



Products

## RUST Borrowing issue on custom concurrentMap Struct [duplicate]

[Log in](#) [Sign up](#)

[Ask Question](#)

Asked 1 year, 9 months ago

Active 1 year, 9 months ago

Viewed 37 times



0



This question already has answers here:

[How to return a reference to a sub-value of a value that is under a mutex?](#) (5 answers)

Closed last year.

Hello Im a RUST begginer Im dealing with some Borrowing issues that I can't solve

I want to make a Concurrent Map, I saw some examples to do it where I need it and it works, but now I want to generalize the solution writing my own mod.

Here it is

```
use std::collections::HashMap;
use std::hash::Hash;
use std::sync::Mutex;

pub struct ConcurrentMap<K, V> {
    elms: Mutex<HashMap<K, V>>,
}

impl<K: Hash + Eq, V> ConcurrentMap<K, V> {
    pub fn new() -> ConcurrentMap<K, V> {
        let map: HashMap<K, V> = HashMap::new();
        ConcurrentMap {
            elms: Mutex::new(map),
        }
    }

    pub fn get(&self, k: &K) -> Option<&V> {
        let map = self.elms.lock();
        match map {
            Ok(t) => {
                return match t.get(k) {
                    None => None,
                    Some(v) => Some(v),
                }
            }
            Err(e) => panic!("GET Concurrent Map Error: {}", e),
        }
    }
}
```

My problem is in `Some(v) => Some(v)` line, I dont know how to copy the value.

If I make that V implement Copy trait then I can't use this Map to store String because String does not implement Copy trait.

Maybe the last matching is not necessary but is the last thing I tried.

[rust](#) [borrowing](#)

Share

Improve this

question

Follow

asked Mar 21 '20 at 2:16



[German Faller](#)

490 ● 6 ● 13

1

Have you explored using `Clone` instead of `Copy` ?

– [loganfsmth](#)

Mar 21 '20 at 2:40

@loganfsmth Im dealing with that at this moment, any link is welcome, If I get the the solution I will Answer my own question.

– [German Faller](#)

Mar 21 '20 at 2:52

[Add a comment](#)

1 Answer

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

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

Accept all cookies Customize settings

0



As @logansmyth suggested, I implemented Clone insted of Copy trait in V(alue) type.

Also, as I mention, I changed last matching with a simple Map combinator

So this is how it looks now

```
use std::collections::HashMap;
use std::hash::Hash;
use std::sync::Mutex;

pub struct ConcurrentMap<K, V> {
    elms: Mutex<HashMap<K, V>>,
}

impl<K: Hash + Eq, V: Clone> ConcurrentMap<K, V> {
    pub fn new() -> ConcurrentMap<K, V> {
        let map: HashMap<K, V> = HashMap::new();
        ConcurrentMap {
            elms: Mutex::new(map),
        }
    }

    pub fn insert(&self, k: K, v: V) {
        let map = self.elms.lock();
        match map {
            Ok(mut t) => t.insert(k, v),
            Err(e) => panic!("INSERT Concurrent Map Error: {}", e),
        };
    }

    pub fn get(&self, k: K) -> Option<V> {
        let map = self.elms.lock();
        match map {
            Ok(t) => t.get(&k).map(|v| v.clone()),
            Err(e) => panic!("GET Concurrent Map Error: {}", e),
        }
    }
}
```

Usage example:

```
lazy_static! {
    static ref PROPERTIES: ConcurrentMap<String, String> = ConcurrentMap::new();
}

PROPERTIES.insert("Key".to_string(), "val".to_string());
PROPERTIES.get("Key".to_string()).unwrap().to_string();
```

[Share](#)

[Improve this answer](#)

[Follow](#)

edited Mar 21 '20 at 3:44

answered Mar 21 '20 at 3:39



[German Faller](#)

490 • 6 • 13

[Add a comment](#)

Not the answer you're looking for? Browse other questions tagged [rust](#) [borrowing](#) 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 return a reference to a sub-value of a value that is under a mutex?](#)

#### Related

336 [How do I print the type of a variable in Rust?](#)

[Why doesn't println! work in Rust unit tests?](#)

[Rust borrowing issue with a getter method on a struct](#)

7 [Rust multiple mutable borrowing](#)






[Implementing constructor function in rust trait](#)

[Rust borrowing rules in 'for...in' loop](#)

[Rust struct within struct: borrowing, lifetime, generic types and more total confusion](#)

[String concatenation in rust and borrowing](#)

## Hot Network Questions

-  [Who \(or what\) created the atropal?](#)
  -  [Why does making a quantum circuit more noise resilient make it easier to simulate classically?](#)
  -  [On what basis do countries repay international loans?](#)
  -  [Schrödinger's cat program](#)
  -  [Why is my reasoning incorrect - probability?](#)
- [more hot questions](#)

## 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