

Dave Crane, Bear Bibault, and Jord Sonneveld
with Ted Goddard, Chris Gray, Ram Venkataraman, and Joe Walker



Ajax

IN PRACTICE

- GET GOING
- GET SAVVY
- 60 PROBLEMS SOLVED

contents

preface xiii
acknowledgments xv
about this book xviii

PART 1 FUNDAMENTALS OF AJAX..... 1

1 *Embracing Ajax* 3

- 1.1 Ajax as a disruptive technology 4
 - Redefining the user's workflow* 5
 - *Redefining web application architecture* 7
- 1.2 Ajax in ten minutes 9
 - Introducing XMLHttpRequest* 9
 - *Instantiating XMLHttpRequest* 10
 - *Sending a request* 11
 - Processing the response* 13
 - *Other XMLHttpRequest methods and properties* 14
- 1.3 Making Ajax simple using frameworks 16
 - Making requests with Prototype's Ajax.Request object* 18
 - *Simplifying Ajax responses* 21
- 1.4 Summary 24

- 2** ***How to talk Ajax*** 26
- 2.1 Generating server-side JavaScript 27
 - Evaluating server-generated code* 27
 - *Utilizing good code-generation practices* 30
 - 2.2 Introducing JSON 34
 - Generating JSON on the server* 36
 - *Round-tripping data using JSON* 40
 - 2.3 Using XML and XSLT with Ajax 44
 - Parsing server-generated XML* 44
 - *Better XML handling with XSLT and XPath* 50
 - 2.4 Using Ajax with web services 56
 - 2.5 Summary 63
- 3** ***Object-oriented JavaScript and Prototype*** 64
- 3.1 Object-oriented JavaScript 66
 - Object fundamentals* 66
 - *Functions are first class* 68
 - Object constructors and methods* 76
 - *Writing a JavaScript class: a button* 82
 - 3.2 The Prototype library 97
 - Generally useful functions and extensions* 98
 - *Array extensions* 100
 - *The Hash class* 102
 - *Binding context objects to functions* 103
 - *Object-oriented Prototype* 105
 - Rewriting the Button class with Prototype* 112
 - 3.3 Summary 116
- 4** ***Open source Ajax toolkits*** 117
- 4.1 The Dojo toolkit 118
 - Asynchronous requests with Dojo* 119
 - *Automatic form marshaling with Dojo* 123
 - 4.2 Prototype 125
 - Asynchronous requests with Prototype* 125
 - *Automatic updating with Prototype* 131
 - *Periodic updating with Prototype* 134
 - 4.3 jQuery 136
 - jQuery Basics* 136
 - *Asynchronous loading with jQuery* 140
 - *Fetching dynamic data with jQuery* 145
 - 4.4 DWR 150
 - Direct Web Remoting with DWR* 151
 - 4.5 Summary 159

PART 2 AJAX BEST PRACTICES..... 161

- 5** **Handling events 163**
- 5.1 Event-handling models 165
 - Basic event-handling registration 165* ▪ *Advanced event handling 169*
 - 5.2 The Event object and event propagation 172
 - The Event object 172* ▪ *Event propagation 173*
 - 5.3 Using Prototype for event handling 178
 - The Prototype Event API 179*
 - 5.4 Event types 180
 - Mouse events 181* ▪ *Keyboard events 182* ▪ *The change event 185* ▪ *Page events 186*
 - 5.5 Putting events into practice 189
 - Validating text fields on the server 190* ▪ *Posting form elements without a page submit 195* ▪ *Submitting only changed elements 198*
 - 5.6 Summary 201
- 6** **Form validation and submission 202**
- 6.1 Client-side validation 203
 - Validating on the client side 203* ▪ *Instant validation 209* ▪ *Cross-field validation 211*
 - 6.2 Posting data 218
 - Anatomy of a POST 218* ▪ *Posting data to a server 220* ▪ *Posting form data to a server 223*
Detecting form data changes 227
 - 6.3 Summary 233
- 7** **Content navigation 234**
- 7.1 Principles of website navigation 235
 - Finding the needle in the haystack 235* ▪ *Making a better needle-finder 237* ▪ *Navigation and Ajax 238*
 - 7.2 Traditional web-based navigation 241
 - A simple navigation menu 241* ▪ *DHTML menus 243*

