ここで差がつくエラー処理

By **Error Handling** your application will be completed

公開用完全版

pixiv

PHP Conference Japan 2017, 8th Oct
Kamata Tokyo, PiO #phpcon2017

2017年10月8日に開催されたPHPカンファレンスで発表した「ここで差がつくエラー処理」の発表資料を補訂した**完全版**です

pixiv

お前誰よ

- うさみけんた（@tadsan）/ Zonu.EXE
  - GitHub/Packagistでは id: zonuexe
- ピクシブ株式会社技術基盤チーム（pixiv.net）
- Emacs Lisper, PHPer
  - 入社前は自宅警備をしながらRuby書いてた
  - Emacs PHP Modeのメンテナ引き継ぎました
- 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

プレスリリース

We are hiring!

お絵描きがもっと楽しくなる場所を創る

pixiv

pixiv 2018年度新卒 エントリー受付中

プレスリリース

はじめに

pixiv

みなさん困ってませんか

pixiv

このページは動作していません

www-tadsan-dev.example.net では現在このリクエストを処理できません。

HTTP ERROR 500

再読み込み

pixiv

こんな画面を見せて
しまってませんか

pixiv

(!) Warning: PDO::__construct(): php_network_getaddresses: getaddrinfo failed: Name or service not known in /path/to/proj/D

Call Stack

#	Time	Memory	Function	Location
1	0.0001	253464	{main}( )	.../index.php:0
2	0.0130	1650808	AppRunner::execute( )	.../index.php:4
3	0.0132	1660512	WwwRouterAndDispatcher::main( )	.../AppRunner.p
4	0.0132	1660952	WwwRouterAndDispatcher::routeAndDispatch( )	.../WwwRouterA

pixiv

本番サービスで
見せちゃってませんか

pixiv

```
Warning: PDO::__construct(): php_network_getaddresses: getaddrinfo failed: Name or service not known in

Call Stack:
    0.0001    253440   1. {main}() /path/to/proj/htdocs/index.php:0
    0.0108   1651472   2. AppRunner::execute() /path/to/proj/htdocs/index.php:4
    0.0109   1661176   3. WwwRouterAndDispatcher::main() /path/to/proj/inc/AppRunner.php:16
    ...
    0.1097   3688416  99. PDO->__construct() /path/to/proj/DB/ExPDOBase.php:29

Fatal error: Uncaught exception 'PDOException' with message 'SQLSTATE[HY000] [2002] php_network_get

PDOException: SQLSTATE[HY000] [2002] php_network_getaddresses: getaddrinfo failed: Name or service r
```

pixiv

HTML 出力途中で落ちたりしませんか

pixiv

書籍一覧

id	書名	内容
1		

(!) Fatal error: Uncaught PDOException: invalid data source name in /Users/megurine/rep

(!) PDOException: invalid data source name in /Users/megurine/repo/php/phpcon2017-e

Call Stack

#	Time	Memory	Function
1	0.0001	358032	{main}( )
2	0.0016	469224	zonuexe\PhpCon2017\{closure}( )
3	0.0017	469456	include('/Users/megurine/repo/php/phpcon2017-error-processing/src/lis
4	0.0017	469864	__construct ( )

pixiv

エラーになっても
Webサイトの体裁は
なんとか保ちたい
……ですよね？

pixiv

pixiv

小説　R-18　ヘルプ　設定

検索

ホーム　作品投稿 | 管理　ブックマーク　フィード　4　ディスカバリー　描き方

エラーが発生しました

戻る

pixiv

画面はpixivの実在のバグではなく
故意に例外を発生させたものです

アジェンダ

pixiv

1. 例外・エラーの種類
2. 本番環境でのエラー処理
3. 開発環境でのエラー処理

pixiv

具体的にはこんな話をします

- エラーと例外の基本概念
- それらを制御するやりかた
- 開発環境でうんざりしない方法

pixiv

参考資料

- だいたいPHPマニュアルに載ってる
  - http://php.net/language.operators.errorcontrol
  - http://php.net/set_error_handler
  - http://php.net/set_exception_handler
  - http://php.net/register_shutdown_function
  - http://php.net/ErrorException
  - http://php.net/display-errors
- whoops! https://github.com/filp/whoops

