

Real World PHP in pixiv改

pixiv 2016 SPRING BOOT CAMP

2016年3月15日

お前誰よ



- うさみけんた / tadsan
- 2012年11月にピクシブ入社
 - 入社前は自宅警備しながらRuby書いてた
- pixivのPHPもろもろやってます
- WEB+DB PressにPHPの記事を書きました
- 好きなリスプはイーマックスリスプです

注意



- ~~時間配分は考慮してない~~
- 開発楽しい！とかじゃなく淡々と羅列するだけ
- ゆっくり話すので、「お前何言ってるんだ」と
感じたらつつこみを入れてください
- 資料はあとで公開します

アジェンダ



- 現状についてだらだら話す
 - **pixiv**のデプロイについて
 - **pixiv**の開発環境について
 - **pixiv**のPHPについて
 - **pixiv**のSQLについて
 - その他

GitLab (2015年10月頃)

The screenshot shows the GitLab web interface for the `pixiv / pixiv` project. The top navigation bar is purple and contains the project name, a search bar, and various icons. The main content area has a white background. On the left, there is a sidebar with icons for repository, files, activity, and other features. The main content area displays the project name `pixiv.net` main repository - README.md, the repository URL `git@gitlab.pixiv.private:pixiv/pixiv.git`, and the repository status (private). A summary box highlights the repository statistics: 61,994 commits, 186 branches, 216 tags, and 1404.61 MB. Below this, the README.md file is shown, followed by the project name `pixiv.git` and a description in Japanese: 「ここはピクシブのむらです」. The word `Setup` is also visible, along with the text `開発環境`.

pixiv / pixiv

Search in this project

pixiv.netメインリポジトリ - README.md

★ Star 0 Fork 0

SSH HTTP git@gitlab.pixiv.private:pixiv/pixiv.git private Download zip

Activity Readme

61,994 commits 186 branches 216 tags 1404.61 MB

README.md

pixiv.git

??? 「ここはピクシブのむらです」

Setup

開発環境

GitLab (2016年3月)

The screenshot displays the GitLab web interface for the 'pixiv / pixiv' project. The top navigation bar includes the project name, a search box, and icons for adding new projects or repositories. Below the navigation bar, the project's statistics are shown: 4 stars, 0 forks, and 0 watchers. The SSH and HTTP clone URLs are provided, along with a 'WATCHING' button. A table of statistics is highlighted with an orange border, showing 74,379 commits, 1,443 branches, 12,409 tags, and 1712.52 MB of size. Below the statistics, there are buttons to 'ADD CHANGELOG', 'ADD LICENSE', and 'ADD CONTRIBUTION GUIDE'. A recent commit is listed with the message 'Merge branch 'premium/text-translations-for-premium-page' into 'master'' by user 'lainbsd'. The main content area shows the project's README, which includes a 'Setup' section with instructions on how to clone the repository using SSH. A code block at the bottom of the README shows the commands: `git clone git@gitlab.pixiv.private:pixiv/pixiv.git` and `cd pixiv`.

pixiv / pixiv

Search in this project

★ 4 0 SSH HTTP git@gitlab.pixiv.private:pixiv/ WATCHING

74,379 COMMITS 1,443 BRANCHES 12,409 TAGS 1712.52 MB ADD CHANGELOG ADD LICENSE ADD CONTRIBUTION GUIDE

5be0754d Merge branch 'premium/text-translations-for-premium-page' into 'master' · about 12 hours ago by lainbsd

pixiv.git

??? 「ここはヒ° クシブ° のむらです」

Setup

開発環境

SSHでpixiv-devにログイン推奨

```
git clone git@gitlab.pixiv.private:pixiv/pixiv.git
cd pixiv
```

pixiv.git (クラスのお友達)

- └ accounts.pixiv.net
- └ admin.pixiv.net
- └ api.pixiv.net
- └ batch
- └ dev-script
- └ embed.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
- └ spapi.pixiv.net
- └ spotlight.pics
- └ ssl.pixiv.net
- └ touch.pixiv.net
- └ util
- └ vendor
- └ web-test
- └ www.pixiv.net

変遷

- 以前(2013年頃まで)は、サブディレクトリごとに独立したGitリポジトリで、共通部分をsubmoduleとして扱ってた
- 共通部分の抽象化もうまくなされて居らず、重複・細部が異なるコードが分散してた

変遷

- submoduleは疎結合なモジュールに適する
- 結合度が高いモジュールのリポジトリを分割すると、変更コストが極めて高い
- リポジトリを統合することで凝集度を高めるリファクタリングも断行しやすくなった
- 細部が異なる重複コード問題も根絶

