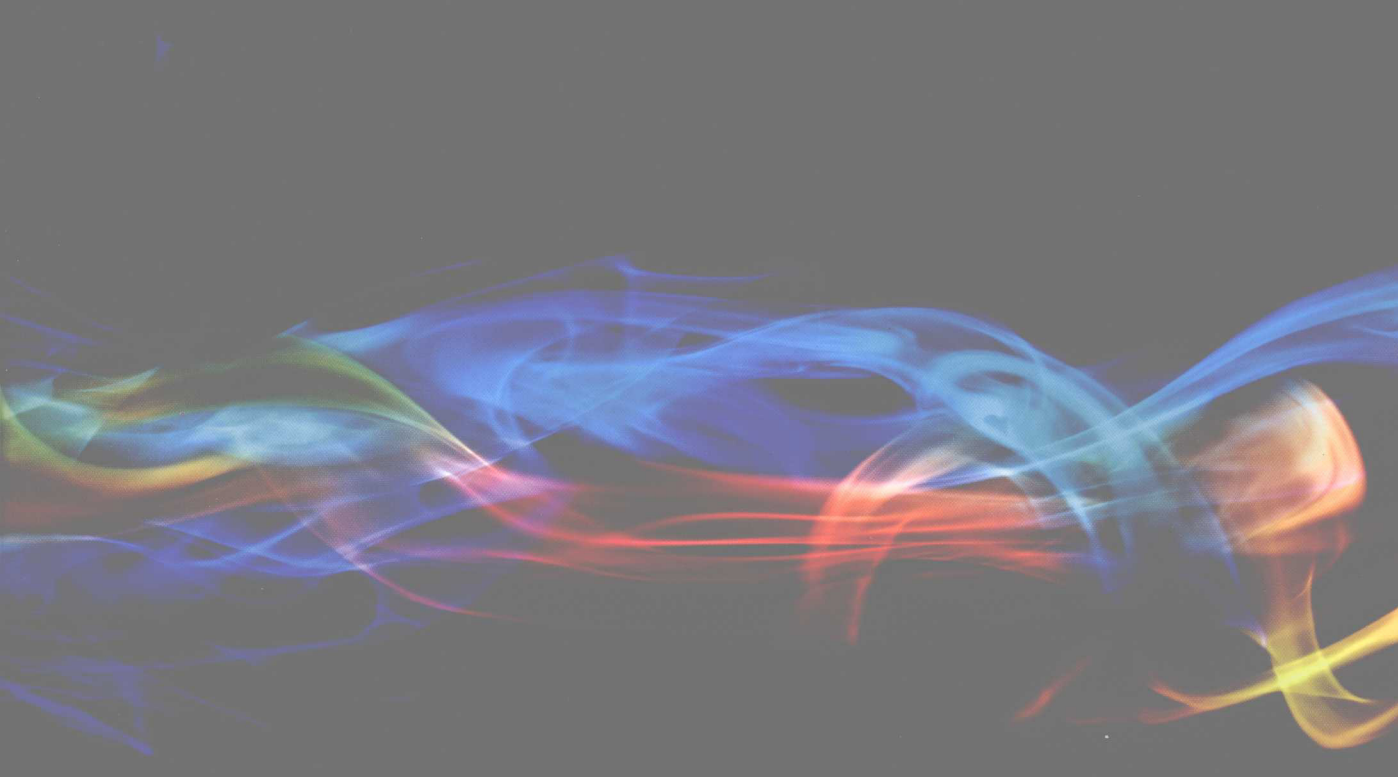




Ajax UI JavaScript Ext JS 3.x

JavaScript 凌后开发

Ext JS 3 详解与实践

张 鑫 黄灯桥 杨彦强 编著

- Ext JS 创始人中文博客参与维护者领衔力作
- 适合开发者的高性能、面向对象的开发框架
- Ajax 凌厉开发模式精密解析
- 内容编排来自 Ext 中文社区 3 年来的用户反馈
- Ext 先锋程序员精雕细琢的代码风格



- Ext JS 开发过程与测试过程全程视频
- 本书实例素材和优质源代码
- 本书中的项目源代码和配置文件
- Ext 3.x 框架文件



清华大学出版社

JavaScript 凌厉开发

——Ext JS 3 详解与实践

张 鑫 黄灯桥 杨彦强 编著

清华大学出版社
北 京

内 容 简 介

本书详细介绍 Ext JS 框架体系结构, 以及利用 HTML/CSS/JavaScript 进行前端设计的方法和技巧。作者为 Ext 中文网站站长领衔的三个国内 Ext JS 先锋, 在开发思维和开发经验上有着无可争议的功力。

本书包含的内容有 Ext.Element.*、事件 Observable、Ext 组件+MVC 原理、Grid/Form/Tree/ComboBox、Ajax 缓存 Store 等, 并照顾 JavaScript 新手, 穿插部分 JavaScript 基础知识。

本书附带的实例丰富, 包括散布各知识点的单独可运行实例、大型 UI 控件、ASP.NET 新闻发布者、Java DWR 整合方案、最新通信协议 Ext.Direct、Ajax 仿资源管理等。

本书附带光盘中的教学视频时长 270 分钟, 包括从零开始学 Ext、如何用 Firebug 调试、ext-base.js 源码级讲解、实例操作演示和源码评点、可视化 IDE 编程、优化工具的使用等。并附带完整的 Ext 3.1 中文文档。

本书适合广大的 Web 开发人员阅读。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

JavaScript 凌厉开发——Ext JS 3 详解与实践/张鑫, 黄灯桥, 杨彦强编著. —北京: 清华大学出版社, 2010.4
ISBN 978-7-302-22284-2

