

われわれは、いかにして
変更点を追うか

ChangeLog/Issueを追う技術

自己紹介

azu

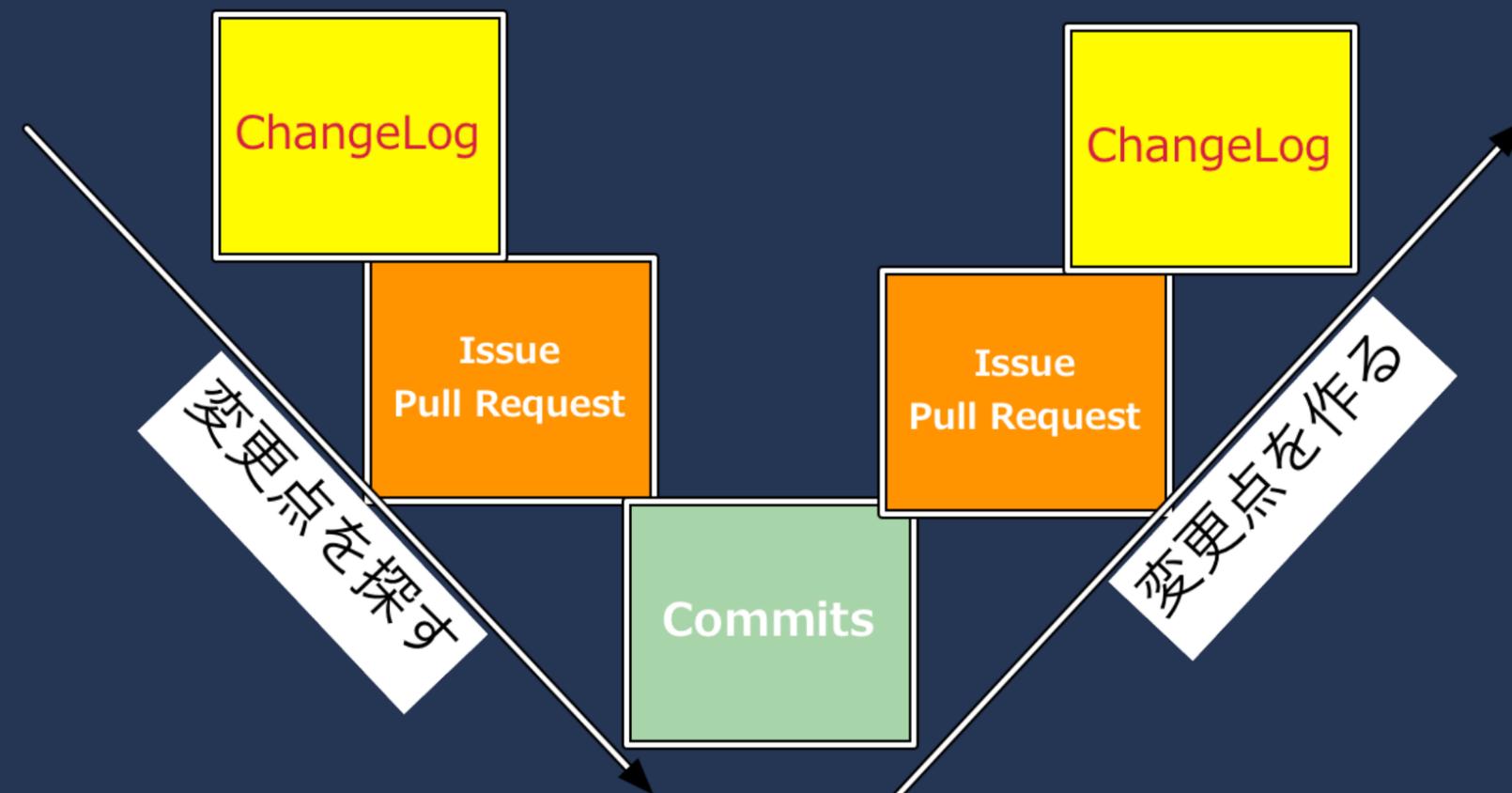
@azu_re

Web scratch, JSer.info

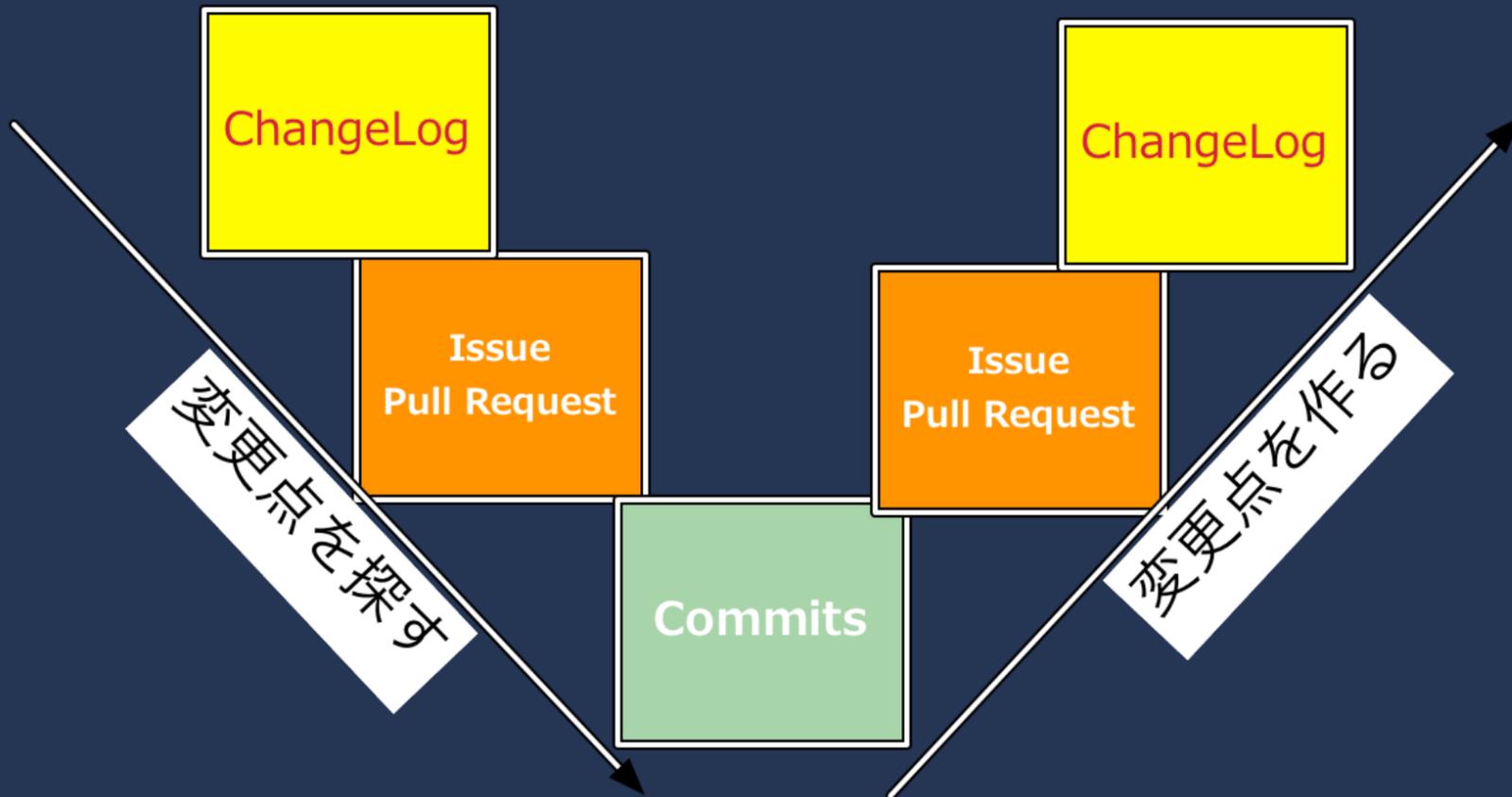


アジェンダ

- 変更点を
 - ChangeLogで知る
 - Issue/Pull Requestで知る
 - Commitで知る



アジェンダ



- 変更点を
 - Commitに書く
 - Issue/Pull Requestを扱う
 - ChangeLogにまとめる

変更点を追う



ChangeLogの追い方

- ChangeLogを追うにはまずChangeLogの更新に気づく事が必要
- GitHubでライブラリのリリースを見ていくためのツールや方法に書いた
- 更新検知の仕組みや補助ツールについて

- リポジトリのWatch => [azu/github-reader](#)
- タイムライン => [azu/github-reader](#)
- Star => [starseeker](#)
- GitHub ReleaseのRSS => [azu/github-releases-to-feedly](#)
- GitHub Releaseの閲覧補助
 - [azu/checkchangelogfrom release](#)
 - [azu/show-diff-from-release](#)

ChangeLogの問題

- 全ての変更がChangeLogからわかるわけじゃない
- そもそも多くの人には面倒なのでChangeLogを書かない
- ChangeLogがあってもそれはサマリだと、細かい内容が分からない
- ChangeLogより更に細かく見たい場合はまずは**Issue/Pull Request**を探す

Issue/Pull Request

- 大きなライブラリほどIssueやPull Requestと関連付けてコミットされる
- ChangeLogにも該当Issueが書いてあることが多い
- => ChangeLogからIssue/PRを探すとより細かい情報が見られる
- ChangeLogにIssue/PRが書いてない場合は?

Issue/PRの探し方 - 検索



Babel 5.3.0の例

Babel 5.3.0の変更点

- 変更内容については書いてある
- **[New Feature]**、**[Bug Fix]**、**[Spec Compliancy]**などジャンル分けされている
- "Speeeeeeed" とあるがどれくらい変わったのか書いてない
- IssueやPR、該当コミットへのリンクはない

5.3.0

Speeeeeeed



- **Spec Compliancy**
 - Allow trailing param commas for methods when using the `es7.trailingCommas` transformer.
- **Bug Fix**
 - Fix `es6.blockScoping` transformer not properly ignoring `break` in `SwitchCase`.
 - Fix lookahead context saving to avoid weird tokenizer state.
 - Explode duplicate identifiers in export/import specifiers and property shorthand to create unique objects.
 - Skip loose mode for class methods when they have decorators.
 - When removing nodes, share their comments with their siblings.
 - Properly hoist temp param declarations when doing TCO.
- **Internal**
 - Add `--harmony_generators` flag to `$ babel-node`.
 - Internal AST traversals have been minimised **drastically**. Transformers have been grouped together which means entire tree traversals are much fewer. Visiting nodes is now also skipped if the traversal context can detect that the handler is a noop. This saves precious cycles as it avoids constructing traversal paths and creating a new traversal context. See issues [#1472](#) and [#1486](#) for related discussion.
- **Polish**
 - Move many `utility` transformers to `minification`.

Speeeeed??

具体的な速度の向上内容を知りたい!

