How to insert a String variable to a global mutable HashMap(using lazy_static and Mutex) without causing "does not live long enough" problem?

Asked 11 months ago
Active 11 months ago
Viewed 160 times

3

I am using Rust and I want to use a global mutable `HashMap` for convenience. However, while it is possible to define a global, mutable `HashMap` using `lazy_static` and `Mutex`, it is hard for my `String` variable defined in my function to have the same life time of the gloabl `HashMap`.

I've tried insert a &str directly and it worked well. Is there any way to convert a String to pure value?

```
lazy_static! {
    static ref USER_TOKEN_HASHMAP: Mutex<HashMap<&'static str, &'static str>> = {
        let mut m = HashMap::new();
        Mutex::new(m)
    };
}

fn func() {
    let mut _map = USER_TOKEN_HASHMAP.lock().unwrap();
    let user_email = String::from("aaa");
    let user_password = String::from("bbb");
    _map.insert(user_email.as_str(), user_password.as_str());
}
```

Error Info:

```
`user_email` does not live long enough
values in a scope are dropped in the opposite order they are defined
rustc(E0597)
```

rust mutex lazy-static

Share
Improve this question
Follow

asked Jan 13 at 8:21

Stacker Dragon
53 ● 3

2

Define `USER_TOKEN_HASHMAP` as `Mutex<HashMap<String, String>>` and drop the `as_str()` , and your code will work as intended.
– user4815162342
Jan 13 at 8:27

Add a comment

1 Answer

Active Oldest Votes

6

✓

> I've tried insert a &str directly and it worked well. Is there any way to convert a String to pure value?

The issue here is that you're taking the problem the wrong way around: `&'static str` will basically only work for literals, because it means "a pointer to a string which lives somewhere else but is never collected" (`'static` meaning "lives forever"). Pretty much the only possible options would be static data (living in the binary itself and thus living as long at the program runs) or leaking the memory (which is not usually a good idea).

Here what you want is that your map store the strings themselves, and when a string is removed from the map it should be collected. That's `String` . That's what it does and that's what it is used for. Maybe `Cow<'static, str>` in the odd case where you have a mix of static data and dynamically allocated data, but that doesn't seem to be the case here.

And thus the fix is:

```
lazy_static! {
    static ref USER_TOKEN_HASHMAP: Mutex<HashMap<String, String>> = Mutex::new(HashMap::new());
}

fn func() {
    let mut _map = USER_TOKEN_HASHMAP.lock().unwrap();
    let user_email = String::from("aaa");
    let user_password = String::from("bbb");
    _map.insert(user_email, user_password);
}
```

Incidentally I'd recommend against the `_` prefix of `_map`: it means you want the item named / alive for some reason but you don't want to use it (unused bindings prefixed with `_` are not warned against). Here it's being actively used, so it should not be prefixed.

Share
Improve this answer
Follow
edited Jan 13 at 9:42

answered Jan 13 at 8:41

Masklinn
20.6k ● 1 ● 17 ● 34

You theoretically *could* leak your strings to convert them to `&'static str`, but then they wouldn't be collected even when removed from the map. You'd need some very good reason for that approach, and I currently can't think of any.
– Sven Marnach
Jan 13 at 9:28

1

@SvenMarnach yeah that's basically what I hint at at the end of the first paragraph, I can see that being an option in a small CLI where releasing memory is just overhead e.g. it runs for a fairly short time and / or nothing is ever removed from the map, so explicitly releasing memory is not really useful (it just wastes time since the OS will release the entire thing when the process dies anyway).
– Masklinn
Jan 13 at 9:42

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 mutex lazy-static 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

How to convert a String into a &'static str

Related

Why does the variable not live long enough?

Factory method: instance does not live long enough

HashMap key does not live long enough

Borrowed value does not live long enough - string slice into HashMap

Value does not live long enough

Variable in loop does not live long enough

Primitive variable does not live long enough

Get item's reference from global HashMap in Rust

Hot Network Questions

Tax implications of large gift

EU ETS: if within-EU flight emissions are limited to climate goals, is there still a reason not to fly as long as you can afford it?

Why is the light source not showing but light is being cast on the object

Can a Pyromancer Sorcerer deal fire damage to a creature under the effect of Invulnerability?

Is it common practice to apply identical processing effects to a batch of photos?

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