slugify


```
var slugify = require('slugify')

slugify('some string') // some-string

// if you prefer something other than
        '-' as separator
slugify('some string', '_')  //
        some_string
```

- Vanilla ES2015 JavaScript
  - If you need to use Slugify with older browsers, consider using version 1.4.7
- No dependencies
- Coerces foreign symbols to their English equivalent (check out the charMap for more details)
- Works in the browser (window.slugify) and AMD/CommonJS-flavored module loaders

Options

```
slugify('some string', {
  replacement: '-',  // replace spaces
        with replacement character,
        defaults to `-`
  remove: undefined, // remove
        characters that match regex,
        defaults to `undefined`
  lower:
        false,      // convert to lower
        case, defaults to `false`
  strict: false,     // strip special
        characters except replacement,
        defaults to `false`
  locale:
        'vi',       // language code of
        the locale to use
  trim:
        true         // trim leading and
        trailing replacement chars,
        defaults to `true`
})
```

Remove

For example, to remove `*+~.()'"!:@` from the result slug, you can use `slugify('..', {remove: /[*+~.()'"!:@]/g})`.

- If the value of `remove` is a regular expression, it should be a character class and only a character class. It should also use the global flag. (For example: `/[*+~.()'"!:@]/g`.) Otherwise, the `remove` option might not work as expected.
- If the value of `remove` is a string, it should be a single character. Otherwise, the `remove` option might not work as expected.

Locales

The main `charmap.json` file contains all known characters and their transliteration. All new characters should be added there first. In case you stumble upon a character already set in `charmap.json`, but not transliterated correctly according to your language, then you have to add those characters in `locales.json` to override the already existing transliteration in `charmap.json`, but for your locale only.

You can get the correct language code of your language from here.

Extend

Out of the box `slugify` comes with support for a handful of Unicode symbols. For example the ☢ (radioactive) symbol is not defined in the `charMap` and therefore it will be stripped by default:

```
slugify('unicode ♥ is ☢') // unicode-
        love-is
```

However you can extend the supported symbols, or override the existing ones with your own:

```
slugify.extend({'☢': 'radioactive'})
slugify('unicode ♥ is ☢') // unicode-
        love-is-radioactive
```

Keep in mind that the `extend` method extends/overrides the default `charMap` for the entire process. In case you need a fresh instance of the slugify's `charMap` object you have to clean up the module cache first:

```
delete require.cache[require.resolve('slugify')]
var slugify = require('slugify')
```

Contribute

1. Add chars to `charmap.json`

2. Run tests `npm test`
3. The tests will build the charmap in `index.js` and will sort the `charmap.json`
4. Commit **all** modified files

Originally this was a vanilla javascript port of node-slug.
Note that the original slug module has been ported to vanilla javascript too.