Speeeeed - 何がどう変わったのか

Twitterで検索

- <https://github.com/babel/babel/blob/master/CHANGELOG.md#530> への言及から探す
- "Speeeeed Babel"で検索する
- [公式アカウント](#)から探す

Speeeeed - 見つかった

- 20-40% ぐらい速度が上がったとのこと(公式アカウントより)



BABEL Babel
@babeljs

5.3.0 has been released with a ~20-40% speed improvement. Upgrade to cut your @emberjs and @reactjs application build times (almost) in half

RETWEETS 240 FAVORITES 171

10:47 AM - 13 May 2015

The screenshot shows a tweet from the official Babel account (@babeljs). The tweet text states that version 5.3.0 has been released, offering a ~20-40% speed improvement and significantly reducing application build times for users of @emberjs and @reactjs. The tweet has received 240 retweets and 171 favorites. The timestamp is 10:47 AM on May 13, 2015. The interface includes a 'Following' button and icons for reply, retweet, favorite, and more options.

Speeeeed - Depth

- パフォーマンスの向上は鵜呑みにしない ⚠
- 何が変更して、どういう計測方法で変わったのかを調べる
- 該当IssueやPRを見つけたい 🌱
- 💡 公式アカウントへのレスで発見 !

Babel @babeljs · May 13
5.3.0 has been released with a ~20-40% speed improvement. Upgrade to cut your @emberjs and @reactjs application build times (almost) in half

240 171

Addy Osmani @addyosmani Following

@babeljs changelog for anyone like me that was curious about where the perf gains came from: [github.com/babel/babel/bl ...](https://github.com/babel/babel/bl...)

Greenwich, London

GitHub

babel/babel
babel - Babel is a compiler for writing next generation JavaScript.

BABEL

View on web

RETWEETS 7 FAVORITES 8

4:52 AM - 14 May 2015

Found Issue links

Reply to @addyosmani @babeljs

Sebastian McKenzie @sebmck · May 14
@addyosmani Whoop, no idea how I missed actually adding what was done. Ref issues: github.com/babel/babel/pu... github.com/babel/babel/is... @babeljs

Speeeeed - 結果

- 以下のIssueを見ていけばいいことがわかった
- [Speeeeed · Issue #1486 · babel/babel](#)
- [Merge internal transformers into single traversal pass by sebmck · Pull Request #1472 · babel/babel](#)
 - 6.3.0でマージされたPR
 - 内部のtransformerを最適化する変更が入った

Speeeeed - 結果

- tserver.js、Traceur、Ember等を対象にして大体20-40%ぐらいコンパイル速度が上がった
- Tweetの根拠が分かった
- 根拠や理論は分かった => 実際に自分で計測する
- プロジェクトによっては誤差レベルだった



```
babel@5.0.0  
npm run build 2.20s user 0.21s sys 120%  
cpu 2.006 total
```

```
babel@5.3.0  
npm run build 2.13s user 0.23s sys 116%  
cpu 1.984 total
```



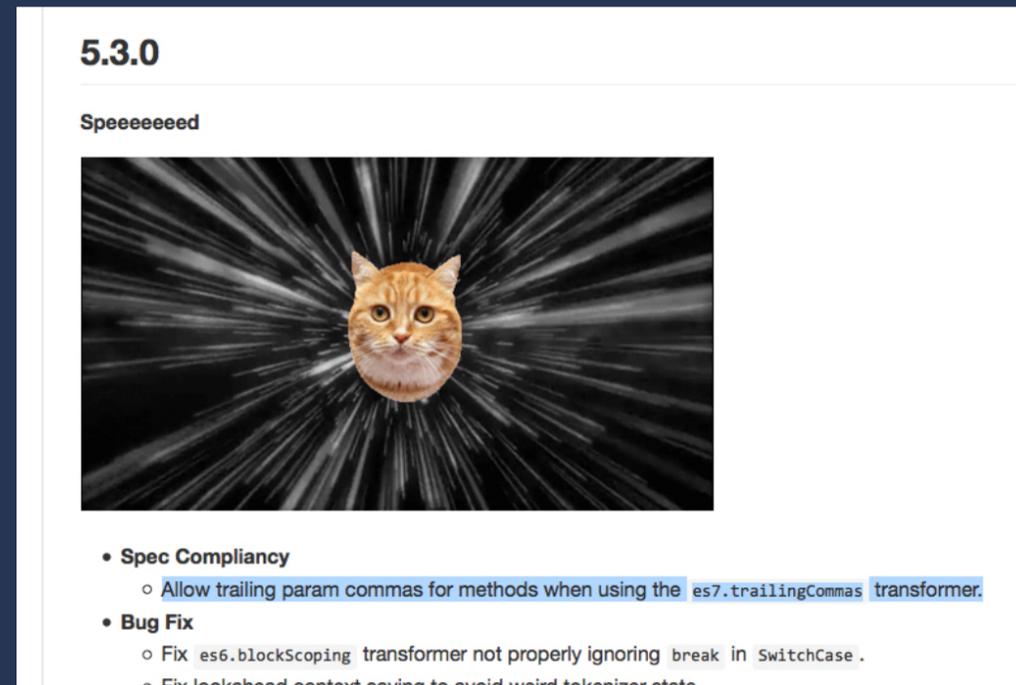
10:40 AM - 13 May 2015

Speeeeed - 結論

- ChangelogからTwitter、GitHub Issueを探せた
- [Merge internal transformers into single traversal pass by sebmck · Pull Request #1472 · babel/babel](#)
 - 複数のtransformerを1 passで通す変更が入った
- transformerを使うプロジェクトでは変換速度が上がった
- transformerを使ってないプロジェクトでは影響が少ない

