

## <<Ext JS权威指南>>

### 图书基本信息

书名：<<Ext JS权威指南>>

13位ISBN编号：9787111380634

10位ISBN编号：7111380630

出版时间：2012-6

出版时间：机械工业出版社

作者：黄灯桥

页数：980

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>

## &lt;&lt;Ext JS权威指南&gt;&gt;

## 前言

近一段时间，随着HTML 5和CSS 3的逐步升温，浏览器大战进行得如火如荼，而开发人员最关心的是JavaScript框架在HTML 5和CSS 3的大潮下会走向何方。

随着iPhone 4S的出现，手机大战也进行得如火如荼。

目前的情况是，不但手机系统进行着混战，手机应用的框架也在进行混战。

应该说Ext公司在这方面触觉是很敏锐的，它选择合并，将公司更改为Sencha，并通过Sencha加入到手机框架大战中。

本以为Ext JS会止步于Ext JS 3.3，想不到Sencha在手机框架中站稳脚跟后，2011年又推出了Ext JS 4.0，对这个颇受欢迎的框架进行了更多的革新，加入了很多HTML 5和CSS 3的元素。

在过渡到HTML 5和CSS 3的时期，Ext JS框架能继续发展，对开发人员来说是天大的喜事。

因为在HTML 5和CSS 3还没完全一统江湖的时候，开发人员要继续使用目前的开发工具进行开发。

而Ext JS 3暴露出来的问题也需要不断修正和更新，这正是Ext JS 4需要解决的。

很不错，Ext JS 4终于来了。

为什么写这本书 从Ext JS 2.0开始，尤其是Ext JS 3.0推出后，使用Ext JS的开发人员越来越多。

而随着Ext JS 4的推出，估计会掀起一个新的技术学习热潮。

Ext JS 4进行了架构的调整，做了一系列的规范化，例如类名的规范化、UI组件的渲染的规范化等，还重构了不少UI组件，这些对熟悉Ext JS 3的开发人员来说都要重新学习。

对于新入门的读者，更是急需Ext JS 4方面的书籍来指导。

因此，笔者决定写一本介绍Ext JS 4的书。

目前市面上介绍Ext JS的图书都是以应用为主，深入介绍Ext JS的很少，这也是笔者写本书的一个缘由。

在本书中，不但介绍了如何使用Ext JS 4，还深入Ext JS 4的源代码，通过抽丝剥茧的分析，让大家对Ext JS 4有更深入的理解，从而使开发人员在使用Ext JS 4时少走弯路。

本书除了深入介绍Ext JS 4外，还有一个主要目的，就是希望大家能掌握Firebug这个工具，学会如何使用Firebug去分析JavaScript代码的运行过程，提高分析能力及调试水平，减少bug的发生和调试时间。本书的特色 笔者在Ext JS的学习和交流中发现，很多使用者对Ext JS的整体架构不太熟悉，造成对应用中的一些问题混淆不清，从而导致应用中出现或多或少的错误。

例如，对Ext JS的三层架构不熟悉，造成在UI中找数据的问题。

而目前各类Ext JS书籍中很少提及这些方面，所以，针对使用者普遍存在的问题和容易混淆的地方，本书通过原理讲解和示例讲述“为什么是这样的”以及“为什么要这样做”。

本书的主要特色就是通过源代码的讲解，让读者明白Ext JS的类是如何运作的，然后通过运行原理及实战来学习如何使用这些类，并循序渐进地学习Ext JS。

本书面向的读者 本书介绍了Ext JS 4中的几乎所有功能，并对其源代码进行了分析和讲解。

因此，读者不但可以学习如何使用Ext JS 4，还可以通过阅读源代码加深对Ext JS的理解，并从源代码中吸取别人的经验，提高自己的脚本编写能力。

只要读者具备Web编程的基础，阅读本书的内容就不会有什么问题。

如果读者是新手，就应根据第1章关于学习Ext JS必需的基础知识一节的建议，去阅读相应的书籍，掌握Web编程的基础知识，尤其是JavaScript、HTML和CSS的相关知识。

当然，能了解面向对象、三层架构等知识对阅读本书是相当有帮助的。

如果读者对Ext JS有一定基础，则可通过本书深入了解Ext JS的运行原理，加深对Ext JS的认识，尤其是对Ext JS新的开发框架的认识。

本书示例中的服务器端代码，使用C#和Java两种语言编写，所以只要熟悉这两种语言中的一种，对阅读本书都会有很大帮助。

不过，Ext JS是一个基于客户端的JavaScript框架，对于使用何种服务器端开发语言没有要求，因此即使是不懂C#和Java语言的读者，阅读本书也不会有太多困难。

使用本书的要求 在使用本书的示例时，最好安装Firefox 4.0以上版本，并且安装Firebug扩展。

## <<Ext JS权威指南>>

要运行本书中带有服务器端代码的示例，需要： IIS或Tomcat 7.0 .Net：Visual Studio 2010 Java：Eclipse Helios Service Release 2或Spket 1.6.18 SQL Server 2005 本书资源包请登录华章网站（[www.hzbook.com](http://www.hzbook.com)）下载。

如何阅读本书 本书是采用循序渐进的方式来介绍Ext JS 4的，各章之间会有一定的关联，因此建议读者按章节的编排顺序逐章阅读。

本书中有些示例是在Firebug中示范的，所以最好在自己的机器上安装好Firefox和Firebug，以备使用。这样做的目的是让大家在阅读过程中熟悉Firebug的一些调试技巧，这些技巧在调试Ext JS时是相当有用的。

例如，有初学者曾咨询为什么在Grid中显示不了数据，查了很久都查不到原因，让笔者帮忙看一下代码，笔者让他先用Firebug查看Store是否已提交数据请求，在Firebug网络面板中，他看到请求已发送，但是返回的是服务器端代码错误，最后查到仅仅是“将公有方法设置成私有方法”这样简单的修正。在很多时候，就是这样的错误造成大麻烦，所以笔者建议多使用Firebug，好的工具可以事半功倍。

本书秉承实践是最好的老师的精神，立足于“自己动手，丰衣足食”的原则，因而希望读者在阅读本书的时候，能亲自动手去实践一下。

Let's do it!

