

# Web 异步通信研究与应用

金日男

(北京邮电大学信息与通信工程学院, 北京 100876)

**摘要:** 传统的 Web 数据提交是以整个表单为单位的也就是与服务器的同步通信, 这种通信方式存在一个明显的缺陷就是当用户提交 Web 表单时往往出现较长时间的等待与页面空白, 这极大地降低了用户获取信息的速度与降低了用户体验。Ajax 的出现打破了这种传统思路, 使我们可以将表单的一部分数据提交到服务器对用户只关心的那部分数据进行数据交互和处理从而减小了传输的流量并且也提升了用户的体验。与这个词相关的“最新”术语就是 XMLHttpRequest 对象 (XHR), 它早在 IE 5 (于 1999 年春天发布) 中就已经出现了。现在, 由于这种传输方式在性能上与用户体验上的提升, 异步传输被越来越多的应用在商业用途, 随着 Google Maps、Google Suggest、Gmail、Flickr 等应用越来越炙手可热, XHR 也成为事实上的标准。这篇文章将探究 Web 异步通信的技术实现并探讨该技术在实际当中的应用。

**关键词:** 异步通信; Ajax; JavaScript;

**中图分类号:** TP311.1

## Investigation and application of web asynchronous communication

Jin Rinan

(Beijing University of Posts and Telecommunications, Beijing 100876)

**Abstract:** The traditional submission of Web data will take the entire form as a unit which is synchronous server communication, but this kind of communication has an obvious flaw is when the user submits the Web form, it often take a long time to wait and the page blank, this reduced the speed of gaining information and the user experience enormously. The appearance of Ajax has broken this traditional mentality, it enables us to be possible to submit a part of form data which is concerned by user to server. This kind of communication can help us can reduce data transmission and also promote user experience. With this word related “newest” the terminology is the XMLHttpRequest object (XHR), it as early as 1999 had already appeared in IE 5. As Ajax can upgrade the capacity of transmission and promote user experience, companies like Google、Flickr use it more and more in there project. We will investagate Web asynchronous communication and discuss about it practical application in project in this article.

**Key words:** Web asynchronous communication; Ajax; JavaScript

## 0 引言

自从采用超文本作为 Web 传输和呈现之后, 我们都是采用这么一套传输方式。当负载比较小的时候, 这并不会体现出有什么不妥。可是当负载比较大, 响应时间要很长, 1 分钟、2 分钟……数分钟的时候, 这种等待就不可忍受了。严重的, 超过响应时间, 服务器干脆告诉你页面不可用。另外, 某些时候, 我只是想改变页面一小部分的数据, 那为什么我必须重新加载整个页面呢? 当软件设计越来越讲究人性化的时候, 这么糟糕的用户体验简直与这种原则背道而驰。为什么老是要让用户等待服务器取数据呢? 至少, 我们应该减少用户等待的时间。现在, 除了程序设计、编码优化和服务器调优之外, 还可以采用 AJAX。

---

**作者简介:** 金日男(1984 年), 男, 硕士研究生, 研究方向: 计算机网络流量监测. E-mail: rinanjin@gmail.com

与传统的 Web 应用不同, AJAX 采用异步交互过程。AJAX 在用户与服务器之间引入一个中间媒介, 从而消除了网络交互过程中的处理—等待—处理—等待缺点。用户的浏览器在执行任务时即装载了 AJAX 引擎。AJAX 引擎用 JavaScript 语言编写, 通常藏在一个隐藏的框架中。它负责编译用户界面及与服务器之间的交互。AJAX 引擎允许用户与应用软件之间的交互过程异步进行, 独立于用户与网络服务器间的交流。现在, 可以用 Javascript 调用 AJAX 引擎来代替产生一个 HTTP 的用户动作, 内存中的数据编辑、页面导航、数据校验这些不需要重新载入整个页面的需求可以交给 AJAX 来执行。这篇文章我们将深入探讨 Ajax 中所用到的技术细节与它在实际当中的应用。

## 1 Web 发展史

讨论 Web 开发技术, 当然要先说说 Web 的起源。众所周知, Web 这个 Internet 上最热门的应用架构是由 Tim Berners-Lee 发明的。Web 的前身是 1980 年 Tim Berners-Lee 负责的 Enquire (Enquire Within Upon Everything 的简称) 项目。1990 年 11 月, 第一个 Web 服务器 nxoc01.cern.ch 开始运行, Tim Berners-Lee 在自己编写的图形化 Web 浏览器 "WorldWideWeb" 上看到了最早的 Web 页面。1991 年, CERN (European Particle Physics Laboratory) 正式发布了 Web 技术标准。目前, 与 Web 相关的各种技术标准都由著名的 W3C 组织 (World Wide Web Consortium) 管理和维护。

