

Index

■ A

- AbstractFacade class, 412–413, 416, 431
- AbstractItemWriter class, 669
- acceptChanges method, 363
- Accessor methods, 62
- acmeFileProcessor.xml file, 664
- AcmeProcessor class, 666
- addListener method, 214
- activationConfig element, 446
- addCookie method, 44
- addListener methods, 20
- Add Server Instance, 672
- afterPhase method, 295, 297
- Ajax
 - asynchronous updates, 289, 292
 - custom conversion, 274
 - custom processing, 269–271
 - functionality, 265–269
 - partial-page updates, 263–265
 - submitting pages, 262–263
 - validation
 - Acme Bookstore application, 256
 - client-side validation, 260
 - jsf.ajax.request() method, 261
 - third-party component library, 260
- AjaxBean managed bean, 254
- AjaxBehaviorEvent object, 261
- Ajaxified, 259
- @Alerter annotation, 593
- Apache Tomcat, 7
- Application server administrators, 650
- ArrayList<Callable>, 657
- asadmin create-managed-executor-service
 - command, 647, 650
- asadmin create-managed-thread-factory command, 664
- AsyncContext.complete method, 51
- Asynchronous consumer, 529
- asyncOperation method, 594

- AsyncResponse resume method, 595
- AsyncResponse.setTimeout() method, 595
- AsyncResponse.setTimeoutHandler() method, 595
- attributeAdded method, 26
- attributeRemoved method, 26
- attributeReplaced method, 26–27
- AuthorController class, 298
- AuthorController managed bean, 247
- AuthorTask, 656
- AuthorTaskTwo, 656
- AuthorWorkController, 424
- AuthorWorkType business interface, 423–424
- automaticTimer() method, 437
- Available Entity Classes, 580

■ B

- Batch applications
 - acmeFileProcessor.xml file, 664
 - AcmeProcessor class, 666–667
 - acmeReader, 665
 - acmeWriter, 665
 - <chunk> element, 668
 - fine-grained experience, 647
 - ItemProcessor, 669
 - ItemWriter class, 669
 - Java EE, 668
 - Java Platform API, 664
 - JobOperator, 668
 - META-INF directory, 665
 - processItem method, 669
 - <processor> element, 668
 - <properties> element, 668
 - readItem method, 668
 - WidgetOutputItem object, 666–667
 - WidgetReportItem object, 665–667
 - writeItems method, 669
- beforePhase method, 295–296
- BOOK database table, 350

BookChatEndpoint class, 606
 bookChatRelay function, 606
 BookController class, 425, 427
 BookController JSF managed bean, 351
 BookController method, 433
 bookDao.queryBooks method, 350
 BookFacade, 431
 BookFacade class, 425
 BookFacade session bean, 429
 BookstoreAppListener, 283
 BookstoreSessionController bean, 435
 BookstoreSessionCounter bean, 434
 book.xhtml view, 230
 bufferedAmount attribute, 609
 builderTasks, 657
 buildPost method, 590
 BytesMessage, 526

■ C

CachedRowSet object, 358, 361–362
 CallableStatement, 354
 cancel method, 654
 CartController class, 275
 Cascading Style Sheets (CSS), 159
 Certificate signing request (CSR), 562
 Character large object (CLOB) data, 355
 CheckPasswordValidator class, 301
 <chunk> element, 668
 CLASSPATH, 318, 671
 ClientRequestFilter, 592
 ClientResponseFilter, 592
 Client setProperty method, 587
 commandButton, 254
 commandLink component, 262
 completeBookList property, 430
 completeMethod, 299
 ComponentSystemEvent object, 286
 Concurrency
 builder task, 655–656
 concurrent/BatchExecutor, 651
 java.lang.Thread, 647
 java.util.concurrent, 647
 java.util.concurrent.ExecutorService, 647
 javax.concurrent.
 ManagedScheduledExecutorService
 interface, 661
 ManagedExecutorService (*see* ManagedExecutor
 Service)
 ManagedScheduledExecutorService, 661
 ManagedScheduleExecutorService, 659
 run method, 660
 scheduleAtFixedRate method, 662
 ScheduleExecutorService interface, 662
 standard techniques, 647
 web.xml deployment descriptor, 651

@ConcurrencyManagement annotation, 436
 concurrent/BatchExecutor, 651
 concurrent/BuilderExecutor, 655
 concurrent/YourExecutorServiceName, 650
 Connection resources, 521
 ContactController class, 205, 216
 ContainerRequestFilter, 592
 ContainerResponseFilter, 592
 Contexts and dependency injection (CDI)
 alternative implementations, 513
 bean metadata, 514
 bean scope determination, 506
 beans.xml file, 499
 CalculationBean class, 498
 ignoring classes, 512
 JavaEERecipesJPA project, 497
 JSF views
 JavaBean, 499
 javax.inject.Named (@Named)
 annotation, 501
 @Named annotation, 499
 @ManagedBean *vs.* @Named annotation, 502
 non-bean objects, 508
 producer fields, disposing of, 513
 specific bean allocation, 503
 Cookie property methods, 44
 COUNT aggregate function, 458
 CreateConnection class, 328
 createContainerEntityManagerFactory
 method, 405
 createEntityManagerFactory method, 405
 create-managed-executor-service
 command, 648
 create-managed-thread-factory command, 662
 createParser() method, 613
 createQuery method, 449
 Create, retrieve, update, and delete (CRUD) database
 code format, 334
 DML, 334
 java.sql.Statement object, 332
 performCreate method, 334
 performDelete method, 335
 performUpdate method, 334
 toUpperCase method, 333
 CriteriaBuilder methods, 448–449
 CriteriaQuery, 447
 currDateAndTime field, 14
 Custom annotation class
 CheckPasswordValidator class, 301
 @Constraint, 302
 ConstraintValidator, 303
 @Documentation, 303
 @Documented, 302
 @Inherited, 302
 inputSecret component value, 300
 @interface, 302–303

