

contents

preface to the second edition xxv

preface to the first edition xxix

acknowledgments xxxi

about this book xxxiii

about the authors xxxviii

authors online xxxix

about the cover illustration xl

***I** Introduction 1*

1.1 What is JSP? 2

1.2 Dynamic content on the web 2

Why dynamic content? 3 ■ Common Gateway

Interface 4 ■ Template systems 5 ■ Java on

the Web 8 ■ How XML fits in 11

1.3 The role of JSP 13

The JavaBeans component architecture 13

JSP and Java 2 Platform Enterprise Edition 15

2 *HTTP and servlets* 17

2.1 The Hypertext Transfer Protocol (HTTP) 18

HTTP basics 18 ▪ *GET versus POST* 21

2.2 Java servlets 23

How a web server uses servlets 24 ▪ *The anatomy of a servlet* 24 ▪ *A servlet example* 26

3 *First steps* 30

3.1 Simple text 31

3.2 Dynamic content 32

Conditional logic 33 ▪ *Iteration* 34

Non-HTML output 37

3.3 Processing requests and managing sessions 38

Accessing request parameters 38 ▪ *Using sessions* 39

3.4 Separating logic from presentation 41

Reusing logic with JavaBeans 42

Abstracting logic with custom tags 44

3.5 Review of examples 45

4 *How JSP works* 46

4.1 The structure of JSP pages 47

Directives and scripting elements 47

Standard and custom actions 48

4.2 Behind the scenes 52

Translation to servlets 52 ▪ *Translation versus execution* 54

4.3 What the environment provides 56

Automatic servlet generation 56 ▪ *Buffered output* 57

Session management 59 ▪ *Exception handling* 63

Implicit objects 64 ▪ *Support for JavaBeans and HTML forms* 64

5	<i>Programming JSP scripts</i>	65
5.1	Scripting languages	66
5.2	JSP tags	68
5.3	JSP directives	68
	<i>Page directive</i>	68 ■ <i>Include directive</i>
	<i>Tag library directive</i>	82
5.4	Scripting elements	83
	<i>Declarations</i>	84 ■ <i>Expressions</i>
	<i>Scriptlets</i>	91
5.5	Flow of control	93
	<i>Conditionalization</i>	93 ■ <i>Iteration</i>
	<i>Exception handling</i>	94 ■ <i>A word of caution</i>
	97	
5.6	Comments	97
	<i>Content comments</i>	98 ■ <i>JSP comments</i>
	<i>Scripting language comments</i>	99

6	<i>Actions and implicit objects</i>	101
6.1	Implicit objects	102
	<i>Servlet-related objects</i>	104 ■ <i>Input/Output</i>
	<i>Contextual objects</i>	112 ■ <i>Error handling</i>
	120	
6.2	Actions	121
	<i>Forward</i>	122 ■ <i>Include</i>
	<i>Plug-in</i>	128
	<i>Bean tags</i>	128

7	<i>Using JSP components</i>	129
7.1	The JSP component model	130
	<i>Component architectures</i>	130 ■ <i>Benefits of a</i>
	<i>component architecture</i>	131 ■ <i>Component design</i>
	<i>for web projects</i>	132 ■ <i>Building applications</i>
	<i>from components</i>	133

- 7.2 JavaBean fundamentals 135
 - The different types of JavaBeans* 138
 - 7.3 JSP bean tags 140
 - Tag-based component programming* 140 ■ *Accessing JSP components* 142 ■ *Initializing beans* 150
 - Controlling a bean's scope* 157
-

8 *Developing JSP components* 165

- 8.1 What makes a bean a bean? 166
 - Bean conventions* 166 ■ *The bean constructor* 167
 - Defining a bean's properties* 168 ■ *Indexed properties* 172 ■ *Implementing bean properties as cursors* 176 ■ *Boolean properties* 178 ■ *JSP type conversion* 179 ■ *Configuring beans* 181
 - 8.2 Some examples 182
 - Example: a TimerBean* 182
 - A bean that calculates interest* 184
 - 8.3 Bean interfaces 189
 - The BeanInfo interface* 189 ■ *The Serializable interface* 190 ■ *The HttpSessionBindingListener interface* 190
 - Other features of the Bean API* 191
 - 8.4 Mixing scriptlets and bean tags 192
 - Accessing beans through scriptlets* 192
 - Accessing scriptlet created objects* 193
-

9 *Working with databases* 198

- 9.1 JSP and JDBC 199
 - JNDI and data sources* 200 ■ *Prepared statements* 201

- 9.2 Database driven JSPs 202
 - Creating JSP components from table data* 202
 - JSPs and JDBC data types* 205 ■ *Maintaining persistent connections* 208 ■ *Handling large sets of results* 211
 - Transaction processing* 216
 - 9.3 Example: JSP conference booking tool 217
 - Project overview* 217 ■ *Our database* 218
 - Design overview* 218
-

