

JSON——开发 AJAX 程序首选数据传输格式

赵金勇

华北计算机系统工程研究所, 北京 (100083)

E-mail: jinyongzhao@126.com

摘 要: JSON 是 JAVASCRIPT 语言的一个子集, 它是一种纯文本的格式, 具有良好的易读性。同时, 它也是轻量级的, 在 JAVASCRIPT 语言中更容易解析。因此, 在开发 AJAX 程序的过程中越来越受到人们的关注。

关键词: JSON, AJAX, XML, JAVASCRIPT

中图分类号: TP311

1. 引 言

近来, Ajax 炙手可热, 它通过避免网页整体刷新, 让 Web 应用程序的响应速度可与桌面应用程序媲美! Ajax 全称是 Asynchronous JavaScript and XML, 它是一种利用 XMLHttpRequest 对象向后台 web 服务器发起异步请求的 web 开发技术, 虽然名称中提到 XML, 但作为前后台的数据交换格式, XML 并不是数据交换的唯一选择。作为 Ajax 程序与后台 web 服务器数据交换的格式, 既可以是 XML、HTML 片段, 也可以是 JSON。如果这些数据的 MIME 类型是“text/plain”, 那么你可以通过分析 XMLHttpRequest 对象的 responseText 属性来存取它; 然而, 如果其 MIME 类型为“text/xml”, 那么 XMLHttpRequest 对象必须采取额外的措施: 它要在返回的文档对象上运行一个 XML 分析器并在内存中构建一棵文档对象模型 (DOM) 树来描述该文档。然后, 你才可以使用 JavaScript 的标准 DOM 方法在树中导航并检索元素、属性及位于该 DOM 树中的其它文本, 此过程十分繁琐!

在我们最近实施的项目中, 前后台数据交换全部采用 JSON 格式, JSON 格式为我们带来了很大的好处, 省去了以往解析操作, 简化了数据访问, 极大的提高了开发的效率!

2. 什么是 JSON

JSON 全称是 JavaScript Object Notation^[1], 它本身就是 JavaScript 语言的一个子集, 是一种轻量级的数据交换格式, 所以它比 XML 更容易解析, 由于 JSON 也是基于文本的, 所以它具有很好的易读性!

JSON 建构于两种结构:

1. “名称/值”对的集合 (A collection of name/value pairs)。不同的语言中, 它被理解为对象 (object), 纪录 (record), 结构 (struct), 字典 (dictionary), 哈希表 (hash table), 有键列表 (keyed list), 或者关联数组 (associative array)。

2. 值的有序列表 (An ordered list of values)。在大部分语言中, 它被理解为数组 (array)。这些都是常见的数据结构。事实上大部分现代计算机语言都以某种形式支持它们。这使得一种数据格式在同样基于这些结构的编程语言之间交换成为可能。

3. JSON 与 XML 比较

首先, 让我们看下面一个关于雇员的描述:

(1) XML 版本:

```
<employees>
```

```
<employee>
```

```
<name>ZhangSan</name>
<sex>Male</sex>
<age>30</age>
<city>BeiJing</city>
<state>China</state>
<department>IT</department>
</employee>
<employee>
  <name>LiSi</name>
  <sex>Female</sex>
  <age>20</age>
  <city>ShangHai</city>
  <state>China</state>
  <department>IT</department>

</employee>
</employees>
```

(2) JSON 版本:

```
{
  "employees" :
  [
    {
      "name" : "ZhangSan",
      "sex" : "Male",
      "age" : "30",
      "city" : "BeiJing",
      "state" : "China",
      "department" : "IT"
    },
    {
      "name" : "LiSi",
      "sex" : "Female",
      "age" : "20",
      "city" : "ShangHai",
      "state" : "China",
      "department" : "IT"
    }
  ]
}
```

正如你所见，JSON 也提供了一种具有嵌套数据元素的结构，就象 XML 一样。与 XML 一样，JSON 也是基于文本的，且它们都使用 Unicode 编码，且其与 XML 一样具有可读性。主观上来看，JSON 更为清晰。JSON 网站提供了对 JSON 语法的严格描述。从总体来看，XML 比较适合于标记文档，而 JSON 却更适于进行数据交换处理。一个 JSON 文档的每一个实例都负责描述一个对象—具体的描述是通过使用嵌套的对象，数组，字符串，数字，布

