

JSON

进入词条

搜索词条

<u>帮助</u>

城市百科 文化

首页

分类频道 ▼

特色百科 ▼

玩转百科 ▼

百科用户 ▼

百科校园

百科合作

□ 手机百科

2 个人中心

★ 收藏

1 3614 **2** 1115

JSON 🗪 编辑



本词条缺少信息栏、补充相关内容使词条更完整、还能快速升级、赶紧来编辑吧!

JSON(JavaScript Object Notation) 是一种轻量级的数据交换格式。它基于JavaScript(Standard ECMA-262 3rd Edition - December 1999)的一个子集。 JSON采用完全独立于语言的文本格式,但是也使用了类似于 C语言家族的习惯(包括C, C++, C#, Java, JavaScript, Perl, Python等)。这些特性使JSON成为理想的数据交 换语言。易于人阅读和编写,同时也易于机器解析和生成。

目录

1 JSON 语法

- JSON 语法规则
- JSON 名称/值对
- JSON 值
- 2 基础结构
- 3 基础示例
 - 名称 / 值对

JSON 语法是 JavaScript 语法的子集。 [1]

■ 表示数组

4 格式应用

- 赋值给变量
- 访问数据
- 修改数据
- 换回字符串
- 5 具体形式

6 概念比较

- 和XML的比较
- 实例比较

7 校验工具

- 前言
- 功能简介

1 JSON 语法

编辑

JSON 语法规则

JSON 语法是 JavaScript 对象表示语法的子集。

- 数据在名称/值对中
- 数据由逗号分隔
- 花括号保存对象
- 方括号保存数组

JSON 名称/值对

JSON 数据的书写格式是: 名称/值对。

名称/值对组合中的名称写在前面(在双引号中),值对写在后面(同样在双引号中),中间用冒号隔开:

"firstName":"John"

这很容易理解,等价于这条 JavaScript 语句:

1 firstName="John"

JSON 值

JSON 值可以是:

- 数字(整数或浮点数)
- 字符串(在双引号中)
- 逻辑值(true 或 false)
- 数组(在方括号中)
- 对象(在花括号中)
- null

2 基础结构

JSON ^[1]结构有两种结构 ^[2]

编辑

JSON图册

词条统计

浏览次数: 1217064次 编辑次数: 100次 历史版本 最近更新: 2014-07-07 创建者: realdodo



百科消息:

- 百度百科最明星——全能偶像鹿晗
- 【詹天佑故居上线】还原真实的詹天佑
- 赤火虹口革命地,中共四大纪念馆上线
- 【百科新玩法】我来建设玉溪城市百科
- 张嘉佳睡前故事《让我留在你身边》

推广链接

ison 货到付款3折起 「亚..

《json》,全球著名中文网上书店;《json》, 货到付款!亚马逊Z.cn... www.amazon.cn

json简单说就是javascript中的对象和数组,所以这两种结构就是对象和数组两种结构,通过这两种结构可以

- 1、对象:对象在js中表示为"{}"括起来的内容,数据结构为 {key: value,key: value,...}的键值对的结构,在面向对象的语言中,key为对象的属性,value为对应的属性值,所以很容易理解,取值方法为 对象.key 获取属性值,这个属性值的类型可以是 数字、字符串、数组、对象几种。
- 2、数组:数组在js中是中括号"[]"括起来的内容,数据结构为 ["java","javascript","vb",...],取值方式和所有语言中一样,使用索引获取,字段值的类型可以是数字、字符串、数组、对象几种。

经过对象、数组2种结构就可以组合成复杂的数据结构了。

3 基础示例

编辑

简单地说^[2],JSON 可以将 JavaScript 对象中表示的一组数据转换为字符串,然后就可以在函数之间轻松地传递这个字符串,或者在异步应用程序中将字符串从 Web 客户机传递给服务器端程序。这个字符串看起来有点儿古怪,但是JavaScript很容易解释它,而且 JSON 可以表示比"名称 / 值对"更复杂的结构。例如,可以表示数组和复杂的对象,而不仅仅是键和值的简单列表。

名称 / 值对

按照最简单的形式,可以用下面这样的 JSON 表示"名称 / 值对":

