



愉快ㄉ貢獻開源

WeiTheShinobi
COSCUP 2024



這個 talk 會講什麼

因為去年聽到有人分享貢獻Linux, 所以有了這篇

- 紿你釣竿, 讓你入門有些方向
- 以 rust 的 linter - clippy 為範例
- 實際範例
- COSCUP 議程表上的標題可能寫得不夠精確, 但應該有表達到受眾是初心者的意思, 不希望有人被標題騙的感覺
投稿跟做簡報是兩件事

這個 talk 會講什麼

- 消除恐懼的最好辦法就是面對恐懼
- 人類的讚歌就是勇氣的讚歌
- 我對這個 talk 的設想是給想參與但覺得很難/不知道如何開始的人
希望聽完有種想馬上進行的感覺



關於我 WeiTheShinobi

- 即使引導早已破碎，也請您當上艾爾登之王

- 是一個忍者
- 之前工作在寫Go
- 興趣是爬山、跑步、寫程式
- 喜歡在網路上欣賞肥美ㄉ鳥類😍
- 特別喜歡鴨子！🦆🦆🦆



為什麼要做？

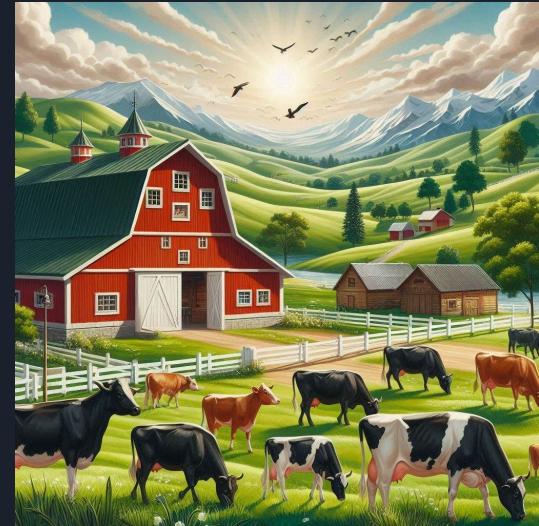


我受夠繁文縟節了。

為什麼要做？

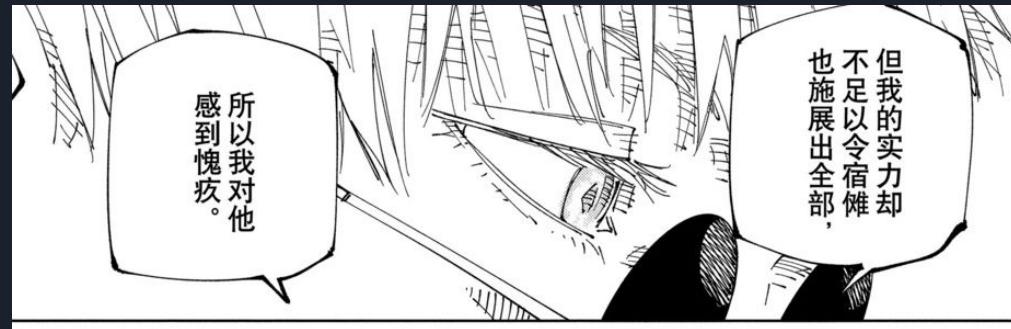
- 我受夠繁文縟節了
- 感覺工作上無法施展全部
- 公司的目的和開源專案不同，身邊有些人也覺得工作和想像中不一樣
- 想爽爽的寫程式碼(要大滿足)
- 開源專案對於程式碼的要求較高

- 牛牛牧場！這裡很多大



為什麼要做？

絕對的強者，由此而生的孤獨，教會你開源專案的是…





從有在使用、有興趣的專案開始

- 選有在使用的專案能讓你更加熟悉
(吃你自己的狗食！)
- 例如在專案中使用的框架，很多人都是用著用著就發pr了
- 來自 wiki:

