

JavaScript RIA开发实战



[JavaScript RIA开发实战_下载链接1](#)

著者:(英)Dennis Odell

出版者:清华大学出版社

出版时间:2010

装帧:

isbn:9787302228400

本书介绍如何采用最合理的方式为RIA编写可靠的、易于维护的HTML、CSS和JavaScript代码，以及如何使用Ajax技术在后台实现浏览器与Web服务器的动态通信。本书将介绍您在构建Web应用程序时可能遇到的性能限制，以及如何以最佳的方式克服这些限制。此外，本书提供的提示可以使用户界面响应更加灵敏。

本书也将介绍如何通过添加使用自定义字体的印刷标题、多媒体回放组件、自定义窗体控件和动态绘图功能改进RIA的用户界面。此外还将演示如何在服务器连接中断时继续运行Web应用程序，如何使用Ajax读取存储在二进制文件中的隐藏数据，以及如何确保JavaScript Web应用程序中最高级别的可访问性。

作者介绍:

目录: 第I部分 最佳实践	
第1章 构建牢固的基础	3
1.1 最佳实践概述	3
1.1.1 确认最佳实践的最佳性的人员	3
1.1.2 可以从最佳实践中受益的人群	4
1.2 常规最佳实践	5
1.2.1 定义项目目标	5
1.2.2 掌握基本规则	6
1.3 标记语言最佳实践: 语义HTML	12
1.3.1 学习HTML标签	13
1.3.2 从文档类型定义开始	14
1.3.3 XHTML扩展HTML的方式	16
1.3.4 实际使用最佳实践	17
1.3.5 Web内容的可访问性指导原则	25
1.4 格式化最佳实践: CSS	27
1.4.1 像素完美再现设计的相关内容	27
1.4.2 W3C CSS标准	28
1.4.3 样式表的指导原则	28
1.4.4 样式的可访问性指导原则	35
1.4.5 注释块	37
1.4.6 浏览器迂回方法	38
1.4.7 本地化考虑事项	38
1.5 结构化文件夹、文件和有价值资源	39
1.5.1 易于阅读的URL	39
1.5.2 文件和文件夹命名	39
1.5.3 文件编码	40
1.5.4 组织有价值资源	40
1.6 设置开发环境	41
1.6.1 编写文件: 集成开发环境	41
1.6.2 存储文件: 版本控制系统	42
1.6.3 测试页面: 浏览器和开发工具	43
1.7 本章小结	44
第2章 面向富互联网应用程序的JavaScript	45
2.1 编码风格指导原则	45
2.1.1 使用一致的格式	45
2.1.2 使用大括号和圆括号	46
2.1.3 使用字母大小写来增加含义	46
2.1.4 使用描述性的变量和函数名	47
2.1.5 维护简短的函数块	48
2.1.6 使用ScriptDoc将注释变为评注	49
2.1.7 使用TODO标记剩余的工作	50
2.2 专业水准的JavaScript编程	50
2.2.1 避免解决不存在的问题	50
2.2.2 使用文档对象模型	51
2.2.3 动态更改CSS属性	52
2.2.4 操作元素及其内容	52
2.2.5 不要将JavaScript和HTML混合	53
2.2.6 将样式与代码分离	53
2.2.7 链接函数调用	54
2.2.8 编写健壮的代码	54
2.2.9 记住对代码进行本地化	56
2.3 面向对象的JavaScript	56
2.3.1 对象、类和构造函数	56
2.3.2 继承: 从现有的类创建新类	60

