Extending

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst, line 1)

Unknown directive type "currentmodule".

```
.. currentmodule:: asyncio
```

The main direction for :mod:`asyncio` extending is writing custom *event loop* classes. Asyncio has helpers that could be used to simplify this task.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst, line 8); *backlink*

Unknown interpreted text role "mod".

Note

Third-parties should reuse existing asyncio code with caution, a new Python version is free to break backward compatibility in *internal* part of API.

Writing a Custom Event Loop

:class:`asyncio.AbstractEventLoop` declares very many methods. Implementing all them from scratch is a tedious job.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst, line 21); *backlink*

Unknown interpreted text role "class".

A loop can get many common methods implementation for free by inheriting from :class:`asyncio.BaseEventLoop`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst, line 24); *backlink*

Unknown interpreted text role "class".

In turn, the successor should implement a bunch of *private* methods declared but not implemented in :class:`asyncio.BaseEventLoop`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst, line 27); *backlink*

Unknown interpreted text role "class".

For example, `loop.create_connection()` checks arguments, resolves DNS addresses, and calls `loop._make_socket_transport()` that should be implemented by inherited class. The `_make_socket_transport()` method is not documented and is considered as an *internal* API.

Future and Task private constructors

:class:`asyncio.Future` and :class:`asyncio.Task` should be never created directly, please use corresponding :meth:`loop.create_future` and :meth:`loop.create_task`, or :func:`asyncio.create_task` factories instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst, line 40); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst, line 40); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst`, line 40); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst`, line 40); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst`, line 40); *backlink*

Unknown interpreted text role "func".

However, third-party *event loops* may *reuse* built-in future and task implementations for the sake of getting a complex and highly optimized code for free.

For this purpose the following, *private* constructors are listed:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst`, line 49)

Unknown directive type "method".

```
.. method:: Future.__init__(*, loop=None)

   Create a built-in future instance.

   *loop* is an optional event loop instance.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst`, line 55)

Unknown directive type "method".

```
.. method:: Task.__init__(coro, *, loop=None, name=None, context=None)

   Create a built-in task instance.

   *loop* is an optional event loop instance. The rest of arguments are described in
   :meth:`loop.create_task` description.

   .. versionchanged:: 3.11

      *context* argument is added.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]asyncio-extending.rst`, line 66)

Unknown directive type "method".

```
.. method:: Tasl._check_future(future)

   Return ``True`` if *future* is attached to the same loop as the task, ``False``
   otherwise.

   .. versionadded:: 3.11
```

Task lifetime support

A third party task implementation should call the following functions to keep a task visible by :func:`asyncio.get_tasks` and :func:`asyncio.current_task`:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\[cpython-main][Doc][library]asyncio-extending.rst, line 77); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]asyncio-extending.rst, line 77); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]asyncio-extending.rst, line 80)

Unknown directive type "function".

```
.. function:: _register_task(task)

   Register a new *task* as managed by *asyncio*.

   Call the function from a task constructor.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]asyncio-extending.rst, line 86)

Unknown directive type "function".

```
.. function:: _unregister_task(task)

   Unregister a *task* from *asyncio* internal structures.

   The function should be called when a task is about to finish.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]asyncio-extending.rst, line 92)

Unknown directive type "function".

```
.. function:: _enter_task(loop, task)

   Switch the current task to the *task* argument.

   Call the function just before executing a portion of embedded *coroutine*
   (:meth:`coroutine.send` or :meth:`coroutine.throw`).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]asyncio-extending.rst, line 99)

Unknown directive type "function".

```
.. function:: _leave_task(loop, task)

   Switch the current task back from *task* to ``None``.

   Call the function just after :meth:`coroutine.send` or :meth:`coroutine.throw`
   execution.
```