Eating your own dog food（直譯為「吃你自家的狗糧」，亦略稱為dogfooding）是一句英語俚語，常用於描述公司（尤指軟體公司）使用自己生產的產品這一情況^[1]。

從有在使用、有興趣的專案開始

- 先了解專案在幹嘛(抽象)，再看看實際上是怎麼做到的(實現)
這個順序倒反的話，你會迷失在虛空之中。
- 自己動手，豐衣足食：從簡單的目標開始
 - > 以 rust compiler 為例，目前約有兩百二十萬行程式碼，一天看一千行，六年都看不完，看完了也不知道在幹嘛



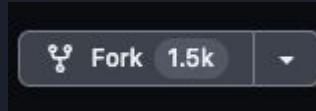
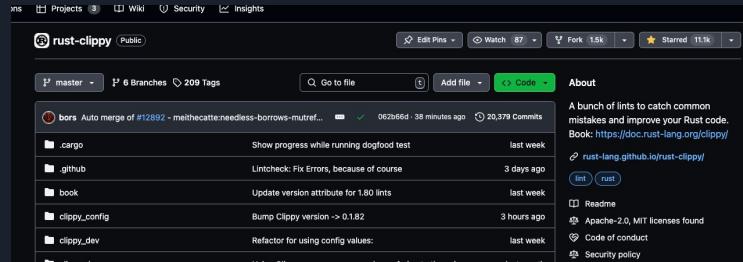
從有在使用、有興趣的專案開始

- 這個人認為只要看書就能學會游泳



Getting Start

1. 隨便選一個有興趣的專案
2. 按下 fork, 這會在你的帳號建立一個 repo
3. 使用你帳號底下的 repo 開發



Getting Start

1. 通常都有會一個 CONTRIBUTING.md, 狠狠地讀它
2. 看 issue 列表, 選一個你能做的(bug fix ...)
3. 和 maintainer 討論, 確定方向
4. 如果想要新功能, 先開 issue, 而不是直接寫一個大 pr



CODE_OF_CONDUCT.RU

CONTRIBUTING.md

COPYRIGHT

1998 Open ✓ 4.67 Closed

Author · Label · Projects · Milestones · Assigned · Sort ·

doc_markdown_footnote_identifier false positive (C-bug) (i-false-positive)
#3103 opened 40 minutes ago by jp-Elie

needless_pass_by_ref_mit should not trigger when values derived from the `list` are used in an unsafe block (C-bug) (i-base-positive) (suggestion-causes-bug)
#3117 opened 2 days ago by swasted

False almost_complete_letter_range when the endpoint is matched separately (C-bug) (i-false-positive)
#3171 opened 3 days ago by couver

needless_borrows_for_generic_arg fires incorrectly on `&mut` variable inside closure (C-bug) (i-false-positive)
#3170 opened 3 days ago by rancas

[Project Goal, Help Wanted] Optimizing MSRV attribute collection (medium) (performance-project)
#3196 opened 3 days ago by blynnas

needless_borrows_for_generic_args on large array (C-bug) (i-false-positive)
#3191 opened 4 days ago by fennott

Incorrect clippy::derivable_impls in the presence of conditional compilation (C-bug) (i-false-positive)
#3180 opened 3 days ago by Vorpahlade

MSRV-aware `std::instead_of_core` (enhancement) (i-false-positive)
#3156 opened 4 days ago by spread

needless_pass_by_value Wrongly emit on rc values (C-bug) (i-false-positive)
#3135 opened last week by dev-erik

Documentation for `from_mut_instead_of_collect` is incorrect (C-bug)
#3117 opened last week by opeenrag