## &lt;&lt;Ext JS权威指南&gt;&gt;

## 内容概要

本书有两大特点：第一，授人以鱼，根据Ext JS的构成体系全面系统地讲解了其开发方法与技巧，每个知识点都辅之以翔实的案例，同时包含大量最佳实践，适合系统学习和开发参考；第二，授人以渔，宏观上对Ext JS的整体架构进行了分析，微观上则通过源代码深刻揭示了Ext JS的工作机制与原理，对于想了解Ext JS工作原理和在开发中碰到疑难问题的读者尤为有帮助。

全书一共22章：第1章简要介绍了学习Ext JS必备的基础知识、JSON、Ext JS 4的新特性，以及其开发工具的获取、安装与配置；第2章介绍了Ext JS 4的获取、Ext JS库的配置与使用、语法、本地化，以及一个经典的入门示例；第3章详细讲解了调试的工具及技巧，这是本书的重要内容，希望所有Web开发者都能掌握；第4章全面介绍了Ext JS的基础架构；第5~9章分别讲解了Ext JS的事件及其应用、选择器与DOM操作、数据交互、模板与组件、容器、面板、布局与视图；第10章和第11章分别详细介绍了重构后的Gird和与Gird同源的树；第12~16章分别讲解了表单、窗口、按钮、菜单、工具条、图形、图表，以及其他组件和实用功能；第17~19章分别介绍了Ext.Direct、动画功能和拖放功能；第20~22章则分别讲解了扩展与插件、主题开发、MVC应用的架构。

## <<Ext JS权威指南>>

### 作者简介

黄灯桥（网名：上将军）资深Web技术专家，拥有10余年工作经验，一直致力于Web开发框架的研究，近年来醉心于Ext JS框架的开发与推广，撰写了大量关于Ext JS开发相关的教程，是社区公认的先锋和布道者，与Sencha和Ext JS的官方开发团队有十分紧密的联系。著有经典著作《JavaScript凌厉开发——Ext详解与实践》和《Ext JS高级程序设计》。

他不仅精通VB、Delphi和C++等传统的面向对象技术，而且还擅长ASP.NET、PHP和Java等Web开发技术。目前重点关注Sencha Touch、jQuery UI、HTML 5、CSS 3以及NoSQL数据库等前沿技术。活跃于CSDN（博客专家）等各大技术社区，在社区内知名度极高，深受网友欢迎。

