

## *contents*

---

<i>preface</i>	<i>xvii</i>
<i>acknowledgments</i>	<i>xix</i>
<i>about this book</i>	<i>xxi</i>
<i>author online</i>	<i>xxvi</i>
<i>about the cover illustration</i>	<i>xxvii</i>

---

## **PART I THE LANGUAGE OF TAGS ..... 1**

---

<b><i>I</i></b>	<b><i>The big picture</i></b>	<b>3</b>
1.1	The JSP custom tags solution	4
1.2	HTTP review	5
	<i>HTTP protocol</i>	<i>5</i>
1.3	Dynamic web servers	9
	<i>Common Gateway Interface</i>	<i>10</i> ▪ <i>Web server APIs</i> 12
	<i>Server pages techniques</i>	<i>13</i> ▪ <i>Separating business and presentation logic</i> 15
1.4	Tag-based programming	17
	<i>Benefits of a tag-based approach</i>	<i>18</i>
1.5	WAP, voice, and other alternative web clients	19
1.6	Summary	22

- 
- ## 2 *Web development with Java* 23
- 2.1 Java and dynamic web content 24
  - 2.2 Introduction to Java servlets 25
    - The servlet API and programming model* 25
    - *Servlets and non-Java web servers* 30
    - *Servlet shortcomings* 31
  - 2.3 Introduction to JavaServer Pages 32
    - Basic JSP syntax elements* 33
    - *A JSP example* 33
    - Scriptlets* 34
    - *Implicit variables* 37
    - *Directives* 38
    - Tags* 40
    - *Tags for JavaBean manipulation* 41
    - Executing a JSP page* 43
  - 2.4 Access models for business/presentation de-coupling 48
    - Model-1* 48
    - *Model-2* 49
  - 2.5 Servlets, JSP, and application configuration 52
    - The WAR file and its structure* 53
    - The application deployment descriptor* 55
  - 2.6 Summary 57
- 
- ## 3 *Developing your first tags* 58
- 3.1 What are JSP custom tags? 59
    - Anatomy of a tag* 59
    - *Using a tag in JSP* 60
    - The tag library descriptor* 61
  - 3.2 Why tags? 62
    - Comparisons of scriptlets and custom tags* 63
  - 3.3 Setting up a development environment 63
    - Installing the JDK* 64
    - *Installing Tomcat* 64
    - Testing your Tomcat installation* 65
  - 3.4 Hello World example 66
    - Compiling the tag* 67
    - *Creating a tag library descriptor (TLD)* 68
    - *Testing HelloWorldTag* 69
    - Did it work?* 72
    - *A tag with attributes* 72
    - Packaging tags for shipment* 75
  - 3.5 A tag with a body 76
    - LowerCaseTag handler* 77
  - 3.6 Summary 79

- 
- 4 Custom JSP tag API and life cycle 80**
- 4.1 Requirements of custom tags 81
  - 4.2 Overview of the tag API 83
    - Tag interface 87* ▪ *Tag life cycle 90*
  - 4.3 The BodyTag interface and its life cycle 94
    - BodyTag interface 94* ▪ *BodyTag life cycle 97*
  - 4.4 Tag API classes 97
    - TagSupport and BodyTagSupport 99* ▪ *PageContext class 99* ▪ *BodyContent class 100*
  - 4.5 Tag-related features of JSP 1.2 101
    - TryCatchFinally interface 101* ▪ *IterationTag interface 102* ▪ *EVAL\_BODY\_BUFFERED, EVAL\_BODY\_AGAIN return codes 102*
    - Updated Tag life cycle 103*
  - 4.6 Summary 106

- 
- 5 Integrating custom tags with the JSP runtime 107**
- 5.1 Tag library descriptor in a nutshell 108
    - The role of the TLD 110*
  - 5.2 Referencing a tag library from a JSP 112
    - The Taglib's uri attribute 113*
  - 5.3 How the JSP runtime works 117
    - Send tag example 117*
    - Translating the JSP into a servlet 121*
  - 5.4 Summary 126

---

**PART II BASIC TECHNIQUES..... 127**

---

- 6 Tag development techniques 129**
- 6.1 Reusable tag programming techniques 130
    - The techniques you'll use most 130*