从技术层面看, Web 架构的精华有三处: 用超文本技术 (HTML) 实现信息与信息的连接; 用统一资源定位技术 (URI) 实现全球信息的精确定位; 用新的应用层协议 (HTTP) 实现分布式的信息共享。这三个特点无一不与信息的分发、获取和利用有关。其实, Tim Berners-Lee 早就明确无误地告诉我们: "Web 是一个抽象的 (假想的) 信息空间。" 也就是说, 作为 Internet 上的一种应用架构, Web 的首要任务就是向人们提供信息和信息服务<sup>[1]</sup>。

## 2 Web 通信技术

### 2.1 同步通信技术

在传统的 Web 应用中, 一般都是采用“请求-刷新-显示”的模式。即当用户通过点击链接或按钮向服务器发送一个请求时, 都是由服务器接收请求并处理, 处理完毕后返回浏览器进行显示。而在服务器进行处理以及数据传输的时间内, 浏览器一直处于 Loading 状态, 显示为空白或者无响应, 用户只能等待。这种请求是同步的, 也就是说客户端和服务端要互相等待: 请求完成才能获得结果, 使用完这些结果才会有新的请求。但是, 用户需要刷新的或许只是一个数据, 却需要刷新整个网页。

### 2.2 异步通信技术

与传统的 Web 通信方式不同, 异步通信不是将整个页面的数据传输到服务器端, 而是将用户感兴趣的数据传输到服务器端进行处理从而实现页面的局部刷新, 在减少了数据传输的同时也没使页面出现整体刷新, 从而提升了响应性能和用户体验。

### 2.3 异步通信技术的优势

- (1) 不刷新整个页面, 在页面内与服务器通信。
- (2) 使用异步方式与服务器通信, 不需要打断用户的操作, 具有更加迅速的响应能力。
- (3) 应用仅由少量页面组成。大部分交互在页面之内完成, 不需要切换整个页面。

### 3 Web 异步通信的实现

2005年2月18日, Adaptive Path 公司的 Jesse James Garrett 发表了《Ajax: A New Approach to Web Applications》, 提出了 Ajax 这个概念, 并分析了 Google Maps 和 Google Suggest 这两个典型应用。Ajax 混合使用 XHTML、CSS、JavaScript、DOM、XMLHttpRequest 进行 Web 开发将会成为一种趋势<sup>[2]</sup>。

Ajax (Asynchronous JavaScript And XML, 异步的 JavaScript 和 XML) 不是一种新的技术, 它实际上是几种成熟技术的融合, 每种技术都有其独特之处, 合在一起就成了一个功能强大的新技术<sup>[3]</sup>。Ajax 的关键包括:

- (1)使用基于 Web 标准的 XHTML 和 CSS 构建页面。
- (2)使用文档对象模型 (DOM, Document Object Model) 作动态显示和交互。
- (3)使用 XML 或 JSON 做数据交互和操作。
- (4)使用 XMLHttpRequest 进行异步数据接收。
- (5)使用 JavaScript 将它们绑定在一起。

#### 3.1 文档对象模型

DOM 的全称是 Document Object Model, 也即文档对象模型。在应用程序中, 基于 DOM 的 XML 分析器将一个 XML 文档转换成一个对象模型的集合 (通常称 DOM 树), 应用程序正是通过对这个对象模型的操作, 来实现对 XML 文档数据的操作。通过 DOM 接口, 应用程序可以在任何时候访问 XML 文档中的任何一部分数据, 因此, 这种利用 DOM 接口的机制也被称作随机访问机制。

目前, DOM 由三部分组成, 包括: 核心 (core)、HTML 和 XML。核心部分是结构化文档比较底层对象的集合, 这一部分所定义的对象已经完全可以表达出任何 HTML 和 XML 文档中的数据了。HTML 接口和 XML 接口两部分则是专为操作具体的 HTML 文档和 XML 文档所提供的高级接口, 使对这两类文件的操作更加方便。

DOM 有两个版本, 一个是由 W3C 于 1998 年 8 月 18 日通过的 DOM Level 1, 另一个则是正在制定过程中的 DOM Level 2, W3C 已于 2000 年 3 月 7 日通过了 DOM Level 2 的候选推荐版本。