やらない話

- エラー監視
- 例外設計
- そもそもエラーって何だ

pixiv

PHPのコードの試しかた

- `php -r 'var_dump(1 + []);'`
- REPL（PsySH または `php -a`）
- https://3v4l.org/
  - 多くのバージョンで串刺し実行

pixiv

エラー・例外とは

- プログラムが正常ではないときに、それを通知する仕組み
- PHPではエラーと例外がある
  - PHP5では処理系は基本的にエラー
  - PHP7で例外を投げることが増えた

pixiv

1. 例外・エラーの種類
2. 本番環境でのエラー処理
3. 開発環境でのエラー処理

pixiv

エラー

pixiv

エラー

- 従来のPHPで異常状態を通知するための仕組み

- 関数呼び出し、未定義変数の参照
  結構いろんなところで発生する

- PHP7で種類が整理された

pixiv

主要なエラー

- Notice よくない処理の通知
- Warning, Error
  - 実行時の警告・エラー
- Deprecated, Strict
  - 非互換／廃止予定の機能
- Fatal Error 致命的エラー

pixiv

Notice

- 配列の未定義位置から値を取り出そうとしたときなどに発生する
- 信号無視みたいな軽犯罪
- ただしバグの温床になるので、無視せずきちんと対処を強く推奨

pixiv

Noticeの例（一部）

```
PHP Notice:  Undefined variable: i in Command line code on line 1
PHP Stack trace:
PHP   1. {main}() Command line code:0

Notice: Undefined variable: i in Command line code on line 1

Call Stack:
    0.0001     349296   1. {main}() Command line code:0
```

```
echo $i;                  // 存在しない変数
echo $_GET['id']; // 存在しないインデクス
echo $o->abc;         // 存在しないプロパティ
```

Warning, Error

- 何かが「間違ってる」ときに発生
- デフォルトの挙動ではWarningで
  は停止せずに進む
- 一貫性がないが、どちらも異常状
  態なので無視せず直した方がよい

pixiv

Warning, Errorの例(一部)

```
// 存在しないファイルを開く
$f = fopen("/hoge.txt", "r");


foo($i); // 引数が足りない、PHP5のみ
// PHP7ではArgumentErrorに変更
function foo($i, $j) { echo $i, $j; }
```

Deprecated, Strict

- 廃止予定や互換性の怪しい機能
- 例: split() 関数（explode()とは別物）
  - 5.3でdeprecated、7.0で削除
- Strictは7.0で廃止(再分類)された
  https://wiki.php.net/rfc/reclassify_e_strict

pixiv

Deprecated, Strictの例(一部)

```
class Hoge {
    /** 古いスタイルのコンストラクタ */
    function Hoge() {
    }
}
```

Deprecatedの例(一部)

```
//** PHP5.3でDeprecatedエラー、7で削除
$a = split(',', '1,2,3');

// エラー抑制演算子、通称なると
// エラーを握り潰せるべんりで危険なやつ
$a = @split(',', '1,2,3');
```

Fatal Error（致命的エラー）

- これ以上は続行できないエラー
- 同名のクラスや関数を
  多重定義しようとしたとき
- ファイルをrequireできなかったとき
- 明らかに不正な式

pixiv

Fatal Errorの例(一部)

```
// PHP5系とPHP7系でエラーになる
// タイミングが異なるコード
$i = mt_rand(1, 2);
echo $i;
if ($i === 1) {
    // 数値と配列は加算できない
    $x = 1 + [];
}
```

Fatal Errorの例(一部)

```php
// PHP5系とPHP7系でエラーになる
// タイミングが異なるコード
$i = mt_rand(1, 2);
echo $i;
if ($i === 1) {
    $x = 1 + [];
}
```

