



Products

Return a closure returning the function argument in Rust

Log in or sign up

Ask Question

Asked 1 year, 2 months ago

Active 1 year, 2 months ago

Viewed 176 times



1



I'm trying to write a function, which will take an argument of type `T` and return a closure, which, when called, will return that argument. How do I write that?

```
pub struct ParseError {}

pub type Parser<T> = Box<dyn Fn(&str) -> Result<(&str, T), ParseError>>;

pub fn pure<T>: 'static>(value: T) -> Parser<T> {
    Box::new(move |input: &str| -> Result<(&str, T), ParseError> { Ok((input, value)) })
}
```

The error:

```
Checking parsec v0.1.0 (/home/arjaz/Documents/code/rust/parsec)
error[E0507]: cannot move out of `value`, a captured variable in an `Fn` closure
  --> src/parsec.rs:16:79
   |
12 | pub fn pure<T>: 'static>(value: T) -> Parser<T>
   |                                ----- captured outer variable
   ...
16 |     Box::new(move |input: &str| -> Result<(&str, T), ParseError> { Ok((input, value)) })
   |                                ^^^^^^ move occurs because `value` has type `T`, which does not implement the `Copy` trait
```

error: aborting due to previous error

For more information about this error, try `rustc --explain E0507`.
error: could not compile `parsec`.

rust lifetime borrow-checker

Share

Improve this question

Follow

edited Oct 8 '20 at 18:00



trent kennedy

20.1k ● 7 ● 42 ● 72

asked Oct 8 '20 at 6:58



Arjaz

11 ● 1

2

It is not working because you are returning a closure that is callable for multiple times, if you change the definition as `FnOnce` it will work but you will lose the ability of calling this closure for multiple times, please check this post for details: stackoverflow.com/a/62563511/1601571

– Ömer Erden

Oct 8 '20 at 7:37

Ok, I'll require `T` to implement `Copy` then. Thank you for clarification!

– Arjaz

Oct 8 '20 at 7:51

Add a comment

1 Answer

Active Oldest Votes



3



Here is a [simplified version in the Rust playground](#), one that avoids `Box` and `Parser`. The function consumes `value`, and returns a closure that, when called, simply returns `value`. Note that the closure is of type `FnOnce` (or, more precisely, is of some type based on `FnOnce`), so we can only call it once. After the call, the memory associated with `value` belongs to the caller.


Your privacy instead we use `Fn`, then the closure can be called multiple times. But after the first call, the closure no longer owns `value` -- it was given to the first caller as the return value -- so calling the closure a second time is not going to work. Hence the error. If a closure is going to be called multiple times, then it must return a copy or clone of `value`, retaining ownership of the original for itself. Since the function is generic over the type of `value`, `T`, you would need to constrain `T` to be `Copy` or `Clone`.

Accept all cookies Customize settings

Share

Improve this answer

Follow
edited Oct 8 '20 at 16:45

answered Oct 8 '20 at 9:02

[AmigoNico](#)
6,232 ● 1 ● 31 ● 44
[Add a comment](#)



Your Answer

Post Your Answer



By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [rust](#) [lifetime](#) [borrow-checker](#) 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

[Error on Future generator closure: Captured variable cannot escape 'FnMut' closure body](#)

Related

- [What is the lifetime of a static variable in a C++ function?](#)
- [Calling an on-heap function argument inside of a closure](#)
- [336 How do I print the type of a variable in Rust?](#)
- [How do I capture variables outside the scope of a closure in Rust?](#)
- [2 How to define a function with two enum parameters where the 2nd type is determined by the 1st?](#)
- [Return lazy iterator that depends on data allocated within the function](#)
- [How to get Result's Ok value if the Result is in a Vector?](#)
- [Declaring Associated Type of Trait Object in Async Function Parameter](#)

Hot Network Questions

-  Repeating slices of an array incrementally
-  Is the science in "Don't Look Up" realistic?
-  Is there any other notation, like tab notation, for piano?
-  Text overflow in tabularx
-  Schrödinger's cat program
- [more hot questions](#)
-  Question feed

[Jobs](#)
[Developer Jobs Directory](#)
[Salary Calculator](#)
[Help](#)
[Mobile](#)

PRODUCTS

[Teams](#)
[Talent](#)
[Advertising](#)
[Enterprise](#)

COMPANY

[About](#)
[Press](#)
[Work Here](#)
[Legal](#)
[Privacy Policy](#)
[Terms of Service](#)
[Contact Us](#)
[Cookie Settings](#)
[Cookie Policy](#)

STACK EXCHANGE NETWORK

[Technology](#)
[Culture & recreation](#)
[Life & arts](#)
[Science](#)
[Professional](#)
[Business](#)
[API](#)
[Data](#)

[Blog](#)
[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)

site design / logo © 2021 Stack Exchange Inc; user contributions licensed under [cc by-sa](#). rev 2021.12.22.41046