

疯狂

Ajax讲义

(第3版)

jQuery/Ext JS/
Prototype/DWR
企业应用前端开发实战

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

第3版

jQuery/Ext JS/
Prototype/DWR
企业应用前端开发实战

李刚 编著

电子工业出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书为《疯狂 Ajax 讲义》的第 3 版,其中 jQuery 升级到 1.8 版本;Prototype 升级到 1.7.1 版本;DWR 升级到 3.0 版本。第 3 版最大更新是详细、全面地介绍了 Ext JS 4.1 的知识。由于 Ext JS 4.1 本身内容比较多,因此本书花了几百页来介绍 Ext JS 的功能和用法,这部分内容独立出来即可作为 Ext JS 4.1 的学习手册。

本书详细介绍了 jQuery 1.8、Ext JS 4.1、Prototype 1.7.1、DWR 这 4 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,让读者理论联系实际。这部分内容是“疯狂软件教育中心”的标准讲义,它既包含了实际 Ajax 开发的重点和难点,也融入了大量学习者的学习经验和感悟。再由笔者以丰富的授课经验为基础,深入浅出地介绍它们,希望读者真正掌握 Ajax 开发的精髓。本书最后提供了 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。电子拍卖系统是一个包含 5 个表、表之间具有复杂关联映射的系统,该案例采用目前最流行、最规范的轻量级 Java EE 架构,将整个应用分为领域对象层、DAO 层、业务逻辑层,然后用 DWR 作为 Ajax 引擎调用业务逻辑方法响应异步请求,系统权限检查推迟到业务逻辑层完成,并使用 HTTP Session 跟踪用户操作状态,从而避免用户刷新浏览器导致的浏览状态丢失。该案例对实际项目具有极好的指导价值和借鉴意义。该案例既提供了 IDE 无关的、基于 Ant 管理的项目源码,也提供了基于 Eclipse IDE 的项目源码,最大限度地满足读者的需求。如果在阅读本书时遇到任何技术问题,都可登录 <http://www.crazyit.org> 与本书庞大的读者群交流。

本书并非针对零基础读者,书中不再包含 HTML、CSS、JavaScript 相关知识,这些知识是阅读本书的基础。本书适合有初步 HTML、CSS、JavaScript 基础的读者,或对企业应用前端开发不太熟悉的开发人员。如果你已经掌握本书上篇:《疯狂 HTML 5/CSS 3/JavaScript 讲义》内容,将非常适合阅读此书。

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: jQuery/Ext JS/Prototype/DWR 企业应用前端开发实战 / 李刚编著.—3 版. —北京: 电子工业出版社, 2013.2
ISBN 978-7-121-19394-1

I. ①疯… II. ①李… III. ①网页制作工具 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2012)第 318332 号

策划编辑: 张月萍

责任编辑: 高洪霞

印 刷: 北京天宇星印刷厂

装 订: 三河市鹏成印业有限公司

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

开 本: 787×1092 1/16

印张: 39.75

字数: 997 千字

彩插: 1

印 次: 2013 年 2 月第 1 次印刷

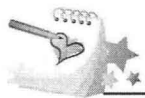
印 数: 4000 册

定价: 79.00 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 zlts@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

无论采用哪种开发平台，只要开发 B/S 结构的应用，Ajax 都是一门不容回避的技术。从某种角度来看，Ajax 比 Java 的应用更广泛，后台语言除了首选 Java 之外，还可以选择 PHP 或 C#……，但 Ajax 技术不容回避。

Ajax 技术是 Web 2.0 的重要技术之一，互联网上各种 Blog 系统、RSS，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。国内大量的游戏运营商纷纷推出自己的网页游戏，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，因此具有很好的市场前景。在这种网页游戏中，每个网页上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。

由于 Ajax 的应用如此广泛，因此 Ajax 相关框架也十分活跃：除了最主流的 jQuery、Ext JS 4.1、Prototype、DWR 之外，还有 Dojo、YUI 等各种框架，本书介绍 jQuery、Ext JS 4.1、Prototype、DWR 这 4 个框架，因为它们不仅是企业开发中最主流的前端框架，而且具有很强的代表性。

- jQuery/Prototype: 代表单纯的 JavaScript 库，主要用于简化 JavaScript 开发，尤其是 jQuery 基本上已经成为行业规范。
- Ext JS: 代表专业的 JavaScript 库+UI 组件库的 RIA 框架。不仅可简化 JavaScript 开发，而且可以轻松地开发专业、强大的 UI 界面。
- DWR: 代表 Java 领域的 RPC 风格的 Ajax 框架。对于 Java EE 应用，DWR 是十分明智的选择。但它不适合其他编程语言，因此影响力不如 jQuery、Ext JS。

由于 Ajax 技术本身丰富多彩，单独掌握一个 Ajax 框架（比如 jQuery），并不能充分满足企业前端开发的要求。企业前端开发不仅需要扎实的 JavaScript 编程基础，还需要掌握 HTML 5、CSS 3 等 UI 技术，但如果总靠原生的 JavaScript、HTML 5、CSS 3 技术进行前端开发，不仅开发效率低，而且应用界面的一致性也不好。因此企业前端开发既需要掌握主流的 JavaScript 库（jQuery 基本已成为行业规范），还需要掌握主流的 RIA 框架，如 Ext JS。

