

# 「ふつうのPHP」が pixivになるまで



pixiv.inc

USAMI Kenta @tadsan

pixiv

2018-07-14 PHPカンファレンス関西 #php

# お前誰よ

- うさみけんた (@**tadsan**) / Zonu.EXE
- ピクシブ株式会社 pixiv運営本部 技術基盤チーム
- 2012年末から現職、WEB APIを実装したりしたよ
- 今回発表するようなところを見つめてきたよ
- Emacs PHP Mode 現行メンテナ/<https://github.com/emacs-php>
- Qiita: <https://qiita.com/tadsan> 適当な記事を書いてきたよ



# pixivの紹介

[ホーム](#)[作品投稿 | 管理](#)[ブックマーク](#)[フィード](#)

10+

[ディスカバリー](#)[描き方](#) [創作アイディア](#)**tadsam**  
premium

フォロー 3514

フォロワー 91

マイピク 13

**プレミアムサービス**

アクセス解析

再投稿機能

予約投稿

sensei

[もっと見る](#)

閲覧履歴

コメント履歴

グッズ購入履歴

グッズ売上

グループ 15

イベント

pixivFANBOX

## お知らせ

公式イベント 「WEBマンガ総選挙2018」決選投票受付開始！ - 7月13日

公式イベント 「不如帰 大乱 カードイラストコンテスト」結果発表 - 7月6日

新機能 ライブ配信者を応援する「エール」機能とライブ配信者向け「リワード」の提供を開始しました - 7月12日

## ディスカバリー

## 人気のライブ



原稿ペン入れ

視聴4人



作画します

視聴34人



お昼のお絵かきタイム

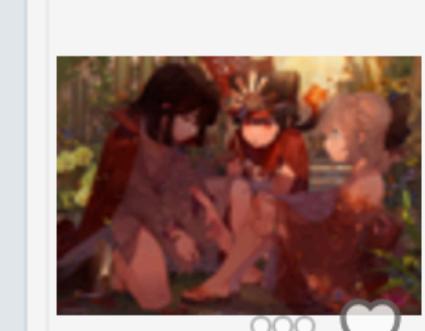
視聴11人

[ライブをはじめる »](#) [もっと見る](#)

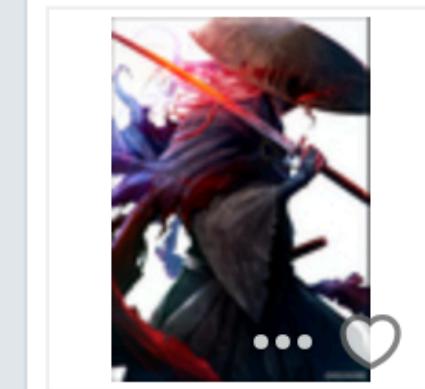
## ショーケース

疲れたあなたを癒すひと♡憧れのお姉さんを  
描いたイラスト特集良い香りがしそう♡柑橘類が描かれたイラスト  
で爽やかな気分をお届け！

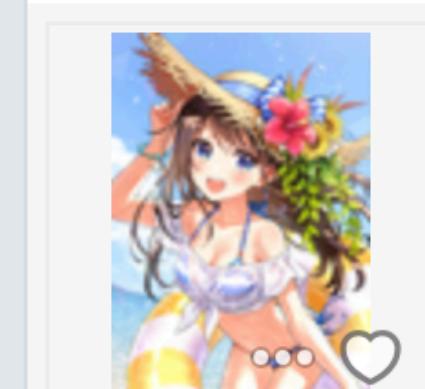
## デイリーランキング

[総合](#) | [イラスト](#) | [うごイラ](#) | [漫画](#) | [小説](#)あねうえ  
by 河CY

1

岡田以蔵  
by ケースワベ  
[K-SUWABE]

