
proxy-lists



Node.js module for getting proxies from publicly available proxy lists. Support for more than two dozen different proxy lists. You can see the full list of proxy sources [here](#).

Missing a proxy list that you think should be supported? Open an issue to suggest it be added as a source. Or you can add a new source and create a pull request to have it added to this module.

- Installation
 - Update Geolp Database
- Command-line interface
- API
 - getProxies
 - ★ Options
 - ★ Proxy Object
 - getProxiesFromSource
 - ★ Options
 - addSource
 - listSources
 - ★ Options
- Usage with Proxy
- Contributing
 - Configure Local Environment
 - Tests
- Changelog
- License
- Funding

Installation

If you wish to use this module as a CLI tool, install it globally via npm:

```
1 npm install -g proxy-lists
```

Otherwise, you can add it to your existing node application like this:

```
1 npm install proxy-lists --save
```

This will install `proxy-lists` and add it to your application's `package.json` file.

Update GeoiP Database

This module uses `geoip-lite` to perform `geoip-country` lookups on IP addresses of proxies. The `geoip-lite` module ships with the free version of MaxMind's `geoip` database. This database stopped being directly included in the module due to a change on MaxMind's side - specifically with their end-user licensing agreements. So it is necessary for each end-user (that's you!) to create their own MaxMind account and then generate a license key.

If you are using this module inside another project (via the API), use the following command to update the `geoip` database:

```
1 npm run update:geoip-database license_key=YOUR_LICENSE_KEY
```

If you are using the CLI:

```
1 proxy-lists updateGeoIpData --license-key YOUR_LICENSE_KEY
```

Command-line interface

This section assumes that you have `proxy-lists` installed globally and that it is available on your current user's `PATH`.

To view the help screen for the CLI tool:

```
1 proxy-lists --help
```

To view the help screen for the `getProxies` command:

```
1 proxy-lists getProxies --help
```

To output the proxies in `.txt` format:

```
1 proxy-lists getProxies --output-format="txt"
```

To output proxies to `STDOUT`:

```
1 proxy-lists getProxies --stdout
```

To output proxies to a different file than `proxies.txt`:

```
1 proxy-lists getProxies --output-file="somefile.txt"
```

To get proxies from specific sources:

```
1 proxy-lists getProxies --sources-white-list="gatherproxy,sockslist"
```

To get proxies from specific countries:

```
1 proxy-lists getProxies --countries="us,ca"
```

To get proxies with specific protocols:

```
1 proxy-lists getProxies --protocols="http,https"
```

To get only anonymous and elite proxies:

```
1 proxy-lists getProxies --anonymity-levels="anonymous,elite"
```

The output of the `getProxies` command is written to a new file (`proxies.txt`) in your current working directory.

API

These are the public methods of the `ProxyLists` module that allow you to get proxies, add custom proxy sources, and list existing sources.

`getProxies`

`getProxies([options])`

Gets proxies from all available proxy lists.

Usage:

```
1 var ProxyLists = require('proxy-lists');
2
3 // `getProxies` returns an event emitter.
4 ProxyLists.getProxies({
5   // options
6   countries: ['us', 'ca']
7 })
8   .on('data', function(proxies) {
9     // Received some proxies.
10    console.log('got some proxies');
11    console.log(proxies);
12  })
```

```

13     .on('error', function(error) {
14         // Some error has occurred.
15         console.log('error!', error);
16     })
17     .once('end', function() {
18         // Done getting proxies.
19         console.log('end!');
20     });

```

Sample proxies:

```

1  [
2      {
3          ipAddress: '123.123.2.42',
4          port: 8080,
5          country: 'us',
6          source: 'superproxies'
7      },
8      {
9          ipAddress: '234.221.233.142',
10         port: 3128,
11         country: 'cz',
12         protocols: ['https'],
13         source: 'someproxysource'
14     },
15     {
16         ipAddress: '234.221.233.142',
17         port: 3128,
18         country: 'cz',
19         anonymityLevel: 'elite',
20         protocols: ['https'],
21         source: 'anotherproxysource'
22     }
23 ]

```

Options for getProxies Method

```

1  var options = {
2      /*
3          The filter mode determines how some options will be used to
4          exclude proxies.
5
6          For example if using this option `anonymityLevels: ['elite']`:
7          'strict' mode will only allow proxies that have the '
8              anonymityLevel' property equal to 'elite'; ie. proxies
              that are missing the 'anonymityLevel' property will be
              excluded.
9
10         'loose' mode will allow proxies that have the '
11             anonymityLevel' property of 'elite' as well as those
              that are missing the 'anonymityLevel' property.
12
13     */
14  }

