

Blog

<https://blogcongnghe.tronghao.com/lodash>

Tách ra

Array

`_.chunk` Tách array chính ra thành những array con với số lượng nhất định.

`_.unzip` Tạo những array theo vị trí của các array.

String

`_.words` Cắt các chữ trong string bỏ vào mảng.

Thay thế

Array

`_.fill` Thay thế thành ... (ký tự nào đó) bắt đầu từ đâu đến đâu (không tính end).

Tim kiếm

Array

`_.findIndex` `_.findIndex(array, [predicate=_.identity], [fromIndex=0])`
x Tìm vị trí của item đầu tiên thỏa điều kiện.

`_.findLastIndex` Tìm vị trí của item cuối cùng thỏa điều kiện.

`_.indexOf` Search item in array, return index tìm thấy.

`_.findKey` Tìm và trả về key của một phần tử.

Lọc

Array

`_.filter` Lọc ra item theo điều kiện.

`_.find` Lọc ra item theo điều kiện. Tương tự hàm `_.filter`

Lọc (cont)

`_.reduce` Lọc giống như filter nhưng trả về dạng object `group_by`.

Lưu ý: Cần return và đổi số thứ ba là {}, là kết quả mặc định khi không có dữ liệu.

Lặp

`_.map` Tạo ra mảng mới sau khi xử lý cái gì đó ở mảng cũ.

`_.mapValues` Trả về object mới từ object cũ sau khi đã xử lý gì đó.

`_.foreach` Lặp qua các element.

Hạ Cấp

Array

`_.flatten` Hạ cấp mảng xuống 1 level.

`_.flattenDeep` Hạ cấp mảng về level 1.

`_.flattenDepth` Hạ cấp mảng xuống n level.

Tạo Object

Array

`_.fromPairs` Tạo object từ [key, value] trong array.

`_.zipObject` Lấy array 1 làm key, Lấy array 2 làm value.

`_.zipObjectDeep` Giống trên nhưng phức tạp hơn.

Object

`_.cloneDeep` Tạo object mới ở địa chỉ mới, để khi thay đổi không ảnh hưởng object ban đầu.

Nghịch đảo

Array

`_.reverse` Đảo ngược phần tử của mảng.

Gom nhóm

Array

`_.groupBy` Gom nhóm object.

Duy nhất

Array

`_.union` Lấy ra những item duy nhất của một hoặc nhiều array.

`_.unionBy` Lấy ra những item duy nhất của một hoặc nhiều array nhưng có thêm điều kiện.

`_.uniq` Lấy ra những item duy nhất của một array.

`_.uniqBy` Lấy ra những item duy nhất của một array nhưng theo điều kiện.

Xóa

Array

`_.compact` Xóa những dữ liệu là false khỏi array.
Những dữ liệu là false: false, 0, empty, null, undefined, NaN

`_.drop` Cắt n phần tử từ vị trí đầu tiên.

`_.dropRight` Cắt n phần tử từ vị trí cuối cùng.

`_.dropRightWhile` Cắt n phần tử từ vị trí cuối cùng với điều kiện nào đó.

`_.dropWhile` Cắt n phần tử từ vị trí đầu tiên với điều kiện nào đó.

`_.initial` Bỏ phần tử cuối cùng của mảng.

`_.pull` Xóa các item có giá trị bằng với item value khỏi mảng.

Xóa (cont)

<code>_.pullAll</code>	Giống <code>_.pull</code> nhưng value dạng array.
<code>_.pullAllBy</code>	Giống <code>_.pull</code> nhưng kèm điều kiện.
<code>_.pullAt</code>	Loại bỏ item tại các index.
<code>_.remove</code>	Xóa theo điều kiện. Xóa luôn item trong array gốc.
<code>_.without</code>	Xóa item có trong mảng giống với các value.
<code>_.sortedUniq</code>	Bỏ phần tử trùng, tối ưu hóa cho mảng đã sắp xếp.
<code>_.sortedUniqBy</code>	Bỏ phần tử trùng, tối ưu hóa cho mảng đã sắp xếp. Nhưng có thêm điều kiện.

