How do you push values to a vec inside an enum stuct in Rust?

Ask Question

Asked 2 years, 9 months ago

Modified 2 years, 8 months ago

Viewed 364 times

6

How do you push values to a vec inside an enum stuct in Rust?

I'm trying to figure out how to push values to a vec inside an enum that's defined as a struct.

Here's the setup along with some of the stuff I tried:

```
enum Widget {
   Alfa { strings: Vec<String> },
}

fn main() {
   let wa = Widget::Alfa { strings: vec![] };

   // wa.strings.push("a".to_string());
   // no field `strings` on type `Widget`

   // wa.Alfa.strings.push("a".to_string());
   // no field `Alfa` on type `Widget`

   // wa.alfa.strings.push("a".to_string());
   // no field `alfa` on type `Widget`

   // wa.Widget::Alfa.strings.push("a".to_string());
   // expected one of `(`, `.`, `;`, `?`, `}`, or an operator, found `::`

   // wa["strings"].push("a".to_string());
   // cannot index into a value of type `Widget`
}
```

Is it possible to update a vec in an emum after it's been created? If so how does one go about that?

(NOTE: It was suggested that this is a duplicate of How do you access enum values in Rust?. I looked at that but it didn't address my question. It addresses how to access values, not how to update them. The two things are related, but the solution in the other answer for accessing does not accommodate updating.)

- rust

Share

Improve this question

Follow

edited Mar 14, 2023 at 19:08

asked Mar 10, 2023 at 21:18

Alan W. Smith

25.6k55 gold badges7373 silver badges101101 bronze badges

1

1

You'll need to use `match` (or `if let`) so the compiler knows which variant is considered.

PitaJ – PitaJ

2023-03-10 21:33:54 +00:00

Commented Mar 10, 2023 at 21:33

Add a comment |

2 Answers

Sorted by: Reset to default

Highest score (default) Trending (recent votes count more) Date modified (newest first) Date created (oldest first)

7

You can't access the fields on an enum variant directly, because the compiler only knows that the value is of the enum type (`Widget`), not which variant of the enum it has. You have to destructure the enum, for example with a `match`:

```
let mut wa = Widget::Alfa { strings: vec![] };
```

```
match &mut wa {
   Widget::Alfa { strings /*: &mut Vec<String> */ } => {
       strings.push("a".to_string());
   }

   // if the enum has more variants, you must have branches for these as well.
   // if you only care about `Widget::Alfa`, a wildcard branch like this is often a
   // good choice.
   _ => unreachable!(), // panics if ever reached, which we know in this case it won't
                        // because we just assigned `wa` before the `match`.
}
```

Alternatively you can use `if let` instead:

```
let mut wa = Widget::Alfa { strings: vec![] };

if let Widget::Alfa { strings } = &mut wa {
   strings.push("a".to_string());
} else {
   // some other variant than `Widget::Alfa`, equivalent to the wildcard branch
   // of the `match`. you can omit this, which would just do nothing
   // if it doesn't match.
   unreachable!()
}
```

Share

Improve this answer

Follow

answered Mar 10, 2023 at 21:34

Freyja

41.3k99 gold badges8989 silver badges130130 bronze badges

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

Comments

Add a comment

1

You can do something like this if you have one matching arm (which has no sense):

```
#[derive(Debug)]
enum Widget {
   Alfa { strings: Vec<String> },
}

fn main() {
   let mut wa = Widget::Alfa { strings: vec![] };

   let Widget::Alfa { strings } = &mut wa;

   strings.push("X".to_string());
   strings.push("Y".to_string());

   println!("{:?}", wa);
}
```

Or by using `match` (`if let`):

```
#[derive(Debug)]
enum Widget {
   Alfa { strings: Vec<String> },
   Beta { string: Vec<String> }
}

fn main() {
   let mut wa = Widget::Alfa { strings: vec![] };

   if let Widget::Alfa { strings } = &mut wa {
```

```
        strings.push("X".to_string());
        strings.push("Y".to_string());
    }

    println!("{:?}", wa);
}
```

Share

Improve this answer

Follow

edited Mar 10, 2023 at 21:47

answered Mar 10, 2023 at 21:36

■or■e Zelji■

1,82522 gold badges1616 silver badges3434 bronze badges

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

- 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

Linked

109
How do you access enum values in Rust?

Related

560
How do you disable dead code warnings at the crate level in Rust?
509
How do I print the type of a variable in Rust?
408
How do I split a string in Rust?
109
How do you access enum values in Rust?
0
How to transform a tiff file to ndarray in rust?
0
Rust proc_macro_derive (with syn crate) generating enum variant for matching
2
How do I access "private" struct's field from other files in Rust?
1
How do you make a non-hashable, C-like enum from a library hashable?
169
How do I get a slice of a Vec<T> in Rust?
0
Strongly typed enum as union in rust: How to determine type of value and retrieve it; How to do "constructors"

Hot Network Questions

- Storm Sphere and Obstruction
- Polygons (and some stars)
- The first Pontrjagin class of Milnor's example of exotic 7-sphere
- Displaying features in a layer of geometrytype = GeometryCollection25D
- Download is performed unsandboxed as root - Why am I getting this message, and how do I fix it?
- What is the difference between an ultraviolet limit and ultraviolet stability?
- "Imagine if..." vs. "Imagine..."
- Is the phrase "murder will out" from Chaucer?
- Could a world tree have different biomes?

- Why does Blender import PLY vertex colors incorrectly (89 → 0.1 instead of 0.349)?
- Looking for a proper formal substitute for "quick fix" for formal letter
- Proportional odds logistic regression for ordered category outcome - how to convert odds ratios to probabilities in this case?
- Generating Irregular, Separated Polygonal Cells along a Curve in Geometry Nodes
- Is "We will review application starting from..." a deadline?
- Can enemies damage each other in Breath of the Wild?
- Misunderstanding the lorentz transformation
- EEPROM Decoupling Capacitor
- A hilarious parody of Western movies where an entire town collapses into tunnels
- Is there a term for forming emotional or relational bonds with AI chatbots? If not, is “cyberpomorphic” a valid neologism?
- Best Welding Method to Fix a Damaged Steel Surly Frame
- How to check if two tables are identical?
- How do Buddhists interpret the Buddha’s explanation of earthquakes in AN 8.70?
- "Creatures your opponents control" after the ability is activated
- Zero Inflated Beta Regression (or not)?

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.

- LinkedIn
- Instagram