```

```
9     filterMode: 'strict',
10
11     /*
12         Whether or not to emit only unique proxies (HOST:PORT).
13     */
14     unique: true,
15
16     /*
17         Get proxies for the specified countries.
18
19         To get all proxies, regardless of country, set this option to
20         NULL.
21
22         See:
23         https://en.wikipedia.org/wiki/ISO_3166-1
24
25         Only USA and Canada:
26         ['us', 'ca']
27     */
28     countries: null,
29
30     /*
31         Exclude proxies from the specified countries.
32
33         To exclude Germany and Great Britain:
34         ['de', 'gb']
35     */
36     countriesBlackList: null,
37
38     /*
39         Get proxies that use the specified protocols.
40
41         To get all proxies, regardless of protocol, set this option to
42         NULL.
43
44         To get proxies with specified protocols:
45         ['socks4', 'socks5']
46     */
47     protocols: null,
48
49     /*
50         Anonymity level.
51
52         To get all proxies, regardless of anonymity level, set this
53         option to NULL.
54
55         To get proxies with specified anonymity-levels:
56         ['elite', 'anonymous']
57     */
58     anonymityLevels: null,
```

```
57     /*
58         Include proxy sources by name.
59
60         Only 'freeproxylists':
61         ['freeproxylists']
62     */
63     sourcesWhiteList: null,
64
65     /*
66         Exclude proxy sources by name.
67
68         All proxy sources except 'freeproxylists':
69         ['freeproxylists']
70     */
71     sourcesBlackList: null,
72
73     /*
74         Full path to the sources directory.
75     */
76     sourcesDir: path.join(__dirname, 'sources'),
77
78     /*
79         Set to TRUE to have all asynchronous operations run in series.
80     */
81     series: false,
82
83     /*
84         Options to pass to puppeteer when creating a new browser
85         instance.
86     */
87     browser: {
88         headless: true,
89         slowMo: 0,
90         timeout: 10000,
91     },
92
93     /*
94         Default request module options. For example you could pass the
95         'proxy' option in this way.
96
97         See for more info:
98         https://github.com/request/request#requestdefaultsoptions
99     */
100     defaultRequestOptions: null,
101
102     /*
103         Use a queue to limit the number of simultaneous HTTP requests.
104     */
105     requestQueue: {
106         /*
107             The maximum number of simultaneous requests.
```

```
106      */
107      concurrency: 1,
108      /*
109         The time (in milliseconds) between each request. Set to 0
           for no delay.
110      */
111      delay: 0,
112    },
113  };
```

Proxy Object The proxy object has the following properties: * **ipAddress** - *string* The IP address of the proxy. * **port** - *integer* The port number of the proxy. * **country** - *string* Alpha-2 country code of the country in which the proxy is geo-located. * **source** - *string* The name of the proxy list from which the proxy was gathered. * **protocols** - *optional array* An array of protocols that the proxy supports. May contain one or more of the following: * **http** - The proxy uses HTTP. * **https** - The proxy uses HTTPS. * **socks5** - The proxy server uses the socks5 protocol. * **socks4** - The proxy server uses the socks4 protocol. * **anonymityLevel** - *optional string* The anonymity level of the proxy. Can be any one of the following: * **transparent** - The proxy does not hide the requester's IP address. * **anonymous** - The proxy hides the requester's IP address, but adds headers to the forwarded request that make it clear that the request was made using a proxy. * **elite** - The proxy hides the requester's IP address and does not add any proxy-related headers to the request.

The attributes marked as *optional* above might not be given for all proxies. Some proxy lists are missing this information.

It's important to note that this module does **NOT** verify all of the information provided by the proxy lists from which the proxies are gathered. If you need to check that proxies work, verify their anonymity level, whether or not they support tunneling; use proxy-verifier.

getProxiesFromSource

`getProxiesFromSource(name, [options])`

Gets proxies from a specific proxy list.

Usage:

```
1 var ProxyLists = require('proxy-lists');
2
3 // `getProxiesFromSource` returns an event emitter.
4 ProxyLists.getProxiesFromSource('freeproxylists', {
5   anonymityLevels: ['elite']
6 })
7   .on('data', function(proxies) {
```

```
8      // Received some proxies.
9      console.log('got some proxies');
10     console.log(proxies);
11   })
12   .on('error', function(error) {
13     // Some error has occurred.
14     console.log('error!', error);
15   })
16   .once('end', function() {
17     // Done getting proxies.
18     console.log('end!');
19   });
```

Options for getProxiesFromSource Method See Options for getProxies Method.

addSource

`addSource(name, source)`

Add a custom proxy source to the list of available proxies. The new proxy source will be used in addition to the existing sources, when calling `getProxies()`.

