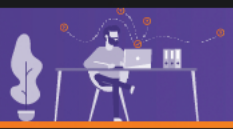
Home
PUBLIC
Stack Overflow
Tags
Users
Jobs
TEAMS What's this?
First 25 Users Free

Return list from async/await method

Ask Question

Asked 5 years, 3 months ago Active 4 months ago Viewed 73k times

Stack Overflow saves developers up to **90 minutes per week** on average.

stackoverflow FOR TEAMS
START SAVING TIME

25

3

I want to make a webservice request asynchron. I call it here:

```
List<Item> list = GetListAsync();
```

Here is the declaration of my function, which should return a list:

```
private async Task<List<Item>> GetListAsync(){
    List<Item> list = await Task.Run(() => manager.GetList());
    return list;
}
```

If I want to compile I get the following error

```
Cannot implicitely convert type System.Threading.Tasks.Task<System.Collections.Generic.List<I
```

As I know If I use the `async` modifier the result is automatically wrapped with Task. I think this doesn't happen because I use `Task.Run`. If I remove the `Task.Run(() =>` part I get

> Cannot await System.Collections.Generic.List expression

I think I haven't fully understood the async/await methods. What I'm doing wrong?

c# list asynchronous async-await task

share improve this question

asked Sep 8 '14 at 9:23

testing
15.9k ● 31 ● 178 ● 335

possible duplicate of Cannot implicitly convert type from Task<> – i3arnon Sep 8 '14 at 20:21

add a comment

4 Answers

active oldest votes

53

You need to correct your code to wait for the list to be downloaded:

```
List<Item> list = await GetListAsync();
```

Also, make sure that the method, where this code is located, has `async` modifier.

The reason why you get this error is that `GetListAsync` method returns a `Task<T>` which is not a

Blog
What senior developers can learn from beginners
WebSockets for fun and profit
Featured on Meta
We're lowering the close/reopen vote threshold from 5 to 3 for good
Why was I just awarded a bunch of "Announcer" badges?

Remote jobs

Front-End Developer (Remote)
Wallethub No office location
REMOTE
angular javascript

Bored of working on CRUD apps? Get paid to make hyper-casual web games!
Koji - Web App Development Platform
No office location
REMOTE
javascript html5

Full Stack JavaScript Developer - Junior
GitStart No office location
$14K - $36K REMOTE
typescript node.js

Senior Vue.js Developer (Remote)
X-Team No office location
REMOTE
angular javascript

Senior Java Engineer
Grassroots Technologies, Inc New York, NY
$50K - $65K REMOTE
java javascript

View more jobs on Stack Overflow

Linked

completed result. As your list is downloaded asynchronously (because of `Task.Run()`) you need to "extract" the value from the task using the `await` keyword.

If you remove `Task.Run()`, you list will be downloaded synchronously and you don't need to use `Task`, `async` or `await`.

One more suggestion: you don't need to await in `GetListAsync` method if the only thing you do is just delegating the operation to a different thread, so you can shorten your code to the following:

```
private Task<List<Item>> GetListAsync(){
    return Task.Run(() => manager.GetList());
}
```

share improve this answer

edited Sep 8 '14 at 9:56

answered Sep 8 '14 at 9:24
takemyoxygen
3,411 ● 17 ● 19

They will also need to ensure that the method calling this code is made an `async` method (if they haven't already). – Daniel Kelley Sep 8 '14 at 9:26

What I don't get is why do I have to `await` two times? – testing Sep 8 '14 at 9:27

@testing, you have to await twice because you have 2 async calls – Ben Robinson Sep 8 '14 at 9:28

@BenRobinson: Do I have two async calls or only one (and the second is for "extracting")? – testing Sep 8 '14 at 9:33

