

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

relateurl

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
<code>http://example.com/dir1/dir1-2/index.html</code>	<code>../dir1-2/</code>
<code>http://example.com/dir2/dir2-1/</code>	<code>/dir2/dir2-1/</code>
<code>http://example.com/dir1/dir1-1/</code>	
<code>https://example.com/dir1/dir1-1/</code>	<code>https://example.com/dir1/dir1-1/</code>
<code>http://google.com:80/dir/</code>	<code>//google.com/dir/</code>
<code>../../../../../../../../..</code>	<code>/#anchor</code>
<code>#anchor</code>	

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

Getting Started

This utility requires [Node.js](#) >= 0.10. To install, type this at the command line:

```
npm install relateurl --save-dev
```

Options

options.defaultPorts

Type: Object

Default value: {ftp:21, http:80, https:443}

Extend the list with any ports you need. Any URLs containing

these default ports will have them removed. Example: http://

example.com:80/ will become http://example.com/.

options.directoryIndexes

Type: Array

Default value: ["index.html"]

Extend the list with any resources you need. Works with

[options.removeDirectoryIndexes](#).

options.ignore_www

Type: Boolean

Default value: false

Release History

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

- 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 code cleanup, options.removeEmptyQueries=true only applied to unrelated URLs
- 0.1.0 initial release

Roadmap

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

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.

options.removeAuth

Type: Boolean
Default value: false
Remove user authentication information from the output URL.

options.removeDirectoryIndexes

Type: Boolean
Default value: true
Remove any resources that match any found in [options.directoryIndexes](#).

options.removeEmptyQueries

Type: Boolean
Default value: false
Remove empty query variables. Example: http://domain.com/?var1&var2=&var3=asdf will become http://domain.com/?var3=asdf. This does not apply to unrelated URLs (with other protocols, auths, hosts and/or ports).

options.removeRootTrailingSlash

Type: Boolean
Default value: true
Remove trailing slashes from root paths. Example: http://domain.com/?var will become http://domain.com/?var while http://domain.com/dir/?var will not be modified.

options.schemeRelative

Type: Boolean
Default value: true
Output URLs relative to the scheme. Example: http://example.com/ will become //example.com/.

options.site

Type: String
Default value: undefined
An options-based version of the [from](#) argument. If both are specified, from takes priority.

options.slashesDenoteHost

Type: Boolean
Default value: true
Passed to Node's [url.parse](#).

Examples

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.