尔值或 null 值来实现的^[2]。

JSON 的另一个优点是它的非冗长性。在 XML 中，打开和关闭标记是必需的，这样才能满足标记的依从性；而在 JSON 中，所有这些要求只需通过一个简单的括号即可满足。在包含有数以百计字段的数据交换中，传统的 XML 标记将会延长数据交换时间。目前还没有正式的研究表明 JSON 比 XML 有更高的线上传输效率；人们只是通过简单的字节数比较发现，对于等效的 JSON 和 XML 有效负载，前者总是小于后者。至于它们之间的差距有多大，特别是在新的 XML 压缩格式下它们的差距有多大，有待进一步的研究。但有一点可以肯定，

较少的冗余意味着更少的传输和解析！换句话说，也就是 JSON 比 xml 占用更少的带宽、内存和 CPU 资源！

上面的实例在数据比较大时，比如，有 1 万个 employee，那么其中字段 "name","sex","age","city","state","department" 会有大量的冗余，事实上以上实例可以简化为：

```
{
  "employees": {
    "field": ["name","sex","age","city","state","department"],
    "value": [
      ["ZhangSan","Male","30","BeiJing","China","IT"],
      ["LiSi","Female","20","ShangHai","China","IT"]
    ]
  }
}
```

其中，field 字段代表了 value 数组中各个值的位置和意义。很明显，数据的冗余进一步减少，事实上，我们可以只保留 value 数组中的有效数据，这样获得最小的冗余。但是，任何收获都要付出代价，这里的代价就是，可读性和可维护性降低！

4. JSON 应用

4.1 表示数据

简单 JSON，按照最简单的形式，可以用下面这样的 JSON 表示名称/值对：

{ "firstName": "Brett" } 这个示例非常简单，实际上比等效的纯文本名称/值对占用更多的空间：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：

```
{ "firstName": "Brett", "lastName": "McLaughlin", "email": "brett@newInstance.com" }
```

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。

4.2 数据访问

以下叙述中所用的数据来自上面（三 JSON 与 XML 比较）中的实例。

首先，看一下对于 XML 数据访问：

```
var root = responseXML;
var employees = root.getElementsByTagName('employees')[0];
for(var i = 0; i < employees.length; i++) {
  var name = employees[i].getElementsByTagName('name')[0];
  .....
}
```

```
}
```

对于 JSON 数据的访问:

```
var employees = responseText.parseJSON();//把 JSON 字符串解析成对象。  
for(var i = 0; i < employees.length; i++) {  
    var name = employees[i].name;  
    .....  
}
```

很显然, 由于 JSON 是 Javascript 的子集, 所以对 JSON 数据访问非常简单! 而且速度也比 XML 格式快很多!

4.3 对象/字符串相互转换

由于客户端和服务端之间传递的参数是字符串, 因此要在客户端访问 JSON 数据, 必须先把 JSON 字符串转换成对象。往后台服务器端传输时, 必须先把对象转换成字符串。

现在, 存在一些开源免费的 JSON 转换工具, 如<http://www.json.org/json.js> 上的 json.js 就是一个很好的转换工具, 在上面的叙述中, 我们已经应用到。它提供了两个方法 `parseJSON()` 和 `toJSONString()` 来做 JSON 对象和字符串之间的转换, 我们在项目中一直应用此解析工具!

4.4 对象传输

以上都是 JSON 的一些基础用法, 当然还有一些较为复杂的应用, 限于篇幅, 我们只作简单介绍, 希望能够抛砖引玉!

首先, 我们先介绍一种 JavaScript 的写法:

```
var obj = {  
    name : "zhangsan"  
    sex  : "male",  
    age  : 20,  
    sayHello : function() {  
        alert("Hello,everyone");  
    }  
}
```

这是 JavaScript 的一种写法, 相信大家已经看出, 这也符合 JSON 格式。既然可以这样, 我们就可以把这个对象 `obj` 转换成 JSON 字符串, 然后传递的后台服务器, 后台 web 服务器可对它进行一定的处理, 再返回或者存入数据库, 以后可以直接传递到页面用 `eval()` 函数执行! 这是一种十分有用的方式。如: 客户端有 JavaScript 函数 `fun(o)`, 后台 web 服务器可以通过计算返回一个形如: `[fun(1), fun(2), fun(3)]` 这样的字符串, 在页面用 `var arg = eval("[fun(1), fun(2), fun(3)]")`, 就可以得到一个数组!