2

Tropical  
Summe...  
by 森倉円

3

# イラストコミュニケーションサービス pixiv

- 2007年9月10日に開始されたイラストSNS・投稿サイト
  - イラスト: 約7000万作品 (IDベース、画像枚数は1億4000万枚以上)
  - ユーザー: 約3000万アカウント (IDベース)
  - イラストブックマーク: 約35億回
  - アクセス数: 月間76億 (HTTPdログベース、一般的な意味のPVとは異なる)
  - サーバー台数: 53台 (pixivのデプロイ対象になっているApache HTTPDのみ)

# 用語

- pixiv本体 [www.pixiv.net](http://www.pixiv.net)
  - デスクトップ版(いわゆるPC向け)
  - モバイル版(いわゆるスマートフォン向け、旧touch.pixiv.net)

# pixiv.git

- pixivとデータベースを共有するコードは一個のリポジトリに含まれる
- それ以前はサービス横断のgit submoduleで共有されてたが効率悪かった

```
├ accounts.pixiv.net
├ admin
├ app-api.pixiv.net
├ batch
├ bin
├ bungei-api.pixiv.net
├ comic-api.pixiv.net
├ embed.pixiv.net
├ fanbox.pixiv.net
├ m.pixiv.net
├ me.pixiv.net
└ oauth.pixiv.net

├ pixiv-lib
├ public-api.pixiv.net
├ rpc.pixiv.private
├ sensei.pixiv.net
├ source.pixiv.net
├ spotlight.pics
├ ssl.pixiv.net
├ tests
├ touch.pixiv.net
├ util
├ www.pixiv.net
└ www.pixivision.net
```

