Class: BalancedPool

Extends: `undici.Dispatcher`

A pool of Pool instances connected to multiple upstreams.

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

`new BalancedPool(upstreams [, options])`

Arguments:

- **upstreams** `URL | string | string[]` - It should only include the **protocol, hostname, and port**.
- **options** `BalancedPoolOptions` (optional)

Parameter: `BalancedPoolOptions`

Extends: `PoolOptions`

- **factory** `(origin: URL, opts: Object) => Dispatcher` - Default: `(origin, opts) => new Pool(origin, opts)`

The `PoolOptions` are passed to each of the `Pool` instances being created. ## Instance Properties

`BalancedPool.upstreams`

Returns an array of upstreams that were previously added.

`BalancedPool.closed`

Implements Client.closed

`BalancedPool.destroyed`

Implements Client.destroyed

`Pool.stats`

Returns `PoolStats` instance for this pool.

Instance Methods

`BalancedPool.addUpstream(upstream)`

Add an upstream.

Arguments:

- **upstream** `string` - It should only include the **protocol, hostname, and port**.

`BalancedPool.removeUpstream(upstream)`

Removes an upstream that was previously addded.

`BalancedPool.close([callback])`

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

`BalancedPool.destroy([error, callback])`

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

`BalancedPool.connect(options[, callback])`

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

`BalancedPool.dispatch(options, handlers)`

Implements `Dispatcher.dispatch(options, handlers)`.

`BalancedPool.pipeline(options, handler)`

See `Dispatcher.pipeline(options, handler)`.

`BalancedPool.request(options[, callback])`

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

`BalancedPool.stream(options, factory[, callback])`

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

`BalancedPool.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'`.