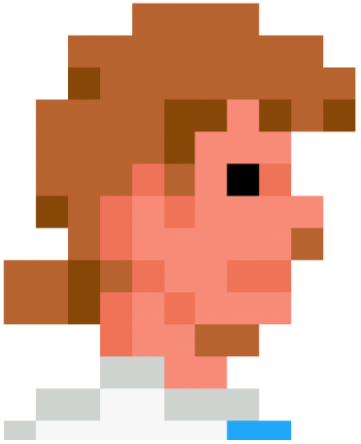
Rust Under the Rug



Software Engineer at Tuenti

MadRust co-organizer

jrvidal

_rvidal

[rust-dev] RFC: Updated RFC process

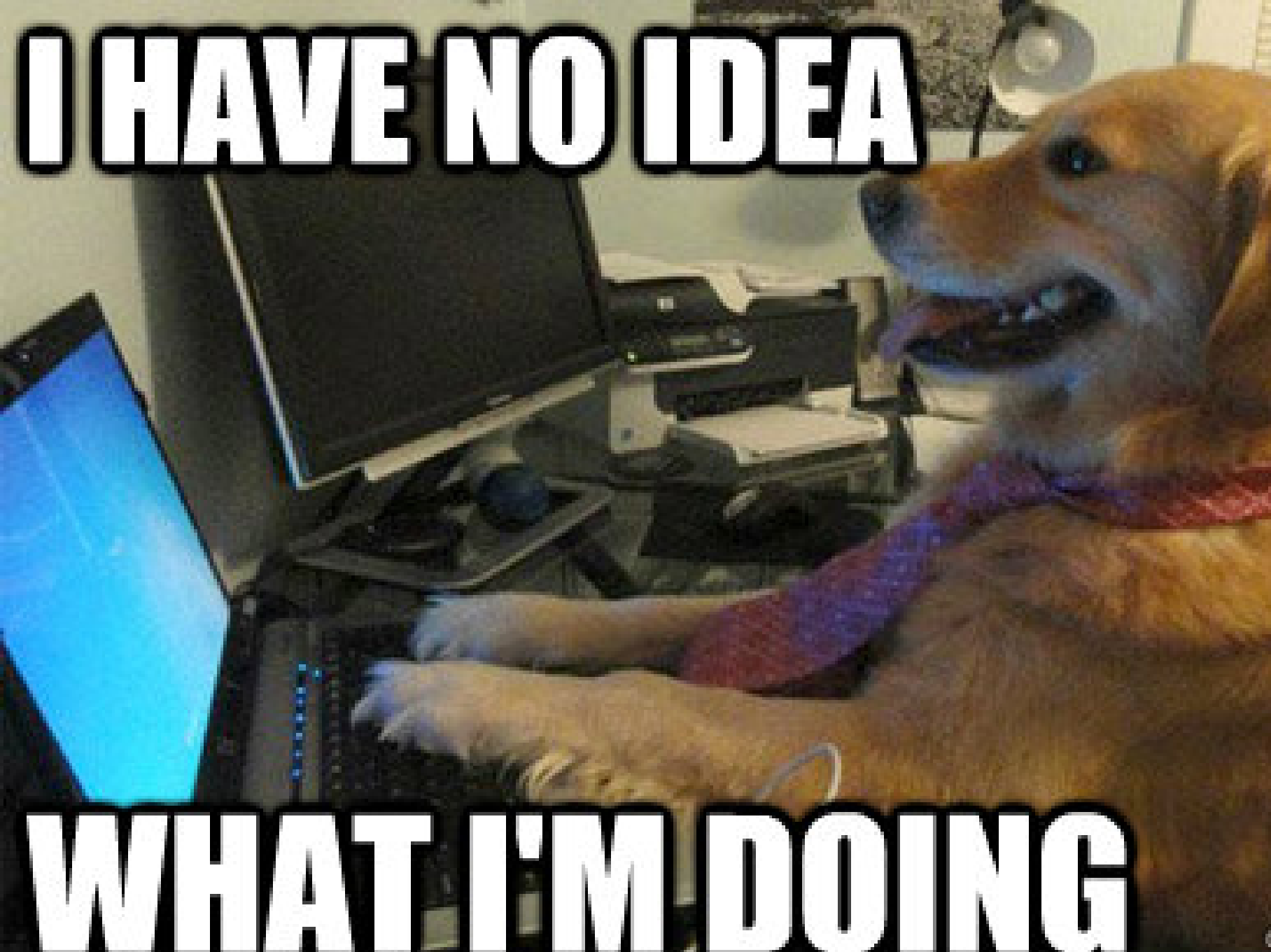
Brian Anderson banderson at mozilla.com

Tue Mar 11 18:11:48 PDT 2014

Hey, Rusties.

The freewheeling way that we add new features to Rust has been good for early development [...]

@brson


I HAVE NO IDEA
WHAT I'M DOING

What "imperfections"?

What curiosities

- Leakpocalypse
- Auto traits
- Implicit bounds

Leakpocalypse

```
#[lang = "drop"]
trait Drop {
    fn drop(&mut self);
}
```

```rust
// libcore/mem.rs

unsafe fn forget<T>(t: T) {
    intrinsics::forget(t)
}
```

```
struct Leak<T> {
    cycle: RefCell<Option<Rc<Rc<Leak<T>>>>>,
    data: T,
}

fn safe_forget<T>(data: T) {
    let e = Rc::new(Leak {
        cycle: RefCell::new(None),
        data: data,
    });
    // Create a cycle
    *e.cycle.borrow_mut() = Some(
      Rc::new(e.clone())
    );
}
```

```rust
let mut v = Box::new(10);

{
  let reference = &*v;
  let jg = thread::scoped(move || {
    println!("{}", reference);
  });
  safe_forget(jg);
}

*v = 20;
```

```
// libcore/mem.rs

fn forget<T>(t: T) {
    unsafe { intrinsics::forget(t) }
}
```

Auto Traits

```
unsafe auto trait Send {}

#[lang="sync"]
unsafe auto trait Sync {}
```

```rust
// Automatic (non-)implementations