Next: ファイルから探す

es7.trailingCommasの仕様への追従

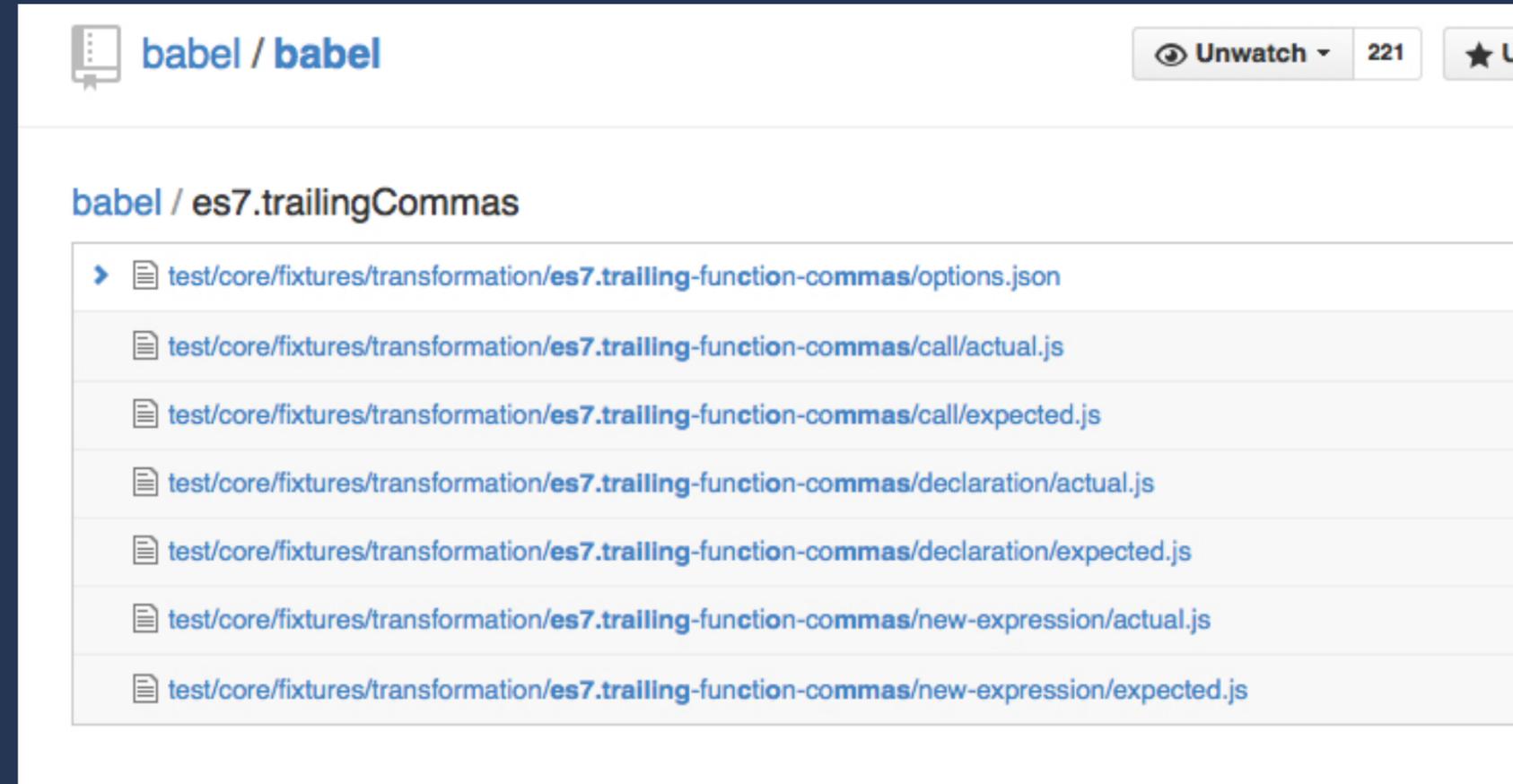


Allow trailing param commas for methods when using the `es7.trailingCommas` transformer.

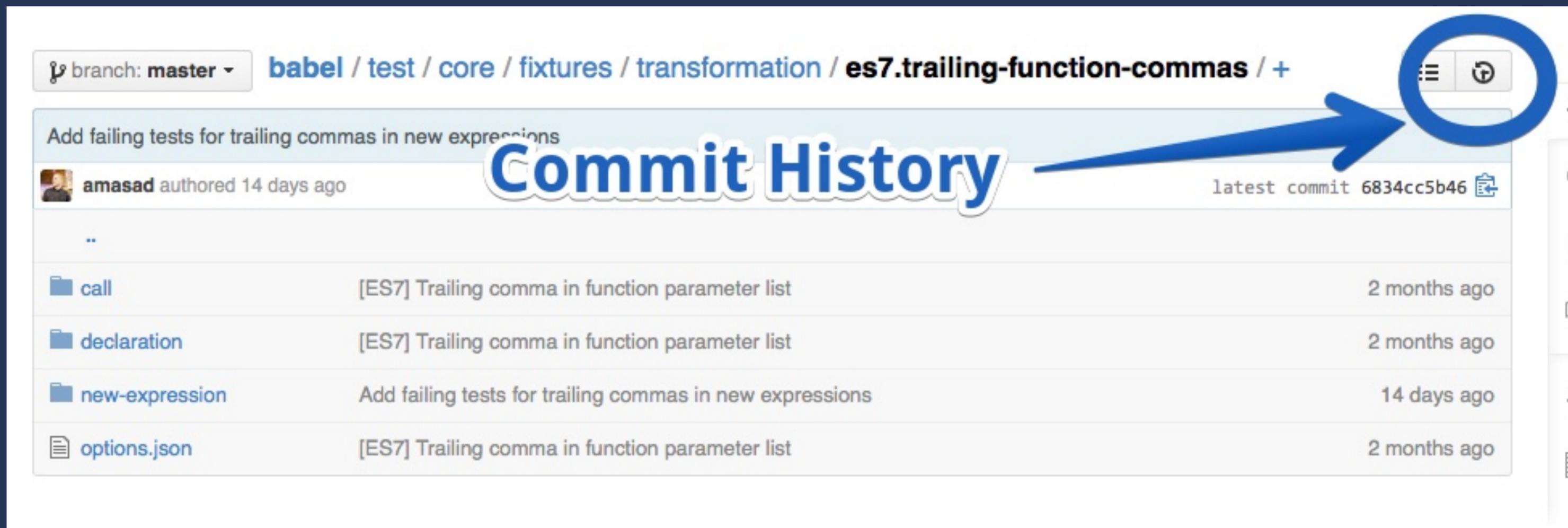
どういう変更だったのか詳細を知りたい💭

es7.trailingCommas の仕様への追従

- Babelではes7.trailingCommasの変換するためのtransformerはファイルを分けて実装されてる
- GitHubでは"T"でファイル検索ができる
- [github](#) のプロジェクトのページでファイルのインクリメンタル検索 - わからん