1 {"firstName":"Brett"}

这个示例非常基本,而且实际上比等效的纯文本"名称 / 值对"占用更多的空间:

1 firstName=Brett

但是,当将多个"名称 / 值对"串在一起时,JSON 就会体现出它的价值了。首先,可以创建包含多个"名称 / 值对"的 记录,比如:

1 {"firstName":"Brett","lastName":"McLaughlin","email":"aaaa"}

从语法方面来看,这与"名称 / 值对"相比并没有很大的优势,但是在这种情况下 JSON 更容易使用,而且可读性更好。例如,它明确地表示以上三个值都是同一记录的一部分;花括号使这些值有了某种联系。

表示数组

当需要表示一组值时,JSON 不但能够提高可读性,而且可以减少复杂性。例如,假设您希望表示一个人名列表。在XML中,需要许多开始标记和结束标记;如果使用典型的名称 / 值对(就像在本系列前面文章中看到的那种名称 / 值对),那么必须建立一种专有的数据格式,或者将键名称修改为 person1-firstName这样的形式。

如果使用 JSON, 就只需将多个带花括号的记录分组在一起:

```
1  {
2   "people":[
3   {"firstName":"Brett","lastName":"McLaughlin","email":"aaaa"},
4   {"firstName":"Jason","lastName":"Hunter","email":"bbbb"},
5   {"firstName":"Elliotte","lastName":"Harold","email":"cccc"}
6   ]
7  }
```

这不难理解。在这个示例中,只有一个名为 people的变量,值是包含三个条目的数组,每个条目是一个人的记录,其中包含名、姓和电子邮件地址。上面的示例演示如何用括号将记录组合成一个值。当然,可以使用相同的语法表示多个值(每个值包含多个记录):

```
{"programmers":[
    {"firstName":"Brett","lastName":"McLaughlin","email":"aaaa"},
    {"firstName":"Jason","lastName":"Hunter","email":"bbbb"},
    {"firstName":"Elliotte","lastName":"Harold","email":"cccc"}
 5
     ],
 6
     "authors":[
    {"firstName":"Isaac","lastName":"Asimov","genre":"sciencefiction"},
    {"firstName":"Tad","lastName":"Williams","genre":"fantasy"},
     {"firstName": "Frank", "lastName": "Peretti", "genre": "christianfiction"}
10
     ],
     "musicians":[
11
     {"firstName":"Eric","lastName":"Clapton","instrument":"guitar"},
12
13
    {"firstName":"Sergei","lastName":"Rachmaninoff","instrument":"piano"}
14
    ]}
```

这里最值得注意的是,能够表示多个值,每个值进而包含多个值。但是还应该注意,在不同的主条目(programmers、authors 和 musicians)之间,记录中实际的名称 / 值对可以不一样。JSON 是完全动态的,允许在 JSON 结构的中间改变表示数据的方式。

在处理 JSON 格式的数据时,没有需要遵守的预定义的约束。所以,在同样的数据结构中,可以改变表示数据的方式,甚至可以以不同方式表示同一事物。

掌握了 JSON 格式之后,在 JavaScript 中使用它就很简单了。JSON 是 JavaScript 原生格式,这意味着在 JavaScript 中处理 JSON 数据不需要任何特殊的 API 或工具包。

赋值给变量

例如,可以创建一个新的 JavaScript 变量,然后将 JSON 格式的数据字符串直接赋值给它:

[3]

```
varpeople={"programmers":[{"firstName":"Brett","lastName":"McLaughlin","email":"aaa
     {"firstName":"Jason","lastName":"Hunter","email":"bbbb"},
    {"firstName":"Elliotte","lastName":"Harold","email":"cccc"}
 4
     ],
     "authors":[
    {"firstName":"Isaac","lastName":"Asimov","genre":"sciencefiction"},
     {"firstName":"Tad","lastName":"Williams","genre":"fantasy"},
     {"firstName": "Frank", "lastName": "Peretti", "genre": "christianfiction"}
 9
     ],
10
     "musicians":[
     {"firstName": "Eric", "lastName": "Clapton", "instrument": "guitar"},
11
    {"firstName": "Sergei", "lastName": "Rachmaninoff", "instrument": "piano"}
12
13
    ]}
```