- 6.2 Writing content back to the user 132
  - Adding data to the output* 133
  - *Exceptions and writing to the user* 137
  - *Flushing the JspWriter's internal buffer* 137
- 6.3 Setting tag attributes 139
  - Specifying tag attributes in the TLD* 140
  - *Providing validity checks at translation time* 142
  - *Using the JavaBeans coding conventions* 145
- 6.4 Logging and Messaging 146
  - Logging* 147
  - *Handling and throwing exceptions* 147
  - Improving logging* 148
- 6.5 Using the Servlet API 150
  - Accessing the implicit JSP objects* 151
  - Accessing the JSP attributes* 160
- 6.6 Configuring tags and bundling resources 169
  - Configuring a web application* 170
- 6.7 Working with the tag's body 176
  - Tag body evaluation* 177
  - *Referencing your tag's body* 180
  - *A BodyTag example—logging messages* 184
- 6.8 Tag cooperation through nesting 188
- 6.9 Cleaning up 189
  - Review of tag life cycle* 190
  - Exceptions and cleanup* 190
  - Improving our base classes to handle cleanup* 192
- 6.10 Summary 194

---

## 7 *Building a tag library for sending email* 195

- 7.1 Sending email from a Java program 196
  - The Simple Mail Transfer Protocol* 196
  - Java-based email* 197
- 7.2 Our first email tag 200
  - SimpleJavaMailTag example* 200

- 7.3 Integrating the tag with the application 206
    - Specifying the SMTP server at the application level* 206
    - Using an existing mail session* 206
    - *Specifying the sender's address at the application level* 207
    - An enhanced SimpleJavaMailTag: JavaMailTag* 207
    - The JavaMailTag in action* 212
  - 7.4 Collecting attributes from the tag's body 215
    - Implementing body processing* 217
    - *Extending the email tag* 217
    - *Creating tags for subject and message* 218
  - 7.5 Adding assertions and input validation 223
    - Performing validation using custom tags* 224
    - Creating a tag for the send mail tag library* 225
  - 7.6 Summary 233
- 

## 8 *Using JavaBeans with Tags* 235

- 8.1 Java reflection 236
  - What is reflection?* 237
  - The reflection API* 238
- 8.2 JavaBeans and reflection 244
  - Tags and JavaBeans* 244
  - *JavaBeans properties* 244
  - JavaBeans introspection* 247
  - *Properties and introspection* 247
- 8.3 The Show tag 251
  - Components of the tag* 252
- 8.4 Exporting bean values from tags 264
  - Informing the runtime of exported scripting variables* 264
  - *The ExportTag* 270
- 8.5 Summary 275

---

**PART III ADVANCED TECHNIQUES ..... 277**

---

**9 *Posing conditions with tags* 279**

- 9.1 Evaluating conditions in JSPs 280
- 9.2 IfTag—A simple condition tag 281
  - Implementing IfTag* 282
  - The problem with IfTag* 283
- 9.3 The advanced condition tag family 285
  - WithTag* 287 ▪ *TestTag* 290 ▪ *TLD for the advanced condition tags* 296 ▪ *Our tag library in action* 297
- 9.4 Improving our advanced condition tags 299
  - Supporting complex conditions with a condition language* 300 ▪ *Supporting complex conditions with JavaBeans* 300
- 9.5 Summary 301

---

**10 *Iterating with tags* 302**

- 10.1 Iterating with tags 101 305
  - Iteration example: SimpleForeachTag* 306
- 10.2 Generalized iterating tags 311
  - A generic iteration interface* 311
  - IterationTagSupport* 312
- 10.3 IterateTag 317
  - Design considerations for IterateTag* 317 ▪ *Wrapping iterators* 317 ▪ *Implementing IterateTag* 319
- 10.4 Look, Mom! No scriptlets—IterateTag in action 323
  - Printing the shopping cart with scriptlets* 323
  - Printing the shopping cart with IterateTag* 325
- 10.5 Making it easier on the JSP author 326
  - Building a better tag* 328 ▪ *The design* 328
  - FieldGetter and ReflectionFieldGetter* 329
  - Integrating FieldGetter with IterationTagSupport* 331
  - Updating IterateTag to perform field substitution* 334 ▪ *Field substitution in action* 335

- 10.6 JSP1.2 and IterationTag 336
    - IterationTag* 336
  - 10.7 Summary 328
- 

## 11 Database access with tags 340

- 11.1 Choosing how to present database information 341
    - Why not just wrap everything in a JavaBean?* 342
  - 11.2 Designing our database presentation tag library 342
    - Handling database connectivity and passing results* 343
    - *Additional design considerations* 343
    - Implementation conclusions* 345
  - 11.3 IterateResultSetTag 345
    - ResultSetIterationSupport class* 346
    - *JDBCFieldGetter class* 347
    - *IterateResultSetTag in action* 348
  - 11.4 Full JDBC connectivity through tags 349
    - Improving our one-tag approach* 350
  - 11.5 Database tag library design 352
    - Requirements* 352
    - *Choosing our tags* 354
  - 11.6 Implementing the library 357
    - DBConnectionTag* 358
    - *SQLQueryTag* 365
    - EnumRsTag* 370
    - *Using our library for the first time* 373
    - *ViewRsTag* 375
  - 11.7 Integrating a controller servlet with our new library 380
    - The controller servlet* 380
    - *The JSP* 382
  - 11.8 Summary 383
- 

