⌐ᵖ **d9f4e23cf4** ▾   ···

loader-utils / **lib** / **parseQuery.js** / <> Jump to ▾

🐙 **Evilebot Tnawi** chore: minimum required `Node.js` version is `8.9.0` (#166) ··· ✓   🕐 History

👥 **1 contributor**

69 lines (54 sloc) │ 1.48 KB   ···

```
 1    'use strict';
 2
 3    const JSON5 = require('json5');
 4
 5    const specialValues = {
 6      null: null,
 7      true: true,
 8      false: false,
 9    };
10
11    function parseQuery(query) {
12      if (query.substr(0, 1) !== '?') {
13        throw new Error(
14          "A valid query string passed to parseQuery should begin with '?'"
15        );
16      }
17
18      query = query.substr(1);
19
20      if (!query) {
21        return {};
22      }
23
24      if (query.substr(0, 1) === '{' && query.substr(-1) === '}') {
25        return JSON5.parse(query);
26      }
27
28      const queryArgs = query.split(/[,&]/g);
29      const result = {};
```

```
30
31      queryArgs.forEach((arg) => {
32        const idx = arg.indexOf('=');
33
34        if (idx >= 0) {
35          let name = arg.substr(0, idx);
36          let value = decodeURIComponent(arg.substr(idx + 1));
37
38          // eslint-disable-next-line no-prototype-builtins
39          if (specialValues.hasOwnProperty(value)) {
40            value = specialValues[value];
41          }
42
43          if (name.substr(-2) === '[]') {
44            name = decodeURIComponent(name.substr(0, name.length - 2));
45
46            if (!Array.isArray(result[name])) {
47              result[name] = [];
48            }
49
50            result[name].push(value);
51          } else {
52            name = decodeURIComponent(name);
53            result[name] = value;
54          }
55        } else {
56          if (arg.substr(0, 1) === '-') {
57            result[decodeURIComponent(arg.substr(1))] = false;
58          } else if (arg.substr(0, 1) === '+') {
59            result[decodeURIComponent(arg.substr(1))] = true;
60          } else {
61            result[decodeURIComponent(arg)] = true;
62          }
63        }
64      });
65
66      return result;
67    }
68
69    module.exports = parseQuery;
```