

[Skip to main content](#)

Stack Overflow

1. [About](#)
2. Products
3. [For Teams](#)
4. [Stack Internal Implement a knowledge platform layer to power your enterprise and AI tools.](#)
5. [Stack Data Licensing Get access to top-class technical expertise with trusted & attributed content.](#)
6. [Stack Ads Connect your brand to the world's most trusted technologist communities.](#)
7. [Releases Keep up-to-date on features we add to Stack Overflow and Stack Internal.](#)
8. [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](#)

[Parsing text file lines into numbers using std::iter::Iterator map](#)

[Ask Question](#)

Asked 10 years, 6 months ago

Modified [10 years, 6 months ago](#)

Viewed 1k times

2

I'm trying to read and parse a text file in Rust. Each line is a signed integer. I'm able to do it using `for line in lines` iteration but I'm unable to do it with a `iter().map(|l| ...)` one-liner. I'm getting a

```
expected `&core::result::Result<collections::string::String, std::io::error::Error>`,
found `core::result::Result<_, _>`
```

when I try to pattern match `Ok(s) => match s.parse()` but I'm unable to get to the bottom of what I am doing wrong. The whole example is below. The code on the bottom is the code that is producing the error.

Can anyone tell what I am doing wrong?

```
use std::error::Error;
use std::fs::File;
use std::io::BufReader;
use std::io::prelude::*;
use std::path::Path;

fn main() {
    // Create a path to the desired file
    let path = Path::new("input/numbers.txt");
    let display = path.display();

    // Open the path in read-only mode, returns `io::Result<File>`
    let file = match File::open(&path) {
        // The `description` method of `io::Error` returns a string that describes the error
        Err(why) => panic!("couldn't open {}: {}", display, Error::description(&why)),
        Ok(file) => file,
    };

    // Collect all lines into a vector
    let reader = BufReader::new(file);
    let lines: Vec<_> = reader.lines().collect();

    // Works.
    let mut nums = vec![];
    for l in lines {
        println!("{}:?", l);
        let num = match l {
            Ok(s) => match s.parse() {
                Ok(i) => i,
                Err(_) => 0
            },
            Err(_) => 0
        };
        nums.push(num);
    }

    // Doesn't work!
    let nums: Vec<i64> = lines.iter().map(|l| match l {
        Ok(s) => match s.parse() {
            Ok(i) => i,
            Err(_) => 0
        },
        Err(_) => 0
    });
}
```

- [file](#)

- [text](#)

- [io](#)

- [rust](#)

[Share](#)

[Improve this question](#)

Follow

[edited Jun 2, 2015 at 23:30](#)

[Shepmaster](#)

439k116116 gold badges1.3k1.3k silver badges1.5k1.5k bronze badges

asked Jun 2, 2015 at 23:20

[neektza](#)

53366 silver badges1111 bronze badges

[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)

8

Let's look at the *complete* error message, which points to the error for us:

```
<anon>:5:9: 5:14 error: mismatched types:
expected `&core::result::Result<&str, ()>`,
  found `core::result::Result<_, _>`  
(expected &-ptr,
  found enum `core::result::Result`) [E0308]
<anon>:5      Ok(s) => match s.parse() {
  ^~~~~~
```

The compiler is expecting a `&Result`, but found a `Result`, and the issue is with the `Ok(s)` pattern. The type of `l` is a *reference* to a `Result` because you are using `iter` - which returns [an iterator of references](#) to the items in the vector.

The shortest fix is to add a `&` to the pattern match for the closure variable:

```
fn main() {
    let lines: Vec<Result<_, ()>> = vec![Ok("1"), Ok("3"), Ok("5")];

    // HERE                                V
    let nums: Vec<i64> = lines.iter().map(|&l| match l {
        Ok(s) => match s.parse() {
            Ok(i) => i,
            Err(_) => 0
        },
        Err(_) => 0
    }).collect();

    println!("{}:{}\n", nums)
}
```

I also had to add `collect` to go back to a `Vec`.

