

Broadview®
www.broadview.com.cn

Ajax

开发精要

概念、案例与框架

柯自聪 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

TP393.09
83

Ajax

开发精要

概念、案例与框架

柯自聪
编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书从概念、案例与框架三个角度来详细阐述 Ajax 开发技术,先介绍 Ajax 的由来、优势及其在 Web 开发中的地位,接着详细介绍 Ajax 的各项组成技术、封装、开源和商业框架等各种相关知识,并提供丰富实用的开发案例和综合案例,引导读者一步步地了解并掌握利用 Ajax 进行 Web 应用程序开发的方方面面。

本书可供各类 Web 开发、网站建设与网页设计人员参考。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Ajax 开发精要:概念、案例与框架 / 柯自聪编著. —北京:电子工业出版社, 2006.5
ISBN 7-121-02468-3

I. A… II. 柯… III. 计算机网络—程序设计 IV. TP393.09

中国版本图书馆 CIP 数据核字 (2006) 第 032200 号

责任编辑:顾慧芳 胡铭娅

印 刷:北京智力达印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×980 1/16 印张:28.25 字数:650 千字

印 次:2006 年 5 月第 1 次印刷

印 数:6000 册 定价:45.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

Ajax (Asynchronous Javascript and XML) 是 2005 年 2 月才正式提出的一项综合技术,其主要特点是为 Web 开发提供异步的数据传输和交换方式,可以在不重载(Reload)刷新(Refresh)界面的情况下与服务器进行数据交换。

随着 Ajax 在 Google Mail、Google Suggestion 等产品的广泛运用, Ajax 正逐渐吸引着 Web 开发界的眼球,成为 Web 开发的热点。Ajax 通过综合运用 Javascript、XHTML 和 CSS、DOM、XML 和 XSTL、XMLHttpRequest 的技术,在提高数据交互速度,改善用户体验等方面发挥着重要作用。其中:

- XHTML 和 CSS 具有灵活、可拼装、基本、结构化等特点,被 Ajax 用来将系统数据呈现给用户,构建一个标准化的用户界面。
- DOM 由于对 HTML、XHTML、XML 拥有强大的操作能力以及结构化的操作方法,故被 Ajax 用来实现数据动态的显示与交互。
- XML 和 XSTL 在结构化的数据表示、传输、交互、共享方面有特别大的优势。因此,XML 和 XSTL 在 Ajax 中主要用做数据交换与处理。
- XMLHttpRequest 是 XMLHttpRequest 组件的对象,它使 Ajax 可以向服务器发起异步的通信请求、并接收服务器返回的数据。这样, Ajax 可以像桌面应用程序一样在不刷新界面的情况下与服务器进行数据交换。
- Javascript 因为具备简单、灵活、丰富的客户端控制和表现等优点,所以被用来操作 HTML、XHTML、CSS,绑定和处理所有的数据。

在 Ajax 出现之前, Javascript、XHTML 和 CSS、DOM、XML 和 XSTL、XMLHttpRequest 基本上是各自为政的, Ajax 让这些技术第一次有了交集,并且组成一个整体。相信随着 Ajax 的逐渐成熟和更加广泛的运用, Ajax 将为 Web 开发应用带来更多的惊喜。

在写作本书之前,笔者在个人 Blog 上发布了《Ajax 开发简略》系列文章,引起了读者的强烈反响,并经 CSDN、dev2dev 等知名网站、论坛转载和收录。本书将继续延续《Ajax 开发简略》系列文章通俗易懂、理论与案例并重等特点,详细介绍 Ajax 的历史、各项组成技术、封装、开源和商业框架等各种相关知识,并提供丰富实用的开发案例和综合案例,引导读者一步步地了解并掌握 Ajax 的开发方法。