ファイルからコミットを探す



branch: master ▾ babel / test / core / fixtures / transformation / es7.trailing-function-commas / +

Add failing tests for trailing commas in new expressions

Commit History latest commit 6834cc5b46

amasad authored 14 days ago

..		
call	[ES7] Trailing comma in function parameter list	2 months ago
declaration	[ES7] Trailing comma in function parameter list	2 months ago
new-expression	Add failing tests for trailing commas in new expressions	14 days ago
options.json	[ES7] Trailing comma in function parameter list	2 months ago

コミットからIssueを探す

- 該当コミットからIssueへのリンクがあるかを見る
- GitHubではPull Requestでマージされたコミットに自動的にリンクが付く
- Linking merged pull requests from commits



The screenshot shows a GitHub commit page for the title "Add failing tests for trailing commas in new expressions". The commit is on the master branch (#1487) and is linked to pull request v5.4.7. A blue arrow points from the pull request link to the commit hash. Below the commit details, the diff shows two files: test/core/fixtures/transformations/es7.trailing-function-commas/new-expression/actual.js and test/core/mixtures/transformations/es7.trailing-function-commas/new-expression/expected.js. The diff highlights the addition of a new expression with a trailing comma. A large blue text overlay reads "Found Pull Request Link".

Add failing tests for trailing commas in new expressions

master (#1487) v5.4.7 v5.3.0

amasad authored 14 days ago 1 parent 9b949e5 commit 68

Showing 2 changed files with 4 additions and 0 deletions.

```
1 test/core/fixtures/transformations/es7.trailing-function-commas/new-expression/actual.js
... @@ -0,0 +1 @@
1 +new Foo(a, b,);
```

Found Pull Request Link

```
3 test/core/mixtures/transformations/es7.trailing-function-commas/new-expression/expected.js
... @@ -0,0 +1,3 @@
1 +"use strict";
2 +
3 +new Foo(a, b);
```

ファイルから探す - 結論

- 変更履歴から関連するファイルを検索
- そのファイルについてのコミットを見る
- 該当コミットからPull Requestを発見出来た 💡
- [\[es7.trailingFunctionCommas\] Allow in new expressions by amasad · Pull Request #1487 · babel/babel](#)

ここまでのまとめ

追う時に何を見つけたら詳細が分かるか？

- GitHubのリポジトリ検索
 - Issueも検索できるのでキーワードから探す
- 関連ファイルからの検索
 - 関連ファイルから変更されたコミットを探す
- Twitter検索
 - その変更に関係することを書いてるを探す
- 公式サイト/ドキュメント
 - リリース時にドキュメントが更新されてるかも

ここまでのまとめ

- Issueを見つければ勝ち 🍑
- Pull Requestを見つければ勝ち 🍑
- Commitを見つければ勝ち 🍑
- もしコミットすら見つからなかったら ?
- ここまでにはブラウザだけでできるので比較的気軽

コミットを検索する

ここからはコマンドラインが必要

GitHubの問題

- コミットメッセージを検索する手段がない
 - [Add support for search in commit messages · Issue #352 · isaacs/github](#)
- repositories、code、Issue、userしか検索できない
 - [Search | GitHub API](#)

ノイズが多い場合

Added "env" option. Especially handy when using the .babelrc.

- 上記の変更の詳細を知りたい！
- Issue検索だとノイズが多くて探せない時
- 直接該当する変更のコミットを探す

5.4.0

• New Feature

- Added `function bind syntax` behind stage 0. Thanks [@RRReverser!](#)
- Added `env` option. Especially handy when using the `.babelrc`.

• Bug Fix

- Fix files not properly being ignored when `babel.transform` ignores the
- Fix scope tracking registering loop head bindings to their `VariableDe`

• Polish

- Normalise path separators for source map paths when using `$ babel`
- Rework `PathHoister` to ignore global references and to not deopt on instead it tries to hoist to the highest scope.
- Added missing exponential operator inlining. Thanks [@nkt!](#)
- Optimise `regenerator` transformer. Thanks [@benjamn!](#)

該当コミットを探す

```
git clone https://github.com/babel/babel.git
git log v5.3.0...v5.4.0 --grep "env"
# v5.3.0からv5.4.0の中で"env"というコミットメッセージを探す
```

```
commit 41b5607ef374945b0e4ca771644d94d3b849ed58
Author: Sebastian McKenzie <sebmck@gmail.com>
Date:   Fri May 15 00:11:28 2015 +0100
```

```
fix env option tests
```

```
commit 024e4454a1e3778b0f9b6d081d5f4e792f6035db
Author: Sebastian McKenzie <sebmck@gmail.com>
Date:   Thu May 14 23:47:56 2015 +0100
```

```
add env option - closes #1531
```

The diagram illustrates the mapping of command-line arguments from the terminal command to their respective sections in the `git log(1)` manual page. The command `git log(1) v5.3.0...v5.4.0 -S "env"` is shown at the top. Three boxes below the command are connected by lines to their corresponding sections in the manual page:

- The box containing `git log(1)` points to the "Show commit logs" section.
- The box containing `v5.3.0...v5.4.0` points to the `<since>..<until>` section.
- The box containing `-S "env"` points to the `-S<string>` section.