Int on OR PATTERN if `alt()` (A-int)
#3142 opened last week by workingblue

如何選擇適合的 issue

- 看標籤
- 可以過濾某些標籤
- 把滑鼠移到標籤上會有詳細資訊
- CONTRIBUTING.md 可能也會寫



A screenshot of a GitHub issue list. The first issue is labeled 'easy' with a green circle icon. Its description reads: 'Call for participation: Easy difficulty. ...'. The second issue is labeled 'mentor' with a green circle icon. Its description reads: 'Call for participation: This issue has ...'. Below these issues, there is a snippet of text about 'T-AST' issues: 'T-AST issues will generally need you to match the AST; it is recommended to run `rustc -Z ui docs`. Usually the lint will end up to be a nested'. There are also snippets for 'E-medium' and 'T-middle' issues.

cannot run C-bug needs-triage O-windows-7 T-compiler
Category: This is a bug.

easy

E-easy
Call for participation: Easy difficulty. ...

mentor

E-mentor
Call for participation: This issue has ...

T-AST issues will generally need you to match the AST; it is recommended to run `rustc -Z ui docs`. Usually the lint will end up to be a nested

E-medium issues are generally pretty easy too, somewhat involved code wise, but not difficult debugging to find the actual problem behind th

T-middle issues can be more involved and rec of the most useful would be `expr_ty` (gives th useful).

淺談 Clippy

- 用一大堆 lint 檢查你的程式碼
- 編譯器會把 source code 轉換成目標程式碼
Rust 編譯器流程大概像是：

source code -> ast -> hir -> mir -> ...

Clippy 大概會接觸到這些

- 把「我是鴨子」看成「我 + 是 + 鴨子」主詞、動詞等等的集合
程式就能檢查：「我」後面一定要「是」
如此便能寫出以下程式碼：

```
if node      == "我" &&
    node.next() != "是":
        do_something()
```

About

A bunch of lints to catch common mistakes and improve your Rust code.
Book: <https://doc.rust-lang.org/clippy/>

🔗 rust-lang.github.io/rust-clippy/

lint rust

📄 Readme

⚠️ Apache-2.0, MIT licenses found

🤝 Code of conduct

⚠️ Security policy

↗️ Activity

☰ Custom properties

☆ 11.1k stars

👁️ 86 watching

🍴 1.5k forks

Report repository

淺談 Clippy

- 診斷訊息看起來像是這樣

```
warning: unnecessary use of `copied`
--> src/main.rs:11:22
|
11 |     for (t, path) in files.iter().copied() {
|          ^^^^^^
|
= help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#unnecessary\_to\_owned
= note: `#[warn(clippy::unnecessary_to_owned)]` on by default
help: use
|
11 |     for (t, path) in files {
```

淺談 Clippy

- talk is cheap, 程式碼邏輯大guy都這樣, 檢查特定的格式

```
if is_copy(cx, ty) {
    let parent_is_suffix_expr: bool = match cx.tcx.parent_hir_node(expr.hir_id) {
        Node::Expr(parent: &Expr) => match parent.kind {
            // &x is a nop, &x.clone() is not
            ExprKind::AddrOf(..) => return,
            // (*x).func() is useless, x.clone().func() can work in case func borrows self
            ExprKind::MethodCall(_, self_arg: &Expr, ..)
                if expr.hir_id == self_arg.hir_id && ty != cx.typeck_results().expr_ty_adjusted(expr) =>
            {
                return;
            },
            // ? is a Call, makes sure not to rec *x?, but rather (*x)?
            ExprKind::Call(hir_callee: &Expr, _) => matches!(
                hir_callee.kind,
                ExprKind::Path(QPath::LangItem(rustc_hir::LangItem::TryTraitBranch, ..))
            ),
            ExprKind::MethodCall(_, self_arg: &Expr, ..) if expr.hir_id == self_arg.hir_id => true,
            ExprKind::Match(_, _, MatchSource::TryDesugar(_) | MatchSource::AwaitDesugar)
            | ExprKind::Field(..)
            | ExprKind::Index(..) => true,
        }
    }
}
```

淺談 Clippy

- 如果最後發現程式有問題, 就拋出一個診斷訊息

```
span_lint_and_then(
    cx,
    lint: MANUAL_INSPECT,
    sp: name_span,
    msg: format!("using `{name}` over `{edit}`"),
    f: |diag: &mut Diag<()>| {
        diag.multipart_suggestion(msg: "try", suggestion: edits, applicability: app);
    },
);
```