本书读者对象

本书系统地介绍了利用 Ajax 进行 Web 应用程序开发的各个方面。适合的读者对象有：

- 需要利用 Ajax 来提供异步数据传输和交换方式的 Web 开发、设计人员。
- 在 Web 开发设计中需要不刷新页面的情况下，能与服务器进行交互的 Web 开发、设计人员。
- 需要提高 Web 系统数据传输速率的 Web 开发、设计人员。
- 需要改善 Web 系统用户体验的 Web 开发、设计人员。
- 对 Web 2.0 有浓厚兴趣的读者。
- 希望详细了解 Ajax 技术的其他读者。

本书特色

目前市场上有关 Ajax 的图书正在纷纷登场，但本书有如下的重要特色：

- 系统概述了 Ajax 开发的方方面面，以帮助读者全面了解 Ajax。
- 遵循认知习惯，从简单到复杂、从单独编码到封装、再到通用框架，帮助读者一步一步地掌握 Ajax 的开发。
- 结合相关理论提供丰富、真实和完整的开发案例，让 Ajax 能真正应用到读者的实际开发工作中。
- 由一线开发人员专门针对国内开发人员撰写，更适合开发人员阅读与实践。
- 针对不同层次读者的需要，在有些章节末尾，推荐深入学习的参考资料。

示例应用和代码清单

本书中涉及的所有应用程序及其源文件都可以从博文视点 (<http://www.broadview.com.cn>) 网站资源下载文件中得到。本书凡涉及需要动态解析的页面都采用 Java, JSP 语言编写，代码提供了必要的注释，读者可以根据需要用其他语言改写。测试某一章节的应用代码是很容易的，只要简单地把应用复制到 JSP Web 容器（比如 Tomcat），并根据需要和书中的说明进行相应的配置即可。每个应用的名称都是和相应的章节对应的，比如，AjaxCh01 是第 1 章随带的应用，AjaxCh05 是第 5 章随带的应用，依次类推。如果一个章节下面有多个示例应用，则分别用类似 AjaxCh01a, AjaxCh01b, AjaxCh01c 的规则命名。

本书内容

本书分为概念篇（第 1~4 章）、开发案例篇（5~9 章）、扩展篇（10~12 章）三个部分共 12 章组成。具体内容如下：

第 1 章 Ajax 介绍

本章介绍 Ajax 的起源及其具体含义，以及 Ajax 在 Web 应用程序大家庭中的地位，并简述如何应用 Ajax，帮助读者形成对 Ajax 的初步印象。

第 2 章 B/S 请求响应机制与 Web 开发模式

本章简要介绍了 HTTP 协议，以及 Web 系统遵循的 B/S 请求响应机制，概括了当前常用的几种 Web 开发技术（CGI，ASP，JSP，CGI，PHP，.NET）所采用的开发模式，并阐述了 Ajax 在 Web 系统中所处的位置。另外，本章还为后续章节的案例开发和测试搭建了一个可用的开发环境。

第 3 章 Ajax 概览

本章从宏观的角度了解为什么使用 Ajax，Ajax 有什么优点和缺点，Ajax 都应用到了哪些系统和领域中，以及采用各种解决方案的开发人员应该如何选择相应的 Ajax 技术。另外，本章还举了一个简单的例子，让读者快速体验 Ajax 的应用效果。

第 4 章 Ajax 用到的技术

Ajax 是一项综合的技术。本章将引导读者掌握开发 Ajax 需要的相关技术知识，尤其是 DOM、XMLHttpRequest 和 XML。通过本章的学习，读者能够具备 Ajax 开发的技术知识。

第 5 章 简单的 Ajax 开发框架

本章结合第 4 章提到的各项技术，在详细的讲解和案例中形成一个简单的典型开发框架。本书的第 6~9 章都将使用并扩展这个框架做示范。

第 6 章 使用 responseText 处理返回的普通文本信息

本章列举了 4 个案例来详细阐述如何运用 responseText 处理返回的普通文本信息。通过这些案例，读者可体会到 Ajax 对于提高数据传输速度、改善用户体验的独特作用。

第 7 章 使用 responseXML 处理返回的 XML 信息

本章列举了 4 个案例来详细阐述如何运用 responseXML 处理返回的 XML 信息。通过这些案例，读者可以体会到 XML 在结构化数据表示方面的重要作用，了解 Ajax 如