PHP7: 何も出力されずFatal

それ以外: 「1」とFatal
または「2」のみ

PHP7は賢いので、実際に
実行されることがなくても
（絶対実行されないifの中だろうと）

コンパイル時に検出してくれる

pixiv

例外

pixiv

例外（Exception, Throwable）

- PHP5で追加された言語機能
- 異常状態のときにthrowで投げる
- 大域脱出（たいいき(Non-local)）と呼ばれる機能
- catch文で捕捉できる

pixiv

例外オブジェクト

- PHP5ではExceptionまたは、Exceptionを継承したクラス
- PHP7ではThrowableインターフェイスと、Exceptionを継承しないErrorとその派生クラスが追加
- 便宜上まとめて「例外」と呼ぶ

pixiv

ErrorとエラーとE_ERROR

- PHP5系でエラーが発生したものが、7系では例外(Errorクラス)を**throw**するように徐々に移行
- すべての移行されれたわけではなく、まだエラーと例外の概念を両方知っておく必要がある
- E_ERRORレベルのエラー…ややこしい

pixiv

エラー

ハンドリング

pixiv

伝統的なエラー処理手法

```
// MySQLに接続失敗したら異常終了
$db = mysql_connect() or die("接続に失敗しました");
```

- die() は exit() と同じ意味
- ここでいきなり強制終了する
- そのあと処理はほとんどできない

せめて例外に

```
if (do_something() === false) {
    throw new FooException('do_something() が失敗した');
}
```

- 現在実行中の処理を中断して、
  try…catch まで遡る
- catchできなかったら、そこで
  おとなしく野垂れ死ぬしかない…?
- ただちに終了されないので、
  まだ挽回の余地がある

エラーハンドリング

- 発生したエラーをなんとかする

pixiv

エラーハンドリング

- 発生したエラーをなんとかする
  - A: どうにか続行を試みる
  - B: どうにもならないので諦める

pixiv

エラーハンドリング

- 発生したエラーをなんとかする
  - A: どうにか続行を試みる
  - B: どうにもならないので諦める
    - せめてログだけでも…
    - なんとか共通ナビゲーションとロゴだけは表示してほしい…

pixiv

我々に三回のチャンスがある

- set_error_handler()
- set_exception_handler()
- register_shutdown_function()

error_reporting()

- 「出力するエラーの種類を設定する」
- どのエラーレベルに対応するか
  ビットフラグで設定する
- 引数なしで呼ぶと現在の値を返す
- error_reporting(E_ALL & ~E_NOTICE)

pixiv

error_reporting() 設定方法

```
// 無視できるもの全部無視
error_reporting(0);
// 全部のエラーを捕捉する
error_reporting(E_ALL);
error_reporting(E_ALL | E_STRICT); // PHP 5.3以下
// Noticeのみ無視
error_reporting(E_ALL & ~E_NOTICE);
// NoticeとDeprecatedを無視
error_reporting(E_ALL & ~E_NOTICE & ~E_DEPRECATED);
```

set_error_handler()

- エラーが発生したときに、set_error_hanlder() に登録した関数が呼ばれる
- エラーに応じて処理を続行させることもできるし終了させてもいい
- Fatal errorは捕捉できない！

pixiv

set_exception_handler()

- 例外がtry-catchでも捕捉されなかったときに、
  set_exception_hanlder() に登録した関数が呼ばれる
- 関数が実行された後は停止する
- 登録できる関数はひとつだけ

pixiv

register_shutdown_function()

- PHPスクリプト終了時に必ず呼ばれる関数を登録する
- set_error_handler()に来ない Fatal Error(致命的エラー)を捕捉してロギングできるチャンス
- 最後のエメラルドスプラッシュ

pixiv

定石

pixiv

エラー処理の定石

- デフォルトのエラー出力を切る
- エラーは例外に変換して、
  未捕捉の例外とまとめて処理する
- 処理すべきエラーかどうかの判定
  にerror_reporting()を利用する

pixiv

デフォルトのエラー出力

(!) Warning: PDO::__construct(): php_network_getaddresses: getaddrinfo failed: Name or service not known in /path/to/proj/D

