された Issue を保存できる
- `--load-baseline` で実行すると新たに増えたものだけ報告される

| suppress_issue_types

- 報告しないIssueをリストアップ
 - フレームワーク側に問題があるなら握り潰すしかない…
 - ある程度の妥協も時には必要
 - CakePHP 4.0になれば
おそらくほぼ消せるでしょう

| tadsanのよくやる手法

- masterブランチの内容をphan-master.logに書き出す
- リファクタリングして変更の都度解析結果をphan.logに書き出す
- `diff -u phan-master.log phan.log`

実際に検出
されたミス

未定義変数の検出

src/Classes/PopAccountCore.php:266 PhanUndeclaredVariable Variable \$part is undeclared

```
public static function getEncoding  
{  
    if (isset($message->getOverview)  
        $decoded = imap_mime_header_decode($message->getOverview());  
        if (isset($part->charset)  
            return $decoded->charset;  
        }  
}
```

\$part変数が未定義

未定義変数の検出

```
src/Controller/ProposalsController.php:197 PhanUndeclaredVariable Variable $slideEmbedHtml is undeclared  
src/Controller/ProposalsController.php:197 PhanUndeclaredVariable Variable $videoEmbedHtml is undeclared
```

```
$this->set(compact('proposal', 'videoEmbedHtml', 'slideEmbedHtml'));  
}
```

\$videoEmbedHtmlと
\$slideEmbedHtmlが未定義

I deprecatedコードの検出

```
src/Controller/SponsorFormController.php:92 PhanDeprecatedProperty Reference to deprecated property \Cake\Http\ServerRequest->data defined at vendor/cakephp/cakephp/src/Http/ServerRequest.php:62 (Deprecated because: 3.4.0 This public property will be removed in 4.0.0. Use getData() instead.)
```

```
if (isset($this->request->data['plans']  
    $selectedPlans = $this->request->
```

CakePHP 4.0.0で
プロパティにアクセス
できなくなる

未定義クラスの検出

```
src/Controller/ReceivedEmailsController.php:168 PhanUndeclaredClassMethod Call to method __construct from undeclared class \App\Controller\Exception  
(Did you mean class \Exception or class \Aura\Intl\Exception or class \Cake\Core\Exception\Exception or class \Cake\Database\Exception)
```

```
/**  
 * deprecated  
 */  
public function reply($uuid = null)  
{  
    throw new Exception('このメソッドはもう使ってません');
```

use Exceptionがないので
現在の名前空間を指してる

！ おまけ: 謎のException

```
use Aura\Intl\Exception;
```

- 3箇所くらいで登場
- どれもAuraを直接使っていない
- たぶんPhpStormの自動挿入

コンストラクタで値を返す

src/View/Helper/MasterHelper.php:16 PhanTypeMagicVoidWithReturn Found a return statement with a value in the implementation of the magic method \App\View\Helper\MasterHelper::__construct, expected void return type

```
public function __construct(View $View)
{
    parent::__construct($View, $config);

    $conferenceSlug = $this->request->get('conferenceSlug');
    if (! $conferenceSlug){
        return null;
    }
}
```

値を返せない
(単にreturn;でよい)

そろそろまとめ

実際稼働しているアプリ
なので致命的な問題は
なかなか見つからない

リファクタリングや
フレームワークの
バージョンアップでは
大きな武器になる

地味なものも
目立つが地道に
漬していいこう

おまけ

今回forteeのコードで
初めてDocker Compose
で構築されたPHPアプリ
ケーションに触れた



なるほどわからん

手持ちのレガシーPHP
をDocker Compose
に載せてみるか

 [pixiv](#) / [intern2015w](#)

 Unwatch ▾ 93

 Unstar 8

 Fork 15

 Code

 Issues 0

 Pull requests 0

 Actions

 Projects 0

 Security

 Insights

Tree: 53e7cebc64 ▾ [intern2015w](#) / README.md

Find file

Copy path

 [zonuexe](#) Fix repository path

53e7ceb on 12 Nov 2015

[1 contributor](#)

90 lines (54 sloc) | 4.38 KB

 Edit on HackMD

Raw

Blame

History



pixiv-winter-internship 2015課題

ピクシブの冬インターンシップでは二日間の短期メニューでpixivのバグフィックスおよび簡単な機能改修を体験してもらいます。

そのための課題として、PHPで書かれたコードへの改善を行ってください。

課題

あなたは**次世代インタラクティブコミュニケーションサービス**の開発チームにアサインされました。

チームメンバーは基本機能の実装が完了してリリース日を待つばかりだと浮かれ気分ですが、あなたはリリース直前にも関わらず多数の問題を発見してしまいました。

無事にサービスをリリースできるように、気がついた脆弱性について対処を行ってください。

開発環境について

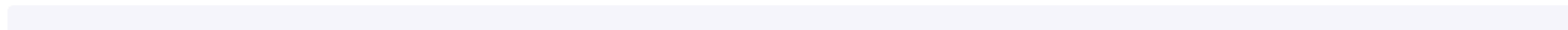
Debian GNU/Linux + PHP 7の環境を利用してください。

[Vagrant](#)で[rlerdorf/php7dev](#)のBoxを利用することがおすすめです。

- Vagrantの動作には[VirtualBox](#)が必要です
- 一部のWindows機ではBIOSの設定により仮想環境が動作しない場合があるので確認してください

環境構築については[サポートチャット](#)で支援しますので、わからないことは気軽に質問してください。

セットアップ



Vagrant →
Docker Compose

phpdotenv
.env → conf/.env

docker-composeを
噛ませてsetupスクリプト
を実行できるように
場所を変更

php -S →
Apache HTTP

初めてでも
できちゃった