I. J… II. ①张… ②黄… ③杨… 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 044034 号

责任编辑: 栾大成 宋延清

封面设计: 杨玉兰

版式设计: 北京东方人华科技有限公司

责任印制: 李红英

出版发行: 清华大学出版社

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

社 总 机: 010-62770175

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

邮 购: 010-62786544

印 装 者: 三河市春园印刷有限公司

经 销: 全国新华书店

开 本: 188×260 印 张: 25.75 彩 插: 1 字 数: 622 千字
附 DVD 光盘 1 张

版 次: 2010 年 4 月第 1 版 印 次: 2010 年 4 月第 1 次印刷

印 数: 1~5000

定 价: 49.00 元

产品编号: 036870-01



A foundation you can build on

序言 (一)

我是很懒的人，认为懒是发展的动力，所以会不遗余力地去找合适的开发资源偷懒。因此每次对于时间不赶的任务，都是寻找合适的资源耗费的工作比较多。于是就从 DHTML 开始一直跟随着技术的发展学习下来，其中经历了 HTC、Ajax，收获不少，教训也不少，希望籍此文章能给大家提供一点提示和经验。

DHTML 篇

从 Web 开发开始，就已经知道 JavaScript 了，不过刚开始对我的作用就是在 form 提交的时候做基于客户端的验证。大概在 2000 年的时候，当时层(Div Layer)的概念刚刚兴起，公司有个项目的某个输入位置要输入日期，就突发奇想，其实可以用层做一个 JavaScript 的日期选择，于是在公司同事的协助下写了第一个 DHTML 的东西——日期选择器。当时还不懂封装和类什么的，所以就是一个简单的 JavaScript 函数。然后按图索骥，又写了一个颜色的选择器。不过在公司没干多久就走了，去搞网络，很久没碰 Web 开发。当再次搞 Web 开发的时候，技术已经有更高的发展，开始流行 HTC 了。

HTC 篇

HTC 的出现是在 IE 浏览器的统治地位确立后，由微软提出的一个组件化的东西，可惜因为微软的发展方向不在这里，所以也没提供太多的支持和大力发展，随着技术的发展，Firefox 等浏览器的兴起，HTC 就如昙花一现，不久就再没人问津了。

一般而言，HTC 是组件化了的脚本过程。尽管引入浏览器的机制不同，但遵循相同的 SDK 规范。无论是脚本运行环境，还是 DOM 文档结构。但 HTC 有着极高的扩展性。也就是说，HTC 可以为我们的 HTML 引入高级的自定义行为(Behavior)。如自定义的 Attribute、Method 或者事件。这就说明，我们可以使用 HTC 机制来开发一个有着高级特性的、可重用的、可扩展的组件。

接触到 HTC 也是懒的结果。当时一个项目需要一个既可以输入又可以选择的列表框，于是就满世界地找文档和例程。其实当时自己有个办法是可以实现的，就是在一个 select 前面放上一个 input，用 input 遮盖住 select 的显示部分，只留下拉部分，可惜做出来的效果看上去比较难看，而且有很大缺憾，最主要还是懒吧，于是就去搜索现成的更好的解决办法。终于在一个论坛 www.51js.com 上找到了，是用 HTC 封装的，感觉效果不错。在论坛上还发现

了一个网站 www.stedy.com(现在这个网站已经没有了, 技术更新太快了), 网站上有很多很好的 HTC。自己找了一套回来用, 可惜啊, 好东西是好东西, 但是用起来并不是很方便, 还要汉化, 修改也因为 JavaScript 经验不足, 只能叹气, 可惜了! 于是继续寻找, 找到了 Webfx.eae.net, 在这里发现宝藏了, 其中的 toolbar、tree、grid 一直用了很长一段时间, 许多项目都用上了, 从这时候开始感叹 JavaScript 的神奇, 可以实现的东西实在是太多。从这时起就开始固定下了一套开发模板。其中用得最多的 HTC 是日期选择和 mask 输入。而在其中的 tree 已经使用到了 XMLHttpRequest, 可以说是 Ajax 开始初露锋芒吧。框架的思想已经开始萌芽了, 只是大多都不成熟, 俺是懒人啊, 自然不去凑热闹了, 偏安于自己的模板。但是感觉写起 Web 应用来还是有点麻烦, 不适合懒人啊。

Ajax

随着 Web 2.0 的报道铺天盖地地砸来, 心动啊, 因为听说适合懒人。于是就不断收集信息。终于在《程序员》杂志上找到一些框架网站。那还等什么, 立马就跑到各大框架网站下载测试版来研究。

1. Qooxdoo

这个是好东西, Widget 很多, 其开发模式也是习惯了的 RAD 开发方式。不过一直没用得上, 因为没有 Splitbar, 非常可惜。等了好几个版本居然还没这东西, 真的有点晕了。现在终于有了, 不过现在更喜欢的是 Ext JS, 更适合懒人, 而且 Demo 效果看起来比 Qooxdoo 好多了。还有就是普遍采用 JSON 的定义方式, 我很喜欢。曾经用 Qooxdoo 做了一个菜单, 用的是非常传统的定义方式, 一行定义一个菜单条目, 然后组合到一个菜单, 然后再组合到屏幕——不适合懒人啊。而且同事反映说慢, 没办法, 我用的那个包足足 900 多 KB, 后来找到新的菜单条就立刻更换不用它了。不过看趋势是越来越大了, 现在居然到 1.1MB 了, 怕怕。不过 Qooxdoo 已经被 CodeGear Delphi for PHP 加入成为开发框架了, 前途看来还不错。但是感觉 Qooxdoo 还有一个缺陷, 就是本地化的问题。因为没深入研究, 而且 Demo 也没这方面的信息, 所以只是感觉。