## <<Ext JS权威指南>>

### 书籍目录

#### 前言

#### 第1章 Ext JS 4开发入门

##### 1.1 学习Ext JS必需的基础知识

##### 1.2 JSON概述

###### 1.2.1 认识JSON

###### 1.2.2 JSON的结构

###### 1.2.3 JSON的例子

###### 1.2.4 在JavaScript中使用JSON

###### 1.2.5 在.NET中使用JSON

###### 1.2.6 在Java中使用JSON

###### 1.2.7 更多有关JSON的信息

##### 1.3 Ext JS 4概述

##### 1.4 Ext JS的开发工具的获取、安装与配置介绍

###### 1.4.1 Ext Designer

###### 1.4.2 在Visual Studio中实现智能提示

###### 1.4.3 Spket

###### 1.4.4 在Eclipse中实现智能提示

##### 1.5 如何获得帮助

##### 1.6 本章小结

#### 第2章 从“Hello World”开始

##### 2.1 获取Ext JS 4

##### 2.2 配置使用Ext JS库

##### 2.3 编写“Hello World”程序

##### 2.4 关于Ext.onReady

##### 2.5 关于Ext.BLANK\_IMAGE\_URL

##### 2.6 关于字体

##### 2.7 Ext JS 4语法

##### 2.8 本地化

##### 2.9 为本书示例准备一个模板

##### 2.10 本章小结

#### 第3章 调试工具及技巧

##### 3.1 使用Firebug进行调试

##### 3.2 在IE中调试

###### 3.2.1 使用Debugbar和Companion.js调试

###### 3.2.2 使用IETester测试

###### 3.2.3 在IE 8或IE 9中调试

##### 3.3 在Chrome中调试

##### 3.4 调试工具的总结

##### 3.5 调试技巧

##### 3.6 本章小结

#### 第4章 Ext JS的基础架构

##### 4.1 Ext JS框架的命名空间：Ext

###### 4.1.1 概述

###### 4.1.2 apply和applyIf方法

###### 4.1.3 不推荐的extend方法

## &lt;&lt;Ext JS权威指南&gt;&gt;

- 4.1.4 数据及其类型检测
- 4.1.5 其他的基础方法
- 4.2 为框架顺利运行提供支持
  - 4.2.1 平台检测工具：Ext.is
  - 4.2.2 当前运行环境检测工具：Ext.supports
- 4.3 Ext JS的静态方法
  - 4.3.1 概述
  - 4.3.2 Ext.Object中的静态方法
  - 4.3.3 Ext.Function中的静态方法
  - 4.3.4 Ext.Array中的静态方法
  - 4.3.5 Ext.Error中的静态方法
- 4.4 深入了解类的创建及管理
  - 4.4.1 开始创建类
  - 4.4.2 创建类的类：Ext.Class
  - 4.4.3 所有继承类的基类：Ext.Base
  - 4.4.4 实现动态加载：Ext.Loader
  - 4.4.5 管理类的类：Ext.ClassManager
  - 4.4.6 类创建的总结
- 4.5 动态加载的路径设置
- 4.6 综合实例：页面计算器
- 4.7 本章小结
- 第5章 Ext JS的事件及其应用
  - 5.1 概述
  - 5.2 浏览器事件
    - 5.2.1 绑定浏览器事件的过程：Ext.EventManager
    - 5.2.2 封装浏览器事件：Ext.EventObject
    - 5.2.3 移除浏览器事件
  - 5.3 内部事件
    - 5.3.1 内部事件对象：Ext.util.Event
    - 5.3.2 为组件添加事件接口：Ext.util.Observable
    - 5.3.3 为组件绑定事件
    - 5.3.4 内部事件的触发过程
    - 5.3.5 移除事件
  - 5.4 特定功能的事件对象
    - 5.4.1 延时任务：Ext.util.DelayedTask
    - 5.4.2 一般任务：Ext.util.TaskRunner与Ext.TaskManager
    - 5.4.3 封装好的单击事件：Ext.util.ClickRepeater
  - 5.5 键盘事件
    - 5.5.1 为元素绑定键盘事件：Ext.util.KeyMap
    - 5.5.2 键盘导航：Ext.util.KeyNav
  - 5.6 综合实例：股票数据的实时更新
  - 5.7 本章小结
- 第6章 选择器与DOM操作
  - 6.1 Ext JS的选择器：Ext.DomQuery
    - 6.1.1 选择器的作用
    - 6.1.2 使用Ext.query选择页面元素
    - 6.1.3 基本选择符

## <<Ext JS权威指南>>

- 6.1.4 属性选择符
- 6.1.5 CSS属性值选择符
- 6.1.6 伪类选择符
- 6.1.7 扩展选择器
- 6.1.8 Ext.DomQuery的使用方法
- 6.1.9 Ext JS选择器的总结
- 6.2 获取单一元素：Ext.dom.Element
  - 6.2.1 从错误开始
  - 6.2.2 使用Ext.get获取元素
  - 6.2.3 使用Ext.fly获取元素
  - 6.2.4 使用Ext.getDom获取元素
  - 6.2.5 获取元素的总结
- 6.3 元素生成器：Ext.dom.Helper
  - 6.3.1 概述
  - 6.3.2 使用createHtml或markup方法生成HTML代码
  - 6.3.3 使用createDOM方法生成DOM对象
  - 6.3.4 使用createTemplate方法创建模板
  - 6.3.5 Helper对象的使用方法
- 6.4 元素的操作
- 6.5 获取元素集合：Ext.CompositeElementLite与Ext.CompositeElement
  - 6.5.1 使用Ext.select获取元素集合
  - 6.5.2 Ext.dom.CompositeElement与Ext.dom.CompositeElementLite的区别
  - 6.5.3 操作元素集合
- 6.6 综合实例：可折叠的面板Accordion
- 6.7 本章小结
- 第7章 数据交互
  - 7.1 数据交互基础
    - 7.1.1 Ajax概述
    - 7.1.2 封装Ajax：Ext.data.Connection与Ext.Ajax
    - 7.1.3 使用Ajax
    - 7.1.4 跨域获取数据：Ext.data.JsonP
    - 7.1.5 为Element对象提供加载功能：Ext.ElementLoader
    - 7.1.6 为组件提供加载功能：Ext.ComponentLoader
  - 7.2 代理
    - 7.2.1 代理概述
    - 7.2.2 基本的代理：Ext.data.proxy.Proxy
    - 7.2.3 进行批量操作：Ext.data.Batch与Ext.data.Operation
    - 7.2.4 服务器端代理：Ext.data.proxy.Server
    - 7.2.5 使用Ajax处理数据的代理：Ext.data.proxy.Ajax与Ext.data.proxy.Rest
    - 7.2.6 跨域处理数据的代理：Ext.data.proxy.JsonP
    - 7.2.7 为Ext.Direct服务的代理：Ext.data.proxy.Direct
    - 7.2.8 客户端代理：Ext.data.proxy.Client
    - 7.2.9 从变量中提取数据的代理：Ext.data.proxy.Memory
    - 7.2.10 使用浏览器存储的代理：Ext.data.WebStorageProxy、Ext.data.SessionStorageProxy和Ext.data.proxy.LocalStorage
  - 7.3 读取和格式化数据
    - 7.3.1 概述



