

```

    // 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.

When the response is still “fresh” in the client’s cache true is returned, otherwise false is returned to indicate that the client cache is now stale and the full response should be sent.

When a client sends the Cache-Control: no-cache request header to indicate an end-to-end reload request, this module will return false to make handling these requests transparent.

Known Issues

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).

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 [jumanji](#) can be used in an Express application to work-around this issue and also provides links to further reading on this Safari bug.

2

Example

API usage

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

Using with Node.js http server

```
var fresh = require('fresh')  
var http = require('http')  
  
var server = http.createServer(function  
  (req, res) {  
    // perform server logic  
    // ... including ETag / Last-  
    Modified response headers  
  
    if (!fresh(req, res)) {
```

3