The manual page content includes:

- Show commit logs**
- `<since>..<until>`
Show only commits between the named two commits. When either `<since>` or `<until>` is omitted, it defaults to HEAD, i.e. the tip of the current branch. For a more complete list of ways to spell `<since>` and `<until>`, see [gitrevisions\(7\)](#).
- `[--] <path>...`
Show only commits that are enough to explain how the files that match the specified paths came to be. See "History Simplification" below for details and other simplification modes.

To prevent confusion with options and branch names, paths may need to be prefixed with `--` to separate them from options or refnames.
- `-S<string>`
Look for differences that introduce or remove an instance of `<string>`. Note that this is different than the string simply appearing in diff output; see the [pickaxe](#) entry in [gitdiffcore\(7\)](#) for more details.

コミットメッセージの検索

- explainshell.com - `git log v5.3.0...v5.4.0 -S "env"`
- `024e4454a...`が該当コミットっぽい
- [motemen/git-browse-remote](https://motemen.github.io/git-browse-remote/)でGitHubのコミットページを開く
- 関連するIssueが見つかった！💡

`git-browse-remote 024e4454a1e3778b0f9b6d081d5f4e792f6035db`



コミットのDiff検索方法色々

- コミットメッセージのみで検索
 - `git log --grep "word"`
- コミットの中身の差分から検索
 - `git log -s "word"`
- [git logでコミットの差分の中身で絞り込む - Qiita](#)

コミットの検索する必要性？

- できればそこまで行く前に解決してた方が嬉しい

Added "env" option. Especially handy when using the .babelrc.

- この例もドキュメントを見るのが正解！
- [Options · Babel](#)
- しかし、どういうユースケースなのかはIssueに残ってる場合が多い

GUIでコミット検索

- [azu/git-scriptable-search](https://azu.github.io/git-scriptable-search/)
- 作った
- JavaScriptでフィルターを書いてコミットの絞込出来るアプリ
- フィルターの処理コストが大きい問題がまだ未解決...

Git directory : /Users/azu/Downloads/babel/.git

```
1 // http://www.nodgit.org/api/commit/  
2 var results = commits.filter(function(commit){  
3   return /test/.test(commit.message())  
4 });  
5 done(results);
```

Run

Filter:

add 5.4.7 changelog

2ec1544f2d469c7f9902b12df36e68fcf1345c54

add 5.4.6 changelog

c7b507e11929ae718d659bc631a45632ae2a6615

add 5.4.5 changelog

82b54794365699b5f637da2a01c5cb17b6f02657

add 5.4.4 changelog

b36cf11ac8fe3b4c811088ed498c00485f65f376

add 5.4.3 changelog

9ae54d2a50a35a120528f1ff3fe7ecd3ec1bef1d

add 5.4.1 changelog

e15ced2922c9bd34d288a5b6c10b9949f4c2656f

add 5.4.0 changelog

3910d18bc9c2b0184a0691d0257eb8c6e03c01d5

Breaking Change検索

Breaking Changeの検索

- ある機能があるバージョンで壊れてた
- 具体的に壊れた/壊した理由を知りたい！
- どこで壊れたのかコミットをみつける
 - コミットが見つかったIssueを探す

git bisectを使った検索

- 任意のテストスクリプトをコミット毎に走らせて、テストが失敗するコミットを見つける
- [AdventCalendar - git bisect で問題箇所を特定する - Qiita](#)
- [git bisectでメソッドが削除されたコミットを探す - しんぷおにゃん](#)
- さすがにここまで行くとやり過ぎ感...