- 2.3.3 this关键字 62
- 2.3.4 访问属性和方法 64
- 2.3.5 对象字面值和JavaScript对象标记 65
- 2.3.6 创建名称空间和层次结构 67
- 2.4 库和架构 68
 - 2.4.1 选择库 68
 - 2.4.2 构建JavaScript库 69
- 2.5 构建RIA 83
 - 2.5.1 构造应用程序 83
 - 2.5.2 管理两组HTML 86
 - 2.5.3 使用设计模式 86
- 2.6 测试和测试驱动的开发 92
- 2.7 使用第三方脚本 94
- 2.8 本章小结 95
- 第II部分 性能
- 第3章 理解Web浏览器 99
 - 3.1 引擎：浏览器的动力室 99
 - 3.1.1 呈现引擎和JavaScript引擎 99
 - 3.1.2 JavaScript引擎的性能基准测试 100
 - 3.2 Web页面请求剖析 103
 - 3.2.1 HTTP：Web背后的通信标准 103
 - 3.2.2 HTTP状态码 108
 - 3.2.3 消息的传输机制 110
 - 3.2.4 HTML页面的加载顺序 112
 - 3.3 页面性能 113
 - 3.3.1 查看页面的性能 113
 - 3.3.2 找出性能的潜在瓶颈 114
 - 3.4 本章小结 116
- 第4章 性能调整 117
 - 4.1 性能是否确实是问题 117
 - 4.2 对Web服务器进行调整以获得出色性能 118
 - 4.2.1 对外部资源使用单独域名 119
 - 4.2.2 使用内容分发网络 119
 - 4.2.3 以数据块方式将HTML文件发送给浏览器 120
 - 4.2.4 定制HTTP头部以驱动浏览器缓存 121
 - 4.2.5 压缩服务器的输出 123
 - 4.3 对HTML页面进行调整以获得出色性能 124
 - 4.3.1 利用HTML Tidy程序缩小HTML文件的大小 124
 - 4.3.2 在HTML代码的结束处引用JavaScript文件 125
 - 4.3.3 减少HTTP请求的数量 126
 - 4.3.4 不要加载来自主页的每个资源 127
 - 4.3.5 减少域名查找次数 127
 - 4.3.6 跨域对组件进行拆分 128
 - 4.3.7 避免链接到重定向 129
 - 4.3.8 减少HTML元素的数量 129
 - 4.3.9 不要链接至不存在的文件 130
 - 4.3.10 缩小HTTP cookie的大小 130
 - 4.4 对样式表进行调整以获得出色性能 131
 - 4.4.1 使用CSSTidy工具压缩CSS文件 131
 - 4.4.2 不要使用@import命令 131
 - 4.4.3 提高表布局的效率 131
 - 4.4.4 在IE中避免使用CSS过滤器和表达式 132
 - 4.4.5 使用简码值 132
 - 4.4.6 使用CSS Sprite技术 136

- 4.4.7 避免低效的CSS选择器 138
- 4.5 对图像进行调整以获得出色性能 139
 - 4.5.1 理解图像格式 140
 - 4.5.2 优化PNG图像 142
 - 4.5.3 不要忘记favicon 142
- 4.6 对JavaScript代码进行调整以获得出色性能 142
 - 4.6.1 使用Dojo ShrinkSafe压缩JavaScript文件 143
 - 4.6.2 通过内容分发网络访问JavaScript库 143
 - 4.6.3 时机就是一切 144
 - 4.6.4 提升核心的JavaScript性能 145
 - 4.6.5 提高Ajax性能 148
 - 4.6.6 提高DOM性能 151
- 4.7 本章小结 156
- 第5章 欺骗性行为：可感知的响应性 157
 - 5.1 提供快速的视觉反馈 157
 - 5.1.1 选择正确的时机 157
 - 5.1.2 对超链接使用CSS伪类 158
 - 5.1.3 让用户知道表单正在提交 159
 - 5.1.4 更改鼠标指针 159
 - 5.1.5 使用Web 2.0样式动画指示器 160
 - 5.1.6 显示进度条 160
 - 5.2 处理运行时间长的脚本 161
 - 5.2.1 将运行时间长的脚本分割为代码块 162
 - 5.2.2 使用定时器多次运行代码块 164
 - 5.3 预测网站访问者的需求 165
 - 5.3.1 预先加载内容 165
 - 5.3.2 高效加载导航层次 166
 - 5.3.3 提前捕捉鼠标单击事件 167
 - 5.4 本章小结 168
- 第III部分 表现
- 第6章 美丽的印刷字体 171
 - 6.1 面临的挑战 171
 - 6.2 字体的基本结构 172
 - 6.3 给文本使用静态图像 172
 - 6.4 为文本动态生成图像 174
 - 6.4.1 使用CSS直接嵌入字体文件 174
 - 6.4.2 让服务器生成文本图像 176
 - 6.4.3 使用Flash生成使用自定义字体的文本 183
 - 6.4.4 使用矢量图生成文本 183
 - 6.5 使用可重用的自定义字体组件 184
 - 6.5.1 Text2PNG 184
 - 6.5.2 Scalable Inman Flash Replacement组件 187
 - 6.5.3 Facelift Image Replacement组件 190
 - 6.5.4 Typeface.js组件 193
 - 6.6 本章小结 194
- 第7章 多媒体回放 195
 - 7.1 处理可访问性 195
 - 7.2 使用可重用的音频回放组件 196
 - 7.2.1 SoundManager组件 196
 - 7.2.2 在没有Flash的浏览器中播放音频文件 199
 - 7.3 使用可重用的视频回放组件 200
 - 7.3.1 YouTube Chromeless Player 203
 - 7.3.2 JW FLV Player 208
 - 7.4 展望：HTML 5中的音频和视频 211