程式實作

- 看看右邊範例
- 可以從錯誤訊息找到 lint
- 跟平常寫程式差不多，就是需要更多準備工作

Reproducer

I tried this code:

```
pub struct TupleStruct(Vec<usize>);

let value = TupleStruct {
    0: vec![0, 1, 2, 3],
};
```

I expected to see this happen (text edited, hopefully accurate):

```
warning: used a field initializer for a tuple struct
--> src/main.rs:5:13
5 | let value = TupleStruct { 0: vec![0, 1, 2, 3] };
   | ^~~~~~ help: try: `TupleStruct(vec![0, 1, 2, 3])`
   |
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#init_numbered_fields
   = note: `#[warn(clippy::init_numbered_fields)]` on by default
```

Instead, this happened:

```
warning: used a field initializer for a tuple struct
--> src/main.rs:5:13
5 | let value = TupleStruct {
   | ^~~~~~
6 | |     0: vec![0, 1, 2, 3],
7 | | };
   | | ^
   |
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#init_numbered_fields
   = note: `#[warn(clippy::init_numbered_fields)]` on by default
help: try
|
5 ~ let value = TupleStruct(<[_]>::into_vec(
6 +         // This rustc_box is not required, but it produces a dramatic improvement in compile
7 +         // time when constructing arrays with many elements.
8 +         #[rustc_box]
9 +         $crate::boxed::Box::new([${$x}, +])
10+     ));
   |
```

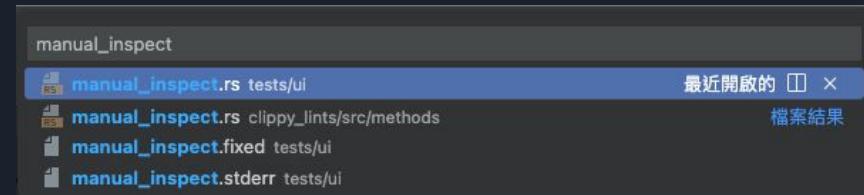
程式實作

1. 從錯誤訊息、有的沒的資訊可以找到線索
2. 找到錯誤是從哪裡開始的
3. 開始 debug, 目標是找到程式從正確轉移到錯誤狀態的那一刻

A -> B -> C -> X -> X -> X

找到 C

4. 撰寫測試



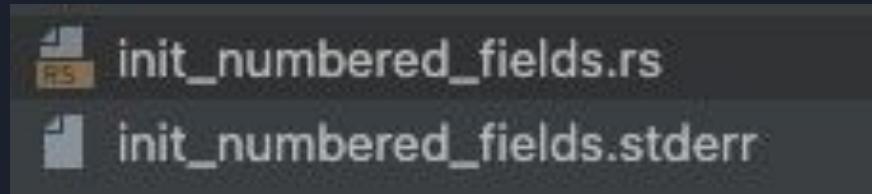
```
warning: using `map` over `inspect`
--> src/main.rs:9:36
9 |     let ref_mut_opt = opt.as_mut().map(|a| {
   |             ^~~~~
   |
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#manual_inspect
   = note: #[warn(clippy::manual_inspect)] on by default
help: try
|
9 ~     let ref_mut_opt = opt.as_mut().inspect(|a| {
10~         a.a += 123;
|
```

程式實作

- 測試三步驟：設置、運行、檢測
- 請把寫測試當作日常
- Clippy 的 ui test 怎麼寫呢？

會有一個 Rust code 的檔案會被 Clippy 掃過（測試案例寫這
生成測試結果在一個檔案中

你會在專案中看到像是右邊那樣的東西：



程式實作

- 回到剛剛的範例，可以直接把遇到的問題照抄當作 test case
- 恭喜你！你已經寫完測試了

Reproducer

I tried this code:

```
pub struct TupleStruct(Vec<usize>);

let value = TupleStruct {
    0: vec![0, 1, 2, 3],
};
```

```
+     struct TupleStructVec(Vec<usize>);
+
+     let _ = TupleStructVec { 0: vec![0, 1, 2, 3] };
```

