

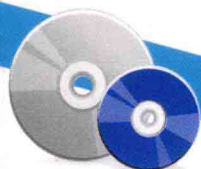
Broadview®
www.broadview.com.cn

快速编程

Ext JS Web

开发技术详解

郭峰
等编著



3小时多媒体教学视频
127个应用实例
2个典型案例
.....

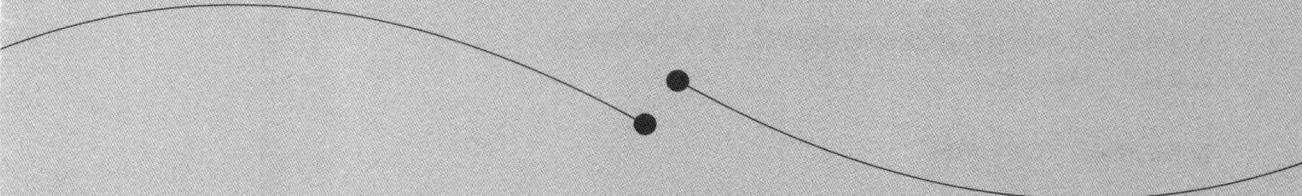
一线人员，为您展示Ext JS Web开发的独特魅力!



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

快意编程

Ext JS Web 开发技术详解



郭峰 等编著

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

内 容 简 介

本书是笔者在多年项目开发过程中的经验总结,它通过丰富的实例由浅入深、循序渐进地介绍了目前采用 Ext JS 进行 Web 开发的使用方法,从而帮助软件设计人员快速掌握 Ext JS 开发技术的使用方法,并能将其应用于实战中。

通过本书的学习,读者可以了解到 Ext JS 的使用方法,熟练掌握如何运用这些技术来帮助前台界面设计人员进行开发。为了帮助众多初学者快速掌握 Ext JS 的开发方法,笔者精心编著了本书。它是笔者在多年项目实践中的经验总结。本书首先对 Ext JS 进行了概述,然后通过一个简单的示例带领读者快速入门,在读者对 Ext JS 有了初步印象后,又重点介绍了 JavaScript 的面向对象技术、Ext JS API 文档的使用方法、Ext JS 的开发流程,然后按照 Ext JS 的开发顺序,首先讲解了页面的布局设定,接着详细讲解了各种常用组件的使用方法,并对实际开发中常用的画面场景进行了总结,最后通过完整的示例演示了 Ext JS+REST+Spring+Hibernate 的整合方案。

本书适用于初、中、高级软件设计人员,同时也可用做高校相关专业师生和社会培训班的教材,但在阅读本书前,需要读者掌握 JavaScript 的基础知识。

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

快意编程: Ext JS Web 开发技术详解 / 郭峰等编著. —北京: 电子工业出版社, 2010.3
ISBN 978-7-121-10435-0

I. 快… II. 郭… III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2010) 第 030400 号

责任编辑: 胡辛征

文字编辑: 江 立

印 刷: 北京智力达印刷有限公司

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: 29.75 字数: 750 千字

印 次: 2010 年 3 月第 1 次印刷

印 数: 4000 册 定价: 59.00 元 (含 DVD 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

书山有路勤为径，学海无涯苦作舟。

——韩愈

当开发人员厌倦采用 JSP 作为前台技术进行 Web 开发，而其他相应的前台技术比如 JSF 等并不成熟时，Ext JS 出现了，它迅速成为前台页面开发技术的主流，紧接着 REST 也异军突起，因此，Ext JS+REST+Spring+Hibernate 的组合成了采用 Java 进行 Web 开发的主要技术。

目前，市场上虽然也有讲解 Ext JS 的书籍，但大多是对 Ext JS 一些组件的属性、方法、事件的简单罗列，其实这些内容通过查看 Ext JS 本身提供的 API 文档即可获知，导致读者即便是看完了一本书，也并不能马上进行开发。本书在编排上并不会对 Ext JS 简单列出组件的属性、方法、事件，而是通过一个个示例，力争使读者读完本书后，不用再看其他书籍，即可进入 Ext JS 项目开发的实战。

