## :c:type:`uv async t` --- Async handle

System Message: ERROR/3 (D:\onboarding-resources\sa aster\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src)async.rst, line 4); *backlink* 

Unknown interpreted text role "c:type".

Async handles allow the user to "wakeup" the event loop and get a callback called from another thread.

## Data types

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\node-
master\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) async.rst, line 14)
```

Unknown directive type "c:type".

```
.. c:type:: uv_async_t
   Async handle type.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\nodemaster\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) async.rst, line 18)

Unknown directive type "c:type".

```
.. c:type:: void (*uv async cb) (uv async t* handle)
   Type definition for callback passed to :c:func:`uv_async_init`.
```

## **Public members**

N/A

 $System\,Message:\,ERROR/3\,(\texttt{D:}\noboarding-resources}\noboarding-resources\\\noboarding$ master\deps\uv\docs\src\ (node-master) (deps) (uv) (docs) (src) async.rst, line 28)

Unknown directive type "seealso".

.. seealso:: The :c:type:`uv\_handle\_t` members also apply.

## API

 $System\,Message:\,ERROR/3\, (\hbox{\tt D:\noboarding-resources}) sample-onboarding-resources \verb|\noboarding-resources|| and the sample-onboarding-resources|| and the sample-onboarding-resources|$ aster\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src)async.rst, line 34)

Unknown directive type "c:function".

```
.. c:function:: int uv_async_init(uv_loop_t* loop, uv_async_t* async, uv_async_cb)
   Initialize the handle. A NULL callback is allowed.
   :returns: 0 on success, or an error code < 0 on failure.
       Unlike other handle initialization functions, it immediately starts the handle.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\nodemaster\deps\uv\docs\src\(node-master) (deps) (uv) (docs) (src) async.rst, line 43)

Unknown directive type "c:function".

```
.. c:function:: int uv_async_send(uv_async_t* async)
   Wake up the event loop and call the async handle's callback.
   :returns: 0 on success, or an error code < 0 on failure.
        It's safe to call this function from any thread. The callback will be called on the
       loop thread.
        c:func:`uv async send` is `async-signal-safe <https://man7.org/linux/man-pages/man7/signal-safety.
       It's safe to call this function from a signal handler.
   .. warning::
       libuv will coalesce calls to :c:func:`uv async send`, that is, not every call to it will
       yield an execution of the callback. For example: if :c:func:`uv_async_send` is called 5
```

times in a row before the callback is called, the callback will only be called once. If :c:func:`uv\_async\_send` is called again after the callback was called, it will be called again.

Unknown directive type "seealso".

.. seealso::
 The :c:type:`uv\_handle\_t` API functions also apply.