这非常简单; people包含前面看到的 JSON 格式的数据。但是,这还不够,因为访问数据的方式似乎还不明显。

访问数据

尽管看起来不明显,但是上面的长字符串实际上只是一个数组;将这个数组放进 JavaScript变量之后,就可以很轻松地访问它。实际上,只需用点号表示法来表示数组元素。所以,要想访问 programmers 列表的第一个条目的姓氏,只需在 JavaScript 中使用下面这样的代码:

people.programmers[0].lastName;

注意,数组索引是从零开始的。所以,这行代码首先访问 people变量中的数据;然后移动到称为 programmers的条目,再移动到第一个记录([0]);最后,访问 lastName键的值。结果是字符串值 "McLaughlin"。

下面是使用同一变量的几个示例。

```
people.authors[1].genre//Valueis"fantasy"
people.musicians[3].lastName//Undefined.Thisreferstothefourthentry,andthereisn'tone
people.programmers[2].firstName//Valueis"Elliotte"
```

利用这样的语法,可以处理任何 JSON 格式的数据,而不需要使用任何额外的 JavaScript 工具包或 API。

修改数据

正如可以用点号和方括号访问数据,也可以按照同样的方式轻松地修改数据:

```
people.musicians[1].lastName="Rachmaninov";
```

在将字符串转换为 JavaScript 对象之后,就可以像这样修改变量中的数据。

换回字符串

最终结论是,如果要处理大量 JavaScript 对象,那么 JSON 是一个好选择,这样就可以轻松地将数据转换为可以在请求中发送给服务器端程序的格式。

5 具体形式

编辑

编辑

- 1、对象是一个无序的"'名称/值'对"集合。
- (1) 一个对象以"{"(左括号)开始,"}"(右括号)结束。
- (2) 每个"名称"后跟一个":"(冒号);
- (3) "'名称/值' 对"之间使用","(逗号)分隔。(如图所示,图中表示数据的方式是类似非确定性自动机的形式,没学过编译原理的人,可能理解起来困难点,实际上也是正则表达式的形式。下同)

```
object string value
```

例子:表示人的一个对象:

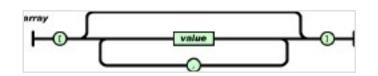
```
1 {
2 "姓名":"大憨",
3 "年龄":24
4 }
```

- 2、数组是值(value)的有序集合。
- (1) 一个数组以"["(左中括号)开始,"]"(右中括号)结束。
- (2) 值之间使用","(逗号)分隔。

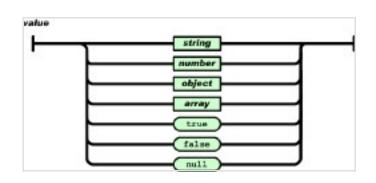
例子: 一组学生

```
1 {
2 "学生":
3 [
4 {"姓名":"小明","年龄":23},
5 {"姓名":"大憨","年龄":24}
6 ]
7 }
```

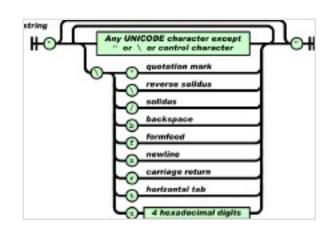
说明:此Json对象包括了一个学生数组,而学生数组中的值又是两个Json对象。



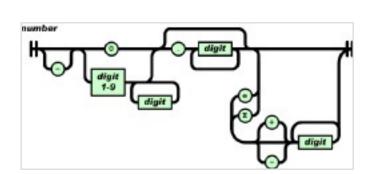
3、值(value)可以是双引号括起来的字符串(string)、数值(number)、true、false、 null、对象(object)或者数组(array)。这些结构可以嵌套。



4、字符串(string)是由双引号包围的任意数量Unicode字符的集合,使用反斜线转义。一个字符(character)即一个单独的字符串(character string)。字符串(string)与C或者Java的字符串非常相似。



5、数值(number)也与C或者Java的数值非常相似。除去未曾使用的八进制与十六进制格式。除去一些编码细节。



6 概念比较

编辑

和XML的比较

可读性

JSON和XML的可读性可谓不相上下,一边是简易的语法,一边是规范的标签形式,很难分出胜负。

可扩展性

XML天生有很好的扩展性,JSON当然也有,没有什么是XML可以扩展而JSON却不能扩展的。不过JSON在 Javascript主场作战,可以存储Javascript复合对象,有着xml不可比拟的优势。

