Ajax技术的数据响应优化[1]

谭力，杨宗源，谢瑾奎
华东师范大学计算机科学技术系，上海(200241)
E-mail：darkwhite29@gmail.com

摘　要：Ajax 技术作为一种新兴的 Web 应用开发技术，相较于传统的 Web 应用开发模式，能提供更流畅而优质的用户体验，为创建类桌面应用程序提供了可能性。本文针对 Ajax 工作原理和数据传输性能进行了分析，随后从响应数据的返回格式角度入手，进一步通过实验对比了两种数据格式 XML 和 JSON 的差异和优劣，并基于实际情况给出了权衡取舍的一些建议，总结出了较为优化的 Ajax 模型，使目前 Ajax 带来的数据响应冗余的缺陷得到了改善。
关键词： Ajax；数据响应；XML；JSON
中图分类号： TP311.52

1 引 言

Ajax 的全称是 Asynchronous JavaScript and XML，即异步 JavaScript 与 XML， 是几项按一定方式组合在一起，协同发挥各自作用的技术组合[8]。Ajax 这个概念在 2005 年 2 月被 Jesse James Garrett 首先提出，并且得到了世界范围内的广泛应用。据非正式不完全统计，目前全世界范围内，使用 Ajax 技术的网站已达数百万个。不刷新网页就能给用户提供动态内容已经成为 Ajax 赢得美誉的决定性因素。Ajax 的出现使得互联网传统的“请求-等待-响应”模式有了重大转变，这正是 Ajax 的核心理念所在。

作为一项越来越流行的“老”技术，Ajax 正以它的各个技术分支所体现出的综合优势提供着更好的用户体验，获得了更广阔的 Web 应用市场。目前，Ajax 应用最普遍的领域是 GIS-Map，GIS 应用交互非常频繁，它强调快速响应，Ajax 的特点正好可以满足这类需求。但享受 Ajax 强大功能的同时，随之而来的问题也不容忽视，由于综合使用了 JavaScript、XHTML 和 CSS、DOM、XML 和 XSTL 以及 XMLHttpRequest 等技术，使得 Ajax 集上述各技术的优缺点于一身。当运用 Ajax 技术来开发一个健壮的软件系统时，需要考虑 Ajax 的可移植性、效用性和可用性等非功能性因素，尤其是效用性。一个 Ajax 应用程序执行效率的高低将在很大程度上影响系统的性能，而在与服务器异步交互的过程中如何高效地传递响应数据则是很重要的一个环节。本文就从 Ajax 的数据响应角度进行分析，探讨如何趋利避害，构建一个高效的 Ajax 应用程序，最后提出了一个整体优化方案。

2 目前 Ajax 性能优化的研究进展

应用程序的性能取决于两个因素：算法复杂度和对系统资源的消耗（最重要的是内存和 CPU）。人们对程序的逻辑性的重视往往高于其性能，但实际上一个程序的性能非常重要，它是用户选择使用某个软件产品的首要考虑因素。一个性能极差的软件，用户会马上放弃它。

我们往往重点关注于如何用 Ajax 实现强大的功能，却淡化了对 Ajax 性能的进一步要求，在 Ajax 广泛应用的今天，如何提高 Ajax 应用程序的效用性已成为当务之急。目前，国内关于 Ajax 性能优化已有的研究主要集中在：对基于 Ajax 的 MVC 模式进行改造[3]，提出一种基于 JSON 的对象序列化算法来解决解析 XML 所造成的缺陷[4]，对 Ajax 的首页加载模式进行改进[5]等。

[1]本课题得到高等学校博士点基金：20060269002 资助。

要对 Ajax 的性能实施优化，首先要考虑的是 Ajax 的基本原理，因为这从根本上决定了其性能。传统的 Web 应用采用同步交互过程。这是一种不连贯的用户体验，原因在于服务器在处理请求的过程中，用户是处于等待的状态，在结果页面没有返回之前，浏览器窗口内是空白的。即使用户只需要看到页面一小部分的数据更新，还是需要将整个页面重新加载一次。而 Ajax 正是从人性化的角度出发，针对用户提交请求后必须等待服务器的响应这一点，对 Web 请求应答机制作出了相应的改进，从而减少了用户的等待时间，提升了用户体验的满意度。

Ajax 采用异步交互方式，在用户与服务器之间引入了一个用 JavaScript 编写的 Ajax 引擎，来代替用户与服务器进行交互。这样用户可以无需等待响应，继续其他的 Web 交互。这是一种连续的用户体验，使用 Ajax，程序员可以创建响应速度接近于本地桌面应用程序的 Web 应用程序。