## <<Ext JS权威指南>>

- 7.3.2 数据的转换过程：Ext.data.reader.Xml、Ext.data.reader.Json和Ext.data.reader.Array
- 7.3.3 Reader对象的配置项
- 7.3.4 格式化提交数据：Ext.data.writer.Writer、Ext.data.writer.JSON和Ext.data.writer.Xml
- 7.3.5 Writer对象的配置项
- 7.4 数据模型
  - 7.4.1 概述
  - 7.4.2 数据类型及排序类型：Ext.data.Types与Ext.data.SortTypes
  - 7.4.3 数据模型的骨架——字段：Ext.data.Field
  - 7.4.4 数据集：Ext.util.AbstractMixedCollection与Ext.util.MixedCollection
  - 7.4.5 数据验证及错误处理：Ext.data.validations与Ext.data.Errors
  - 7.4.6 模型的关系：Ext.data.Association、Ext.data.HasManyAssociation和Ext.data.BelongsToAssociation
  - 7.4.7 管理数据模型：Ext.AbstractManager与Ext.ModelManager
  - 7.4.8 定义数据模型：Ext.data.Model
  - 7.4.9 数据模型的定义过程
  - 7.4.10 数据模型的创建
  - 7.4.11 数据模型的配置项、属性和方法
- 7.5 Store
  - 7.5.1 概述
  - 7.5.2 Store对象的实例化过程
  - 7.5.3 TreeStore对象的实例化过程
  - 7.5.4 Ext.data.Store加载数据的方法
  - 7.5.5 Ext.data.TreeStore加载数据的方法
  - 7.5.6 Store的配置项
  - 7.5.7 Store的分页
  - 7.5.8 Store的排序：Ext.util.Sorter与Ext.util.Sortable
  - 7.5.9 Store的过滤：Ext.util.Filter
  - 7.5.10 Store的分组：Ext.util.Grouper
  - 7.5.11 树节点：Ext.data.NodeInterface与Ext.data.Tree
  - 7.5.12 Store的方法
  - 7.5.13 Store的事件
  - 7.5.14 Store管理器：Ext.data.StoreManager
- 7.6 综合实例
  - 7.6.1 远程读取JSON数据
  - 7.6.2 读取XML数据
  - 7.6.3 Store的数据操作
- 7.7 本章小结
- 第8章 模板与组件基础
  - 8.1 模板
    - 8.1.1 模板概述
    - 8.1.2 Ext.Template的创建与编译
    - 8.1.3 格式化输出数据：Ext.String、Ext.Number、Ext.Date和Ext.util.Format
    - 8.1.4 超级模板：Ext.XTemplate（包括Ext.XTemplateParser和Ext.XTemplateCompiler）
    - 8.1.5 模板的方法
  - 8.2 组件的基础知识
    - 8.2.1 概述
    - 8.2.2 组件类的整体架构

## <<Ext JS权威指南>>

- 8.2.3 布局类的整体架构
- 8.2.4 组件的创建流程
- 8.2.5 常用的组件配置项、属性、方法和事件
- 8.3 为组件添加功能
  - 8.3.1 为元素添加阴影：Ext.Shadow与Ext.ShadowPool
  - 8.3.2 为组件提供阴影和shim功能：Ext.Layer
  - 8.3.3 让组件实现浮动功能：Ext.util.Floating
  - 8.3.4 记录组件状态：Ext.state.Stateful
  - 8.3.5 实现调整大小功能：Ext.resizer.Resizer与Ext.resizer.ResizeTracker
  - 8.3.6 为组件提供拖动功能：Ext.util.ComponentDragger
  - 8.3.7 为组件实现动画功能：Ext.util.Animate
  - 8.3.8 其他的组件辅助功能类
- 8.4 组件的管理
  - 8.4.1 组件管理及查询：Ext.ComponentManager与Ext.ComponentQuery
  - 8.4.2 焦点管理：Ext.FocusManager
  - 8.4.3 z-order管理：Ext.ZindexManager与Ext.WindowManager
  - 8.4.4 状态管理：Ext.state.Manager、Ext.state.Provider、Ext.state.Local-StorageProvider和Ext.state.CookieProvider
- 8.5 综合实例
  - 8.5.1 使用子模板
  - 8.5.2 递归调用模板
- 8.6 本章小结
- 第9章 容器、面板、布局和视图
  - 9.1 容器与布局的关系
  - 9.2 容器
    - 9.2.1 容器的创建过程：Ext.container.AbstractContainer与Ext.container.Container
    - 9.2.2 Ext.container.AbstractContainer和Ext.container.Container的配置项、属性、方法和事件
    - 9.2.3 将body元素作为容器：Ext.container.Viewport
  - 9.3 面板
    - 9.3.1 面板的结构
    - 9.3.2 构件的放置：dockedItems
    - 9.3.3 面板标题栏构件：Ext.panel.Header与Ext.panel.Tool
    - 9.3.4 记录和恢复面板属性：Ext.util.Memento
    - 9.3.5 面板常用的配置项、方法和事件
  - 9.4 布局
    - 9.4.1 布局概述
    - 9.4.2 布局的运行流程：Ext.layout.Layout
    - 9.4.3 容器类布局基类：Ext.layout.container.Container
    - 9.4.4 盒子布局、垂直布局与水平布局：Ext.layout.container.Box、Ext.layout.container.VBox与Ext.layout.container.HBox
    - 9.4.5 为盒子模型提供调整大小的功能：Ext.resizer.Splitter
    - 9.4.6 手风琴布局：Ext.layout.container.Accordion
    - 9.4.7 锚固布局：Ext.layout.container.Anchor
    - 9.4.8 绝对定位布局：Ext.layout.container.Absolute
    - 9.4.9 边框布局：Ext.layout.container.Border
    - 9.4.10 自动布局：Ext.layout.container.Auto