ChangeLog

ChangeLog

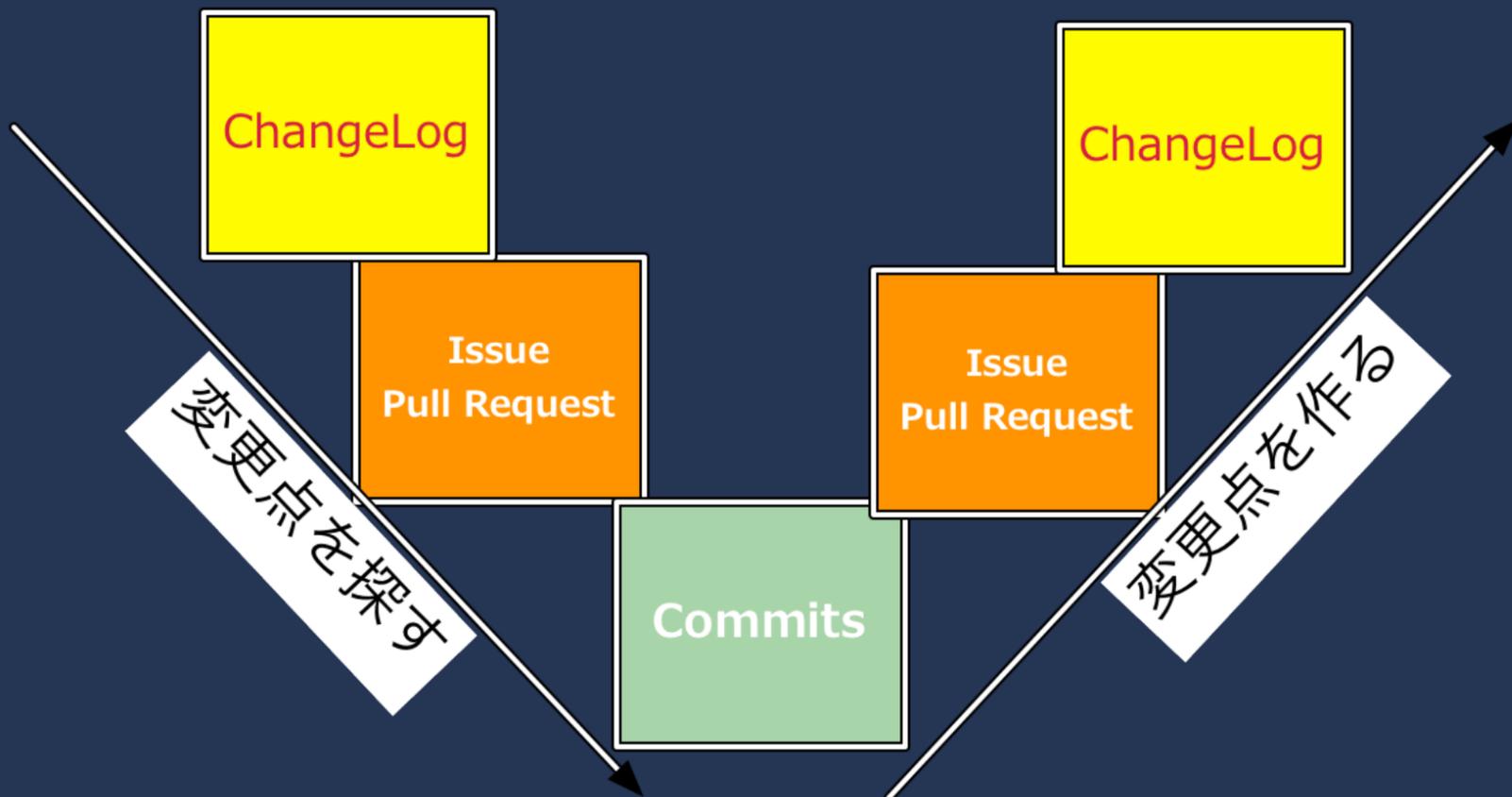
Issue
Pull Request

Issue
Pull Request

Commits

変更点を探す

変更点を作る



探すから作るへ

変更点の追い方≡変更点の作り方

- 変更点が追いやすい == いい変更点
- 変更点の追い方を知ってる ≡ いい変更点を作れるのでは？

追いやすい変更点って？

- 良いコミット
- 良いIssue/Pull Request
- 良いChangeLog

変更点を作る 🧑‍🔧

良いコミット

コミットメッセージを書く

- 今更感があるけど、要約と本文で書くのが基本
- [How to Write a Git Commit Message](#)
- [見えないチカラ: 【翻訳】Gitのコミットメッセージに関する
注意点](#)

良いコミットメッセージ?

- 追いやすい変更点からの良いコミットメッセージとは? 🌸
 - サマリと本文がある
 - 検索しやすいキーワードが存在する
 - 関連するIssue/Pull Requestが書かれてる
- 実現するための規約の一種として **conventional changelog**

conventional changelog

- ChangeLog自動生成のためのコミットメッセージ規約のこと
- [ajoslin/conventional-changelog](https://github.com/ajoslin/conventional-changelog)

```
feat(ngInclude): add template url parameter to events
```

```
The `src` (i.e. the url of the template to load) is now provided to the  
`$includeContentRequested`, `$includeContentLoaded` and `$includeContentError`  
events.
```

```
Closes #8453
```

```
Closes #8454
```

conventional changelog

- 規約の形は何でも良い = ツールとの相性
- コミット検索向けのキーワードとしてtypeとcomponentの利用

```
commit type      component      commit title
      \          |          /
feat(ngInclude): add template url parameter to events

body -> The 'src` (i.e. the url of the template to load) is now provided to the
`$includeContentRequested`, `$includeContentLoaded` and `$includeContentError`
events.

referenced -> Closes #8453
issues      Closes #8454
```

コミットメッセージへの追加情報

- コミットの種類 : feat?、fix?、Breaking Change?
- コミットの対象: (スコープ)
 - 多分コミット書くときに悩む
 - コミットメッセージを検索しやすくするためのキーワードとして考えるといいのかも

コミットメッセージへの追加情報

- 関連するIssueの場所
 - GitHubはIssueの#42的なIDをコミットメッセージに書くと連動する
 - IssueのURLでも問題ない
 - [Closing issues via commit messages - User Documentation](#)

