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

Returning references nested a few methods deep inside RefCell

I am able to get my reference fine until the Inner object. inner.get_bar() works fine. How do it return the same reference with Outer. I have tried a few things based on the reading but would appreciate the help.

foo_addr is some address in memory that is passed from outside.

attempt1 rightfully fails with temporary value created here. attemt2, attempt3 cause lifetime issues and requires that Inner must outlive the Bar and also complain about the temporary value due to the reference to inner.

```
use alloc::rc::Rc;
use core::cell::{Ref, RefCell};
use owning_ref::RefRef;

struct Outer {
    inner: Rc<RefCell<Inner>>,
}

impl Outer {
    pub fn new(foo_addr: usize) -> Self {
        Self {
            inner: Rc::new(RefCell::new(Inner::new(foo_addr))),
        }
    }

    pub fn get_bar(&self, idx: usize) -> Result<&Bar, SomeErr> {
        todo!()
    }

    pub fn get_bar_attempt1(&self, idx: usize) -> Result<&Bar, SomeErr> {
        self.inner.borrow().get_bar(idx)
    }

    pub fn get_bar_attempt2(&self, idx: usize) -> Ref<'_, Result<&Bar, SomeErr>> {
        Ref::map(self.inner.borrow(), |inner| &inner.get_bar(idx))
    }

    pub fn get_bar_attempt3(&self, idx: usize) -> RefRef<Inner, Result<&Bar, SomeErr>> {
        RefRef::new(self.inner.borrow()).map(|inner| &inner.get_bar(idx))
    }
}

struct Inner {
    foo_addr: usize,
}

impl Inner {
    fn new(foo_addr: usize) -> Self {
        Self { foo_addr }
    }

    fn get_bar(&self, idx: usize) -> Result<&Bar, SomeErr> {
        let foo = self.get_foo()?;
        foo.get_bar(idx)
    }

    fn get_foo(&self) -> Result<&Foo, SomeErr> {
        let foo = unsafe { &*(self.foo_addr as *const Foo) };
        Ok(foo)
    }
}

#[repr(C)]
struct Foo {
    bar_list: [Bar; 10],
```

```
}

impl Foo {
    fn get_bar(&self, idx: usize) -> Result<&Bar, SomeErr> {
        Ok(&self.bar_list[idx])
    }
}

#[repr(C)]
struct Bar {
    val: u32,
}

impl Bar {
    pub fn getter(&self) -> u32 {
        self.val
    }

    pub fn setter(&mut self, new_val: u32) {
        self.val = new_val
    }
}

#[derive(Debug)]
enum SomeErr {}
```

- rust

Share

Improve this question

Follow

edited Nov 8, 2023 at 20:40

asked Nov 8, 2023 at 20:16

scoplexian

12311 silver badge99 bronze badges

1

4

Have you read How do I return a reference to something inside a RefCell without breaking encapsulation? The answer is you can't return a bare reference to a value derived from within a `RefCell`, a `Ref` must be alive and guarding the value.

kmdreko – kmdreko

2023-11-08 20:44:10 +00:00

Commented Nov 8, 2023 at 20:44

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)

0

You can use `Ref::filter_map()`. Since it requires an `Option` and not a `Result` this is quite clunky, but it works:

```
pub fn get_bar(&self, idx: usize) -> Result<Ref<'_, Bar>, SomeErr> {
    let mut err = None;
    Ref::filter_map(self.inner.borrow(), |inner| match inner.get_bar(idx) {
        Ok(value) => Some(value),
        Err(error) => {
            err = Some(error);
            None
        }
    })
    .map_err(|_| err.unwrap())
}
```

Share

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.

Linked

[65](#)

[How do I return a reference to something inside a RefCell without breaking encapsulation?](#)

Related

[3](#)

[Struct with a mutable reference to another instance of the same type](#)

[3](#)

[Returning iterator to RefCell](#)

[1](#)

[Trouble with bidirectional references using Option<Rc<RefCell<T>>>](#)

[13](#)

[Returning iterator of a Vec in a RefCell](#)

[6](#)

[Why does RefCell not have the same scoping as regular references?](#)

[0](#)

[How to iterate over an Rc<RefCell<T>> that returns raw mutable references](#)

[65](#)

[How do I return a reference to something inside a RefCell without breaking encapsulation?](#)

Hot Network Questions

- [How to protect 100-year-old wooden floors for the next 100 years?](#)
- [2 ways of resolution of vectors](#)
- [What do the terms postponement, slip and scrub mean in the context of the space shuttle program?](#)
- [How to use Bézier curves in `TikZ` to draw a quotation mark?](#)
- ["Imagine if..." vs. "Imagine..."](#)
- [Is there any point in sharing a scientific discovery as a "nobody"?](#)
- [Critiques of the real vs. nominal definition distinction](#)
- ["Creatures your opponents control" after the ability is activated](#)
- [Is Brian Ripley's Pattern Recognition and NN book still an insightful read?](#)
- [What is the difference between an ultraviolet limit and ultraviolet stability?](#)
- [Correct website for Thailand ETA application form](#)
- [Documentation for DUSHIN software](#)
- [Could a world tree have different biomes?](#)
- [I can't add a loop cut on one face](#)
- [Is there a term for forming emotional or relational bonds with AI chatbots? If not, is "cyberpomorphic" a valid neologism?](#)
- [When did "doubt" meaning question or point of uncertainty fall out of use outside of Indian English?](#)
- [is it possible to keep the space when using url in latex](#)
- [Statistical testing for bimodal/multimodal sample](#)
- [Trying to understand UK tax brackets for salary above 150k](#)
- [Download is performed unsandboxed as root - Why am I getting this message, and how do I fix it?](#)
- [Installing root certificates by government](#)
- [Is there any way to restore hearts without consuming anything, without hot springs, and without shrines?](#)

Subscribe to RSS

Question feed

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