何与 XML 交互。

第 8 章 在普通文本和 XML 文档之间权衡

responseText 和 responseXML 是 Ajax 处理返回服务器信息的两种方式。本章列举了两者的优缺点，帮助读者权衡如何在需要的时候选择合适的处理方式。

第 9 章 模拟 MSN Space

MSN Space 是 Ajax 的一个典型应用。本章运用 Ajax 模拟了 MSN Space 的几个重要功能：固定链接、操作评论，以及管理列表项。

第 10 章 Ajax 封装

本章根据第 5~9 章的内容，总结出四种封装 Ajax 的方法，以达到简化开发、提高开发效率、复用代码的目的。作为第 11 章、第 12 章的引子，本章的目的在于引导读者思考如何简化 Ajax 开发，通过适当的封装提高开发效率。

第 11 章 开源 Ajax 框架

本章简要介绍了包括 DWR, Ajax Tags, AjaxAnywhere, SWATO 等在内的几个开源框架，并对如何使用这些框架进行了示范。通过对比介绍，读者可以选用相应的开源框架、加入到自己的系统中，以实现加速 Ajax 编码的目的。

第 12 章 商业 Ajax 框架

本章列举了两个比较典型的商业框架，讲述了 Ajax 在商业产品开发上的运用，以作为开源 Ajax 框架的补充、借鉴和参考。

本书开发案例篇和扩展篇部分的案例需求，使用自然语言的描述方法。

排版约定

本书使用下列排版约定：

粗体 (**Bold**)

用于强调重点，或者标识术语的第一次使用。

斜体 (*Italic*)

此外，电子邮件地址、网址、FTP 地址、文件名、目录名和新闻组也用斜体表示，Java 的类名称也用斜体表示。

由于笔者水平和时间有限，书中错误在所难免，恳请读者批评指正。

在本书的写作过程中，我的家人以及玉锦等朋友给予了源源不断的支持和鼓励，在

此表示特别的感谢。希望在本书完成之后，能有时间多陪伴家人和朋友。

本书编写过程中得到博文视点公司的大力支持，本书的策划编辑张毅给予了特别的帮助，在此表示衷心的感谢。

同时，也感谢一直关注《Ajax 开发简略》和本书的朋友。

笔者的联系方式如下：

Blog: <http://www.blogjava.net/eamoi>

电子邮件: eamoi@163.com

柯自聪

2006 年 4 月

目 录

概 念 篇

第 1 章 Ajax 介绍.....	2	3.5 Ajax 的缺陷.....	41
1.1 Ajax 的由来.....	2	3.6 一个简单的 Ajax 例子.....	42
1.2 Ajax 的定义.....	3	3.7 小结.....	47
1.3 Web 应用程序的解决方案.....	5	第 4 章 Ajax 用到的技术.....	48
1.4 Ajax 的工作方式.....	7	4.1 HTML.....	48
1.5 小结.....	8	4.1.1 HTML 与 Ajax.....	49
第 2 章 B/S 请求响应机制与		4.1.2 HTML 常用标记.....	49
Web 开发模式.....	9	4.2 XML.....	62
2.1 HTTP 请求响应模型.....	9	4.2.1 XML 与 Ajax.....	63
2.2 B/S 架构的请求响应机制.....	12	4.2.2 XML 基础.....	63
2.3 Web 开发模式.....	13	4.2.3 文档类型定义及声明.....	68
2.4 开发环境的设置与配置.....	19	4.2.4 XML 名域.....	71
2.4.1 下载 Java 开发		4.2.5 操纵和解析 XML.....	73
工具包 (JDK).....	19	4.2.6 选择合适的 XML 生成方式.....	78
2.4.2 下载安装 Tomcat.....	20	4.3 XHTML 与 CSS.....	83
2.4.3 下载安装 Eclipse 3.1.1.....	20	4.3.1 XHTML、CSS 与 Ajax.....	84
2.4.4 创建开发目录.....	22	4.3.2 XHTML 与 CSS 基础.....	84
2.5 小结.....	24	4.3.3 XHTML 语法约束.....	85
2.6 参考资料.....	25	4.4 Javascript.....	87
第 3 章 Ajax 概览.....	26	4.4.1 Javascript 与 Ajax.....	88
3.1 为什么使用 Ajax.....	26	4.4.2 Javascript 基本数据结构.....	89
3.2 谁在使用 Ajax.....	30	4.4.3 Javascript 表达式和运算符.....	91
3.3 Ajax 的应用方向.....	32	4.4.4 Javascript 的流程控制	
3.4 基于各种 Web 解决方案		和函数.....	91
的 Ajax 技术选择路线.....	35	4.4.5 Javascript 的对象.....	94
		4.4.6 窗口及输入输出.....	98