- isValid method, 304
- PasswordLength, 300, 304
- @Retention, 302–303
- syntactic metadata, 302
- @Target, 302
- validator class, 300

D

- Data access object (DAO)
 - application code base, 347
 - AUTHOR database table, 341
 - AuthorDAO class, 347
 - Author object, 347
 - classes, 335
 - Java enterprise object, 347
 - performCreate method, 347
 - performDelete method, 347
 - performUpdate method, 347
- Data definition language (DDL), 406
- Data Manipulation Language (DML), 334
- DataSource object, 319
- dataTable component, 203, 245, 249, 350
- Date() object, 292
- dayAndTime property, 292
- db_props.properties file, 328, 330
- @DELETE, 584
- DisplayCookieServlet, 44
- @Documentation annotation, 303
- doDownload method, 34
- doFilter method, 24
- doGet method, 6, 14
- doPost method, 6, 17
- DriverManager, 323
- DriverManager.getConnection() method, 323
- DUMMY_PROC procedure, 354
- Durable message subscribers
 - create and publish message, 533
 - delay message, 536
 - initial durable subscriber, 533
 - receive message, 534
 - run application, 535
 - string-based identifier, 536
 - Topic connection, 532, 535
 - unsubscribing, 535–536

E

- EJB 3.0, 373
- Employee entity, 589
- Enterprise archive (EAR), 378, 483
- Enterprise JavaBeans (EJBs)
 - application server container, 409
 - asynchronous message process, 444
 - data

- BookViewNode, 631–632
- build() method, 633
- create() method, 633
- FXCollections.observableArrayList() method, 634
- SceneBuilder statement, 633–634
- start() method, 633
- TableView, 629–630
- TabPane, 632
- dataTable model
 - AbstractFacade class, 431
 - ArrayList of HashMaps, 433
 - BookController, 430–431
 - BookController.getCustomBookList() method, 432
 - completeBookList field, 433
 - customBookList field, 433
 - customBookList property, 432
 - findAll() method, 433
 - List of entity objects, 430, 433
 - List of Map objects, 432–433
 - obtainCustomList method, 432
- Entity Manager, 409
 - local and remote interfaces, 443
 - object persistence, 429
 - object updation, 429
 - optional transaction (*see* Optional transaction life-cycle callbacks)
 - query language, 409
 - scheduling timer service
 - automatic timer, 437
 - calendar-based timer expressions, 439
 - e-mail, 438
 - org.javaee.recipes.jpa.timer.
 - ProgrammaticTimerExample class, 438
 - org.javaee.recipes.jpa.timer.TimerBean file, 437
 - programmatically timer, 438, 440
 - @Schedule annotation, 439
 - ScheduleExpression helper class, 440
 - types, 439
 - session beans with JSF
 - AuthorWorkController, 424
 - AuthorWorkFacade session bean, 423–424
 - BookFacade class, 425
 - EJB 3.1+, 427–428
 - findBookById method, 425
 - HTML client, 427
 - javax.ejb.EJB annotation, 428
 - javax.naming.EJB annotation, 428
 - javax.naming.InitialContext interface's
 - lookup method, 428
 - @Local annotation, 424, 427
 - @Remote annotation, 427
 - web-based client development, 422

Enterprise JavaBeans (EJBs) (*cont.*)
 singleton bean (*see* Singleton bean)
 stateful session bean (*see* Stateful session beans)
 stateless session bean (*see* Stateless session beans)
 technology, 370
 types, 409

Enterprise solution
 Groovy servlet
 BookstoreServletAction.groovy, 601
 BookstoreServlet.groovy file, 598
 groovy-all-xxx.jar file, 600
 groovy.servlet.GroovyServlet, 599
 MarkupBuilder, 599
 ParameterExample.groovy, 600
 print statement, 600
 sql.eachRow method, 598

Jython servlet
 BookstoreJython.py servlet, 602
 convertHTML function, 603
 doGet and doPost, 603
 org.python.util.PyServlet, 602
 response.getWriter(), 603

Entity Bean classes, 678
 EntityBSubtype, 468
 EntityExistsException, 429
 Entity Manager, 409
 equals() method, 374
 executeQuery method, 330–331, 340
 executeUpdate method, 341

■ F

Facelets, 103
 CSS, 159
 custom resolver, 197–198
 debugging view content, 195–197
 definition language, 159
 handling variable-length data
 Acme Bookstore application, 188
 AuthorController, 191–192, 194–195
 data collection, 187–188
 displayAuthor method, 191
 markup and JSF tag, 190
 object collection, 190
 recipe04_05c.xhtml, bio view, 188–189
 ui:repeat tag, 188–190