## <<Ext JS权威指南>>

- 9.4.11 表格布局：Ext.layout.container.Table
- 9.4.12 列布局：Ext.layout.container.Column
- 9.4.13 自适应布局：Ext.layout.container.AbstractFit与Ext.layout.container.Fit
- 9.4.14 卡片布局：Ext.layout.container.AbstractCard与Ext.layout.container.Card
- 9.5 标签面板
  - 9.5.1 标签面板的构成及其运行流程：Ext.tab.Panel、Ext.tab.Bar与Ext.tab.Tab
  - 9.5.2 标签面板的配置项、属性、方法和事件
  - 9.5.3 使用标签页
  - 9.5.4 可重用的标签页
- 9.6 视图与选择模型
  - 9.6.1 视图与选择模型概述
  - 9.6.2 视图的运行流程：Ext.view.AbstractView与Ext.view.View
  - 9.6.3 选择模型的工作流程
  - 9.6.4 选择模型的配置项、属性、方法和事件
  - 9.6.5 视图的配置项、属性、方法和事件
  - 9.6.6 使用视图
- 9.7 页面布局设计
- 9.8 综合实例
  - 9.8.1 布局设计实例：仿Eclipse界面
  - 9.8.2 在单页面应用中使用卡片布局实现“页面”切换
- 9.9 本章小结
- 第10章 重构后的Grid
  - 10.1 Grid的基类及其构成
    - 10.1.1 概述
    - 10.1.2 表格面板的运行流程：Ext.panel.Table
    - 10.1.3 表格视图的运行流程：Ext.view.Table与Ext.view.TableChunker
    - 10.1.4 列标题容器的运行流程：Ext.grid.header.Container
    - 10.1.5 列标题的运行流程：Ext.grid.column.Column
    - 10.1.6 虚拟滚动条的工作原理：Ext.grid.PagingScroller
    - 10.1.7 锁定列的运行流程：Ext.grid.Lockable与Ext.grid.LockingView
  - 10.2 使用Grid
    - 10.2.1 最简单的Grid
    - 10.2.2 列的配置项
    - 10.2.3 自定义单元格的显示格式
    - 10.2.4 通过列对象定义单元格的显示格式
    - 10.2.5 设置行的背景颜色
    - 10.2.6 列标题的分组
    - 10.2.7 使用锁定列
    - 10.2.8 Grid的配置项、属性、方法和事件
  - 10.3 Grid的附加功能
    - 10.3.1 概述
    - 10.3.2 附加功能基类：Ext.grid.feature.Feature
    - 10.3.3 为行添加附加信息：Ext.grid.feature.RowBody
    - 10.3.4 数据汇总功能：Ext.grid.feature.AbstractSummary与Ext.grid.feature.Summary
    - 10.3.5 分组功能：Ext.grid.feature.Grouping
    - 10.3.6 分组汇总功能：Ext.grid.feature.GroupingSummary

