

+future sodom+ std future cppreference std future<T> get cppreference What is a Future and how do I use it? Stack.

â â â â â Rating: 5 (8.395.244 reviews) - Free • Future • Access

Original URL: <https://tools.orientwatchusa.com/future-sodom.pdf>

Mar 12 2024 The class template `std::future` provides a mechanism to access the result of asynchronous operations. An asynchronous operation created via `std::async`, `std::packaged_task` or `std::promise` can provide a `std::future` object to the creator of that asynchronous operation.

The creator of the asynchronous operation can then use a variety of methods to query, wait for, or extract a value from the `std::future`. Feb 22 2024 The `get` member function waits by calling `wait` until the shared state is ready, then retrieves the value stored in the shared state, if any.

Right after calling this function, `valid` is `false`. If `valid` is `false` before the call to this function, the behavior is undefined. Jul 21 2020 A `future` represents the result of an asynchronous operation and can have two states: `uncompleted` or `completed`.

Most likely, as you aren't doing this just for fun, you actually need the results of that `Future<T>` to progress in your application. You need to display the number from the database or the list of movies found. Aug 27 2021 Checks if `theFuture` refers to a shared state.

This is the case only for futures that were not default-constructed or moved from, i.e., returned by `std::promise::get_future()`, `std::packaged_task::get_future()` or `std::async` until the first time `get` or `share` is called.

The behavior is undefined if any member function other than the destructor, the move assignment operator, or `valid` is called. Mar 19 2025 Specifies state of a `future` as returned by `wait_for` and `wait_until` functions of `std::future` and `std::shared_future`.

Constants Aug 2 2020 If `theFuture` is the result of a call to `async` that used lazy evaluation, this function returns immediately without waiting. The behavior is undefined if `valid` is `false` before the call to this function or `Clock` does not meet the `Clock` requirements.

The program is ill-formed if `std::chrono::is_clock_v<Clock>` is `false`.

since C++20 Nov 27 2023 `future` `const future = delete` `~future` `future operator =` `const future = delete` `future operator =` `future` `noexcept` `shared_future<R>` `share` `noexcept` retrieving the value * see description * `get` functions to check state `bool valid` `const noexcept void wait` `const template<class Rep, class Period>` Mar 6 2020 `impl<F>Future` for `Box<F>` where `F` `Unpin` `+Future+` ? `Sized` `Boxed` futures only implement the `Future` trait when `theFuture` inside the `Box` implements `Unpin`.

Since your function doesn't guarantee that the returned `future` implements `Unpin`, your return value will be considered to not implement `Future`. You'll not be able to await it because your type is basically not a `Future`.

Related Links:

1. <<six stud swingathon>> Six Degrees Zotero 6 Zotero R GTA 6 2026 5 26 ...
2. #???? ???? ??????# KRISHAN kahit kayo nalang Krishna RONALD DELOS SANT...
3. +young and curious 4+ Empowering young people is future proofing busin...
4. #naty delgado# Naty Certified Safe. Made For Skin. Built For Trust Ama...
5. +double lives+ c float vs. double precision Stack Overflow What is the...
6. +hot rod chicks 1+ Outlook free personal email and calendar from Micro...
7. <<oil overload 7>> Crude Oil Prices Today OilPrice Crude Oil News Page...
8. <<disney stars naked>> Disney The official home for all thingsDisney D...
9. <craving anal 3> Salt craving A symptom of Addison s disease? Mayo Cli...
10. <peachycm erome> YouTube Help Google Help Download the YouTube mobile ...