

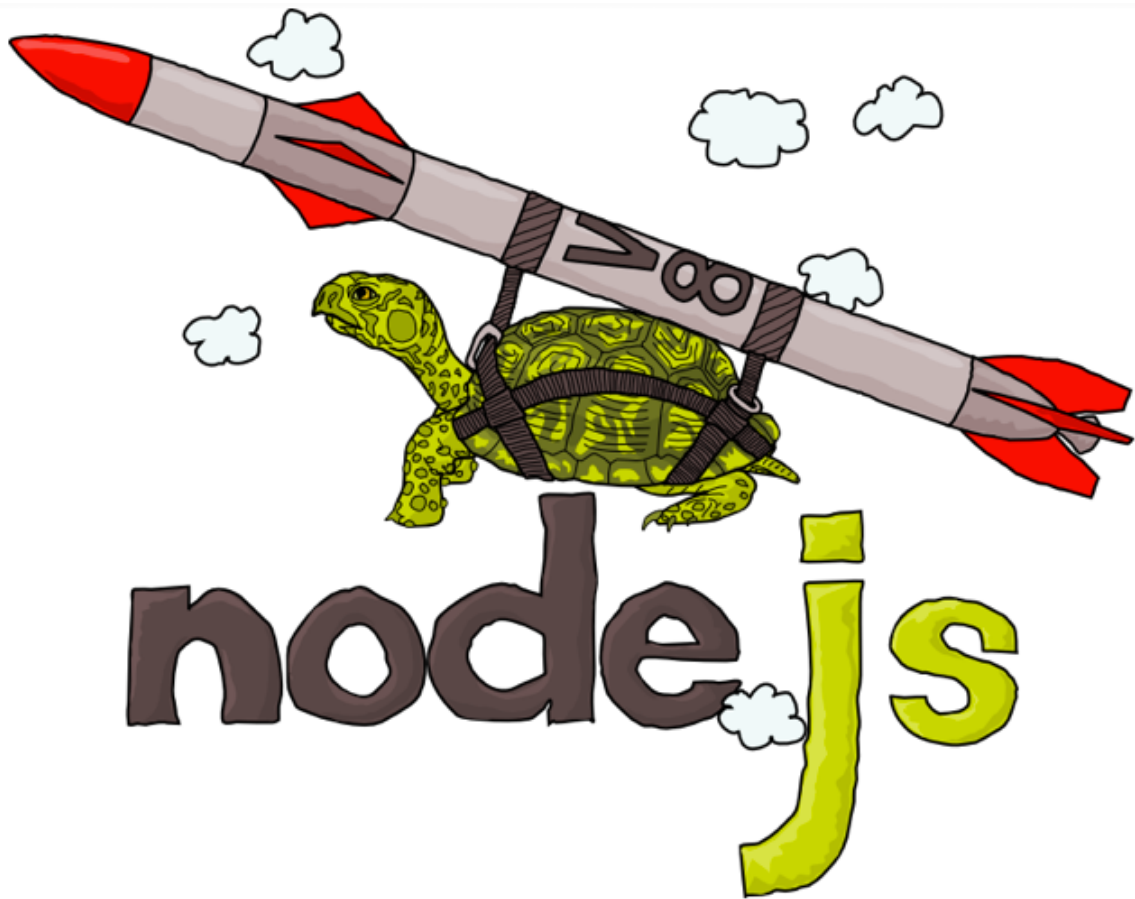


[首页](#) / [nodejs](#) / node.js 下依赖Express 实现post 4种方式提交参数

## node.js 下依赖Express 实现post 4种方式提交参数

© 2014-12-31 nodejs 33518

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



上面这个图好有意思啊,哈哈, v8威武啊....

在2014年的最后一天和大家分享关于node.js 如何提交4种格式的post数据.

上上一篇说到了关于[http协议里定义的4种常见数据的post方法](#),详细介绍请点击查看.

分别是这四种:

www-form-urlencoded,

form-data,

application/json,

text/xml

Express 依赖 `bodyParser` 对请求的包体进行解析, 默认支持: `application/json`, `application/x-www-form-urlencoded`, `multipart/form-data`. 可惜对 `xml` 没有支持, 需要自己代码来实现, 下面我们一一简单介绍.

### (一) `www-form-urlencoded`

`http` 默认的 `post` 请求是这种方式, 注意这是是默认的提交方式, 比如你写一个 `<form>....<input type="submite" /></form>` `form` 表单, 里面的 `submite` 按钮默认就是这种 `www-form-urlencoded` 方式提交的.

`node.js` 下使用 `Express` 如何接收这种提交方式. 需要 `body-parse` 插件支持, Express 和 `body-parse` 的依赖关系 在 4.0 以下版本和 4.0 以上版本是不同的, 可以搜索本站 `body-parse` 关键词查看.

```
var express = require('express');
var bodyParser = require('body-parser');

var app = express();
var server = require('http').createServer(app);

app.use(bodyParser.urlencoded({
  extended: true
}));

var PORT = process.env.PORT || 8002;
server.listen(PORT);
```

上面搭建起一个简单的 `server`, `node app.js` 运行起来, 现在是可以接收 `post www-form-urlencoded` 类型参数的.