デプロイ

- 基本的にサーバー全台(××台程度)にrsync
- 以前(2013年)はPHPファイルの増減を伴ったデプロイ時には(ページにもよるが)、×××件程度のエラーが発生してしまってた
- 現在はBlue-GreenDeploymentの確立で、ファイルの追加/削除も怖いものではない

便利なリンク

[ガイドライン / デプロイ後にすべきこと / What do after deploying](#)
[エラーログ \(error log viewer\) / munin \(server resource monitor\)](#)
[www / touch / m / ssl / spapi / api / embed / pixiv.me / admin](#)
[www.stage / touch.stage / m.stage / ssl.stage / spapi.stage / api.stage / embed.stage / me.stage\(error\) / admin.stage](#)

最近のコミット

93ea7d2	uchien	HEAD release/2015.10.23.15.29.07.uchien origin/master	origin/HEAD	October 23, 2015 3:28:03 PM JST
Merge branch 'feature/halloween2015-ranking'				
M pixiv-lib/eventDataSet.php				
dcbec33	uchien	origin/feature/halloween2015-ranking		October 23, 2015 3:26:10 PM JST
ハロウィン2015ランキング表示用の設定				
M pixiv-lib/eventDataSet.php				
8f0463e	tadsan	release/2015.10.22.23.00.53.tadsan		October 22, 2015 11:00:14 PM JST
投稿後画面で小説の場合はデータを引かない				
M www.pixiv.net/htdocs/content_upload_fix.php				
dc28e66	tadsan	release/2015.10.22.20.53.20.tadsan		October 22, 2015 8:52:43 PM JST
pixiv-lib: テスト修正しわすれ				
M pixiv-lib/unit_test/PixivTest/DB/NovelRankingDao/GetByDateTest.php				

tadsan

デプロイを開始する

プロジェクト

- app-setting.pixiv.public
- chat.pixiv.net
- contest-cms
- mascala
- pixiv
- pixiv-log-transmitter
- pixpedia
- plaza
- pploy-dist
- pxvadm
- special-resource
- testploy
- yufulight-deliver

デプロイ (pploy)

- 汎用的なデプロイシステム
- /.deploy ディレクトリにファイルを配置
 - bin/deploy スクリプトが実行される
 - readme.html で画面カスタマイズ可能
- Gitに依存するので、特定のバージョンを
チェックアウトして反映することも容易

デプロイと変遷 (ブログ記事紹介)

- 開発・デプロイ環境の変遷 2014年春版
- 履歴を残したまま複数のgitリポジトリを統合する
- pixivのデプロイを支えるpploy
- github: edvakf/pploy
- WEB+DB PRESS Vol. 84

pixivの開発環境

- DBなどの依存が多く、ローカルに環境を構築することは容易ではない(不可能でもない)
- 共有の環境にSSHでログインして開発
 - 端末のEmacsやVimで開発するひとも
 - rsyncでローカルと同期してPhpStormなど
 - @tadsanはTRAMPを利用してる

pixivの開発環境

- ApacheのVirtualHostを大胆に活用
- /home/www/にシンボリックリンクを作成
 - 社内ネットワークでDNS名前解決できる
 - www.tadsan.example.com のような形式
- pixiv関係なくとも任意のPHPを動作可能
- OpenStackで同じ仕様のサーバー増やせる

pixiv-lib

- DBの操作を抽象化する層(DAO)
- データの入出力などをまとめて扱う層
 - 社内では「Common層」と呼ばれる
- セッション・国際化などの基盤機能
- その他の共通処理いろいろ！
 - むかしはextlibって呼ばれてた (懐古)

pixivのPHPコード

- PHP 5.5 + Apache
 - URLを見ればわかる通り、ページ=PHP
 - search.php, member_illust.php など...
 - 一部のリクエストはHHVMで捌いてる
 - ルーティングのためmod_rewriteに依存

pixivのPHPコード

- 既存のフレームワークに依存しない
- 独自フレームワークを運用してる部分もある
 - 口にすると顔をしかめられがちだが、
用途に合わせて設計ができるのは悪くない
- 何を使ったとしても、サービス提供者として
セキュリティの責任は変らない！

pixivのPHPコード

- 動的な言語機能は必要ない限り利用しない
- コードの把握しやすさ、
 - クラスの静的メソッドを多用する
 - 継承は排除する方向
 - トレイトもあまり利用しない
 - リファレンスは原則排除する
 - PhpStormを使ってもつらいものはつらい

pixivのPHPコード (昨年10月)