程式實作

- 回到剛剛的範例，可以直接把遇到的問題照抄當作 test case
- 恭喜你！你已經寫完測試了
- 有時候要可能到 FP, FN, 需要更多測試

Reproducer

I tried this code:

```
pub struct TupleStruct(Vec<usize>);

let value = TupleStruct {
    0: vec![0, 1, 2, 3],
};
```

» Common Abbreviations

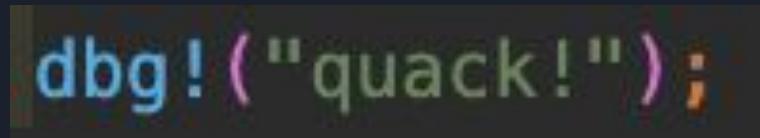
Abbreviation	Meaning
UB	Undefined Behavior
FP	False Positive
FN	False Negative
ICE	Internal Compiler Error
AST	Abstract Syntax Tree
MIR	Mid-Level Intermediate Representation
HIR	High-Level Intermediate Representation
TCX	Type context

```
+ struct TupleStructVec(Vec<usize>);
+
+ let _ = TupleStructVec { 0: vec![0, 1, 2, 3] };
```

This is a concise list of abbreviations that can come up during Clippy development. An extensive general list can be found in the [rustc-dev-guide glossary](#). Always feel free to ask if an abbreviation or meaning is unclear to you.

程式實作

- 是時候熟練的使用 terminal 了！



輸入：

cargo clippy uitest

-> 測試

cargo clippy uibless

-> 生成測試

TESTNAME=\$name cargo clippy uibless -> 限定測試名

蠻重要的，不然每次全測試一次會跑很久

- `dbg!()` 會是你的好朋友，他能：
talk is cheap



```
[clippy_lints/src/init_numbered_fields.rs:46:9] e = Expr {
    hir_id: HirId(DefId(0:19 ~ init_numbered_fields[1a70]::{impl#0}::default).23),
    kind: Block(
        Block {
            stmts: [],
            expr: Some(
```

程式實作

- 開始測試驅動開發吧！
配合測試指令，你開始研究 ... 時光飛逝 ...



程式實作

- 開始測試驅動開發吧！
配合測試指令，你開始研究... 時光飛逝 ...
- 最終你改了一行 code，成功解決了這個問題



```
62             snippet_with_applicability(cx, path.span(), "...", &mut appl),
63             expr_spans
64             .into_iter_sorted()
65 +             .map(|(_, span)| snippet_with_context(cx, span, path.span().ctxt(), "...", &mut appl).0)
66             .intersperse(Cow::Borrowed(", "))
67             .collect::<String>()
68     );
```

- 測試正確的生成

```
26 + error: used a field initializer for a tuple struct
27 +   --> tests/ui/numbered_fields.rs:49:13
28 +   |
29 + LL |     let _ = TupleStructVec { 0: vec![0, 1, 2, 3] };
30 +   |           ~~~~~ help: try: `TupleStructVec(vec![0, 1, 2, 3])`
31 +
32 + error: aborting due to 3 previous errors
```

程式實作

- 什麼？就改一行
- 台下的你：



程式實作

- 什麼？就改一行
- 台下的你：



程式實作

- 什麼？就改一行
- 台下的你：



程式實作

- Clippy 依賴了很多 Compiler 的模組
巨大複雜的專案感覺就像一個特定領域語言一樣
(我自己的感覺啦)
- 承上
需要對編譯器有些了解
但不急, Clippy 只用到一小部分
- 多多查閱文件, 文件有提供一些常用手法 : Clippy Book