# Real World PHP in pixiv 改

pixiv 2016 SPRING BOOT CAMP

2016年3月15日

<https://niconare.nicovideo.jp/watch/kn1259>

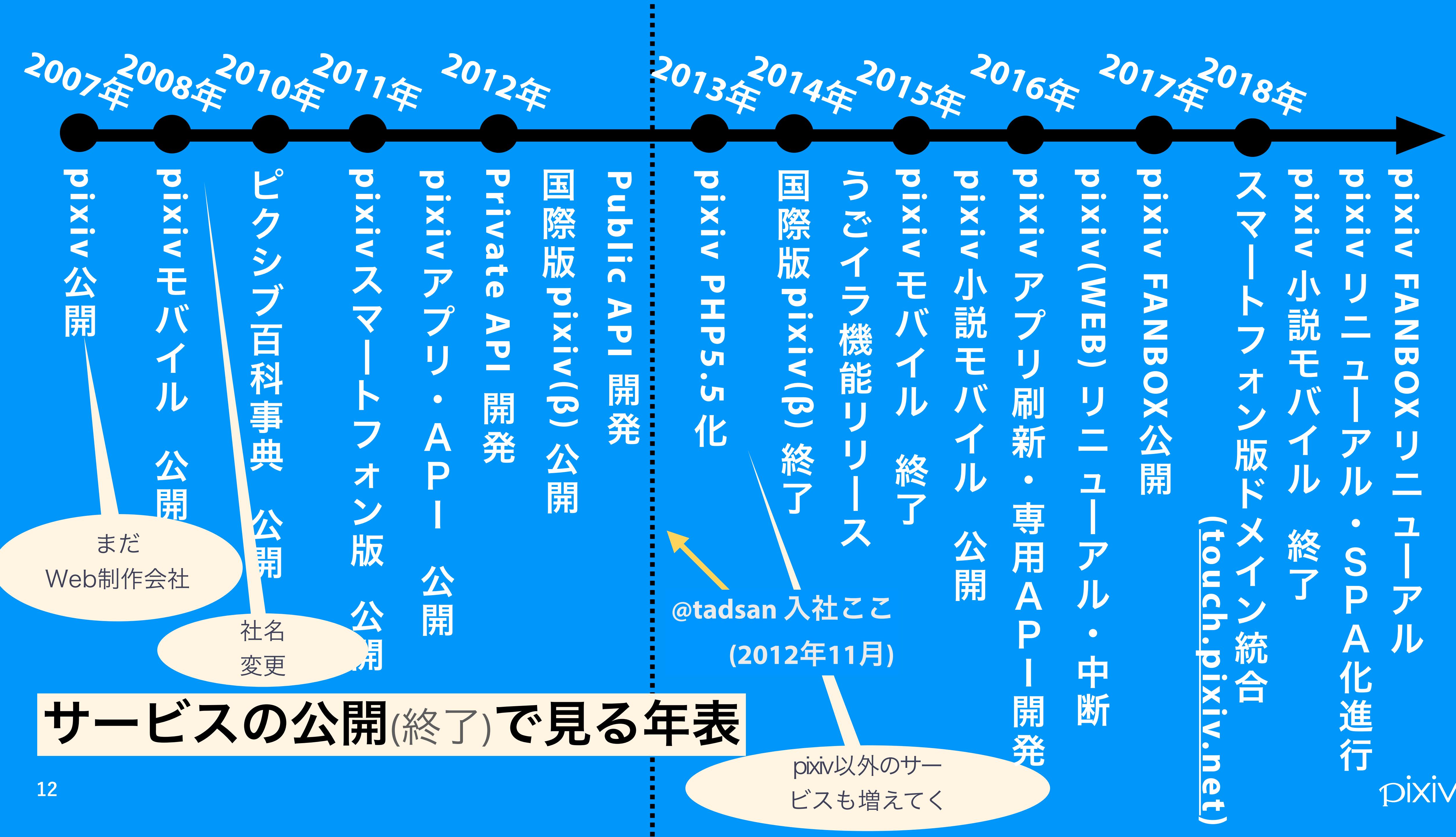
# よくある質問

- フレームワークは使ってないの？
- 使ってません。マイクロフレームワークのようなのはいくつもある
- べんりユーティリティの集合で実際フレームワークになっている
- フレームワークを指向して開発されたものもいくつかある
- フレームワーク／言語の移行やフルスクラッチしないの...？
- 直近での予定はありません。過去に痛い目にも遭ってるし...

