

## <<企业级Web开发实战>>

### 图书基本信息

书名：<<企业级Web开发实战>>

13位ISBN编号：9787121084768

10位ISBN编号：7121084767

出版时间：2009-4

出版时间：电子工业出版社

作者：雨虹 等编著

页数：482

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<企业级Web开发实战>>

### 前言

最近几年，Java和Web技术都发生了巨大的变化。

对企业应用而言，从外围的门户系统到核心的业务系统，Web化已经相当普遍，随之而来的则是计算交互模式的再一次变化。

从主机应用、C/S应用与Web应用，再到Enterprise Web 2.0的提出，背后体现的都是计算交互方式的变化。

表现在这几年与用户的交互，成为技术关注的重点。

Java Web表示层的技术多种多样，变化也比较大。

丰富的用户体验，对企业级应用不只是展现方式的炫和酷，而是通过个性化推送、用户行为模式分析等，提高应用系统实效的重要手段。

根据我这几年的咨询经历，企业的领导者而不仅仅是过去的IT技术人员，都在考虑如何快速应用新的IT交互技术，从而增强企业的竞争力。

大约在2004年年初，我的一个同事与好友在做Rich Client的项目。

闲聊起来，他向我介绍说有个Java设计师不错，基本功比较扎实；而他的肯定通常是相当可靠的。

那是我第一次认识了本书的作者之一雨虹，爱打太极拳的小伙子，朴实的笑容。

当时给我的印象是他真心地喜爱编程，喜欢钻研，不仅仅看做一份工作的技能，而热爱是最好的老师。

。

这些年雨虹和他的团队出色地完成了许多项目，他们也都成为了有丰富经验的架构师。

本书的作者们结合多年的工作经验，在书中详细地介绍了Web表现层编程的现状、发展以及大量的实践经验。

总体来看，这本书以JSF为主线，同时又介绍了Web 2.0的一些开发技术：Ajax及一些流行的Ajax框架，如ExtJS，并将两者有机地融合起来，涵盖了Web开发的基本方面。

相信本书会为读者提供非常有益的帮助。

新的企业级应用将从交互模式方式上而重新改写。

## <<企业级Web开发实战>>

### 内容概要

本书汇集了许多大型企业级Web开发的优秀实践经验，内容包含了大量详实的实例：从Web开发基础，到JSF组件（RichFaces），再到纯JavaScript框架（ExtJS），以及结合两者优点的ExtFaces，都详细地进行了分析和总结，其中很多实践解决方案和样例代码稍作修改就可以直接应用于项目开发之中。