2. Dojo

又是一个不错的框架。有我想要的东西, 但是 Grid 功能太弱了, 于是没再深入研究, 继续等待吧。

3. YUI

也是一个不错的框架。看过演示后, 因其 Grid 功能不是很好, 也没再深入研究。

4. jQuery 与 Prototype.js

忘记了是在搜索什么东西的时候搜索到了这两个框架的。很喜欢这两个框架, 曾经在二者之间进行过艰难的选择。jQuery 的 Widget 很多, 但是感觉它的代码有点晦涩, 可能是太灵

活的缘故,有时候看那些 **Widget** 的例子简直是看天书。而且对它全部采用正则表达式做全文搜索感觉不是很好,老这样搜索,得花多少时间呢。所以最后还是选择了 **Prototype**。**Widget** 不多就自己想办法变通一下吧。最开始的成果是将网上基于 **Prototype** 的 **Validation** 的 **Plugin** 和正在使用的 **Smartform** 的优点进行组合,写自己的校验类,通过一个 **JSON** 来定义 **Form** 元件的验证。然后写了一个 **Splitbar**,可惜不尽如人意,功力太浅了,这个 **Splitbar** 最大的问题就是在窗口的变化后会有很大问题,将就先用着吧,以后再慢慢改。第三个目标就是通过定义 **JSON** 自动生成 **Form** 和窗口,这个难度不大,就是在如何生成 **Label** 和定义宽度上尝试了几种办法,花了不少时间。**Prototype** 确实是一个不错的框架,可惜就是基于上面的开发太少了,很多时候要自己动手,对于懒人来说,实在是太痛苦了。不过终于发现了我一直期待的框架 **Ext JS**,实在是太令我兴奋了。

5. Ext JS

应该怎样来表达我对 **Ext JS** 的喜爱呢?无法形容!为什么那么喜爱这个框架?因为适合我这样的懒人啊!适合我习惯的开发方式!其对话框、**Layout**、**Form** 的组织形式及 **Toolbar** 和 **Menu** 的组织形式,都是我所喜欢的,也正是我自己想通过 **Prototype** 实现的东西,可惜自己做一个框架工程实在太太,所以只能实现部分功能,而且因自己水平有限,实现的功能也不好。现在忽然有一个那么适合我这懒人的东西出现,当然是无比兴奋了!越深入了解 **Ext JS**,越发觉这东西好使。**Grid**、**Layout**、**Form**,一个基本的开发模板也于近日完成了,懒人的未来工作就是复制、粘贴然后修改,轻松啊。

黄灯桥

序言 (二)

天天都说把 HTML/CSS/JavaScript 学好，可这个过程咋又觉得那么地难呢？

的确，HTML/CSS/JavaScript，尤其当中的 JavaScript(简称为 JS)，并非什么神乎其技，又不是我们干活的主力“语言”，更不要说让我们追之、慕之了。

比如一个开发 Web 的从业人员，他大概会如此——后台业务已经有了，现在写一些页面特效，加点 Ajax 效果。嗯，花上数小时恶补一下 JS，甚至倒是不必用恶补去形容此番过程，自己明白 JS 语法、if... else、switch 流程、六大数据类型……就是了。

“脚本语言 + 谁都会的 HTML/CSS，本身用起来就是要简单、傻瓜式的”。

一些接受 Ajax 并非儿戏的朋友可能认为这种学习态度已经要不得。且慢，咱们先不评价是否重视了 Ajax，从另外一个角度来讲，其动机是如此简单，可以实现功能的话已经不错了，没必要想得那么多。

可是，换 Ext 这么老练成熟的框架来说，如此简单的学习动机或者学习心态，是不是足以成为一个说服我们学习 Ext 的理由呢？这样的答案很明显是否定的，所以我们想先说说“学习动机”，理顺一下想法，这也是解开 Ajax 学习难的症结的一个侧面所在。

先跟读者交待一点，单刀直入式的说教只会凸显功利性动机，与我们追求自发式学习的目的不符，甚至苦口婆心式的或喂饭式的使招也没有益处——大家不容易接受。

我们不妨从一个故事说起——

话说大片《2012》，在灾难发生前的阶段，主角 Jackson(杰克逊)偶遇疯狂地理学家 Richard(查理)，对 Jackson 进行仙人指路，说“世界末日在即……而书架上有避难所”云云。安贫乐道的主角肯定不会相信。未料不过几天，竟被全然说中，就猛想起避难逃生图，连忙又开汽车、又开飞机、又开摩托，才得到了那份逃生图，于是打开一看，堂堂 Map of China！结果顿时令那个 Jackson 产生了错愕不已的表情，意外之余却又不容置否，因为根本没有其他的选择余地：逃难的目的地就是东方的国土。

戏剧性的效果是编剧、导演们最愿意送给观众的；观众们也乐于接受这种情理之中、意料之外的剧情营造。

但是科幻片与科技工作是两码事，怎么能扯在一起呢。不是我们极力推崇某种 Ext 开发观念，不是我们眼中无它样的技术，也不是我们言之旦旦，而是当两年、五年……过后，在浏览器上选择菜单“查看源文件(v)” (View Source)，却发现皆是 HTML/CSS/JavaScript。试问大家一句，还会有像 Jackson 那样错愕不已的表情流露？

