



疯狂 Ajax 讲义

—Prototype/jQuery+DWR+
Spring+Hibernate**整合开发**

李刚 编著

疯狂源自梦想

技术成就辉煌

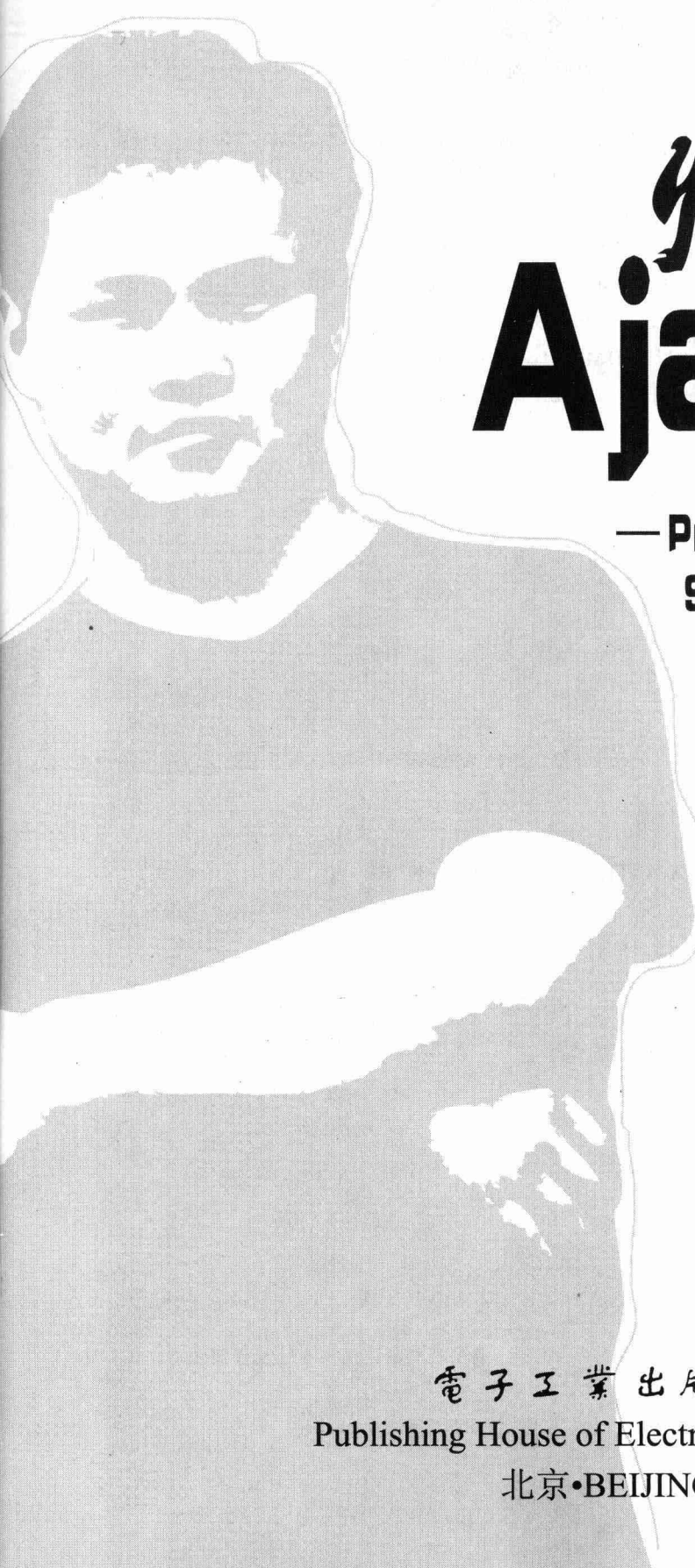
疯狂源自梦想
技术成就辉煌



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>



疯狂 Ajax 讲义

— Prototype/jQuery+DWR+
Spring+Hibernate 整合开发

李刚 编著

電子工業出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是《基于 J2EE 的 Ajax 宝典》的第二版。《基于 J2EE 的 Ajax 宝典》面市近 2 年, 作为 Ajax 领域最全面、实用的图书, 一直深受读者的好评。

全书主要分为三个部分。第一部分介绍了 XHTML、CSS、JavaScript 和 DOM 编程等内容。第二部分详细介绍了 Prototype、jQuery、DWR、AjaxTags 等四个最常用的 Ajax 框架的用法, 并针对每个框架提供了一个实用案例。这两个部分是笔者在“疯狂 Java 实训营”的培训讲义, 是本书的重点部分。第三部分则提供了 2 个综合性案例: Blog 系统和电子拍卖系统, 让读者将前面所学真正应用到实际项目中。

本书绝大部分章节后都提供了相应的编程习题, 供开发者巩固所学, 将理论融入实际开发之用。关于这些编程习题的解题思路和参考答案可登录 <http://www.crazyjava.org> 获取。

本书是疯狂 Java 体系丛书之一, 前 8 章基本以 XHTML、JavaScript 和 DOM 编程为主, 无须任何基础即可阅读; 第 9 章以后的内容则需要掌握 Spring、Hibernate 等 Java EE 知识, 建议先认真阅读疯狂 Java 体系的《轻量级 Java EE 企业应用实战》一书。

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

疯狂 Ajax 讲义——Prototype/jQuery+DWR+Spring+Hibernate 整合开发/李刚编著. —北京: 电子工业出版社, 2009.4
ISBN 978-7-121-08440-9

I. 疯… II. 李… III. ①计算机网络—程序设计②JAVA 语言—程序设计 IV. TP393.09 TP312

中国版本图书馆 CIP 数据核字 (2009) 第 030104 号

责任编辑: 朱沐红

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

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

开 本: 850×1168 1/16 印张: 39.5 字数: 1127 千字

印 次: 2009 年 4 月第 1 次印刷

印 数: 4000 册 定价: 69.00 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



前言

Ajax 技术已经不再是新技术，它已经成为企业开发中应用最广泛的技术之一，不管采用什么样的开发平台：Java EE 也好，.NET 也好，PHP 也好，Ruby on Rails 也好，只要开发 B/S 架构的应用，那么表现层就一定会使用 Ajax 技术。

Ajax 技术采用异步方式发送请求，避免了每个请求对应一个页面的模式，允许在一个页面发送多个请求，从而可以更大程度地利用已下载的页面，服务器每次响应生成的只是必需的数据，无须响应生成整个页面。对用户而言，发送异步请求不会阻塞当前的浏览器线程，浏览器可以继续下一步操作：比如继续浏览或再次发送异步请求。因此用户将不会处于等待状态，而是感觉自己一直与应用处于交互状态，从而带给了用户连续的体验。

