



# nodejs获取GET和POST数据以及读取响应请求文件



jCodeLife



0.21 2020.09.06 19:38:31 字数 513 阅读 76

编辑文章

用户发送数据请求的时候，常见的方式是 `GET` 和 `POST` 请求，无论是发送 `GET` 还是 `POST`，其数据都在请求内容对象 `request` 中

## 1. 获取 GET 数据的三种方式

`get` 数据发送的时候都是在 `request.url` 中的，`request` 表示请求的内容对象，里面包括了很多请求相关的信息。

方法一：手动切

通过字符串中的 `split` 方法 将 `url` 字符串中的请求数据切成json格式

```
1 const http=require("http");//引入http模块
2 http.createServer(function(req,res){//通过http模块对象创建服务器
3     //req获取前台发的数据
4     var GET={};
5     if (req.url.indexOf('?')!=-1) {
6         var arr = req.url.split('?');
7         var url=arr[0];//地址
8         var arr2 = arr[1].split('&');
9         for (var i = 0; i < arr2.length; i++) {
10             var arr3 = arr2[i].split('=');
11             GET[arr3[0]]=arr3[1];
12         }
13     }else{
14         var url=req.url;
15         // GET={}
16     }
17     console.log(url,GET);
18     res.write("aaa");//响应写在页面上的内容
19     res.end();//结束响应
20 }).listen(8080);//要个端口并监听
```

方法二：使用 `querystring` 模块

`querystring` 跟 `http` 一样都是 `node` 自带的模块，通过 `querystring` 自带的 `parse` 方法，可以将查询字符串变成 `json` 形式

```
1 const http = require('http');
2 const querystring=require('querystring');
3 http.createServer(function(req,res){
4     var GET={};
5     if (req.url.indexOf('?')!=-1) {
6         var arr=req.url.split('?');
7         var url = arr[0];
8         GET=querystring.parse(arr[1]);//
9     }else{
10         var url=req.url;
11     }
12     console.log(GET);
13     res.write('aaaa');
14     res.end();
15 }).listen(8080);
```

方法三：使用 `url` 模块



jCodeLife

总资产32 (约2.44元)

一篇搞清AJAX及实现手动封装

阅读 73

Vue3为什么引入composition API

阅读 34

使用vite创建vue3项目

阅读 136

## 推荐阅读

如何理解JavaScript中的对象?

阅读 243

js 判断字符串中是否包含某个字符串

阅读 219

尤雨溪：全职做开源年薪百万，非科班程序员也能成功

阅读 74

Vue插件-饼状图

阅读 807

vue数据更新了，视图不更新，怎么解决?

阅读 304



```
3 var GET={};
4 http.createServer(function(req,res){
5   var obj=urllib.parse(req.url,true);
6   var url=obj.pathname;
7   var GET=obj.query;
8
9   console.log(url,GET);
10  res.write('aaaaa');
11  res.end();
12 }).listen(8080);
```

以上三种方式获取GET请求的数据，推荐使用第三种

## 2. 获取 POST 请求的数据

与 GET 数据相比，POST 数据量大，得分段。通过 `req.on('data',function(data){})` 监听，当有一段数据到达的时候执行回调，回调函数参数 `data` 为每段达到的数据。当数据全部到达时会触发 `req.on('end',function(){})` 里面的回调函数。可以通过 `querystring.parse(str)` 解析成我们想要的 POST 请求数据格式

```
1 var http=require('http');
2 var querystring=require('querystring');
3 http.createServer(function(req,res){
4   var str='';
5   var i=0;
6   req.on('data',function(data){
7     console.log(`第${++i}次收到数据`);
8     str+=data;
9   });//当有一段数据到达的时候
10  req.on('end',function(){
11    var POST=querystring.parse(str);
12    console.log(POST);
13  });//数据全部到达
14 }).listen(8080);
15
```

## 3. 文件的读写

通过 `fs` 模块完成文件的读取，里面有个 `readFile(要读取的文件, 回调函数)` 用于读取文件，读取结束执行回调

针对三种请求：请求文件或者get请求或者post请求，我们来搭建一个完整一点的服务器雏形

```
1 //引入相应模块
2 const http=require('http');//http协议相关
3 const fs=require('fs');//文件系统相关
4 const querystring=require('querystring');//查询字符串相关
5 const urllib=require('url');//处理url相关
6
7 var server=http.createServer(function(req,res){
8   //GET
9   var obj=urllib.parse(req.url,true);
10  var url=obj.pathname;
11  const GET=obj.query;
12
13  //POST
14  var str='';
15  var POST='';
16  req.on('data',function(data){
17    str+=data;
18  });
19  req.on('end',function(){
20    POST=querystring.parse(str);
21  });
22  // console.log(url,GET,POST)
23
24  //文件读写
```



```
30     res.write(data);
31   }
32   res.end();
33 });
34
35 });
36
37 server.listen(8080); // 要个端口并且监听该端口
```

以上就是nodejs如何获取GET和POST数据以及如何读取所请求文件并响应的相关内容



1人点赞 >



笔记 (2)



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



jCodeLife 书山有路勤为径，学海无涯苦作舟

总资产32 (约2.44元) 共写了13.6W字 获得334个赞 共25个粉丝



写下你的评论...

全部评论 0

只看作者

关闭评论

按时间倒序

按时间正序

被以下专题收入，发现更多相似内容

投稿管理

+ 收入我的专题



前段开发那些事儿

写下你的评论...

评论 0

赞 1

