<https://www.cnblogs.com/htoooth/p/7242217.html>

# [htoooth](https://www.cnblogs.com/htoooth/)

## 日拱一卒,功不唐捐

* [**博客园**](https://www.cnblogs.com/)
* [**首页**](https://www.cnblogs.com/htoooth/)
* [**新随笔**](https://i.cnblogs.com/EditPosts.aspx?opt=1)
* [**联系**](https://msg.cnblogs.com/send/htoooth)
* [**订阅**](https://www.cnblogs.com/htoooth/rss)
* [**管理**](https://i.cnblogs.com/)

随笔 - 94  文章 - 1  评论 - 23

# [$.ajax 中的contentType](https://www.cnblogs.com/htoooth/p/7242217.html)

作者：[@htoooth](https://github.com/htoooth)  
本文为作者原创，转载请注明出处：<https://www.cnblogs.com/htoooth/p/7242217.html>

#### 目录

[**正文**](https://www.cnblogs.com/htoooth/p/7242217.html#_caption_0)  
[**参考文章**](https://www.cnblogs.com/htoooth/p/7242217.html#_caption_1)

# $.ajax 中的contentType

在 cnodejs.org 论坛中有一个[问题](https://cnodejs.org/topic/5976df443f0ab31540ed4d3e)，让我也很奇怪，说是 $.ajax 设置数据类型 applicaiton/json之后，服务器端(express)就拿不到数据，好奇之下，就去翻了翻资料，发现了一个自己也没有注意到的事。

于是有了这篇文章。

## 正文

$.ajax contentType 和 dataType , contentType 主要设置你发送给服务器的格式，dataType设置你收到服务器数据的格式。

在http 请求中，get 和 post 是最常用的。在 jquery 的 ajax 中， contentType都是默认的值：application/x-www-form-urlencoded，这种格式的特点就是，name/value 成为一组，每组之间用 & 联接，而 name与value 则是使用 = 连接。如： wwwh.baidu.com/q?key=fdsa&lang=zh 这是get , 而 post 请求则是使用请求体，参数不在 url 中，在请求体中的参数表现形式也是: key=fdsa&lang=zh的形式。

键值对这样组织在一般的情况下是没有什么问题的，这里说的一般是，不带嵌套类型JSON，也就是 简单的JSON，形如这样：

{

a: 1,

b: 2,

c: 3

}

但是在一些复杂的情况下就有问题了。 例如在 ajax 中你要传一个复杂的 json 对像，也就说是对象嵌数组，数组中包括对象，兄果你这样传：

{

data: {

a: [{

x: 2

}]

}

}

这个复杂对象， application/x-www-form-urlencoded 这种形式是没有办法将复杂的 JSON 组织成键值对形式(当然也有方案这点可以[参考](https://stackoverflow.com/questions/6243051/how-to-pass-an-array-within-a-query-string) ) ,你传进去可以发送请求，但是服务端收到数据为空， 因为 ajax 没有办法知道怎样处理这个数据。

这怎么可以呢？

聪明的程序员发现 http 还可以自定义数据类型，于是就定义一种叫 application/json 的类型。这种类型是 text ， 我们 ajax 的复杂JSON数据，用 JSON.stringify序列化后，然后发送，在服务器端接到然后用 JSON.parse 进行还原就行了，这样就能处理复杂的对象了。

$.ajax({

dataType: 'json',

contentType: 'application/json',

data: JSON.stringify({a: [{b:1, a:1}]})

})

这样你就可以发送复杂JSON的对象了。像现在的 restclient 都是这样处理的。