Ajax 技术是 Web 2.0 的重要技术之一，互连网上各种 Blog 系统、RSS，以及 Wiki 系统和 SNS 交友网络等，都大量使用了 Ajax 技术。

Ajax 技术还催生了大量的网页游戏。国内的很多游戏运营商纷纷推出了自己的网页游戏。网页游戏具有无须下载、安装，即开即玩、简单便捷的特征，尤其对办公室上班族具有较大的吸引力，因此也具有很好的市场前景。在这种网页游戏中，每个网页上都包含了大量制作精美的图片，当游戏玩家单击、双击这些图片时，系统将采用 Ajax 技术与远程服务器通信，这是绝大部分网页游戏的底层运行机制。

2007 年出版的《基于 J2EE 的 Ajax 宝典》具有全面、专业的特征，书中不仅深入介绍了 Ajax 编程的底层原理和技术，还全面介绍了 Prototype、Dojo、DWR、JSON-RPC-Java 和 AjaxTags 等 5 个 Ajax 框架。不过，Dojo 版本更新太快，这一点限制了它在实际企业开发中的应用。本书作为《基于 J2EE 的 Ajax 宝典》的第二版，详细介绍了 XHTML、CSS、JavaScript、DOM 和 JavaScript 事件机制等基础知识，重点分析了 XMLHttpRequest 对象的运行机制和运行原理。Ajax 框架的介绍部分，Prototype、DWR、AjaxTags 等框架升级到了最新版本，另外新增介绍了 jQuery 框架。

本书的创作感言



写一本书真的很累！每次一本书写到最后几章时，都会有一种近似虚脱的感觉。此外，如果还遭遇一些来自外界的困扰，就让人更加难以静下来做事。不过，笔者现在的主要职业是培训，需要不断地面对新的学生，有责任引导他们进入软件开发行业，这大概是支持自己继续写下去的一个动力吧。

最初，笔者写书仅仅是为了作为笔者的培训教材，帮助自己的学生能更好地理解自己所讲授的内容，所以总是尽量使用清晰条理的方式来组织内容，用实用、易操作的实例来演示开发，用通俗易懂的语言进行表达。希望把实际企业开发中解决问题的方法，用通俗、简单的语言告诉学生。

在本书创作过程中，笔者一度感到非常困惑：其实笔者这些书的学术价值真的很少，因为基本上没有什么创新，绝大部分都是前人的智慧。充其量，笔者只是进行了再归纳、总结，于是难免感到意兴阑珊。

无聊中和一位美籍华人（一位资深 CTO）在 Skype 上聊天，他告诉笔者：印度的程序员数量大概是中国程序员的 100 倍，因为印度有大量的程序员基数，所以就孕育出了大量优秀的程序员。中国人，虽然是世界上非常聪明的种族之一，但中国程序员太少了，所以中国产生的优秀程序员也很少。你写的书虽然没有什么创新，但只要能把实际软件开发的方法和经验传播开来，让更多的年轻人走进软件开发行业，你就为中国的软件开发业做出了贡献。如果有一天中国拥有 1 亿以上的软件开发工程师，那中国软件行业就真正发展起来了。

那天之后，萧索的心情开始慢慢好转，毕竟自己还在做一件“切实爱国”的事情。笔者心想：爱

国，不是光喊口号的事情，而是需要埋头做事的。

本书有什么特点



不知道是否有人仔细研究过笔者写的书，书中的长句是很少的——因为很多语句自己都会反复地调整，有兴趣的读者可以仔细体会一下这个特点。

《基于 J2EE 的 Ajax 宝典》上市一年半了，其间收到不少读者来信，对书中内容也提了一些自己的见解。此外，该书也一直作为“疯狂 Java 实训营”的讲义，这些对本书的升级起到了很大的帮助。

此外，本书还有如下特点：

1. 通俗易懂，适合自学

该书第一版作为培训教材近 2 年了，在吸收大量学习者的学习体会和心得的基础上，本书重点讲解了学习过程中难以理解和掌握的知识点，降低了学习者的学习难度。

2. 知识丰富，内容全面

正如该书的第一版，书中知识非常全面：XHTML、CSS、JavaScript、DOM、Event 机制、XMLHttpRequest、Prototype 库、jQuery、DWR、AjaxTags 等 Ajax 知识的相关内容，都可在本书中找到详细的讲解。

3. 深入实用，实践性强

本书并不是一本 Ajax 的入门图书，本书将 Ajax 技术融入轻量级 Java EE 开发，深入介绍了 Ajax+Java EE 整合开发的方法和步骤，对实际企业开发具有极好的指导意义。

衷心感谢



本书创作过程中有一些小事情一度影响了笔者的心情，让笔者产生过心灰意懒、意兴阑珊的感觉，使得写作也一度中断。在此，要衷心感谢电子工业出版社的朋友，以及疯狂 Java 联盟的杨恩雄、heyitang、petrelsky5 等广大网友和所有给我鼓励的朋友，是你们的支持让我再度燃起创作热情。