- 7.3 Borrowing navigational aids from the desktop app 247
 - The gooxdoo tab view* 248
 - *The gooxdoo toolbar and windows* 250
 - *The gooxdoo tree widget* 254
- 7.4 Between the desktop and the Web 259
 - The OpenRico Accordion control* 259
 - *Building an HTML-friendly tree control* 263
- 7.5 Summary 270

8 **Handling back, refresh, and undo** 271

- 8.1 Removing access to the browser's navigation controls 272
 - Removing the toolbars* 272
 - *Capturing keyboard shortcuts* 274
 - *Disabling the right-click context menu* 275
 - *Preventing users from navigating history or refreshing* 276
- 8.2 Working with a browser's navigation controls 280
 - Using the JavaScript history object* 280
 - *Hashes as bookmarks* 281
 - *Introducing the Really Simple History (RSH) framework* 283
 - *Using RSH to maintain state at the client level* 284
 - *Using RSH to maintain state at the server level* 289
- 8.3 Handling undo operations 293
 - When to provide undo capability* 294
 - *Implementing an undo stack* 295
 - *Extending the undo stack for more complex actions* 300
- 8.4 Summary 309

9 **Drag and drop** 311

- 9.1 JavaScript drag-and-drop frameworks 313
- 9.2 Drag and drop for Ajax 314
 - Drag-and-drop Ajax shopping cart* 314
 - *Manipulating data in lists* 321
 - *The Ajax shopping cart using ICEfaces* 326
- 9.3 Summary 335

10 **Being user-friendly** 336

- 10.1 Combating latency 338
 - Countering latency with feedback* 338
 - *Showing progress* 345
 - *Timing out Ajax requests* 351
 - Dealing with multiple clicks* 355

- 10.2 Preventing and detecting entry errors 359
 - Displaying proactive contextual help* 359
 - *Validating form entries* 366
- 10.3 Maintaining focus and layering order 374
 - Maintaining focus order* 375
 - *Managing stacking order* 381
- 10.4 Summary 387

11 **State management and caching 388**

- 11.1 Maintaining client state 390
- 11.2 Caching server data 392
 - Exchanging Java class data* 393
 - *Prefetching* 402
- 11.3 Persisting client state 406
 - Storing and retrieving user state with JSON* 406
 - *Persisting JSON strings through AMASS* 409
- 11.4 Summary 413

12 **Open web APIs and Ajax 415**

- 12.1 The Yahoo! Developer Network 416
 - Yahoo! Maps* 417
 - *The cross-server proxy* 421
 - *Yahoo! Maps Geocoding* 430
 - *Yahoo! Traffic* 436
- 12.2 The Google Search API 443
 - Google search* 443
- 12.3 Flickr photos 454
 - Flickr identification* 455
 - *Flickr photos and thumbnails* 459
- 12.4 But wait! As they say, there's more... 464
 - Amazon services* 464
 - *eBay services* 464
 - MapQuest* 465
 - *NOAA/National Weather Service* 465
 - *More, more, more...* 465
- 12.5 Summary 465

13 **Mashing it up with Ajax 466**

- 13.1 Introducing the Trip-o-matic application 467
 - Application purpose* 467
 - *Application overview and requirements* 468
- 13.2 The Trip-o-matic data file 469
 - What format should we use?* 469
 - *The trip data format* 470
 - *Setting up Flickr photo sets* 471

- 13.3 The TripomaticDigester class 473
 - The dependency check* 473
 - *The TripomaticDigester constructor* 474
 - *Digesting the trip data* 475
 - Loading the points of interest* 476
 - *Collecting element text* 477
- 13.4 The Tripomatic application class 479
 - The Tripomatic class and constructor* 480
 - *Creating the content elements* 482
 - *Filling in the trip data* 484
 - Showing the map* 487
 - *Loading the thumbnails* 488
 - Displaying the photos* 491
- 13.5 The Trip-o-matic application page 492
 - The Trip-o-matic HTML document* 492
 - *Tripping along with style* 494
- 13.6 Summary 496
 - index* 499