String

<code>_.trim</code>	Bỏ khoảng trắng hai đầu.
<code>_.trimStart</code>	Bỏ khoảng trắng bên trái.
<code>_.trimEnd</code>	Bỏ khoảng trắng bên phải.
<code>_.truncate</code>	Cắt string và thêm dấu "..." vào nếu string dài quá length.

Sắp xếp

Array

<code>_.sortedIndex</code>	Xác định vị trí chèn vào mảng, có giá trị gần với giá trị cần chèn nhất và nhỏ hơn giá trị cần chèn.
<code>_.sortedIndexBy</code>	Xác định vị trí chèn vào mảng, có giá trị gần với giá trị cần chèn nhất và nhỏ hơn giá trị cần chèn. Nhưng có thêm điều kiện.

Sắp xếp (cont)

<code>_.sortedIndexOf</code>	Xác định vị trí chèn vào mảng đã được sắp xếp, có giá trị gần với giá trị cần chèn nhất và nhỏ hơn giá trị cần chèn.
<code>_.sortedLastIndex</code>	Xác định vị trí chèn vào mảng, có giá trị gần với giá trị cần chèn nhất và lớn hơn giá trị cần chèn.
<code>_.sortedLastIndexBy</code>	Xác định vị trí chèn vào mảng, có giá trị gần với giá trị cần chèn nhất và lớn hơn giá trị cần chèn. Nhưng có thêm điều kiện.
<code>_.orderBy</code>	Sắp xếp mảng.

Kiểm tra

<code>_.isNull</code>	
<code>_.isFunction</code>	
<code>_.isUndefined</code>	
<code>_.isNumber</code>	
<code>_.isObject</code>	
<code>_.isDate</code>	
<code>_.isEmpty</code>	
<code>_.startsWith</code>	Kiểm tra chuỗi có nằm ở đầu string cần so sánh.
<code>_.endsWith</code>	Kiểm tra chuỗi có nằm ở cuối string cần so sánh.

Object

<code>_.has</code>	Kiểm tra object có property hay function nào đó không.
--------------------	--

Object and Array

<code>_.every</code>	true => nếu tất cả phần tử thỏa điều kiện.
----------------------	--

Kiểm tra (cont)

<code>_.some</code>	true => nếu một trong các phần tử thỏa điều kiện
<code>_.includes</code>	Kiểm tra giá trị có nằm trong collection không.
<code>_.isEqual</code>	Kiểm tra có giống nhau không.

Biến đổi

String

<code>_.camelCase</code>	Chuyển chữ dạng camel case.
<code>_.capitalize</code>	Viết hoa chữ cái đầu.
<code>_.pad</code>	Bổ sung chữ bằng cách thêm ký tự 2 bên.
<code>_.padStart</code>	Bổ sung chữ bằng cách thêm ký tự bên trái.
<code>_.padEnd</code>	Bổ sung chữ bằng cách thêm ký tự bên phải.
<code>_.startCase</code>	Viết hoa chữ cái đầu mỗi chữ.
<code>_.deburr</code>	Chuyển chữ có dấu thành chữ không dấu.

Toán học

<code>_.max</code>	Trả về item lớn nhất trong array.
<code>_.ceil</code>	Làm tròn số lên. 5.1 ==> 6 5.6 ==> 6
<code>_.floor</code>	Làm tròn số xuống. 5.1 ==> 5 5.6 ==> 5
<code>_.round</code>	Làm tròn số theo cách thông thường. > 5 lên. < 5 xuống. 5.1 ==> 5 5.6 ==> 6

Nối

Array

<code>_.concat</code>	Nối các phần tử của một hoặc nhiều mảng vào mảng khác.
-----------------------	--

Nối (cont)

<code>_.join</code>	Nối các phần tử lại bằng ký tự gì đó.
<code>_.zip</code>	Gom những vị trí lại thành một mảng.
<code>_.zipWith</code>	Giống trên nhưng có xử lý thêm điều kiện gì đó.