2012 年很快就会到来。其中定有必然因素、偶然因素左右着技术潮流的走向。我们无法有决定性的言论，我们只是想籍此说明，HTML/CSS/JavaScript 本身并不难，只要我们明白了为什么学 Ajax 这一点，心里有底了——便真的不难。

既然明晰了学习动机后，就应该对这门课的掌握过程有个大致的规划。每个人的知识背景不同，学习的方式方法也不同，所以作者也很难交待所谓的终南捷径，因此只能就作者自己一直以来的学习历程，结合 Ext，相当浅薄地进行一些补充。

阶段一：摸索

参考序言(一)所言就很有代表性。落实到语言层面，首先对 JavaScript 的“二义性(Ambiguity)”有一定的认识。怎么去理解这个“二义性”呢？实际上好比我们的语言，不是有一词多义的吗？如果不结合上下文理解，所产生的语境自然是不同的。JS 的“二义性”也大抵一样，而且埋藏在非常微小的细节中，如 `function(arg1, arg2){...}` 中的括号 `()`，除了限定参数位置以外，我们还可以在表达式中像数学算式一样表示优先，如 `(x+y)*z`，但 JS 告诉您圆括号另外的一个作用就是“执行”，前提是必须与“函数类型的对象”放在一起，因此我们可以写成 `(function(){alert("hihi");...})()`，表示刚声明的一个函数立马就执行了，就会运行 `alert()`；不加最后的 `()` 就不执行，仅仅是声明一个函数而已。

像 `()` 这样的用法在 JS 里面还有更多，例如 `object[int/string]` 的 `object` 既可以是数组，也可以就是对象，方括号内的又可以是 `int` 类型或 `string` 类型的，都在一个“筐”里面装，另有引申。难怪初涉及的用户对这种小细节的地方视为唐突、乃至混乱。

既然有很多“二义性”，那么有没有“反二义性”存在呢——真的是有的。这里的关系比较微妙。以“同义词/近义词”的例子做比较：汉语的词(或其他自然语言)的文本是已确定的，围绕它有不同的意思；但 JS 是意思已经抓住了，但可以允许有不同的表达方式，解析器也会接受。例如在 JS 中写一个正则表达式，可以用 `new RegExp('\w\d+', 'ig')`，这时第一个参数就是字符串类型了，也可以用 `/\w\d+/ig`，这时不用字符串就可避免转义时的烦琐，使代码更可读。例子中的同一含义两种表述就是“反二义性”的表现，应该分别根据不同的应用场合来选择。

我们仍然对前面的例子补充一下，变换成这样 `(new Function('alert("hihi");'))()`——不知道聪明的读者看出区别了没有？

JS 是一种轻松的脚本语言，对相同的定义有多种实现方式。这样的语言，在没有深刻认识之前，太多的语法总给人一种混乱和不够严谨的感觉。到一定程度了，你便能理解这种语言的种种轶事、每个方法的种种由来，甚至可以领悟出一点关于编码哲学的道理。

还有一点是大家心中抱怨得比较多的——Ajax 开发如果有个像样的 IDE 该多好呀，问题是现在许多 IDE 都不成熟，这点一时还难以被克服，有待以后观察。作者用过的 JS IDE 也不下 N 个，后来还是回头用那个 JsEclipse 感觉顺手，但一般还是以 Notepad2 直接修修改改(Notepad2 v3.1 才 236KB 大小)。至于 Dreamweaver/Visual Web Developer 等虽然还算不上 UI IDE(另有 Flex Builder 就很不错)，但可视化 Web 设计 HTML/CSS 的方面做得比较成熟，学习 HTML/CSS 有了好帮手。HTML/CSS 是纯粹的陈述式语言，本身几乎不存在流程控制，

因此都是被 JS 操控的对象。大家是否看过 Ext 生成的 HTML，都是一层层的 div 嵌套、再嵌套，周而复始……

应当注意当前讨论的过程中没有使用 Server 服务端的语言控制 HTML/CSS/JS，就是不用我们熟悉的 Server 生成 HTML，或静态模板这些方法。熟悉浏览器的编程工作后，通过动态控制浏览器的元素，就可以开辟一个新的园地，这便是接下来的“GUI 编程”。

阶段二：GUI 编程

尽管“GUI 编程”说得有点堂而皇之的感觉，但我们的想法是尽量不停留在 DOM 编程的级别上。我们认为，到现在为止或者将来，收获的知识财富真的应该是一步一个脚印的，也应该明确下一个目标将是怎样。诚如业界前辈 DLEE 所说“……JavaScript 允许开发者以一种长时间循序渐进的方式来学习和进步，将学习曲线拉得很平，所以 JavaScript 对于开发者来说是非常平易近人的……向我们展示出了 DHTML 的巨大潜力，我们总是可以通过修改和重构使得这些代码容易维护”。本来不是 JavaScript 的使命，却要担负 MFC UI/Swing/WinForms 的重任，我们开发者也只好亦步亦趋了。

在这个阶段中，我们要知道 JS 的新型理念，了解所谓的 New Wave JavaScript 是怎样的，是如何贴近 OO 的，怎样在 Ext 中构建一个符合面向对象特征的 UI 系统，编码的时候如何更能够发挥 Ext 的特点，如何使用 CSS Selector 等。对于大部分 Java/C# 开发者而言，框架便利性的意义通常体现于此，学习和使用变得非常自然和轻松了。当然，我们心里还要明白，原理上相互借鉴不等于直接套用，还需要充分认识其不足之处，而且不等于前面的难题都解决了，背后还有非常多繁重的工作需要完成，例如跨浏览器始终就不容易解决。