5. 客户端和服务端对 JSON 支持

到目前为止, 我们一直集中于讨论如何把 JSON 应用于客户端浏览器上的基于 AJAX 技术的 Web 应用程序。当然, Web 服务器端必须存在一定的技术支持才能实现首先生成 JSON, 然后由客户端使用 JSON。客户端对 JSON 的支持已经有很多开源的工具, 如我们上面提到的 json.js

5.1 在服务器上解释 JSON

一旦您编写完客户端 JavaScript 代码、允许用户与 Web 表单和 Web 页的交互、收集发送给服务器端程序以做处理所需的信息，此时，服务器就成为了应用程序（如果调用了异步使用的服务器端程序，则可能是我们认为的所谓的“Ajax 程序”）中的主角。在此时，您在客户端所做的选择（比如使用 JavaScript 对象，然后将其转换成 JSON 字符串）必须要与服务器端的选择相匹配，比如使用哪个 API 解码 JSON 数据。

5.2 处理 JSON 的两步骤

不管在服务器端使用何种语言，在服务器端处理 JSON 基本上就需要两个步骤。

<1> 针对编写服务器端程序所用的语言，找到相应的 JSON 解析器/工具箱/帮助器 API。

<2> 使用 JSON 解析器/工具箱/帮助器 API 取得来自客户机的请求数据并将数据转变成脚本能理解的东西

5.3 寻找 JSON 解析器

寻找 JSON 解析器或工具箱最好的资源是 JSON 站点。在这里，除了可以了解此格式本身的方方面面之外，还可以通过各种链接找到 JSON 的各种工具和解析器，从 ASP 到 Erlang，到 Pike，再到 Ruby，应有尽有。您只需针对自己编写脚本所用的语言下载相应的工具箱即可。为了让服务器端脚本和程序能够使用此工具箱，可以根据情况对其进行选择、扩展或安装（如果在服务器端使用的是 C#、PHP 或 Lisp，则可变性更大）。

例如，如果使用的是 PHP，可以简单将其升级至 PHP 5.2 并用它完成操作；在 PHP 这个最新版本默认包含了 JSON 扩展。实际上，那也是在使用 PHP 时处理 JSON 的最好方法。如果使用的是 Java servlet，json.org 上的 org.json 包显然就是个不错的选择。在这种情况下，可以从 JSON Web 站点下载 json.zip 并将其中包含的源文件添加到项目构建目录。编译完这些文件后，一切就就绪了。对于所支持的其他语言，同样可以使用相同的步骤；使用何种语言取决于您对该语言的精通程度，最好使用您所熟悉的语言。

6. 结论

JSON 的基本思想是，把自己实现为 JavaScript 的一个子集，从而使其成为一种轻量级的和高度灵活的处理 AJAX 数据交换的方式。与传统的 XML 处理技术相比，这个工具的分析速度更快且更为易用。从该工具的性能和使用现状来看，它有可能会成为下一代 Web 2.0 开发中 XML 数据操作的主要替代者。任何开发者，无论是开发标准桌面应用程序还是开发 Web 应用程序，只要使用 XML 数据处理，都会欣赏 JSON 的简易特征。JSON 的最大价值在于可以将 JavaScript 真的作为 JavaScript 而非数据格式语言进行处理。

参考文献

- [1] Nicholas C. Zakas . 《JavaScript 高级程序设计》，人民邮电出版社，2006.9
- [2] Alessandro Lacava. 《Speed Up Your AJAX-based Apps with JSON》，2006.5
- [3] Kris Hadlock. 《Advanced AJAX With JSON》2006.10

JSON——the First Option for the AJAX Program

Zhao Jinyong

National Computer System Engineering Research Institute of China, Beijing (100083)

Abstract

JSON is a subset of JAVASCRIPT, because it is text-based it is easy for humans to read and write. It is also a lightweight data-interchange format that's easy to parse programmatically. So more and more people begin to pay attention to it in the development of AJAX Program.

Keywords: JSON, AJAX, XML, JAVASCRIPT