## <<Ext JS权威指南>>

### 10.4 可编辑的Grid

#### 10.4.1 概述

#### 10.4.2 Grid实现可编辑功能的运行流程：Ext.grid.plugin.Editing

#### 10.4.3 单元格编辑的运行流程：Ext.grid.plugin.CellEditing、Ext.grid.CellEditor与Ext.Editor

#### 10.4.4 行编辑的运行流程：Ext.grid.plugin.RowEditing与Ext.grid.RowEditor

#### 10.4.5 在Grid中使用单元格编辑模式

#### 10.4.6 在Grid中使用行编辑模式

#### 10.4.7 Grid编辑插件的配置项、属性、方法和事件

### 10.5 关于列表视图：ListView

### 10.6 属性Grid

#### 10.6.1 概述

#### 10.6.2 使用属性Grid

#### 10.6.3 自定义编辑组件

#### 10.6.4 PropertyGrid的配置项、属性、方法和事件

### 10.7 综合实例

#### 10.7.1 使用不同选择模型的Grid以及设置默认选择行

#### 10.7.2 Grid的本地排序和过滤

#### 10.7.3 使用分页工具条（PagingToolbar）实现远程分页、排序和过滤

#### 10.7.4 使用分页滚动条（PagingScroller）实现远程分页、排序和过滤

#### 10.7.5 使用CellEditing实现数据的增删改

#### 10.7.6 使用RowEditing实现数据的增删改

#### 10.7.7 主从表的显示

### 10.8 本章小结

## 第11章 与Grid同源的树

### 11.1 树的构成

#### 11.1.1 概述

#### 11.1.2 树面板的运行流程：Ext.tree.Panel

#### 11.1.3 TreeStore的运行流程：Ext.data.TreeStore

#### 11.1.4 TreeColumn的运行流程：Ext.tree.Column

#### 11.1.5 视图的运行流程：Ext.tree.View与Ext.data.NodeStore

#### 11.1.6 树的选择模型：Ext.selection.TreeModel

### 11.2 树的使用

#### 11.2.1 一个最简单的树

#### 11.2.2 树节点的默认字段

#### 11.2.3 为树节点添加附加字段

#### 11.2.4 显示多列数据（TreeGrid效果）

#### 11.2.5 在树中使用复选框

#### 11.2.6 树的配置项、属性、方法和事件

### 11.3 综合实例

#### 11.3.1 树的远程加载

#### 11.3.2 树的动态加载及节点维护

#### 11.3.3 XML树及节点维护

#### 11.3.4 使用树动态控制Grid的显示

### 11.4 本章小结

## 第12章 表单

### 12.1 表单的构成及操作

#### 12.1.1 表单面板的运行流程：Ext.form.Panel与Ext.form.FieldAncestor

## <<Ext JS权威指南>>

- 12.1.2 表单面板的配置项、属性、方法和事件
- 12.1.3 表单的管理：Ext.form.Basic
- 12.1.4 BasicForm的配置项、属性、方法和事件
- 12.1.5 表单的操作：Ext.form.action.Action
- 12.1.6 加载操作的运行流程：Ext.form.action.Load与Ext.form.action.DirectLoad
- 12.1.7 提交操作的运行流程：Ext.form.action.Submit、Ext.form.action.DirectSubmit与Ext.form.action.StandardSubmit
- 12.1.8 字段的构成
- 12.1.9 BaseField的配置项、属性、方法和事件
- 12.1.10 常用的验证函数：Ext.form.field.VTypes
- 12.2 使用字段
- 12.3 使用Trigger类字段
  - 12.3.1 具有单击功能的字段：Ext.form.field.Trigger
  - 12.3.2 实现微调功能的Spinner字段
  - 12.3.3 使用NumberField字段
  - 12.3.4 下拉选择类字段的基类：Ext.form.field.Picker
  - 12.3.5 使用DateField字段
  - 12.3.6 使用时间Field字段
- 12.4 使用ComboBox字段
  - 12.4.1 概述
  - 12.4.2 BoundList对象的运行流程
  - 12.4.3 ComboBox字段的配置项、属性、方法和事件
  - 12.4.4 最简单的ComboBox
  - 12.4.5 自定义列表显示格式的ComboBox
  - 12.4.6 动态调整ComboBox的列表数据
  - 12.4.7 实现ComboBox的联动
  - 12.4.8 使用ComboBox的查询功能
  - 12.4.9 设置ComboBox的默认值
- 12.5 表单的验证和加载数据
  - 12.5.1 表单的验证及错误显示方式
  - 12.5.2 为表单加载数据
- 12.6 在表单中使用布局
  - 12.6.1 分列显示表单的字段
  - 12.6.2 使用Fieldset作为列容器
  - 12.6.3 使用两列布局加HtmlEditor的表单
  - 12.6.4 在表单中使用标签页
- 12.7 综合实例：实现Products表的管理功能
- 12.8 本章小结
- 第13章 窗口
  - 13.1 窗口：Ext.window.Window.
    - 13.1.1 窗口的构成
    - 13.1.2 窗口的配置项、属性、方法和事件
    - 13.1.3 使用窗口
    - 13.1.4 在窗口内使用布局
  - 13.2 信息提示窗口：Ext.window.MessageBox
    - 13.2.1 概述
    - 13.2.2 信息提示窗口的构成

## <<Ext JS权威指南>>

- 13.2.3 使用信息提示窗口
- 13.2.4 信息提示窗口按钮的本地化
- 13.2.5 使用alert方法
- 13.2.6 使用confirm方法
- 13.2.7 使用progress方法
- 13.2.8 使用prompt方法
- 13.2.9 使用wait方法
- 13.2.10 使用信息提示窗口要注意的问题
- 13.3 综合实例：实现登录窗口
- 13.4 本章小结
- 第14章 按钮、菜单与工具条
- 14.1 按钮
  - 14.1.1 按钮的构成：Ext.button.Button
  - 14.1.2 按钮的配置项、属性、方法和事件
  - 14.1.3 使用按钮
  - 14.1.4 带分割线的按钮：Ext.button.Split
  - 14.1.5 多状态按钮：Ext.button.Cycle
  - 14.1.6 按钮组：Ext.container.ButtonGroup
- 14.2 菜单及菜单项
  - 14.2.1 Ext JS的菜单（Menu对象）是什么
  - 14.2.2 菜单管理器：Ext.menu.Manager
  - 14.2.3 菜单项：Ext.menu.Item
  - 14.2.4 可复选的菜单项：Ext.menu.CheckItem
  - 14.2.5 菜单分隔条：Ext.menu.Separator
  - 14.2.6 颜色选择器菜单：Ext.menu.ColorPicker
  - 14.2.7 日期选择菜单：Ext.menu.DatePicker
  - 14.2.8 使用菜单
- 14.3 工具栏及工具栏组件
  - 14.3.1 工具栏：Ext.toolbar.Toolbar
  - 14.3.2 非交互式工具栏条目的基类：Ext.toolbar.Item
  - 14.3.3 文本项：Ext.toolbar.TextItem
  - 14.3.4 填充项：Ext.toolbar.Fill
  - 14.3.5 工具栏分隔条：Ext.toolbar.Separator
  - 14.3.6 空白项：Ext.toolbar.Spacer
  - 14.3.7 分页工具栏：Ext.toolbar.Paging
  - 14.3.8 使用工具栏
- 14.4 使用Ext.Action
  - 14.4.1 概述
  - 14.4.2 Action对象配置项和方法
  - 14.4.3 使用示例
- 14.5 综合实例：在Grid中使用右键菜单
- 14.6 本章小结
- 第15章 图形与图表
- 15.1 基础知识
  - 15.1.1 SVG简介
  - 15.1.2 VML简介
- 15.2 图形介绍

## <<Ext JS权威指南>>

- 15.2.1 概述
- 15.2.2 画布的工作流程：Ext.draw.Component
- 15.2.3 图形引擎及接口：Ext.draw.Surface、Ext.draw.engine.Svg和Ext.draw.engine.Vml
- 15.2.4 画笔：Ext.draw.Sprite
- 15.2.5 图层：Ext.draw.CompositeSprite
- 15.2.6 调色板：Ext.draw.Color
- 15.2.7 辅助对象：Ext.draw.Draw与Ext.draw.Matrix
- 15.3 使用图形功能
  - 15.3.1 简单的开始
  - 15.3.2 DrawComponent对象的配置项、属性、方法和事件
  - 15.3.3 Surface对象的配置项、属性、方法和事件
  - 15.3.4 DrawSprite对象的配置项、属性、方法和事件
  - 15.3.5 CompositeSprite对象的配置项、属性、方法和事件
  - 15.3.6 使用基本图形
  - 15.3.7 使用图片
  - 15.3.8 使用路径
  - 15.3.9 移动、旋转和缩放图形
  - 15.3.10 使用渐变效果
  - 15.3.11 使用图层
- 15.4 图表介绍
  - 15.4.1 概述
  - 15.4.2 图表的工作流程
- 15.5 使用图表
  - 15.5.1 从一个简单例子开始
  - 15.5.2 坐标轴的配置项
  - 15.5.3 Series对象的配置项、属性、方法和事件
  - 15.5.4 折线图的配置项
  - 15.5.5 显示多个折线图及使用图例
  - 15.5.6 使用面积图
  - 15.5.7 简单条形图（Bar和Column Chart）及使用标签（Label对象）
  - 15.5.8 堆积条形图
  - 15.5.9 分组条形图
  - 15.5.10 自定义条形颜色
  - 15.5.11 使用散点图
  - 15.5.12 使用饼图
  - 15.5.13 自定义饼块颜色
  - 15.5.14 使用表盘图
  - 15.5.15 使用雷达图
  - 15.5.16 使用时间轴
  - 15.5.17 实现实时动态的图表
  - 15.5.18 使用组合图
  - 15.5.19 在图表中使用背景
  - 15.5.20 在图表中自定义主题
- 15.6 本章小结
- 第16章 其他组件及实用功能
  - 16.1 其他组件
  - 16.2 使用滑块