随着 Ext 一路偕手发展，今天许多的用户都达到了“GUI 编程”这个阶段，也就是精通的阶段，可以写出不错的 Ext 扩展组件，合理地发挥着 API 的作用。我们希望的就是在 GUI 这一环之中，有比较好的解决方案。如果 Ext 得以成功实践，大家对 GUI 的认识应该在不知不觉中有提高。不少项目的 GUI 是稍微压后考虑的一环。所幸的是，Ext 并非跟我们熟悉的开发技能认知相左，大家可以清晰地了解其套路，这样解决 GUI 问题时就不必太伤脑筋，这已经比以前的 DHTML 好很多了。其实大家都期盼有这样的项目，让开发者可以将更多的精力放在业务上，让 Ext 替自己解决一些一般性的界面问题。而专职的 UI 开发者就可以研究 Ext 的里里外外，毕竟 Ext 是开源的，只要在架构设计上预先考虑，适合于扩展，那么专职的 UI 开发人员也可以创建更多 UI 控件来，甚至通过贡献自己的代码创建一个控件市场。Ext 官方就有一个类似这样的(参考 Apple iPhone)软件交易市场的计划。

用 Ext 构建 UI 的目的已经达到，我们已经可以对 Ext 轻松地指点江山了。整个社区也有一定的知识储备，平时遇到问题可以透过百度、Google 的搜寻查找解决问题的线索，加入各个 QQ 群也是挺不错的主意。

对于大部分学习的用户而言，达到阶段二就会满足工作上的需求了，毕竟日常工作不可能围着 Ext 转；也可能有用户此时此刻的心情可以用“骑虎难下”来形容，因为用了 Ext 之后并没有我们先前所说的效果，反而不得其门而入，别太担心，闻道有先后，术业有专攻，社区一大批高手也非虚名，多请教一下自会有释然解惑之道。

阶段三：回归 UI 本质

若是年复一年地都搞代码，渐渐地就会感到乏味。工作和学习中累积的过程不断地使得作者往上层思考——什么是 UI 之道？莫非 Jack 练就出的 Ext 本领就只是 Jack 脑海中的私有产物？这些人如何能把浏览器里的各种元素玩得出神入化？jQuery 又是什么样的定位？这些都是很有趣的问题。

从接触 Web 前端开发起，就抱着一边是有机的逻辑程序，一边是唯美的界面这样一种兴趣去看这两者如何合二为一，所以反而忽略了他用户体验、人体工程学等的课题。但相信这些学术上的价值就反映在用户程序界面上，而且可以深入得更广。这里举一个实证，就是 Google 没有绚丽的界面，其服务却能提供细致入微的用户体验，与 Ext 相比，这又是一种什么样的体验哲学呢？

估计 Ext 的用户群中大多数人都与作者的年龄接近，正值当打之年，如果比较感兴趣，希望就这些话题一同学习，一同交流。

再版说明

前面对“学习动机”和“学习过程”都已交待完毕了，最后说说关于本书的一些情况。自第一版发行以后，陆续收到很多读者的评价，无论是主动发过来的，抑或是大伙 QQ 对话中的只言片语。这里收集的若干条是比较具有代表性的：

“该书对于 Ext 基础方面还是说得不错的，但需要有一定基础的人看。”

“视频太烂了，书本内容对于初学者过于难懂。”

“……买本 Ext 的书还附赠 Web/DHTML 啊？无论怎么说，还是觉得代码非常多。”

对于读者的评价，我们真的是心存感激，并借此鼓励做好这一版的升级更新；对于认为我们做得不够好的读者，将心比心，我们的确是辜负了厚爱（也的确有违初衷），现在唯一可做的，只有反思和改良——具体便是删减内容和继续精益求精，务求不再抱憾。

就是后来大成编辑都说，既然到 3.0 了，而且内容也调整了不少，要不干脆换个书名吧，都已婉言相拒。因为经过某些事件，作者一心一意想着的都是“哪里跌倒，就从哪里爬起来”。

斗胆再说前面提到的可以做什么，多少说明了我们百感交集的心态。为此，对于第二版目录、内容的敲定，我们耗费了大量的时间，不断讨论、不断调整、不断重新获取灵感，希望能有更好的形式传达 Ext 的精神和内容。关于 Ext JS 之所以成为业界首屈一指的不凡框架的原因，我们愿意成为它的一个重要或不重要的注脚。

张鑫

前言

Web 前端设计是近年来业界关注的热点，其中 Ajax 凭借显著的优势成为优先考虑的计算机软件交付平台。Ajax 不算新技术，在许多人眼中只是“旧瓶装新酒”的概念，不过摆在大家面前的是，Ajax 像呼吸的空气一样无处不在，有浏览器的地方便有 Ajax 的市场。在此明显的优势下，催生出许多 Ajax GUI 库，Ext JS 便是其中流行的框架，用户群不在少数。

揭开 Ext JS 漂亮的外观皮肤，其内部的实现可以说没有太多的新意可言。这样说皆因 GUI 理论早在 Web 普及之前就已成熟并在不同平台上加以实现，同属 GUI 门下的 Ext JS 只是一个运行于 Web 的 GUI，没有可能也没有必要构建一次新的 GUI 理论体系。换言之，Ext JS 或 Ajax GUI 的关键任务在于如何采用 HTML/CSS/DOM 等浏览器技术，进行 GUI 理论的实践——当然这是非常宏大的命题，背后所覆盖的技术细节、开发的难度、平衡度的拿捏都难以逐一述说。框架的任务正是为了解决此类纷杂的难题，从而为开发者提供清晰、易用、可维护的一线工具集合。不错，既然问题提出了，新型的框架就有必要成为“承前启后”的、成熟的框架，Ext JS 也不例外。论起实质，框架的设计要求便是：一边使用较先进的 GUI 理论体系来支撑整个框架的运作，调度有条不紊；一边告别过去 JavaScript 组件零散的编码方式，应该是善于利用浏览器技术的优雅编程；另外还要提交易于扩展的、对程序员友好的 API 接口。