Object

<code>_.merge</code>	Gộp object thành 1 object, không ghi đè undefined.
<code>_.assign</code>	Gộp object thành 1 object, có ghi đè undefined.

So sánh và lấy

Array

<code>_.difference</code>	Lấy các item có trong mảng này mà không có trong mảng kia.
<code>_.differenceBy</code>	Tương tự <code>_.difference</code> nhưng nó sẽ áp dụng tham số thứ 3 apply vào hoặc làm điều kiện so sánh.
<code>_.head</code>	Lấy phần tử đầu tiên của mảng. Giống với <code>_.first(array)</code>
<code>_.intersection</code>	Lấy những phần tử giống nhau của các mảng.
<code>_.intersectionBy</code>	Giống <code>_.intersection</code> nhưng nó sẽ áp dụng tham số thứ 3 apply vào hoặc làm điều kiện so sánh.
<code>_.last</code>	Lấy phần tử cuối cùng của array.
<code>_.lastIndexOf</code>	Tìm phần tử cuối cùng giống value search
<code>_.nth</code>	Lấy phần tử thứ i của array
<code>_.slice</code>	Lấy từ đâu tới đâu, không bao gồm phần tử cuối.
<code>_.tail</code>	Lấy tất cả nhưng bỏ phần tử đầu.

So sánh và lấy (cont)

<code>_.take</code>	Tạo một mảng với n phần tử cắt từ vị trí đầu tiên.
<code>_.takeRight</code>	Tạo một mảng với n phần tử cắt từ vị trí cuối cùng.
<code>_.takeRightWhile</code>	Lấy các phần tử theo điều kiện từ vị trí cuối.
<code>_.takeWhile</code>	Lấy các phần tử theo điều kiện bắt đầu từ vị trí đầu tiên.
<code>_.xor</code>	Lấy những item khác nhau của những array.
<code>_.xorBy</code>	Giống trên nhưng có xử lý thêm gì đó.
<code>_.xorWith</code>	Giống trên nhưng có thêm điều kiện.
<code>_.size</code>	Length của array, object, string
<code>_.keyBy</code>	Lấy dữ liệu dễ hơn dạng mảng và object.
<code>_.reject</code>	Trả về array chứa các element không thỏa điều kiện của tham số thứ hai.

Object

<code>_.functions</code>	Trả về tên toàn bộ những function của 1 object.
<code>_.keys</code>	Trả về toàn bộ key của object.
<code>_.pick</code>	Chỉ lấy 1 số thuộc tính của object.
<code>_.pickBy</code>	Chỉ lấy 1 số thuộc tính của object theo điều kiện.
<code>_.omit</code>	Bỏ một số thuộc tính của object, lấy những thuộc tính còn lại.

So sánh và lấy (cont)

<code>_.omitBy</code>	Bỏ một số thuộc tính của object theo điều kiện, lấy những thuộc tính còn lại.
-----------------------	---

Tiện ích

<code>_.times</code>	Thực hiện số lần cụ thể cho function kết quả.
<code>_.delay</code>	Sau một khoảng thời gian, chỉ c <code>_.setTimeout</code> .
<code>_.debounce</code>	Gọi một function sau một khoảng đó được gọi. Giống <code>_.setInterval</code>
<code>_.get</code>	<code>_.get(object, 'a.b.c', ...)</code> Lấy dữ liệu mà không bị lỗi undefined
<code>_.set</code>	<code>_.set(object, 'a.b.c', value)</code> Gán dữ liệu mà không bị lỗi undefined



By tronghao

cheatography.com/tronghao/

Published 5th February, 2023.

Last updated 12th February, 2023.

Page 3 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>