

- 0.3.0 test options.slashesDenoteHost=false, add something like options.externalDirectoryIndexes=[] for external sites

relateur l

Minify URLs by converting them from absolute to relative.

If you were to use this library on a website like `http://example.com/dir1/dir1-1/`, you would get results such as:

Before	After
http://example.com/dir1/ dir1-2/index.html	.. / dir1-2 / dir1-2-1 / index.html
http://example.com/dir2/ dir2-1 /	.. / dir2 / dir2-1 / dir2-1-1 /
http://example.com/dir1/ dir1-1 /	.. / dir1 / dir1-1 /
https://example.com/ dir1/dir1-1 /	https:// example.com / dir1 / dir1-1 /
http://google.com:80/ dir /	//google.com / dir /
../../../../../../../../ #anchor	/ #anchor

All string parsing. No directory browsing. It is thoroughly tested, very fast and lightweight with zero external dependencies.

Release History

path. relative are both slower and less powerful than this library.

Getting Started

This utility requires `Node.js >= 0.10`. To install, type this at the command line:

```
npm install relativeurl --save-dev
```

- 0.2.7 Node v6 support
- 0.2.6 minor enhancements
- 0.2.5 added options. removeRootTrailingSlash
- 0.2.4 added options. site
- 0.2.3 added browserify npm-script
- 0.2.2 removed task runner
- 0.2.1 shorten resource- and query-relative URLs, test variations list with other site URLs
- 0.2.0 cleanup, code
- 0.1.0 initial release

- options.removeEmptyQuerries=true only applied to unrelated URLs
- options.removeOptions.list with other site URLs
- 0.2.1 shorten resource- and query-relative URLs, test variations list with other site URLs
- 0.2.0 cleanup, code
- 0.1.0 initial release

Roadmap

- 0.2.8 check if queries are the same, regardless of param order
- 0.2.8 possible scheme exceptions such as tel :
- 0.2.8 decipher and return invalid input (special cases) to complete test suite

Type: Boolean	options.ignore_www	Default value: false
Type: Array	options.directoryIndexes	Default value: ["index.html"]
Extend the list with any resources you need. Works with	options.removedirectoryIndexes	example.com:80/ will become http://example.com/.
these default ports will have them removed. Example: http://	options.directoryIndexes	example.com:80/test will become http://example.com/test
Extend the list with any ports you need. Any URLs containing	options.directoryIndexes	example.com:80/test will become http://example.com:80/test

Single Instance

```
var RelateUrl = require("relateurl");

var result = RelateUrl.relate(from, to,
    options);
```

Reusable Instances

```
var RelateUrl = require("relateurl");

var instance = new RelateUrl(from,
    options);

var result1 = instance.relate(to1);
var result2 = instance.relate(to2,
    customOptions);
var result3 = instance.relate(to3);
```

FAQ

1. Why bother writing/using this?

To aid in further minifying HTML, mainly for the purpose of faster page loads and SEO. It's been integrated into [HTMLMinifier](#).

2. Why not just use Node's url.parse, url.resolve and path.relative?

url.parse is used, but url.resolve and

This will, for example, consider any domains containing <http://www.example.com/> to be related to any that contain <http://example.com/>.

options.output

Type: constant or String

Choices:

RelateUrl.ABSOLUTE, RelateUrl.PATH_RELATIVE, RelateUrl.

Choices:

"absolute", "pathRelative", "rootRelative", "shortest"

Default value: RelateUrl.SHORTEST

RelateUrl.ABSOLUTE will produce an absolute URL.
Overrides [options.schemeRelative](#) with a value of false.

RelateUrl.PATH_RELATIVE will produce something like .. / child-of-parent / etc / .

RelateUrl.ROOT_RELATIVE will produce something like / child-of-root / etc / .

RelateUrl.SHORTEST will choose whichever is shortest between root- and path-relative.

options.rejectedSchemes

Type: Array

Default value: ["data", "javascript", "mailto"]

Extend the list with any additional schemes. Example:
javascript:something will not be modified.

modified.

var while `http://domain.com/dir/?var` will not be

`domain.com/?var` will become `http://domain.com?`

remove trailing slashes from root paths. Example: `http://`

`domain.com` trailing slashes from root paths. Example: `http://domain.com?domain.com/?var`

Default value: true

Type: Boolean

options.removeRootTrailingSlash

unrelated URLs (with other protocols, auths, hosts and/or ports).

`http://domain.com/?var3=adsf`. This does not apply to

`domain.com/?var1&var2=&var3=adsf` will become

`http://domain.com` empty query variables. Example: `http://`

Default value: false

Type: Boolean

options.removeEmptyQueries

options.rmtreeIndexes.

Remove any resources that match any found in

Default value: true

Type: Boolean

options.rmtreeIndexes

Remove user authentication information from the output URL.

Default value: false

Type: Boolean

This library can be used as a [function for single-use](#) or as a

class for [multiple conversions](#).

Upon successful conversion, a `String` will be returned. If an

issue is encountered while parsing from, an error will be thrown.

Examples

Passed to Node's [URL.parse](#).

Default value: true

Type: Boolean

options.slashesDenoteHost

An options-based version of the `from` argument. If both are

specified, from takes priority.

Default value: undefined

Type: String

options.site

example.com/will become //example.com/.

Output URLs relative to the scheme. Example: `http://`

Default value: true

Type: Boolean

options.schemeRelative

var while `http://domain.com/dir/?var` will not be

`domain.com/?var` will become `http://domain.com?`

remove trailing slashes from root paths. Example: `http://`

`domain.com` trailing slashes from root paths. Example: `http://domain.com?domain.com/?var`

Default value: true

Type: Boolean

Default value: false

Type: Boolean

Default value: true

Type: Boolean

Default value: true