编码难度

XML有丰富的编码工具,比如Dom4j、JDom等,JSON也有提供的工具。无工具的情况下,相信熟练的开发人员一样能很快的写出想要的xml文档和JSON字符串,不过,xml文档要多很多结构上的字符。

解码难度

XML的解析方式有两种:

一是通过文档模型解析,也就是通过父标签索引出一组标记。例如:

xmlData.getElementsByTagName("tagName"),但是这样是要在预先知道文档结构的情况下使用,无法进行通用的封装。

另外一种方法是遍历节点(document 以及 childNodes)。这个可以通过递归来实现,不过解析出来的数据仍旧是形式各异,往往也不能满足预先的要求。

凡是这样可扩展的结构数据解析起来一定都很困难。

JSON也同样如此。如果预先知道JSON结构的情况下,使用JSON进行数据传递简直是太美妙了,可以写出很实用美观可读性强的代码。如果你是纯粹的前台开发人员,一定会非常喜欢JSON。但是如果你是一个应用开发人员,就不是那么喜欢了,毕竟xml才是真正的结构化标记语言,用于进行数据传递。

而如果不知道JSON的结构而去解析JSON的话,那简直是噩梦。费时费力不说,代码也会变得冗余拖沓,得到的结果也不尽人意。但是这样也不影响众多前台开发人员选择JSON。因为json.js中的toJSONString()就可以看到JSON的字符串结构。当然不是使用这个字符串,这样仍旧是噩梦。常用JSON的人看到这个字符串之后,就对JSON的结构很明了了,就更容易的操作JSON。

以上是在Javascript中仅对于数据传递的xml与JSON的解析。在Javascript地盘内,JSON毕竟是主场作战, 其优势当然要远远优越于xml。如果JSON中存储Javascript复合对象,而且不知道其结构的话,我相信很多程序 员也一样是哭着解析JSON的。

除了上述之外,JSON和XML还有另外一个很大的区别在于有效数据率。JSON作为数据包格式传输的时候具有更高的效率,这是因为JSON不像XML那样需要有严格的闭合标签,这就让有效数据量与总数据包比大大提升,从而减少同等数据流量的情况下,网络的传输压力 [4]。

实例比较

XML和JSON都使用结构化方法来标记数据,下面来做一个简单的比较。

用XML表示中国部分省市数据如下:

```
<?xmlversion="1.0"encoding="utf-8"?>
    <country>
    <name>中国</name>
    ovince>
    <name>黑龙江</name>
     <cities>
    <city>哈尔滨</city>
 7
    <city>大庆</city>
9
    </cities>
    </province>
10
     ovince>
11
12
    <name>广东</name>
13
    <cities>
    <city>广州</city>
14
    <city>深圳</city>
15
    <city>珠海</city>
16
17
    </cities>
    </province>
18
     19
    <name>台湾</name>
20
21
    <cities>
    <city>台北</city>
22
    <city>高雄</city>
23
24
    </cities>
25
    </province>
    ovince>
26
27
    <name>新疆</name>
28
    <cities>
    <city>乌鲁木齐</city>
29
30
    </cities>
31
    </province>
32
    </country>
```

用JSON表示如下:

```
1 {
2 "name":"中国",
3 "province":[
```

```
4
    {
    "name":"黑龙江",
    "cities":{
    "city":["哈尔滨","大庆"]
 9
    },
10
    "name":"广东",
11
    "cities":{
12
    "city":["广州","深圳","珠海"]
13
14
    }
15
    },
16
    "name":"台湾",
17
18
    "cities":{
    "city":["台北","高雄"]
19
20
    }
21
    },
22
23
    "name":"新疆",
    "cities":{
24
    "city":["乌鲁木齐"]
25
26
27
    }
28
29
    }
```

编码的可读性,xml有明显的优势,毕竟人类的语言更贴近这样的说明结构。json读起来更像一个数据块,读起来就比较费解了。不过,我们读起来费解的语言,恰恰是适合机器阅读,所以通过json的索引.province[0].name就能够读取"黑龙江"这个值。

