

# Contents

<b>Acknowledgments</b>	<b>ix</b>
<b>Introduction</b>	<b>xxi</b>
<b>Part I: An Introduction to RIAs</b>	<b>1</b>
<b>Chapter 1: What Characterizes Rich Internet Applications?</b>	<b>3</b>
<b>Rich Internet Applications are “Plastic”</b>	<b>4</b>
An Example of Plastic (Dynamic) Web Pages	4
How Style Sheets Create a Better Page	6
<b>Rich Internet Applications: The Web Is Disruptive (Finally)</b>	<b>8</b>
<b>Rich Internet Applications Are Invisible</b>	<b>9</b>
Invisibility of Google	10
Focusing Your RIAs	11
<b>Rich Internet Applications Break Down Walled Gardens</b>	<b>12</b>
<b>Rich Internet Applications Create New (Unwalled?) Gardens</b>	<b>13</b>
Constraints in the Unwalled Garden	14
Understanding the Counter-Manifesto	15
<b>Rich Internet Applications Are Always Up-to-Date</b>	<b>17</b>
<b>Rich Internet Applications Are OS Killers</b>	<b>19</b>
Web 2.0 Layering	19
Cascading Style Sheets	20
Leaving the Desktop Behind	20
Is Java a Winner or Loser?	20
<b>Rich Internet Applications Are Browser Centric</b>	<b>21</b>
<b>Rich Internet Applications Are Network Centric</b>	<b>21</b>
<b>Rich Internet Applications Are Mind-Altering</b>	<b>23</b>
Observations from a Developer’s Altered Mind	23
Some Observations from Our Altered Minds	23
<b>Rich Internet Applications Are Software as Service</b>	<b>24</b>
<b>Rich Internet Applications Are User Centered</b>	<b>25</b>
<b>Rich Internet Applications Are Inherently Collaborative</b>	<b>25</b>
<b>Rich Internet Applications: Small and Agile Development</b>	<b>26</b>
<b>Rich Internet Applications Are Bound to Change</b>	<b>26</b>
<b>How Do I Change the Way I Develop Applications?</b>	<b>27</b>
<b>How Am I Going to Make Money from RIAs?</b>	<b>28</b>
<b>Rich Internet Applications Are Seductive</b>	<b>29</b>

<b>Chapter 2: RIA Foundations</b>	<b>35</b>
<b>XML: The Lingua Franca of the Internet</b>	<b>36</b>
Back to the Hip-Bone Song — XML Fundamentals	36
Using Attributes	38
The DOM Tree	39
<b>XHTML: A Language for Documents</b>	<b>42</b>
Document Structure	43
Text Structure	44
Describing Text Type	45
In-text Objects	46
Document Setup	47
<b>CSS: A Language for Style</b>	<b>48</b>
Attaching CSS to a Web Page	50
Painting By Numbers	51
The CSS Cascade Shimmy	53
Taking Full Advantage of CSS in Your RIA	54
<b>JavaScript: A Language for Action</b>	<b>54</b>
Attaching JavaScript to a Web Page	55
Binding JavaScript to User Actions	57
Removing and Adding from the DOM Tree	59
Inspecting and Setting an Element's Contents	60
<b>A Concluding Example — Angela's Ristorante</b>	<b>62</b>
<b>Summary</b>	<b>67</b>
<b>Chapter 3: The Programmable Web: The Mashup Ecosystem</b>	<b>69</b>
<b>Getting Ready to Mash</b>	<b>71</b>
<b>The Big Picture</b>	<b>71</b>
XmlHttpRequest Object	72
A Simple First Mashup	73
Working with XML Output	74
Working with JSON Output	86
<b>What's a Proxy and Why Do I Need One?</b>	<b>92</b>
Working with a Proxy	94
Working without a Proxy	96
XML or JSON	96
<b>The Yahoo and Google Traffic Mashup</b>	<b>97</b>
Getting a Geocode	97
Getting the Traffic Report	102
Dynamically Injecting a SCRIPT Tag into the HEAD	103
Putting Together the Final Mashup	106
<b>Summary</b>	<b>116</b>