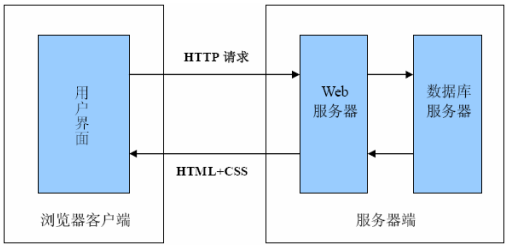

图 1　传统的 Web 应用模式——同步交互方式

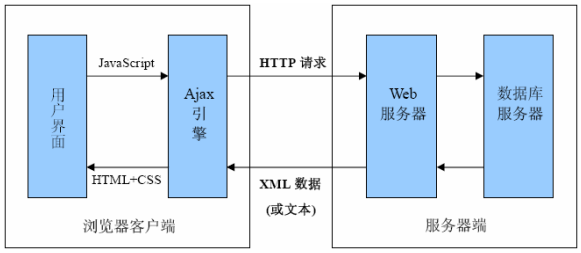

图 2　基于 Ajax 的 Web 应用模式——异步交互方式

从图 1 和图 2[1]的对比中可以看到，在基于 Ajax 的 Web 应用模式中，用户与服务器之间多了一个 Ajax 引擎，正是它使得客户端与服务器之间的交互实现了异步化，因为它可以代替用户产生一个 HTTP 请求，而同时用户的页面操作不被打断。但正是因为 Ajax 引擎的存在，带来了传输响应数据的问题。在传统 Web 应用中，Web 服务器将响应数据封装在 HTML 页面中，并返回给客户端显示；在基于 Ajax 的 Web 应用中，Web 服务器把响应数据封装在 XML 文档或纯文本中，并返回给 Ajax 引擎，后者从中解析出响应数据，作为局部更新的内容嵌入原 HTML 页面中并在本地显示。

3 响应数据传输格式分析

正如第 1 节中讨论的，在服务器返回的响应数据传输过程中，Ajax 引擎可以用以下两种数据响应格式来获取从服务器返回的信息：一是纯文本，用 XMLHttpRequest 对象的 responseText 属性来获取；二是 XML 文档，用 XMLHttpRequest 对象的 responseXML 属性来获取。若使用纯文本格式，当前最常用的格式是 JSON。用 JSON 和 XML 来返回响应数据各有利弊，以下将从几个方面作进一步分析，通过对比两者的优劣，就可根据实际情况作出最优的选择。

3.1 XML 和 JSON 简介

XML (eXtensible Markup Language)是一种类似于 HTML 的，用户自定义标签的标记语言。XML 是 Ajax 技术的一个重要组成部分。在 Ajax 应用程序中，通常由 XML 作为数据传输的媒介，来承载服务器处理并返回的数据。随后 JavaScript 操作 DOM 元素对返回的 XML 文档进行解析，提取响应结果，从而更新页面内容。而服务器返回的数据不光可以是 XML 格式，也可以是 JSON 格式的。使用 XML 确实有其方便之处，因为 XML 是目前 Web 数据表示较为通用的标准，兼容性较好。但当 XML 数据量大，结构复杂时，会在一定程度上影

响服务器的响应速度和处理效率。因为若使用 XML，我们不得不加上一些并不需要的冗余数据如自定义的开始标签和结束标签。

而 JSON 正是作为一种 XML 的替代品提出的。JSON (JavaScript Object Notation)是一种基于 JavaScript 的，比 XML 更简洁的轻量级数据交换格式，它采用“名/值”序偶和值的有序列表的形式，本质上是一个关联数组（associative array）或者一个哈希表（hash table）[6]。JSON 作为一种轻量级的数据交换格式，具有简洁、数据量小以及便于解析等优点，受到了开发人员的欢迎。虽 JSON 仍处于起步阶段，但其应用前景是非常广阔的。当 JSON 被浏览器和服务器端编程语言完全支持的时候，相信一个网络时代的新阶段即将来临。

