

XML 与 JSON 在 WEB 中对数据封装解析的对比

Comparison of Data Encapsulation and Analysis of XML and JSON in WEB

雷映喜 LEI Ying-xi; 习淑婷 XI Shu-ting; 彭俊峰 PENG Jun-feng; 周应光 ZHOU Ying-guang

(玉溪师范学院, 玉溪 653100)

(Yuxi Normal University, Yuxi 653100, China)

摘要: 通过介绍 XML 与 JSON 两种数据交换的封装和浏览器如何解析, 网络上传输的性能的比较, 在 WEB 中使用 JSON 作为数据传输格式。

Abstract: The paper introduces the data encapsulation of both XML and JSON and how the browser analyzes data, compares the transmission on the network, use JSON as data transformat in WEB.

关键词: XML; JSON; 封装解析

Key words: XML; JSON; encapsulation and analysis

中图分类号: TP39

文献标识码: A

文章编号: 1006-4311(2013)09-0210-02

0 引言

信息技术的发展, AJAX 技术成熟运用越来越受到广大程序员的喜爱, 作为封装数据的 XML 和 JSON 成为了主要的交换格式, 在一个 WEB 应用中, 如何选择一个合适的交换格式, 直接影响到整个应用的性能。

1 概述

1.1 XML 和 JSON 简介 XML 是 Extensible Markup Language (可扩展标记语言, 有时候也写作 eXtensible Markup Language) 的简写形式, 它使信息以有意义的结构进行编码, 计算机和人在一定程度上都可以理解这种编码^[1]。可以用来标记数据和定义数据类型, 是一种允许用户对自己的标记语言进行定义的源语言。具有良好的可扩展性、开放性、跨平台性和互操作性。

XML 最初设计的目的是弥补 HTML 的不足, 以强大的扩展性满足网络信息发布的需要, 后来逐渐用于网络数据的转换和描述。

JSON (JavaScript Object Notation) 是一种数据交换格式, 是以 JavaScript 为基础的数据表示语言, 是在以下两种数据结构的基础上定义基本的数据描述格式的:

1) 含有名称/值对的集合;

2) 一个有序的列表^[2]。JSON 技术借鉴了 javascript 规范当中对象创建的一种语法结构, 相对于 XML 是一种轻量级的数据交换标准。

1.2 XML 和 JSON 语法对比

1.2.1 XML 语法 XML 语法相对简单, 所有的 XML 元素必须合理包含, 且所有的 XML 文档必须有一个根元素。

如同 HTML 一样, XML 元素同样也可以拥有属性。XML 元素的属性以名字/值成对的出现。

XML 语法规规范要求 XML 元素属性值必须用引号引着。用户可以自定义标签。

如:

```
<person> <user><name>张三</name></user></per-
```

son>

注意: 以 <person> 开头的必须要以 </person> 结尾, 并且区分大小写。

1.2.2 JSON 语法

1) 表示一个对象

{属性名 1: 属性值 1, 属性名 2: 属性值 2...}

注意: 属性名必须有引号 (单引号, 或者双引号) 引起来。属性值如果是字符串, 也必须使用引号引起来。

属性值的类型可以是 string, number, Boolean, object, null。

如:

```
{'name': 'zs', 'age': 22};
```

```
{'name': 'ww', 'address': {'room': 2012}}
```

2) 表示一个数组

```
var arr=[{属性名 1: 属性值 1, 属性名 2: 属性值 2}, {属性名 1: 属性值 1, 属性名 2: 属性值 2}];
```

2 封装对比

现在我们做一个实验, 环境如下:

硬件: 在处理器为 Intel Core i5; 内存为 4G; 硬盘: 500G 的笔记本电脑上。

软件: 浏览器: IE9; 实验服务器为: Tomcat 6.0; 实验开发工具: MyEclipse 8.5。

在数据库中字段: name, sex, phone, address; 服务器端取出数据用 JSON 和 XML 对数据封装传输部分代码:

```
Xml: <users><user><name>user1</name><sex>男</sex> <phone> 123345677 </phone> <address>北京</address></user>
```

```
</users>
```

```
JSON: [{"address": "北京", "name": "user1", "phone": "123345677", "sex": "男 1"}, {"address": "云南 0", "name": "user2", "phone": "123345677", "sex": "男 1"}, ...]
```

JSON 在服务器的封装:

```
list=findDao.getListData(); //从数据库中取出数据, 返回一个 List 集合
```

```
JSONArray obj = JSONArray.fromObject(list);
```

作者简介: 雷映喜 (1989-), 男, 云南文山山人, 玉溪师范学院学生, 计算机科学与技术专业。

```
String str = obj.toString();
```

XML 在服务器端从数据库中取出数据的封装, 这里采用 dom4j 封装:

```
list = findDao.getListData (); //从数据库中取出数据, 返回一个 List 集合
Document document = DocumentHelper.createDocument();
Element root = document.addElement("users");
for (Info info : list) {
    Element employee = root.addElement("user");
    Element name = employee.addElement("name");
    name.setText(info.getName());
    //同理可封装 sex, phone, address 节点
}
String str = document.asXML();
```

3 浏览器解析对比

3.1 在浏览器中使用 JAVASCRIPT 对 JSON 和 XML 进行解析:

