QA Consulting

Lodash.

TRAINING MATERIALS

Contacts

Elliot.Womack@qa.com

team.qac.all.trainers@qa.com

www.consulting.qa.com

Lodash

DESCRIPTION

lodash is a library with a bunch of functionality, the idea behind it is that pretty much every developer ends up coding some generic utility functions all the time and it's just a waste of time, so lodash is a library with a bunch of popular/common programming constructs which in turn makes your code tidier.

It also has the benefit that they methods are proven to run on all of the popular browsers, so you don't have to worry about compatibility issues.

The majority of the functions are based on the functional programming paradigm.

INSTALLATION

WEBSITES

To use it simply download it and add it into your project via a script tag.

```
<script src=”lodash.js”></script>
```

NPM

```
npm install lodash
```

EXAMPLES

_.TIMES

```
// 1. Basic for loop.
for(var i = 0; i < 5; i++) {
    // ....
}
// 2. Using Array's join and split methods
Array.apply(null, Array(5)).forEach(function(){
    // ...
});
// Lodash
_.times(5, function(){
    // ...
});
```

Loop through a collection and return a nested property

```
// Fetch the name of the first pet from each owner
var ownerArr = [{
    'owner': 'Colin',
    'pets': [{'name':'dog1'}, {'name': 'dog2'}]
}, {
    'owner': 'John',
    'pets': [{'name':'dog3'}, {'name': 'dog4'}]
}];
// Array's map method.
ownerArr.map(function(owner){
   return owner.pets[0].name;
});
// Lodash
_.map(ownerArr, 'pets[0].name');
```

Lodash

_.CLONEDEEP

This is a difficult thing to accomplish in native JS, the best solution we have natively is JSON.parse(JSON.stringify(object)); however this won't work if there are any functions within the object.

```
var objA = {
    'name': 'colin'
}
// Normal method? Too long. See Stackoverflow for solution:
http://stackoverflow.com/questions/4459928/how-to-deep-clone-
in-javascript
// Lodash
var objB = _.cloneDeep(objA);
objB === objA // false
```

_.RANDOM

```
// Get a random number between 15 and 20.
// Naive utility method
function getRandomNumber(min, max){
    return Math.floor(Math.random() * (max - min + 1)) + min;
}
getRandomNumber(15, 20);
// Lodash
_.random(15, 20);
```

If we wanted our random method to do a lot more, floating point numbers, single params for the maximum number etc. It gets a lot more complicated, however lodash' random method has a lot of utility built in

```
_.random(20); // Return random number between 0 to 20
_.random(15, 20, true); // Return random floating numbers
between 15 and 20
```

Lodash

_.OMIT

Removing properties from an object

```
// Naive method: Remove an array of keys from object
Object.prototype.remove = function(arr) {
    var that = this;
    arr.forEach(function(key){
        delete(that[key]);
    });
};
var objA = {'name': 'colin', 'car': 'suzuki', 'age': 17};
objA.remove(['car', 'age']);
objA; // {'name': 'colin'}
// Lodash
objA = _.omit(objA, ['car', 'age']); // {'name': 'colin'}
var objA = {'name': 'colin', 'car': 'suzuki', 'age': 17};
// Lodash
objA = _.omit(objA, 'car'); // {'name': 'colin', 'age': 17};
objA = _.omit(objA, _.isNumber); // {'name': 'colin'};
```

_.SAMPLE

Used for selecting a random item from a list
Instead of generating a random number and then accessing something out of the array, lodash has a function that will sort all that out for us.

```
var luckyDraw = ['Colin', 'John', 'James', 'Lily', 'Mary'];
function pickRandomPerson(luckyDraw){
    var index = Math.floor(Math.random() * (luckyDraw.
length));
    return luckyDraw[index];
}
pickRandomPerson(luckyDraw); // John
// Lodash
_.sample(luckyDraw); // Colin
```

It also has functionality to grab multiple items out of the collection

```
// Lodash - Getting 2 random item
_.sample(luckyDraw, 2); // ['John','Lily']
```

Lodash

_.DEBOUNCE

_.debounce will invoke a function after a certain amount of time since the last time it was invoked.

```
function validateEmail() {
// Validate email here and show error message if not valid
}
var emailInput = document.getElementById('email-field');
emailInput.addEventListener('keyup',
_.debounce(validateEmail, 500));
```

In this example the function validateEmail will be invoked after 500ms so the error message won't display instantly, this timer will reset on each keyup. This way the user won't see an error message until he stops typing.

_.DEBURR

This one is extremely simple yet incredibly useful, it will remove all combining diacritical marks, aka accents on letters. Really useful when you're parsing out of a search function.

```
_.deburr('déjà vu');
// -> deja vu
_.deburr('Juan José');
// -> Juan Jose
```

_.KEYBY

keyBy will take an array and an attribute of the objects in that array, and will construct an object that has all the other objects as the value, and the attribute you specified as the key, almost converting an array of indexed based values into a hashmap with the entire object as the value and the attribute (usually an id) as the value.

```
var posts = [
    { id: '1abc', title: 'First blog post', content: '...'},
    { id: '2abc', title: 'Second blog post', content: '...'
},
    // more blog posts
    { id: '34abc', title: 'The blog post we want', content:
'...' }
    // even more blog posts
];
posts = _.keyBy(posts, 'id');
var post = posts['34abc']
// post -> { id: '34abc', title: 'The blog post we want',
content: '...' }
```

It also lets us call the filter/map/reduce commands on array-like objects, if necessary.

_.COMPACT

Creates an array with falsey values removed, false/null/0/""/undefined/NaN

```
_.compact([0, 1, false, 2, '', 3]);
// => [1, 2, 3]
```

_.JOIN

Converts all elements in array into a string separated by a separator

```
_.join(['a', 'b', 'c'], '~');
// => 'a~b~c'
```

_.REVERSE

Reverse an array

```
var array = [1, 2, 3];
_.reverse(array);
// => [3, 2, 1]
console.log(array);
// => [3, 2, 1]
```

CONCLUSION

Lodash is an amazing useful library, what has been discussed is less than 10% of the total functions lodash provides, but some of the most useful ones.

If you want further examples or more of the methods head over to the documentation (Really well written documentation too!)

https://lodash.com/docs/4.17.4