JSF 2.0+ application, 159
 page template creation
 composition, 162
 CSS style sheet, 161
 custom_template.xhtml, 160
 JSF managed bean, 161
 mitigates issues, 161
 NetBeans, 162–163
 similar and function, 161

tag library, 162
 template window, 163–164
 ui:insert element, 160
 ui:insert tag, 162
 view definition language, 159, 161
 visual representation, 161

resources
 Author table, 176
 css, 175
 directory structure, 174
 h:dataTable component, 176
 JSF component, 177
 libraries, 177
 URL creation, 174

reusable template component
 Acme bookstore layout, 185
 AuthorController.java, 180–183
 composite component, 186
 custom_template_search.xhtml, 184
 default implementation method, 186–187
 @ManagedProperty annotation, 187
 opening and closing tag, 186
 SearchController.java, 178–180
 searchText and errorText property, 187
 search.xhtml component, 178
 search.xhtml document, 186
 ui:component tag, 185
 XHTML document, 177
 XML namespace, 187

template view
 application, 171
 AuthorController, 168, 170
 page control and template tags, 172
 recipe04_01a.xhtml, 165–166
 recipe04_01b.xhtml, 166–167
 recipe04_01c.xhtml, 167–168
 template clients, 171
 ui:composition tag, 164
 ui:define tag, 173–174

Tiles, 159

faces-config.xml file, 284, 308
 facesFlowScope, 314

f:ajax tag
 action component, 263
 action listener, 261
 Ajaxified, 259, 261
 Ajax validation, 259
 attributes, 260
 execute keywords, 264
 execute and render attributes, 261
 functionality, 269
 JSF arsenal, 260
 partial-page updates, 264

FETCH join, 469
 f:event tag, 285

f:facet tag, 251
 findBookByChapterTitle method, 460–461
 findBookById method, 425
 findBooksByAuthor method, 450
 findByTitle method, 413
 <flow-definition> section, 308
 @FlowScoped, 312
 flush method, 35
 forward method, 40
 f:phaseListener tag, 296
 fterPhase method, 296
 f:validateBean tag, 225
 f:validateDoubleRange validator, 224
 f:validateLength, 221
 f:validateLongRange, 224
 f:validateRegex validator, 224
 f:validateRequired validator, 224

G

GenericServlet, 5, 7
 @GET, 584
 getAsciiStream method, 358
 getAsObject method, 272, 274
 getAsString method, 272, 274
 getAttribute() method, 29
 getBookCategories method, 467
 getCart() method, 418
 getCompleteBookList method, 349, 433
 getConnection method, 319, 323
 getParameter method, 17
 getPhaseId method, 295–296
 getPort method, 576
 getRequestDispatcher method, 39
 getResourceAsStream method, 34
 getResultList method, 451
 getServletConfig().getInitializationParameter(), 22
 getSession method, 29
 getSingleResult method, 452, 458
 getUserMessage method, 584
 Glassfish deployment directory, 7
 Glassfish JDBC connection pools, 320
 Glassfish lib directory, 318
 GlassFish Server Administration, 649
 Glassfish v4 application server, 447
 graphicImage component, 219, 230
 Groovy servlet
 BookstoreServletAction.groovy, 601
 BookstoreServlet.groovy file, 598
 groovy-all-xxx.jar file, 600
 groovy.servlet.GroovyServlet, 599
 MarkupBuilder, 599
 ParameterExample.groovy, 600
 print statement, 600
 sql.eachRow method, 598

H

hashCode() and equals() methods, 386
 HashMap object, 433
 h:dataTable component, 307
 h:graphicImage tag, 230
 h:outputLink tags, 218
 HTML5 project, 679–681
 HTML5 technology, 605
 HttpServlet methods, 6
 HttpServletRequest object, 6
 HttpServletResponse sendRedirect method, 40
 HttpSessionAttributeListener, 25–26
 HttpSessionBindingEvent, 26
 HttpSessionListener, 20
 HttpSession object, 31

I

IllegalArgumentException, 429–430
 init method, 6
 Inner join, 461, 469
 inputFile component, 253
 inputHidden component, 207
 inputSecret component, 207
 inputTextarea component, 207
 inputText component, 207
 Integrated development environment (IDE), 97, 671
 @interface declaration, 303
 Invocation, 590
 InvocationCallback, 596
 isBookInCart, 285
 isCanceled method, 654
 isDone method, 654
 isListenerForSource, 285
 isListenerForSource(Object source) method, 283–284
 ItCategory entity, 467–468
 ItemProcessor class implementation, 669
 ItemWriter class implementation, 669

J, K

JavaBeans, 97
 javac command, 7
 javac command-line utility, 5
 java:comp/DefaultManagedExecutorService, 651
 Java Database Connectivity (JDBC), 369
 acceptChanges method, 363
 API, 369
 CachedRowSet object, 358–362
 calling PL/SQL stored procedures, 354
 CLASSPATH, 318
 connection management
 code intensive, 329
 CreateConnection class, 328

