

This PDF is generated from: <https://www.extremeweekend.pl/Wed-14-Feb-2024-29594.html>

Title: Future trends of energy storage in Armenia

Generated on: 2026-09-15 07:27:38

Copyright (C) 2026 EXTREME POWER. All rights reserved.

For the latest updates and more information, visit our website: <https://www.extremeweekend.pl>

`future (const future & ) = delete; ~future (); future & operator =(const future & ) = delete; future & operator =(future & & ) noexcept; shared_future &R> share () noexcept; // ...`

The class template `std::future` provides a mechanism to access the result of asynchronous operations: An asynchronous operation (created via `std::async`, ...

If the future 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 ...

If the future is the result of a call to `std::async` that used lazy evaluation, this function returns immediately without waiting. This function may block for longer than ...

Transfers the shared state of `*this`, if any, to a `std::shared_future` object. Multiple `std::shared_future` objects may reference the same shared state, which is not possible with ...

Checks if the future 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 ()`, ...

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 ...`

Specifies state of a future as returned by `wait_for` and `wait_until` functions of `std::future` and `std::shared_future`. Constants

The error: `SyntaxError: future feature annotations is not defined` usually related to an old version of python,

but my remote server has Python3.9 and to verify it - I also added it ...

Unlike `std::future`, which is only moveable (so only one instance can refer to any particular asynchronous result), `std::shared_future` is copyable and multiple shared future ...

Web: <https://www.extremeweekend.pl>