本书写给谁看



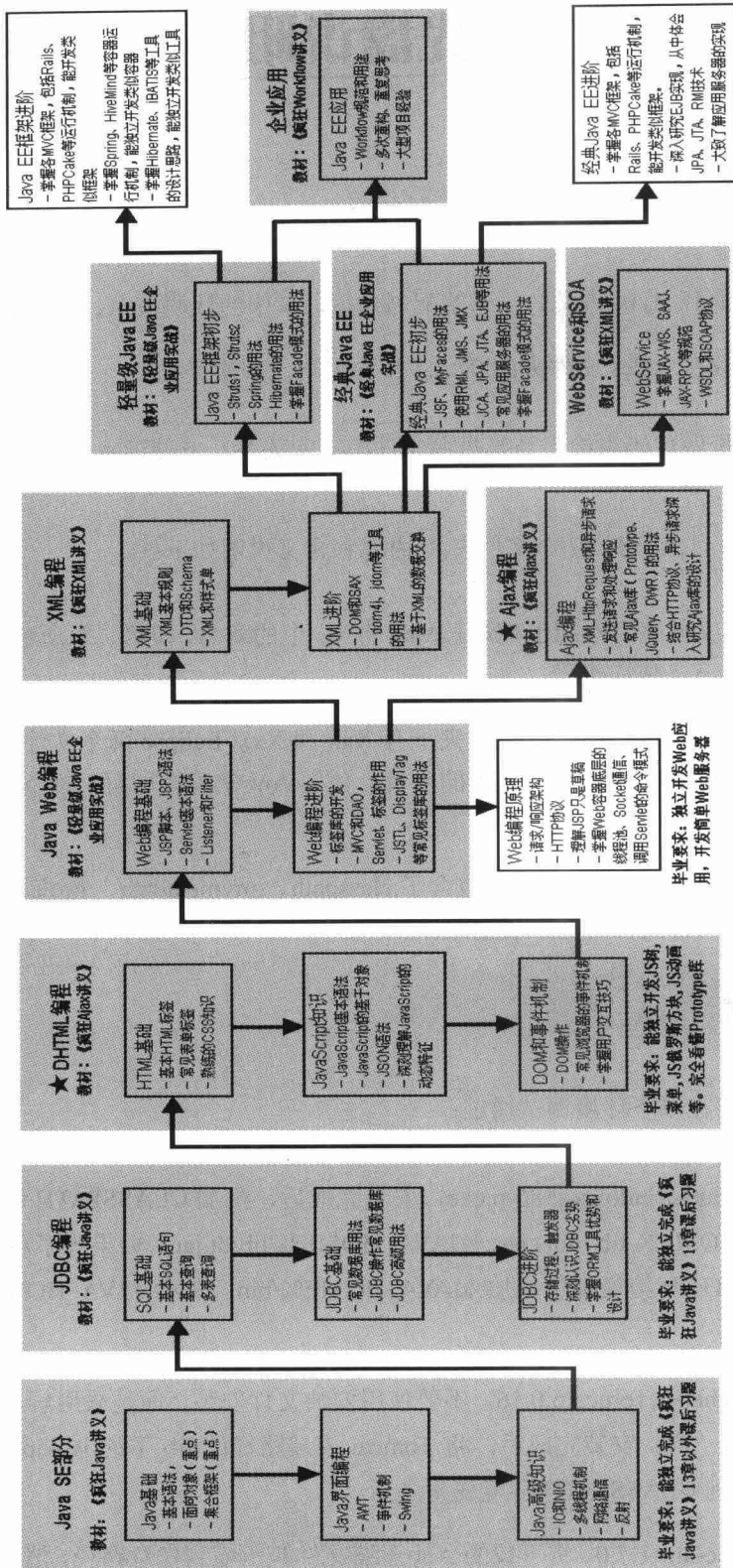
本书是疯狂 Java 体系丛书之一，前半部分（前 8 章）没有基础即可阅读，后半部分则需要一定的 Spring、Hibernate 等 Java EE 基础。如果读者只希望掌握 JavaScript 编程、DHTML 和 Ajax 基础，则无须任何基础；如果读者希望将 Ajax 融入实际的 Java EE 开发，则建议先阅读《轻量级 Java EE 企业应用实战》一书。

2008 年 3 月 17 日



疯狂 Java 学习路线图

笔者就自己对 Java EE 体系的理解，对 Java EE 学习者给出一个粗略线路图：



备注：

1. 没有灰色覆盖的区域稍有难度，请谨慎尝试。
2. 本学习线路图不涉及设计模式、软件方法学等概念，但希望大家能从开发中悟道。
3. 本人并不认为 Spring、Hibernate 很复杂，只要基础扎实，掌握框架是水到渠成的。

光盘说明

一、光盘内容

 本光盘是《疯狂 Ajax 讲义》一书的配书光盘，书中的代码按章、按节存放，即第 2 章、第 2 节所使用的代码放在 codes 文件夹的 02\2.2 文件夹下，依次类推。

 另：书中每份源代码也给出与光盘源文件的对应关系，方便读者查找。

 本光盘 codes 目录下有 17 个文件夹，其内容和含义说明如下：

(1) 01~17 个文件夹名对应于《疯狂 Ajax 讲义》中的章名，即第二章所使用的代码放在 codes 文件夹的 02 文件夹下，依次类推。

(2) 其中 10、12、14、16、17 文件夹下有 Xxx 和 Xxx_Eclipse 两个文件夹，它们是同一个项目的源文件，其中 Xxx 是 IDE 平台无关的项目，使用 Ant 来编译即可；而 Xxx_Eclipse 是该项目在 Eclipse IDE 工具中的项目文件。

(3) codes 文件夹下大量文件夹下包含了 .classpath、.mymetadate、.project、.springBeans 等文件，它们是 Eclipse 项目文件，请不要删除。

二、运行环境

 本书中的程序在以下环境调试通过：

(1) 安装 jdk-6u6-windows-i586-p.exe，安装完成后，添加 CLASSPATH 环境变量，该环境变量的值为;%JAVA_HOME%/lib/tools.jar;%JAVA_HOME%/lib/dt.jar。如果为了可以编译和运行 Java 程序，还应该在 PATH 环境变量中增加%JAVA_HOME%/bin。其中 JAVA_HOME 代表 JDK（不是 JRE）的安装路径。

(2) 安装 Apache 的 Tomcat6.0.16，不要使用安装文件安装，而是采用解压缩的安装方式。安装 Tomcat 请参看第一章。安装完成后，将 Tomcat 安装路径的 lib 下的 jsp-api.jar 和 servlet-api.jar 两个 JAR 文件添加到 CLASSPATH 环境变量之后。

(3) 安装 apache-ant-1.7.0。将下载的 Ant 压缩文件解压缩到任意路径，然后增加 ANT_HOME 的环境变量，让变量的值为 Ant 的解压缩路径。并在 PATH 环境变量中增加%ANT_HOME%/bin 环境变量。

(4) 安装 MySQL5.0 或更高版本, 安装 MySQL 时候选择 GBK 的编码方式。

(5) 安装 Eclipse3.3, 并安装 MyEclipse6.0 的插件。

三、注意事项

(1) 独立应用程序的代码中都包括 build.xml 文件, 在 Dos 或 Shell 下进入 build.xml 文件所在路径, 执行如下命令:

```
ant build -- 编译程序
```

```
ant run --运行程序
```

(2) 对于 Web 应用, 将该应用复制到 %TOMCAT_HOME%/webapps 路径下, 然后进入 build.xml 所在路径, 执行如下命令:

```
ant build -- 编译应用
```