本书有何特点

(1) 由易到难、通俗易懂：本书从最基本的 JavaScript 面向对象技术讲起，这是掌握 Ext JS 的必备基础，只有掌握了最基本的 JavaScript 面向对象技术，才能真正掌握好 Ext JS，在讲解各种组件的使用方法时，我们将通过简单的示例来演示各种组件的使用方法。

(2) 技术涵盖广：本书讲解了 Ext JS 的大部分常用组件，比如表单、表格、树、AJAX 和数据存储等，这些组件都是日常开发能够用到的。

(3) 不是简单的 API 罗列：为了让读者真正快速地掌握 Ext JS 的开发技术，本书没有对 Ext JS 常用的 API 进行简单罗列，而是通过实际的应用场景让读者一步一步地掌握这些属性、方法和事件的使用方法。

(4) 最新的技术组合：本书最后讲解了 Ext JS+REST+Spring+Hibernate 的组合，这是目前开源领域最新的技术组合。

(5) 案例丰富，实战性强：每一个案例都是笔者在实际项目开发过程中使用过的，非常具有典型性。

本书包括的内容

本书从大的方面讲主要包含 3 部分内容。

第一部分首先对 Ext JS 相关基础知识进行介绍,然后带领读者搭建 Ext JS 的开发环境,并快速实现 Ext JS 的入门。

第二部分首先对 JavaScript 的面向对象技术进行了简单讲解,因为这是掌握 Ext JS 的基础,接着按照 Ext JS 的开发顺序,先讲解 Ext JS 的整体框架、开发模式,接着详细讲解了它的布局、表单组件、表格组件、树形组件、窗口组件、工具栏和菜单栏组件以及 Ext JS 在 AJAX 方面的应用,然后讲解了 REST 的相关知识,最后通过对 Ext JS 的封装,从而提高开发人员的开发效率。

第三部分通过对博客管理系统和工时统计管理系统示例的讲解,详细演示了 Ext JS+REST+Spring+Hibernate 的组合方式,从而使读者通过本书的学习,马上可以进行项目实战。

适合阅读本书的读者

本书具有技术涵盖广、案例丰富、实战性强、通俗易懂的特点,力求以通俗的语言及丰富的实例来指导读者透彻掌握如何将目前流行的 Ext JS、REST、Spring2、JPA 等技术运用在实际的程序开发中,但本书主要是对 Ext JS 进行讲解,如果读者想要了解 REST、Spring2、JPA 的相关知识,需要参考相应的书籍,本书适用于初、中、高级软件设计人员,同时也可用做高校相关专业师生和社会培训班的教材,但在阅读本书前,需要读者掌握 JavaScript 的基础知识。

本书作者

本书主要由郭锋编写,其他参与编写的人员有吴燃、方振宇、陈冠佐、傅奎、陈勤、梁洋洋、毕梦飞、陈庆、柴相花、陈非凡、陈华、陈嵩、承卓、陈先在。在此表示感谢!

作 者

2010 年 1 月

目 录

第一篇 Ext JS 相关知识介绍

第 1 章 Ext JS 介绍	2
1.1 Ext JS 概述	2
1.1.1 Ext JS 的起源	2
1.1.2 Ext JS 的许可协议	4
1.1.3 Ext JS 的应用示例	4
1.2 Ext JS 与 AJAX	9
1.2.1 XMLHttpRequest 技术	9
1.2.2 JavaScript (Java 脚本语言)	11
1.2.3 DOM (文档对象模型)	13
1.2.4 CSS (样式表)	14
1.2.5 XML	14
1.3 与 Ext JS 同类的 AJAX 框架	15
1.3.1 Prototype	16
1.3.2 Dojo	16
1.3.3 DWR	18
1.3.4 GWT	19
1.3.5 Ext JS 与上述框架的区别	19
1.4 小结	20
第 2 章 Ext JS 快速入门	21
2.1 建立 Ext JS 开发环境	21
2.1.1 下载 Ext JS	21
2.1.2 下载 Eclipse	24
2.1.3 下载并整合 Ext JS 的开发工具 Spket	25
2.1.4 下载 Ext JS 的调试工具 Firebug	29
2.2 使用 JavaScript 实现 HelloWorld 的示例	32
2.3 使用 Ext JS 实现 HelloWorld 的示例	33
2.3.1 编写 HTML 页面 myHelloWorld.htm	33
2.3.2 引入 Ext JS 类库	34