6. Development

- 6.1. Basics
- 6.2. Adding Lints
- 6.3. Defining Lints
- 6.4. Writing tests
- 6.5. Lint Passes
- 6.6. Emitting lints
- 6.7. Type Checking
- 6.8. Trait Checking
- 6.9. Method Checking
- 6.10. Macro Expansions
- 6.11. Common Tools
- 6.12. Infrastructure
 - 6.12.1. Syncing changes between Clippy and rust-lang/rust
 - 6.12.2. Backporting Changes
 - 6.12.3. Updating the Changelog
 - 6.12.4. Release a New Version
 - 6.12.5. The Clippy Book
- 6.13. Proposals
 - 6.13.1. Roadmap 2021
 - 6.13.2. Syntax Tree Patterns
- 6.14. The Team

程式實作

- 給大家看看，過過乾癮

Dealing with macros and expansions

Keep in mind that macros are already expanded and desugaring is already applied to the code representation that you are working with in Clippy. This unfortunately causes a lot of false positives because macro expansions are "invisible" unless you actively check for them. Generally speaking, code with macro expansions should just be ignored by Clippy because that code can be dynamic in ways that are difficult or impossible to see. Use the following functions to deal with macros:

- `span.from_expansion()` : detects if a span is from macro expansion or desugaring. Checking this is a common first step in a lint.

```
if expr.span.from_expansion() {  
    // just forget it  
    return;  
}
```

Checking for a specific type

There are three ways to check if an expression type is a specific type we want to check for. All of these methods only check for the base type, generic arguments have to be checked separately.

```
use clippy_utils::ty::{is_type_diagnostic_item, is_type_lang_item};  
use clippy_utils::{paths, match_def_path};  
use rustc_span::symbol::sym;  
use rustc_hir::LangItem;  
  
impl LateLintPass<'_> for MyStructLint {  
    fn check_expr(&mut self, cx: &LateContext<'_>, expr: &Expr<'_>) {  
        // Getting the expression type  
        let ty = cx.typeck_results().expr_ty(expr);  
  
        // 1. Using diagnostic items  
        // The last argument is the diagnostic item to check for  
        if is_type_diagnostic_item(cx, ty, sym::Option) {  
            // The type is an `Option`  
        }  
  
        // 2. Using lang items  
        if is_type_lang_item(cx, ty, LangItem::RangeFull) {  
            // The type is a full range like `.`.drain(..)`  
        }  
  
        // 3. Using the type path  
        // This method should be avoided if possible  
        if match_def_path(cx, def_id, &paths::RESULT) {  
            // The type is a `core::result::Result`  
        }  
    }  
}
```

Prefer using diagnostic items and lang items where possible.

程式實作

- Rust Compiler Dev Guide 也能提供很多幫助
- Clippy 大部分都在 AST, HIR 的部分
- 不會很難, 對編譯器有基礎理解即可
不需要太多深入知識

Source Code Representation

[33. Prologue](#)

[34. Command-line arguments](#)

[35. rustc_driver and rustc_interface](#) ➔

[36. Syntax and the AST](#) ➔

[37. The HIR \(High-level IR\)](#) ➔

可以參考的資料

- 文件、CONTRIBUTE.md
- Clippy book、rust compiler dev guide
- 以前的討論
有時候你會看到: see issue1234
-> github.com/{org}/{project}/issues/1234
- 查看以前的 commit(有一些工具更方便看)

The screenshot shows a GitHub repository page for 'rust-clippy'. The 'CONTRIBUTING.md' file is open, displaying the following content:

```
rust-clippy / CONTRIBUTING.md ⓘ  
Alexendoo Destructure Conf in register_lints ✓  
Preview Code Blame 280 lines (215 loc) · 12.9 KB  
  
Contributing to Clippy  
Hello fellow Rustacean! Great to see your interest in Clippy.  
First: if you're unsure or afraid of anything, just ask!  
  
The Clippy book  
If you're new to Clippy and don't know where to start, the Clippy book includes:  
High level approach  
Rust Compiler Development Guide  
Getting Started  
Thank you for your interest in contributing to Rust! There are many ways to contribute, and we appreciate all of them.

- Asking Questions

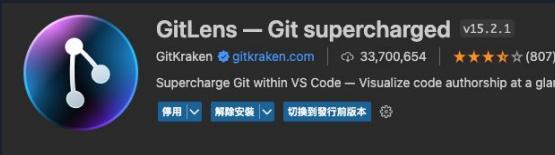
```