毋庸置疑，值得肯定的是，我国的 Ext JS 用户群是世界数量第一的 Ext 用户群。不知读者是否察觉到，官方教程的中文文化程度也是最高的，并且围绕 API 的文档、例子、学习社区/博客资源遍地开花……至于本书作者的任务，则是在“承前启后”的基础上一方面把 Ext JS 2.0 到 3.1 的更新部分讲清楚，另一方面是“起承转合”，把 Ext 认知中的各个环节做个系统的理顺，不回避晦涩之处，让 Ext 用户学有所用。本书适合广大 Web 开发人群阅读。

本书各章的主要内容如下。

第 1 章 开篇

为第一次认识 Ext 的用户介绍 Ext 的情况，简单说明 Ajax 的开发大背景和 Ext 成长历史。

第 2 章 Ext 起步

这里初步接触了 Ext 的方方面面，包括获取 Ext 的方法、运行环境、如何兼容其他 JavaScript 库等一般问题，其中“如何学好 Ext 框架”是此章的重点，道破了 Ext.extend() 继承方法的奥秘。希望读者给予充分的关注，理解作者行文的用意。

第3章 Ext 基础

该章正式介绍 Ext 语法内容,属于整个 Ext 框架体系之中的基础知识部分。Ext 各对象层次中处于最底层、不可分解的对象便为“元素”,因此该章内容很大部分是围绕元素进行的,从 Ext.Element 对象的用法开始,到 HTML 模板的应用都有详细的解说。其中 CSS 选择器的使用和一些实用的 DOM 方法更是初学者不可缺少的内容。此章难度不大,一般 Web 程序员可轻松掌握。末尾的一节“制作一个对话框”的例子,开启了我们的 GUI 组件之旅,同时也呼应了 Ext 一系列的基础用法,包括引出了后面“事件”的内容,难度正好适中。该章内容适用于 Ext JS 及其精简库 Ext Core。

第4章 Ext 事件

事件是 GUI 的灵魂,抓住了事件就等于理解了 GUI 编程的要义。许多有经验的开发者都坦言,乐于见到对事件的深入介绍——这就是我们独立安排一章来讨论事件的原因。读者万不可忽略这一章。

就文本内容而言,该章又可粗略地划分为以下三大部分。其一,是浏览器 JavaScript 事件与 Function 的用法;其二,是适用于 Ext 的事件模式机制——Ext.util.Observable 类;其三,是例子。该章一开始的时候没有直奔 Ext 事件的主题,而是如过场般地先向读者介绍登记浏览器事件、事件的处理函数、函数的作用域、Function.apply()/call()等知识,这些完全可作为纯 Ajax 知识点,如此一来便可作为后面高级事件内容的铺垫和导引,逐级展开。紧接着是 Ext 的事件体系的引入,说明观察者模式为组件通讯的基本消息模型,指明具体涉及 Ext 包的哪几个类。由此可见,OO 对 Ext 施加的影响还是很大的。观察者模式是该章的难点之一,可能有部分读者未有设计模式的经验,或不深刻。不过可以放心的是,只要通过阅读这一章的内容,便可充分掌握其奥妙。末段的例子是详述制作一个“打分”Widget 的过程,遵循 HTML/CSS/JavaScript 介绍的三部曲。同样不能忽视的是,例子中也做 Ext 继承上的横向切入,以求加强对 Ext.extend()的说明。

此章内容同样适用于 Ext JS 及其精简库 Ext Core。

第5章 创建图形界面

从该章开始,以介绍理论知识点为先,开篇的时候就介绍新型 Ajax 架构与旧有 DHTML 编程的不同。从比较之中进一步理解 MVC 模式。对于读者正确理解 MVC 模式至关重要,Ext 中屡屡采用的就是 MVC 模式。又与前 4 章比较,此章的内容集中于那些“看得见”的 UI 组件,不仅揭示所谓的“图形界面”如何在 Ext 中制作出来,而且深入其背后的原理和源码。

Ext 2.0 中首次实现了“生存周期”的 UI 组件模型,Ext 3.0 中得到了进一步的加强。“生存周期”与整个架构密不可分,无论哪一个 Ext 组件都符合“生存周期”的特征。Ext 是希望大家扩展 API 的时候遵循一种良好定义的规则方法,因此学习好“生存周期”的 UI 组件模型是研究 Ext 源码或编写扩展组件的前提,所谓“模板方法模式”也是如此。

该章的第二大部分就是介绍 Ext 组件模型中的 Component/Container/Panel/Viewport,此处行文不是铺陈罗列 API 的要点,而是协助发掘 Component/Container/Panel/Viewport 的

细微之处，诱发自我兴趣来探究。若说完整度，当然 API 上的介绍比作者的介绍还要完整。

第三部分是“组件的高级应用”，实际上是有关组件各专题的简述，并有例子辅助之。

第四部分“容器的布局”说明各容器之间是怎样布局的。不严格讨论的时候，布局是一种布局“风格”。Ext 的 Grid 也好，Window 也好……涉及到 UI 布局就是表明容器与容器上一级/下一级、外一级/内一级之间的关系。当然这是容器“嵌套”关系之外的另外一种关系。如何将此关系阐述清楚就是该部分的内容。同样地举一反三，我们罗列 13 种布局风格的时候仅用一个大的例子来说明，但是经常用的 `AnchorLayout`、`FormLayout`、`ColumnLayout` 就详细说明了拖放操作能够为用户带来相当有价值的 UI 体验，我们立足于 `Ext.DD.*` 的底层去理解 Ajax 的拖放过程。