- Java Database Connectivity (JDBC) (*cont.*)
 - CreateConnection.java, 325
 - db_props.properties, 328
 - DriverManager, 329
 - getDsConnection, 329
 - org.javaerecipes.chapter07, 326
- CRUD database (*see* Create, retrieve, update, and delete (CRUD) database)
- database connection
 - DataSource Classname, 320
 - DataSource object, 319, 322–323
 - DriverManager, 323
 - DriverManager.getConnection() method, 323
 - getConnection() method, 319
 - Glassfish, connection pool, 320
 - InitialContext, 324
 - Oracle, 319
 - properties, 321
 - resource creation, 322
 - String, 323
- database driver, 317–318
- JSF views
 - bookDao.queryBooks method, 350
 - BOOK database table, 348, 350
 - dataTable component, 348–351
 - getCompleteBookList method, 349
 - queryBooks method, 350
- moveToInsertRow method, 363
- PL/SQL stored procedures, 353
- query, 329–330
- querying and storing
 - CLOB data, 355, 357
 - createClob method, 358
 - getAsciiStream method, 358
 - performCreate method, 358
 - readClob method, 355
 - string values, 357
 - VARCHAR fields, 357
- RowSet objects (*see* RowSet objects)
- scrollable ResultSets (*see* Scrollable ResultSets)
- SQL injection (*see* SQL injection)
- SyncProviderException, 363
- try-catch block, 324–325
- JavaEERecipes application, 17, 580
- JavaEERecipes project, 578
- JavaEERecipesJPA.war file, 447
- JavaEERecipesJTA persistence unit, 415
- JavaEERecipes.war file, 1
- JavaEERecipes web.xml configuration file, 564
- Java Enterprise Applications, 673
- Java Enterprise platform, 647
- JavaFX
 - Button node, 620
 - database
 - BookView class, 626
 - cellValueFactory, 628
 - EntityManagerFactory, 628
 - findAllBooks() method, 628
 - getColumns().addAll() method, 629
 - observableArrayList method, 628
 - org.javaerecipes.javafx.entity.
 - Book, 621–624
 - org.javaerecipes.javafx.session.
 - AbstractFacade, 624–625
 - org.javaerecipes.javafx.session.
 - BookFacade, 624
 - persistence.xml, 628
 - PropertyValueFactory class, 629
 - TableColumn, 629
 - TableView, 621
 - EJB data
 - BookViewNode, 631–632
 - build() method, 633
 - create() method, 633
 - FXCollections.observableArrayList()
 - method, 634
 - SceneBuilder statement, 633–634
 - start() method, 633
 - TableView, 629–630
 - TabPane, 632
 - EventHandler, 620
 - “Hello World” application, 619
 - HelloWorld JavaFX main class, 618–619
 - JavaEERecipesFX, 615–616
 - javafx.application.Application, 620
 - javafx.scene.Node class, 620
 - javafx.scene.Scene object, 620
 - javafx.stage.Stage object, 620
 - Label node, 620
- MVC
 - AcmeBookstoreMainController
 - class, 637–638
 - acme_bookstore_main.fxml, 636
 - <AnchorPane> element, 636–637
 - fx:controller, 640
 - FXML, 638–639
 - initialize() method, 640
 - javafx.scene.control.Label node, 639
 - javafx.scene.layout.AnchorPane
 - class, 639
 - java.lang package, 639
- NetBeans New Project dialog, 616–617
- NetBeans Platform Manager, 619
- NetBeans 7.x, 619
- New JavaFX Application dialog, 617
- REST services
 - acme_bookstore_rest.fxml, 643
 - clientLabel, 645
 - createTask method, 645
 - EventHandler<WorksrStateEvent>, 645
 - javafx.application.Application
 - class, 644

- obtainData method, 645
 - RESTFulCientController, 643–644
 - service client, 641–642
 - user interface, 640–641
- setOnAction() method, 620
- setText() method, 620
- start method, 619
- VBox node, 620
- java.lang.Thread, 647
- Java Message Service (JMS), 409
 - create and send message
 - run application, 525
 - using simplified API, 525–526
 - using standard API, 524
 - durable subscribers (*see* Durable message subscribers)
 - message filtering
 - run application, 530
 - sendMessage1 method, 529
 - sendMessage2 method, 529
 - string-based expressions, 531
 - queues
 - browseMessages() method, 531
 - QueueBrowser object, 532
 - run application, 532
 - receive message
 - run application, 528
 - using simplified API, 528
 - using standard API, 528
 - using standard JMS API, 527
 - resources
 - connection factory resources, 521
 - destination resources, 522
 - using NetBeans IDE, 518
 - within GlassFish application server environment, 520
 - session
 - connection, 523
 - createSession method, 523
 - message acknowledgment, 524
 - nontransactional, 524
 - run application, 523
- java.naming.NamingException, 324
- Java Persistence API (JPA), 369, 447
 - EclipseLink, 369–370
 - Hibernate, 369–370
 - KODO, 369
 - OpenJPA, 369
 - TopLink, 369, 370
 - translating code, 369
- Java Persistence Query Language (JPQL), 401
 - aggregate functions, 458
 - bulk updates and deletes
 - Criteria API, 465–466
 - CriteriaBuilder, 466
 - Employee entity, 465
 - EmployeeSession bean, 465
 - javax.persistence.criteria.CriteriaDelete object, 466
 - javax.persistence.criteria.CriteriaUpdate object, 466
 - MetaModel API, 467
 - org.javaeerecipes.jpasession.EmployeeSession.java class, 465
 - CriteriaQuery, 447
 - database stored procedures, 459
 - entity subclasses, 467
 - filtering query results, 450
 - functional expressions
 - arithmetic functions, 464
 - authorList, 463
 - datetime functions, 464
 - findAuthorByLast method, 463
 - HAVING clause, 463
 - org.javaeerecipes.jpasession.BookAuthorFacade class, 463
 - string functions, 463
 - strings, 462
 - WHERE clause, 462–463
 - joining entities
 - instances retrieval, 460
 - rows retrieval, 461
 - more than one entity
 - authorBooks SqlResultSetMapping, 456
 - @ColumnResult annotation, 457
 - @FieldResult annotation, 457
 - HashMap, 455, 457
 - JSF dataTable source, 456
 - Object[] list, 457
 - org.javaeerecipes.jpasession.entity.BookAuthor entity class, 454
 - org.javaeerecipes.jpasession.BookAuthorFacade session bean, 455
 - @SqlResultSetMapping annotation, 454, 456–457
 - string-based key, 457
 - native queries
 - createNativeQuery method, 452–453
 - database records, 452
 - dynamic query, 454
 - getResultSet method, 453
 - javax.persistence.Query methods, 452
 - @NamedNativeQuery annotation, 452–453
 - org.javaeerecipes.jpasession.entity.Book entity, 453
 - org.javaeerecipes.jpasession.BookFacade bean, 453
 - SqlResultSetMapping, 453–454
 - ON conditions, 468
 - query execution, 464
 - querying all entity instances