启动 Tomcat 服务器, 使用浏览器即可访问该应用。

(3) 对于 Eclipse 项目文件, 导入 Eclipse 开发工具即可。

(4) 代码中有大量代码需要连接数据库, 读者应修改数据库 URL 以及用户名、密码让这些代码与读者运行环境一致。如果项目下有 SQL 脚本, 导入 SQL 脚本即可, 如果没有 SQL 脚本, 系统将在运行时自动建表, 读者只需创建对应数据库即可。

(5) 在使用本光盘的程序时, 请将程序拷贝到硬盘上, 并去除文件的只读属性。

(6) 本书绝大部分章节后都提供了相应的编程习题, 供开发者巩固所学, 将理论融入实际开发之用。关于这些编程习题的解题思路和参考答案可登录 <http://www.crazyjava.org> 获取。

四、技术支持



如果您使用本光盘中遇到什么问题, 您可以登录如下网站与我们联系:

网站: <http://www.crazyjava.org>

或发 Email 到 kongyeeku@163.com



CONTENTS

第 1 章 Ajax 概述	1	2.2 开始传统的 JSP 聊天室	28
1.1 重新思考 Web 应用	2	2.2.1 实现业务逻辑组件	28
1.1.1 应用系统的发展史	2	2.2.2 实现控制器	31
1.1.2 传统 Web 应用的优势和缺点	4	2.2.3 实现视图	33
1.2 重新设计 Web 应用	5	2.2.4 JSP 聊天室的问题	34
1.2.1 富 Internet 应用	5	2.3 Ajax 聊天室	34
1.2.2 异步发送请求, 避免等待	7	2.3.1 异步发送请求	35
1.2.3 使用 Ajax	7	 使用 Ajax 技术是不是会带来更大的工作量?	37
1.3 Ajax 介绍	7	2.3.2 解决多余刷新的问题	37
1.3.1 Ajax 的工作方式	8	2.3.3 解析服务器响应	39
1.3.2 Ajax 的核心: XMLHttpRequest	8	2.3.4 何时发送请求	40
1.3.3 Ajax 的编程脚本: JavaScript 语言	9	 客户端频繁发送请求, 难道不会加重服务器负担?	41
1.3.4 HTML 页面的 DOM 模型	9	2.3.5 Ajax 聊天室的特点	44
1.3.5 数据交换和显示	10	2.4 Ajax 编程的技术难点	44
1.4 Ajax 的基本特征	10	2.5 传统 Web 应用与 Ajax 应用的对比	45
1.4.1 异步发送请求	10	2.6 本章小结	46
1.4.2 服务器响应是数据, 而不是页面内容	11	第 3 章 XHTML 语言详解	47
1.4.3 浏览器中的是应用, 不是简单视图	11	3.1 XHTML 简介	48
1.5 Ajax 的替代技术	11	3.1.1 HTML 的作用和历史	48
1.5.1 Sun 的 Java Web Start 技术	11	 在保存 HTML 文件时, 到底采用 .htm 扩展名还是采用 .html 扩展名呢?	48
1.5.2 Microsoft 的 ClickOnce 技术	12	 我应该使用 FrontPage 学习 HTML 文档呢? 还是使用 Dreamweaver 好?	49
1.5.3 基于 Flash 的 Flex	12	3.1.2 HTML 4.01 和 XHTML	49
1.6 搭建 Ajax 开发环境	13	 如果我使用 XHTML 编写互联网页, 会不会有浏览器不支持?	50
1.6.1 本书的 Ajax 开发环境	13	3.2 XHTML 的基本语法	50
1.6.2 安装 Tomcat 服务器	13	3.2.1 XHTML 的基本结构和规则	50
1.6.3 配置 Tomcat 的服务端口	15	3.2.2 XHTML 和 DTD	52
1.6.4 进入 Tomcat 控制台	15	3.3 XHTML 的常用标签	54
1.6.5 部署 Web 应用	17	3.3.1 基本标签	54
1.6.6 配置 Tomcat 的数据源	18	3.3.2 文本格式化标签	55
1.6.7 安装 Ant	19	 如果我希望 HTML 页面内的文本更美观, 例如改变它们的颜色、背景等, 那该用什么标签呢?	57
1.6.8 Eclipse 的下载和安装	21	3.3.3 超级链接和锚点	57
1.6.9 在线安装 Eclipse 插件	21	3.3.4 列表相关标签	58
1.6.10 手动安装 Eclipse 插件	22	3.3.5 图像相关标签	59
1.7 调试 JavaScript 脚本	23		
1.8 本章小结	24		
第 2 章 Ajax 初体验	25		
2.1 Ajax 带来的优势	26		
 即使使用 Ajax 技术, 客户端和服务			
器一样有网络通信延迟, 尤其是当网			
络状况不好时, 通信延迟将更严重,			
用户一样感受不到更新延迟吗?	27		

3.3.6 表格相关标签	61	4.5.7 三目运算符	102
3.3.7 框架相关标签	64	4.5.8 逗号运算符	103
3.4 XHTML 的表单标签	65	4.5.9 void 运算符	103
3.4.1 表单标签	66	4.5.10 typeof 和 instanceof 运算符	104
3.4.2 使用 input 元素	67	4.6 语句	104
 前面的页面中包含 5 个单选框, 为何前面 3 个只能选中一个, 后面 2 个只能选中一个, 但一共可以选择 2 个呢?	69	4.6.1 语句块	105
3.4.3 使用 label 定义标签	69	4.6.2 空语句	105
 在表单里直接定义普通文本不可以作为标签吗? 专门使用<label.../>元素定义标签有什么作用?	69	4.6.3 异常抛出语句	105
3.4.4 使用 button 定义按钮	70	4.6.4 异常捕捉语句	106
3.4.5 列表框和下拉菜单	71	4.6.5 with 语句	107
3.4.6 使用 textarea 定义文本域	72	4.7 流程控制	108
3.5 XHTML 头部和元信息	73	4.7.1 分支	108
3.6 本章小结	74	4.7.2 while 循环	110
本章练习	74	4.7.3 do while 循环	111
第 4 章 JavaScript 语法详解	75	4.7.4 for 循环	111
4.1 JavaScript 简介	76	4.7.5 for in 循环	112
4.1.1 运行 JavaScript	77	4.7.6 break 和 continue	113
4.1.2 导入 JavaScript 文件	77	4.8 函数	116
4.2 数据类型和变量	77	4.8.1 函数定义	116
4.2.1 定义变量的方式	78	4.8.2 局部变量和局部函数	117
4.2.2 类型转换	78	4.8.3 匿名函数	118
4.2.3 变量	80	4.8.4 函数和类	120
4.3 基本数据类型	82	4.8.5 函数的实例属性和静态属性	121
4.3.1 数值类型	82	4.8.6 递归函数	123
4.3.2 字符串类型	86	4.9 函数的参数处理	124
4.3.3 布尔类型	89	4.9.1 基本类型和复合类型的参数传递	125
4.3.4 undefined 和 null	90	4.9.2 空参数	126
4.3.5 正则表达式	91	4.9.3 参数类型	127
4.4 复合类型	93	4.10 对象	128
4.4.1 对象	93	4.10.1 面向对象的概念	129
4.4.2 数组	93	4.10.2 对象和关联数组	129
4.4.3 函数	94	4.10.3 继承和 prototype	130
4.5 运算符	96	4.11 建对象	135
4.5.1 赋值运算符	96	4.11.1 使用关键字 new 创建对象	135
4.5.2 算术运算符	97	4.11.2 使用 Object 直接创建对象	135
4.5.3 位运算符	98	4.11.3 使用 JSON 语法创建对象	137
4.5.4 加强的赋值运算符	99	4.12 本章小结	140
4.5.5 比较运算符	100	本章练习	140
4.5.6 逻辑运算符	101	第 5 章 级联样式单详解	141
		5.1 样式单概述	142
		5.2 CSS 的基本使用	143
		5.2.1 引入外部样式文件	143
		5.2.2 使用内部 CSS 样式	144
		5.2.3 使用内联样式	146