# pixivの歴史

# お断り

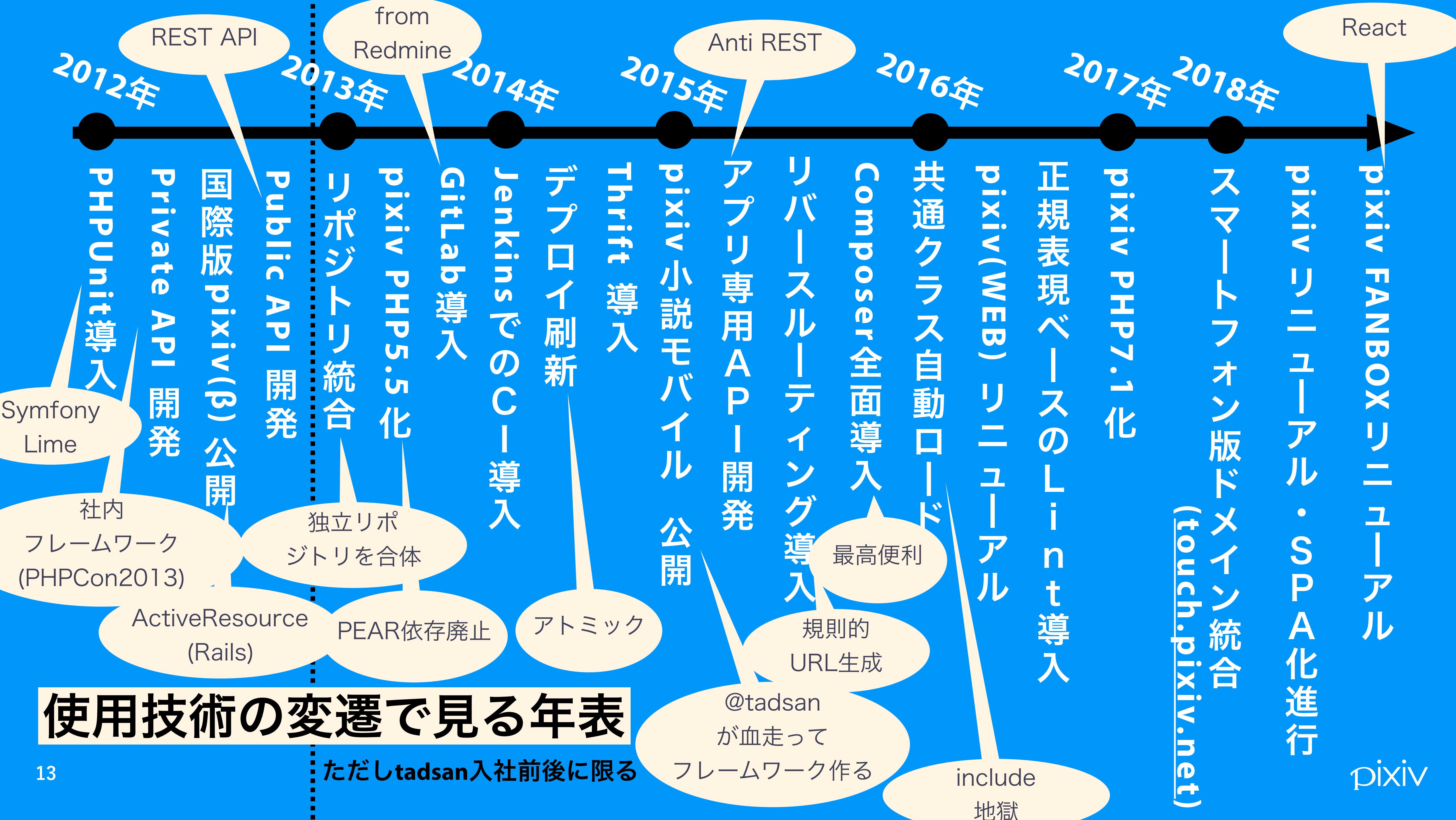
数々の変更は@tadsanが直接行ったものではなく、歴代の開発者が改善を積み重ねてきたものです  
(むしろ恩恵を受けてきた立場)



# 使用技術の変遷で見る年表

13

ただしtadsan入社前後に限る



# お断り

今回の発表で言及されないものは概ね時間の制約で取捨選択されたか忘れてるだけなので、懇親会やTwitterで@tadsanにきいて

# ふつうのPHP #とは

# pixivのPHP

- PHPはApache+mod\_php (一時fpmを使ってた箇所あり)
  - フロントのnginxから、リバースプロキシしてApacheにリクエスト
- むかしながらのDocumentRootにある .php ファイル = URL
  - 例: <https://www.pixiv.net/member.php?id=105589>
- LinuxとかMySQLとか、よくあるLAMPのスタックに載ってる

# 2012年のコードのイメージ

```
<?php // www.pixiv.net/htdocs/hoge.php
require_once __DIR__ . '/../inc/bootstrap.php';
include_once INC_PATH . '/Hoge/Fuga.php';
include_once INC_PATH . '/Hoge/Piyo.class.php';
try {
    display()
} catch (Exception $e) {
    error::exception_error($e);
}

function display() {
    // ...
}
```

include\_onceいっぱい

エラーハンドリング  
(していないページもあった)

