
nodash

Release 1.1.0a

Crepac4

Nov 19, 2020

CONTENTS

1	Description	3
2	To be implemented	5
3	What does ‘nodash’ mean	7
	Index	11

An ‘extension’ of lodash for node

[MIT license](#) [GitHub last commit](#) [Node.js CI nycrc config on GitHub](#) [Documentation](#) [Status](#)

DESCRIPTION

An extension of lodash with functions and utilities that I use for nodejs.

TO BE IMPLEMENTED

1. Functions for distinguishing between async Function Generator, Function Generators, async Functions, Functions, Classes, Literal objects. (Done)
2. syntactic sugar for synchronous get of a json file (Done)
3. Other functionality that I miss from python (in progress)

WHAT DOES 'NODASH' MEAN

lodash - nodash - node dash.

Note: This library is at it's very early stages. Any suggestions are welcome.

writeAsCSV (*p*, *data*)

Write a literal object to file as a comma-separated values (CSV) file. Like standard `writeFile`, `writeAsCSV()` is Async

Arguments

- **p** (*String*) – Absolute path to the location of the comma-separated values (CSV) file.
- **data** (*Object*) – The literal Object to save as a comma-separated values (CSV) file.

Returns Promise – Void - Represents the completion of an asynchronous operation.

Examples:

```
let jsonData = {id: 1, first_name: 'john', last_name: 'white'}
nodash.writeAsCSV(`${__dirname}/tmp/data.csv`, jsonData); // returns a promise
```

isObject (*a*)

Determines if a value is an Object.

Arguments

- **a** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is an Object. False Otherwise. NOTE: Object does not necessarily mean only a literal object in javascript.

Examples:

```
let arr = [1,3,4,5]
isObject(arr) //returns true
```

isGenerator (*fn*)

Determines if a value is a function generator.

Arguments

- **fn** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is a function generator. False Otherwise.

Examples:

```
function *test() {}  
isGenerator(test) //returns true
```

isClass (*fn*)

Determines if a value is a class.

Arguments

- **fn** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is a class. False Otherwise.

Examples:

```
class Test {}  
isClass(test) //returns true
```

isFunction (*fn*)

Determines if a value is a function.

Arguments

- **fn** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is a function. False Otherwise.

Examples:

```
function test() {}  
isFunction(test) //returns true
```

isGeneratorFunction (*fn*)

Determines if a value is a function generator.

Arguments

- **fn** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is a function generator. False Otherwise.

isAsyncFunction (*fn*)

Determines if a value is an asynchronous function.

Arguments

- **fn** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is an asynchronous function. False Otherwise.

Examples:

```
async function test() {}  
isAsyncFunction(test) //returns true
```

isAsyncGenerator (*fn*)

Determines if a value is an asynchronous function generator.

Arguments

- **fn** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is an asynchronous function generator. False Otherwise.

Examples:

```

async function* test() {}
isAsyncGenerator(test) //returns true

```

isArray (*a*)

Determines if a value is an array.

Arguments

- **a** (*any*) – The value to be checked.

Returns Boolean – Returns True if the value is an array. False Otherwise.

Examples:

```

let arr = [1,2,3,4,5]
isArray(arr) //returns true

```

isUrl (*a*)

Determines if a string is a valid url.

Arguments

- **a** (*any*) – The string to be checked.

Returns Boolean – Returns True if the value is a valid url. False Otherwise.

Examples:

```

let test = 'https://google.com'
isAsyncGenerator(test) //returns true

```

get (*thing*)

Synchronous get request. Only to be used at the start of the program so as to avoid crashes and other bottlenecks.

Arguments

- **thing** (*string*) – A string containing the address from which to make the get request.

Returns Object – Returns the result as a literal Object (JSON) - Default Encoding: utf8

Examples:

```

async function* test() {}
let test_url = https://sometesturl.com/api/someendpoint.json
req('GET', test_url) //returns parsed json response

```

uniqSortBy (*data*, *key*)

This function combines lodash's uniq and sortBy in one.

Arguments

- **data** (*Object*) – The data that we need to sort and uniq.
- **key** (*Object*) – The key to sort againsts.

Returns Object –

dieWhen (*cond*, *hard*)

Similar to php's die with the added convinience of conditional death.

Arguments

- **cond** (*Boolean*) – The condition of death.
- **hard** (*Object*) – An array of messages print out upon death.

Returns Object –

die (*cond*, ...*hard*)

Similar to php's die. It exits the current node process while also providing the ability to print a message (useful for debugging)

Arguments

- **cond** (*Boolean*) – The condition of death.
- **hard** (*Object*) – An array of messages print out upon death.

Returns Object –

INDEX

D

`die()` (*built-in function*), 10

`diewhen()` (*built-in function*), 9

G

`get()` (*built-in function*), 9

I

`isArray()` (*built-in function*), 9

`isAsyncFunction()` (*built-in function*), 8

`isAsyncGenerator()` (*built-in function*), 8

`isClass()` (*built-in function*), 8

`isFunction()` (*built-in function*), 8

`isGenerator()` (*built-in function*), 7

`isGeneratorFunction()` (*built-in function*), 8

`isObject()` (*built-in function*), 7

`isUrl()` (*built-in function*), 9

U

`uniqSortBy()` (*built-in function*), 9

W

`writeAsCSV()` (*built-in function*), 7