

```

// client has a fresh copy of
// resource
res.statusCode = 304
res.end()
return
}

// send the resource
res.statusCode = 200
res.end('hello, world!')
})

function isFresh (req, res) {
  return fresh(req.headers, {
    etag: res.getHeader('ETag'),
    'last-modified':
      res.getHeader('Last-Modified')
  })
}

server.listen(3000)

```

License

[MIT](#)

fresh

HTTP response freshness testing

Installation

This is a [Node.js](#) module available through the [npm registry](#).
Installation is done using the [npm install command](#):

\$ npm install fresh

API

fresh(reqHeaders, resHeaders)

Check freshness of the response using request and response headers.

```
    if (isFresh(req, res)) {
```

```
        Modified response headers
        // ... including adding Etag / Last-
        // perform server logic
        (req, res) {
```

var server = http.createServer(function

```
        var http = require('http')
        var fresh = require('fresh')
```

Using with Node.js http server

```
// => true
fresh(reqHeaders, resHeaders)
var resHeaders = { etag: 'foo' }
        'foo' }
var reqHeaders = { 'if-none-match':
```

```
// => false
fresh(reqHeaders, resHeaders)
var resHeaders = { etag: 'bar' }
        'bar' }
var reqHeaders = { 'if-none-match':
```

API usage

Example

There is a known issue that in certain versions of Safari, Safari will incorrectly make a request that allows this module to validate freshness of the resource even when Safari does not have a representation of the resource in the cache. The module [jumani](#) also provides links to further reading on this Safari bug.

This module is designed to only follow the HTTP specifications, not to work-around all kinds of client bugs (especially since this module typically does not receive enough information to understand what the client actually is).

Known Issues