Call Stack

#	Time	Memory	Function	Location
1	0.0001	253464	{main}( )	.../index.php:0
2	0.0130	1650808	AppRunner::execute( )	.../index.php:4
3	0.0132	1660512	WwwRouterAndDispatcher::main( )	.../AppRunner.p
4	0.0132	1660952	WwwRouterAndDispatcher::routeAndDispatch( )	.../WwwRouterA

デフォルトのエラー出力

```
// 必ず読まれるPHPファイルで設定する
// 少なくとも本番環境では絶対offにする
ini_set('display_errors', 'off');
```

set_exception_handler() 設定方法

```
// 例外をハンドリング
set_exception_handler(function ($e){
    // 例外をログに記録
    YourApp::logException($e);
    // エラー時の画面を表示
    YourApp::displayErrorPage($e);
});
```

エラー抑制とerror_reporting

- @(エラー抑制演算子)は一時的にerror_reporting(0)に設定して処理を実行するだけ
- 独自設定したエラーハンドラがerror_reporting()を考慮してくれないと、エラーが握り潰されない

pixiv

set_error_handler() 設定方法

```
set_error_handler('my_error_handler');
function my_error_handler (
    $level, $message, $file = null, $line = null) {
    // エラーレベルのAND(ビット積)をとる
    if (error_reporting() & $level > 0) {
        throw new ErrorException(
            $message, 0, $level, $file, $line);
    }
}
```

error_reporting()

- error_reporting()で設定した
  エラーレベルはデフォルトの
  エラーハンドラで利用される
- 自作のエラーハンドラを設定する
  ときは、error_reporting()を
  考慮して実装しなければならない
- 実際グローバル変数みたいなもの

pixiv

register_shutdown_function() 設定方法

```
register_shutdown_function(function(){
    if (empty($error = error_get_last())) return;
    // エラーハンドラーで捕捉できないエラーを判定
    $fatal = E_ERROR | E_PARSE | E_COMPILE_WARNING
          | E_CORE_ERROR|E_CORE_WARNING|E_COMPILE_ERROR;
    if (($error['level'] & $fatal_errors) > 0) {
        $e = new ErrorException($error['message'], 0,
           $error['type'], $error['file'], $error['line']);
        my_error_handler($e);
    }
});
```

例外のロギング

- PHPの標準機能error_log()は、ちょっと弱い
- rollbar, faultlineなどに送る
- pixivの場合は例外情報をバックトレースも含めてJSONエンコードしてファイルに追記、Fluentdで収集

pixiv

2017-06-13

PHPカンファレンス福岡2017でfaultlineについて発表してきました #phpconfuk

PHP

前回とほぼ同じタイトルですが、無事発表してきました。

プロフィール

id:k1LoW

検索

記事を検索

最新記事

PHPカンファレンス2017でServerless Frameworkとその知見について発表します #phpcon2017

Serverless Meetup Fukuoka #1で発表してきました #serverlessfukuoka

いかに危険なコマンド実行間違いを防ぐか。そのためにExeCopを書いてみた

Webフロントエンドハイパフォーマンスチューニング

http://k1low.hatenablog.com/entry/2017/06/13/083000

開発時の原則

pixiv

error_reporting()

- 可能な限り無視しない
  - 例: 本番ではDeprecatedを無視
- ただしプロジェクト全体にNotice発生箇所が多い場合は無理しない
  - 一気にリファクタリングしようとしてエンバグする方がずっと怖い

pixiv

エラー設定は初期化処理で実行する

- error_reportingはphp.iniでも設定可能だが、PHPで明示的に設定すれば環境依存がなくなる
- 深いところで仕方なく変更する必要に迫られたときは、小さなスコープで変更してすぐに戻す

pixiv

段階的にエラーレベルを上げる

- pixivの開発時にもまだNoticeが発生する箇所が残ってる…
- やむを得ず変更する際は浅い位置(例:コントローラ)で設定する
- Deprecatedはユニットテストで追い詰めていく

pixiv

pixivの設定

```
// 全体設定
error_reporting(E_ALL & ~E_NOTICE & ~E_DEPRECATED);
// ユニットテスト
error_reporting(E_ALL);


// 新規ファイルや動作確認できたページ
// (コントローラー)単位でレベルを上げる
setErrorReportingAll();
```

