

## 让URLSearchParams帮你生成和解析参数字符串



Henry  
不会做一手好菜的全栈前端不是好产品经理

已关注

54 人赞同了该文章

无论在 Node.js 程序还是 HTML 前端程序里，我们都免不了像下面这样手工拼 URL 搜索字符串（URL Search String，或称 Query String）：

```
// 备注：习惯上我们用 k 和 p 分别代表 keywords 和 pageIndex 的缩写：
function getQueryString(params) {
  if (params) {
    const {
      keywords,
      pageIndex = 1
    } = params;
    return `?k=${encodeURIComponent(keywords || '')}&p=${pageIndex}`;
  }
  throw new Error("params 参数不能为空。");
}

// 返回“?k=%E5%85%B3%E9%94%AE%E5%AD%97&p=1”
console.log(getQueryString({ keywords: '关键字', pageIndex: 1 }));
```

如果你逐行阅读上面的代码，不难发现这段程序并**不是很严谨**，而且虽然只有两个参数，但是却需要考虑参数为 null 或 undefined，此外还需通过 `encodeURIComponent()` 函数来处理 UTF-8 编码问题。

而将 JSON 对象转为 URL 字符串还并不是最大的难题，如何从 `location.search` 解析出参数才是更大的问题。例如，从 react-router v4 开始，不再支持对 URL 参数，你需要自己写代码从 `location.search` 中解析出你需要的参数。

### URLSearchParams

那么你可能会想有没有横跨不同浏览器和 Node.js 的现成解决方案或者开源项目呢？答案一定是有的，而且还是一个标准这就是 **URLSearchParams** ([浏览器端文档](#) | [Node.js 文档](#))。

### 浏览器支持列表



Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari	Edge
Basic support	49	29.0 (29.0)	未实现	46	10.1	未实现
<code>entries()</code> , <code>keys()</code> , <code>values()</code> , and support of <code>for...of</code>	49	44.0 (44.0)	未实现	46	?	未实现
<code>USVString</code> for constructor <code>init</code> object	(Yes)	53.0 (53.0)	未实现	(Yes)	?	未实现
Record for constructor <code>init</code> object	未实现	54.0 (54.0)	未实现	?	?	未实现

	Desktop	Mobile				
Feature	Android	Chrome for Android	Firefox Mobile (Gecko)	IE Mobile	Opera Mobile	Safari Mobile
Basic support	49	49	29.0 (29.0)	未实现	未实现	10.3

## 引用

### 浏览器代码

```
// 如果需要在较低版本的 PC 浏览器，需要额外安装和引用 polyfill:
// $ npm install --save url-search-params
// import URLSearchParams from 'url-search-params';
```

```
const urlSearchParams = new URLSearchParams();
```

### Node.js 代码

```
const { URLSearchParams } = require('url');
```

```
const urlSearchParams = new URLSearchParams();
```

### 生成搜索字符串

```
const params = new URLSearchParams();
params.set('k', '关键字');           // 设置参数
params.set('p', 1);                  // 支持 Boolean、Number 等丰富类型
console.log(params.toString());       // k=%E5%85%B3%E9%94%AE%E5%AD%97&p=1
```

### 解析搜索字符串

```
const params = new URLSearchParams('k=%E5%85%B3%E9%94%AE%E5%AD%97&p=1');
console.log(params.get('k'));         // 返回字符串“关键字”，支持自动 UTF-8 解码
console.log(params.get('p'));         // 返回字符串“1”
console.log(params.get('xxx'));        // 如果没有 xxx 这个键，则返回 null
console.log(params.has('xxx'));        // 当然也可以通过 has() 方法查询是否存在指定的键
console.log(params.keys());            // 返回一个 ES6 Iterator，内含 ['k', 'p']
console.log(params.values());          // 返回一个 ES6 Iterator，内含 ['关键字', '1']
console.log(params.entries());         // 与 Map.prototype.entries() 类似
```



## URL 搜索参数列表

相信不少同学都有过不知该如何在 HTTP GET 参数中传递数组的经历，我看到过不少人使用的这种方式：

```
http://example.com/search?brands=bmw,audi,mercedes
```

事实上，上面这种方式是不够标准的，并且需要处理原本字符串中就包含 “,” 的情况，而在业界广为**推荐的范式**是这样的：

```
http://example.com/search?brands=bmw&brands=audi&brands=mercedes
```

在 Node.js 中，以 express 为例，可以通过下面的方法获得上面 brands 参数的值：

```
app.get('/search', (req, res) => {
  console.log(req.query.brands); // 直接返回一个数组 ['bmw', 'audi', 'mercedes']
  ...
});
```