2.3.3 运行调试 Ext JS	34
2.4 小结	34

第二篇 Ext JS 技术详解

第 3 章 JavaScript 面向对象技术	36
3.1 为什么要先讲 JavaScript	36
3.2 JavaScript 的自定义函数	42
3.2.1 函数的定义	42
3.2.2 函数的使用	43
3.3 JavaScript 的面向对象编程	43
3.3.1 一切皆是对象	43
3.3.2 prototype 原型对象	44
3.3.3 如何实现类的继承	45
3.4 如何模仿 Java 类库的引入机制实现 JS 文件的引入	45
3.5 JavaScript 操作 DOM 示例	46
3.6 小结	57
第 4 章 Ext JS 框架	58
4.1 Ext JS 类库	58
4.1.1 Ext JS 类结构	58
4.1.2 Ext.extend 介绍	59
4.1.3 Ext.apply 介绍	68
4.1.4 Ext.onReady 介绍	69
4.2 Ext JS 组件	70
4.2.1 Ext JS 组件结构	70
4.2.2 Ext.Component 介绍	72
4.2.3 Ext.ViewPort 介绍	73
4.2.4 Ext.Panel 介绍	75
4.3 Ext JS API 文档使用指南	76
4.3.1 Ext JS API 文档结构	76
4.3.2 如何使用 Ext JS API 文档	77
4.4 小结	85
第 5 章 Ext JS 开发模式	86
5.1 程序规划	86
5.1.1 设计 HTML 文件	86
5.1.2 设计对应的 JS 文件	87
5.2 使用 Ext.onReady 函数	95

5.2.1	创建命名空间	95
5.2.2	自定义函数	95
5.3	确定页面布局	97
5.4	使用组件	98
5.5	与后台进行交互	100
5.6	小结	101
第 6 章	Ext JS 布局	102
6.1	Ext JS 的布局方式	102
6.2	Border 布局	106
6.3	Fit 布局	110
6.4	Accordion 布局	111
6.5	Form 布局	115
6.6	Column 布局	116
6.7	Card 布局	120
6.8	Table 布局	121
6.9	小结	124
第 7 章	表单组件	125
7.1	Form 表单简介	125
7.1.1	Form 表单的基本属性	125
7.1.2	Ext JS 对 Form 表单的封装	128
7.2	Ext JS 的表单组件	132
7.2.1	文本输入	134
7.2.2	多行文本输入	136
7.2.3	单选框	138
7.2.4	复选框	139
7.2.5	下拉列表	141
7.2.6	日期元件	141
7.3	Ext JS 表单组件的使用	142
7.4	小结	147
第 8 章	表格组件	148
8.1	Table 表格简介	148
8.1.1	Table 表格的基本属性	148
8.1.2	Ext JS 对 Table 表格的封装	150
8.2	Ext JS 的表格元件	152
8.2.1	常用配置	152
8.2.2	常用事件	154