#### 3.2 数据类型

##### 3.2.1 XML 数据类型

XML (eXtensible Markup Language, 可扩展置标语言) 是由 W3C (World Wide Web Consortium, 互联网联合组织) 于 1998 年 2 月发布的一种标准, 是 SGML (Standard Generalized Markup Language, 标准通用置标语言) 的一个简化子集。由于它将 SGML 的丰富功能与 HTML 的易用性结合到了 Web 的应用中, 以一种开放的自我描述方式定义了数据结构, 在描述数据内容的同时能突出对结构的描述, 从而体现出数据之间的关系。这样所组织的数据对于应用程序和用户都是友好的、可操作的。自 XML 推出以来, 迅速得到软件开发商的支持和程序开发人员的喜爱, 显示出强大的生命力。

XML 的主要使用标签来表示数据。标签由包围在一个小于号 (<) 和一个大于号 (>) 之间的文本组成, 例如 <tag>。起始标签 (start tag) 表示一个特定区域的开始, 例如 <start>; 结束标签 (end tag) 定义了一个区域的结束, 除了在小于号之后紧跟着一个斜线 (/) 外,

和起始标签基本一样，例如</end>。SGML 还定义了标签的特性（attribute），它们是定义在小于号和大于号之间的值，例如中的 src 特性。

### 3.2.2 JSON 数据类型

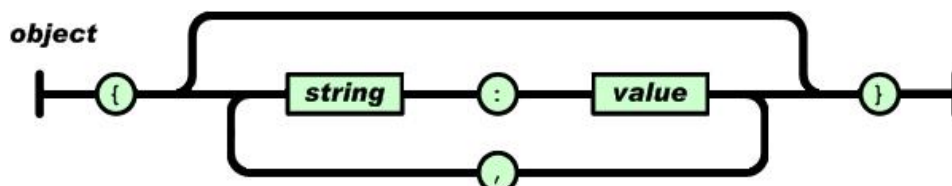
JSON 协议是一种数据传输协议，你也可以把它理解为表达数据协议，JSON 与 XML 不同，JSON 只能用来传输数据，而不能用作文档格式。

JSON 即 JavaScript Object Notation，它是一种轻量级的数据交换格式，非常适合于服务器与 JavaScript 的交互。

尽管有许多宣传关于 XML 如何拥有跨平台，跨语言的优势，然而，除非应用于 Web Services，否则，在普通的 Web 应用中，开发者经常为 XML 的解析伤透了脑筋，无论是服务器端生成或处理 XML，还是客户端用 JavaScript 解析 XML，都常常导致复杂的代码，极低的开发效率。实际上，对于大多数 Web 应用来说，他们根本不需要复杂的 XML 来传输数据，XML 的扩展性很少具有优势，许多 AJAX 应用甚至直接返回 HTML 片段来构建动态 Web 页面。和返回 XML 并解析它相比，返回 HTML 片段大大降低了系统的复杂性，但同时缺少了一定的灵活性。

JSON 具有以下这些形式：

(1) 对象是一个无序的“名称/值”对集合。一个对象以“{”（左括号）开始，“}”（右括号）结束。每个“名称”后跟一个“:”（冒号）；“名称/值”对之间使用“,”（逗号）分隔[4]。



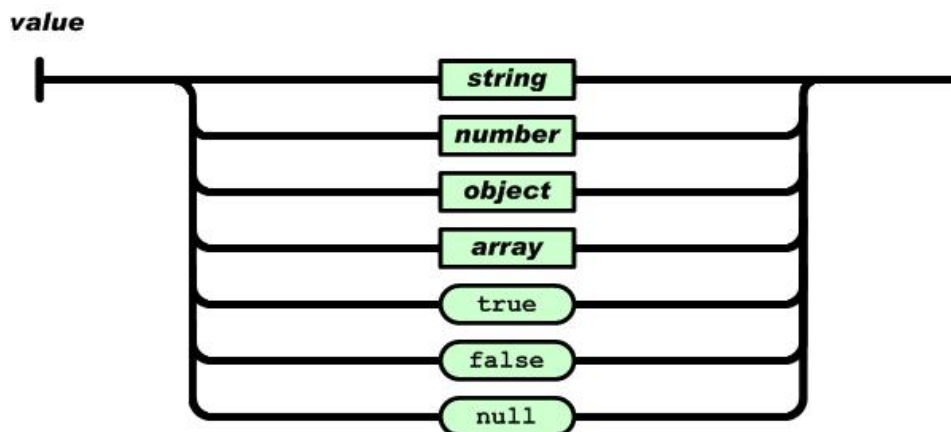


图 3 JSON 值表达式  
Fig.3 "JSON value expression"