我们在上面的代码中加上一个路由.

```
app.post('/urlencoded', function (req, res) {
  console.log(req.body);
  res.send(req.body);
});
//注意, 这个路由应该是加在 var PORT=process.env.PORT || 8002; 上面
```

现在用 `postman` 模拟一个 `post` 请求,

红框指定了提交方式是 `www-form-urlencoded` 方式

http://localhost:8002/urlencoded POST

form-data x-www-form-urlencoded raw

eee	111	✕
fff	222	✕
Key	Value	

Send Preview Add to collection

Body Cookies (2) Headers (5) STATUS 200 OK TIME 407 ms

Pretty Raw Preview JSON XML

```
1 {  
2   "eee": "111",  
3   "fff": "222"  
4 }
```

可以看到 200 状态,成功识别了请求参数并且返回来.

## (二) form-data

也是比较常用的提交数据的方式,和上面一种不同的是form表单里需要有 `enctype` 标识.

比如我们在上传文件的时候,必须在form标签里做这样的标识 `enctype="multipart/form-data"`

而这种提交方式一般用在文件上传,所以 在node.js 里处理这类表单还需要 中间件 `connect-multiparty` ,他是专门处理此类 post数据相关的依赖包.github官方地址: <https://github.com/andrewrk/connect-multiparty>

```
npm install connect-multiparty
```

使用也非常简单,上面的代码不变,只需要在你的相关路由里指定此中间件,它就可以帮你处理请求.

```
var multipart = require('connect-multiparty');  
var multipartMiddleware = multipart();  
  
app.post('/formdata',multipartMiddleware, function (req, res) {  
  console.log(req.body);  
  res.send(req.body);  
});
```

我用postman模拟,没有办法上传文件,只模拟表单...参数同样直接返回客户端,

http://localhost:8002/formdata POST

form-data x-www-form-urlencoded raw

ccc	111	Text	✕
ddd	222	Text	✕
Key	Value	Text	

Send



Preview

Add to collection

Body Cookies (2) Headers (5)

STATUS 200 OK TIME 19 ms

Pretty Raw Preview



JSON XML

```
1 {
2   "ccc": "111",
3   "ddd": "222"
4 }
```

### (三) application/json

一般api喜欢此类提交方式,数据使用方法,节省带宽.

bodyParser 支持此类参数解析.

注意: 在提交之前需要指定http 请求头为 content-type=application/json

代码如下:

```
var express = require('express');
var bodyParser = require('body-parser');
var multipart = require('connect-multiparty');
var multipartMiddleware = multipart();

//var morgan = require('morgan');
var app = express();
var server = require('http').createServer(app);

app.use(bodyParser.json({limit: '1mb'})); //这里指定参数使用 json 格式
app.use(bodyParser.urlencoded({
  extended: true
}));

app.post('/json', function (req, res) {
  console.log(req.body);
  res.send(req.body);
});

var PORT = process.env.PORT || 8002;
server.listen(PORT);
```

注意上面的配置参数,指定参数使用json格式.

```
app.use(bodyParser.json({limit: '1mb'}));
```

http://localhost:8002/json POST

Content-Type application/json

Header Value

form-data x-www-form-urlencoded raw **JSON**

```
1 {
2   "ccc": "111",
3   "ddd": "222"
4 }
```

Send Preview Add to collection

body Cookies (2) Headers (5) **STATUS** 200 OK **TIME** 238 ms

Pretty Raw Preview JSON XML

```
1 {
2   "ccc": "111",
3   "ddd": "222"
4 }
```

#### (四) text/xml

这种请求类型不是特别常见, **body-parse**默认也不解析这种数据格式,目前腾讯微信平台在使用这种数据交换格式.**node.js** 在 **express** 如何解析这种格式,没有好的办法,只能自己用代码处理,把请求体参数按照字符串读取出来,然后使用 **xml2json** 包把字符串解析成**json**对象,使用起来就方便多了.

注意:

我们还是要使用 **body-parse** 得到字符串,然后再转化.

**xml**格式请求需要指定 **http** 请求头 **content-type=text/xml**

利用**req**上定义的事件 **data** 来获取**http**请求流, **end** 事件结束请求流的处理.

利用 **xml2json** 把上面得到的请求参数流(我们直接转化为字符串)转化为 **json** 对象.

```
npm install xml2json
```

代码如下:

```
var express = require('express');
var bodyParser = require('body-parser');
var xml2json=require('xml2json');

var app = express();
var server = require('http').createServer(app);

app.use(bodyParser.urlencoded({
  extended: true
}));

app.post('/xml', function (req, res) {
  req.rawBody = '';
  var json={};
  req.setEncoding('utf8');

  req.on('data', function(chunk) {
    req.rawBody += chunk;
  });
  req.on('end', function() {

    json=xml2json.toJson(req.rawBody);

    res.send(JSON.stringify(json));
  });
});

var PORT = process.env.PORT || 8002;
server.listen(PORT);
```

我们最后也是把提交的xml格式字符串转化成json对象输出到客户端.

http://localhost:8002/xml

POST

content-type

text/xml

×

Manage presets

Header

Value

form-data

x-www-form-urlencoded

raw

XML

1 <methodcall>

2 <a>aaa</a>

3 </methodcall>

Send

Preview

Add to collection

Body

Cookies (2)

Headers (5)

STATUS 200 OK

TIME 235 ms

Pretty

Raw

Preview

JSON

XML


1 {"methodcall":{"a":"aaa"}}

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!



私密脱毛

充气娃娃专卖

页游

吃什么生精

微整形

北大青鸟

不够硬进不去