本书全面、细致地介绍了 jQuery、Ext JS 4.1、Prototype、DWR，将本书的任何部分独立出来，都可作为这门技术的学习手册；整体来看，它们是企业前端开发框架最主流、最具代表性的技术。因此，一个功底扎实的前端开发者必须熟练地掌握并应用它们。

衷心感谢



疯狂 Java 体系图书已走过多年的“风风雨雨”，收到过无数读者的感激、期待，也遭遇过某些嫉妒者的恶意诋毁……但“群众的眼睛是雪亮的”，广大读者的认同已让疯狂 Java 体系图书的销量稳居国内第一。以 Ajax 为例（搜索 Java、Android、XML 等都会看到类似效果），在当当网搜索 Ajax 图书，可以看到《疯狂 Ajax 讲义》排在第 1 位，截图如下。



在京东网搜索 Ajax 图书、按销量排行，可以看到《疯狂 Ajax 讲义》排在第 2 位，截图如下。



诚挚地感谢广大读者的支持与爱护，你们的支持让疯狂 Java 图书没有放弃，你们的激励让疯狂 Java 图书茁壮成长，你们的反馈让疯狂 Java 图书日臻完善！同时也感谢博文视点张月萍等编辑，以及疯狂软件教育中心技术团队一贯的支持。

本书有什么特点



本书只是一本介绍 Ajax 前端开发的图书，不是一本关于所谓“思想”的书，更不是一本看完之后可以“吹嘘、炫耀”的书——因为本书并没有堆砌一堆“深奥”的新名词、一堆“高深”的思想，本书保持了“疯狂 Java 体系”的一贯风格：操作步骤详细，编程思路清晰，语言平实。

本书带给读者的只是 9 个字：“看得懂、学得会、做得出”，本书并不能让你认识一堆新名词，只能帮助你掌握扎实的企业前端开发功底。对于本书，光“看”是不够的，一定要“做”，阅读本书的同时，应该把所有知识点的配套实例都做出来，这样才能真正掌握本书的知识。

无论如何，读者在阅读本书时遇到知识上的问题，都可以登录疯狂 Java 联盟 (<http://www.crazyit.org>) 与广大 Java 学习者交流，笔者也会通过该平台与大家一起交流、学习。

除此之外，本书还有如下几个特点。

1. 通俗易懂，适合自学

本书作为培训教材 5 年多了，吸收了大量学习者的学习体会和心得，并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

本书全面、详细地介绍了 jQuery、Ext JS 4.1、Prototype、DWR，它们既是企业开发中最主流的前端框架， also 具有很强的代表性。掌握本书内容即可具备扎实的前端开发功底。

3. 深入实用，实践性强

本书并不是一本 Ajax 的入门图书，而是全面、深入地介绍了企业开发中最主流、最具代表性的前端框架，并将它们真正融入 Java 企业应用开发，这对实际企业应用开发具有极好的指导意义。

本书写给谁看



本书并非针对零基础的读者，如果你具有 HTML 5、CSS 3、JavaScript 基础，认真学习此书即可让你成为前端开发的实战型人才；如果你已经阅读过本书上篇：《疯狂 HTML 5/CSS/JavaScript 讲义》，那么阅读本书非常合适；如果你完全没有 HTML、CSS、JavaScript 基础知识，建议暂时不要购买、阅读本书。