如何在 XML 和 JSON 之间权衡，选用一个当前场景下最合适的响应数据载体，是优化一个 Ajax 应用程序需要考虑的问题。

3.2 用户可读性

XML 采用用户自定义标签来包含数据，牺牲了存储空间，可文档显得整洁容易理解，换取了用户可读性；相反 JSON 虽具有简洁的语法，但当用户面对一系列的冒号、逗号和大括号时，却显得有些茫然。对于习惯了 HTML 和 XML 中一脉相承的开始结束标签的用户来讲，JSON 的代码不是很容易阅读，像 XML 这种基于标签的语法，要更整洁一些[7]。明显地，具有明确语义的开始和结束标签比起符号来更能展现文档的结构和含义。

但是在 Ajax 这个场景中，用 XML 和 JSON 表示的只是服务器返回的中间数据，对于用户是不可见的。当数据真正呈现在页面上时，使用 XML 和 JSON 最后的页面效果是一样的。因此，从优化 Ajax 的数据响应过程的角度来说，两种数据响应格式对于用户的可读性似乎是不需要考虑的，更应该考虑的是它们针对系统性能的可读性。下面的讨论将会给出答案。

3.3 数据量

下面通过两个示例：在 Google Suggest 中输入查询字符串”compiler”后返回搜索建议（示例 1）和电子邮件（示例 2），来对两种数据响应格式作出对比。示例 1 的 XML 格式代码如下：

```
<?xml version=”1.0” encoding=”utf-8”?>
<suggestions terms=”compiler”>
    <suggestion term=”compiler compliance level” results=”451, 000 results” />
    <suggestion term=”compiler design” results=”952, 000 results” />
    <suggestion term=”compiler options” results=”253, 000 results” />
    <suggestion term=”compiler optimization” results=”417, 000 results” />
    <suggestion term=”compiler services” results=”51, 400 results” />
    <suggestion term=”compiler course” results=”5, 400, 000 results” />
    <suggestion term=”compiler conference” results=”5, 080, 000 results” />
    <suggestion term=”compiler error” results=”1, 700, 000 results” />
    <suggestion term=”compilers and compiler generators” results=”314, 000 results” />
</suggestions>
```

示例 1 的 JSON 格式代码如下：

```
{"suggestions":{
```

```
"terms":"compiler",
"suggestion":[{
    "term":"compiler compliance level","results":"451, 000 results"},{
    "term":"compiler design","results":"952, 000 results"},{
    "term":"compiler options","results":"253, 000 results"},{
    "term":"compiler optimization","results":"417, 000 results"},{
    "term":"compiler services","results":"51, 400 results"},{
    "term":"compiler course","results":"5, 400, 000 results"},{
    "term":"compiler conference","results":"5, 080, 000 results"},{
    "term":"compiler error","results":"1, 700, 000 results"},{
    "term":"compilers and compiler generators","results":"314, 000 results"}
]
}
}
```