4.5	DOM	98		XML 文档	113
4.5.1	DOM 与 Ajax	99	4.6	XMLHttpRequest 对象	116
4.5.2	DOM 眼中的 HTML 文档——树	99	4.6.1	XMLHttpRequest 对象与 Ajax	116
4.5.3	HTML 文档的节点	100	4.6.2	XMLHttpRequest 对象的属性和方法	116
4.5.4	使用 DOM 操作 HTML 文档	102	4.7	小结	122
4.5.5	使用 DOM 操作		4.8	参考资料	123

开发案例篇

第 5 章	简单的 Ajax 开发框架	126	6.3.2	案例解析	172
5.1	如何使用 Ajax	126	6.3.3	应用范围	183
5.2	初始化对象	128	6.4	案例 4——级联的下拉列表	184
5.3	指定响应处理函数	129	6.4.1	需求说明	187
5.4	发出 http 请求	129	6.4.2	案例解析	188
5.5	处理服务器返回的信息	130	6.4.3	应用范围	201
5.6	一个初步的 Ajax 开发框架	131	6.5	小结	202
5.7	异常处理机制	136	第 7 章	使用 responseXML 处理返回的 XML 信息	204
5.8	中文乱码问题及其解决方案	137	7.1	案例 5——获取元对象数据	205
5.9	小结	140	7.1.1	需求说明	206
第 6 章	使用.responseText 处理返回的普通文本信息	141	7.1.2	范例解析	206
6.1	案例 1——数据校验	141	7.1.3	应用范围	224
6.1.1	需求说明	142	7.2	案例 6——处理列表数据	224
6.1.2	案例解析	143	7.2.1	需求说明	226
6.1.3	应用范围	152	7.2.2	案例解析	227
6.2	案例 2——级联菜单	152	7.2.3	应用范围	239
6.2.1	需求说明	153	7.3	案例 7——处理级联的数据	240
6.2.2	案例解析	154	7.3.1	需求说明	240
6.2.3	应用范围	170	7.3.2	案例解析	242
6.3	案例 3——条目内容	170	7.3.3	应用范围	251
6.3.1	需求说明	172	7.4	案例 8——更新保存数据	251

7.4.1 需求说明	252	9.3 实体类和服务类	296
7.4.2 案例解析	252	9.4 初始化共享空间首页	324
7.4.3 应用范围	280	9.5 固定链接	328
7.5 小结	281	9.6 操作评论	330
第 8 章 在普通文本和 XML 文档 之间权衡	282	9.6.1 添加评论	330
8.1 何时返回普通文本	282	9.6.2 阅读评论	336
8.2 何时返回 XML 文档	283	9.7 引用通告	336
8.3 innerHTML 的应用	284	9.8 管理列表项	339
8.4 小结	285	9.8.1 初始化自定义列表首页	339
第 9 章 模拟 MSN Space	286	9.8.2 添加列表项目	346
9.1 MSN Space 中的 Ajax	286	9.8.3 编辑列表项目	349
9.2 持久化机制	292	9.8.4 删除列表项目	352
		9.9 小结	354

