

range-parser

npm v1.2.1 downloads 348.1M/month node >= 0.6
travis 404 coverage 100%

Range header field parser.

Installation

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

```
$ npm install range-parser
```

API

```
var parseRange = require('range-parser')
```

```
parseRange(100,  
  'bytes=50-55,0-10,5-10,56-60',  
  { combine: true })  
  // [ { start: 0, end: 10 },  
  //   { start: 50, end: 60 } ]  
  // size of the resource. An array of ranges will be returned or  
  // negative numbers indicating an error parsing.  
  // 2 signals a malformed header string  
  // 1 signals an unsatisfiable range  
  var range = parseRange(size,  
  // parse header from request  
  // the type of the range  
  // if (range.type === 'bytes') {  
  //   the ranges  
  //   range.forEach(function (r) {  
  //     do something with r.start and  
  //     r.end  
  //   })  
  // }  
  // {  
  //   start: 50, end: 60  
  // }  
  // {  
  //   start: 0, end: 10  
  // }  
  // ]
```

License

MIT

`parseRange(size, header, options)`

Specifies if overlapping & adjacent ranges should be combined, defaults to `false`. When `true`, ranges will be combined and returned as if they were specified that way in the header.

These properties are accepted in the options object.

Options

```
{  
}  
r.end  
// do something with r.start and  
range.forEach(function (r) {  
  // the ranges  
  if (range.type === 'bytes') {  
    // the type of the range  
    // if (range.type === 'bytes') {  
    //   the ranges  
    //   range.forEach(function (r) {  
    //     do something with r.start and  
    //     r.end  
    //   })  
    // }  
    // {  
    //   start: 50, end: 60  
    // }  
    // {  
    //   start: 0, end: 10  
    // }  
    // ]  
  // size of the resource. An array of ranges will be returned or  
  // negative numbers indicating an error parsing.  
  // 2 signals a malformed header string  
  // 1 signals an unsatisfiable range  
  var range = parseRange(size,  
  // parse header from request  
  // the type of the range  
  // if (range.type === 'bytes') {  
  //   the ranges  
  //   range.forEach(function (r) {  
  //     do something with r.start and  
  //     r.end  
  //   })  
  // }  
  // {  
  //   start: 50, end: 60  
  // }  
  // {  
  //   start: 0, end: 10  
  // }  
  // ]
```

- 1 signals an unsatisfiable range
- 2 signals a malformed header string
- Parse the given header string where `size` is the maximum size of the resource. An array of ranges will be returned or negative numbers indicating an error parsing.