# proxy-addr

[NPM Version](https://npmjs.org/package/proxy-addr) [NPM Downloads](https://npmjs.org/package/proxy-addr) [Node.js Version](https://nodejs.org/en/download) [Build Status](https://github.com/jshttp/proxy-addr/actions?query=workflow%3Aci) [Test Coverage](https://coveralls.io/r/jshttp/proxy-addr?branch=master)

Determine address of proxied request

## Install

This is a [Node.js](https://nodejs.org/en/) module available through the [npm registry](https://www.npmjs.com/). Installation is done using the [npm install command](https://docs.npmjs.com/getting-started/installing-npm-packages-locally):

$ npm install proxy-addr

## API

var proxyaddr = require('proxy-addr')

### proxyaddr(req, trust)

Return the address of the request, using the given trust parameter.

The trust argument is a function that returns true if you trust the address, false if you don't. The closest untrusted address is returned.

proxyaddr(req, function (addr) { return addr === '127.0.0.1' })

proxyaddr(req, function (addr, i) { return i < 1 })

The trust arugment may also be a single IP address string or an array of trusted addresses, as plain IP addresses, CIDR-formatted strings, or IP/netmask strings.

proxyaddr(req, '127.0.0.1')

proxyaddr(req, ['127.0.0.0/8', '10.0.0.0/8'])

proxyaddr(req, ['127.0.0.0/255.0.0.0', '192.168.0.0/255.255.0.0'])

This module also supports IPv6. Your IPv6 addresses will be normalized automatically (i.e. fe80::00ed:1 equals fe80:0:0:0:0:0:ed:1).

proxyaddr(req, '::1')

proxyaddr(req, ['::1/128', 'fe80::/10'])

This module will automatically work with IPv4-mapped IPv6 addresses as well to support node.js in IPv6-only mode. This means that you do not have to specify both ::ffff:a00:1 and 10.0.0.1.

As a convenience, this module also takes certain pre-defined names in addition to IP addresses, which expand into IP addresses:

proxyaddr(req, 'loopback')

proxyaddr(req, ['loopback', 'fc00:ac:1ab5:fff::1/64'])

* loopback: IPv4 and IPv6 loopback addresses (like ::1 and 127.0.0.1).
* linklocal: IPv4 and IPv6 link-local addresses (like fe80::1:1:1:1 and 169.254.0.1).
* uniquelocal: IPv4 private addresses and IPv6 unique-local addresses (like fc00:ac:1ab5:fff::1 and 192.168.0.1).

When trust is specified as a function, it will be called for each address to determine if it is a trusted address. The function is given two arguments: addr and i, where addr is a string of the address to check and i is a number that represents the distance from the socket address.

### proxyaddr.all(req, [trust])

Return all the addresses of the request, optionally stopping at the first untrusted. This array is ordered from closest to furthest (i.e. arr[0] === req.connection.remoteAddress).

proxyaddr.all(req)

The optional trust argument takes the same arguments as trust does in proxyaddr(req, trust).

proxyaddr.all(req, 'loopback')

### proxyaddr.compile(val)

Compiles argument val into a trust function. This function takes the same arguments as trust does in proxyaddr(req, trust) and returns a function suitable for proxyaddr(req, trust).

var trust = proxyaddr.compile('loopback')

var addr = proxyaddr(req, trust)

This function is meant to be optimized for use against every request. It is recommend to compile a trust function up-front for the trusted configuration and pass that to proxyaddr(req, trust) for each request.

## Testing

$ npm test

## Benchmarks

$ npm run-script bench

## License

[MIT](http://license)