10 Architecting JSP applications 229

- 10.1 Web applications 230
 - Web application flow* 232
 - Architectural approaches* 233
- 10.2 Page-centric design 233
 - Role-based pages* 233 ■ *Managing page flow with action targets* 236 ■ *Building composite pages* 238
 - Limitations of the page-centric approach* 241
- 10.3 Servlet-centric design 242
 - Hello, World—with servlets* 243 ■ *JSP and the servlet API* 244 ■ *Servlets for application control* 247
 - Servlets for handling application logic* 248 ■ *Servlets as single entry points* 249 ■ *Handling errors in the servlet* 252 ■ *Example: servlet-centric employee browser* 253 ■ *EmployeeBean* 255
 - FetchEmployeeServlet* 258 ■ *JSP employee list* 261
 - JSP page viewer* 262
- 10.4 Enterprise JavaBeans 263
 - What are Enterprise JavaBeans?* 263 ■ *JavaBeans vs. EJBs* 264 ■ *Application servers and EJB containers* 264
 - Application design with EJBs* 265

- 10.5 Choosing an appropriate architecture 266
 - Application environment* 267 ■ *Enterprise software requirements* 268 ■ *Performance, scalability, and availability* 269 ■ *Technical considerations* 269
 - Organizational considerations* 270
-

11 An example JSP project 272

- 11.1 An FAQ system 273
 - Project motivations* 273 ■ *Application requirements* 273
 - Application modules* 275
 - Building an FAQ component* 276
 - 11.2 The storage module 278
 - Database schema* 279 ■ *The FaqRepository class* 279
 - Storage module exceptions* 285
 - 11.3 The administration module 286
 - The administration servlet* 287 ■ *The main menu* 293
 - Adding an FAQ* 297 ■ *Deleting an FAQ* 300
 - Updating an FAQ* 306
 - 11.4 The web access module 311
 - The FaqServlet* 312 ■ *Viewing a single FAQ* 313
 - Viewing all the FAQs* 314 ■ *A table of contents view* 315
 - Plain text view* 317
-

12 Introducing filters and listeners 318

- 12.1 Life-cycle event listeners 319
 - Session listeners* 319 ■ *Application listeners* 324
- 12.2 Filters 326
 - How filters work* 327 ■ *Filter classes* 330
 - Wrapper classes* 332
- 12.3 Using filters and listeners 333

13	<i>Applying filters and listeners</i>	334
13.1	Application description	335
13.2	User authentication	337
	<i>User account representation</i>	337 ■ <i>User management interface</i>
	338 ■ <i>User management implementation</i>	339
13.3	Web authentication	341
	<i>Session interactions</i>	341 ■ <i>Login servlet</i>
	344	
	<i>Login pages</i>	350 ■ <i>Content pages</i>
	353	
	<i>Logout servlet</i>	357 ■ <i>Logout pages</i>
	358	
13.4	Access control filters	360
	<i>Authentication filter</i>	361 ■ <i>Role filter</i>
	364	
13.5	Logging listener	368
	<i>HttpSessionListener methods</i>	369
	<i>HttpSessionAttributeListener methods</i>	369
13.6	Content filter	372
	<i>Filter methods</i>	373 ■ <i>Response wrapper inner class</i>
	375	
	<i>Output stream inner class</i>	376 ■ <i>More filter methods</i>
	377	
	<i>Filter results</i>	380 ■ <i>Other content filters</i>
	381	

14	<i>Deploying JSP applications</i>	384
14.1	This means WAR	385
	<i>WAR is XML</i>	386 ■ <i>Waging WAR</i>
	389	
14.2	The art of WAR	390
	<i>WAR materiel</i>	390 ■ <i>Drafting deployment descriptors</i>
	396	
14.3	Maintaining a WAR footing	415

15

Performing common JSP tasks 418

- 15.1 Handling cookies 419
 - Managing cookies* 419 ▪ *The Cookie class* 420
 - Example 1: setting a cookie* 421
 - Example 2: retrieving a cookie* 422
- 15.2 Creating error pages 425
 - An erroneous page* 426 ▪ *Data collection methods* 427
 - Sending electronic mail* 432 ▪ *The error page* 433
- 15.3 Mixing JSP and JavaScript 437
- 15.4 Building interactive interfaces 441
 - Sticky widgets* 441 ▪ *Utility methods* 442
 - The example form* 443 ▪ *Setting up the form* 445
 - Text and hidden fields* 446 ▪ *Text areas* 447
 - Radio buttons* 447 ▪ *Select boxes* 448
 - Check boxes* 448 ▪ *Form source* 449
- 15.5 Validating form data 451
 - Client- and server-side validation* 451
 - Example: server-side validation* 452
- 15.6 Building a shopping cart 458
 - Overview* 459 ▪ *The catalog page* 460
 - ShoppingCartItem and InventoryManager* 460
 - The ShoppingCart bean* 464
 - Displaying the shopping cart* 466
- 15.7 Miscellaneous tasks 467
 - Determining the last modification date* 467
 - Executing system commands* 468