2012-10-28

目 录 CONTENTS

第 1 章 Ajax 概述与 Ajax 初体验	1	2.2 XMLHttpRequest 的属性	38
1.1 重新思考 Web 应用	2	2.3 发送请求	41
1.1.1 应用系统的发展史	2	2.3.1 发送简单请求	41
1.1.2 传统 Web 应用的优势和缺点	5	2.3.2 发送 GET 请求	43
1.2 重新设计 Web 应用	5	2.3.3 发送 POST 请求	46
1.2.1 富 Internet 应用	6	2.3.4 发送请求时的编码问题	47
1.2.2 使用 Ajax	7	2.3.5 发送 XML 请求	51
1.3 Ajax 技术介绍	7	2.4 处理服务器响应	54
1.3.1 Ajax 的工作方式	8	2.4.1 处理的时机	54
1.3.2 Ajax 技术核心： XMLHttpRequest	8	2.4.2 使用文本响应	54
1.3.3 Ajax 的编程脚本： JavaScript 语言	8	2.4.3 使用 JSON 响应	54
1.3.4 HTML 页面的 DOM 模型	9	2.5 XMLHttpRequest 对象的运行周期	57
1.3.5 动态加载和显示数据	9	2.6 Ajax 必须解决的问题	58
1.3.6 Ajax 的特征	10	2.6.1 跨浏览器问题	58
1.3.7 Ajax 带来的典型优势	11	2.6.2 安全性问题	59
1.4 开发传统的 B/S 聊天室	13	2.6.3 性能问题	61
1.4.1 实现业务逻辑组件	14	2.7 本章小结	65
1.4.2 实现控制器	16		
1.4.3 实现视图	18	第 3 章 jQuery 库详解	66
1.4.4 传统 B/S 聊天室的问题	19	3.1 jQuery 入门	67
1.5 Ajax 聊天室	20	3.1.1 理解 jQuery 的设计	67
1.5.1 异步发送请求	20	3.1.2 下载和安装 jQuery	68
1.5.2 解决多余刷新的问题	22	3.1.3 让 jQuery 与其他 JavaScript 库共存	69
1.5.3 解析服务器响应	24	3.2 获取 jQuery 对象	69
1.5.4 何时发送请求	25	3.2.1 jQuery 核心函数	70
1.5.5 Ajax 聊天室的特点	29	3.2.2 jQuery 与 jQuery.holdReady	71
1.6 Ajax 编程的技术难点	29	3.2.3 以 CSS 选择器访问 DOM 元素	72
1.7 传统 Web 应用与 Ajax 应用的对比	31	3.2.4 选择器的附加限定词	74
1.8 本章小结	32	3.2.5 表单相关的选择器	77
第 2 章 XMLHttpRequest 对象详解	33	3.3 jQuery 操作类数组的工具方法	80
2.1 XMLHttpRequest 对象概述	34	3.3.1 过滤相关方法	81
2.2 XMLHttpRequest 的方法和属性	34	3.3.2 仿 DOM 导航查找的相关方法	83
2.2.1 XMLHttpRequest 的方法	35	3.3.3 链接方法	85
2.2.2 XMLHttpRequest 的属性	38	3.4 jQuery 支持的方法	86
		3.4.1 jQuery 命名空间的方法	86
		3.4.2 数据存储的相关方法	90

3.4.3 操作属性的相关方法	91	4.4.2 处理用户登录	145
3.4.4 操作 CSS 属性的相关方法	93	4.4.3 获得用户相片列表	148
3.4.5 操作元素内容的相关方法	96	4.4.4 处理翻页	149
3.4.6 操作 DOM 节点的相关方法	98	4.4.5 通过隐藏 iframe 实现文件上传	150
3.5 jQuery 事件相关方法	103	4.4.6 页面加载时的处理	154
3.5.1 绑定事件处理函数	103	4.5 本章小结	155
3.5.2 特定事件相关的方法	105		
3.5.3 事件对象	106		
3.6 动画效果相关的方法	108	第 5 章 Ext JS 核心技术详解	157
3.6.1 简单动画和复杂动画	108		
3.6.2 操作动画队列	111	5.1 Ext JS 入门	158
3.7 jQuery 的回调支持	112	5.1.1 下载和安装 Ext JS	158
3.7.1 回调支持的基本用法	112	5.1.2 Ext 入门示例	159
3.7.2 创建 Callbacks 支持的选项	114	5.1.3 使用 Ext JS 文档	160
3.8 Ajax 相关方法	116	5.2 核心工具方法	162
3.8.1 三个工具方法	116	5.2.1 Ext 的属性和获取浏览器特性	162
3.8.2 使用 load 方法	117	5.2.2 onReady 方法	163
3.8.3 jQuery.ajax(options)与 jQuery.ajaxSetup(options)	119	5.2.3 Ext.defer 方法	163
3.8.4 使用 get/post 方法	120	5.2.4 Ext.apply 和 Ext.applyIf 方法	164
3.9 jQuery 的 Deferred 对象	123	5.2.5 Ext.define、Ext.create 和 Ext.extend 方法	166
3.9.1 jQuery 的异步调用	123	5.2.6 Ext.isXxx 方法	169
3.9.2 为多个耗时操作指定回调函数	127	5.2.7 Ext.each 和 Ext.iterate 方法	169
3.9.3 为普通对象增加 Deferred 接口	127	5.2.8 Ext.namespace 方法	170
3.9.4 jQuery 对象的 promise 方法	128	5.3 Ext JS 扩展的工具方法	171
3.10 扩展 jQuery 和 jQuery 插件	129	5.3.1 为 Array 扩展的工具方法	171
3.11 本章小结	130	5.3.2 为 Date 扩展的工具方法	173
		5.3.3 为 Function 扩展的工具方法	173
第 4 章 基于 jQuery 的应用：电子相册系统	131	5.3.4 为 Number 扩展的工具方法	174
4.1 实现持久层	132	5.3.5 为 Object 扩展的工具方法	175
4.1.1 实现持久化类	132	5.3.6 为 String 扩展的工具方法	175
4.1.2 配置 SessionFactory	134	5.3.7 JSON 支持	176
4.2 实现 DAO 组件	135	5.4 简化的 DOM 操作	177
4.2.1 开发通用 DAO 组件	136	5.4.1 Ext.get、Ext.getXxx 和 Ext.fly 方法	177
4.2.2 DAO 接口定义	139	5.4.2 Ext.query、Ext.select 方法与 Ext.dom.Query 工具类	178
4.2.3 完成 DAO 组件的实现类	139	5.4.3 DOM 元素的常用操作	182
4.3 实现业务逻辑层	141	5.4.4 使用 Ext.DomHelper、Ext.Template 动态生成 HTML	188
4.3.1 实现业务逻辑组件	141	5.5 动画处理	192
4.3.2 配置业务逻辑组件	143	5.5.1 使用 Ext.fx.Animation 实现简单动画	192
4.4 实现客户端调用	144		
4.4.1 访问业务逻辑组件	144		