8.3	Ext JS 表格组件应用示例	154
8.3.1	合理划分 JS 文件	154
8.3.2	定义表格组件	155
8.4	采用表格组件展示数据	161
8.5	小结	169
第 9 章	表格组件的扩充	170
9.1	可编辑表格组件的继承模型	170
9.2	Ext JS 可编辑表格组件的使用	171
9.3	实现分页功能	174
9.3.1	使用 GridPanel 展现数据	174
9.3.2	为 store 配置 proxy 用于动态请求分页数据	175
9.3.3	使用 PagingToolbar 实现分页工具栏	175
9.4	实现分组功能	176
9.4.1	配置 GroupStore	176
9.4.2	配置 GroupView	176
9.5	表格组件扩充综合示例	179
9.6	小结	185
第 10 章	树形组件	186
10.1	树形组件介绍	186
10.2	如何使用树形组件	187
10.2.1	创建树的节点	187
10.2.2	定义单击事件	188
10.3	树形组件的扩充	190
10.3.1	可以多选的树形组件	190
10.3.2	可以显示更多信息的树形组件	191
10.3.3	可以拖动的树形组件	192
10.4	树形组件综合示例	194
10.5	小结	203
第 11 章	窗口组件	204
11.1	JavaScript 的弹出对话框	204
11.1.1	alert	204
11.1.2	prompt	205
11.1.3	confirm	205
11.1.4	open	206
11.2	Ext JS 的弹出对话框	208
11.2.1	MessageBox	208

11.2.2	各种对话框的使用方法	212
11.3	Ext.window 的使用方法	216
11.4	小结	218
第 12 章	工具栏与菜单栏组件	219
12.1	工具栏组件介绍	219
12.2	如何使用工具栏组件	222
12.2.1	添加按钮事件的工具栏	222
12.2.2	创建带分隔条的工具栏	223
12.3	菜单栏组件介绍	224
12.4	如何使用菜单栏组件	231
12.5	小结	232
第 13 章	AJAX 应用与数据交互	233
13.1	Ext.Ajax	233
13.2	Ext.data.Store	237
13.2.1	存储数据	237
13.2.2	更新数据	238
13.3	Ext.data.Reader	239
13.3.1	Ext.data.ArrayReader	240
13.3.2	Ext.data.JsonReader	242
13.3.3	Ext.data.XmlReader	243
13.4	小结	245
第 14 章	REST 介绍	246
14.1	基础知识概述	246
14.1.1	HTTP 简介	246
14.1.2	REST 简介	247
14.1.3	JSR 311 规范简介	247
14.1.4	Jersey 简介	248
14.2	REST 开发示例	248
14.2.1	资源类的命名规则	249
14.2.2	URI 接口组织规则	249
14.2.3	示例 URI 接口的定义	250
14.2.4	Resource 类的编写	250
14.2.5	统一异常处理类的编写	251
14.3	Jersey 项目的部署	252
14.3.1	包结构规范	252
14.3.2	部署文件 (web.xml) 设定说明及运行	252

14.4	Jersey 的常用 annotation 表	253
14.5	Jersey 的使用示例	254
14.6	小结	261
第 15 章	封装 Ext JS 提高开发速度	262
15.1	完整的 Ext JS 开发流程	262
15.1.1	定义画面的组成	262
15.1.2	定义数据模型	263
15.1.3	确定页面组件的相互引用	264
15.1.4	确定前后台的数据交互	265
15.2	设计通用的工具类	265
15.2.1	全局变量类 Global.js	266
15.2.2	工具类 Util.js	276
15.3	扩展 Ext JS 的组件功能	278
15.3.1	表单补丁类	279
15.3.2	表格补丁类	280
15.4	定义常见的应用场景	282
15.4.1	单笔 CRUD 场景	282
15.4.2	主从画面场景	298
15.4.3	树形结构场景	315
15.5	小结	323

第三篇 AJAX 整合实践

第 16 章	Ext3+REST+Spring2+ Hibernate3 整合方案	326
16.1	博客管理系统简介	326
16.1.1	博客管理系统功能介绍	326
16.1.2	架构设计	330
16.2	Spring 简介	331
16.2.1	Spring 的核心思想	332
16.2.2	Spring 的下载配置	333
16.3	Hibernate 简介	336
16.3.1	Hibernate 介绍	336
16.3.2	JPA 的关系映射	342
16.3.3	Ext JS、Jersey、Hibernate 与 Spring 的整合	344
16.4	博客管理系统示例	353
16.4.1	持久层模型设计	354
16.4.2	添加博客类别功能的实现	364
16.4.3	发布博客功能的实现	366

