How To Get Query String in JavaScript

Summary: in this tutorial, you will learn how to use the `URLSearchParams` to get query string parameters in JavaScript.

To get a query string you can access the `search` property of the `location` object:

```
location.search
```

Assuming that the value of the `location.search` is:

```
'?type=list&page=20'
```

To work with the query string, you can use the `URLSearchParams` object.

```
const urlParams = new URLSearchParams(location.search);
```

The `URLSearchParams` is an iterable object, therefore you can use the `for...of` structure to iterate over its elements which are query string parameters:

```
const urlParams = new URLSearchParams(location.search);

for (const [key, value] of urlParams) {
    console.log(`${key}:${value}`);
}
```

Output:

```
type:list
page:20
```

Useful URLSearchParams methods

The `URLSearchParams` has some useful methods that return iterators of parameter keys, values, and entries:

- `keys()` returns an iterator that iterates over the parameter keys.
- `values()` returns an iterator that iterates over the parameter values.
- `entries()` returns an iterator that iterates over the (key, value) pairs of the parameters.

keys() example

The following example uses the `keys()` method to list all parameter names of a query string:

```
const urlParams = new URLSearchParams('?type=list&page=20');

for (const key of urlParams.keys()) {
    console.log(key);
}
```

Output:

```
type
page
```

values() example

The following example uses the `keys()` method to list all parameter values of a query string:

```
const urlParams = new URLSearchParams('?type=list&page=20');

for (const value of urlParams.values()) {
    console.log(value);
}
```

Output:

```
list
20
```

entries() example

The following example uses the `entries()` method to list all pairs of parameter key/value of a query string:

```
const urlParams = new URLSearchParams('?type=list&page=20');

for (const entry of urlParams.entries()) {
    console.log(entry);
}
```

Output:

```
["type", "list"]
["page", "20"]
```

Check if a query string parameter exists

The `URLSearchParams.has()` method returns `true` if a parameter with a specified name exists.

```
const urlParams = new URLSearchParams('?type=list&page=20');

console.log(urlParams.has('type')); // true
console.log(urlParams.has('foo'));  // false
```

Output

```
true
false
```

Summary

- The `URLSearchParams` provides an interface to work with query string parameters
- The `URLSearchParams` is an iterable so you can use the `for...of` construct to iterate over query string parameters.
- The `has()` method of the `URLSearchParams` determines if a parameter with a specified name exists.