べんりヘルパー関数

```
/**
 * 開発時のみエラーレベルを最大に上げる
 *
 * 本番(運用環境)には影響しない
 */
function setErrorReportingAll() {
    if (is_development()) {
        error_reporting(E_ALL);
    }
}
```

深いところで一時的にエラーレベルを変更

```
/** mcryptは7.1で非推奨、7.2で削除予定 */
function mcrypt_wrapper($text) {
    $err = error_reporting();
    error_reporting(E_ALL & ~E_DEPRECATED);
    try {
        // mcryptを使った処理
    } finally {
        error_reporting($err);
    }
    return $result;
}
```

手を出せないライブラリ/SDK

```
/** なんかエラー吐きまくる古いSDKのラッパー */
function hoge_legacy_wrapper($text) {
    $err = error_reporting();
    error_reporting(E_ALL &~E_NOTICE & ~E_DEPRECATED);
    try {
        $result = \Hoge\Legacy\SDK::proc();
    } finally {
        error_reporting($err);
    }
    return $result;
}
```

whoops!

pixiv

whoops!

- *"PHP errors for cool kids"*
- エラー処理用フレームワーク
- set_exception_handler()は
  ひとつしかハンドラを登録できな
  いが、whoopsは複数登録できる

pixiv

Error

Unsupported operand types

COPY

Stack frames (3)

2 Error

…/src/routing.php:10

1 zonuexe\PhpCon2017\{closure}

…/public/index.php:39

0 call_user_func

…/public/index.php:39

/Users/megurine/repo/php/phpcon2017-error-processing/src/routing.php

```
1.  <?php
2.
3.  namespace zonuexe\PhpCon2017;
4.
5.  $routing_map = [];
6.
7.  $routing_map['show_index'] = ['GET', '/', function () {
8.      $i = 1;
9.      $a =  [];
10.     var_dump($i + $a);
11.     return view('index');
12. }];
13.
14. $routing_map['error_notice'] = ['GET', '/error/notice/undefined_variable', function
    () {
15.     echo $a;
16.     return null;
17. }];
18.
19. $routing_map['list_feature'] = ['GET', '/feature', function () {
20.     return view('list_feature', [
```

Arguments

```
1.  "Unsupported operand types"
```

No comments for this stack frame.

whoops! ハンドラー

- ハンドラー(関数)は複数登録できるスタック（FIFO, 後入れ先出し）
  - ロギング・デバッグだけでなくエラー画面の表示にも利用できる
- WEB+DB Press Vol.96サンプルにwhoopsを使った設定例書きました

http://gihyo.jp/magazine/wdpress/archive/2017/vol96

pixiv

whoops! 注意点

- PrettyPageHandlerはべんりだがもし本番環境で有効化されてしまうと脆弱性や情報漏洩の原因にすらなりかねない
- 設定に自信がなければ、単に開発環境用のカッコイイエラー画面として割り切るのが安全

pixiv

whoopsの有効化

```
// 開発時のみWhoopsを有効化する場合
if (is_development()) {
    $handler = (PHP_SAPI === 'cli')
        ? new \Whoops\Handler\PlainTextHandler
        : new \Whoops\Handler\PrettyPageHandler;
    $whoops = new \Whoops\Run;
    $whoops->register();
}
```

落穂

pixiv

エラーレベルとビットフラグ

- エラー発生時のレベルと
  error_reporting()はビットフラグ
- E_ALLは全種類のエラーの
  フラグを立てた状態
- E_ALL & ~E_NOTICEのように
  反転してビット積をとるのが簡単

pixiv

ビットフラグの確認方法

```
printf('%015b', E_ALL);
// => "1111111111111111"
printf('%015b', E_DEPRECATED);
// => "0100000000000000"
printf('%015b', E_ALL & ~E_DEPRECATED);
// => "1011111111111111"
```

trigger_error()