## <<Ext JS权威指南>>

- 16.3 使用提示信息
- 16.4 实用功能
- 16.5 本章小结
- 第17章 可简化通信的Ext.Direct
  - 17.1 准备工作
  - 17.2 Ext.Direct的工作原理及构成
    - 17.2.1 工作原理
    - 17.2.2 Ext.Direct的构成
    - 17.2.3 RemotingProvider对象的具体工作流程
    - 17.2.4 PollingProvider对象的具体工作流程
  - 17.3 配置Ext.Direct的使用环境
    - 17.3.1 概述
    - 17.3.2 .NET环境的配置
    - 17.3.3 Java环境的配置
  - 17.4 使用Ext.Direct
    - 17.4.1 概述
    - 17.4.2 使用DirectProxy及进行CURD操作
    - 17.4.3 使用Ext.Direct实现树的动态加载及节点维护
    - 17.4.4 使用DirectLoad为表单加载数据
    - 17.4.5 使用DirectSubmit提交表单及使用Session
    - 17.4.6 使用Ext.Direct上传文件
    - 17.4.7 使用PollingProvider对象
    - 17.4.8 错误处理
  - 17.5 本章小结
- 第18章 动画功能
  - 18.1 动画功能的构成及工作流程
    - 18.1.1 概述
    - 18.1.2 动画功能的工作流程：Ext.fx.Anim
    - 18.1.3 分步动画的工作流程：Ext.fx.Animator
  - 18.2 使用动画
    - 18.2.1 由最简单的动画开始
    - 18.2.2 过渡效果使用的功能函数介绍
    - 18.2.3 使用分步动画
    - 18.2.4 注意的问题
  - 18.3 在Element对象中使用动画
  - 18.4 本章小结
- 第19章 拖放功能
  - 19.1 拖放功能的构成及工作流程
    - 19.1.1 概述
    - 19.1.2 DragDropManager对象的工作流程
    - 19.1.3 注册节点：Ext.dd.Registry
    - 19.1.4 一般拖动功能的工作流程：Ext.dd.DD
    - 19.1.5 DragSource对象的工作流程
    - 19.1.6 DropTarget对象的工作流程
    - 19.1.7 DragZone对象的工作流程
    - 19.1.8 DropZone对象的工作流程
  - 19.2 使用拖放功能



## <<Ext JS权威指南>>

- 19.2.1 最简单的拖动效果
- 19.2.2 使用DragSource对象与DropTarget对象
- 19.2.3 使用DragZone对象与DropZone对象（使用Registry对象）
- 19.2.4 使用DragZone对象与DropZone对象（不使用Registry对象）
- 19.2.5 通过拖动实现节点排序
- 19.2.6 使用GridViewDropZonePlugin插件
- 19.2.7 使用TreeViewDragDropPlugin插件
- 19.2.8 关于Grid和Tree拖动后的数据保存问题
- 19.3 本章小结
- 第20章 扩展与插件
- 20.1 扩展与插件的区别
- 20.2 扩展与插件如何选择
- 20.3 如何编写扩展
  - 20.3.1 命名空间
  - 20.3.2 定义扩展
  - 20.3.3 定义别名
  - 20.3.4 定义备用名
  - 20.3.5 要求加载的类：requires与uses
  - 20.3.6 混入功能
  - 20.3.7 构造函数与initComponent方法
  - 20.3.8 静态属性和方法与单件模式
  - 20.3.9 可自动生成set和get方法的属性与initConfig方法
  - 20.3.10 在扩展中常用的方法
  - 20.3.11 编写扩展：TreeComboBox
- 20.4 如何编写插件
  - 20.4.1 概述
  - 20.4.2 AbstractPlugin对象
  - 20.4.3 插件的别名
  - 20.4.4 编写插件：RowColor
- 20.5 扩展和插件介绍
  - 20.5.1 概述
  - 20.5.2 本地分页代理：Ext.ux.data.PagingMemoryProxy
  - 20.5.3 标签滚动菜单：Ext.ux.TabScrollerMenu
  - 20.5.4 编辑器TinyMCE
- 20.6 本章小结
- 第21章 主题开发
- 21.1 准备工作
  - 21.1.1 安装Ruby
  - 21.1.2 安装Compass
  - 21.1.3 SASS介绍
- 21.2 为Ext JS 4创建新主题
  - 21.2.1 概述
  - 21.2.2 目录结构
  - 21.2.3 修改配置
  - 21.2.4 编译
  - 21.2.5 测试主题
- 21.3 通过ui配置项设置组件样式