5.5.2	使用 Ext.fx.Animator 实现多 关键帧动画	194	6.1.4	使用 Ext.toolbar.ToolBar 添加 工具条	226
5.5.3	为元素增加动画	195	6.1.5	使用 Ext.tab.Panel 创建 Tab 面板	229
5.6	Ext JS 的事件处理	197	6.1.6	使用 Ext.container.Viewport 整体布局	231
5.6.1	使用 Ext.EventManager 管理事 件处理函数与 Ext.EventObject 事件对象	197	6.1.7	使用 Ext.window.Window 创建 窗口	234
5.6.2	事件处理函数的执行范围和附 加选项	200	6.1.8	使用 Ext.window.MessageBox 创建对话框	235
5.6.3	使用 Ext.util.Observable 实现内 部事件	201	6.2	布局	238
5.7	实用工具集	203	6.2.1	容器与布局的关系	238
5.7.1	Ext.util.HashMap 与 Ext.util.MixedCollection	203	6.2.2	垂直布局与水平布局	238
5.7.2	使用 Ext.util.Cookies 操作 Cookie	204	6.2.3	使用 Ext.resizer.Resizer 为组件 增加大小调整功能	239
5.7.3	使用 Ext.util.DelayedTask 处理延时任务	205	6.2.4	使用 Ext.resizer.Splitter 为盒布 局增加大小调整的功能	241
5.7.4	包装单击事件: Ext.util.ClickRepeater	205	6.2.5	使用 Ext.layout.container.Accordion 实现折叠布局	242
5.7.5	使用 Ext.util.TaskRunner 与 Ext.TaskManager	206	6.2.6	使用 Ext.layout.container.Border 实现边框布局	244
5.7.6	使用 Ext.util.KeyMap 与 Ext.util. KeyNav 为元素处理按键事件	207	6.2.7	使用 Ext.layout.container.Table 实现表格布局	245
5.8	拖放	209	6.2.8	使用 Ext.layout.container.Column 实现列布局	246
5.8.1	使用 Ext.dd.DD 和 Ext.dd. DDProxy 创建可拖放元素	209	6.2.9	使用 Ext.layout.container.Auto 实现自动布局	248
5.8.2	用 Ext.dd.DragSource 创建拖动 源与用 Ext.dd.DropTarget 创建 放目标	210	6.2.10	使用 Ext.layout.container.Fit 实现自适应布局	249
5.8.3	用 Ext.dd.DragZone 创建拖动源与 用 Ext.dd.DropZone 创建放目标	212	6.2.11	使用 Ext.layout.container.Card 实现卡片布局	250
5.9	Ajax 支持	214	6.2.12	使用 Ext.layout.container.Anchor 实现锚布局	252
5.10	本章小结	217	6.2.13	使用 Ext.layout.container.Absolute 实现绝对布局	253
第 6 章	Ext JS 界面组件详解	218	6.3	菜单及菜单项	254
6.1	容器、面板和窗口	219	6.3.1	使用菜单	254
6.1.1	使用 Ext.Component	220	6.3.2	使用 Ext.menu.CheckItem 实现 可勾选菜单项	256
6.1.2	使用 Ext.container.Container 创建容器	222	6.3.3	使用 Ext.menu.ColorPicker 创建 颜色选择器菜单	257
6.1.3	使用 Ext.panel.Panel 创建面板	223	6.3.4	使用 Ext.menu.DatePicker 创建 日期选择菜单	258