16.4.4	Resources 模型设计	367
16.4.5	管理博客类别的画面设计	370
16.4.6	发布博客内容的画面设计	377
16.4.7	浏览博客内容的画面设计	385
16.4.8	编写配置文件	386
16.4.9	程序演示	388
16.5	小结	391
第 17 章	工时统计管理系统	392
17.1	工时统计系统简介	392
17.1.1	工时统计系统功能简介	392
17.1.2	采用 REST 进行架构组合	393
17.2	持久层模型设计	400
17.2.1	设计数据模型	400
17.2.2	设计 DAO 模型	406
17.3	业务逻辑模型设计	410
17.3.1	工时大类业务逻辑	410
17.3.2	工时细分类业务逻辑	412
17.3.3	工时填写业务逻辑	415
17.3.4	工时细项填写业务逻辑	417
17.4	Resources 模型设计	420
17.4.1	工时大类 Resources	420
17.4.2	工时细分类 Resources	423
17.4.3	工时填写 Resources	425
17.5	前台画面设计	428
17.5.1	设计工时类别画面	429
17.5.2	设计工时填写画面	444
17.6	编写配置文件	454
17.7	运行演示程序	457
17.8	小结	460

第一篇 Ext JS 相关知识介绍

第 1 章 Ext JS 介绍

第 2 章 Ext JS 快速入门

第 1 章 Ext JS 介绍

在众多的 AJAX 框架中，最流行的应该是 Ext 了，它不但提供了 AJAX 的基本功能，更重要的是它提供了大量封装好的 UI 组件，在此基础上，开发人员可以很方便地使用这些组件来实现美观的画面。

本章将从 Ext JS 的起源讲起，讲解 Ext JS 的工作原理、发展趋势和应用示例，然后分别讲解 Ext JS 的核心技术及目前比较流行的 Ext JS 实现框架，并对此进行比较，目的是让读者真正明白什么是 Ext JS，为什么要用 Ext JS 以及使用 Ext JS 都要掌握哪些技术和框架才能帮助开发人员快速应用 Ext JS。

1.1 Ext JS 概述

Ext JS 整合了 XMLHttpRequest、JavaScript、DOM、CSS、XML 等众多技术，它采用 JavaScript 编写，因为它与后台技术无关，所以开发人员可以直接使用 Ext JS 编写前台页面，无须建立任何应用服务器。

1.1.1 Ext JS 的起源

Ext JS 最初的目的是对 Yahoo-UI 进行扩展，由开发人员 Jack Slocum 开发，在开发过程中，他参考了 Java Swing 的实现机制，并深受《Head First Design Patterns》这本书的影响。如果读者查看 Ext JS 的源代码，就会发现各种模式的大量应用。Jack Slocum 还对 Ext JS 的源代码进行了大量的封装，使得 JavaScript 具有了面向对象的特性。Ext JS 的发展历史如图 1.1 所示。

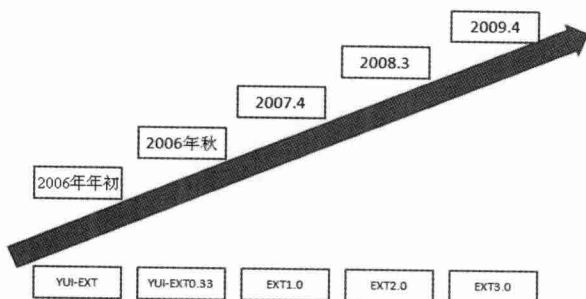


图 1.1 Ext JS 的发展历史

Ext JS 在设计之初就决定对跨浏览器进行支持，因此，开发人员在编写前台画面时，

不需要考虑具体的浏览器。Ext JS 支持的浏览器主要有：

- Windows Internet Explorer 6 及更高版本
- Mozilla Firefox 1.5 及更高版本
- Apple Safari 2 及更高版本
- Opera 9 及更高版本