扩展篇

第 10 章 Ajax 封装	358	11.2 Ajax Tags	379
10.1 封装客户端 Ajax 操作	358	11.2.1 Ajax Tags 简介	380
10.2 自定义标签扩展现有 的 HTML 控件	359	11.2.2 Ajax Tags 标签及其使用 方法	383
10.3 规范服务器的返回数据	360	11.2.3 Ajax Tags 的适用范围	399
10.4 客户端脚本与服务器端 对象的交互	361	11.3 AjaxAnywhere	400
10.5 小结	361	11.3.1 AjaxAnywhere 简介	400
第 11 章 开源 Ajax 框架	363	11.3.2 Ajax Anywhere 安装	401
11.1 DWR	363	11.3.3 AjaxAnywhere 的类库 及其用法	402
11.1.1 DWR 简介	363	11.3.4 试用 AjaxAnywhere	417
11.1.2 DWR 使用入门	365	11.3.5 Ajax Anywhere 的适用 范围	421
11.1.3 试用 DWR	373	11.4 SWATO	422
11.1.4 DWR 的适用范围	379	11.4.1 SWATO 简介	422

11.4.2	SWATO 快速安装	423	第 12 章	商业 Ajax 框架	431
11.4.3	第一个服务类	424	12.1	Google Ajaxstl	431
11.4.4	注册服务	426	12.2	锐道 Dorado	431
11.4.5	使用服务	427	12.3	小结	434
11.4.6	SWATO 的适用范围	429	参考文献		435
11.5	小结	430			

概 念 篇

- ◇ 第 1 章 Ajax 介绍
- ◇ 第 2 章 B/S 请求响应机制与 Web 开发模式
- ◇ 第 3 章 Ajax 概览
- ◇ 第 4 章 Ajax 用到的技术

Ajax 介绍

目前, Web 应用程序正在以几何数量级的速度增长, Web 应用程序的解决技术也正逐步成熟。Java (JSP)、.NET (C#)、PHP、ASP 已逐渐成为 Web 应用程序开发的技术主流。Web 应用程序不再是简单显示信息的网站, 而逐步融合核心的业务逻辑, 成为了 IT 领域的业务处理平台。大家也习惯把 Web 应用程序部署在因特网或者内部网中, 将日常处理业务交给 Web 应用程序处理。因此, 如今的 Web 应用程序承载了太多的期待。

2005 年 2 月, 有一项“老”技术让业界重新审视 Web 应用程序的通信模式, 对 Web 应用程序的数据传输速度和用户体验提出了更高的要求 and 期望, 也让业界看到了成功的曙光。业界都希望借此技术, 使 Web 应用程序继承桌面应用程序反应灵敏、胖客户端、客户体验优秀等优点。这项技术就是 Ajax。

本章将简单阐述 Ajax 的由来、定义及其使用方法, 在讲述当前几种主流 Web 应用程序开发技术的基础上让读者认识 Ajax 在 Web 应用程序大家庭中的位置。

1.1 Ajax 的由来

2005 年 2 月, Adaptive Path 公司的 Jesse James Garrett 在他的“Ajax: A New Approach to Web Applications”(<http://www.adaptivepath.com/publications/essays/archives/000385.php>) 一文中首次提出了 Ajax 的概念。Garrett 将一种综合使用 Javascript、XHTML 和 CSS、DOM、XMLHttpRequest、XML 和 XSTL 的技术称之为 Ajax。一直以来, Web 应用程序的交互效果都不如桌面应用程序, 比如客户端功能和用户体验效果, 这点让很多 Web 应用程序开发人员和用户耿耿于怀。Garrett 期待 Ajax 能够消除 Web 应用程序和桌面应用程序在系统人机交互和用户体验方面的差距, 至少 Google Suggest、Google Maps 等产品已经让人们看到了胜利的曙光。