16 *Generating non-HTML content* 470

- 16.1 Working with non-HTML content 471
 - The importance of MIME* 471 ▪ *Controlling the content type* 472 ▪ *Detecting your client* 472
 - Designing multiformat applications* 473
 - Controlling the file extension* 474
- 16.2 Text content formats 475
 - Plain text output* 475 ▪ *WYGIWYG output (what you generate is what you get)* 476
- 16.3 XML documents 477
 - Creating voice XML documents* 479
- 16.4 External content 482
 - JSP style sheets* 483 ▪ *JavaScript* 485
- 16.5 Advanced content formats 487
 - Excel spread sheets* 488 ▪ *Code generation* 489

17 *JSP by example* 493

- 17.1 A rotating banner ad 494
 - The BannerBean* 494 ▪ *Using the bean* 495
- 17.2 A random quote generator 497
 - The QuoteBean* 497 ▪ *Using the bean* 498
- 17.3 The Tell a Friend! sticker 499
 - The sticker* 500 ▪ *The MailForm page* 502
 - Sending the mail* 503
- 17.4 A JSP Whois client 505
 - The Whois protocol* 505 ▪ *Requirements and design considerations* 507 ▪ *The WhoisBean* 507
 - Building the front end* 515

- 17.5 An index generator 517
 - A basic implementation* 518 ▪ *An improved version* 520
 - Going further* 525
 - 17.6 A button to view JSP source 525
 - Displaying the source* 525 ▪ *Limitations of the view source program* 527 ▪ *Adding a view source button \to a page* 527 ▪ *Viewing source through a bookmark* 528
-

18 *Creating custom tags* 529

- 18.1 Role of custom tags 530
- 18.2 How tag libraries work 531
- 18.3 Tag library descriptors 535
 - Library elements* 535 ▪ *Validator elements* 537
 - Listener elements* 538 ▪ *Tag elements* 538
 - Variable elements* 540 ▪ *Attribute elements* 541
 - Example element* 543
- 18.4 API overview 544
 - Tag handlers* 544 ▪ *Tag handler life-cycle* 550
 - Helper classes* 556 ▪ *Auxiliary classes* 559
- 18.5 Example tag library 559
- 18.6 Content substitution 560
- 18.7 Tag attributes 563
- 18.8 Content translation 567
 - URL rewriting* 568 ▪ *HTML encoding* 572
- 18.9 Exception handling 575
- 18.10 To be continued 580

19 *Implementing advanced custom tags* 582

- 19.1 Tag scripting variables 583
 - Example tag* 583 ■ *Scripting variable JavaBean* 585
- 19.2 Flow of control 587
 - Conditionalization* 588 ■ *Iteration* 595
- 19.3 Interacting tags 613
 - Interaction mechanisms* 614 ■ *Index tag* 616
- 19.4 The final ingredient 619

20 *Validating custom tag libraries* 621

- 20.1 Two representations of JSP 622
- 20.2 JSP pages as XML documents 624
 - The root element* 625 ■ *Template text* 626
 - Scripting elements* 627 ■ *Request-time attribute values* 627
 - Directives and actions* 629 ■ *Sample page* 629
- 20.3 Tag library validation 631
- 20.4 Example validators 634
 - Copying validator* 635 ■ *Script-prohibiting validator* 638
 - Error handling* 642 ■ *Content handler* 645
 - Nesting validator* 651
- 20.5 Packaging the tag library 660
 - Packaging a single library* 661
 - Packaging multiple libraries* 662
- 20.6 For further information 666

A *Changes in the JSP 1.2 API* 669

- A.1 Introduction 669
- A.2 Changes to the API 670
 - Java 2, Version 1.2 now a requirement* 670

- Servlet API 2.3 required* 670 ▪ *XML syntax now fully supported* 670 ▪ *Determining the real path* 671
 - Redirects are not relative to the servlet context* 671
 - Restricted names* 671 ▪ *Page encoding attribute* 671
 - Flush on include no longer required* 671
 - A.3 Web application changes 672
 - New 2.3 web application DTD* 672 ▪ *Handling of white space* 672 ▪ *Resolving path names in the web.xml file* 672
 - Request mappings* 672 ▪ *Dependency on installed extensions* 672
 - A.4 Custom tag improvements 673
 - Translation time validation* 673 ▪ *New tag interfaces* 673 ▪ *Changes to the TLD* 673
 - A.5 JavaBean changes 674
 - Bean tags cannot implicitly access scriptlet objects* 674
 - Fully qualified class names required* 674
 - A.6 New servlet features 674
 - Servlet filters* 675 ▪ *Application events* 675
-