那么我们应该如何通过 URLSearchParams 来生成及解析这种标准参数形式呢？使用 append() 和 getAll() 方法就行：

```
let params = new URLSearchParams();
// 通过 append() 方法添加所有的选项值
params.append('brands', 'bmw');
params.append('brands', 'audi');
params.append('brands', 'mercedes');
const searchString = params.toString();

// 解析也同样简单
params = new URLSearchParams(searchString);
console.log(params.getAll('brands')); // 返回一个数组 ['bmw', 'audi', 'mercedes']
```

## 写在最后

URLSearchParams 无疑是前端做查询字符串拼接、特别是实现前端路由的好帮手。这里还要啰嗦一句，由于还在试验阶段，因此 Node.js、Chrome、Firefox 以及各种 Polyfill 类库的实现略有不同，例如：

```
const params = new URLSearchParams({
  k: '关键字',
  p: 1
});
console.log(params.toString());
```

在 Chrome 和 Node.js 上可以获得我们预期的结果，而在 FireFox 和 Polyfill 类库上则不支持 JSON 的构造函数，起初还差点害我出了一次生产事故，另外 UC 浏览器的实现也有不同（不支持 “?” 开头的字符串作为构造参数），因此大家在初期使用的时候要格外小心，不多说了，直接给出 URLSearchParams API 文档地址：

- [Node.js 文档](#)
- [Firefox 文档](#)
- [Polyfill Github 仓库及文档](#)



### 关于专栏

如果你喜欢这篇文章，就请关注我的专栏 《前端零栈》，在这里我们一起聊一聊前端技术和前端工程。

《前端零栈》 专栏开张后，非常每篇文章都被友商转载或入选周刊推荐，欢迎继续关注 我的专栏前端零栈 - 知乎专栏。

### 关于作者

**Henry**，就职于阿里巴巴南京研发中心，10 岁开始学习计算机编程，高二暑假获得江苏省青少年信息奥林匹克一等奖。2000 年开始自学 JavaScript 及网页制作，2006 年起正式开始从事前端开发工作，从此一干就是 10 多年。加入阿里巴巴前，曾在 SAP 中国研究院担任智慧交通大数据产品经理。


Github： MagicCube (Henry Li)

编辑于 2017-10-05

前端开发   JavaScript   HTML


▲ 赞同 54   ▼   ● 13 条评论   ➤ 分享   ❤ 喜欢   ★ 收藏   📄 申请转载   ...

### 文章被以下专栏收录



**前端零栈**  
让我们一起聊聊『前端人工智能』和『前端技术』


关注专栏



**阿里 CCO 体验技术专刊**  
阿里客户体验 CCO 技术部前端 / 全栈技术专刊


关注专栏

### 推荐阅读




是时候换个姿势使用  
JavaScript 数组了

技能树IT修真院



ES6 - 标签函数

HuminiOS



JavaScript

晓舟报告

13 条评论   ➡ 切换为时间排序

写下你的评论...

 柯琳   2017-09-23

上周就在mdn上琢磨这个api，当时就想着抄着mdn上的解释说明来水一篇文章....

👍 1



找兵头也是0月才刚用到来玉的一个线上项目里的，感觉非常靠谱！

👍 赞



冷静

2017-09-24

微信小程序生产环境使用会抛一个异常，也不知道是哪里不对，就改成手动解析了

👍 赞



熊杰

2017-09-25

一直用querystring

👍 3



hacke2

2017-09-25

[querystring-uti](#) 这个是我们团队出的，api 设计几乎一致

👍 2



Henry (作者) 回复 hacke2

2017-09-25

赞一个，虽然此时还是比较推崇标准

👍 赞



fluency03 回复 hacke2

2017-10-05

[npmjs.com/package/qs](#)

👍 赞



fluency03

2017-10-05

有个node包叫 qs: [npmjs.com/package/qs](#)

👍 赞



a cat

2017-10-09

iOS10.3才支持，上次用没有Plyfill没注意看下兼容性被坑了。。

👍 赞



roastwind

2018-01-24

楼主有说数组，如果是一个object呢，该怎么操作呢？

👍 赞



陈爱国

2019-05-27

如果一个url是: [zhuanlan.zhihu.com/#...](#)，使用urlParams.searchParams.append('key', 'value');最后获得的值是"[zhuanlan.zhihu.com/?...](#)"，为什么不把参数插入到后面呢？

👍 赞



哎哟不错哦 回复 陈爱国

2019-09-03

因为url格式就是这样的：<scheme>://<user>:<password>@<host>:<port>/<path>;<br><params>?<query>#<frag>

#<frag> 部分就是放在最后面的

👍 赞



淡定路过请无视

2019-12-16

马克一下 刚用到了

👍 赞