6.4 按钮	259	6.6.12 使用 Ext.form.field.HtmlEditor 创建富文本编辑器	310
6.4.1 使用按钮	259	6.6.13 使用 Ext.form.field.File 上传文件	311
6.4.2 带菜单的按钮	261	6.6.14 使用 Ext.form.field.Checkbox 创建复选框	314
6.4.3 带分割线的按钮: Ext.button.Split	263	6.6.15 使用 Ext.form.field.Radio 创建 单选按钮	317
6.4.4 多状态按钮: Ext.button.Cycle	264	6.6.16 使用 Ext.form.field.ComboBox 创建复合框	319
6.4.5 使用 Ext.container.ButtonGroup 创建按钮组	266	6.6.17 使用远程数据、queryMode 为 local 的复合框	321
6.4.6 使用 Ext.Action	267	6.6.18 queryMode 为 remote 的复合框	323
6.5 其他组件及实用功能	270	6.6.19 为 Ext.form.field.ComboBox 增加分页功能	325
6.5.1 使用 Ext.ProgressBar 创建 进度条	270	6.6.20 实现 Ext.form.field.ComboBox 二级联动	327
6.5.2 使用 Ext.slider.Single 创建滑动条	274	6.6.21 使用 Ext.form.FieldSet 组合表 单控件	330
6.5.3 使用 Ext.slider.Multi 创建多 滑块滑动条	276	6.7 Ext JS 的数据中心	332
6.5.4 使用 Ext.Image 创建图片	277	6.7.1 Ext.data.Model 与 Ext.data.Field	332
6.5.5 使用 Ext.Editor 创建编辑器	278	6.7.2 使用 Ext.data.proxy.Proxy 和 Ext.data.reader.Reader 读取数据	335
6.5.6 使用 Ext.tip.ToolTip 创建提示	279	6.7.3 Ext.data.Store 及其子类	336
6.5.7 使用 Ext.tip.QuickTipManager 和 Ext.tip.QuickTip 创建提示	283	6.8 使用 Ext.grid.Panel 生成表格	337
6.5.8 使用 Ext.ux.GMapPanel 实现 Google 地图	284	6.8.1 使用 Ext.grid.Panel 生成表格	337
6.6 表单	286	6.8.2 使用 Ext.grid.column.Column 定义列	339
6.6.1 使用 Ext.form.Panel 与 Ext.form.Basic 创建表单	288	6.8.3 使用 Ext.grid.Panel 显示远程数据	342
6.6.2 配置输入校验选项	292	6.8.4 对 Ext.grid.Panel 表格进行分页	343
6.6.3 使用 vtype 选项配置输入校验	294	6.8.5 对本地数据进行分页	344
6.6.4 使用 Ext.form.action.Submit 提 交表单	297	6.8.6 表格数据排序和过滤	346
6.6.5 使用 Ext.form.action.Load 为表 单装载数据	300	6.8.7 通过表格特性对表格数据进行 分组	347
6.6.6 使用 Ext.form.action. StandardSubmit 提交表单	303	6.8.8 开发动态编辑的表格	349
6.6.7 单行文本框、密码框	303	6.8.9 允许拖动数据的表格	352
6.6.8 使用 Ext.form.field.Number 创建数值框	305	6.8.10 使用 Ext.grid.column.Action 操作表格数据	356
6.6.9 使用 Ext.form.field.Date 创建 日期选择框	306	6.9 使用 Ext.tree.Panel 生成树	360
6.6.10 使用 Ext.form.field.Spinner 创建 微调选择框	308	6.9.1 树有关的类及简单树	360
6.6.11 使用 Ext.form.field.TextArea 创 建文本域	309	6.9.2 添加、删除、修改树节点	362
		6.9.3 带复选框的树节点	365

6.9.4 表格树	366	7.6.1 使用 Ajax.Request 类	416
6.9.5 异步加载树节点	368	7.6.2 利用 Form.request 方法	420
6.9.6 树节点的拖放与通过拖放更新 底层数据	372	7.6.3 使用 Ajax.Responders 对象	421
6.10 本章小结	376	7.6.4 使用 Ajax 对象	422
第 7 章 Prototype 库详解	377	7.6.5 使用 Ajax.Updater 类	423
7.1 Prototype 的下载和安装	378	7.6.6 使用 Ajax.PeriodicalUpdater 类	426
7.1.1 什么是 Prototype 库	378	7.7 本章小结	428
7.1.2 下载和安装 Prototype 库	379	第 8 章 基于 Prototype 库的应用: 自动完成	429
7.1.3 使用 Prototype 对象	379	8.1 应用的基本分析和持久层设计	430
7.2 Prototype 的工具函数	380	8.1.1 数据要求	430
7.2.1 使用 \$() 函数	380	8.1.2 实现持久化类	430
7.2.2 使用 \$\$() 函数	382	8.1.3 实现 DAO 组件	432
7.2.3 使用 \$A() 函数	384	8.1.4 部署 DAO 组件	434
7.2.4 使用 \$F() 函数	385	8.2 实现 Service 组件	435
7.2.5 使用 \$w() 函数	386	8.3 使用 Servlet 提供服务器响应	437
7.2.6 使用 \$H() 函数	386	8.3.1 根据前缀查询品牌	438
7.2.7 使用 \$R() 函数	387	8.3.2 根据品牌查询型号	439
7.2.8 使用 Try.these() 函数	388	8.3.3 根据型号查询详细信息	439
7.3 Prototype 的 JSON 支持	389	8.4 客户端 HTML 页面实现	440
7.4 Prototype 的自定义对象和类	390	8.5 增加 HTML 页面的事件响应能力	442
7.4.1 使用 Element 对象	390	8.5.1 实现品牌输入框的事件处理器	442
7.4.2 使用 Element.Methods	394	8.5.2 实现键盘事件的处理器	444
7.4.3 使用 Enumerable	394	8.5.3 根据品牌提示型号	446
7.4.4 使用 ObjectRange	398	8.5.4 根据型号显示描述	447
7.4.5 使用 Form.Element 操作表单 控件	399	8.5.5 注册 Ajax 事件监听器	448
7.4.6 使用 Form 操作表单	400	8.6 本章小结	449
7.4.7 使用 Hash 对象	402	第 9 章 DWR 框架详解	451
7.4.8 使用 Event	404	9.1 DWR 的下载和安装	452
7.4.9 使用 Template	405	9.1.1 什么是 DWR	452
7.4.10 使用 Class	406	9.1.2 下载和安装 DWR	453
7.4.11 两个常用的监听器	407	9.2 使用 DWR	457
7.5 Prototype 常用的扩展	409	9.2.1 编写处理类	458
7.5.1 扩展 Array	409	9.2.2 配置 DWR	460
7.5.2 扩展 document	410	9.3 使用 DWR 的转换器	461
7.5.3 扩展 String	411	9.3.1 基本转换器	461
7.5.4 扩展 Function	414	9.3.2 对象转换器	462
7.5.5 扩展 Number	415	9.3.3 数组转换器	464
7.6 Prototype 的 Ajax 支持	416	9.3.4 集合类型转换器	464