ファイルローカルな  
グローバル関数

この下にもいっぱい

# 2015年のコードのイメージ

```
<?php // www.pixiv.net/htdocs/hoge.php  
require_once __DIR__ . '/../inc/bootstrap.php';  
  
AppRunner::execute(new Www_HogeController);
```

include\_once一個だけ

このクラスは  
自動ロードされる

# AppRunner (名前の由来は知らない)

```
<?php

final class PCAppRunner {
    public static function execute(Controller $controller) {
        try {
            Controller_Util::turnOnWhoops();
            Controller_Util::redirectToHttps();

            $controller->main();
        } catch (\Throwable $exception) {
            PCAppRunner::setHttpStatus(500);
            Controller_Util::displayWhoops($exception);
        }
    }
}
```

whoops

パソコン版

# AppRunnerとエラーハンドリング

- サービスごとの例外ハンドリングはAppRunnerが受け持つ
- whoops! はエラー処理用のフレームワークなので、これだけを使って例外／エラー処理を完結させることも可能ではあるが、割と複雑になるので、開発環境でエラー画面を表示するだけの目的に専念している
- エラーログはPHPの標準機能は利用せず`file_put_contents()`でログファイルに書き込んでる
- whoops! とエラーログの話はWEB+DB PRESS Vol.96 に書きました

# 2018年のコードのイメージ

```
final class Controller_WwwRoutes {
    public static function getRoutes() {
        $route_map = [
            '/' => [
                'action' => function () {
                    Wwww_IndexController::main();
                },
            ],
            '/hoge.php' => [
                'action' => function () {
                    Wwww_HogeController::main();
                },
            ],
        ];
    }
}
```

htdocs/hoge.phpは消滅

URLとファイルは切り離され  
マップの一要素に

全URLが1ファイルに

技術基盤チームの山際はWebサイトのURL設計と刷新の方針について話しました。

The slide features a blue header bar with the title '大規模ウェブサイトのURL刷新の方針と実装' and subtitle 'by Kohki YAMAGIWA'. A small 'SD' logo is in the top right corner. The main content area has a large white title '大規模Webサイトの URL刷新の方針と実装' and a date '2017/10/08 PHP Conference 2017'. At the bottom, there are navigation icons: a left arrow, a right arrow, and an upward arrow.

pixivは2007年から開始されたWebサービスです。イラストを投稿するサービスという点では今まで一貫していますが、今までの間には多くの修正が行われました。当初開発された当時のURL設計と現在のページ構成には非常に。

**PHPカンファレンス2017で発表**

# 命名規則

- クラス名はPSR-0的な \_ を使った擬似名前空間
  - 組織内で一貫してればPSRに拘る必要がない
  - 「PSRの誤解」 <https://qiita.com/tadsan/items/942a381e952e12a8fa5a>
- 基本はstaticで、クラスはインスタンス化せずに利用
  - クラスを擬似的な関数・定数置き場にしている
  - どうしてこんなことを続けてるのかは去年のLTで話した

公開版

# staticおじさんになろう

War on Dynamics

pixiv

2017-07-15 PHPカンファレンス関西2017 LT

去年の当日募集LTで話した

pixiv

# 開発しやすさの ための取り組み

# PHPで開発しにくいところ

- クエリパラメータのハンドリング
- PDOの機能の貧弱さ
- テンプレートエンジンとURLの問題
- Railsにあるようなカッコイイ機能がない
  - かっこいいエラー表示
  - 対話環境 (rails console)

# クエリパラメータの問題

```
$id = $_GET['user_id'];
if (is_numeric($id)) User_Common::getById($id);
else error("不正な入力です");
```

- こういうコードを書いてはいけない
  - クエリパラメータは数字ではない値が入ってくる可能性
  - 入力が空、入力が任意の文字列、入力が不正な数字列、入力が配列
  - filter\_input() をかけると安全にはなるが、それでも面倒

# → ParamHelper

```
$id = ParamHelper::getPositiveInt('id');  
User_Common::getById($id);
```

- クエリパラメータ(またはform)から値を取り出すヘルパー
- 未入力や不正な入力があると例外を投げる
- 前述のAppRunnerでキャッチしてエラー画面を描画する

# → ParamHelper

```
$mode = ParamHelper::getEnum('mode', ['hoge', 'fuga'], ['post' => 'only']);
```

- ?mode=hoge または ?mode=fuga のみを期待するようなパターン
- 不正な形式や ?mode=piyo のような期待しない入力で例外を投げる