前面讲过，Ext JS 建立在 Yahoo-UI 基础之上，并深受 Java Swing 实现机制和设计模式的影响，Ext JS 的构建图如图 1.2 所示。

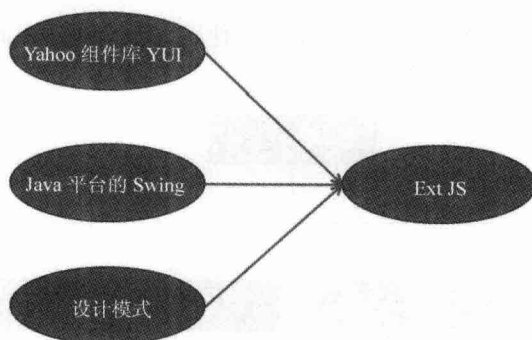


图 1.2 Ext JS 的构建图

在 Ext JS 中，首先要判断客户端使用的是哪种浏览器，然后根据不同的浏览器提供相应的支持，比如在 Ext JS 提供的 AJAX 中，会使用循环来创建 XMLHttpRequest 对象，代码如下所示：

```
//判断客户端使用的浏览器
isOpera = check(/opera/),
isChrome = check(/chrome/),
isWebKit = check(/webkit/),
isSafari = !isChrome && check(/safari/),
isSafari3 = isSafari && check(/version\/3/),
isSafari4 = isSafari && check(/version\/4/),
isIE = !isOpera && check(/msie/),
isIE7 = isIE && check(/msie 7/),
isIE8 = isIE && check(/msie 8/),
isGecko = !isWebKit && check(/gecko/),
isGecko3 = isGecko && check(/rv:1\.\9/),
isBorderBox = isIE && !isStrict,
isWindows = check(/windows|win32/),
isMac = check(/macintosh|mac os x/),
isAir = check(/adobeair/),
isLinux = check(/linux/),
isSecure = /^https/i.test(window.location.protocol);
//存储创建 XMLHttpRequest 的 active 对象
var activeX = ['MSXML2.XMLHTTP.3.0',
               'MSXML2.XMLHTTP',
               'Microsoft.XMLHTTP'];
function createXhrObject(transactionId) {
    var http;
    //首先通过 new XMLHttpRequest() 创建 XMLHttpRequest，如果异常，则循环创建。
    try {
        http = new XMLHttpRequest();
```

```
    } catch(e) {  
        for (var i = 0; i < activeX.length; ++i) {  
            try {  
                http = new ActiveXObject(activeX[i]);  
                break;  
            } catch(e) {}  
        }  
    } finally {  
        return {conn : http, tId : transactionId};  
    }  
}
```