为什么还没有介绍 Grid、Form……等的大头组件？因为如果一章写完就会篇幅太大了。其实第 5、6 章都是讨论 Ext 的 UI 组件的，但划分为前后两章。按照是否绑定数据的 UI 组件来讨论，那么第 6 章就是专门讨论可绑定数据的控件，即与“布局、面板”这类没有绑定数据的对象区别开来，也就是“展现数据”的控件。

第 6 章 展示数据

UI 的任务不外乎“交互”和“表示层”，其中尤以数据的“表示/展示”为首要考虑，所以与我们打交道较多的便是负责此项任务的 UI 组件。该章第一小节是帮助后台产生 Ext 读取的 JSON 格式数据，当然 XML 的也可以。熟悉后台的用户应该按部就班即可。

Grid 在 Ext 中是比较复杂的组件，也是典型的“数据/表现分离”的代表。讲述 Grid 离不开讲述 Store 对象，因此讲述应以 Store 为先。怎样更系统地讲述 Store 与 Grid 之间的关系呢？介绍 MVC 模式是比较好的方法，首先，该模式厘清了 UI 组件中各逻辑对象的关系，以及如何在“观察者模式上”产生耦合/内聚的结果。我们打算在说明 Grid 真正用法之前，重点说说这层关系。又因为理论部分着墨较多，结合 Ext 说清楚真是件不容易的事情。当然本章提到 Store 的地方只是扼要地提及其理论，有关更全面的 Store 用法或者实践部分就要根据第 7 章的专题去阅读了。

一般用户关心的 Grid 配置方法，即“分页、查询、排序、列模型与单元格”均有细致提及，并给读者提供了配有充分说明的例子。需要特别说明的是，由于时间的关系，后台例子只提供了 C# 版本，但不是没有照顾其他类型的后台主机。实际上 C# 代码还是比较简单的，除 JSON 解析库外，没有使用特别的第三方组件。另外，网上讨论 Grid 的资源信手可捻，足见 Ext/Grid 发展之快。鉴于此，我们挑选了 Grid 的几种高级用法：多层表头、Grid 拖放和 `PropertyGrid` 等，并包括两个常用的 Grid 扩展组件 (`Ext.ux.Grid.*`)。这些都是 Ext 例子中没有提及而又比较常用的扩展或插件。言外之意，通过较多的案例和扩展工具，不仅可掌握其用法，而且可以帮助 Ext 用户进一步熟悉 Grid 的原理。

`DataView` 作为另外一种展现数据方式的 Ext 组件在本章中也有所提及。

Form 表单的应用是 Ext 的一大特色，改变了 HTML 表单传统的布局。我们前面已经交待了 Form 的排版和怎样控制外观的问题，现在我们聚焦于 Form 的数据加载、验证、提交等的

一系列流程。“Ajax 式提交”与“传统方式提交”清晰地说明了 Ext 表单的运作原理，以及我们应该采用何种手段控制 Form。Form 就是说明表单数据绑定的具体操作，当然还有表单验证、特殊字段控制、集成 HTML 编辑器等的方法。

列表组件 ComboBox 是 Ext 中强大的组件。我们透过不同的例子来说明 ComboBox 的用法，常见的“联动 ComboBox”说明了两个组件一起工作的情形，对理解 ComboxBox 很有帮助。最后则介绍典型的扩展“TreeComboBox”。

Tree 树状组件一般都是项目中必选的控件，开发者很容易把层次结构的数据放在 Tree 中展示。正如所料，Tree 在 Ext 中的“数据/页面”分离的思想中也得到很好的贯彻。本部分内容中我们先说明 Tree 的结构，其次说明 Tree 的样式修改，然后说明 Tree 的拖放行为，特别是对 Tree 与 Grid 的拖放都有例子演示。

纯 UI 部分大致介绍完毕，接着应该介绍的便是“数据驱动”。

第 7 章 数据驱动

说到底 Ext 是一个 UI 库、一个控件渲染引擎，然而“绑定数据”偏偏又是不可或缺的内容之一。不妨想想看，我们专精于控件如何漂亮、如何快速、如何流畅、如何友好……但没有数据绑定接口的 UI 又有何作用？这正是第 6 章“展示数据”开篇回答了的问题，我们特别在第 7 章做专题讨论，说明这个数据绑定接口究竟如何使用。

再回想起上世纪的 Web 页面中 JS 代码片段的做法，要插入文字或图片？哦，容易，请放到一个数组或 hash 中保存……一维数组不够？嗯，二维的，三维的……代码跑起来了，庆祝！但若干日之后，回头看自己写过的数组，发现却不容易维护啊！Ext 作者大抵也清晰地意识到了这个问题，认为新型 Ajax 库必须有一个强壮的数据模型，来应付愈加复杂的数据结构；同时其接口肯定是统一一致的，即同一个数据实体可以在不同 UI 控件之间共享，可靠地进行绑定。其实再横向地看看，其他稍大型的 JS 库(Dojo/YUI/Qooxdoo 等)的架构都有相似的独立数据模型包含在内。