Usage:

```
1  var ProxyLists = require('proxy-lists');
2
3  ProxyLists.addSource('my-custom-source', {
4    homeUrl: 'https://somewhere.com',
5    getProxies: function(options) {
6
7      var emitter = options.newEventEmitter();
8
9      _.defer(function() {
10        // When an error occurs, use the 'error' event.
11        // The 'error' event can be emitted more than once.
12        emitter.emit('error', new Error('Something bad happened!'))
13        ;
14
15        // When proxies are ready, use the 'data' event.
16        // The 'data' event can be emitted more than once.
17        emitter.emit('data', proxies);
18
19        // When done getting proxies, emit the 'end' event.
20        // The 'end' event should be emitted once.
21        emitter.emit('end');
22      });
23      // Must return an event emitter.
```

```
24     return emitter;
25   }
26 });
```

Your proxy source is required to return the following for each proxy: `ipAddress`, `port`. See Proxy Object above for more information.

Please consider sharing your custom proxy sources by creating a pull request to have them added to this module so that others can use them too.

Important Options to Note Please note that there are a couple options that you should respect in your custom proxy source: * **sample** - **boolean** If `options.sample` is **true** then you should do your best to make the fewest number of HTTP requests to the proxy source but still get at least some real proxies. The purpose of this option is to reduce the strain caused by this module's unit tests on each proxy sources' servers. * **series** - **boolean** If `options.series` is **true** you should make sure that all asynchronous code in your custom source is run in series, NOT parallel. The purpose is to reduce the memory usage of the module so that it can be run in low-memory environments such as a VPS with 256MB of RAM.

listSources

`listSources([options])`

Get list of all available proxy sources.

Usage:

```
1 var ProxyLists = require('proxy-lists');
2
3 var sources = ProxyLists.listSources();
```

Sample sources:

```
1 [
2   {
3     name: 'freeproxylists',
4     homeUrl: 'http://www.freeproxylists.com'
5   },
6   {
7     name: 'gatherproxy',
8     homeUrl: 'http://www.gatherproxy.com'
9   }
10 ]
```

Options for listSources Method

```
1 var options = {
2   /*
3     Include proxy sources by name.
4
5     Only 'freeproxylists':
6     ['freeproxylists']
7   */
8   sourcesWhiteList: null,
9
10  /*
11    Exclude proxy sources by name.
12
13    All proxy sources except 'freeproxylists':
14    ['freeproxylists']
15  */
16  sourcesBlackList: null
17 };
```

Usage with Proxy

It is possible to use a proxy while getting proxies, using the `"browser"` and `"defaultRequestOptions"` options. This module uses both request and puppeteer under-the-hood to scrape web pages. So you will have to configure both of those to use a proxy while getting proxies from every possible source.

Here is an example using the API:

```
1 var ProxyLists = require('proxy-lists');
2
3 ProxyLists.getProxies({
4   browser: {
5     // arguments passed to puppeteer browser instance:
6     args: [ '--proxy-server=127.0.0.1:9876' /* your proxy */ ]
7   },
8   defaultRequestOptions: {
9     // Passed as default options to the request module.
10    // Read the following for details about proxy usage and request
11    // : https://github.com/request/request#proxies
12    proxy: 'http://127.0.0.1:9876',
13  }
14 })
15 .on('data', function(proxies) {
16   console.log(proxies);
17 });
```

It is not currently possible to pass the above options via the CLI. But if you'd like to add this feature,

pull requests are welcome ;)

Contributing

There are a number of ways you can contribute:

- **Improve or correct the documentation** - All the documentation is in this [readme.md](#) file. If you see a mistake, or think something should be clarified or expanded upon, please submit a pull request
- **Report a bug** - Please review existing issues before submitting a new one; to avoid duplicates. If you can't find an issue that relates to the bug you've found, please create a new one.
- **Request a feature** - Again, please review the existing issues before posting a feature request. If you can't find an existing one that covers your feature idea, please create a new one.
- **Fix a bug** - Have a look at the existing issues for the project. If there's a bug in there that you'd like to tackle, please feel free to do so. I would ask that when fixing a bug, that you first create a failing test that proves the bug. Then to fix the bug, make the test pass. This should hopefully ensure that the bug never creeps into the project again. After you've done all that, you can submit a pull request with your changes.

Configure Local Environment

Step 1: Get the Code First, you'll need to pull down the code from GitHub:

```
1 git clone https://github.com/chill117/proxy-lists.git
```

Step 2: Install Dependencies Second, you'll need to install the project dependencies as well as the dev dependencies. To do this, simply run the following from the directory you created in step 1:

```
1 npm install
```

Tests

This project includes an automated regression test suite. To run the tests:

```
1 npm test
```

Changelog

See changelog.md

License

This software is MIT licensed: > A short, permissive software license. Basically, you can do whatever you want as long as you include the original copyright and license notice in any copy of the software/-source. There are many variations of this license in use.

Funding

This project is free and open-source. If you would like to show your appreciation by helping to fund the project's continued development and maintenance, you can find available options here.