# テンプレートとURL生成の問題

```
<a href="{$smarty.const.SYSTEM_URL_WWW|escape}member_illust.php?id={$user.id|escape}">{$user.name|escape}</a>
```

- こういうコードを書くと事故が入り込みやすい
  - 純粋にtypoの危険性
  - idなどのパラメータに不正な値を入れるリスク

# → ReverseRoute

```
<a href="{reverse_route page='fullWwwMemberProfile' id=$user.id}">  
{$user.name|escape}</a>
```

- あるページに名前をつけて、reverse\_route関数に page 引数で渡す
- 生成結果は変更前と同じ
- ルーティングの逆関数にあたるので、一部のフレームワークは ReverseRoutingやURLヘルパーなどの名前でサポートしている

# → ReverseRoute

```
/**
 * @route\example https://www.pixiv.net/member.php?id=12345 {id: 12345}
 * @route\example https://www.pixiv.net/member.php?id=12345&utm_source=xxxxxx
{id: 12345, utm_source: "xxxxxx"}
 */
public static function fullWwwMemberProfile(array $params)
{
    Util Assert::num($params['id']);

    return ReverseRoute::buildUrl(SYSTEM_URL_WWW, '/member.php', ['id'],
$params);
}
```

# pixiv inside [archive]



pixiv insideは移転しました！» <https://inside.pixiv.blog/>

2016-10-25

## DocCommentでPHPのユニットテストの書きやすさを劇的に改善する手法

44  
B!ブックマーク

0  
シェア

ツイート

こんにちは、pixivでPHPをやってる@tadsanです。好きなテスティングフレームワークは [PHPUnit](#) です！ 好きな某CALOIDは [テトさん](#) です！

みなさまはユニットテストを書いていらっしゃいますか？ 今回はDocCommentとPHPUnitのタプロバイダーをうまく利用してテストの記述を省力化する手法を紹介いたします バ(ノ)

### pixiv insideとは

<https://inside.pixiv.blog/> に移転しました。こちらは2016年末までのアーカイブです。

+ 読者になる 221

### 月別アーカイブ

- ▼ 2016 (52)
  - 2016 / 12 (28)
  - 2016 / 11 (6)
  - 2016 / 10 (1)

blogに書きました  
<https://devpixiv.hatenablog.com/entry/2016/10/25/093000>

# PDOの問題

- SQLは自動生成じゃなくて手で書きたい...  
でもPDOで複雑なクエリを書こうとすると文字列結合が避けられない...
- PDOで書きにくいクエリの代表例: IN
- もう?を並べてインデックスをインクリメントしながらbindValueはやだお
- PDOのクエリのplaceholderに?を書くか :hogeで書くか問題
- 型チェックできるように書くのもちょっとめんどう

# PxvSql

```
/*
 * @param int[] $novel_ids
 * @param int    $limit
 * @param int    $offset
 * @param string $order  ASC|DESC
 */
public static function getByNovelIds(array $novel_ids, $limit, $offset, $order = 'ASC')
{
    $con  = self::getConnection(DBConf::DB_NOVEL, DBConf::REP_READ);
    $stmt = PxvSql::build($con, NovelRankingDao::getById_sql, [
        ':novel_ids' => $novel_ids,
        ':limit'      => $limit,
        ':offset'     => $offset,
        ':order'      => $order,
    ]);
    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}
const getById_sql = '
SELECT `novel_id`, `mode`, `date`, `rank`
FROM `pix_novel_ranking`
WHERE `novel_id` IN ( :novel_ids@int[] )
ORDER BY `date` :order@ascdesc
LIMIT  :limit@int
OFFSET :offset@int
';
```

