



WICKET IN ACTION

Martijn Dashorst
Eelco Hillenius

contents

foreword xvii
preface xix
acknowledgments xxi
about this book xxiii

PART 1 GETTING STARTED WITH WICKET.....1

1 *What is Wicket? 3*

- 1.1 How we got here 4
 - A developer's tale 4* ▪ *What problems does Wicket solve? 5*
- 1.2 Wicket in a nutshell 10
 - Just Java 11* ▪ *Just HTML 12* ▪ *The right abstractions 13*
- 1.3 Have a quick bite of Wicket 14
 - Hello, uhm ... World! 15* ▪ *Having fun with links 17*
The Wicket echo application 21
- 1.4 Summary 23

2 *The architecture of Wicket 24*

- 2.1 How Wicket handles requests 25
 - Request-handling objects 25* ▪ *The processing steps involved in request handling 29* ▪ *Thread-safety 31*

- 2.2 Introducing Wicket components 31
 - The component triad* 32
 - *Wicket's Java components* 33
 - Page: one component to rule them all* 34
 - *Components and markup* 35
 - *Separation of presentation and logic: a good thing?* 38
 - *The component's data brokers: models* 39
 - Extending components with behaviors* 42
- 2.3 Summary 44

3 **Building a cheesy Wicket application** 45

- 3.1 Introducing Cheesr 46
 - Setting up shop* 46
 - *Designing the user interface* 51
- 3.2 Creating the store front 53
 - Cutting to the cheese* 54
 - *Adding the shopping cart* 58
 - Going to check out* 62
 - *Adding pagination to the list of cheeses* 64
- 3.3 Creating the checkout page 66
 - Adding the billing address form* 67
 - *Adding validation to the billing-address form* 72
 - *Creating a reusable shopping cart* 74
- 3.4 Summary 78

PART 2 INGREDIENTS FOR YOUR WICKET APPLICATIONS79

4 **Understanding models** 81

- 4.1 What are models? 82
- 4.2 A taste of the standard models 84
 - Using the simple Model* 84
 - *Using PropertyModels for dynamic models* 90
 - *Saving code with CompoundPropertyModels* 92
- 4.3 Keeping things small and fresh: detachable models 96
 - What is detaching?* 96
 - *Working around a serialization problem with detachable models* 98
 - Using LoadableDetachableModel* 100
- 4.4 Nesting models for fun and profit 102
- 4.5 Summary 104

- 5 Working with components: labels, links, and repeaters 106**
- 5.1 What are components? 107
 - 5.2 Displaying text with label components 109
 - Using the Label component to render text 109*
 - Displaying multiple lines using a MultiLineLabel 111*
 - Displaying formatted text using labels 112*
 - 5.3 Navigating using links 113
 - Linking to documents using static links 113*
 - Using ExternalLink to render links programmatically 114*
 - Linking to Wicket pages with BookmarkablePageLinks 115*
 - Adding bookmarkable links automatically with wicket:link 119*
 - 5.4 Responding to client actions with a link 120
 - Using Link to respond to client actions 120*
 - Using AjaxFallbackLink to respond to client actions 122*
 - 5.5 Using repeaters to repeat markup and components 124
 - Using the RepeatingView to repeat markup and components 124*
 - Using a ListView to repeat markup and components 128*
 - 5.6 Performing common tasks with components 130
 - Hiding parts of a page 131* ▪ *Manipulating markup attributes 133* ▪ *Removing excess markup 136*
 - 5.7 Summary 137
- 6 Processing user input using forms 139**
- 6.1 What are forms? 140
 - 6.2 How does form processing work? 141
 - Submitting a form from the browser to the server 141*
 - Processing the form submission on the server 144*
 - 6.3 Components for text input 145
 - Using a TextField to process single-line text 146*
 - Using a PasswordTextField to process a password 147*
 - Using a TextArea to process multiline text 148*
 - 6.4 Selecting from a list of items 149
 - Selecting a single value from a list of choices 149*
 - Selecting multiple values from a list of choices 152*
 - Mapping an object to a choice and back using a ChoiceRenderer 153* ▪ *Using check boxes for boolean properties 155*

- 6.5 Components for submitting form data 157
 - Using buttons to submit data* 157
 - *Using links to submit data* 158
 - *Using Ajax to submit data* 159
 - Skipping Wicket's form processing* 161
- 6.6 Validating user input 162
 - Making a field required* 162
 - *Converting user input from strings to domain types* 163
 - *Using Wicket's supplied validators* 163
 - *Writing your own validator* 165
- 6.7 Providing feedback 166
 - Feedback messages* 166
 - *Using the info, error, and warn methods for general messages* 169
 - *Displaying feedback messages using a FeedbackPanel* 170
- 6.8 Summary 172

7 **Composing your pages** 173

- 7.1 Grouping components 174
 - Grouping components on a page: WebMarkupContainer* 175
 - Reusing grouped components by creating a Panel* 178
 - Grouping components using fragments* 182
- 7.2 Page composition: creating a consistent layout 183
 - Creating consistent layouts using plain pages* 184
 - Creating consistent layouts using markup inheritance* 187
 - Creating consistent layouts using panels* 191
 - Which is the best?* 193
- 7.3 Summary 195