The screenshot shows a GitHub commit message from the 'rust-clippy' repository. The message contains a warning about inlining code:

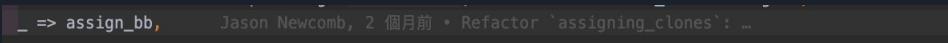
```
/// The macro statement may be lowered due to the compiler  
/// not inlining the call to cmp. See issue [#5354] (https://github.com/rust-lang/rust/pull/5354)  
///
```

活用工具

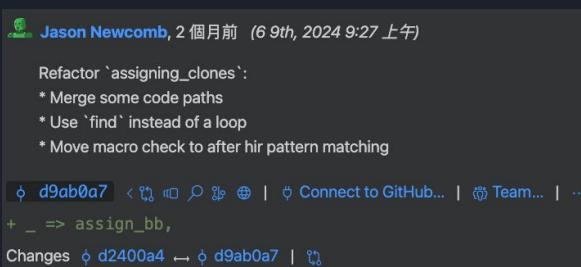
- 看看這份 code 之前的歷史



- 安裝後就會顯示資訊



- 滑鼠移過去可以看更多



Express.js 的災難

- 我從 Reddit 上的討論串猜測
- youtube 的一個教學影片以 Express.js 為範例，示範如何發出 pull request
- 大批網友前往 Express.js 的專案濫發 pr
- 至今的 Express.js 依然三天兩頭就有人來鬧場
- 舉例：只是在 README.md 加上無意義的字

72 Open	✓ 1,358 Closed
Update Readme.md	
#5780 by Deep7704 was closed 3 days ago • Review required	
Update Readme.md	
#5779 by Babu-Bhai-No-1 was closed 3 days ago • Review required	
Update by adarsh	
#5778 by ADARSH0104 was closed 3 days ago • Review required	
Update Readme.md	
#5777 by Abhinav7585 was closed 4 days ago • Review required	
Update Readme.md	
#5745 by upendraPal was closed 3 weeks ago • Review required	
Update Readme.md	
#5744 by shubhamft2004 was closed 3 weeks ago • Review required	
Update index.js	
#5742 by harsh7913 was closed 3 weeks ago • Review required	
SPAM	
#5741 by AMIT-NAIKK was closed 3 weeks ago • Review required	
Update Readme.md	
#5740 by Ishikarao30 was closed 3 weeks ago • Review required	
updated package to version 4.3.5	
#5738 by ritik772 was closed 3 weeks ago • Review required	
Update Readme.md (invalid pr)	
#5737 by hub02 was closed last month • Review required	
Update Readme.md for heading	
#5736 by harshitsheesh was closed 3 weeks ago • Review required	
Update Readme.md	
#5735 by Charlie15082020 was closed last month • Review required	
Update Readme.md	
#5733 by Kartikaydon was closed last month • Review required	
Update Readme.md	
#5732 by Sankha-Dalai was closed last month • Review required	
Replace node:querystring with URLSearchParams	
#5730 by TheDevMinerTV was closed last month	
Update eslintrigore	
#5718 by Sana-Shah-een was closed on Jun 23 • Review required	
awwwwUpdate Readme.md (invalid pr)	
#5717 by shahilahamad was closed on Jun 23 • Review required	
Update Readme.md (invalid)	
#5716 by Dhruvsingh9999 was closed on Jun 23 • Review required	

Express.js 的災難

Update Readme.md #5802

Closed Sunder-Kumar wants to merge 1 commit into expressjs:master from Sunder-Kumar:master