由上可得出下图（由于篇幅关系，示例 2 不给出具体表示，表示格式类似）：

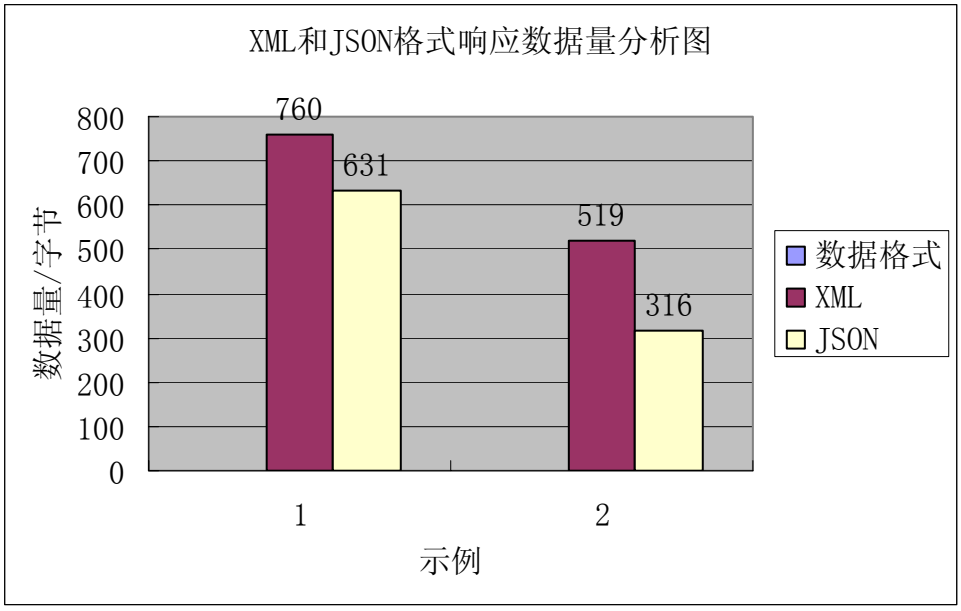

图 3　XML 和 JSON 格式响应数据量分析图

可以看到若服务器端返回的是 XML 格式的数据，除了冗余的开始结束标签之外，还必须确保该 XML 文档含有特定内容的首部信息，如 XML 版本号和编码格式；而若采用 JSON 返回响应数据，使用的结构化符号将短小精悍许多，如逗号、冒号等只占一个字符的简单符号，这将在很大程度上减少浏览器和服务器之间传输的数据量。随着应用程序中的数据交换量的增长，数据结构的复杂化，JSON 数据量小的优势将更为明显（如示例 2）。这样客户端与服务器端进行数据交换时，占用的带宽就会尽可能小，应用程序就运行得更快、更有效率。虽然表面上在一次请求中数据缩减量不大，但如果这样的并发请求数量众多，节省的数据量就有很大差别了。设想这样一个一天处理上万条并发请求的服务器，一个月所节省的系统资源开销是相当可观的。

3.4 客户端解析效率

JSON 跟 XML 都是结构化的数据交换格式，两者的不同在于 XML 本身是 DOM 树结构

的，需要 JavaScript 操作 DOM 元素来进行解析才能获取其中的数据，而且 DOM 在各个浏览器中的实现也不尽相同，所以针对 XML DOM 的编程会变的更为复杂。而 JSON 本身就是 JavaScript，因此不需要 Ajax 引擎来解析，一旦调用 JavaScript 的 eval()方法将 JSON 字符串序列化成为 JavaScript 对象之后，就可以直接读取其属性来获取数据。相比之下，访问和处理数据时 JSON 比 XML 要快捷许多。

下面通过一个简单的测试程序来计算代码的执行时间，从而比较两种格式读取数据时的解析效率。其中，解析 XML 中响应数据的代码如下：

```
var t1 = new Date().getTime();
for(var i=0; i<10000; i++){
   var data = request.responseXML;
   var name = data.getElementByTagName
            (“name”)[0].firstChild.nodeValue;
   var website = data.getElementsByTagName
            (“website”)[0].firstChild.nodeValue;
   var email = data.getElementsByTagName
            (“email”)[0].firstChild.nodeValue;
}
var t2 = new Date().getTime();
alert(t2-t1);
```

解析 JSON 中响应数据的代码如下：

```
var t1 = new Date().getTime();
for(var i=0; i<10000; i++){
   var data = eval(‘(’+request.responseText+‘)’);
   var name = data.person.name;
   var email = data.person.email;
   var website = data.person.website;
}
var t2 = new Date().getTime();
alert(t2-t1);
```

在上面两段分别解析 XML 和 JSON 中响应数据的代码中，for 循环中的代码是两者各自的解析操作，在解析操作代码段的开始和结束分别使用了 JavaScript 中的 getTime()函数来记录时间，最后计算差值，即代码的运算时间。为消除偶然误差，重复试验（10 次）可统计出下图：