## 12 Custom tags and J2EE 385

- 12.1 What is J2EE? 386
  - J2EE server components and client access* 388
  - Deployment in J2EE* 389
  - *Why custom tags and J2EE are a good fit* 389
- 12.2 What are EJBs, and why learn of them? 390
  - EJBs—What are they?* 390

- Types of EJB components* 392
      - *EJBs and their functions* 395
      - *Example: catalogue entry EJB* 397
      - *Points to keep in mind* 406
- 12.3 Using and configuring J2EE services 406
  - *Getting services* 407
  - *Tag and Servlet API integration* 408
  - *Setting environment entries* 409
  - *Setting EJB reference entries* 410
  - *Setting resource factory reference entries* 411
  - *Wrap it up* 412
- 12.4 J2EE database connection tags 413
  - *DataSource* 413
  - *Updating database tags to use J2EE conventions* 414
- 12.5 J2EE email tag 421
  - *Defining a mail service* 421
  - *Referencing the mail service* 422
  - *J2EE send tag* 424
- 12.6 Using EJBs from within JSP 429
  - *Writing custom tags for EJB access* 429
  - *Retrieving the EJB home interface* 430
  - *Using the EJB home interface* 434
- 12.7 Summary 442

## PART IV CASE STUDIES ..... 443

---

### 13 *JDBC-driven WebStore* 445

- 13.1 Introduction 446
  - *Custom tags used* 446
- 13.2 Overview 447
- 13.3 Requirements 448
- 13.4 Design 452
  - *Model* 452
  - *View* 456
  - *Control* 461
  - *Utilities* 461
- 13.5 Implementation 461
  - *Tags* 462
  - *Views* 474
- 13.6 Summary 526

- 
- 14** *EJB-driven WAPStore* 527
- 14.1 Introduction 528
    - Custom tags used* 528 ▪ *WAP* 529 ▪ *EJB* 529
  - 14.2 Overview 529
  - 14.3 Implementation 530
    - Normal flow of events* 530 ▪ *Model* 531
    - View* 532 ▪ *Welcome view* 535 ▪ *Controller* 561
  - 14.4 Summary 563

---

**PART V DESIGN** ..... 565

---

- 15** *JSP tag libraries—tips and tricks* 567
- 15.1 The case for custom tags 568
    - Tags and nonprogrammers* 568
    - Reusing tags* 569 ▪ *Maintaining tags* 570
    - Tags and application performance* 570
  - 15.2 Development considerations 571
    - Tag development dos* 571
    - Tag development don'ts* 576
  - 15.3 Further development and testing 578
    - Debugging tags* 578 ▪ *Testing tags on more than one JSP container* 578
  - 15.4 Design recommendations 579
    - Opening library internals* 579 ▪ *Generalizing your tags* 580 ▪ *Integration and the surrounding environment* 583 ▪ *Tags and general purpose libraries* 584
  - 15.5 Additional points to remember 584
    - The tag life cycle* 584 ▪ *The case for scriptlets* 585 ▪ *Freeing allocated resources* 585
    - Caching expensive results* 586
    - Supporting JSP1.1 and JSP1.2* 586
  - 15.6 Summary 584

- A** *What is XML?* 589
- A.1 XML vs HTML 590
  - A.2 XML syntax 590
    - DTDs* 592
  - A.3 XML pitfalls 593
  - A.4 Why XML? 594
  - A.5 Summary 595
  - A.6 Additional reading 595
- 

- B** *The Tag Library Descriptor* 597
- B.1 TLD elements 598
    - The taglib element* 598
    - *The tag element* 599
    - Element Recap* 601
  - B.2 A sample TLD 602
  - B.3 JSP1.2 and the new TLD entries 604
    - New taglib elements* 604
    - *New tag elements* 605
    - New attribute elements* 607
  - B.4 Summary 607
- 

- C** *Using the code examples* 608
- C.1 Using the example tags 609
    - Obtaining example tags* 609
    - *Using the example tags* 609
    - Compiling the example tags* 610
  - C.2 Using the case studies 611
    - The WebStore application* 611
    - *The WAPStore application* 613
- 

*references* 615

*index* 617