The other change you could make would be to *consume* the input vector using `into_iter` and then iterate on each value in the vector:

```
// HERE                                V~~~~~
let nums: Vec<i64> = lines.into_iter().map(|l| match l {
```

And for good measure, you could use `ok`, `and_then`, and `unwrap_or` to say the same thing a bit more succinctly:

```
let nums: Vec<i64> = lines.into_iter().map(|l| {
    l.ok().and_then(|s| s.parse().ok()).unwrap_or(0)
}).collect();
```

[Share](#)

[Improve this answer](#)

Follow

[edited Jun 2, 2015 at 23:50](#)

answered Jun 2, 2015 at 23:28

[Shepmaster](#)

439k116116 gold badges1.3k1.3k silver badges1.5k1.5k bronze badges

Sign up to request clarification or add additional context in comments.

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

- [file](#)
- [text](#)

- [io](#)
- [rust](#)

See similar questions with these tags.

- The Overflow Blog
 - [Introducing Stack Overflow AI Assist—a tool for the modern developer](#)
 - [Treating your agents like microservices](#)
- Featured on Meta
 - [Chat room owners can now establish room guidelines](#)
 - [AI Assist is now available on Stack Overflow](#)
 - [Policy: Generative AI \(e.g., ChatGPT\) is banned](#)

Related

- 7
[How to combine reading a file line by line and iterating over each character in each line?](#)
- 4
[How to simplify parsing a text file to a vector of values?](#)
- 3
[Parsing strings into integers](#)
- 14
[Why can't I collect the Lines iterator into a vector of Strings?](#)
- 0
[How do I create an iterator of lines from a file that have been split into pieces?](#)
- 2
[Take numbers from a file and put into a Vec<i32> but keep getting error](#)
- 2
[How to read a text File in Rust and read multiple Values per line](#)
- 0
[How can I parse the .nth\(\) line in a file as an integer?](#)
- 1
[Reading .txt file into HashMap](#)
- 1
[Why can I not iterate over the lines of a file after counting them?](#)

Hot Network Questions

- [TV program where real actors interact with 2-Dimensional characters](#)
- [Schottky OR-ing diodes passing reverse voltage; is it a faulty diode?](#)
- [Can a 10*10 square be paved with 1*4 rectangular stone plates?](#)
- [Preservation of universal sentences under substructures in higher-order logic](#)
- [Statistical testing for bimodal/multimodal sample](#)
- [How can I determine whether a complex rational equation has real solutions?](#)
- [Critiques of the real vs. nominal definition distinction](#)
- [Why Select Within Distance set to 1m selects every waterbody?](#)
- [Ping sweep of IP subnets](#)
- [How to avoid an undesired "nil" in defmacro](#)
- [Finding real variable where a 2x2 complex matrix is singular in Mathematica](#)
- [How do we define time-ordering operations in QFT?](#)
- [King defense setups](#)
- ["Creatures your opponents control" after the ability is activated](#)
- ["Imagine if..." vs. "Imagine..."](#)
- [How to protect 100-year-old wooden floors for the next 100 years?](#)
- [Documentation for DUSHIN software](#)
- [is it possible to keep the space when using url in latex](#)
- [variable name change causes segfault](#)
- [What do the terms postponement, slip and scrub mean in the context of the space shuttle program?](#)
- [2 ways of resolution of vectors](#)
- [Could a world tree have different biomes?](#)
- [Explain this seeming contradiction in Euclid Book 1 Proposition 16](#)

- [Is it possible to use LaTeX to create RPG-related PDFs, such as character sheets or adventure modules?](#)

[more hot questions](#)

[Question feed](#)

Subscribe to RSS

[Question feed](#)

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

[Stack Overflow](#)

- [Questions](#)
- [Help](#)
- [Chat](#)

[Business](#)

- [Stack Internal](#)
- [Stack Data Licensing](#)
- [Stack Ads](#)

[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](#)