Ext JS 除了使用自己的核心底层外，还可以使用其他常用的 AJAX 框架，比如 YUI、Dojo、jQuery、Prototype 等，如图 1.3 所示。

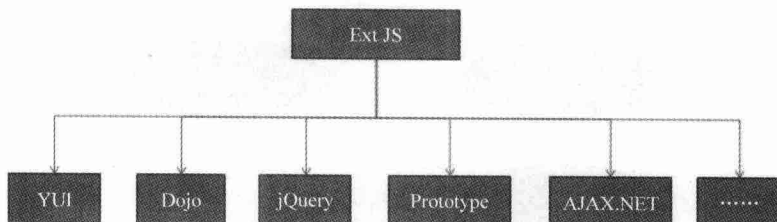


图 1.3 Ext JS 能够使用的核心底层

1.1.2 Ext JS 的许可协议

在使用开源框架 Ext JS 之前，一定要了解框架基于哪种许可协议条款。Ext JS 提供以下几个许可协议选项：

- 开放源码许可证：采用 Open Source LGPL 3.0 许可证的条款。如果打算在另一个开放源码项目或者个人、教育或非赢利项目中使用 Ext JS，则要使用这个许可证。
- 商用许可证：如果希望在商业项目中使用 Ext JS 时避免开发源码许可证的某些限制，或者由于内部原因必须拥有许可证，或者希望在商业上支持 Ext JS 的开发，这是最合适的许可证。
- 原始设备生产商（OEM）/转售商许可证：如果打算对 Ext JS 进行重新打包，或者作为软件开发库销售 Ext JS，这种许可证是最合适的。

1.1.3 Ext JS 的应用示例

前面讲解了 Ext JS 的基础知识，读者可能会迫不及待地想体验一下 Ext JS 的效果，下面我们就通过一些示例来展示 Ext JS 美轮美奂的效果，先来看一下分页组件在 Ext JS 中的应用，如图 1.4 所示。

上述分页组件的源代码如下所示：

```
Ext.onReady(function() {  
    // 创建一个 Store  
    var store = new Ext.data.JsonStore({  
        root: 'topics',  
        totalProperty: 'totalCount',
```

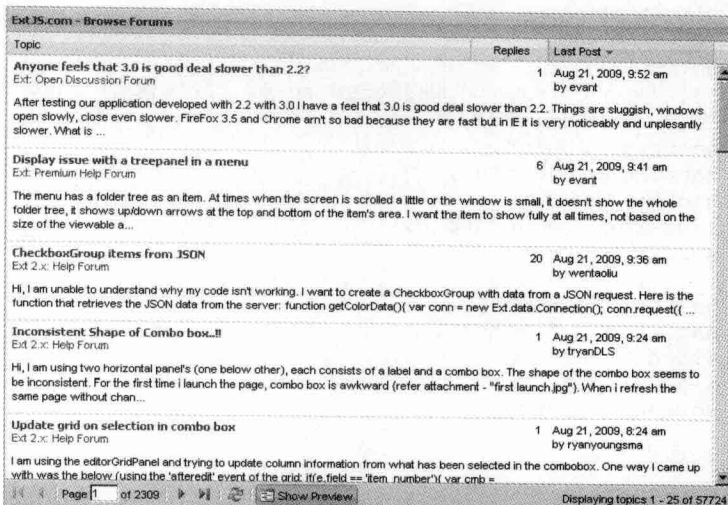


图 1.4 分页组件在 Ext JS 中的应用

```

idProperty: 'threadid',
remoteSort: true,
//定义要加载的数据域
fields: [
    'title', 'forumtitle', 'forumid', 'author',
    {name: 'replycount', type: 'int'},
    {name: 'lastpost', mapping: 'lastpost', type: 'date', dateFormat:
    'timestamp'},
    'lastposter', 'excerpt'
],
// 数据加载的方式和 url
proxy: new Ext.data.ScriptTagProxy({
    url: 'http://extjs.com/forum/topics-browse-remote.php'
});
store.setDefaultSort('lastpost', 'desc');
// 表示格式化要显示的内容
function renderTopic(value, p, record){
    return String.format(
        '<b><a href="http://extjs.com/forum/showthread.php?t={2}"
target="_blank">{0}</a></b><a href="http://extjs.com/forum/forumdisplay.
php?f={3}" target="_blank">{1} Forum</a>',
        value, record.data.forumtitle, record.id, record.data.forumid);
    }
    function renderLast(value, p, r){
        return String.format('{0}<br/>by {1}', value.dateFormat('M j, Y, g:i
a'), r.data['lastposter']);
    }
    //创建一个表格组件
    var grid = new Ext.grid.GridPanel({
        el: 'topic-grid',
        width: 700,
        height: 500,
        title: 'ExtJS.com - Browse Forums',
        store: store,
        trackMouseOver: false,
        disableSelection: true,

```