- エラーを意図的に発生させる
- ただし全てのエラーを発生させられるわけではなくE_USER系のみ
- 今日エラーはあまり考慮されてないので、役立たないかも…
- USERの有無は別のエラーなので

pixiv

trigger_error利用例

```
// 関数をリネームしたいとき
/**
 * @deprecated {@see grant()} を使ってください
 */
function credential(array $data){
    $msg = 'credential() is obsolete function.  Use
grant() instead.';
    trigger_error($msg, E_USER_DEPRECATED);
    return grant($data);
}
```

もうひとつのエラーハンドリング

- set_exception_handler()を使ったハンドリングは…実際(説明)難しい
- 細かなエラーハンドリングは try-catchを使った方がやりやすい
- エラーの仕組み(このセッションで説明した内容！)よりは把握しやすい
- https://gist.github.com/zonuexe/577d089815e241d5da26

pixiv

pixivの場合（AppRunnerパターン）

- アプリケーション(PC版, スマートフォン版, etc...)ごとにAppRunner(社内用語)と呼ばれるクラスを定義する
- 例外の種類ごとにcatchして、適切な処理(ロギング・エラー表示)などをする
- http://inside.pixiv.net/entry/2015/12/18/210721

pixiv

pixivの場合（なぜこのパターンか）

- 例外にも種類がある
  - アプリケーションのバグ
  - クエリパラメータの異常入力
- それらのアプリケーションの事情に沿ってロギングする

pixiv

参考資料

pixiv

参考資料

- だいたいPHPマニュアルに載ってる
  - http://php.net/language.operators.errorcontrol
  - http://php.net/set_error_handler
  - http://php.net/set_exception_handler
  - http://php.net/register_shutdown_function
  - http://php.net/ErrorException
  - http://php.net/display-errors
- whoops! https://github.com/filp/whoops

参考(にすると良い)資料

- PHPのエラーと例外再入門
  by Hiraku NAKANO
  https://speakerdeck.com/hirak/php-error-and-exception

- faultline/faultline-phpによるサーバ管理不要なエラートラッキングとその効果 by Ken'ichiro Oyama
  http://k1low.hatenablog.com/entry/2017/06/13/083000

参考資料（pixiv AppRunner）

- （ピクシブ株式会社 Advent Calendar 2015）
  健全なpixivは健康なPHPに宿る～
  モダンPHPを保つ7つの鍵
  http://inside.pixiv.net/entry/2015/12/18/210721

- ソースコード（サンプル）
  https://gist.github.com/zonuexe/577d089815e241d5da26

最後だ

pixiv

メンバー全員が知るべき？

- 知っておくとためにはなる…かも
- でも完全に理解してないといけないわけではない
- とは言っても社内で誰かが知っておかないと困ることではある
- 「Noticeを潰せ」だけ周知したい