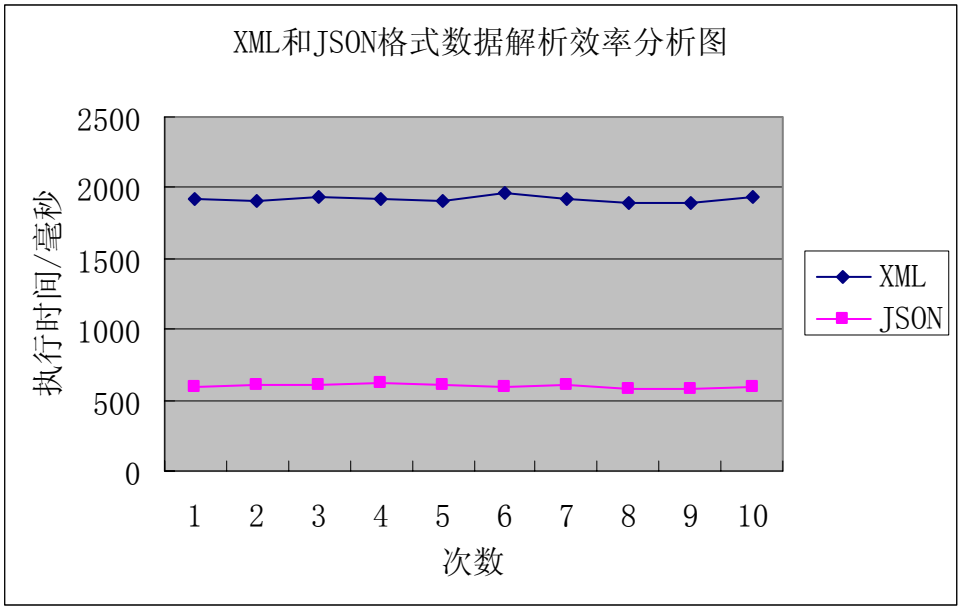

图 4　XML 和 JSON 格式数据解析效率分析图

从上图可以看出，JSON 的数据解析效率要远比 XML 高，也就是说，若使用 JSON 作为响应数据载体，将会加快页面响应速度，使得运用 Ajax 而获得的即时页面更新效果更加显著，用户体验更加流畅。因此，从系统性能而非用户的角度来讲，JSON 比 XML 的可读性更好，这也回答了 2.1 小节中的问题。

3.5 服务器端开发效率

对于如何将一段数据序列化为一个 XML 文档，各种服务器端编程语言都有提供多种方式来实现，如.NET 框架下，C#中的 XmlSerializer 类，通过结合 TextWriter 类，它可以序列化一个对象成为 XML 格式。在 Java 中则提供了 XmlStreamWriter 和 XMLEncoder 类以及 API 函数 SAX 来处理普通对象到 XML 的转换。另外，在 PHP 以及.NET 等多种开发环境下，开发人员都可以使用 XmlWriter 类等工具来直接生成 XML 文档。

而对于 JSON 的自动生成支持工具，目前来讲还比较少，开发人员可能需要自己手动编写一段代码或者使用一些零散的开源类库来完成这项工作。而且对于较为复杂的数据，作为一种新兴的数据交换格式，JSON 的生成效率和稳定性会比 XML 差一些。

因此，从服务器端开发效率上来讲，较早出现并且更加标准化、规范化的 XML 目前会比 JSON 有更好的表现。

3.6 安全分析及优化

JSON 是一种用于在两台机器之间的传输数据的数据交换格式。由于它承载的只是数据，不会含有赋值和调用，所以它是安全中立的。使用 JSON 的系统安全与否，取决于该系统自身设计的好坏。JSON 本身并不会引入安全隐患[2]。而当开发人员用 eval()函数把 JSON 数据作为 JavaScript 代码执行，从而转化为 JavaScript 对象时，可能会带来意想不到的安全性问题。攻击者可以在 JSON 数据中携带恶意的 JavaScript 代码发送给客户端，这样 eval()函数就会执行这些恶意代码，这就存在相当大的风险，系统可能会因此而崩溃。虽然，编程过程中使用 JSON 时只会提取其属性值，而忽略其所有方法，但并不能就此保证系统安全。

另外，JSON 本质上就是 JavaScript，由于 Web 应用程序的所有访问者都可以阅读到程序中的 JavaScript 源代码，那么每一个用户都可以从服务器上获取 JSON 数据。因此，JSON 只能用于可公开的数据，其他敏感数据则不能使用 JSON 来做传输载体。除非使用了代码混淆器之类的工具来使得 JSON 文档的 URL 无法预测，使用 JSON 才是安全的。

因此，如果使用 JSON 作为数据载体，必须确保 JSON 是安全的。一方面，可以通过口令验证来保障 JSON 数据来源的可信度，即读取的响应数据不是来自于某个不可靠的第三方；另一方面，可以在客户端使用正则表达式来检查 JSON 数据是否包含有恶意代码关键字。从而降低系统遭受攻击的可能性。

从安全性角度来讲，XML 由于解析时不含有任何本地执行过程，因此相对 JSON 来讲更安全一些。

3.7 权衡与取舍

在以上的讨论中，本文从各个角度分析了 XML 和 JSON 各自的优劣。可以看到在用户可读性、服务器端处理效率和安全性上，XML 优于 JSON；而在数据量和客户端解析效率这两个核心指标中 JSON 都胜过 XML 一筹。在和 JSON 的竞争中，XML 已经略显颓势。现实情况也是如此，开发人员越来越多地用 JSON 当作 XML 的一个轻量级替代品来承载响应数据。

根据上面的分析，我们可以列出下表作一个总结：

表 1　XML 和 JSON 的对比表

数据格式	用户可读性	数据量	客户端解析效率	服务器端开发效率	安全性
XML	好	大	低	高	较好
JSON	一般	小	高	低	差

根据上表中的对比可以看出，XML 表现较好的方面是用户可读性，服务器端处理效率和安全性，而 JSON 优于 XML 的方面是数据量和客户端解析效率。于是，针对具体的开发场景，可以给出如下选取方案：