9.4	方法声明定义	465
9.5	使用 DWR 的创建器	466
9.5.1	创建器的配置	466
9.5.2	使用 new 创建器	468
9.5.3	使用 none 创建器	469
9.5.4	使用 script 创建器	469
9.6	调用服务器端的方法	470
9.6.1	调用服务器端方法的通用配置	471
9.6.2	使用简单回调	472
9.6.3	使用 JSON 格式的回调	476
9.6.4	将客户端参数传递到回调函数	479
9.7	使用 engine.js	480
9.7.1	设置调用顺序	480
9.7.2	设置全局超时时长	481
9.7.3	设置全局 Hook 函数	481
9.7.4	设置全局处理函数	481
9.7.5	设置常用的全局选项	482
9.7.6	批处理	482
9.8	使用 util.js	483
9.8.1	使用 \$()	483
9.8.2	处理列表	483
9.8.3	处理表格	488
9.8.4	访问 HTML 元素值	492
9.8.5	几个工具函数	494
9.9	整合第三方 Java EE 框架	496
9.9.1	访问 Servlet API	496
9.9.2	整合 Spring	500
9.10	使用 DWR 注释	502
9.10.1	初始配置	502
9.10.2	标注创建器和转换器	503
9.11	异常处理	504
9.12	反向 Ajax	506
9.12.1	配置反向 Ajax	506
9.12.2	在 Java 方法中操作 Web 页	507
9.12.3	在客户端调用反向 Ajax 方法	509
9.13	DWR 3 简化的文件上传	510
9.13.1	文件上传的处理类	510
9.13.2	调用 JavaScript 完成文件上传	512
9.14	本章小结	514

第 10 章	基于 DWR 的应用：即时消息系统	515
10.1	实现 Hibernate 持久层	516
10.1.1	实现持久化类	516
10.1.2	将 POJO 映射成持久化对象	518
10.2	实现 DAO 组件	519
10.2.1	实现 DAO 组件	519
10.2.2	部署 DAO 组件	520
10.3	实现业务逻辑组件	521
10.3.1	业务逻辑组件的接口	521
10.3.2	业务逻辑组件的实现类	522
10.3.3	部署业务逻辑组件	525
10.4	开发 DWR 前端处理类	526
10.4.1	实现 DWR 前端处理组件	526
10.4.2	为前端处理器增加权限控制	528
10.5	处理异步请求	529
10.5.1	将 Spring 容器中的 Bean 创建成 JavaScript 对象	529
10.5.2	处理用户登录	531
10.5.3	处理用户注册	532
10.5.4	处理消息发布	534
10.5.5	获取消息列表	535
10.5.6	处理分页	536
10.5.7	查看消息内容	537
10.5.8	页面加载时的处理	538
10.6	本章小结	539
第 11 章	基于 Ext JS 4.1 的应用：简易 Blog 系统	540
11.1	实现 Hibernate 持久层	541
11.1.1	设计 Hibernate 的持久化类	541
11.1.2	完成映射文件	543
11.2	实现 DAO 组件	545
11.2.1	DAO 接口定义	546
11.2.2	实现 DAO 组件	546
11.2.3	配置 DAO 组件	547
11.3	实现业务逻辑组件	548
11.3.1	业务逻辑组件的接口	548
11.3.2	业务逻辑组件的实现类	549
11.3.3	配置业务逻辑组件	552
11.4	使用 Ext JS 开发前端	553