编码的手写难度来说,xml还是舒服一些,好读当然就好写。不过写出来的字符JSON就明显少很多。去掉空白制表以及换行的话,JSON就是密密麻麻的有用数据,而xml却包含很多重复的标记字符。

7 校验工具

编辑

前言

JSON格式取代了xml给网络传输带来了很大的便利,但是却没有了xml的一目了然,尤其是json数据很长的时候,我们会陷入繁琐复杂的数据节点查找中。

但是国人的一款在线工具 BeJson 给众多程序员带来了一阵凉风。

功能简介

1 JSON格式化校验

很多人在得到JSON数据后,一时没有办法判断JSON数据格式是否正确,是否少或多符号而导致程序不能解析,这个功能正好能帮助大家来完成JSON格式的校验。

2 JSON视图

想必很多程序员都会遇到当找一个节点的时候,会发现如果直接对着一行行数据无从下手,就算知道哪个位置,还要一个节点一个节点的往下找,万一一不留神又得从头开始找的麻烦事。

有了这个功能,一切JSON数据都会变成视图格式,一目了然,什么对象下有多少数组,一个数组下有多少对象。

这个功能非常实用。不光有视图功能还有格式化、压缩、转义、校验功能。总之很强大。

3 压缩转义

程序员在写JSON语句测试用例的时候,很多时候为了方便直接写了个JSON字符串做测试,但是又陷入了无止境的双引号转义的麻烦中。这款功能集压缩、转义于一身,让你在写测试用例的时候,如鱼得水。

4 JSON在线编辑器

如果你现在的电脑刚巧没有装你所熟悉的编辑器,如果你想针对拿到的JSON数据的某个节点做数据修改时,这个功能可以满足你的需求。

5 在线发送JSON数据

大家都知道,JSON用的最多的还是web项目的开发,那你要测试一个接口是否能准确的接受JSON数据,那你就得写一个页面发送JSON字符串,重复的做着这件事。随着这个功能的横空出世,你可以摆脱写测试页面了,因为这个功能可以将指定的JSON数据发送指定的url,方便吧。

6 JSON着色

很多人在写文档时,总希望文档能一目了然,但是面对着白底黑字的JSON数据总是提不起精神没关系,使用这个功能,所有的关键字都会被着色,数据结构一目了然。

7 JSON-XML互转

顾名思义,将JSON格式的数据转化成XML格式、或者XML格式的数据转化成JSON格式,一切都不是问题。

8 JSON-VIEW

JSON查看实用工具,在开发过程中(windows平台)可以对JSON数据进行格式化和视图显示。

9 它和xml一样都是一种数据交换格式

词条图册

更多图册▶



1/1 < >

WEB技术		<i>!</i> ^
HTML	XHTML - HTML 5 - CSS - TCP/IP	
XML	XML • XSL • XSLT • XSL-FO • XPath • XPointer • XLink • DTD • XML Schema • DOM • XForms • SOAP • WSDL • RDF • RSS • WAP • Web Services	
Web脚本	JavaScript - HTML DOM - DHTML - VBScript - AJAX - jQuery - JSON - E4X - WMLScript	
Serv脚本	SQL = ASP = ADO = PHP	
.NET	Microsoft.NETNET Mobile	
多媒体	SMIL • SVG	

参考资料

- 1. ★ JSON 教程 🛂 . W3Cschool 菜鸟教程 [引用日期2014-01-21] .
- 2. ★ JSON 🛂 . 李岩的博客 [引用日期2013-04-23] .
- 3. **☆ java解析JSON操作说明 ☑** . 中国Linux联盟 [引用日期2012-11-29] .
- 4. ★ JSON代码格式化高亮 | 格式校验工具 aTool在线工具 🗹 . 在线工具 aTool [引用日期2013-12-23] .

词条标签: ajax web开发 XML web web技术 AJAX , Web 2.0 , JSON , Web开发 , JSON-VIEW



新手上路

成长任务 编辑入门 编辑规则 百科术语



我有问题

 常见问题
 我要提问

 参加讨论
 意见反馈



投诉建议

举报不良信息 未通过词条申诉 投诉侵权信息 封禁查询与解封