其实在 Garrett 的文章发表之前, Ajax 已经在一些内部的 Web 系统中得到了应用, Google 也已将 Ajax 技术应用到包括 Google Suggest、GMail 等多个产品中, 只是这个时候 Ajax 还“藏在深闺人不识”, 业界知之甚少。原来的 Ajax 有一个冗长的名字: 异步的 Javascript、CSS、DOM 和 XMLHttpRequest (Asynchronous Javascript+CSS+DOM+XMLHttpRequest)。Garrett 则扮演了“临门一脚”的角色。Google 所用的这种基于多种交互操作技术的新式动态 Web 应用技术被他称为 Ajax, 即 Asynchronous Javascript and XML (异步 Javascript 和 XML) 的缩写, 这个形象简短的名字概括了这种技术的特点。现在, “Ajax”这个简短形象的缩写已风靡全世界。

Garrett 的文章发表后, 业界对 Ajax 兴趣浓厚, 咨询的信件纷至沓来。Garrett 也专门整理了一份 FAQ, 来解答读者的问题。从 Garrett 的文章和 FAQ 中, 可以完整地看出 Ajax 的轮廓, 了解 Ajax 到底是什么。

1.2 Ajax 的定义

Ajax 不是一项新的技术, 只是多种技术的综合, 或者是设计方式, 包括 Javascript、XHTML 和 CSS、DOM、XML 和 XSTL、XMLHttpRequest 等技术。其中:

- 使用 XHTML 和 CSS 实现标准化的呈现界面。
- 使用 DOM 实现动态的显示和交互。
- 使用 XMLHttpRequest 实现与服务器的异步通信。
- 使用 Javascript 将 XHTML、DOM、XML、XMLHttpRequest 绑定。

各种技术在 Ajax 引擎中的作用如图 1-1 所示。

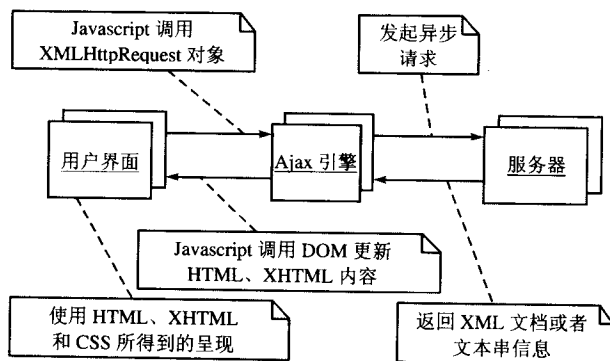


图 1-1 各种技术在 Ajax 引擎中的作用

整个交互通信过程是异步进行的, 所以 Garrett 采用了 Ajax (Asynchronous Javascript and XML) 这个名称。

Ajax 的主要功能在于，将浏览器客户端和服务端传统的同步交互通信方式改变为异步通信交互方式，从而丰富了浏览器客户端功能，解决了浏览器频繁刷新页面等待数据传输的问题，改善了 Web 应用程序的用户体验。使用 Ajax，就算不重载刷新 Web 页面，用户也可以顺利地快速得到 Web 服务器的数据。

典型地，看看 Google Maps 的功能 (<http://maps.google.com>)。当用户拖动地图左上角的放大标尺的时候，Web 页上的地图立即被放大，页面却没有刷新。当按住鼠标左键移动地图的时候，地图跟着移动。这个过程是快速的，而期间用户并没有向服务器提交表单或者单击一个超链接。如果用传统的 Web 应用程序交互思维来理解，这个过程是难以理解的。这正是 Ajax 的魅力。

作为 Web 应用程序一部分的 Ajax 的生命周期更像桌面系统的 GUI，而 DOM 则扮演了类似 GUI 控件的角色。Javascript 脚本对 DOM 注册事件监听器，操作 DOM 响应事件。在响应事件的过程中，Web 服务器可能被调用。这个调用是异步进行的，所以事件监听阶段和事件响应阶段是分开的。