## <<Ext JS权威指南>>

### 21.4 本章小结

### 第22章 MVC应用架构

#### 22.1 MVC应用架构的构成及工作流程

##### 22.1.1 构成

##### 22.1.2 控制器的工作流程：Ext.app.Controller

##### 22.1.3 Application对象的工作流程

#### 22.2 一步一步实现MVC框架

##### 22.2.1 概述

##### 22.2.2 创建目录

##### 22.2.3 创建首页

##### 22.2.4 创建启动脚本：app.js

##### 22.2.5 定义登录对话框

##### 22.2.6 创建应用脚本：Application.js

##### 22.2.7 创建Viewport视图

##### 22.2.8 菜单视图及控制器

##### 22.2.9 实现订单管理

##### 22.2.10 实现产品管理

##### 22.2.11 示例效果

#### 22.3 本章小结

### 附录 简写类名与Ext JS类名对照表

## &lt;&lt;Ext JS权威指南&gt;&gt;

## 章节摘录

版权页： 插图： 第14章 按钮、菜单与工具条 按钮、菜单和工具条也是使用比较频繁的组件，可以说是简单而平凡的组件。

本章将讲述这些简单而平凡的组件是如何构成的，以及如何去使用它们。

14.1 按钮 14.1.1 按钮的构成：Ext.button.Button Button对象派生于Component对象，因而基本就是由自己的模板构成的，图14—1是按钮的元素构成。

从图中可以看到，一个按钮总共有5个元素，最外层的div负责监听单击事件，em元素的作用是根据定义使用不同的样式显示下拉的小箭头，button元素的作用是作为图标和文本的容器，两个span元素一个用于显示文本，一个用于显示图标，通过不同的样式，就可实现按钮的多种样式。

当为按钮定义了url配置项的时候，button元素会换成a元素，成为一个链接，而且是target为“\_\_blank”的链接，这适合用在一些需要打开链接的地方。

14.1.2 按钮的配置项、属性、方法和事件 1.配置项 allowDepress：布尔值，如果设置为false，会禁止按钮的按下动作。

该配置项只有在enableToggle为true时才有效。

默认值是undefined。

arrowAlign：设置小箭头的位置，值可以为right和bottom，默认值为right。

arrowCls：应用于小箭头的样式名称。

autoWidth：在默认状态下，如果没有设置按钮的宽度，它会尝试拉长其宽度以适应容器的宽度。

如果按钮的宽度是由可调整大小的布局管理的，该值设置为false可阻止按钮自动调整大小。

默认值为undefined。

baseCls：应用于按钮的样式名称，默认值为“x-btn”。

baseParams：参数对象，当定义了url配置项，该参数将作为链接的提交参数。

clickEvent：设置可以触发按钮句柄函数的DOM事件，默认值是click，值还可以是dblclick或contextmenu。

## <<Ext JS权威指南>>

### 编辑推荐

《Ext JS 权威指南》为Ext JS领域的集大成之作，不仅全面地讲解了Ext JS框架的使用和深入分析了其运行机制与原理，还对Firebug使用方法和原理进行了深入的讲解，目的是让读者在掌握Ext JS使用的基础上，能利用Firebug去深入研究Ext JS的原理并对开发中遇到的各种bug进行调试，是系统学习和开发参考的必备之选。

## &lt;&lt;Ext JS权威指南&gt;&gt;

## 名人推荐

相比之前的版本，Ext JS 4发生了革命性的变化。

它在创建之初就以“最全面现代化”为目标，采用了改进的架构并加入了许多新的特性，从而使开发者能以更低的成本开发出更加可靠的下一代Web应用。

Ext JS 4几乎对框架的每个部分都进行了升级和优化，力图充分利用现今各主流浏览器的能力，同时它也与之前的版本兼容。

本书是目前最全面和最系统化地讲解Ext JS 4的著作，包含Ext JS 4的所有功能模块以及开发方法和技巧，是一本实实在在的“权威指南”，适合系统学习和开发参考，强烈推荐。

——51CTO 中国领先的IT技术网站 上将军是Ext JS领域当之无愧的专家，他在Ext JS领域的造诣之深，令人兴叹。

难能可贵的是，他这些年来一直专注于Ext JS的研究和推广，紧跟Ext JS官方团队的步伐，撰写了大量关于Ext JS的最新资讯和技术文章，深受社区欢迎。

同时，他还撰写了两本Ext JS的专著，广获读者好评。

本书是他的第三本Ext JS著作，也是国内第一本真正全面针对Ext JS 4的专著，它不仅从应用的角度全面讲解了Ext JS的使用，而且还从源代码的角度深入分析了Ext JS的工作原理，可谓广度和深度兼备。

——Ext JS中国用户组 Ext JS 4是Web开发者的福音，本书则是希望系统学习并深入了解Ext JS4的读者的最佳选择！

目前市面上出版了很多关于Ext JS的书，但是真正基于Ext JS 4的著作仅此一本。

大家都知道，Ext JS 4较之以前的版本发生了很大的变化。

更为重要的是，目前市面上的同类著作要么只讲Ext JS如何使用，要么只是枯燥地去分析源代码，本书则是以教会读者“更好地使用Ext JS”为核心目的，以讲解Ext JS的开发方法和技巧为主，同时还对一些重要的功能模块和知识点的原理进行了源代码分析，不仅能让读者知道如何做，而且还能知道为什么要这样做，最终实现做得更好的目的。

还要重点提到的是，本书为了照顾不同读者的需要，服务器端的案例同时给出了Java和C#两个版本，作者之用心可见一斑！

——Ext JS中文社区

#### 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>