Rust lifetime - variable does not live long enough error [duplicate]

Asked 4 years, 11 months ago

Active 4 years, 11 months ago

Viewed 311 times

▲

1

▼

This question already has answers here:

Is there any way to return a reference to a variable created in a function? (4 answers)

Closed 4 years ago.

I'm running into an issue with lifetimes in rust that I'm having trouble figuring out. I've tried a lot of tweaks to the below but I keep introducing new errors. I want index to return a Vector object.

I have:

```
struct Matrix<T> {
    num_rows: i32,
    num_cols: i32,
    data: Vec<T>
}

struct Vector<T> {
    data: Vec<T>
}
```

And I'm trying to do

```
impl<T: Clone> Index<usize> for Matrix<T> {
    type Output = Vector<T>;

    fn index(&self, i: usize) -> &Vector<T> {
        let index = i as i32;
        let start = (index * &self.num_cols) as usize;
        let end = (((index + 1) * &self.num_cols) - 1) as usize;
        let data_slice = &self.data[start..end];
        let data = data_slice.to_vec();
        let vector_temp = Vector::<T>::new(data);
        return &vector_temp;
    }
}
```

But I'm getting

```
error: `vector_temp` does not live long enough
  --> src\main.rs:45:17
   |
45 |         return &vector_temp;
   |                 ^^^^^^^^^^^ does not live long enough
46 |     }
   |     - borrowed value only lives until here
   |
note: borrowed value must be valid for the anonymous lifetime #1 defined on     the block at 38:44...
  --> src\main.rs:38:45
   |
38 |     fn index(&self, i: usize) -> &Vector<T> {
   |                                             ^

 error: aborting due to previous error

 error: Could not compile `hello_world`.
```

I haven't fully grokked lifetimes in rust yet so was hoping someone could help me out. Thanks!

rust lifetime

Share
Improve this question
Follow

asked Jan 8 '17 at 3:30

user5534715
55 ● 2 ● 6

Add a comment

1 Answer

Active Oldest Votes

▲

3

▼

You're trying to return a reference to an object that will be destroyed once the function ends. `vector_temp` will no longer exist once `index` returns and so returning a reference to it is illegal (because that reference would point to nowhere).

What you want to do, is return the slice you're creating and let the caller decide what to do with it:

```
impl<T: Clone> Index<usize> for Matrix<T> {
    type Output = [T];

    fn index(&self, i: usize) -> &[T] {
        let index = i as i32;
        let start = (index * &self.num_cols) as usize;
        let end = (((index + 1) * &self.num_cols) - 1) as usize;

        &self.data[start..end]
    }
}
```

You can then have the caller do what you did in your original implementation:

```
let m1 = Matrix { ... };
let owned_vector = m1[index_here].to_owned();
```

I am not 100% sure where you go from here, given I'm not sure how much further you're going to take this. Returning the unsized slice can have its problems and so without knowing your specific use case I'm not sure if there is a better way you could handle this.

Hopefully this helps with your immediate issue though.

Share
Improve this answer
Follow

edited Jan 8 '17 at 4:03

answered Jan 8 '17 at 3:42

Simon Whitehead
59.3k ● 7 ● 101 ● 130

Add a comment

Not the answer you're looking for? Browse other questions tagged rust lifetime 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

85
Is there any way to return a reference to a variable created in a function?

Related

What is the lifetime of a static variable in a C++ function?

Factory method: instance does not live long enough

Borrowed value does not live long enough compiler error for struct

Borrow data out of a mutex "borrowed value does not live long enough"

Rust lifetime for database connection bundle

3
Value does not live long enough with a generic function that creates a container, adds items to it, then iterates over the items

Problems with Tuple's lifetime in rust.

Can't reference a slice of &str because it doesn't live long enough, even though what it references does

2
What does the Rust error code E0495 mean?

Store data that implements a trait in a vector

Hot Network Questions

What does the numbers mean in this guitar tab if they already gave the chords to play?

Send Geometry nodes value into Shading tab

What does this entry on the Rocinante's pilot quick-menu mean?

Question on OEIS A000085

Who (or what) created the atropal?

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