下面是一个典型的浏览器中 Ajax 应用的生命周期。

- 用户访问：用户访问一个网站，比如点击链接或者在浏览器中输入网站的 URL 地址。
- 页面初始化：页面初始化加载，准备处理用户输入或者刷新页面内容。
- 触发浏览器事件：浏览器触发一个事件，比如鼠标单击或者按下键盘。
- 向服务器发起请求：浏览器向服务器发出一个请求。
- 服务器处理请求：服务器收到浏览器发出的请求，调用业务逻辑接口处理请求。
- 服务器响应请求：服务器响应浏览器发出的请求，将处理结果返回。这个返回结果传递给在发出请求时指定的请求调用函数。
- 浏览器更新页面：请求调用函数根据响应结果更新 DOM 内容，比如 DOM 变量或者任何的 Javascript 变量，更新页面内容。

“触发浏览器事件→向服务器发起请求→服务器处理请求→服务器响应请求→浏览器更新页面”这个过程是可以多次循环的，如图 1-2 所示。在循环过程中通常会有很多的变量产生，很多事件也可能在客户端解决而不提交到服务器。有些 Ajax 应用的生命周期可能是短暂的，随着用户提交表单或者重新刷新页面而结束。其他则持续呈现在浏览器上（用户不重新刷新页面，也不提交表单），一直响应用户的行为。

值得指出的是，浏览器触发事件和向服务器发起请求是在一个线程上的，而服务器响应和浏览器更新页面是在不同线程上的。这是由服务器请求的异步特性所决定的。也可以配置 XMLHttpRequest 对象实现异步调用，只是现在还缺少比较有效的处理办法。

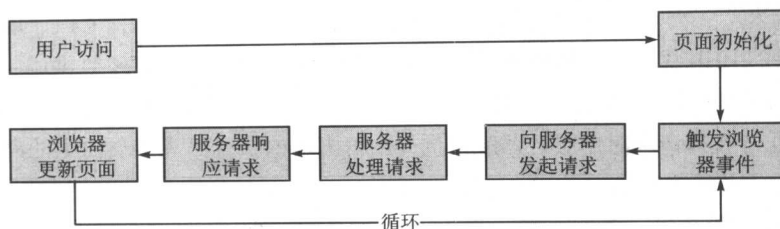


图 1-2 Ajax 的生命周期

1.3 Web 应用程序的解决方案

随着 Web 的发展，Web 应用程序设计也日趋复杂。Web 编程语言简化了原本非常困难的设计，并且将很多不可能的设想变成现实。

如今，Java、.NET、PHP、ASP 已经承担起建立 Web 应用程序的任务，成为 Web 应用程序大家庭的重要成员，使开发人员可以高效率地创建服务器端应用程序。

1. Perl、CGI

Perl (Practical Extraction and Report Language) 是比较早期的动态网站编程技术，具有对文本进行相当好的处理和格式化能力，可以通过使用代码模块进行扩展。

在 Web 发展早期，CGI (Common Gateway Interface, 公共网关接口) 作为 Perl 在 Web 服务器环境中的具体应用，曾经是开发动态 Web 内容的惟一方案。然而，CGI 并非是一种高效率的解决方案，对每一个传过来的请求，Web 服务器都必须创建一个新的操作系统进程，载入一个解释器和一个脚本，执行这个脚本，然后把这些全部释放掉。这使得服务器承受着繁重的负担，而且当 Web 通信量增加时性能会变得很低。

2. ASP

ASP (Active Server Pages) 是继 CGI 之后一项广受欢迎的动态网站开发技术。ASP 允许开发人员在普通网页中加入脚本代码，比如 VBScript 和 Javascript，来产生动态部分。对于复杂的代码，脚本代码还可以调用 DOM (ActiveX) 组件。当一个 ASP 页面被请求时，服务器执行页面中的代码；执行结果被放入页面中，然后将静态和动态部分组合起来，并发回浏览器。ASP 将业务逻辑代码包含在 <%%> 中间，通过类似 VBScript 的 ASP 语法，实现复杂业务逻辑。

ASP 与微软的 IIS 服务器绑定在一起。由于其依赖本地的 COM 代码作为组件模型，所以主要用于 Windows 平台。

3. PHP

PHP 则是一种开源的 Web 脚本语言。与 ASP 类似，其允许开发人员在普通网页中加