5.3 使用 CSS 属性	147	6.5.2 添加节点	185
5.3.1 文字相关属性	148	6.5.3 为列表框、下拉菜单增加选项	185
5.3.2 整体段落相关属性	150	6.5.4 动态添加表格内容	187
5.3.3 背景相关属性	151	6.6 删除 XHTML 元素	188
5.3.4 表格相关属性	152	6.6.1 删除节点	188
5.3.5 大小相关属性	155	6.6.2 删除列表框、下拉菜单的选项	189
5.3.6 位置相关属性	155	6.6.3 删除表格的行或单元格	191
5.3.7 边框相关属性	157	6.7 传统 DHTML 模型	192
5.3.8 轮廓相关属性	159	6.8 使用 window 对象	194
5.3.9 三个常用属性	160	6.8.1 访问历史	196
5.4 选择器定义	161	6.8.2 浏览器对象	197
5.4.1 属性选择器	162	6.8.3 访问页面 URL	197
5.4.2 ID 选择器	163	6.8.4 客户机屏幕信息	198
5.4.3 class 选择器	164	6.8.5 弹出新窗口	199
5.4.4 包含选择器和子元素选择器	165	6.8.6 确认对话框和输入对话框	199
5.4.5 超级链接相关选择器	166	6.8.7 使用定时器	200
5.5 在脚本中修改显示样式	166	6.9 使用 document 对象	201
5.5.1 随机改变页面的背景色	167	6.9.1 动态页面	202
5.5.2 卷帘效果	167	6.9.2 读写 Cookie	203
5.5.3 动态增加立体效果	169	6.10 两个常用范例	204
5.6 本章小结	170	6.10.1 可编辑表格	204
第 6 章 DOM 模型详解	171	6.10.2 导航菜单	206
6.1 DOM 模型概述	172	6.11 DOM 模型和 XML 文档	210
6.2 DOM 模型和 XHTML 文档	173	6.11.1 使用 DOM 解析 XML 文档	210
6.2.1 XHTML 元素之间的继承图	173	6.11.2 使用 DOM 解析器创建 XML	212
6.2.2 XHTML 元素之间常见的包含关系	174	6.12 本章小结	214
6.3 访问 XHTML 元素	175	本章练习	214
6.3.1 根据 ID 访问 XHTML 元素	175	第 7 章 事件处理机制	215
 如何让每个 XHTML 元素都有唯一的 id 属性呢? 以前我见到很多 XHTML 页面元素并没有 id 属性啊。	175	7.1 基本事件模型	216
 程序中为了访问<div.../>元素和<textarea.../>元素的“内容”, 为何一个用 innerHTML 属性, 另一个用 value 属性?	176	7.1.1 绑定 XHTML 元素属性	216
6.3.2 利用节点关系访问 XHTML 元素	176	7.1.2 绑定 DOM 对象的属性	218
6.3.3 访问表单域控件	178	7.1.3 事件处理函数和关键字 this	219
6.3.4 访问列表框、下拉菜单的选项	179	7.1.4 使用返回值改变默认行为	221
6.3.5 访问表格子元素	180	7.1.5 在代码中触发事件	222
6.4 修改 XHTML 元素	182	 为什么在<form.../>元素中<input.../>元素的 id 属性值不能是 submit 呢? ..	223
6.5 新增 XHTML 元素	183	7.2 Ajax 应用的 MVC	224
6.5.1 创建或复制节点	183	7.3 Internet Explorer 的事件模型	226
		7.3.1 使用 script for 绑定	227
		7.3.2 使用 attachEvent 方法执行绑定	227
		7.3.3 访问事件对象	229
		 此处介绍的是 Internet Explorer 中访问事件的方式, 那么其他浏览器呢? ..	232
		7.3.4 事件冒泡	232