B *Running the reference implementation* 676

- B.1 Prerequisites 677
 - B.2 Downloading and installing Tomcat 677
 - B.3 Web applications and Tomcat 681
-

C *Incorporating Java applets* 683

- C.1 Browser support for Java 683
- C.2 The plug-in action 685
 - Required attributes* 685 ▪ *Optional attributes* 687
 - Parameters* 688 ▪ *Fallback text* 689
- C.3 Example: applet configuration 690

D JSP resources 697

- D.1 Java implementations 697
- D.2 JSP-related web sites 697
- D.3 JSP FAQs and tutorials 698
- D.4 JSP containers 698
- D.5 Java application servers with JSP support 699
- D.6 JSP development tools 700
- D.7 Tools for performance testing 700
- D.8 Mailing lists and newsgroups 700

E JSP syntax reference 702

- E.1 Content comments 702
- E.2 JSP comments 703
- E.3 `<jsp:declaration>` 704
- E.4 `<jsp:directive.include>` 705
- E.5 `<jsp:directive.page>` 706
- E.6 `<jsp:directive.taglib>` 707
- E.7 `<jsp:expression>` 708
- E.8 `<jsp:forward>` 709
- E.9 `<jsp:getProperty>` 710
- E.10 `<jsp:include>` 711
- E.11 `<jsp:plugin>` 712
- E.12 `<jsp:scriptlet>` 713
- E.13 `<jsp:setProperty>` 714
- E.14 `<jsp:useBean>` 715

F *JSP API reference* 718

F.1 JSP implicit objects 719

F.2 Package javax.servlet 719

Interface Filter[†] 719 ■ *Interface FilterChain*[†] 719

Interface FilterConfig[†] 720 ■ *Class GenericServlet* 720

Interface RequestDispatcher 720 ■ *Interface servlet* 721

Interface ServletConfig 721 ■ *Interface ServletContext* 721

Interface ServletContextAttributeEvent[†] 722

Interface ServletContextAttributeListener[†] 722

Interface ServletContextEvent[†] 722

Interface ServletContextListener[†] 723

Class ServletException 723 ■ *Class ServletInputStream* 723

Class ServletOutputStream 724

Interface ServletRequest 724

Class ServletRequestWrapper[†] 725

Interface ServletResponse 726

Class ServletResponseWrapper[†] 726

Interface SingleThreadModel 727

Class UnavailableException 727

F.3 Package javax.servlet.http 727

Class cookie 727 ■ *Class HttpServlet* 728

Interface HttpServletRequest 729

Class HttpServletRequestWrapper[†] 730

Interface HttpServletResponse 730

Class HttpServletResponseWrapper[†] 732

Interface HttpSession 733

Interface HttpSessionActivationListener[†] 733

Interface HttpSessionAttributeListener[†] 733

Class HttpSessionBindingEvent 734

	<i>Interface HttpSessionBindingListener</i>	734
	<i>Class HttpSessionEvent[†]</i>	734
	<i>Interface HttpSessionListener[†]</i>	735
	▪ <i>Class HttpUtils</i>	735
F.4	Package javax.servlet.jsp	735
	<i>Interface HttpJspPage</i>	735
	▪ <i>Class JspEngineInfo</i>	736
	<i>Class JspException</i>	736
	▪ <i>Class JspFactory</i>	736
	<i>Interface JspPage</i>	737
	▪ <i>Class JspTagException</i>	737
	<i>Class JspWriter</i>	737
	▪ <i>Class PageContext</i>	738
F.5	Package javax.servlet.jsp.tagext	740
	<i>Class BodyContent</i>	740
	▪ <i>Interface BodyTag</i>	740
	<i>Class BodyTagSupport</i>	740
	▪ <i>Interface IterationTag[†]</i>	741
	<i>Class PageData[†]</i>	741
	▪ <i>Interface Tag</i>	741
	<i>Class TagAttributeInfo</i>	742
	▪ <i>Class TagData</i>	742
	<i>Class TagExtraInfo</i>	743
	▪ <i>Class TagInfo</i>	743
	<i>Class TagLibraryInfo</i>	744
	<i>Class TagLibraryValidator[†]</i>	744
	<i>Class TagSupport</i>	744
	▪ <i>Class TagVariableInfo[†]</i>	745
	<i>Interface TryCatchFinally[†]</i>	745
	▪ <i>Class VariableInfo</i>	745