---

<b>Chapter 4: Getting Started: Creating Your First RIA</b>	<b>117</b>
<b>Two Scenarios: If Restaurants Were Run Like Web Apps</b>	<b>117</b>
The Unusual Restaurant	118
An Alternative Experience	119
How Does This Relate to My Web App?	120
<b>AJAX versus the Traditional Approach</b>	<b>120</b>
Understanding the Traditional Web Application	121
Limitations of the Traditional Web Application	122
Understanding the Web Application That Uses AJAX	123
Advantages of Using AJAX over the Traditional Approach	125
AJAX Components	125
<b>Development Environment Setup</b>	<b>126</b>
<b>Your First RIA: Hello World</b>	<b>126</b>
Creating HelloWorldv1.html	126
Adding a JavaScript Event Handler	128
Defining the JavaScript Body	128
Creating the Asynchronous Request	128
Defining the Event Handler Function	128
Defining the Callback Function	130
Processing the Server's Response	131
The Complete HelloWorldv3.html file	132
Creating the helloFiles Directory	133
<b>Adding the Server Side</b>	<b>135</b>
Programming the Server-Side PHP	135
Programming the Java Servlet	138
Enhancing the Client Application	142
<b>Summary</b>	<b>145</b>
<b>Part II: RIAs Explored</b>	<b>147</b>
<b>Chapter 5: Debugging the Client Side</b>	<b>149</b>
<b>Debugging Tools</b>	<b>150</b>
Firefox	150
The Mozilla DOM Inspector	151
The JavaScript Console	152
Venkman	152
FireBug	154
Markup Validation	155
But Wait, There's More!	156

# Contents

---

<b>Logging</b>	<b>156</b>
The Alert Function	158
A Very Simple Logger	158
A Homework Assignment	160
<b>Summary</b>	<b>161</b>
<b>Chapter 6: The Model-View-Controller (MVC) Pattern</b>	<b>163</b>
<hr/>	
<b>The Model-View-Controller</b>	<b>164</b>
Origins: The Thing-Model-View-Editor	164
The Model-View-Controller Pattern in Detail	165
<b>Using MVC in Java</b>	<b>170</b>
The Model 2 Architecture	171
The Design of Health History in Java	173
MVC Frameworks for Java	174
<b>Web Application Frameworks</b>	<b>176</b>
Ruby on Rails: a Ruby-based MVC Framework	178
TurboGears: A Python-based Alternative to Rails	181
MVC Implementation Quick Reference	184
<b>Summary</b>	<b>184</b>
<b>Chapter 7: JavaScript Library Survey</b>	<b>185</b>
<hr/>	
<b>JavaScript Library Basics</b>	<b>186</b>
Why Use JavaScript Libraries?	186
The Various Libraries	188
Library Properties	189
Frequently Provided Library Capabilities	189
<b>JavaScript Libraries in Detail</b>	<b>191</b>
Dojo	191
MochiKit	195
Script.aculo.us	198
Yahoo! UI Library	201
<b>JavaScript Library Quick Reference</b>	<b>203</b>
<b>Summary</b>	<b>204</b>
<b>Chapter 8: Compiling to JavaScript</b>	<b>205</b>
<hr/>	
<b>Enter the First Contenders</b>	<b>206</b>
<b>The Google Web Toolkit (GWT)</b>	<b>206</b>
GWT Benefits	207
GWT Drawbacks	208
Developing with GWT	208