结合书中的内容，读者可以根据自己的项目的特点，采用书中的一种或者多种技术灵活地制定实际项目的解决方案。

本书内容广泛，深入浅出，既适合开发人员作为多种Web UI技术的入门指导，也适合架构师作为技术选型的参考手册。

# <<企业级Web开发实战>>

## 书籍目录

第 部分 Web UI开发基础	第1章 Web UI编程综述	1.1 Web 2.0 介绍	1.1.1 Web的发展历史
1.1.2 Web 2.0相关技术	1.1.3 SOA与Web 2.0	1.2 UI 编程技术	1.2.1 Rich Client
Internet 编程	1.2.2 Ajax 编程	1.2.3 Ajax in JSF编程	1.3 本章小结
2.1 Ajax的基本知识	2.1.1 Ajax基本概念	2.1.2 Ajax的基本特点	第2章 Ajax 基础知识
2.2.1互联网和HTML的诞生	2.2.2 HTML的发展	2.3 DOM 简介	2.2 一切从浏览器说起
2.3.2 DOM事件	2.3.3 DOM事件流	2.3.4 事件处理函数	2.3.1 HTML
2.4.1 CSS简介	2.4.2 基本的CSS语法	2.4.3 CSS属性	2.4 CSS
2.4.4 CSS选择器	2.4.5	2.5 JavaScript简介	2.5.1 JavaScript入门示例
2.5.2 JavaScript基本数据	2.5.3 JavaScript的基本构成	2.5.4 事件驱动及事件处理	2.6 XML简介
2.6.1	2.6.2 XML的优点	2.6.3 一个简单的XML文档	2.6.4 XML文档的整体结构
2.6.5 XML文档的实质内容——元素	2.6.6 字符数据与实体引用	2.6.7 标记	2.6.8 CDATA
2.6.9 注释	2.7 JSON简介	2.8 XMLHttpRequest对象简介	2.8.1
XMLHttpRequest对象的属性和事件	2.8.2 XMLHttpRequest对象的方法	2.8.3 发送请求和处理请求	2.9 本章小结
第3章 Ajax框架介绍	3.1 Prototype	3.1.1 什么是Prototype	3.1.2 Prototype的下载和引入
3.1.3 Prototype常用函数介绍	3.1.4 Prototype的Ajax功能	3.2 script.aculo.us	3.2.1 script.aculo.us 简介
3.2.2 script.aculo.us 的引入和使用	3.2.3	3.3 jQuery	3.3.1 jQuery 简介
3.3.2 jQuery的使用	3.4 Sarissa	3.4.1 Sarissa 介绍	3.4.2 Sarissa的使用
3.5 本章小结	第 部分 JSF应用开发	第4章 JSF介绍	4.1 什么是JSF
4.1.1 JSF简介	4.1.2 JSF的体系结构	4.1.3 为什么要使用JSF	4.2
4.3 配置JSF	4.3.1 下载	4.3.2 安装配置	4.3.3 HelloWorld
4.3.4 JSF的元素	4.4.1 UI组件	4.4.2 JSF生命周期	4.4.3 数据转换与验证
4.4.4 JSF事件处理	4.4.5 JSF表达式语言	4.5 JSF与Spring结合	4.6 JSF解决方案
4.6.1 DataTable分页	4.6.2 一般分页	4.6.3 On-Demand分页	4.6.4 Exception统一处理
4.6.5 Shale框架验证	4.7 本章小结	第5章 Facelets	5.1 Facelets简介
5.2 配置Facelets	5.2.1 下载	5.2.2	5.2.3 HelloWorld示例
5.3 Facelets模板和扩展机制	5.3.1 UI Component	5.3.2 Facelets include标签的用法	5.4 Facelets自定义标签
5.5 本章小结	第6章	6.1 RichFaces简介	6.2 使用条件
6.3 配置RichFaces	6.3.1 下载	6.3.2 安装及配置	6.3.3 HelloWorld示例
6.3.4 RichFaces配置进阶	6.4 RichFaces的基本	6.4.1 简介	6.4.2 RichFaces的体系架构
6.4.3 如何发送Ajax请求	6.4.4	6.4.5 决定要重绘的区域	6.5 RichFaces开发工具介绍
6.5.1 Red Hat	6.5.2 Eclipse WTP	6.6 RichFaces常用组件介绍	6.6.1 Ajax日志
6.6.2 Ajax监听器	6.6.3 参数	6.6.4 按钮	6.6.5 链接
6.6.6 Ajax状态	6.6.7 扩展Ajax事件	6.6.8 日历控件	6.6.9 列表移动
6.6.10 可排序列表	6.6.11	6.6.12 模式对话框	6.6.13 面板条
6.6.14 可滚动的数据表格	6.6.15 可	6.6.16 标签页	6.6.17 工具提示
6.6.18 自动完成对话框	6.6.19 树	6.6.20 内容菜单	6.6.21 页面效果
6.6.22 数据表格	6.7 优化Ajax请求	6.7.1 优	6.7.2 优化数据
6.8 异常处理	6.8.1 Ajax请求错误处理	6.8.2 Session过	6.9 局限和不足
6.10 本章小结	第7章 Seam入门	7.1 Seam简介	7.2 配置Seam
7.2.1 下载	7.2.2 安装和配置	7.2.3 HelloWorld	7.3 Seam与JSF
7.4 再谈数据表	7.4.1 单纯RichFaces的解决方案	7.4.2 RichFaces + Seam的解决方案	7.5 本
7.5 本章小结	第 部分 Ext应用开发	第8章 ExtJS框架的介绍和使用	8.1 JavaScript面向对象编程
8.2	8.3 ExtJS布局 ( layout )	8.4 嵌套布局 ( NestedLayout )	8.5 表单组件 ( Ext.form.FormPanel )
8.6 树组件 ( Ext.tree.TreePanel )	8.7 对话框组件	( Ext.Window )	8.8 表格组件 ( Ext.grid.GridPanel )
8.9 菜单组件 ( Ext.menu.Menu )	8.10	8.10.1 Ajax 组件	8.10.2 Template和XTemplate组件
8.10.3 DomHelper组件			

## <<企业级Web开发实战>>

8.11 国际化    8.12 开发工具    8.13 本章小结    第9章 DWR和JSON-RPC-Java的介绍及使用  
9.1 DWR框架的介绍和使用    9.1.1 DWR基本概念    9.1.2 使用DWR    9.1.3 使用自定义对象  
9.1.4 DWR的配置    9.2 JSON-RPC-Java框架的介绍和使用    9.2.1 基本概念  
9.2.2 安装和配置    9.3 DWR和JSON-RPC-Java的简单对比    9.4 本章小结    部分 JSF与Ext的结合应用  
第10章 基于Ext的JSF组件——ExtFaces    10.1 ExtFaces简介    10.1.1 ExtFaces的来源  
10.1.2 ExtFaces采用的技术    10.1.3 JSF开发现状    10.1.4 Ext和JSF的结合方案    10.2 ExtFaces原理  
10.3 ExtFaces目标    10.4 ExtFaces组件介绍    10.4.1 简单组件    10.4.2 嵌套组件  
10.5 开发计划    10.6 本章小结