Java Persistence Query Language (JPQL) (*cont.*)

- BookAuthor entity, 448–449
- createQuery method, 448
- CriteriaQuery object, 448–449
- findAuthor method, 448
- FROM clause, 449
- javax.persistence.Query interface
 - methods, 449
- record, 448
- ResultList, 448
- SELECT clause, 449
- TypedQuery object, 450
- returning single object, 451
- JavaScript Object Notation (JSON), 605
- JavaServer Faces (JSF)
 - Ajax(*see* Ajax)arithmetic and reserved words
 - EvaluationController, 147
 - JSF EL expressions, 146
 - working principle, 148
- bean actions invoking, 157
 - f:viewAction facet, 157
 - JSF Life-Cycle Phases, 158
 - onPostback attribute, 158
 - preRenderView, 157
 - process validation phase, 158
 - String validateUser, 158
- bookmarkable URLs
 - authorList property, 151
 - f:viewParam tag, 150
 - org.javaeerecipes, 151
 - working principle, 152
- check box
 - attributes, 234
 - false value, 231
 - managed bean controllers, 232
 - populateNotificationTypes method, 235
 - recipe05_07.xhtml, 232
 - selectBooleanCheckbox component, 234–235
 - selectManyCheckbox component, 234–235
 - true value, 234
- component and tag primer
 - attributes, 201–202
 - binding components, 203
 - core tags, 201
 - HTML, 199
- data collection
 - attributes, 249
 - dataTable component, 245, 249, 251
 - f:facet tag, 251
 - h:column Attributes, 250
 - managed bean, 247–248
 - recipe05_10.xhtml, 245
 - styles.css sheet, 247
- evolution, 97
- FacesServlet, 97

- file upload component, 253
- graphics, 230–231
- information message display
 - h:message component, 119
 - Java EE 7 Recipes, 119
 - MessageController, 119
 - working principle, 121
- input form creation
 - h:form tags, 207
 - input component tag attributes, 208
 - inputHidden component, 208
 - inputSecret component, 208
 - inputText component, 208
 - inputTextarea component, 208
 - managed bean, 205–207
 - recipe05_01.xhtml, 204
- invoke action methods
 - action attribute, 214
 - actionListener method, 214
 - commandButton component, 209, 212
 - commandLink component, 209, 212–213
 - managed bean, 210
 - recipe05_02.xhtml, 209
- JavaBeans, 97
- library, 252
- managed bean writing, 103
 - CalculationController, 104
 - JSF view, 107
 - Plain Old Java Object, 104
 - scopes, 109
 - string list creation, 104
 - working principle, 108
- message updation
 - FacesContext, 130
 - manage bean, 129
 - resource bundle creation, 129
 - working principle, 131
- navigation, 122
 - action attribute, h:commandButton
 - components, 124
 - conditional navigation, 124
 - faces-config.xml file, 127
 - h:commandButton components, 123
 - implicit navigation, 125
 - NavigationController, 123, 125
 - techniques, 123
 - working principle, 128
- NetBeans
 - downloading and installing, 100
 - HelloWorldController, 101
 - index.xhtml file, 101
- object list display
 - h:dataTable components, 153
 - style sheet source, 154
 - working principle, 156