struct Sendable { numbers: Vec<usize> }

struct NonSendable { shared: Rc<usize> }



// Overrides
struct YouCanSendThis { numbers: *const usize }
unsafe impl Send for YouCanSendThis {}

struct DontSendThis { numbers: Vec<usize> }
impl !Send for DontSendThis {} // Unstable
```

RFCs #19, #127

```
pub struct Banana { /* normal stuff */ }
```

```rust
pub struct Banana { /* normal(?) stuff */ }

unsafe impl Send for Banana {}
```

```rust
pub struct Banana { /* normal(?) stuff */ }

trait SendForReal : Send {}

impl SendForReal for Banana {}
```

```
let m = Mutex::new(Cell::new(0));
let g : MutexGuard<Cell<i32>> =
    m.lock().unwrap();

{
    rayon::join(
        || { g.set(g.get() + 1) },
        || { g.set(g.get() + 1) });
}
```

?Sized


CANON ZOOM LENS EF 24-105mm
EFS 17-85mm
MACRO 0.35m/1.2ft
IMAGE STABILIZER
Nikolai Chernichenko

```
#[lang = "sized"]
trait Sized {}
```

```
enum Option<T> { /* ... */ }

struct Rc<T>
where
    T: ?Sized { /* ... */ }
```

```
fn my_api<T>()
  where T : ?Sized +
            ?Move +
            ?Leak +
            ?DynSize
```

More implicit bounds RFC

```rust
unsafe auto trait Unpin {}
```

Unpin RFC

More!

- `#[fundamental]`
- Specialization
- `DerefMove` and `*boxed`
- `Read::chars`

Bibliography

- What's a lang item
- Nomicon
- `catch_panic` RFC, on different types of guarantees.


Keep Calm and Read RFCs
Thanks!