pixiv

お絵描きがもっと楽しくなる場所を創る

プレスリリース

Webサービスの開発基盤を設計したいひとも

お絵描きがもっと楽しくなる場所を創る

pixiv

pixiv 2018年度新卒エントリー受付中

プレスリリース

創作文化のための開発に興味があるひとも

お絵描きがもっと楽しくなる場所を創る

pixiv

pixiv 2018年度新卒エントリー受付中

プレスリリース

今回の内容がいまいち ぴんと来なくても大丈夫

お絵描きがもっと楽しくなる場所を創る

pixiv

プレスリリース

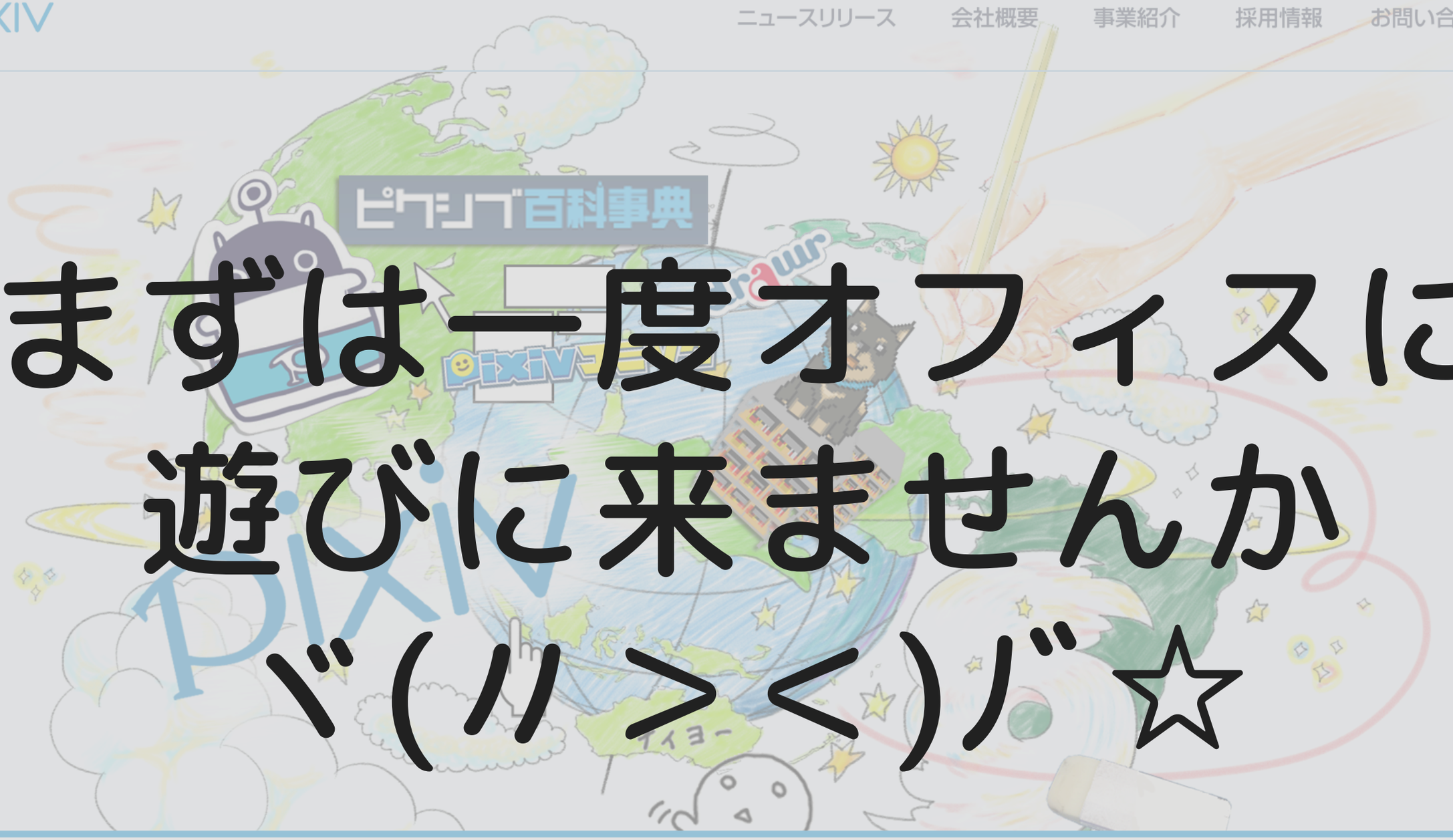
PHP, JavaScript, Ruby, Scala, Go, データ分析, インフラ, その他…

お絵描きがもっと楽しくなる場所を創る

pixiv

プレスリリース

pixiv 2018年度新卒 エントリー受付中



まずは一度オフィスに遊びに来ませんか ヾ(//＞＜)ゞ☆

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

pixiv

プレスリリース

Twitter @tadsan宛
または
recruit.pixiv.net

使用フォント

セプテンバーＭ・Ｌ

Migu 1M

Rounded M+ 2p