- output components
 - attributes, 217
 - graphicImage component, 219
 - link Component Additional Attributes, 220
 - managed bean, 215, 217
 - outputFormat component, 218
 - outputLabel additional attributes, 219
 - outputLabel component, 217
 - outputLink additional attributes, 218
 - outputLink componenet, 217
 - outputLink and outputLabel components, 218
 - outputText, 217
 - outputText component, 218
 - recipe05_03.xhtml, 214
- page expression evaluation, 139
 - immediate attribute, 139
 - recipe03_08.xhtml, 139
 - working principle, 141
- page layout
 - Cascading Style Sheets, 242
 - panelGrid components, 238, 245
 - panelGroup component, 238, 244
 - recipe05_09.xhtml, 238
 - ValueExpressions, 243
- page pass parameters
 - ArrayList, 143
 - Author class, 144
 - displayAuthor method, 142
 - JSF EL expression, 141
 - working principle, 145
- radio button, 236, 238
- request-driven, 97
- selection
 - component attributes, 229
 - f:selectItem or f:selectItems tags, 229
 - itemValue and itemLabel attributes, 229
 - LinkedHashMap, 229
 - managed bean, 227
 - recipe05_05.xhtml, 226
 - SelectItem objects, 230
 - selectManyListbox component, 226, 230
 - selectManyMenu component, 226, 230
 - selectOneListbox component, 226, 230
 - selectOneMenu component, 226
- simple JSF application
 - creation, 97
 - Java IDE, 98
 - JSF managed bean examination, 99
 - JSF managed bean field value display, 98
 - pre-JSF 2.0 environment, 100
- sophisticated user interface, 110
 - AuthorController, 112
 - form id, 117
 - h:commandButton component, 118
 - h:dataTable, 118
 - h:graphicImage tag, 117
 - h:inputText component, 118
 - h:outputLabel tag, 117
 - working principle, 117
 - XHTML, 110
- user input validation
 - custom validator, 133
 - Employee class, 136
 - JSF managed bean, 134
 - JSF validator, 132
 - tags, 132
 - working principle, 138
- validation
 - constraint annotations, 224
 - email property, 222
 - FacesContext, 226
 - f:validateBean tag, 225
 - f:validateDoubleRange validator, 224
 - f:validateLength, 221, 224
 - f:validateLongRange, 224
 - f:validateRegex validator, 224
 - f:validateRequired validator, 224
 - javax.faces.validator.ValidatorException, 226
 - recipe05_04.xhtml, 221
 - UIComponent, 226
 - validatePassword method, 222-223
 - validator attribute, 222
- working principle, 102
 - Apply Request Values phase, 103
 - Facelets, 102-103
 - Invoke Applications phase, 103
 - JSF managed bean, 103
 - JSF parts, 102
 - Process Validations phase, 103
 - Render Response phase, 103
 - restore view, 102
 - Update Model Values phase, 103
 - web.xml file, 103
- JavaServer pages (JSP), 1
 - business logic separation
 - field's getter method, 58
 - <jsp:useBean> element, 60
 - jsp:useBean element scopes, 60
 - RandomBean, 58-59
 - conditional expression
 - code, 64
 - and expression, 67
 - functions.tld file, 65-66
 - isPrimitive function, 65-67
 - <jsp:useBean>, 63
 - JSTL, 67
 - static modifier, 66
 - String value, 63
 - taglib directive, 66

JavaServer pages (JSP) (*cont.*)

- TLD, 63
- typename field, 65
- custom tag creation
 - authorName attribute, 77
 - custom.tld TLD, 80
 - cust:signature element, 79
 - default signature, 77
 - doEndTag() method, 80
 - doStartTag() method, 80
 - doTag method, 79–80
 - getjspContext method, 80
 - <html> element, 80
 - javax.servlet.jsp, 77–78
 - JSP code, 79
 - JSP 2.0 simple tag support, 77
 - mapping, 78
 - MVC architecture, 81
 - PageContext's getOut method, 80
 - short-name element, 80
 - Signature class, 77–78
 - SimpleTagSupport class, 77, 79
 - tag-class element, 80
- document creation, 67–69
- EL expressions
 - arithmetic expressions, 71
 - arithmetic operators, 74
 - #{ and } character sequences, 72
 - <c:if> tags, 73
 - conditional and/or arithmetic expressions, 69
 - conditional logic, 69
 - conditional page rendering, 71
 - implicit objects, 72–73
 - JavaBean property, 72
 - JSTL <c:if> tag, 71
 - JSTL tag library, 69–70
 - language, 71
 - reserved words, 73–74
 - #{ } syntax, 72
 - turn off, 95–96
- embedding Java code, 56–58
- error handling, 91–94
- input form creation, database record, 83–87
- life cycle, 55–56
- parameter accessing, multiple pages, 74–77
- records display, database table
 - AuthorBean class, 89–91
 - Author class, 88–89
 - code, 87–88
 - getAuthorList method, 91
 - Java classes, 87
 - jsp:useBean element, 91
 - JSTL c:forEach element, 87, 91
 - queryAuthors method, 91
 - var attribute, 91

- scriptlets
 - disabling, 94–95
 - embedded Java code, 53, 57–58
- static/dynamic pages embedding, 81–82
- values setting/yielding
 - accessor methods, 62
 - EasyBean, 60, 62
 - EL expressions, 60
 - getFieldValue, 62
 - input form, 61
 - JavaBean class, 61–62
 - jsp:getProperty element, 63
 - jsp:setProperty element, 62
 - jsp:useBean element, 62
 - POST method, 62
 - setFieldValue, 62
- web page development
 - dynamic code, 54
 - getCurrentDate method, 55
 - JavaBean, 54–55
 - .jsp extension, 55
 - <jsp:useBean> tag, 55
- Java servlets. *See* Servlets
- Java specification requests (JSRs), 369
- java.sql.PreparedStatement object, 335
- java.sql.SQLException, 324–325
- Java standard tag library (JSTL), 67
- java.util.concurrent, 647
- java.util.concurrent.Callable, 657
- java.util.concurrent.ExecutorService, 647
- java.util.concurrent.TimeUnit, 595
- java.util.Runnable, 654
- Java Web Project, 673–674
- Java Web Services
 - binding filters and interceptors, 593
 - consuming and producing, REST, 585–586
 - enterprise application development, 563
 - filtering request and responses, 591–592
 - binding filters and interceptors, 593
 - @BindingPriority annotation, 593
 - filters, 592
 - entity interceptors, 592
 - Java EE Project, 577–578
 - JAX-RS, 563
 - JAX-RS Client (*see* JAX-RS Client)
 - JAX-WS, 563
 - JAX-WS Web Service (*see* JAX-WS web service)
 - Jersey JARs, 564
 - RESTful, 564
 - @DELETE, 584
 - entity class, 580–582
 - @GET, 584
 - getUserMessage method, 584
 - Java class, 583
 - JavaEERecipes project, 579