# PxvSql

- 文字列結合を一切やらなくてもクエリが書けるようになった
- 実行時に型検査をするのでnullなどの不穏な値は入り込まない
  - エラーチェックのボイラープレートがグッと減る
  - SQLi(脆弱性)のリスクなし (ただしmysqlndの実装の不備はないと信じる)
  - :hoge@int で整数の埋め込み、:fuga\_ids@int[] で複数の整数を展開する
- %if や %for の記法で条件分岐や複数SETなども可能
- 問題意識はQiitaに書いた <https://qiita.com/tadsan/items/e615a779baa6eabdab47>

# PDOのラッパー

- これは、少なくとも2011年以前からあった仕組み
- 書き込み用(master)と読み込み用(slave)の系統を明示的に分けて利用可能
  - slaveにINSERTやUPDATEなどのクエリを発行しようと例外
  - 全クエリにSQLのクエリ單一行にして、トレースをコメントにして埋め込む
    - スロークエリのログにどのコントローラに起因するか集計しやすい
    - スロークエリを一覧できるビューアーもある

# 新機能をリリース時にどうする？

- A: 新機能が実装されたブランチをリリースするときマージする
- B: リリース前にあらかじめマージして、実行されないようにしておく
- よくある問題
  - Aは機能の規模が大きいとビッグバンマージを引き起こし、予期せぬ問題を引き起こすことがある
  - 理想としてはBで、一般ユーザー向けに実行されないとしても鮮度のよいうちにマージしておきたい(リリース時の変更を最小限に)

# A/Bテスト／機能の有効化／無効化

```
if ($_SERVER['REMOTE_ADDR'] === OFFICE_IP) {  
    hogehoge();  
}
```

- 新しい機能をサービスに実装して、社内でだけ有効化したいとする。
- 無造作にこんなコードを書かれると後から意味不明になる
- そもそも社外と社内での挙動が別物になるので嫌な予感しかしない

# DevToggle (A/Bテストフレームワーク)

```
if (ABTest_DevToggle::isEnabledDevToggle('hogefunc')) {  
    hogehoge();  
}
```

- 専用のコンソール画面でボタンを押すことで有効化／無効化される
- 設定を一行足すだけで一般ユーザーに対してリリースできる
  - リリース後に障害などが発生しないことが確認できたらifを消す
- 元はA/Bテストのための機構だった(一般ユーザーに対して確率で適用)

# まとめ

# 何もない野原に秩序をつくる

- べんり機能を導入すると、部分的に生産性が体感で数倍になったりする
- 一からサービスを作るとしたら同じことを繰り返せるかは、悩む
  - いまだったら Laravelを入れるかもしれない
  - ただ、それは2018年のチート未来人だからできる発言
- サービスの価値はコードではなく、その結果のユーザーエンタメニメント
- フレームワークがなかったとしても、  
pixivが2007年からユーザーに価値を提供してきた結果は変わらない