---

Developer Tools	210
Look and Feel of GWT	212
Automatic Serialization	214
<b>Ruby JavaScript (RJS)</b>	<b>215</b>
RJS Benefits	216
RJS Drawbacks	216
Developing with RJS	217
Look and Feel of RJS	218
<b>Summary</b>	<b>220</b>
<b>Chapter 9: An Introduction to ideaStax</b>	<b>221</b>
<b>ideaStax Concept of Operation</b>	<b>222</b>
Creating a New Stack, Card, or Concept	224
Creating Cards	226
<b>Getting Started: the Essential Elements</b>	<b>230</b>
ideaStax	230
MySQL	231
Ruby on Rails Setup	232
<b>ideaStax: the Implementation Detail</b>	<b>234</b>
Deconstructing Concept, Card, and Stack	234
Verifying Concepts, Cards, and Stacks	235
<b>Implementing the Views</b>	<b>237</b>
<b>Implementing the Controllers</b>	<b>239</b>
Creating Tests for HTTP POST or GET	240
Using the Ruby JavaScript Construct as a Partial Renderer	240
<b>Rails Control Flow</b>	<b>243</b>
<b>Summary</b>	<b>245</b>
<b>Part III: RIA Development in Depth</b>	<b>247</b>
<b>Chapter 10: Form Validation</b>	<b>249</b>
<b>Incorporating Form Validation into Your System</b>	<b>250</b>
Server-Side Validation of the Entire Form	250
Client-Side Validation with JavaScript	251
Asynchronous Server-Side Validation of a Single Form Element	251
<b>Capturing User Information</b>	<b>252</b>
<b>The Unchecked User Form</b>	<b>252</b>
<b>Validation in Practice: Client-Side Validation with Dojo</b>	<b>258</b>
Installing the Dojo Toolkit	258
Getting Started with Dojo	259

# Contents

---

Using Client-Side Validation Functions	259
Using Dojo Validation Widgets	260
<b>Validation in Practice: Server-Side Validation of the Completed Form</b>	<b>272</b>
Implementing the Server Side	272
Defining the Model	273
Defining the Controller	276
Defining the Helper Class	277
Compiling and Running the Servlet	282
Implementing the Client-Side by Enhancing the View	282
Enhancing the Helper by Adding Validation of the Other Form Fields	284
<b>Validation in Practice: Server-Side Validation of the Username</b>	<b>288</b>
Enhancing the View	289
Enhancing the Controller	291
<b>Summary</b>	<b>293</b>
<b>Chapter 11: Form Usability</b>	<b>295</b>
<b>Overview</b>	<b>296</b>
<b>Approaches to the Usable Form</b>	<b>296</b>
Limitations to Synchronous JavaScript in Forms	296
Advantages to AJAX and JavaScript Effects in Forms	297
<b>Form Usability in Practice: Populating a Field with Server Data</b>	<b>298</b>
Implementing the Server Side	299
Implementing the Client Side by Enhancing the View	305
<b>Form Usability in Practice: Using a Server Backed Combo Box</b>	<b>309</b>
An Auto-Complete Combo Box Example: Car Make	309
A Server-Populated Combo Box Example: Car Model	310
Implementing the Server Side	311
Implementing the Client Side by Enhancing the View	314
<b>Summary</b>	<b>321</b>
<b>Chapter 12: Drag and Drop</b>	<b>325</b>
<b>Introducing Prototype and script.aculo.us</b>	<b>326</b>
Understanding Prototype and script.aculo.us Capabilities	326
Obtaining Prototype and script.aculo.us	327
<b>DOM Elements in the Physical World</b>	<b>327</b>
Draggables	328
Droppables	330
Sortable Lists	332
<b>Drag and Drop in Context: The ideaStax Editor</b>	<b>333</b>
Basic Design	333
Planning for Drag and Drop	335
Implementation Plan	336

---

<b>A Client-Only ideaStax Editor</b>	<b>336</b>
Creating Slides Dynamically with script.aculo.us' Builder	336
Populating the Library with Draggable Slides	339
Creating a Droppable the Editor Pane	342
Implementing the Sortable Timeline	344
Client-Only Implementation Summary	347
<b>A Server-Backed ideaStax Editor</b>	<b>347</b>
Getting Started with Rails	348
Creating the Model	349
Creating the View	355
Creating the Controller	360
The End-to-End ideaStax Editor	364
<b>Summary</b>	<b>365</b>
<b>Chapter 13: User Interaction, Effects, and Animation</b>	<b>367</b>
<hr/>	
<b>Effects Introduction: Hide and Seek</b>	<b>368</b>
<b>The script.aculo.us Effect Class</b>	<b>369</b>
Using Effects	370
Example: Effects-Based Notifications	372
Binding Effects to Other script.aculo.us Actions	374
Combining Effects	377
<b>Modal Windows and the ideaStax Previewer</b>	<b>377</b>
Positioning the Preview Window	379
Fetching the Preview Window	381
Transitioning In and Out of the Preview Window	383
Making the Preview Modal	384
A Completed Modal Preview	386
<b>Summary</b>	<b>387</b>
<b>Chapter 14: Tagging and Rating (I): Creating an Infrastructure</b>	<b>389</b>
<hr/>	
<b>Understanding the Tagging Problem</b>	<b>390</b>
<b>A Tagging Application</b>	<b>391</b>
<b>Getting Started: the Essential Elements</b>	<b>395</b>
Python	396
MySQL	396
MySQLdb	396
Flickr Photo API	397
Apache Web Server	397
MochiKit	398
<b>Data Mining</b>	<b>400</b>
Downloading Photo Metadata	400
Accessing Photo Metadata	401