3.1.1 JAVASCRIPT 解析 JSON 的部分代码:

```
var txt = xhr.responseText;
var ojson = txt.evalJSON();
for (int i = 0; i < ojson.length; i++) {
    var showDate = ojson[i].name + ' 性别: ' + ojson[i].sex + ' 电话: ' +
    ojson[i].phone + ' 地址: ' + ojson[i].address;
}
```

3.1.2 JAVASCRIPT 解析 XML 的部分代码:

```
var doc = xhr.responseXML;
var xmlDoc = doc.documentElement;
var root = doc.getElementsByTagName("person");
var msg = xmlDoc.getElementsByTagName("user");
for (int i = 0; i < msg.length; i++) {
    var elem = node[0];
    var name = elem.getElementsByTagName("name")[i].textContent != undefined ?
    elem.getElementsByTagName("name")[0].textContent :
    elem.getElementsByTagName("name")[i].text;
}
```

3.2 对浏览器上解析 JSON 和 XML 的数据做一个对比:

根据数据的增多, 浏览器解析数据所用的时间, 如下表:

表 1

记录数/条	所用时间 (ms)	
	JSON	XML
500	14	2166
1000	21	8943
1500	30	22796

数据传输时文件大小对比:

表 2

记录的条数	所占用的大小 (kB)	
	JSON	XML
500	37.81	51.53
1000	75.89	103.28
1500	115.93	157.00

(注: 每次试验都清空浏览器缓存, 由于解析速度过慢, 以上是测试 5 次的平均值。CPU 使用率为 3%, 内存占 1.49G。)

从上面可以看出, 在浏览器上解析 JSON 明显要比 XML 的解析要简单的多, 而且 XML 解析代码量大, 比较复杂。从 (表 1) 中可以看出, 当解析相同记录的数据时, 其效率 JSON 是 XML 的 150 倍左右, 并且随着记录数不断的增多, 其倍数明显的在翻倍增长。

从 (表 2) 可以看出, 在传输相同的数据, 使用 JSON 要比使用 XML 封装格式传输文件要小。使用 XML 传输格式, 引入了一些不必要的字符, 这样也需要花费一定时间, 和占用一定带宽, 不利于网络传输。虽然, 在 XML 数据访问与数据传输是可以优化的^[2], 这个已经有人写过。这需要一些编程的老手进行数据的分析才能得出优化方式。这不合适初学者和一些开发经验较少的程序员。

4 结论

不管是选择哪种技术来对数据进行封装传输, 都要考虑系统的开销。考虑系统传输数据的复杂程度, 和数据量, 才能减少开发成本, 提高用户的体验度。从上面的解析和封装的比较得出, 在一般的 WEB 应用中传输数据用 JSON 作为数据封装格式, 解析速度快, 代码量少, 操作简单。在性能上要明显优于 XML。可以降低 XML 的数据冗余。而在一些需要特殊功能的 WEB 应用中, 传输的数据复杂, 这样可以采用 XML。XML 更倾向于企业级的应用。严格的规范与众多的周边技术, 使得 XML 应用范围易于扩展。而大量的商业化的应用程序支持 XML, 容易的编写与阅读 XML。

参考文献:

- [1] Mitch Amiano Conrad D'Cruz 等著, 焦瑜净, 闫雷鸣译. XML 案例解析教程[M]. 北京: 华大学出版社, 2007.
- [2] 李传良. XML 数据访问与数据传输优化[J]. 科学之友科学之友, 下旬. 2011, (08).
- [3] Daniel Rubio. An introduction to JSON [OL]. [2007/02/28].
- [4] 高静, 段会川. JSON 数据传输效率研究[J]. 计算机工程与设计, 2011, (07).
- [5] 李浩, 王恂. 用 JSON 改进 AJAX 数据传输[J]. 中国高新技术企业, 2008, (16).
- [6] 黄芳. SQL 数据支持 XML 技术[J]. 山东省农业管理干部学院学报, 2006, (05).
- [7] 高静. Xml 在数据挖掘中的应用[J]. 科技创新导报, 2010, (24).
- [8] 彭亚依. XML 在电子商务中的应用[J]. 软件导刊, 2009, (02).
- [9] 罗天抒. 基于 Flex 的无纸化考试系统的设计与实现[J]. 企业家天地半月刊(理论版), 2009, (07).
- [10] 谷东峰. 基于 XML 电子商务的研究[J]. 现代情报, 2009(02).

论文降重、修改、代写请扫码



免费论文查重，传递门 >> <http://free.paperyy.com>

阅读此文的还阅读了：

- [1. JSON格式数据在Web开发中的应用](#)
- [2. JSON与XML的比较研究](#)
- [3. 基于Web的在线复杂公式编辑器的实现](#)
- [4. 无线控制器上对WAPI数据处理与实现](#)
- [5. Infineon推出高速蓝牙芯片](#)
- [6. 比较XML与JSON在Web中的应用](#)
- [7. XML与JSON在WEB中对数据封装解析的对比](#)
- [8. XML在移动电子商务中的应用](#)
- [9. JSON与XML在网络数据传输中的应用分析](#)
- [10. 宇瞻 DDR II 533](#)