1 @testing You have 2, you are awaiting the task in you `GetListAsync` method, then you are awaiting the method itself in your calling code – Ben Robinson Sep 8 '14 at 9:35

add a comment

10

In addition to @takemyoxygen's answer the convention of having a function name that ends in `Async` is that this function is truly asynchronous. I.e. it does not start a new thread and it doesn't simply call `Task.Run`. If that is all the code that is in your function, it will be better to remove it completely and simply have:

```
List<Item> list = await Task.Run(() => manager.GetList());
```

share improve this answer

answered Sep 8 '14 at 11:30
Ned Stoyanov
11.4k ● 4 ● 26 ● 53

I also have a try/catch block to catch the exceptions and I'm showing a spinner. So my method name is correct than? – testing Sep 8 '14 at 11:45

yes i think it should be OK, starting a new thread is not ideal but i think the guideline is more about calling a synchronous version of the function inside `Task.Run` – Ned Stoyanov Sep 9 '14 at 9:35

add a comment

Home
PUBLIC
Stack Overflow
Tags
Users
Jobs
TEAMS What's this?

3

Works for me:

```
List<Item> list = Task.Run(() => manager.GetList()).Result;
```

in this way it is not necessary to mark the method with async in the call.

share improve this answer

answered Jun 25 at 23:59
sergiokml
31 ● 3

63 Cannot implicitly convert type from Task<>
2 is it possible to convert Task<List<>> to List<>?
1 How do I asynchronously enumerate a collection in an ASP.NET environment?

Related

3235 How do I check if a list is empty?
2924 Finding the index of an item given a list containing it in Python
3119 What is the difference between Python's list methods append and extend?
3003 How to make a flat list out of list of lists?
2294 How do I concatenate two lists in Python?
2328 How to clone or copy a list?
5249 How do I return the response from an asynchronous call?
993 How and when to use 'async' and 'await'
290 Do you have to put Task.Run in a method to make it async?
897 Using async/await with a forEach loop

Hot Network Questions

Even and Odd Numbers
Ubuntu: show what keys are pressed in real-time
How does the Starlink satellite shine?
Which font's digits resemble Nixie tube digits?
How to review a manuscript where the software is difficult to install?
Does ascending one step burn 0.1 kcal of energy?
How did the Telstar 525/405-line conversion system from 1962 work?
Get Object Label from API Name in LWC using wire
Sorting a large list of test scores
How do airlines compute weight allowances for carry-on bags?
How are short passwords not safe on the web?
What are the round things near the main front cog?
How unique should articles be for better Google SEO
Unit Circle and Circle of Radius R
How to correctly say Star Wars in Latin?
Integer Interpretator
How does one tell (or conclude) if a substance is

First 25 Users Free

add a comment

-1

Instead of doing all these, one can simply use ".Result" to get the result from a particular task.

eg: List list = GetListAsync().Result;

Which as per the definition => Gets the result value of this Task < TResult >

share improve this answer

answered Jul 31 at 10:56

avinava basu

67 • 2

add a comment

Your Answer

B I

Sign up or log in

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

Post as a guest

Name

Email

Required, but never shown

Post Your Answer

By clicking "Post Your Answer", you agree to our terms of service, privacy policy and cookie policy

Not the answer you're looking for? Browse other questions tagged c# list asynchronous async-await task or ask your own question.

How does one tell (or conclude) if a substance is carcinogenic?

How long would a medium sized city last if transplanted to another world?

meaning of "claim" in "claim asylum"

Idioms for saying when one's bad situation compounded with more problems

Are there statements from IBM insiders revealing why OS/2 was dropped?

If an american is attacked or discriminated by his status as american, is it protected by anti discrimination laws?

Headset labeling - Does this qualify as EC30?

Phone number with different formats lookup

Question feed

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS What's this?

First 25 Users Free

STACK OVERFLOW

Questions

Jobs

PRODUCTS

Teams

Talent

COMPANY

About

Press

STACK EXCHANGE NETWORK

Technology

Blog Facebook Twitter LinkedIn

Developer Jobs Directory
Salary Calculator
Help
Mobile

Advertising
Enterprise

Work Here
Legal
Privacy Policy
Contact Us

Life / Arts
Culture / Recreation
Science
Other

site design / logo © 2019 Stack Exchange Inc; user contributions licensed under cc by-sa 4.0 with attribution required

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.