(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*) [4]。

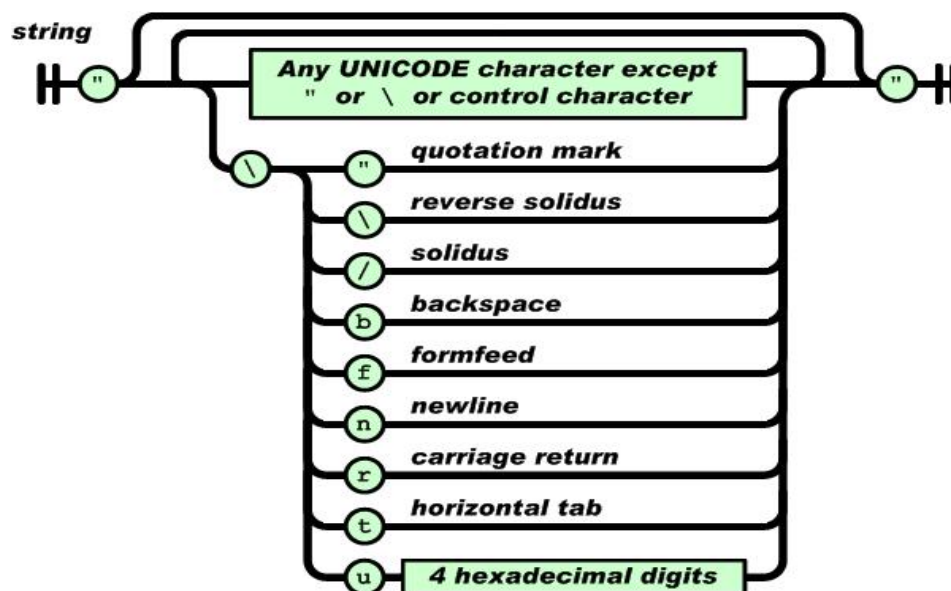


图 4 JSON 字符串表达式  
Fig.4 "JSON string expression"

(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 [4]。



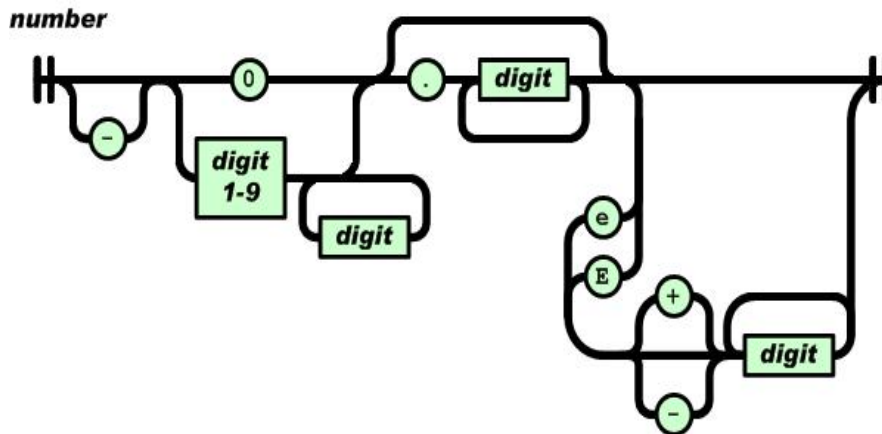


图 5 JSON 数值表达式  
Fig.5 "JSON number expression"

### 3.2.3 XMLHttpRequest 对象

XMLHttpRequest 对象提供了对 HTTP 协议的完全的访问，包括做出 POST 和 HEAD 请求以及普通的 GET 请求的能力。XMLHttpRequest 可以同步或异步地返回 Web 服务器的响应，并且能够以文本或者一个 DOM 文档的形式返回内容。

使用 XMLHttpRequest 脚本化 HTTP 有 3 个步骤<sup>[5]</sup>：

- (1) 创建一个 XMLHttpRequest 对象。
- (2) 指定 HTTP 请求并向一个 Web 服务器提交。
- (3) 同步地或异步地获得服务器的响应。

## 4 Web 系统资源动态监测

### 4.1 开发意义

在开发过程中开发人员往往因为疏漏或者某个功能模块的运行而造成系统濒临瘫痪，这时候就需要对系统运行状况进行检查，针对一个 Web 应用架构，通常所包含的系统资源有 JVM 内存情况与数据库连接池情况，如果将这些资源的实时状况直观显示给开发人员将会大大降低开发过程中对某个模块性能评估的复杂程度。

### 4.2 功能实现

由于对系统功能的监测需要具备实时性，因此不可能采用传统的同步 Web 通信方式，所以这里采用异步传输方式，每隔 5 秒钟就向服务器端发送一次请求刷新各个资源的占用情况。这里所监测的资源包括 JVM 内存使用情况和数据库连接池的占用情况。