7.3.5 重定向事件.....	234	9.1.2 下载 Prototype 库	282
7.3.6 取消事件默认行为.....	236	9.1.3 安装 Prototype 库	283
7.3.7 捕获鼠标事件.....	237	9.1.4 使用 Prototype 对象	283
7.4 DOM 2 的事件模型.....	239	9.2 Prototype 的工具函数	284
7.4.1 绑定事件处理器.....	239	9.2.1 使用 \$() 函数	284
7.4.2 访问事件对象.....	241	9.2.2 使用 \$\$() 函数	286
 DOM 2 事件模型和 Internet Explorer 事件模型里访问事件对象的方式完 全不同, 如果我们需要写一个跨浏 览器的程序, 是不是只能将事件处 理函数绑定到 XHTML 元素, 并将 event 显式作为参数传入事件处理 函数?	241	9.2.3 使用 \$A() 函数	288
7.4.3 事件传播.....	243	9.2.4 使用 \$F() 函数	289
7.4.4 转发事件.....	247	9.2.5 使用 \$H() 函数	290
7.4.5 取消事件的默认行为.....	249	9.2.6 使用 \$R() 函数	291
7.5 本章小结	250	9.2.7 使用 Try.these() 函数	291
本章练习	250	9.3 Prototype 的 JSON 支持	293
第 8 章 XMLHttpRequest 对象详解	251	9.4 Prototype 的自定义对象和类	294
8.1 XMLHttpRequest 对象概述	252	9.4.1 使用 Element 对象	294
8.2 XMLHttpRequest 的方法和属性	252	9.4.2 使用 Element.Methods	298
8.2.1 XMLHttpRequest 的方法	252	9.4.3 使用 Enumerable	298
8.2.2 XMLHttpRequest 的属性	256	9.4.4 使用 ObjectRange	302
8.3 发送请求	258	9.4.5 使用 Form.Element 操作表单控件	302
8.3.1 发送简单请求	259	9.4.6 使用 Form 操作表单	304
8.3.2 发送 GET 请求	261	9.4.7 使用 Hash 对象	305
8.3.3 发送 POST 请求	263	9.4.8 使用 Event	307
8.3.4 发送请求时的编码问题	264	 element() 和 findElement() 的关系到 底是怎么回事呢?	307
8.3.5 发送 XML 请求	268	9.4.9 使用 Template	308
8.4 处理服务器响应	270	9.4.10 使用 Class	309
8.4.1 处理的时机	270	9.4.11 两个常用的监听器	310
8.4.2 使用文本响应	271	9.5 Prototype 常用的扩展	312
8.4.3 使用 XML 响应	271	9.5.1 扩展 Array	312
8.4.4 使用 DOM 模型生成页面	273	9.5.2 扩展 document	313
8.5 XMLHttpRequest 对象的运行 周期	273	9.5.3 扩展 String	314
8.6 Ajax 必须解决的问题	274	9.5.4 扩展 Function	316
8.6.1 跨浏览器问题	274	9.5.5 扩展 Number	318
8.6.2 安全性问题	275	9.6 Prototype 的 Ajax 支持	319
8.6.3 性能问题	277	9.6.1 使用 Ajax.Request 类	319
8.7 本章小结	280	9.6.2 使用 Form.request 方法	322
第 9 章 Prototype 库详解	281	9.6.3 使用 Ajax.Responders 对象	324
9.1 Prototype 的下载和安装	282	9.6.4 使用 Ajax 对象	325
9.1.1 什么是 Prototype 库	282	9.6.5 使用 Ajax.Updater 类	325
		9.6.6 使用 Ajax.PeriodicalUpdater 类	328
		9.7 本章小结	329
		第 10 章 基于 Prototype 库的应用: 自动 完成	330
		10.1 应用的基本分析和设计	331
		10.1.1 数据要求	331

10.1.2 数据表结构.....	331	11.6 动画效果相关的方法.....	380
10.2 Domain Object 和持久层.....	331	11.7 Ajax 相关方法.....	383
10.2.1 Domain Object.....	332	11.7.1 两个工具方法.....	383
10.2.2 实现 DAO 组件.....	333	11.7.2 使用 load 方法.....	384
10.3 实现 Service 组件.....	337	11.7.3 使用 jQuery.ajax(options)方法.....	385
10.4 使用 Servlet 提供服务器响应.....	341	11.7.4 使用 get/post 方法.....	387
10.4.1 根据前缀查询品牌.....	341	11.8 扩展 jQuery 和 jQuery 插件.....	389
10.4.2 根据品牌查询型号.....	342	11.9 本章小结.....	390
10.4.3 根据型号查询详细信息.....	343	第 12 章 基于 jQuery 的应用：电子相册系统.....	391
10.5 客户端 HTML 页面实现.....	344	12.1 实现持久层.....	392
10.6 增加 HTML 页面的事件响应能力.....	346	12.1.1 实现持久化类.....	392
10.6.1 实现品牌输入框的事件处理器.....	346	12.1.2 配置 SessionFactory.....	394
10.6.2 实现键盘事件的处理器.....	347	12.2 实现 DAO 组件.....	395
10.6.3 根据品牌提示型号.....	350	12.2.1 DAO 接口定义.....	395
10.6.4 根据型号显示描述.....	351	12.2.2 完成 DAO 组件的实现类.....	397
10.6.5 注册 Ajax 事件监听器.....	352	12.3 实现业务逻辑层.....	402
10.7 本章小结.....	352	12.3.1 实现业务逻辑组件.....	402
第 11 章 jQuery 库详解.....	353	12.3.2 配置业务逻辑组件.....	405
11.1 jQuery 入门.....	354	12.4 实现客户端调用.....	405
11.1.1 理解 jQuery 的设计.....	354	12.4.1 访问业务逻辑组件.....	406
 上面的程序中 target 对象到底是什么？它怎么会拥有 height、width、css 这些方法？.....	355	12.4.2 处理用户登录.....	406
11.1.2 下载和安装 jQuery.....	355	12.4.3 获得用户相片列表.....	408
11.1.3 让 jQuery 与其他 JavaScript 库共存.....	356	12.4.4 处理翻页.....	409
11.2 获取 jQuery 对象.....	356	12.4.5 处理文件上传.....	411
11.2.1 jQuery 核心函数.....	356	 当 Servlet 重定向到 album.html 页面后，如何弹出如图 12.5 所示对话框？.....	413
11.2.2 以 CSS 选择器访问 DOM 元素.....	357	12.4.6 页面加载时的处理.....	413
11.2.3 选择器的附加限定词.....	359	 HttpSession 里的 curImg 属性是从哪里来的呢？.....	415
11.2.4 表单相关的选择器.....	362	12.5 本章小结.....	415
11.3 jQuery 操作类数组的工具方法.....	363	本章练习.....	415
11.3.1 过滤相关方法.....	364	第 13 章 DWR 框架详解.....	416
11.3.2 仿 DOM 导航的相关方法.....	365	13.1 DWR 的下载和安装.....	417
11.3.3 链接方法.....	367	13.1.1 什么是 DWR.....	417
11.4 jQuery 支持的方法.....	368	13.1.2 下载和安装 DWR.....	418
11.4.1 jQuery 命名空间的方法.....	368	13.2 使用 DWR.....	422
11.4.2 数据存储的相关方法.....	370	13.2.1 编写处理类.....	422
11.4.3 操作属性的相关方法.....	370	13.2.2 配置 DWR.....	424
11.4.4 操作 CSS 属性的相关方法.....	371	13.3 使用 DWR 的转换器.....	425
11.4.5 操作元素内容的相关方法.....	373	13.3.1 基本转换器.....	425
11.4.6 操作 DOM 节点的相关方法.....	374	13.3.2 对象转换器.....	426
11.5 jQuery 事件相关方法.....	378	13.3.3 数组转换器.....	428