此章介绍 Ext 中数据的提交与响应方式，然后对目前最流行的数据格式 XML 与 JSON 进行深入的介绍与比较，接着继续深入探讨 Ext 数据驱动的核心——Ext 的容器类、数据存储基本单元、数据存储类型、数据的读取方式、数据代理等。然后，我们结合目前最流行的 Ajax 远程框架 DWR，讲解 DWR 如何在 Ext 中使用，以及如何创建基于 DWR 的 Ext 数据读取器和数据代理。对于 REST 在 Ext 中的支持，也有一个例子对应说明。

对于 3.0 版更新变化幅度最大的，非 Ext.Direct 莫属。作者根据官方文档的指导介绍了这个新型通信机制。需要说明的是，相关的例子尽管还没有立刻给出，但是在后面第 9 章大型 UI 控件的例子中，就使用了 Ext.Direct 方案。结合实例讨论应该会更加立竿见影地说明问题。

第 8 章 JavaScript 高级应用

来自 extjs.com 的 Animal 同学说“在真正开发之前，掌握好高级 JavaScript 知识，就像鱼儿游在水里一样，对 JS 运用自如”。的确，学习 Ext 必须有相当程度的 JavaScript 知识，这也是摆在我们面前不争的事实。Web 开发人群对 JS 不陌生，但也不一定很熟悉，而且水平

非常参差不齐。总体来说，这一章将针对 Ajax 开发中的需要进行的专项讲解，读者可以根据自己的需要和使用水平有选择地阅读。

闭包(Closure)在诸多脚本语言中有的有，有的没有，但如果是动态语言的话，那这个特性就常见多了，而且非常有价值。该部分内容通过简单却深刻的例子，充分说明闭包在 JavaScript 中的升华，使得我们能以 C 的语法，运用“闭包”的思想来高效地使用代码——读者不容错过。

“JavaScript 的继承”既可作为独立于 Ext 以外的知识点，又是大家讨论得比较多的话题。我们从 HTML/CSS/JavaScript 的演变历史得知，那个行业委实充满着无奈、戏剧性的练历，就像在夹缝中求生存……撇开人文情怀，站在纯学术的角度去探究，就必须开动脑筋，一眼地分析源码，甚至每一行代码都反复揣摩。在介绍这方面内容的过程中，从 function 开始一步步介绍 Ext 继承的基础原理，帮助读者深入了解 JavaScript 语言机制以及 OO 的策略。我们希望该段内容是很好的“敲门砖”，读懂 Ext.extend()源码后可以立刻进入更广阔的 Ext 底层空间！由于本书不是专述 JavaScript 的缘故，因此该部分内容点到即止。有兴趣的读者可浏览我们的博客，参考一下其他的文章。

此外，该章还交待了以下两点内容。

(1) 处理 i18n 多语种的问题，这在跨国项目中可能是必须考虑到的一点。该章的这部分内容通过汉语、英语、日语的例子来说明具体的控制界面语种的方法。

(2) Ext 皮肤之“冷艳”、“绝代”……使其一跃成为靠面子吃饭的众多 CSS 程序中的翘楚。如果用户感觉厌倦，可以根据 CSS Sprites 和“九宫格”的理论来修改皮肤。该章含有这方面的内容。

第 9 章 大型 UI 控件

随着 Ext 应用的不断加深，一旦 Ext 无法满足我们的需求，可行之道就是扩展它的新组件。我们可以比较方便和容易地扩展组件，不过随着扩展组件复杂程度的增加，则带来了更多的 JavaScript 代码和对象，该章中的例子就是一个直接说明此问题的大型示例，演示了怎样从一个 UI 框架发展到一个更能促进生产力的封装方案，从而修补缺乏先进 IDE 的不足。

例子中还演示了新型通信机制 Ext.Direct。如果没有使用 Ext.Direct 的经验，该例子是一个直观而翔实的范本资源。

第 10 章 新闻管理系统

此框架被开发人员接受和使用的同时，我们晓得，新型 Ajax 架构明显是打破原有“表示层”整合模式的。在此启发下，大家很容易得到的结论便是，要使用 Ext 就要用得彻底——为此本书最后安排一个例子来说明，这是一个后台为 C# + SQL Server Express 的新闻管理系统，UI 部分就应该完全使用 Ext 来实现，完全摒弃传统的 ASP.NET 前台模式，把 Ext 的思想不折不扣地放到开发平台中。

非.NET 的读者就可能会抱怨，为什么没有其他类型的后台案例？确实，由于篇幅的关系，我们难以给出更多的例子，不过在前面的许多章节之中，例如介绍 Form 的数据加载时，就

使用了 JSP 的方案；第 7 章的“数据驱动”的作者本身是资深 J2EE 架构师，其例子便是偏 Java 的。最后一章出现的.NET 例子就是大体上的一个 50/50 平衡的构想，希望读者能够充分予以理解——当然不排除我们日后在网上图书专区推出其他平台的例子。

本书由张鑫、黄灯桥、杨彦强执笔。在编写过程中，以 InfoQ 中文站(infoq.com/cn)总编辑霍泰稳先生为首的许多朋友对本书的编写提出了许多建设性的意见和肯定的反馈，对本书的创作给予了极大的帮助。清华大学出版社栾大成编辑统筹下的工作小组对本书的编排和润色工作付出了辛勤的劳动，在此一一向他们表示衷心的感谢，没有他们给予的指导，很难顺利地完成此书的出版。

本书在编写过程中，作者参考了 extjs.com 文档及部分教程，在此特向这些资源的作者表示感谢。信息总在万变之中，我们笔下对 Ext 的班门弄斧尚不足以窥见 Ext 的全貌。由于水平有限，因此希望得到读者的批评和指正，这样才能反映出我们与 Ajax/Ext 爱好者或开发社区的共鸣。