

Title: The future of energy storage cabinet

Generated on: 2026-06-05 14:30:18

Copyright (C) 2026 MARZENIA SOLAR SOLUTIONS. All rights reserved.

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

-----

However, this is many years in the future, giving affected decorators plenty of time to update their code. Make the future import a no-op in the future: Instead of eventually making from ...

Blocks until the result becomes available. `valid() == true` after the call. The behavior is undefined if `valid() == false` before the call to this function.

The issue here is that the `future = m.make_future_dataframe` method creates a dataset future where the only column is the `ds` date column. In order to predict using a model with regressors ...

```
future (const future & ) = delete; ~future (); future & operator =(const future & ) = delete; future & operator =(future & & ) noexcept; shared_future &lt;R&gt; share () noexcept; // retrieving the value ...
```

In this case it does work. In general, it probably doesn't. I'm wondering how this break in backwards compatibility should in general be navigated. Perhaps installing a previous version of ...

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. ...

`impl&lt;F&gt;` Future for `Box&lt;F&gt;`; where `F: Unpin + Future + ?Sized`, `Boxed` futures only implement the `Future` trait when the future inside the `Box` implements `Unpin`. Since your function ...

`wait_until` waits for a result to become available. It blocks until specified `timeout_time` has been reached or the result becomes available, whichever comes first. The return value indicates why ...

The promise is the `&quot;push&quot;` end of the promise-future communication channel: the operation that stores a value in the shared state synchronizes-with (as defined in `std::memory_order`) ...

The class template `std::future` provides a mechanism to access the result of asynchronous operations: An



# The future of energy storage cabinet

asynchronous operation (created via std::async, std::packaged\_task, ...

Web: <https://www.malemarzenia.com.pl>

