ip-regex

> *Regular expression for matching IP addresses*

Install

```
$ npm install ip-regex
```

This module targets Node.js 8 or later and the latest version of Chrome, Firefox, and Safari. If you want support for older browsers, use version 2.1.0: `npm install ip-regex@2.1.0`

Usage

```js
const ipRegex = require('ip-regex');

// Contains an IP address?
ipRegex().test('unicorn 192.168.0.1');
//=> true

// Is an IP address?
ipRegex({exact: true}).test('unicorn 192.168.0.1');
//=> false

ipRegex.v6({exact: true}).test('1:2:3:4:5:6:7:8');
//=> true

'unicorn 192.168.0.1 cake 1:2:3:4:5:6:7:8 rainbow'.match(ipRegex());
//=> ['192.168.0.1', '1:2:3:4:5:6:7:8']

// Contains an IP address?
ipRegex({includeBoundaries: true}).test('192.168.0.2000000000');
//=> false

// Matches an IP address?
'192.168.0.2000000000'.match(ipRegex({includeBoundaries: true}));
//=> null
```

API

ipRegex([options])

Returns a regex for matching both IPv4 and IPv6.

ipRegex.v4([options])

Returns a regex for matching IPv4.

ipRegex.v6([options])

Returns a regex for matching IPv6.

options

Type: `Object`

exact

Type: `boolean`
Default: `false` *(Matches any IP address in a string)*

Only match an exact string. Useful with `RegExp#test()` to check if a string is an IP address.

includeBoundaries

Type: `boolean`
Default: `false`

Include boundaries in the regex. When `true`, `192.168.0.2000000000` will report as an invalid IPv4 address. If this option is not set, the mentioned IPv4 address would report as valid (ignoring the trailing zeros).

Related

- is-ip - Check if a string is an IP address
- is-cidr - Check if a string is an IP address in CIDR notation
- cidr-regex - Regular expression for matching IP addresses in CIDR notation

License

MIT © Sindre Sorhus