## &lt;&lt;企业级Web开发实战&gt;&gt;

## 章节摘录

第 部分 Web UI开发基础      第1章 Web UI编程综述      1.1 Web 2.0介绍      在介绍web 2.0之前，还是让我们从Web的发展历史开始说起吧。

1.1.1web的发展历史      1.1.1.1从无到有      web是一种典型的分布式应用架构。web应用中的每一次信息交换都要涉及客户端和服务端两个层面。因此，Web开发技术大体上也可以被分为客户端技术和服务器端技术两大类。我们先来谈谈客户端技术的萌芽和演进过程。

web客户端的主要任务是展现信息内容，而HTML语言则是信息展现的最有效载体之一。作为一种实用的超文本语言，HTML的历史最早可以追溯到20世纪40年代。

1945年，vannevar Bush在一篇文章中阐述了文本和文本之间通过超级链接相互关联的思想，并在文中给出了一种能实现信息关联的计算机Memex的设计方案。

Doug Engelban等人则在1960年前后，对信息关联技术做了最早的实验。

与此同时，TedNelson正式将这种信息关联技术命名为超文本（Hypertext）技术。

1969年，IBM的Charles Goldfarb发明了可用于描述超文本信息的GML（Generalized Markup Language）语言。

1978到1986年间，在ANSI等组织的努力下，GML语言进一步发展成为著名的标准通用标记语言（Standard Generalized Markup Language，SGML）。

1989年，欧洲粒子物理研究所（European Council for Nuclear Research，CERN）的Tim Berners-Lee意识到，与其简单地引用其他人的工作，为什么不干脆链接呢？

读一篇文章时，科学家可以打开所引用的文章。

超文本（Hypertext）当时相当流行，并利用了他先前在文档和文本处理方面的研究成果。

Tim Berners-Lee同时感到SGML是描述超文本信息的一个上佳方案，但美中不足的是SGML。

过于复杂，不利于信息的传递和解析。

于是，Tim Berners-Lee对SGML语言做了大刀阔斧的简化和完善，提出了SGML的一个子集，称为超文本标记语言（HyperText Markup Language，HTML）。

Tim Berners-Lee不仅创建了一个称为超文本传输协议（HyperText Transfer Protocol，HTTP）的简单协议，还发明了第一个Web浏览器，叫做WorldWideWeb。

Tim Berners-Lee提出的HTML获得巨大的应用，他本人也由此获得了“互联网之父”的称号。

1.1.1.2Web 1.0发展概述      在HTML提出之后，Web获得了飞快的发展，下面从服务器端和客户端两个方面说起。

1.客户端Web的发展      最初的HTML语言只能在浏览器中展现静态的文本或图像信息，这满足不了人们对信息丰富性和多样性的强烈需求——这件事情最终的结果是，由静态技术向动态技术的转变成为了Web客户端技术演进的永恒定律。

能存储、展现二维动画的GIF图像格式早在1989年就已发展成熟。

Web出现后，GIF第一次为HTML页面引入了动感元素。

但更大的变革来源于1995年Java语言的问世。

Java语言天生就具备的平台无关的特点，让人们一下子找到了在浏览器中开发动态应用的捷径。

1996年，著名的Netscape浏览器在其2.0版中增加了对Java Applet和JavaScript的支持。

Netscape的竞争对手，Microsoft的IE 3.0浏览器也在这一年开始支持Java技术。

现在，开发人员可以用Java和JavaScript语言开发丰富HTML页面的功能了。

顺便说一句，JavaScript语言在所有客户端开发技术中占有非常独特的地位：它是一种以脚本方式运行的、简化了的Java语言，这也是脚本技术第一次在Web世界里崭露头角。

为了用Microsoft的技术与JavaScript抗衡，Microsoft还为1996年的IE 3.0设计了另一种后来也声名显赫的脚本语言——VBScript语言。

真正让HTML页面表现更加丰富和强大的，则是CSS（Cascading Style Sheets）和DHTML（Dynamic HTML）技术。



## &lt;&lt;企业级Web开发实战&gt;&gt;

1996年年底，W3C提出了CSS的建议标准，同年，IE 3.0引入了对CSS的支持。

CSS大大提高了开发者对信息展现格式的控制能力。

1997年的Netscape 4.0不但支持CSS，而且增加了许多Netscape公司自定义的动态HTML标记，这些标记在CSS的基础上，让HTML页面中的各种要素相互交互。

