Consecutively run a list/queue of async tasks

help

Arwenac July 24, 2025, 8:30am 1

Hi there,

To preface: I usually work in embedded C, and a little C++ so learning rust is quite a nice exercise! And I am learning about async functions by coding a program for a Artifacts MMO bot. I am having a lot of fun but am also a little lost.

I am trying to do a list of reqwest post/api calls (which are all async functions) and handle the result data one after the other.
For this I created a struct `Player` which will be updated by data received from these reqwest calls.

I am now using tokio::spawn to have two threads, which each handle their own data/player struct.
They will never have to share data but will have to run at the same time.

```
    let mut player1 = Player::new("NAME1");
   let mut player2 = Player::new("NAME2");

   if !player1.init_player().await {
       return;
   }
   if !player2.init_player().await {
       return;
   }

   let handler1 =
       tokio::spawn(handle_1(player1));
   let handler2 =
       tokio::spawn(handle_2(player2));
```

And the handle functions is something like

```
async fn handle_1(mut player: Player) {
   let player_mutex = Arc::new(Mutex::new(player));
  do_and_handle_api_call1(player_mutex.clone(), some_data).await;
  do_and_handle_api_call2(player_mutex.clone(), some_data).await;
}
```

This is working fine for now. But I would like to edit my code to have to list/vector/queue for each Player struct I can add these calls/tasks to.
And then have the program wait until there are tasks to execute, do them and go back to sleep and wait until new tasks are added. For this I have found std::sync::Condvar but I am not sure I am going in the right direction.

For context: Later I will use a gui/CLI (like ratatui or egui or a simple CLI) to add tasks to this queue but for now it is fine to add them all in one go in the main.

So now my question:
Could someone help me get into the right direction with code examples or reading material for managing this kind of queue?
Should I use a vector, FuturesOrdered or mpsc?

Is there some kind of crate for this? I am reading a lot about tokio async and futures and async/atomics but getting a bit overwhelmed and could use some directions on where to go and what route to take.

1 Like

jofas July 24, 2025, 11:36am 2

Sounds to me like your `Player` struct should be an actor, right? Alice has a blog post on how to write basic actors with tokio:

ryhl.io

Actors with Tokio – Alice Ryhl

This article is about building actors with Tokio directly, without using any actor libraries such as Actix. This turns out to be rather easy to do, however there are some details you should be aware of:

1 Like

Arwenac July 25, 2025, 6:19am 3

Thank you, I think that design pattern is exactly what I needed! Will experiment more with it.
I already got a little test program out of it.

Now going to figure out a way to implement it in my already existing crate

1 Like

system Closed October 23, 2025, 6:19am 4

This topic was automatically closed 90 days after the last reply. We invite you to open a new topic if you have further questions or comments.

Related topics

Topic	Replies	Views	Activity
Execute async code every milliseconds then create a future help	3	874	April 3, 2021
How to handle a vector of async function pointers	10	14009	July 12, 2020
Write threaded async code with least number of crates help	4	642	June 30, 2021
Asynchronous queue for use with tokio (or: how to scrape tile servers with tokio?) help	4	7810	August 17, 2019
Tokio execute async for loop concurrently	10	5782	September 1, 2020