- `include_once`ってたくさん書いてた！！
- オートローディングの利用は限定的
 - `pixiv-lib`では現状は利用不可 (改善してた)
- リポジトリ内に混在する都合上、
Composerのautoloadは利用しにくい
- 一部ではphpabで生成してinclude

pixivのPHPコード (現在)

- `include_once`を撲滅した！！
- オートローディングを大幅導入
 - `pixiv-lib`では完全移行が完了
 - シンプルなクラスローダを手書き
 - 一部ではphpabで生成してincludeしてる

pixivのPHPコード

- コミットログやコード内コメントは、
英語または日本語で各人の得意な方で
 - 意識高く「全部英語で」など拘らない
 - 必要な文章は論理的に誤りがないように
- コミットのルールは厳格に定めない
 - 各人のやりかたを尊重する
- 典型的な Github Flow で運用

loadset.php

- 各プロジェクトの初期化ファイル
 - エラーハンドラーの設定
 - 必要なライブラリのロード
 - 定数(define)の定義
- loadsetのみrequire_onceで読み込む
 - そのほかは全部 include_once で

pixivのSQL

- MySQL 5.5
- 文字列結合によるSQL組立つらすぎ...
- ORMマッパー/自動クエリビルダーは利用せず
- 2015年の初旬からSQLに型をつけて安全に書けるライブラリを社内で運用
 - 脆弱性対策の文脈で生まれたが書きやすい


```

/**
 * @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);
    $stmt = PxxSql::build($con, NovelRankingDao::getByNovelId_sql, [
        ':novel_ids' => $novel_ids,
        ':limit' => $limit,
        ':offset' => $offset,
        ':order' => $order,
    ]);
    $stmt->execute();

    return $stmt->fetchAll(PDO::FETCH_ASSOC);
}
const getByNovelId_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
';

```

pixivのSQL

- Q: なぜ自動クエリビルダー利用しないの？
 - いままでが文字列結合で組み立ててたから
 - クエリが単純ではないパターンも多い
 - 例: ブックマークなど

pixivのWebAPI

- いろんな試行錯誤をしてきた
 - 一部で有名なCSVのAPI (最初期)
 - ActiveResourceで利用するためのAPI
 - 提携企業向けRESTful API
 - 社内向けのシンプルなAPI (RPC)
 - スマートフォン用API (not REST)

pixivのWebAPI

- なぜRESTfulか？なぜRESTじゃないのか？
- 慎重に判断する必要がある
 - API(RPC)から実行したい処理の粒度と
REST/CRUDの概念が馴染むとは限らない
- 外部公開するのでなければ、
綺麗なURLの価値はあまり高くない

pixivのWebAPI 設計

- モバイルアプリはRESTを捨てる(採用しない)
決断をした
 - HTTP Methodも全てPOSTで受ける
- 社内API(RPC)は Apache Thrift を採用した
- どんなAPIが最適かは時と場合による

パッケージ管理

- Composerを利用
 - composer.pharはpixiv.gitに同梱
- 社内ネットワーク内でSatisを運用
 - Jenkinsで定期的に更新
 - GitHubの障害があっても支障なし
 - 手動でパッケージ更新もできる

pixivのテスト

- 基本的にPHPUnitでテストを書く
- 2013年頃にテストを書く文化が定着した
- GitlabのMRにpushしたら自動テスト走る
- うっかりmasterにpushされたら即座に
idobataに通知される

pixivのレビュー体制

- 毎週pixiv.gitに携る各チームのレビューアーで設計方針や最新の情報などを共有・相談
- レビューアーが各チームに持ち帰る
 - 各自メンバーに説明したり、レビュー過程で指摘することで全体のコード品質を保つ
 - ガイドラインや推奨スタイルの確立

そのほか

- [pixiv](#)のデプロイ回数は一日およそ40回
 - ディレクターが文言変更をすることが多い
- Jenkinsをいっぱい使ってる
 - cronの代替として
 - スクリプトのWebUIとしても利用できる
- 各種エディタの勢力は均衡してる

ふろく (WEB+DB PHP連載)

- 特におもしろい記事
 - Vol.81: テストしにくいコード対策
 - Vol.83: 安全なコーディングスタイル
 - Vol.84: デプロイ・Composer運用
 - Vol.87: PHPDoc記法で型注釈
 - Vol.88: HHVMの運用
 - Vol.91: 名前空間とオートローディング

続きはWebで



- <https://github.com/pixiv>
- <http://inside.pixiv.net/>
- **GitHub** (tadsan個人で公開してるもの)
 - [BaguettePHP/simple-routing](#)
 - [BaguettePHP/http-accept-language](#)
 - [BaguettePHP/TetoSQL](#)