PART 3 GOING BEYOND WICKET BASICS 197

8 **Developing reusable components** 199

- 8.1 Why create custom reusable components? 200
- 8.2 Creating a component that selects the current locale 200
 - What are reusable custom components?* 201
 - *Implementing the locale-selector component* 202
 - *Creating a compound component* 204
 - *Adding a Reset link* 207
- 8.3 Developing a compound component:
 - DateTimeField** 208
 - Composite input components* 209
 - *Embedding form components* 209
 - *Synchronizing the models of the embedded components* 211

- 8.4 Developing a discount list component 213
 - The container* 214
 - *The read-only discounts list* 216
 - The edit-discounts list* 217
- 8.5 Summary 221

9 *Images, CSS, and scripts: working with resources* 223

- 9.1 Using packaged resources 224
 - Including packaged resources using auto-linking* 226
- 9.2 Building export functionality as a resource 226
 - Creating the resource* 227
 - *Letting a component host the resource* 228
 - *Making the export available as a shared resource* 229
 - *Initializing the shared resource* 230
 - *An alternative implementation* 231
- 9.3 Resources and third-party libraries 232
 - A Jcaptcha image component* 232
 - *Implementing a complete Jcaptcha form* 234
- 9.4 Summary 236

10 *Rich components and Ajax* 238

- 10.1 Asynchronous JavaScript and XML (Ajax) 239
 - Ajax explained* 239
 - *Ajax support in Wicket* 242
 - Ajax components* 243
 - *Ajax behaviors* 245
- 10.2 Header contributions 247
 - Using header-contributing behaviors* 248
 - *Using the header contributor interface* 249
 - *Using the wicket:head tag* 250
- 10.3 Ajaxifying the cheese discounts 251
 - Implementing in-place editing* 251
 - *Refactoring the discount list* 252
 - *How AjaxEditableLabel works* 254
- 10.4 Creating your own Ajax components 258
 - Using third-party Ajax engines* 258
 - Detecting client capabilities* 261
- 10.5 Gotchas when working with Wicket and Ajax 262
- 10.6 Summary 264

PART 4 PREPARING FOR THE REAL WORLD265

- 11** ***Securing your application*** 267
- 11.1 Session-relative pages 268
 - 11.2 Implementing authentication 269
 - Keeping track of the user* 269
 - *Authenticating the user* 270
 - *Building a user panel* 272
 - Building a page for signing out* 273
 - 11.3 Implementing authorization 274
 - Introducing authorization strategies* 274
 - *Protecting the discounts page* 276
 - *Disabling the Edit link for unauthorized users* 278
 - 11.4 Summary 281
- 12** ***Conquer the world with l10n and i18n*** 282
- 12.1 Supporting multiple languages 284
 - Localizing the UserPanel* 284
 - *Using <wicket:message> tags* 285
 - *The message-lookup algorithm* 288
 - Localized markup files* 289
 - 12.2 Customizing resource loading 291
 - 12.3 Localized conversions 293
 - Wicket converters* 293
 - *Custom converters* 295
 - 12.4 Summary 298
- 13** ***Multitiered architectures*** 299
- 13.1 Introducing the three-tiered service architecture 300
 - Advantages of utilizing a layered architecture* 301
 - Who is in charge of the dependencies?* 301
 - *Code without dependency injection* 302
 - *Dependency injection to the rescue* 303
 - 13.2 Layering Wicket applications using Spring 304
 - Spring time!* 305
 - *The simplest way to configure Wicket to use Spring* 306
 - *Using proxies instead of direct references* 307
 - *Using proxies from the wicket-spring project* 308
 - Wicket's Spring bean annotations* 309
 - *Using Spring bean annotations with objects that aren't Wicket components* 312

- 13.3 Implementing the data tier using Hibernate 313
 - Introducing Hibernate* 314
 - *Configuring Hibernate* 314
 - Implementing data access objects using Hibernate* 316
 - Wicket/Hibernate pitfalls* 317
- 13.4 Summary 319

14 ***Putting your application into production*** 321

- 14.1 Testing your Wicket application 322
 - Unit-testing Hello, World* 322
 - *Having fun with link tests* 324
 - Testing the Wicket Echo application* 326
 - *Testing validators on Cheers's checkout page* 327
 - *Testing a panel directly with the ShoppingCartPanel* 328
- 14.2 Optimizing URLs for search engines and visitors 330
 - Bookmarkable requests vs. session-relative requests* 330
 - Extreme URL makeover: mounting and URL encodings* 332
- 14.3 Configuring your application for production 336
 - Switching to deployment mode for optimal performance* 336
 - Providing meaningful error pages* 341
- 14.4 Knowing what is happening in your application 344
 - Logging requests with RequestLogger* 344
 - Using JMX to work under the hood while driving* 347
- 14.5 Summary 350
 - index* 351