



Products

Returning a structure referencing a new &str

Log in Sign up

Ask Question

Asked 2 years, 3 months ago

Active 2 years, 3 months ago

Viewed 67 times



0



I'm currently writing a program and at some point I have to deal with a structure containing `&str` values. Let's consider it has this look:

```
struct Book<a> {  
    url: &a str,  
}
```

I have a function that fetches a book from a remote storage and returns an URL pointing to that book, which looks like this:

```
fn fetch_book<a>(id: u64) -> Book<a> {  
    // do some stuff to get a 'key' variable that allows to access the book  
    Book {  
        // Here we turn the formatted String into an &str  
        url: &format!("https://<ip-address>/books?id={}&key={}", id, key)  
    }  
}
```

This function won't compile because Rust tells we are returning a value that references data owned by the current function.

So, what can I do to make this function works properly (without going `unsafe`)?

[string](#) [reference](#) [rust](#)

Share

Improve this question

Follow

asked Sep 21 '19 at 19:04



ClementNerma

1,041 ● 1 ● 8 ● 16

2

Looks like you want `String` instead of `&str` as the type of `url`.

– Sven Marnach

Sep 21 '19 at 19:14

Indeed, but I'd like to *not* store a `String` in my structure - the strings in there are not supposed to be mutable.

– ClementNerma

Sep 21 '19 at 19:39

@ClementNerma You *build* the URL, it's not borrowed from elsewhere. You have to store it somewhere, mutable or not. The fact `String` is mutable is just a bonus here, the important point is that it owns the str

– Denys Séguret

Sep 21 '19 at 19:40

That's true, but I can't figure out *why* I can't put this new `&str` inside this structure, as it should live long enough for that.

– ClementNerma

Sep 21 '19 at 19:41

2

At the end of the function, the values that are created are either dropped or moved. If you don't move the str (ie return it in an owning structure), it's dropped and references to it makes no sense anymore.

– Denys Séguret

Sep 21 '19 at 19:44

Show 3 more comments

1 Answer

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



0



Your privacy

As @DenysSéguret pointed out in the comments:

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

Accept all cookies At the end of the function, the values that are created are either dropped or moved. If you don't move the str (ie return it in an owning structure), it's dropped and references to it makes no sense anymore.

Plus, according to [this question](#) posted on Rust's users forum, it is indeed not possible to return a `&str` from a "freshly-generated" string.

[Share](#)

[Improve this answer](#)

[Follow](#)

edited Sep 22 '19 at 21:19



[Azat Ibrakov](#)

7,912 ● 9 ● 35 ● 43

answered Sep 21 '19 at 19:58



[ClementNerma](#)

1,041 ● 1 ● 8 ● 16

[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 [string](#) [reference](#) [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

Related

[What are the differences between Rust's 'String' and 'str'?](#)

[Unsafe references to attributes of the same struct](#)

[TypeError: a bytes-like object is required, not 'str' when writing to a file in Python3](#)

[Create *mut *mut to a struct](#)

[In Rust, can you own a string literal?](#)

[Entry::Occupied.get\(\) returns a value referencing data owned by the current function even though hashmap should have the ownership](#)

[Semantics of lifetime parameters](#)

Hot Network Questions

- [Text overflow in tabularx](#)
- [What is this large long-legged orange and black insect?](#)
- [Changing a color of a link](#)
- [Why does the prophecy imply Macbeth has to murder the king?](#)
- [Numbers, Racked Up](#)
- [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](#)

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