13.3.4 集合类型转换器	428
13.4 方法声明定义	429
13.5 使用 DWR 的创建器	430
13.5.1 创建器的配置	430
13.5.2 使用 new 创建器	432
13.5.3 使用 none 创建器	433
 既然 none 创建器不创建任何对象，哪有对象暴露给 JavaScript 代码？	433
13.5.4 使用 script 创建器	433
13.6 调用服务器端的方法	434
13.6.1 调用服务器端方法的通用配置	434
13.6.2 使用简单回调	435
13.6.3 使用 JSON 格式的回调	440
13.6.4 将客户端参数传递到回调函数	442
13.7 使用 engine.js	443
13.7.1 设置调用顺序	443
13.7.2 设置全局超时时长	443
13.7.3 设置全局 Hook 函数	444
13.7.4 设置全局处理函数	444
13.7.5 设置常用的全局选项	444
13.7.6 批处理	445
13.8 使用 util.js	445
13.8.1 使用 \$()	446
13.8.2 处理列表	446
13.8.3 处理表格	450
13.8.4 访问 HTML 元素值	454
 getValues() 可以一次获取多个 HTML 元素的值，那返回的值如何保存呢？	455
13.8.5 几个工具函数	456
13.9 整合第三方 Java EE 框架	458
13.9.1 访问 Servlet API	458
 老师你以前教我们：谁调用方法，谁负责为形参赋值。现在我们调用 addSession() 方法时没有为第二个参数赋值，那第二个参数从哪里获得参数值呢？	460
13.9.2 整合 Spring	461
13.10 使用 DWR 注释	463
13.10.1 初始配置	464
 如果我有很多类需要列出，那岂不是很臃肿，classes 属性值是否支持通配符？如果想列出内部类应该怎么写？	464

13.10.2 标注创建器和转换器	464
13.11 异常处理	465
13.12 反向 Ajax	467
13.12.1 配置使用反向 Ajax	468
 反向 Ajax 技术不是依赖 HTTP 协议的，它怎么可以违反请求-响应架构的规律呢？	468
13.12.2 在 Java 方法中操作 Web 页	469
13.12.3 在客户端调用反向 Ajax 方法	471
13.13 本章小结	472
第 14 章 基于 DWR 的应用：即时消息系统	473
14.1 实现 Hibernate 持久层	474
14.1.1 Hibernate 持久层的 POJO	474
14.1.2 将 POJO 映射成持久化对象	476
14.2 实现 DAO 组件	477
14.2.1 扩展 HibernateDaoSupport 来实现分页	478
 为什么不扩展 Hibernate Template 类来实现分页？扩展 HibernateDaoSupport 是不是会引起一些混乱？	478
14.2.2 实现 DAO 组件	480
14.3 实现业务逻辑组件	482
14.3.1 业务逻辑组件的接口	483
14.3.2 业务逻辑组件的实现类	484
14.3.3 部署业务逻辑组件	487
14.3.4 基于 AOP 的权限控制	488
14.4 调用业务逻辑组件	490
14.4.1 将 Spring 容器中的 Bean 创建成 JavaScript 对象	490
14.4.2 处理用户登录	491
 既然已经在 JavaScript 代码里进行了权限控制，为何还要在业务逻辑层控制呢？	492
14.4.3 处理用户注册	493
14.4.4 处理消息发布	494
14.4.5 获取消息列表	495
14.4.6 处理分页	496
14.4.7 查看消息内容	497
14.4.8 页面加载函数	498
14.5 本章小结	498
本章练习	498
第 15 章 AjaxTags 框架详解	499
15.1 AjaxTags 的下载和安装	500
15.1.1 什么是 AjaxTags	500

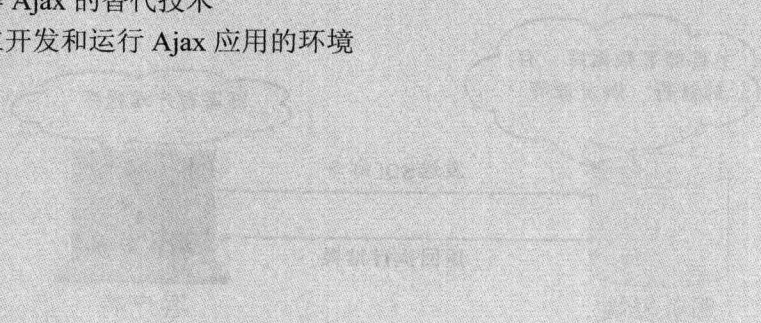
15.1.2 下载和安装 AjaxTags	500	16.5.3 页面加载时的动作	551
15.2 AjaxTags 入门	501	16.5.4 查看评论	552
15.2.1 编写处理类	502	16.5.5 控制回复的翻页	554
15.2.2 使用标签	503	16.5.6 添加回复	554
15.3 处理类的几种形式	505	16.5.7 查看 Blog 文章内容	556
15.3.1 使用普通 Servlet 生成响应	505	16.5.8 添加新的 Blog 文章	556
15.3.2 使用 AjaxXmlBuilder 辅助类	507	16.6 本章小结	558
15.3.3 使用 BaseAjaxServlet 生成响应	509	本章练习	558
15.3.4 使用非 Java 响应	510	第 17 章 电子拍卖系统	559
15.4 使用 AjaxTags 标签	511	17.1 总体说明和概要设计	560
15.4.1 使用自动完成标签	511	17.1.1 系统的总体架构设计	560
15.4.2 使用 area 标签	515	17.1.2 数据库设计	561
15.4.3 使用 anchors 标签	516	17.2 实现 Hibernate 持久化类	562
15.4.4 使用 callout 标签	517	17.2.1 设计 Domain Object	562
15.4.5 使用 htmlContent 标签	518	17.2.2 实现 Domain Object	563
15.4.6 使用 portlet 标签	520	17.3 DAO 层实现	568
15.4.7 使用 select 标签	522	17.3.1 DAO 的基础配置	568
15.4.8 创建 Tab 页	523	17.3.2 实现 DAO 组件	569
15.4.9 使用 displayTag 标签	524	17.3.3 部署 DAO 组件	574
15.4.10 使用 tree 标签创建树	526	17.4 业务逻辑层实现	575
15.4.11 使用 updateField 标签	528	17.4.1 设计业务逻辑组件	575
15.5 关于 AjaxTags 的选择	530	17.4.2 业务逻辑组件的异常处理	576
15.5.1 AjaxTags 的优势和使用场景	530	17.4.3 发送竞价通知邮件	578
15.5.2 AjaxTags 的缺点	531	17.4.4 实现业务逻辑层组件	578
15.6 本章小结	531	17.4.5 业务层的权限控制	587
第 16 章 Ajax 实例: 简易 Blog 系统	532	17.4.6 业务层的任务调度	588
16.1 实现 Hibernate 持久层	533	17.4.7 事务管理	589
16.1.1 设计 Hibernate 的持久化类	533	17.5 暴露业务逻辑方法	590
16.1.2 完成映射文件	535	17.5.1 初始化 Spring 容器	590
16.1.3 数据表的结构	537	17.5.2 配置 DWR 的核心 Servlet	590
16.2 实现 DAO 组件	538	17.5.3 暴露业务逻辑方法	591
16.2.1 DAO 接口定义	538	17.6 调用业务逻辑方法响应用户	
16.2.2 实现 DAO 组件	539	请求	592
16.2.3 配置 DAO 组件	542	17.6.1 页面加载时的函数	592
16.3 实现业务逻辑组件	543	17.6.2 处理返回首页的请求	593
16.3.1 业务逻辑组件的接口	543	17.6.3 浏览所有流拍物品	593
16.3.2 业务逻辑组件的实现类	544	17.6.4 处理用户登录	596
16.3.3 配置业务逻辑组件	547	17.6.5 管理物品	600
16.4 整合 DWR 框架	548	17.6.6 管理物品种类	603
16.4.1 配置 web.xml 文件	548	17.6.7 查看竞得物品	606
16.4.2 将 Spring 容器中的 Bean 转化成		17.6.8 查看自己的竞价记录	608
JavaScript 对象	549	17.6.9 浏览拍卖物品	609
16.5 在客户端调用 JavaScript 对象	550	17.6.10 参与竞价	611
16.5.1 获取 Blog 文章列表	550	17.7 本章小结	614
16.5.2 控制 Blog 文章列表的翻页	551	本章练习	614