客户端采用了 jQuery 技术，这是一个简洁快速也是一个新型的 JavaScript 库，它能让你在你的网页上简单的操作文档、处理事件、运行动画效果或者添加 Ajax 交互。jQuery 的设计会改变你写 JavaScript 代码的方式。jQuery 适合于设计师、开发者以及那些爱好者，同样适合于商业开发，可以说 jQuery 适合任何 JavaScript 应用的地方，可用于不用的应用程序。jQuery 是一个轻量级的脚本，其代码非常小巧，jQuery 包只有 19KB 左右。jQuery 支持 CSS1-CSS3，以及基本的 XPath，jQuery 是跨浏览器的，它支持的浏览器包括 IE 6.0+，FF 1.5+，Safari 2.0+，Opera 9.0+ 等各大主流浏览器<sup>[6]</sup>。

服务器端采用了 Struts2 框架，Apache Struts2 是一个优雅的，可扩展的 Web 框架。框架设计的目标贯穿整个开发周期，从开发到发布，包括维护的整个过程。

客户端与服务端端的通信过程如下图所示:

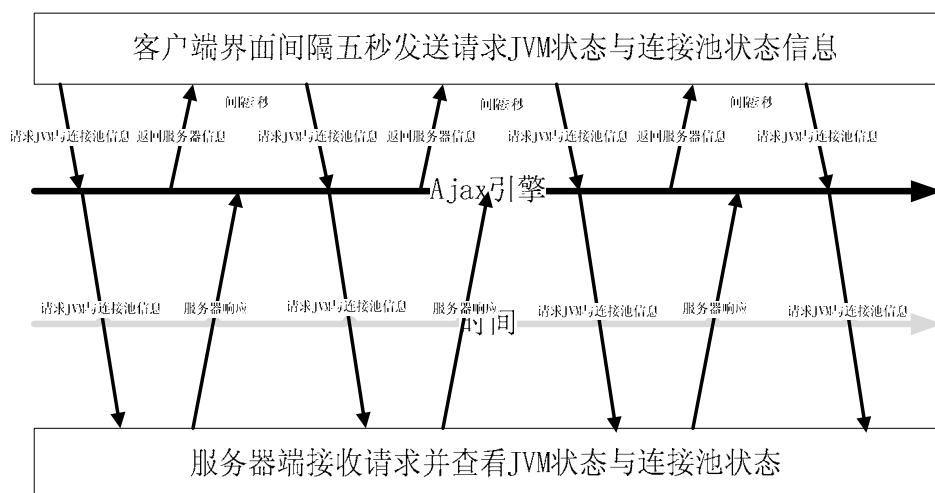


图 6 系统资源监测模块通信过程  
Fig.6 "Communication process of system monitor"

服务器端页面如下:

Java VM (Java虚拟机) 内存
内存总量: 6.64 MB    已用内存: 4.67 MB    29.6% 空闲
数据库连接池运行状况
1、DBPool Key: defaultkey RuntimeInfo: total num: 2, busy num: 0, idle num: 2

图 7 初始系统状态  
Fig.7 "Begin of system"

<b>Java VM (Java虚拟机) 内存</b>
内存总量: 8.64 MB    已用内存: 4.56 MB    31.3% 空闲
<b>数据库连接池运行情况</b>
1. DBPool Key: defaultkey RuntimeInfo: total num: 2, busy num: 0, idle num: 2

图 8 5 秒钟之后系统状态  
Fig.8 "Five seconds later"

## 5 结论

本文介绍了 Web 异步通信的原理以及在实际中的实现与应用，但是介于篇幅所限很难将 Web 异步通信的所有情景都列举出来。由于 Ajax 带给客户的全新体验和便捷的操作相信在不久的将来他将在 Web 应用领域发挥出越来越大的作用。

## 致谢

在自己的研究过程中我得到了实验室周文莉老师和冯韬略老师的支持,在此表示衷心的感谢。

### [参考文献] (References)

- [1] Behrouz A. Forouzan, Sophia Chung Fegan. TCP/IP SUITE[M]. The McGraw-Hill Companies. Inc, 2006
- [2] Jesse James Garrett. The Element of User's Expreience[M]. New Riders Publishing, 2005
- [3] 孙鑫. Struts2 深入详解[M]. 电子工业出版社, 2008
- [4] Douglas Crockford. Introducing JSON[OL]. [1999].<http://www.json.org/>
- [5] David Flanagan. JavaScript: The Definitive Guide[M]. O'Reilly Media, Inc. 2007
- [6] John Resig. The jQuery Project [OL]. [2010]. <http://jquery.com/>