# Contents

---

<b>Setting Up the Data Model</b>	<b>403</b>
Setting Up the Data Tables	404
<b>Populating the Pictures Data Base</b>	<b>408</b>
Setting Up the Flickr Photo Interface	411
<b>Scoring Tags</b>	<b>415</b>
Using Styles to Implement the Tag Cloud	415
JSON Data from the Tagger	417
Database Lookups	418
Server-Side CGI for Getting a Tagged Photo	420
The Dynamically Generated RIAPage	422
<b>Summary</b>	<b>422</b>
<b>Chapter 15: Tagging and Rating (II): Using Social Capability</b>	<b>423</b>
<b>Stax So Far</b>	<b>423</b>
Stax Design Center	424
Retrieving Concepts and Cards	426
Getting Started: the Essential Elements	427
Implementing the Controllers and Views	437
<b>Summary</b>	<b>448</b>
<b>Part IV: Advanced Selected Topics</b>	<b>451</b>
<b>Chapter 16: Providing an API</b>	<b>453</b>
<b>Choosing How to Expose Your Service</b>	<b>454</b>
The Home-Grown Solution	454
The Web Service Solution	454
Choosing Casual over Formal	455
Exposing a Service over HTTP	455
<b>Metering Access</b>	<b>455</b>
<b>Defining the Interface</b>	<b>458</b>
<b>An Example API for Stax</b>	<b>459</b>
Writing Documentation	459
Documentation Sample	459
Constructing the API	462
<b>Summary</b>	<b>465</b>

---

<b>Chapter 17: RIA Widgets</b>	<b>467</b>
<b>Surveying the Field</b>	<b>468</b>
<b>Getting Started: the Essential Elements</b>	<b>468</b>
Unpacking a Widget with the Widget Converter	469
File Structure and Packaging	470
Widget Specification File Overview	471
<b>Defining Your First Widget: a Hello World Exercise</b>	<b>472</b>
The Basic Hello World	472
Packing the Widget with the Widget Converter	474
Extending Hello World	475
<b>A More Useful Example: Incorporating the Stax API</b>	<b>482</b>
Setting Up the Example	483
Creating the Basic Widget	483
Adding File Drag and Drop	486
Uploading the Concept to the Stax Server	488
Adding User Preferences	490
<b>Summary</b>	<b>495</b>
<b>Chapter 18: Rich Immersive Environments</b>	<b>497</b>
Looking Back in Rich Immersion: The World of Magic Cap	498
Forward in Time: The World of Snow Crash	500
And Now: The World of Second Life	500
Second Life: The Programmer's View	502
Getting Started: The Essential Second Life Elements	510
Connecting Second Life to Real Life	511
A Web 2.0 Demonstration of Second Life	515
<b>Summary</b>	<b>518</b>
<b>Part V: Appendix</b>	<b>523</b>
<b>Appendix: Dojo Validation Functions and Flags</b>	<b>525</b>
<b>Basic Dojo Validation Functions and Flags</b>	<b>525</b>
<b>Dojo Internet Validation Functions and Flags</b>	<b>529</b>
<b>Locale-Specific Validation Functions in Dojo</b>	<b>531</b>
<b>Index</b>	<b>533</b>