在开发一个基于 Ajax 技术的 Web 应用程序时，在安全性要求不高及服务器处理能力较强的场景下，选用 JSON 更好；在用户体验要求不高的安全敏感场景下，选用 XML 较为合适；在用户体验和安全性要求都较高的场景下，应从大局着眼，选用较为安全的 XML 而牺牲一部分系统性能。

总之，开发人员需要认真评估在不同场景下两种响应数据表示方式的成本和效率，了解两者的差异后，再来根据实际需要进行合理选择，或者直接采用页面重载刷新的方式而不是 Ajax。

结 论

Ajax 是一种很强大的创建类桌面应用程序的 Web 应用开发工具，可以在不需要刷新页面的情况下进行异步的后台数据库交互，从而在不会打断用户在应用程序中其他操作的同时更新页面内容，大大提高了 Web 应用程序的响应速度。

但作为开发一个成熟软件的角度来讲，Ajax 的性能优化是亟待解决的问题。如何高效地运行 Ajax 程序越来越被重视。首先面临的是响应数据传输的问题，我们需要根据实际情况选择数据响应的返回格式，从而使服务器和客户端负载都尽量达到最小化。能否高效地利用系统资源，这是在富互联网应用（Rich Internet Application, RIA）日渐发达的今天，开发

人员面临的一大考验。我们希望看到的是，在不改变浏览器机制和用户习惯的前提下，Web 访问者更好地体验 Ajax 带来的丰富动态功能，就像使用桌面应用程序那样。

本文从数据响应的角度出发，考虑如何优化 Ajax 的性能，就开发人员容易忽视的一些重要问题，作了具体分析并提出了相应的参考建议。尚存的不足之处在于提出的部分观点尚无法给出量化的模型来衡量，有的只能通过主观经验来判断，如怎样根据安全隐患的严重程度来决定选用 XML 还是 JSON。因此，从理论化的角度来讲，还需要进一步的研究。

参考文献

[1] GARRETT J J. Ajax: A New Approach to Web Applications[EB/OL], http://www.adaptivepath.com/ideas/essays/archives/000385.php, 2005-02-18.
[2] CROCKFORD D. JSON and Browser Security[EB/OL], http://yuiblog.com/blog/2007/04/10/json-and-browser-security, 2007-4-10.
[3] 王东，孙彬. 基于 Ajax 的 MVC 框架的改造分析[J]. 计算机应用，2007，27(S1)：293-295.
[4] 张涛，黄强，等. 一个基于 JSON 的对象序列化算法[J]. 计算机工程与应用，2007，43(15)：98-100.
[5] 阳锋，徐建波. AJAX 技术的性能改进研究[J]. 计算机工程与科学，2008，30(6)：146-148.
[6] HADLOCK K. Ajax——Web 开发、可重用组件及模式[M]. 叶俊 译. 北京：清华大学出版社，2007.
[7] KEITH J. Bulletproof Ajax 中文版[M]. 刘申，宋薇 译. 北京：人民邮电出版社，2007.
[8] 汤代禄. 互联网的变革--Web 2.0 理念与设计[M]. 北京：电子工业出版社，2007.

Data response optimization of the Ajax technology

TAN Li, YANG Zongyuan, XIE Jinkui

Department of Computer Science and Technology, East China Normal University, Shanghai (200241)

Abstract

As an emerging developing technology of web applications, in contrast to traditional ways, Ajax is capable of providing smoother user experience and possibility to create desktop-like applications. To summarize an optimized model of Ajax, this paper analyzes the work principle and data transmission performance of Ajax, and then discusses the difference between XML and JSON by experiments with a focus on the format of response data returned. Furthermore, some suggestions to choose either based on the current conditions are given, which improves the shortcomings such as data response redundancy caused by Ajax nowadays.

Keywords: Ajax; data response; XML; JSON

作者简介：

谭力（1986－），男，四川绵阳人，硕士研究生，主要研究方向：Web 开发技术、形式化方法；

杨宗源（1953－），男，上海人，教授，博士生导师，主要研究方向：软件工程、工具及环境、形式化方法；

谢瑾奎（1975－），男，广西桂林人，讲师，主要研究方向：程序语义、软件自动化及智能信息处理、抽象状态机、交互计算及复杂性。