- 7.4.1 [audio]和[video]标签 211
- 7.4.2 JavaScript API 212
- 7.4.3 当前采用情况 213
- 7.5 本章小结 213
- 第8章 表单控件 215
 - 8.1 自定义现有表单控件 215
 - 8.1.1 按钮 215
 - 8.1.2 文本字段 219
 - 8.1.3 文件上传控件 220
 - 8.2 添加新型表单控件 223
 - 8.2.1 用于日期选择的日历构件 224
 - 8.2.2 滑块控件 241
 - 8.3 使用可重用的表单组件 253
 - 8.3.1 SWFUpload: 包含进度条的多文件上传 254
 - 8.3.2 TinyMCE:富文本编辑 258
 - 8.4 验证表单 260
 - 8.5 本章小结 261
- 第9章 脱机存储 263
 - 9.1 使用cookie保存数据 263
 - 9.1.1 创建cookie 263
 - 9.1.2 cookie的不足之处 266
 - 9.2 使用Internet Explorer的数据存储 266
 - 9.3 数据存储API简介 268
 - 9.3.1 本地存储API 269
 - 9.3.2 Mozilla的全局存储API 270
 - 9.3.3 客户端数据库存储API 271
 - 9.4 使用Flash共享对象保存数据 275
 - 9.5 创建跨浏览器的本地数据存储API 277
 - 9.6 使用可重用的脱机存储组件 282
 - 9.7 本章小结 282
- 第10章 二进制Ajax 283
 - 10.1 纯文本文件与二进制文件 283
 - 10.2 使用Ajax读取二进制文件 283
 - 10.3 从照片文件中提取图像数据 290
 - 10.3.1 理解EXIF格式 290
 - 10.3.2 使用JavaScript读取EXIF数据 291
 - 10.3.3 显示文件的EXIF数据 299
 - 10.4 本章小结 303
- 第11章 在浏览器中绘制 305
 - 11.1 创建可缩放矢量图 305
 - 11.1.1 创建SVG图像文件 306
 - 11.1.2 在HTML中指定SVG 307
 - 11.1.3 通过JavaScript指定SVG 308
 - 11.2 使用矢量标记语言绘制 309
 - 11.3 使用可重用绘图库构建动态图形 310
 - 11.4 使用HTML 5 [canvas]标签 317
 - 11.5 本章小结 319
- 第12章 富互联网应用程序的可访问性 321
 - 12.1 迎合需求的用户 321
 - 12.1.1 使用辅助技术的用户 321
 - 12.1.2 移动设备用户 321
 - 12.1.3 不使用鼠标的用户 322
 - 12.1.4 面向所有用户的可访问性 322
 - 12.2 恰当地使用前进以及后退按钮导航 323

- 12.3 设备无关的JavaScript 327
 - 12.3.1 设备无关事件 327
 - 12.3.2 设备无关事件委托 329
 - 12.3.3 已更新内容提示以及焦点 330
- 12.4 Web页面可访问性倡议——无障碍富互联网应用程序 333
 - 12.4.1 角色 333
 - 12.4.2 状态以及特性 335
 - 12.4.3 焦点管理 336
 - 12.4.4 使用键盘与ARIA构件交互 336
 - 12.4.5 WAI-ARIA示例 337
 - 12.4.6 验证 340
 - 12.4.7 测试 341
- 12.5 本章小结 342

• • • • • [\(收起\)](#)

[JavaScript RIA开发实战_下载链接1](#)

标签

web

JavaScript

软件开发

Web开发

RIA

评论

简单翻了翻, 一般吧. 南图借的.

学校图书馆里借的，前面部分值得一读，讲的内容挺实际的，不过都是较散的知识点，不适合刚入门的人阅读。。。

[JavaScript RIA开发实战_下载链接1](#)

书评

多的不说，这本书不是我买的，只是换工作的时候我用的那台机器里面有这书的电子版，拿着没事做就读了一下，没读多少，发现了一些问题。
简单来说，得到这本书的朋友们学习一下书中的代码思想就行了，代码看看就好，印在书上的代码你原版搬到项目里是不一定能用的。吐槽一下翻...

[JavaScript RIA开发实战_下载链接1](#)