Lodash

Stop reinventing the wheel

Santiago Camargo Rodríguez

Agenda

- Contexto
- API
- Programación funcional
- Usar o no usar lodash
- Webpack, custom builds

Problema

- Escasez de funciones para manipular datos y conjuntos. Necesitamos filtrar, agrupar, cambiar propiedades, ordenar, etc.
- El desarrollo de una aplicación tiene tiempo límite.
- Hay que enfocarse en resolver problemas de negocio, no resolver problemas que resuelven problemas de negocio.

¿Solución?

- Creación de helpers/utils cuyas APIs pueden llegar a ser inconsistentes:
  - `¿ addToGroup(collection, element) ?`
  - `¿ addToGroup(element, collection) ?`
  - `¿ collection.addToGroup(element)  ?`
- Hay que añadirles pruebas unitarias para garantizar un código de calidad y fiable.
- Estamos reinventando la rueda.

Librerías de utilidades

sugar.js (https://sugarjs.com)

valentine.js (https://github.com/ded/valentine)

wu.js (http://fitzgen.github.io/wu.js)

lazy.js (http://danieltao.com/lazy.js)

ramda (http://ramdajs.com/)

underscore (http://underscorejs.org/)

lodash (https://lodash.com/)

Lodash

- Librería cargada con una buena batería de funciones útiles (más de 300).
- Enfocada sobre todo en arrays, objetos, funciones y strings.
- Creada por John-David Dalton.
- Superset de underscore.

¿Qué ofrece Lodash?

- Buen rendimiento.
- Consistencia.
- Seguridad, menos bugs (más de 6.500 pruebas unitarias)
- Personalización.
- No extiende las clases base de JavaScript.

Conceptos de lodash

- `truthy, falsy` → verdadero o falso
- `iteratee` → función que va a ser aplicada sobre cada elemento.
- `predicate` → función que recibe un argumento y devuelve verdadero o falso
- `SameValueAsZero` → Algoritmo equivalente a ≡ incluyendo NaN
- `comparator` → función que recibe dos argumentos y devuelve verdadero o falso
- `customizer` → función que recibe dos argumentos y devuelve verdadero o falso

API

- Lang
- Strings
- Array
- Colecciones
- Objetos
- Funciones

Leyenda:

función inmutable

función mutable

Lang

_.is* _.to*

castArray	isDate	isNil	gt, gte
isArrayLike	isEmpty	isUndefined	lt, lte
isArrayLikeObject	isEqual	toArray	clone
IsBoolean	isError	toInteger	cloneDeep
IsString	isFunction	toString	
isNumber	IsNull	toNumber	

String

```
camelCase
capitalize
deburr
escape
unescape
kebabCase
pad
padStart
padEnd
snakeCase
truncate
words
```

Array

chunk

difference ⇔ pullAll

differenceBy ⇔ pullAllBy

differenceWith ⇔ pullAllWith

drop

dropRight

dropWhile

dropRightWhile

flatten

flattenDeep

flattenDepth

fromPairs

toPairs

intersection

intersectionBy

intersectionWith

sortedIndex

sortedIndexBy

sortedLastIndex

sortedLastIndexBy

union

unionBy

unionWith

uniq, sortedUniq

uniqBy, sortedUniqBy

uniqWith

without ⇔ pull

xor

xorBy

xorWith

Collection

```
countBy
groupBy
keyBy
orderBy
partition
reject
sample
sampleSize
shuffle
sortBy
```

Objects

defaults	functions	merge
defaultsDeep	functionsIn	
forIn	get	
forInRight	set	
forOwn	has	
forOwnRight	hasIn	

Function

```
after
before
ary, unary
curry, curryRight
partial, partialRight
debounce
defer
memoize
once
negate
overArgs
throttle
flow, flowRight (compose)
cond
```

¿Necesito lodash? API ES5/6+

Array

```
concat
fill
find
findIndex
indexOf
join
lastIndexOf
reverse
slice
(for)each
every
filter
includes
map
reduce
reduceRight
some
```

Object

```
assign
keys
toPairs (entries)
values
```

Number

```
isNaN
```

String

```
repeat
template (literal)
toLower
toUpper
trim
replace
```

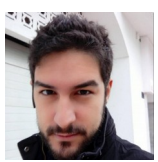
Métodos implementables

`compact`	`without`	`nth`
`faltten`	`minBy`	`minBy`
`flattenDeep`	`maxBy`	`maxBy`
`head`	`range`	`range`
`tail`	`size`	`size`
`initial`	`after`	`after`

Custom builds y uso

Webpack plugins: babel-plugin-lodash, & lodash-webpack-plugin

- ES Modules: lodash-es
- lodash-cli (https://lodash.com/custom-builds)
- Función por paquete
- Aviso de discontinuidad de lodash-cli y función por paquete



Santiago

Lemoncoder

@crsantiblack

www.lemoncode.net