

Class: Pool

Extends: `undici.Dispatcher`

A pool of `Client` instances connected to the same upstream target.

Requests are not guaranteed to be dispatched in order of invocation.

new Pool(url[, options])

Arguments:

- **url** `URL | string` - It should only include the **protocol**, **hostname**, and **port**.
- **options** `PoolOptions` (optional)

Parameter: PoolOptions

Extends: `ClientOptions`

- **factory** `(origin: URL, opts: Object) => Dispatcher` - Default: `(origin, opts) => new Client(origin, opts)`
- **connections** `number | null` (optional) - Default: `null` - The number of `Client` instances to create. When set to `null`, the `Pool` instance will create an unlimited amount of `Client` instances.

Instance Properties

Pool.closed

Implements `Client.closed`

Pool.destroyed

Implements `Client.destroyed`

Pool.stats

Returns `PoolStats` instance for this pool.

Instance Methods

Pool.close([callback])

Implements `Dispatcher.close([callback])`.

Pool.destroy([error, callback])

Implements `Dispatcher.destroy([error, callback])`.

Pool.connect(options[, callback])

See Dispatcher.connect(options[, callback]).

Pool.dispatch(options, handler)

Implements Dispatcher.dispatch(options, handler).

Pool.pipeline(options, handler)

See Dispatcher.pipeline(options, handler).

Pool.request(options[, callback])

See Dispatcher.request(options [, callback]).

Pool.stream(options, factory[, callback])

See Dispatcher.stream(options, factory[, callback]).

Pool.upgrade(options[, callback])

See Dispatcher.upgrade(options[, callback]).

Instance Events

Event: 'connect'

See Dispatcher Event: 'connect'.

Event: 'disconnect'

See Dispatcher Event: 'disconnect'.

Event: 'drain'

See Dispatcher Event: 'drain'.