RUSTFIX

…and the journey to getting better code automatically

Pascal Hertleif

2018-02-04

HI, I'M PASCAL HERTLEIF

- Web dev & Rust freelancer
- Co-organizer of Rust Cologne
- {twitter,github}.com/killercup
- Rust-centric blog: deterministic.space

RUST PROGRAMMERS LIKE COMPILER ERRORS

```
error[E0507]: cannot move out of borrowed content
   --> src/lib.rs:126:29
    |
    |
126 |         if let Some(code) = diagnostic.code {
    |                     ----    ^^^^^^^^^^ cannot move out of b
    |                     |
    |                     hint: to prevent move, use `ref code` o
```

WHY?

Rust is all about compile-time checks

But also: Rust's errors are *pretty* and *helpful*

WARNINGS, TOO

- Lints
- *Prevent* errors
- Examples
  - unused `Result`s or `Iterator`s
  - Missing docs

CLIPPY

It looks like you're enjoying errors. Would you like some more?

- 208 lints written by the community
- Find common bugs
  - For example: `if_same_then_else`

LINTS AS A TEACHING TOOL

```
iter.filter(|x| x == 0).next()
```

Did you know about `iter.find(|x| x == 0)`?

```
if !m.contains_key(&k) { m.insert(k, v) }
```

Have you heard the good news about the Entry API?

```
m.entry(k).or_insert(v);
```

- Learn new APIs when you are *almost* using them
- Each clippy lint is documented with examples

MAKING THIS INTERACTIVE

SUGGESTIONS

```
error: this range is empty so this for loop will never run
   --> $DIR/for_loop.rs:148:14
    |
    |
148 |     for i in 10..0 {
    |              ^^^^^
    |
    = note: `-D reverse-range-loop` implied by `-D warnings`
help: consider using the following if you are attempting to itera
    |
148 |     for i in (0..10).rev() {
    |              ^^^^^^^^^^^^^
```

WHY FIX IT YOURSELF?

I’m sold. These suggestions are great. But…

- don’t make me type
- don’t make me wait
- just fix my code already

PRESENTING: RUSTFIX

IT'S ALREADY THERE

No magic! Just:

- Take existing errors
- Parse the compiler output (JSON)
- Apply suggestions (search & replace)

MAKE IT INTERACTIVE (1)

rustfix as CLI tool is interactive

- for each error (that has a suggestion) it
  - shows you the error and the fix(es)
  - asks what you want to do
  - fixes your code

MAKE IT INTERACTIVE (2)

You need to manually call the CLI tool

You do that after working on the code

We can do better

FIXES VIA IDE

RLS

- Rust Language Server exposes warnings and suggestions
- VSCode plugin can already apply fixes

DEEP DIVE!

HOW DOES THIS WORK?

- Compiler plugin
- Find nodes and items of interest
- ‘Annotate’ spans of code
- Compiler generates human/JSON output

WHAT IS AUTO-FIXABLE?

- A lot!
- Do we need to make a list?
- Rust PR #47540 adds flag for “approximate suggestions”
- rustfix-like tools will use this in the future

THANKS!

YOU CAN CONTRIBUTE LINTS AND SUGGESTIONS!

Search for “Rust Clippy” or go to

https://github.com/rust-lang-nursery/rust-clippy

There are a lot of issues labelled *good first issue*

You get to call yourself a compiler hacker!

ANY QUESTIONS?

Visit rust-lang.org

Follow me on Twitter: @killercup

Slides available at git.io/fosdem-rustfix