

[Skip to main content](#)

[Stack Overflow](#)

1. [About](#)

2. [Products](#)

3. [For Teams](#)

1. [Stack Internal](#) Implement a knowledge platform layer to power your enterprise and AI tools.

2. [Stack Data Licensing](#) Get access to top-class technical expertise with trusted & attributed content.

3. [Stack Ads](#) Connect your brand to the world's most trusted technologist communities.

4. [Releases](#) Keep up-to-date on features we add to Stack Overflow and Stack Internal.

5. [About the company](#) [Visit the blog](#)

Loading...

[current community](#)

[Stack Overflow](#)

[help](#) [chat](#)

- [Meta Stack Overflow](#)

your communities

[Sign up](#) or [log in](#) to customize your list.

[more stack exchange communities](#)

[company blog](#)

3. [Log in](#)

4. [Sign up](#)

[AI Assist is now on Stack Overflow](#). Start a chat to get instant answers from across the network. Sign up to save and share your chats.

[Home](#)

[Questions](#)

[AI Assist](#)

[Tags](#)

[Challenges](#)

7. [Chat](#)

[Articles](#)

[Users](#)

[Companies](#)

[Collectives](#)

Communities for your favorite technologies. [Explore all Collectives](#)

Stack Internal

Stack Overflow for Teams is now called **Stack Internal**. Bring the best of human thought and AI automation together at your work.

[Try for free](#) [Learn more](#)

[Stack Internal](#)

Bring the best of human thought and AI automation together at your work. [Learn more](#)

Collectives™ on Stack Overflow

Find centralized, trusted content and collaborate around the technologies you use most.

[Learn more about Collectives](#)

Stack Internal

Knowledge at work

Bring the best of human thought and AI automation together at your work.

[Explore Stack Internal](#)

[How to return a function from a function for the purpose of parameter injection?](#)

[Ask Question](#)

Asked 4 years, 2 months ago

Modified [4 years, 2 months ago](#)

Viewed 655 times

2

I am trying to return a function from another function.

```
// Immediately run function passed in
fn set_timeout(f: &dyn Fn()) {
    f();
}

fn callback() -> dyn Fn() {
    return fn() {
        println!("hi");
    };
}

fn main() {
    set_timeout(&callback());
}
```

My end goal is to achieve a long form variation of the following code where I move `fooRef1` into the inner function through a parameter on `callback(fooRef1)` (parameter injection).

```
fn main() {
    let fooRef = Arc::new(Mutex::new(String::from("foo")));
    let fooRef1 = fooRef.clone();

    set_timeout(&(move || {
        let mut foo = fooRef1.lock().unwrap();
        foo.push_str("bar");
        println!("{}", foo);
    }));
}
```

Why does the inline callback closure work but the separated function does not?

EDIT: Here's the error messages from rustc, as obtained via Rust playground.

```
Compiling playground v0.0.1 (/playground)
error: expected one of `.`, `;`, `?`, `}`, or an operator, found keyword `fn`
--> src/main.rs:7:12
|
7 |     return fn() {
|           ^^ expected one of `.`, `;`, `?`, `}`, or an operator

error[E0746]: return type cannot have an unboxed trait object
--> src/main.rs:6:18
|
6 | fn callback() -> dyn Fn() {
|           ^^^^^^^^^ doesn't have a size known at compile-time

= note: for information on trait objects, see <https://doc.rust-lang.org/book/ch17-02-trait-objects.html#using-trait-objects>
= note: if all the returned values were of the same type you could use `impl Fn()` as the return type
= note: for information on `impl Trait`, see <https://doc.rust-lang.org/book/ch10-02-traits.html#returning-types-that-implement-trait>
= note: you can create a new `enum` with a variant for each returned type
help: return a boxed trait object instead
|
6 | fn callback() -> Box<dyn Fn()> {
7 |     Box::new(return fn() {
8 |         println!("hi");
9 |     });
|

error[E0277]: the size for values of type `(dyn Fn() + 'static)` cannot be known at compilation time
--> src/main.rs:13:16
|
```

```
13 |   set_timeout(&callback());
    |           ^^^^^^^ doesn't have a size known at compile-time
    |
= help: the trait `Sized` is not implemented for `(dyn Fn() + 'static)`
= note: the return type of a function must have a statically known size
```

Some errors have detailed explanations: E0277, E0746.

For more information about an error, try `rustc --explain E0277`.

error: could not compile `playground` due to 3 previous errors

- [rust](#)

[Share](#)

[Improve this question](#)

Follow

[edited Oct 3, 2021 at 2:49](#)

[cadolphs](#)

9,72311 gold badge3030 silver badges4545 bronze badges

asked Oct 3, 2021 at 2:01

[David Alsh](#)

8,05788 gold badges4747 silver badges8383 bronze badges

3

1

Please include the errors you are seeing. They are likely to be illuminating.

Peter Hall – [Peter Hall](#)

2021-10-03 02:06:51 +00:00

Commented Oct 3, 2021 at 2:06

1

In the meantime, one of these is likely what you want, depending on your requirements: [play.rust-lang.org/...](https://play.rust-lang.org/)

Peter Hall – [Peter Hall](#)

2021-10-03 02:09:27 +00:00

Commented Oct 3, 2021 at 2:09

Note that `fn` and `Fn` are very different. `fn` is a function pointer (a type) and refers to static functions that do not capture their environment. `Fn` is a trait and can be implemented by closures as well as static functions.