11.4.1	初始化界面	553	12.4.4	处理用户竞价	588
11.4.2	显示指定 Blog 文章	557	12.4.5	判断拍卖物品状态	590
11.4.3	显示回复列表	559	12.4.6	事务管理	592
11.4.4	添加回复	561	12.4.7	配置业务层组件	592
11.4.5	发表新 Blog 文章	565	12.5	暴露前端处理方法	595
11.5	本章小结	569	12.5.1	定义 DWR 前端处理类	595
第 12 章	Ajax 实例: 电子拍卖系统	570	12.5.2	初始化 Spring 容器	597
12.1	总体说明和概要设计	571	12.5.3	配置 DWR 的核心 Servlet	597
12.1.1	系统的总体架构设计	571	12.5.4	暴露前端处理方法	598
12.1.2	数据库设计	572	12.5.5	对前端处理方法增加权限检查	600
12.2	实现 Hibernate 持久化类	573	12.6	调用前端处理方法响应用户请求	601
12.2.1	设计 Domain Object	574	12.6.1	页面加载时的函数	601
12.2.2	实现 Domain Object	575	12.6.2	处理返回首页的请求	602
12.3	DAO 层实现	579	12.6.3	浏览所有流拍物品	602
12.3.1	DAO 的基础配置	579	12.6.4	处理用户登录	605
12.3.2	实现 DAO 组件	580	12.6.5	管理物品	609
12.3.3	部署 DAO 组件	583	12.6.6	管理物品种类	612
12.4	业务逻辑层实现	584	12.6.7	查看竞得物品	616
12.4.1	设计业务逻辑组件	584	12.6.8	查看自己的竞价记录	617
12.4.2	依赖注入 DAO 组件	586	12.6.9	浏览拍卖物品	619
12.4.3	业务逻辑组件的异常处理	587	12.6.10	参与竞价	621
			12.7	本章小结	624

第 1 章

Ajax 概述与 Ajax 初体验

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 应用的改进和优势
- ✎ Ajax 的基本特征
- ✎ Ajax 依赖的核心技术
- ✎ Ajax 技术带来的改变
- ✎ Ajax 技术的优势
- ✎ 体验 Ajax 聊天室的便捷
- ✎ 开发 JSP 聊天室
- ✎ 开发 Ajax 聊天室
- ✎ 发送异步请求
- ✎ 使用 Servlet 生成文本响应
- ✎ 使用 JSP 生成文本响应
- ✎ 获取服务器的响应内容
- ✎ 通过 DOM 加载服务器响应
- ✎ Ajax 编程的技术难点
- ✎ Ajax 和传统 Web 应用的对比

Ajax 的全称是 Asynchronous JavaScript And XML, 即异步 JavaScript 和 XML, 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。

Ajax 并非一种全新的技术, 它所使用的 JavaScript、CSS、DOM 等技术早已存在。Ajax 通过这些传统的对象来改善用户的交互体验, 让用户能异步发送请求: 用户可以在浏览页面的同时, 向服务器发送请求。Ajax 技术的核心在于异步请求, 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 技术带给互联网一场全新的革命。目前, 几乎所有的 B/S 应用都广泛地使用了 Ajax 技术。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。

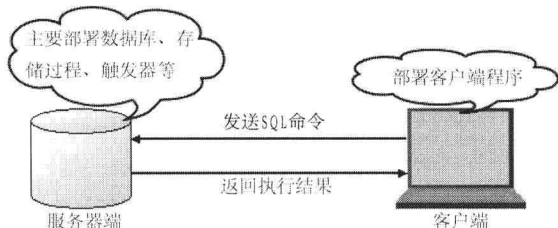


图 1.1 C/S 结构应用

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,这种结构的软件适用于用户数目不多的局域网内。早期的大部分 ERP 软件产品即属于此类结构。

随着 Internet 技术的兴起, B/S (浏览器/服务器模式) 结构得到了大规模应用。B/S 结构是对 C/S 结构的一种改进。在这种结构下,应用的业务逻辑完全在应用服务器端实现,用户表现完全在 Web 服务器上实现,客户端只需要浏览器即可进行业务处理,是一种全新的软件系统结构技术。这种结构是当今应用软件的首选体系结构。

在这种应用结构下,客户端的所有处理请求都以 HTTP 请求形式发送,而服务器端则将响应以 HTML 页面的形式送回客户端,由客户端浏览器负责显示 HTML 页面。B/S 结构应用的结构图如图 1.2 所示。

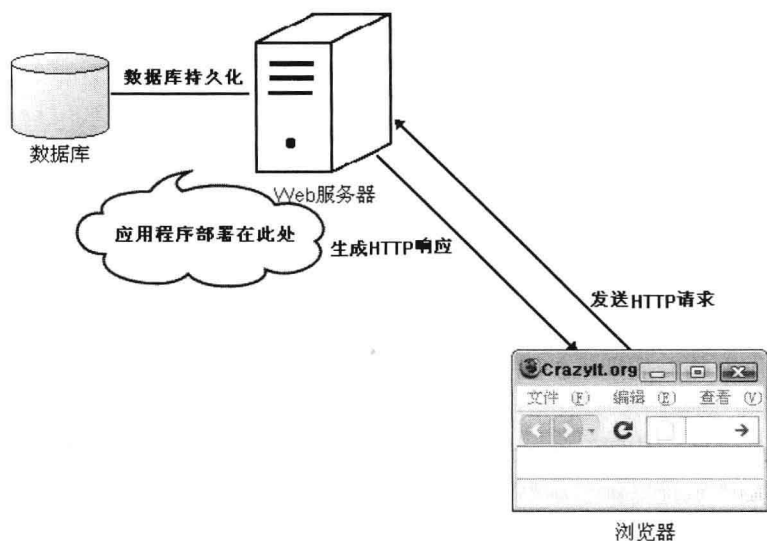


图 1.2 B/S 结构的应用

B/S 结构得到迅速推广是有原因的。在大部分情况下, B/S 结构的应用比 C/S 结构的应用更加优秀,适应性更广。相对而言, B/S 结构的系统有如下方面的优势:

- 数据安全性高。由于 C/S 结构软件的数据分布特性,客户端所发生的火灾、地震、盗抢、病毒、黑客攻击等都成了可怕的数据杀手。另外,对于集团级的异地软件应用, C/S 结构的软件必须在各地安装多台服务器,并在多台服务器之间进行数据同步。因此,每个数据点上的数据安全都会影响到整个应用的数据安全。对于跨区域的大型应用而言, C/S 结构软件的安全性是令人无法接受的。而对 B/S 结构的系统而言,数据集中存放于总部的数据库服务器(服务器数据可以通过多种方式备份存放),客户端不保存任何业务数据和数据库持久化信息,无须进行数据同步,所以这些安全问题自然也就不存在了。
- 数据一致性好。在 C/S 结构的解决方案里,对于跨区域的大型企业应用都采用各地安装区域级服务器,然后再进行数据同步的模式。这些服务器每天必须同步完毕之后,总部才可得到最终的数据。由于局部网络故障等原因,可能造成个别数据库无法正常同步。即使都能正常同步,但各服务器往往不能同时同步,因而数据也无法绝对一致,

不能用于决策。而对于 B/S 结构的系统而言,数据是集中存放的,客户端发生的每次数据修改都直接进入中央数据库,不存在数据一致性的问题。

- 数据实时性好。对于大型的跨区域应用而言, C/S 结构不可能随时跟踪各客户端的业务发生情况,因为数据都不是实时更新的,因而看到的数据都是滞后的;而 B/S 结构则不同,因为其数据都是实时存入服务器端的数据库,因而服务器端可以实时看到当前发生的所有业务,能提供更好的企业决策支持。
- 系统更新方便。软件供应商提供的软件不可能是完美无缺的。即使是一个绝对完美的软件系统,当具体的业务环境发生改变后,系统也应随之改变。因而,必须经常对已部署的软件产品进行维护、升级。对 C/S 结构的软件而言,由于其应用是分布的,需要对每一个节点手动进行程序安装,所以,即使非常小的程序缺陷都需要很长的时间来重新部署。重新部署时,为了保证各程序版本的一致性,必须暂停业务进行更新(即“休克更新”)。在很多情景下,这是不可忍受的。而 B/S 结构的软件则不同,其应用程序集中于服务器上,各应用节点没有任何程序,应用的更新只需要更新服务器端程序即可,因而可以做到快速的服务响应。

传统的 C/S 结构软件开发工具有早期的 Visual Basic、Visual FoxPro 等,这些开发工具目前已经趋于淘汰。目前依然使用的开发工具有 PowerBuilder、Delphi 等。除早期的一些系统外,现在新开发的系统大部分使用 B/S 结构。

B/S 结构的应用开发早期有 ASP、JSP 和 PHP 等。早期这些 B/S 结构开发技术相当混乱,系统中业务逻辑、数据持久化、控制逻辑混在一起,这些处理逻辑都通过页面的脚本实现。早期的 B/S 结构应用面临着后期维护困难、难以扩充的问题。

MVC 设计模式重新定义了 B/S 结构应用的开发模式,规定 B/S 结构应用应该分成 Model (M: 模型)、View (V: 视图) 和 Controller (C: 控制器) 三个部分。MVC 模式分离的数据访问和数据表现,给系统提供了更好的解耦。MVC 架构的核心思想是,将程序分成相对独立而又能协同工作的三个部分。通过使用 MVC 架构,可以降低模块之间的耦合,提供应用的扩展性。MVC 中的每个组件只关心组件内的逻辑,不应与其他组件的逻辑混合。

Java EE 的出现,则更加规范了 B/S 结构应用的开发。Java EE 推荐将应用分为数据持久层、业务逻辑层和 Web 层,各层之间以松耦合的方式组织在一起。

目前, Ajax 的出现,再次完善了传统的 Web 应用。Ajax 应用强调异步发送用户请求:用户在浏览页面的同时可以发送请求,在第一个请求的服务器响应还没有完全结束时,可以再次发送请求。这种请求的发送方式非常类似于传统的 C/S 结构的应用。

在传统的 Web 应用里,因为用户总是需要加载新页面时才提交请求,而提交请求后又需要等待服务器响应,所以如果服务器响应还没有完全结束,则用户只能等待,不能继续发送请求。

与传统 Web 应用不同的是, Ajax 将请求与页面分离:在传统的 Web 应用里,每个请求即对应一个页面,不管客户端以 POST 还是 GET 方式提交请求,每次请求都会丢弃当前页面,等待服务器生成新页面。在等待期间,旧的页面已经丢弃,新的页面还没有完全生成,整个浏览器将一片空白,而用户什么都做不了,只能等待,对于用户而言,这是一种不连续的体验,感觉非常不好。