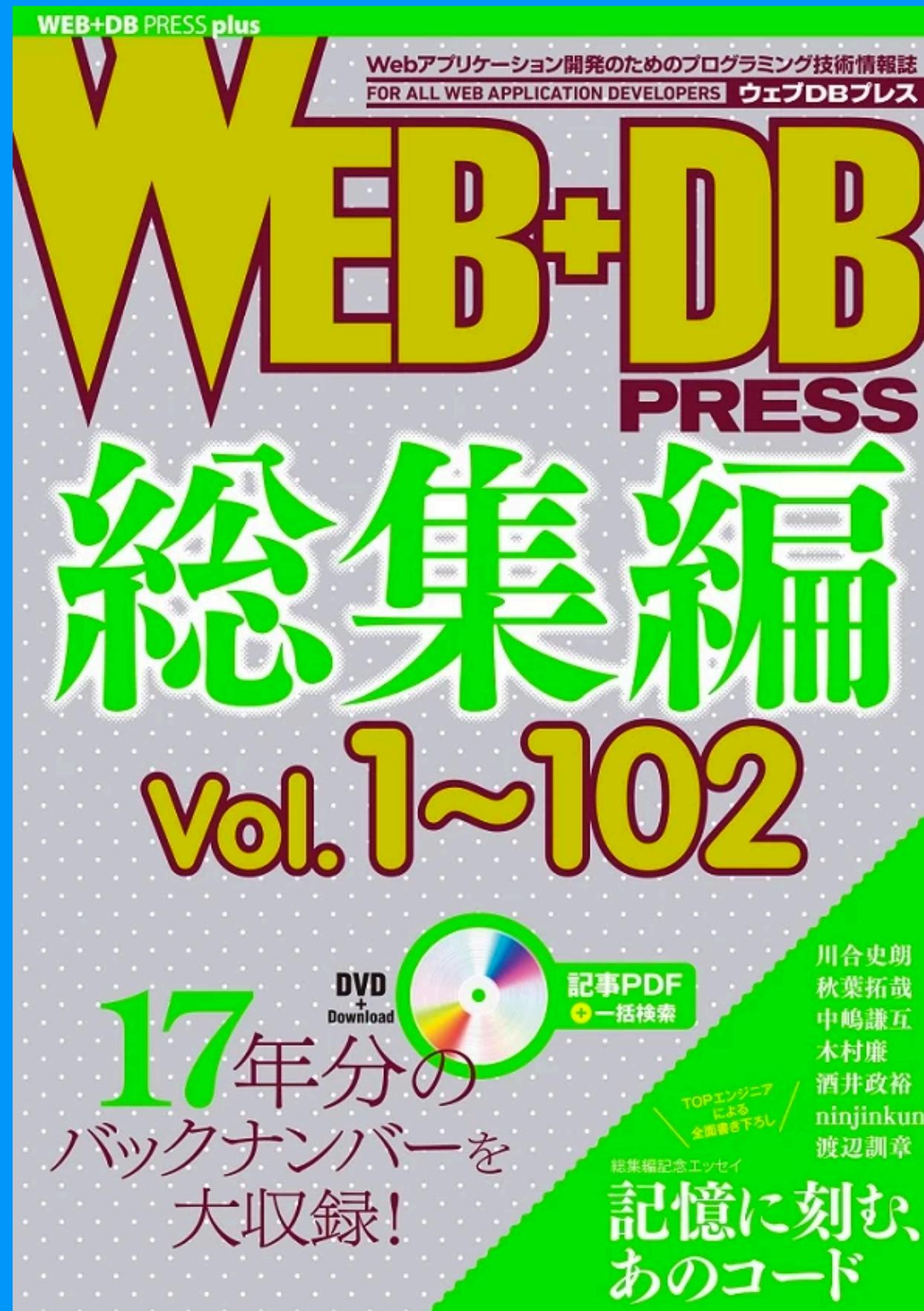
# 続きはWebで(あとWEB+DB PRESS)

- ニコナレ <https://niconare.nicovideo.jp/users/5962>
- Real World PHP in pixiv改
- pixiv inside
- DocCommentでPHPのユニットテストの書きやすさを劇的に改善する手法
- WEB+DB PRESS連載『PHP大規模開発入門』を振り返る
- pixivの基盤ノウハウ大公開！PHPカンファレンス2017登壇レポート

# 続きはWebで(あとWEB+DB PRESS)

- Qiita
  - 憂鬱なSQLのためのアレ、またはPDOと仲良くして枕を高くしてねむる
  - PSRの誤解
  - インスピイヤされて掲示板を作りたくなった シリーズ
  - includeって書きたくない僕たちのためのオートローディングとComposer
- GitHub
  - whoopsやエラー処理の例 <https://github.com/zonuexe/wdb-php-96-sample>

# 続きはWebで(あとWEB+DB PRESS)



- これ一冊を購入すれば歴代のPHP連載が全部読めるのでおすすめ
- まとめ <https://inside.pixiv.blog/tadsan/3991>
- 今回の内容に近いものだと
  - vol.81, vol.87, vol.91, vol.94, vol. 96
- 買って読んでね！Webにサンプルコードもあるよ