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾

Readme.md

```
@@ -35,7 +35,7 @@ app.get('/', function (req, res) {  
35   app.listen(3000)  
36   ...  
38 -  ## Installation  
39 + # Sunder-Kumar  
40  ## Installation  
41  This is a [Node.js] (ht
```

Update Readme.md #5799

Closed hiteshpanditpatil wants to merge 1 commit into expressjs:master from hiteshpanditpatil:master

Conversation 2 Commits 1 Checks 0 Files changed 1

Changes from all commits ▾ File filter ▾ Conversations ▾ Jump to ▾

Readme.md

```
@@ -1,6 +1,7 @@  
1  ![[Express Logo]](https://i.cloudup.com/zfY6LL7eFa-3000x3000.png)](http://expressjs.com/)  
2  **Fast, unopinionated, minimalist web framework for [Node.js](http://nodejs.org).**  
3  **This project has a [Code of Conduct]().**  
4  **Fast, unopinionated, minimalist web fr  
5  **This project has a [Code of Conduct]()  
6  + ** dataease directory  
7  + ** dataease directory
```

Express.js 的災難

 SPAM

#5741 by AMIT-NAIKK was closed 3 weeks ago • Review required

Express.js 的災難

某個網友 : *我知道這是spam 但我還是要發

 SPAM

#5741 by AMIT-NAIKK was closed 3 weeks ago • Review required

Express.js 的災難

某個網友 : *我知道這是spam但我還是要發

```
SPAM
 35  MIT-NAIKK was closed 3 weeks ago
 36 + # baap edit done
 37 + # vishal bhardwaj
 38 ## Installation
```

Express.js 的災難

某個細

還是要發

```
18 * [Triagers] (#triagers)
19 * [License] (#license)
20 + # mj ...
21 ! [NPM Version] [npm-version-image]] [npm-url]
22 ! [NPM Install Size] [npm-install-size-image]] [npm-install-size-url]
```

SPAM

atulkrsdevops commented 3 weeks ago

Hello There

Update Readme.md
Third Party License
+ # Rishi

```
6
7 ## Table of contents
8 + Hello Hello
```

```
5 **This project has a [Code of Conduct] []
6 + **Author Affan Mansuri **
7 ## Table of contents
```

SPAM #5771

Closed

Gaurav18982 wants to merge 1 commit into branch

Conversation

Commits 1

Gaurav18982 commented last week

tp

```
app.listen(3000)
...
+ # Raj Sir
## Installation
```

```
6
7 ## Table of contents
8 + #vishal bhardwaj
9 ## Installation(#Insta
```

我ㄉ心得

- 就像是超級馬拉松
一個人在無人區, 不知道自己能某抵達終點
(不知道大家有沒有長跑或登山過)
- 要嘗試去面對未知
做只知道終點不知道路徑的事情
- 心裡還是有點壓力在, 不希望浪費 maintainer 的時間
既期待又怕受傷害



我ㄉ心得

- 花了很多時間什麼也沒做是正常的
有些問題本來就是死胡同
- 99% 的時間都花在理解問題
能理解為什麼有大佬常常說自己整天沒做事
花了數十個小時只為了改一個小 bug
- 承上
對於一個生命週期長且規模巨大的專案
程式碼的要求較高



POV: 在 COSCUP 學到如何貢獻



* 一想到自己知道了如何貢獻，使你充滿了決心。

roadmap && END

1. 會寫 Rust
 2. 了解資料結構和演算法
 3. 懂編譯器在幹嘛
 4. 仔細閱讀 CONTRIBUTING.md
 5. fork 專案
 6. 挑簡單的 issue 開始動手 / 發 issue 提 feature
(label: good-first-issue)
 7. 撰寫程式碼和測試
 8. 通過測試、format
 9. 提交
 10. 爽
-
- 還有很多類似的專案可以貢獻
像是 rustfmt, rust compiler
 - 某個矮人曾說過：「努力過的人都是戰士」