- NetBeans, 580
- @Path annotation, 584
- REST Service Annotations, 583
- root resource class, 579
- web service implementations, 563
- javax.concurrent.ManagedScheduledExecutorService
 - interface, 661
- javax.ejb.Singleton annotation, 436
- javax.enterprise.concurrent.ManagedExecutorService, 655
- javax.enterprise.concurrent.ManagedTask interfaces, 655, 657
- javax.enterprise.concurrent.ManagedThreadFactory, 663
- javax.faces.validator.ValidatorException, 226
- javax.jms.Queue, 445
- javax.json.Json class, 613
- javax.jws.WebMethod annotation, 564
- javax.persistence.cache.retrieveMode hint, 464–465
- javax.persistence.criteria.CriteriaBuilder interface, 466
- javax.persistence.Persistence
 - createEntityManagerFactory method, 410
- javax.servlet.Filter interface, 24
- javax.servlet.http.Cookie object, 44
- javax.servlet.http.HttpSessionListener, 28
- javax.transaction.UserTransaction interface, 658–659
- javax.ws.rs.client.Client, 587
- javax.ws.rs.client.ClientBuilder newClient method, 587
- javax.xml.ws.Endpoint object's, 572
- JAX-RS Client
 - Client setProperty method, 587
 - Employee entity, 589
 - get method, 589
 - Invocation, 590
 - JavaEERecipes application, 587
 - javax.ws.rs.client.Client, 587
 - MediaType fields, 588
 - readEntity method, 589
 - request method, 589
 - RESTful web service, 586
 - Web Resource Targets, 587
 - WebTarget injection, 590
- JAX-RS method Long-Running Operations
 - Asynchronously, 594–596
- JAX-WS web service
 - Acme Bookstore database, 565
 - coding directly/IDE, 569
 - deploy, 572–573
 - endpointInterface attribute, 569
 - javax.jws.WebMethod annotation, 564
 - NetBeans 7.x IDE, 566–569
 - obtainContactList, 565
 - service implementation class, 569
 - @SOAPBinding interface, 569
 - stand-alone Java client application, 574–576
 - web-based client application, 564
 - @WebMethod annotation, 569–570
 - @WebService annotation, 566, 570
 - WSDL document, 571
 - WSDL file, 573–574
- JMS 2.0 aligned activationConfig properties, 446
- JobOperator, 668
- JoinRowSet, 367
 - addRowSet method, 367
 - CachedRowSet objects, 367
 - JOIN query, 364
- JPQL tool, 679
- JSF
 - autocompletion, 297–299
 - custom annotation class (*see* Custom annotation class)
 - custom conversion, 271–273
 - customized data table
 - CartController class, 305
 - h:dataTable component, 307
 - p:dataTable component, 304, 307
 - p:rowEditor component, 307
 - PrimeFaces, 306–307
 - recipe06_16.xhtml, 304
 - RichFaces library, 306
 - RowEditEvent, 305, 307
 - third-party component library, 304
 - updateRowData method, 305, 307
 - HTML5 component
 - cc:implementation tags, 295
 - cc:interface tags, 294
 - composite component, 292
 - Recipe 4, 294
 - recipe06_12.xhtml, 293
 - recipe06_18.xhtml, 314
 - taglib URI, 314–315
 - video component, 292
 - web components, 294
 - listener method, 285–286
 - managed bean scopes
 - CartController, 275, 283
 - cart.xhtml, 279
 - dataTable component, 279
 - review cart item, 282
 - @SessionScoped annotation, 282
 - shopping cart, 274, 276, 278
 - page flow development (*see* Page flow development)
 - phase event, 295–296
 - system-level events, 283–284
 - viewAction component, 286–288
- jsf.ajax.request() function, 269
- jsf.ajax.request() method, 261, 270–271
- JSF application projects, 675–678
- JSF dataTable component, 348
- Json.createReader() method, 613

JSON object

- building
 - beginArray() method, 611
 - beginObject(), 611
 - JsonObjects, 610
 - JsonObjectBuilder, 610
 - JsonObjectBuilder.beginObject() method, 611
 - technicalReviewer objects, 611
 - reading from input source, 612
 - writing, 612
- JsonObjectBuilder class, 610–611
- JsonParser, 614
- JsonWriter class, 612
- JsonWriter writeObject() method, 612
- Jython servlet
 - BookstoreJython.py servlet, 602
 - convertHTML function, 603
 - doGet and doPost, 603
 - org.python.util.PyServlet, 602
 - response.getWriter(), 603

■ L

- LinkedHashMap, 229
- @Local annotation, 415, 443–444
- @Lock annotation, 436
- LowerConverter, 272