1997年，Microsoft发布了IE 4.0，并将动态HTML标记、CSS和动态对象模型（DHTML Object Model）发展成了一套完整、实用、高效的客户端开发技术体系，Microsoft称其为DHTML。

同样是实现HTML页面的动态效果，DHTML技术无须启动Java虚拟机或其他脚本环境，可以在浏览器的支持下，获得更好的展现效果和更高的执行效率。

为了在HTML页面中实现音频、视频等更为复杂的多媒体应用，1996年的Netscape 2.0成功地引入了对QuickTime插件的支持，插件这种开发方式也迅速风靡了浏览器的世界。

在Windows平台上，Microsoft将客户端应用集中在了20世纪90年代中期刚刚问世的COM和ActiveX身上。

1996年，IE 3.0正式支持在HTML页面中插入ActiveX控件的功能，这为其他厂商扩展Web客户端的信息展现方式开辟了一条自由之路。

1999年，RealPlayer插件先后在Netscape和IE浏览器中取得了成功，与此同时，Microsoft将媒体播放软件Media Player也被预装到了各种Windows版本之中。

同样值得纪念的还有Flash插件的横空出世：20世纪90年代初期，Jonathan Gay在Futurewave公司开发了一种名为Future Splash Animator的二维矢量动画展示工具，1996年，Macromedia公司收购了FutureWave，并将Jonathan Gay的发明改名为我们熟悉的Flash。

从此，Flash动画成了Web开发者表现自我、展示个性的最佳方式。

除了编写HTML页面之外，客户端应用的开发者还可以利用一些成熟的技术将浏览器的功能添加到自己的应用程序中。

从1992年开始，W3C就免费向开发者提供libwww开发库。

借助libwww，我们可以自己编写Web浏览器和Web搜索工具，也可以分析、编辑或显示HTML页面。

1999年，Microsoft在IE 5.0中引入的HTAs（HTML Applications）技术则允许我们直接将HTML页面转换为一个真正的应用程序。

从1997年的IE 4.0开始，Microsoft为开发者提供了WebBrowser控件和其他相关的COM接口，允许程序员在自己的程序中直接嵌入浏览器窗口，或调用各种浏览器的功能，如分析或编辑HTML页面等。Windows 98及其后的Windows操作系统甚至还利用WSH（Windows Script Host）技术将原本只在浏览器中运行的JavaScript、VBScript变成了可以在Win32环境下使用的通用脚本语言，这些都极大地扩展和增强了Web客户端开发技术。

**2.服务器端技术的成熟与发展** 与客户端技术从静态向动态的演进过程类似，Web服务器端的开发技术也是由静态向动态逐渐发展、完善起来的。

最早的Web服务器简单地响应浏览器发来的HTTP请求，并将存储在服务器上的HTML文件返回给浏览器。

一种名为SSI（Server Side Includes）的技术可以让Web服务器在返回HTML文件前，更新HTML文件的某些内容，但其功能非常有限。

第一种真正使服务器能根据运行时的具体情况，动态生成HTML页面的技术是大名鼎鼎的CGI（Common Gateway Interface）技术。

1993年，CGI 1.0的标准草案由NCSA（National Center for Super computing Applications）提出；以享受到信息检索、信息交换、信息处理等更为便捷的信息服务。

早期的CGI程序大多是编译后的可执行程序，其编程语言可以是C、C++、Pascal等任何通用的程序设计语言。

为了简化CGI程序的修改、编译和发布过程，人们开始探寻用脚本语言实现CGI应用的可行方式。

在此方面，不能不提的是Larry Wall于1987年发明的Perl语言。

Perl结合了C语言的高效，以及sh、awk等脚本语言的便捷，似乎天生就适用于CGI程序的编写。

1995年，第一个用Perl写成的CGI程序问世。

## <<企业级Web开发实战>>

很快，Perl在CGI编程领域的风头就盖过了它的前辈C语言。

随后，Python等著名的脚本语言也陆续加入了CGI编程语言的行列。

1994年，Rasmus Lerdorf发明了专用于Web服务器端编程的PHP（Personal HomePage Tools）语言。与以往的CGI程序不同，PHP语言将HTML代码和PHP指令合成为完整的服务器端动态页面，Web应用的开发者可以用一种更加简便、快捷的方式实现动态Web功能。

1996年，Microsoft借鉴PHP的思想，在其Web服务器IIS 3.0中引入了ASP技术，ASP使用的脚本语言是我们熟悉的VBScript和JavaScript。

借助Microsoft visualStudio等开发工具在市场上的成功，ASP迅速成为了Windows系统下Web服务器端的主流开发技术。



## <<企业级Web开发实战>>

### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>