Peter Hall – [Peter Hall](#)

2021-10-03 02:11:47 +00:00

Commented Oct 3, 2021 at 2:11

[Add a comment](#) | .

1 Answer

Sorted by: [Reset to default](#)

Highest score (default) Trending (recent votes count more) Date modified (newest first) Date created (oldest first)

1

So, looking at the errors from `rustc`, the first problem is that `return` expects something other than a function declaration. Instead, as pointed out in the comment by @peter-hall you have to define the function first, and then just use it as the return value:

```
def outer() -> fn() {
  fn inner() {
    //blabla
  }
  inner
}
```

And then the next problems are about `dyn`, and that you can't just return a naked `dyn` object.

One of the solutions to this is pointed out by the compiler: Put your trait object into a `Box`, i.e., allocate it on the heap.

Alternatively, use the `impl Trait` syntax: https://doc.rust-lang.org/rust-by-example/trait/impl_trait.html

Or just state that your return type is that of `fn`. Credit to those suggestions all to `peter-hall`.

[Share](#)

[Improve this answer](#)

Follow

answered Oct 3, 2021 at 2:54

[cadolphs](#)

9,72311 gold badge3030 silver badges4545 bronze badges

Sign up to request clarification or add additional context in comments.

Comments

Add a comment

Your Answer

Thanks for contributing an answer to Stack Overflow!

- Please be sure to *answer the question*. Provide details and share your research!

But *avoid* ...

- Asking for help, clarification, or responding to other answers.
- Making statements based on opinion; back them up with references or personal experience.

To learn more, see our [tips on writing great answers](#).

Sign up or [log in](#)

Sign up using Google

Sign up using Email and Password

Submit

Post as a guest

Name

Email

Required, but never shown

Post as a guest

Name

Email

Required, but never shown

Post Your Answer Discard

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

Start asking to get answers

Find the answer to your question by asking.

[Ask question](#)

Explore related questions

- [rust](#)

See similar questions with these tags.

- The Overflow Blog
[Introducing Stack Overflow AI Assist—a tool for the modern developer](#)
[Treating your agents like microservices](#)
- Featured on Meta
[Chat room owners can now establish room guidelines](#)
[AI Assist is now available on Stack Overflow](#)
[Policy: Generative AI \(e.g., ChatGPT\) is banned](#)

Related

[2](#)

[How can I return a function?](#)

[15](#)

[How to pass a function as argument in Rust](#)

[2](#)

[Pass function as argument to another function](#)

[3](#)

[How can I pass a function with 2 parameters as a parameter to a function?](#)

[182](#)

[How do you pass a Rust function as a parameter?](#)

[1](#)

[How to assign function return value to a variable in Rust?](#)

[5](#)

[How would one return a function from a function in Rust?](#)

[15](#)

[What is the correct syntax to return a function in Rust?](#)

[0](#)

[Returning reference from a function in Rust](#)

[6](#)

[Return function that returns a reference](#)

Hot Network Questions

- [Vertical alignment of equations inside tabularx](#)
- [How to protect 100-year-old wooden floors for the next 100 years?](#)
- [When did "doubt" meaning question or point of uncertainty fall out of use outside of Indian English?](#)
- [Is there a term for forming emotional or relational bonds with AI chatbots? If not, is "cyberpomorphic" a valid neologism?](#)
- [How do I add a text editor to my \(debian-based\) initramfs?](#)
- [Is zero-point energy a mathematical artifact?](#)
- [What's the point of swapping polarity between mod slots?](#)
- [Best Welding Method to Fix a Damaged Steel Surly Frame](#)
- [Schottky OR-ing diodes passing reverse voltage; is it a faulty diode?](#)
- [Storm Sphere and Obstruction](#)
- ["Save as webpage, HTML Only" Fail while offline, Firefox](#)
- [Generating Irregular, Separated Polygonal Cells along a Curve in Geometry Nodes](#)
- [is it possible to keep the space when using url in latex](#)
- [The state of icy worlds during differentiation and accretion, and how they survive with their water](#)
- [When should I use tied notes vs. the whole note duration?](#)
- ["Imagine if..." vs. "Imagine..."](#)
- [Curve has weird bumps when triangulated](#)
- [King defense setups](#)
- [The fourth moment of the Riemann zeta function without absolute values](#)
- [Can I Automatically List Supported Kernel Parameters in GRUB?](#)
- [PSE Advent Calendar 2025 \(Day 3\): A gift from my travel agent](#)

- ["Creatures your opponents control" after the ability is activated](#)
- [Why does clang zero "eax" before calling a function with unspecified parameters, but gcc doesn't?](#)
- [Misunderstanding the lorentz transformation](#)

[more hot questions](#)

[Question feed](#)

Subscribe to RSS

Question feed

To subscribe to this RSS feed, copy and paste this URL into your RSS reader.

[Stack Overflow](#)

- [Questions](#)
- [Help](#)
- [Chat](#)

[Business](#)

- [Stack Internal](#)
- [Stack Data Licensing](#)
- [Stack Ads](#)

[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 © 2025 Stack Exchange Inc; user contributions licensed under [CC BY-SA](#) . rev 2025.12.4.37651