第 1 章

Ajax 概述

本章要点

- ✎ C/S 模式应用的结构和缺点
- ✎ B/S 模式应用的结构和优势
- ✎ 传统 Web 应用的不足
- ✎ 如何改进传统的 Web 应用
- ✎ RIA 的改进和优势
- ✎ Ajax 的基础
- ✎ Ajax 的基本特征
- ✎ Ajax 依赖的核心技术
- ✎ 了解 Ajax 的替代技术
- ✎ 建立开发和运行 Ajax 应用的环境



Ajax (Asynchronous JavaScript And XML, 异步 JavaScript 和 XML) 是个相当新的名词, 它在 2005 年由 Jesse James Garrett 首先提出。在接下来的极短时间内, Ajax 被广泛应用到大量 B/S 结构的应用中, 改进了传统的 Web 应用, 给浏览者一种更连续的体验。Ajax 的最大优势在于异步交互, 即浏览者在浏览页面时, 可同时向服务器发送请求, 甚至可以不用等待前一次请求得到完全响应, 便再次发送请求。这种异步请求的方式, 非常类似于传统的桌面应用。通过使用 Ajax 技术, 可以使互联网网页具有更友好的人机交互和更美观的浏览界面。

使用 Ajax 的异步请求方式, 浏览器无须频繁地重新加载新页面, 服务器的响应不再是整个页面内容, 而只是必须更新的部分数据。Ajax 可以减轻服务器和带宽的负担, 提供更好的服务响应。使用 Ajax 的异步模式, 浏览器无须重新加载整个页面, 就可以显示新的数据。浏览器通过 JavaScript 代码向服务器发送请求, JavaScript 代码负责解析服务器的响应数据, 并把样式表加到数据上, 然后在现有网页中显示出来。

Ajax 技术给互联网带来了一场革命——Web 2.0, 而且它也正是这场革命中的核心技术。到目前为止, 已很难找到一个没有使用 Ajax 技术的 Web 应用。Ajax 技术甚至催生了一种新的网络游戏平台: 网页游戏——游戏玩家无须下载任何客户端, 直接打开网页就可开始游戏。

1.1 重新思考 Web 应用

传统的 Web 应用经过多年的发展, 在很多方面都是相当完善的。特别是 Java EE、.NET、Ruby on Rails 等平台的出现, 更是规范了 Web 应用的开发。Ajax 的出现, 让人不得不重新思考传统的 Web 应用。Ajax 给浏览者一种全新的体验: 浏览者可以无须等待服务器响应, 而多次以异步方式向服务器发送请求。这种体验方式, 非常类似于传统的桌面应用。Ajax 并不是要颠覆传统的 B/S 结构的应用, 而只是让 B/S 结构的应用更加完善。

1.1.1 应用系统的发展史

早期应用软件系统大都采用 C/S (客户机/服务器模式) 结构, C/S 结构的软件分为客户机和服务器两层。客户机不是毫无运算能力的输入/输出设备, 在客户端需要部署大量的应用程序, 而且可能还具有一定的数据存储能力。

C/S 结构应用的服务器端通常主要安装数据库管理系统, 当然也可能包含一些业务逻辑实现 (这些业务逻辑实现通常以函数、存储过程和触发器的形式存在)。通过把软件系统的计算和数据合理地分配在客户机和服务器两端, 可以有效地降低网络通信量和服务器运算量。

C/S 结构应用的结构图如图 1.1 所示。

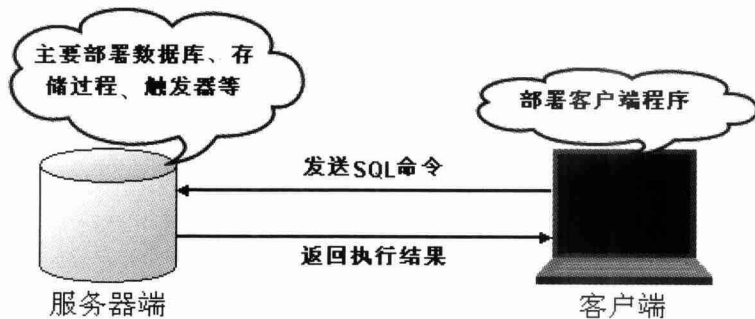


图 1.1 C/S 结构应用

对于 C/S 结构的应用而言, 因为可以直接在客户端部署应用程序, 所以可以让应用的人机交互界面更加友好, 并可充分美化应用程序的人机界面。但由于服务器连接个数和数据通信量的限制, 这种结构的软件适于在用户数目不多的局域网内使用。早期的大部分 ERP 软件产品即属于此类结构。