



Products

Rust closures from factory functions

[log in](#) [sign up](#)

[Ask Question](#)

Asked 5 years, 4 months ago

Active 4 years, 8 months ago

Viewed 226 times



2



I have some Rust code I'm trying to get working but I'm not sure how to go about it.

```
fn main() {
    let names = vec!["foo", "bar", "baz"];
    let print = printer(names);
    let result = print();
    println!("{}", result);
    do_other_thing(names.as_slice());
}

fn printer(names: Vec<&str>) -> Box<Fn() -> String> {
    Box::new(move || {
        let text = String::new();
        for name in names {
            text = text + name;
        }
        text
    })
}

fn do_other_thing(names: &[&str]) {}
```

This compiles with:

```
error[E0477]: the type `[closure@src/main.rs:10:14: 16:6 names:std::vec::Vec<&str>]` does not fulfill the required lifetime
--> src/main.rs:10:5
|
10|     Box::new(move || {
|     ^ starting here...
11|         let text = String::new();
12|         for name in names {
13|             text = text + name;
14|         }
15|         text
16|     })
|     ^ ...ending here
= note: type must outlive the static lifetime
```

I have a vague idea of what's going on. It looks like there's a chance the closure will outlive the `names` parameter. I can annotate as `'static` but that doesn't feel right and even then I'd like to not move the vector so that `do_other_thing` works. I need to copy somehow.

[rust](#)

[Share](#)

[Improve this question](#)

[Follow](#)

edited Apr 27 '17 at 15:00



[Shepmaster](#)

305k • 59 • 824 • 1083

asked Aug 3 '16 at 15:37



[sommrid](#)

57 • 5

[Add a comment](#)

1 Answer

[Active](#) [Oldest](#) [Votes](#)



2



Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

The error says that `names` must outlive the static lifetime, this is because the boxed `Fn` has static lifetime. You have two options:

Accept all cookies [Customize settings](#)
1. Add the `'static` lifetime to `names` :

```
fn printer(names: Vec<&'static str> -> Box<Fn() -> String>){
    Box::new(move || {
        // ...
    })
}
```

2. Change the lifetime of the boxed `Fn` to match the `names` lifetime:

```
fn printer<'a>(names: Vec<&'a str>) -> Box<Fn() -> String + 'a>{
    Box::new(move || {
        // ...
    })
}
```

Note that the body of closure needs to be adjusted and that you are giving the ownership of `names` to `printer`, so you cannot use `names` in `do_other_thing`. Here is a fixed version:

```
fn main() {
    let names = vec!["foo", "bar", "baz"];
    let print = printer(&names);
    let result = print();
    println!("{}", result);
    do_other_thing(names.as_slice());
}

fn printer<'a>(names: &'a Vec<&'a str>) -> Box<Fn() -> String + 'a>{
    Box::new(move || {
        // this is more idiomatic
        // map transforms &&str to &str
        names.iter().map(|s| *s).collect()
    })
}
```

[Share](#)

[Improve this answer](#)

[Follow](#)

edited Apr 27 '17 at 15:02



[Shepmaster](#)

305k ● 59 ● 824 ● 1083

answered Aug 3 '16 at 15:55



[malbarbo](#)

9,673 ● 1 ● 35 ● 52

Thanks! The + 'a to tie the closure lifetime was exactly what I was looking for.

— [sommid](#)

Aug 3 '16 at 18:39

[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](#) 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

Passing in functions and returning a boxed closure in Rust

Returning non-trivial gstreamer "pad callbacks" as boxed closures

Related

336

How do I print the type of a variable in Rust?

Why doesn't println! work in Rust unit tests?

How to disable unused code warnings in Rust?

How do I split a string in Rust?

Rust package with both a library and a binary?

Factory method: instance does not live long enough







Returning non-trivial gstreamer "pad callbacks" as boxed closures

Store data that implements a trait in a vector

Why does the Rust compiler not optimize code assuming that two mutable references cannot alias?

Understanding lifetimes: max lifetime and 'static

Hot Network Questions

-  Applied mathematics or Computer science PhD?
-  Why does the prophecy imply Macbeth has to murder the king?
-  How can I let my opponent know that I'm touching a piece to adjust it?
-  Changing a color of a link
-  Schrödinger's cat program
- [more hot questions](#)
-  Question feed

STACK OVERFLOW

Questions
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