How do I give a TcpStream to a BufReader but then take it back?

Asked 5 years, 4 months ago

Active 5 years, 4 months ago

Viewed 207 times

1

I have a struct:

```
pub struct Paradise {
    cstream: TcpStream,
}
```

with a method:

```
pub fn write_message(&mut self, code: i32, message: &str) {
    let foo = format!("{} {}\r\n", code, message);
    let _ = self.cstream.write(foo.as_bytes());
}
```

That works great. It's an FTP server so when I get a new `TcpStream` from `TcpListener::bind` I do:

```
let mut p = Paradise::new(stream);
p.start();
```

And inside that `start` method I call:

```
self.write_message(220, "Welcome to Paradise");
```

and sure enough, I see that message in the FTP client. So far so good.

But then I do:

```
let mut br = BufReader::new(&self.cstream);
loop {
    let mut buffer = String::new();
    let _ = br.read_line(&mut buffer);
    println!("{:?}", buffer);
    self.write_message(550, "Testing");
}
```

And when I get to the next `write_message` call inside the loop:

> cannot borrow `*self` as mutable because `self.cstream` is also borrowed as immutable [E0502]

Full code:

https://github.com/andrewarrow/tinted_paradise/blob/169cc5f7025c417814f47a1fb3e3fc78ce4f9516/src/paradise.rs

https://github.com/andrewarrow/tinted_paradise/blob/169cc5f7025c417814f47a1fb3e3fc78ce4f9516/src/starter.rs

How can I change stuff around so I *can* call `write_message` inside the loop?

memory-management reference rust borrowing

Share
Improve this question
Follow

edited Aug 21 '16 at 19:15

Shepmaster
305k ● 59 ● 824 ● 1083

asked Aug 21 '16 at 18:59

Andrew Arrow
4,117 ● 8 ● 41 ● 75

1

There are currently 72 questions for the error message you've specified. – Shepmaster Aug 21 '16 at 19:22

"are you saying I'm using stack overflow incorrectly to learn rust? Why not use it as a learning..."

– Andrew Arrow Aug 21 '16 at 19:27

1

Haha, I'm certainly not against *learning*. I *am* (obliquely) pointing out that multiple people have spent significant time explaining that error message to ~70 people in ~70 contexts. It would be courteous to read (a good portion of) them in addition to the normal Rust documentation and explain what makes your question different. Note that I answered the question you asked in the title, which seemed unique, but perhaps is actually a duplicate of stackoverflow.com/q/35869078/155423 or those linked from there.
– Shepmaster
Aug 21 '16 at 19:32

3

Also, I'm pretty sure that "Why are you so against learning?" is some kind of logical fallacy or similar... ^_^
– Shepmaster
Aug 21 '16 at 19:35

2

@AndrewArrow: The purpose of stackoverflow is to get an answer to your question *without* having to ask it. It also aims at *consolidating* knowledge, rather than having it scattered across a dozen half-hearted explanations among so many pages. This means that, in general, if you ask a question that has already been asked, it should be closed. If the answers on the already asked question are insufficient, then it is best to improve *those* rather than leave incomplete answers there. As a corollary, a different question need to clearly indicate *how* it is different; it may not be obvious.
– Matthieu M.
Aug 21 '16 at 20:22

Show 3 more comments

1 Answer

Active Oldest Votes

4

> How do I give a TcpStream to a BufReader but then take it back?

By calling BufReader::into_inner :

```
impl<R: Read> BufReader<R> {
    fn into_inner(self) -> R
}
```

> Unwraps this BufReader , returning the underlying reader.
>
> Note that any leftover data in the internal buffer is lost.

Share
Improve this answer
Follow

answered Aug 21 '16 at 19:15

Shepmaster
305k ● 59 ● 824 ● 1083

Add a comment

Your Answer

Post Your Answer

Not the answer you're looking for? Browse other questions tagged memory-management reference rust borrowing or ask your own question.

The Overflow Blog

- Sequencing your DNA with a USB dongle and open source code
- Don't push that button: Exploring the software that flies SpaceX rockets and...

Featured on Meta

- Providing a JavaScript API for userscripts
- Congratulations to the 59 sites that just left Beta

Linked

If BufReader takes ownership of a stream, how can I read and write lines on it?

Related

How can I give eclipse more memory than 512M?

BufReader::lines() over TcpStream stops iteration

Will OpenJDK JVM ever give heap memory back to Linux?

How to access the BufReader twice?

Can std::io::BufReader on a TcpStream lead to data loss?

How to store TcpStream with BufReader and BufWriter in a data structure

How to store TcpStream inside a HashMap?

Hot Network Questions

EU ETS: if within-EU flight emissions are limited to climate goals, is there still a reason not to fly as long as you can afford it?

Are they already planning a successor to the JWST?

Could mass timber construction techniques have been used in the past?

On what basis do countries repay international loans?

Changing a color of a link

more hot questions

Question feed