



[首页](#) / [nodejs](#) / node.js post json格式数据到服务器的几种方法

node.js post json格式数据到服务器的几种方法

🕒 2014-07-17 📁 [nodejs](#) 👁 36389

分享到: [QQ空间](#) [新浪微博](#) [腾讯微博](#) [人人网](#) [微信](#)



json格式被越来越多的开发者说青睐,我们常常在接口定义时使用这种格式参数进行数据交换.

今天主要给大家从繁到简的几个node.js下使用的提交 json个数参数的方式方法.

下面示例中出现的参数都做了处理,拷贝代码后需要更改这些参数同时自己写一个接收json格式的api,请求成功后再返回json格式

(一) node.js 原生自带http模块,可以解决基于http协议下的请求及回发,执行效率高,但是好多东西需要开发人员自己动手来实现,看下面代码

```

var http=require('http');

var body = {
  "data":{
    "channel" : "aaa",
    "appkey" : "bbb"
  },
  "sign" : "22334455",
  "token" : "bb667788"
};

var bodyString = JSON.stringify(body);

var headers = {
  'Content-Type': 'application/json',
  'Content-Length': bodyString.length
};

var options = {
  host: '127.0.0.1',
  port: 3005,
  path: '/Config',
  method: 'POST',
  headers: headers
};

var req=http.request(options,function(res){
  res.setEncoding('utf-8');

  var responseString = '';

  res.on('data', function(data) {
    responseString += data;
  });

  res.on('end', function() {
    //这里接收的参数是字符串形式,需要格式化成json格式使用
    var resultObject = JSON.parse(responseString);
    console.log('-----resBody-----',resultObject);
  });

  req.on('error', function(e) {
    // TODO: handle error.
    console.log('-----error-----',e);
  });
});
req.write(bodyString);
req.end();

```

http模块比较原始,请求参数里我们要手动指定请求头类型,头长度,请求方式,主机头和端口等.

当我们将json格式参数post上去以后,得到的response 对象需要我们在 data 事件上手动处理获取到的数据, end 事件表示接收数据流已结束, error 事件是当回发数据发生错误时将自动触发此事件响应函数

请求结果:

```
zhangzhi@moke:~/code/test$ node request.js
-----resBody----- { error_code: 0,
  error: '',
  data: { energy_list: [ [Object] ] },
  server_time: 1405575465 }
```

下面我们介绍第二种相对简单的方式来post json格式参数.

(二)

大名鼎鼎的 **request** 模块上场了,这个模块帮我们做了很多我们不关心的东西,比如请求后得到的 **response** 对象,我们将不会非常繁琐的去手动捕获 **data** ,**end** ,**error** 事件.取而代之的是回调函数里直接将我们关心的回发数据放到了一个变量里,看下面的代码:

此模块并发node.js 原生自带模块,首先需要 `npm install request` ,下面代码的参数同样做了处理

```
var request=require('request');

var options = {
  headers: {"Connection": "close"},
  url: 'http://127.0.0.1:3005/Config',
  method: 'POST',
  json:true,
  body: {data:{channel : "aaa",appkey : "bbb"},sign : "ccc",token : "ddd"}
};

function callback(error, response, data) {
  if (!error && response.statusCode == 200) {
    console.log('----info-----',data);
  }
}

request(options, callback);
```

代码明显比第一种方式少了很多,callback 里面的 **data** 参数就是我们请求路径后得到的内容,而且我们在 **options** 中指定了 **json:true** ,所以请求和回发的数据自动转变成了 **json** 对象.我们看下这次的运行结果:

```
zhangzhi@moke:~/code/test$ node request.js
----info----- { error_code: 0,
  error: '',
  data: { energy_list: [ [Object] ] },
  server_time: 1405576317 }
```

下面还有一种更简单的方法和大家分享

(三)

也是一个第三方模块,叫做 **request-json**

看名字就非常容易理解,有关**json** 格式参数的请求模块,使用也非常方便.

首先 `npm install request-json` 安装此模块,定义好 **json**请求参数对象,见下面代码:

```
request = require('request-json');
var client = request.newClient('http://127.0.0.1:3005/');

var data = {data:{channel : "aaa",appkey : "bbb"},sign : "4444",token : "555"};
client.post('Config', data, function(err, res, body) {
  console.log(res.statusCode,body);
});
```

返回的body 也是自动序列化成json对象,见运行结果

```
zhangzhi@moke:~/code/test$ node request.js
200 { error_code: 0,
  error: '',
  data: { energy_list: [ [Object] ] },
  server_time: 1405576675 }
```

前面的 200 是 res.statusCode 回发状态值,接着输出的是我们得到的json对象,无论从使用还是理解上,都是最简单的一种方法.

你可能感兴趣

- [微信第三方活动 关注微信公众号才能参与](#)
- [node-webkit 项目中的 package.json 格式](#)
- [mac 系统下 如何 彻底 删除 node.js](#)
- [ejs母版页让你的node.js应用开发更快更有效- 一介布衣](#)
- [node.js 实现一个简单的登录拦截器](#)

本文由 [一介布衣](#) 创作, 采用 [知识共享署名 3.0](#) 中国大陆许可协议。
可自由转载、引用, 但需署名作者且注明文章出处。

本站部署于「[UCloud云计算](#)」。
右侧微信扫描「打赏」,感谢支持!



北大青鸟	吃什么生精	充气娃娃专卖	移民美国
页游	移民美国的条件		男士去眼袋方法

[node.js](#) | [post_json](#) | [request](#) | [request-json](#)



Copyright © 2015 一介布衣

京ICP备14006909号-1