conventional changelogのメリット

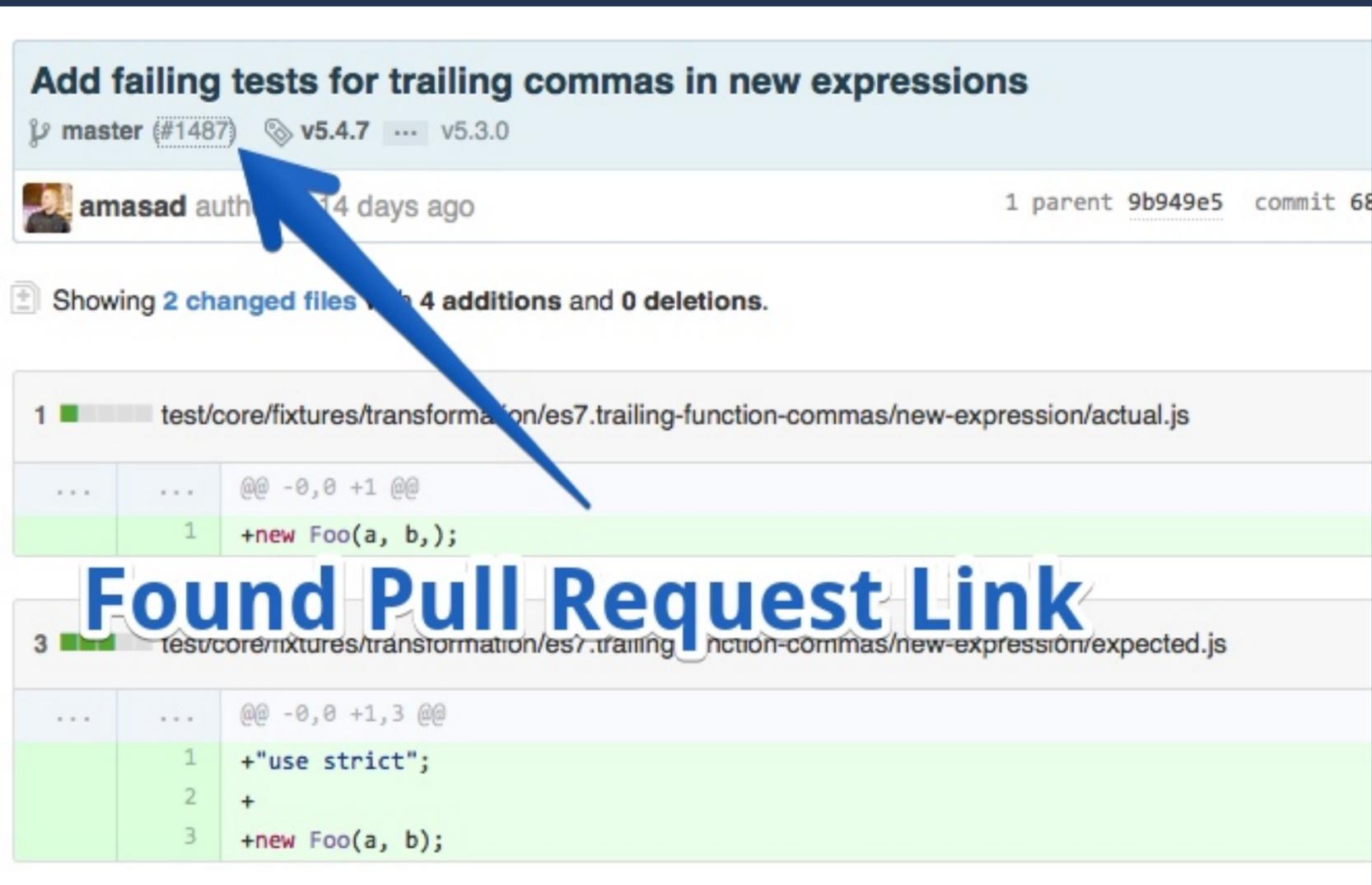
- コミットから自動的にChangeLogを生成出来る
 - ChangeLogに該当Issueへのリンクが生成できる事が大事
- [ajoslin/conventional-changelog](#)
 - Rust: [clog - A conventional changelog generator for the rest of us](#)
- [rvagg/changelog-maker](#)

コミットメッセージを書くの面倒問題

- 特に関連するIssueを書くのが大変(補完の問題)
- GitHubではPull Requestでマージされたコミットに自動的にリンクが付く
- つまり、常にPull Requestで開発すれば自動的にIssueと紐づく！

Pull Request 駆動

Pull Request 駆動



Add failing tests for trailing commas in new expressions

master (#1487) v5.4.7 v5.3.0

amasad authored 14 days ago 1 parent 9b949e5 commit 68

Showing 2 changed files with 4 additions and 0 deletions.

```
1 test/core/fixtures/transformations/es7.trailing-function-commas/new-expression/actual.js
... @@ -0,0 +1 @@
1 +new Foo(a, b,);
```

Found Pull Request Link

```
3 test/core/fixtures/transformations/es7.trailing-function-commas/new-expression/expected.js
... @@ -0,0 +1,3 @@
1 +"use strict";
2 +
3 +new Foo(a, b);
```

- Pull Requestで取り込まれたコミットは自動的にIssueと紐づく
- Linking merged pull requests from commits
- Pull Requestに詳細を書くことで、自動的に情報が集約される + コミットからのリンク

Pull Request駆動の副作用(感想)

- コミットが常にサーバにあるので、複数端末での開発がしやすい
- 開発時のメモ欄がある感覚
- Watchしてる人へ通知が行って、他の人が参加するチャンスがある

ChangeLog

ChangeLogは人が読むもの

- ChangeLogは人が読むものなので、無意味な自動生成は無意味
- 自動生成はできるが、自動生成のソースがダメだとChangeLogもダメ
- 良いコミットの副産物として良いChangeLogが作れる
- 良いコミットではない場合は、手動で書いたほうがマシ

ChangeLogにあると良いもの

- 日付
- バージョン
- 変更内容
 - グループ핑グされてるとより良い
- 関連するIssue/Pull Request
- Keep a Changelog

GitHub ReleaseとCHANGELOG

- GitHub Release
- CHANGELOGファイル
- どっちに書く？

GitHub Release

- Pros
 - 日付、バージョン、パーマネントリンク、RSSを得られる
 - git tagと連動できる
- Cons
 - メッセージの自動化がイマイチ難しい
 - Git tagとGitHub ReleasesとCHANGELOG.mdの自動化について

CHANGELOG

- Pros
 - 伝統的なやり方なのでツール等が多い
 - 自動化がしやすい
- Cons
 - パーマネントリンクが微妙になりやすい(Markdownじゃないとリンクができない、ハッシュ)

まとめ

- ChangeLogの追い方を知ること、いいChangeLogを書くことに繋がる
- GitHub上だけでも情報の詳細度が異なる
 - ChangeLog > Issue/Pull Request -> Commits
- この3つが繋がるような変更点をできるのが理想的

おわり

- This presentation is also Pull Request Driven
- [われわれは、いかにして変更点を追うか - Issueを追う技術 by azu · Pull Request #10 · azu/slide](#)