■ M

- @ManagedBean annotation, 168
- ManagedExecutorService, 594–595
 - application server administrators, 650
 - asadmin create-managed-executor-service, 647
 - AuthorTask, 656
 - AuthorTaskTwo, 656
 - concurrent/BatchExecutor, 648
 - concurrent/BuilderExecutor, 655
 - GlassFish, 647
 - GlassFish Managed Executor Services panel, 649
 - GlassFish Server Administration Console, 649
 - JNDI name, 650
 - List<Future<object>>, 657
 - New Managed Executor Service panel, 649–650
 - @Resource annotation, 651, 654, 657
 - Runnable, 655
 - submit method, 595, 654
 - UserTransaction interface, 658
 - web.xml, 653
- @ManagedProperty annotation, 187
- ManagedScheduledExecutorService, 661
- ManagedThreadFactory, 663–664
- ManagedThreadFactory resource, 662

- ManagedThreadFactory newThread method, 663
- MapMessage, 526
- matheval field, 36
- MathServlet, 15
- MessageBodyWriter implementations, 592
- message component, 268
- Message consumer, 528
- @MessageDriven annotation, 445–446
- Message-driven EJBs, 409
- MessageListener interface, 445
- messageReceiver method, 605
- META-INF directory, 665
- Model view controller (MVC)
 - AcmeBookstoreMainController class, 637–638
 - acme_bookstore_main.fxml, 635–636
 - <AnchorPane> element, 636–637
 - fx:controller, 640
 - FXML, 638–639
 - initialize() method, 640
 - javafx.scene.control.Label node, 639
 - javafx.scene.layout.AnchorPane class, 639
 - java.lang package, 639
- moveToInsertRow method, 363

■ N

- @NameBinding annotation, 593
- @NamedNativeQuery annotation, 453
- @NamedQuery annotation, 402
- @NamedStoredProcedure solution, 460
- NavigationHandler, 288
- NetBeans IDE, 161
 - Add Server, 671
 - EJB entity classes, 678
 - expand and administer server, 673
 - HTML5, 679–681
 - Java Enterprise Applications, 673
 - Java Web Project, 673–675
 - JPQL, 679
 - JSF application files, 675–678
 - remote server, 671
 - Set Server Location, 672
- newsletterSubscriptionMsgs, 264
- newThread method, 664
- NonUniqueResultException, 452

■ O

- ObjectMessage, 526
- Object-relational mapping (ORM), 369, 458
 - automatic schema generation
 - annotations, 407–408
 - DDL script, 406
 - EJB 3.2, 405
 - NetBeans/Eclipse, 405

- object-relational metadata, 405
 - PersistenceProvider generateSchema method, 405
 - properties, 406
 - SQL scripts, 405
- creation and performing standard database transaction, 369
- data type mapping
 - CONTACT database table, 375–376
 - database column’s data type, 376
 - entity class creation, 377
 - Oracle and Java, 377
- definition, 369
- entity class creation
 - annotations, 374
 - BOOK_AUTHOR database table, 370, 372
 - camel-case lettering, 373
 - definition, 372
 - EJB technology, 370
 - java.io.Serializable interface, 373
 - javax.persistence.Entity annotation, 373
 - @Lob annotation, 373
 - @NotNull annotation, 373
 - String variable, 373
 - @Table annotation, 373
- entity field
 - bean validation mechanism, 403
 - data validation, 404
 - JavaScript field validation, 404
 - @Pattern annotation, 404
 - persistent field/property, 403
 - POJO, 404
- entity objects, 369
- @ManyToMany annotation, 397
- ManyToMany relation
 - cascade, 399
 - entity creation, 400
 - fetch, 399
 - @JoinTable annotation, 400
 - mappedBy, 399
 - owning entity, 398
 - persistent field/property, 399
 - targetEntity, 399
- ManyToOne relation
 - AuthorDetail object, 396
 - field/property, 394
 - persistent field or property, 396
- named query
 - BookAuthor entity class, 401–402
 - JPQL query, 402
 - @NamedQuery annotation, 402
 - persistence.xml file, 401
 - while loop, 402
- @OneToMany annotation, 397
- OneToMany relation
 - AuthorDetail class, 395
 - AuthorDetail objects, 396
 - cascade, 397
 - fetch, 397
 - field/property, 394
 - nonowning entity, 396
 - orphanRemoval, 397
 - owning entity, 396
 - owning object, 394
 - targetedEntity, 397
- One-to-One relation
 - AUTHOR_DETAIL table, 392–393
 - Author entity, 392
 - cascade operation, 393
 - fetch, 393
 - mappedBy attribute, 394
 - @OneToOne annotation, 391
 - optional attribute, 394
 - owned relation, 391
 - unique identifier, 393
- persistence unit creation
 - database connection pool, 377
 - database transaction, 377
 - EntityManagerFactory object, 379
 - Java Transaction API/Resource-Local entity manager, 378
 - local JDBC connection, 377
 - persistence.xml code, 378
 - persistence.xml configuration file, 380
 - properties of element, 379
 - WAR/EAR file, 378
- primary key value creation
 - AuthorWorkPKEmbedded class, 385–386
 - AUTO value, 383
 - BookAuthor entity, 380
 - BOOK_AUTHOR_S sequence, 381
 - bookId and authorId columns, 384, 387–389
 - embeddable techniques, 389
 - equals() method, 390
 - @GeneratedValue annotation, 382–383
 - @Id annotation, 386, 390
 - ID column, 384–385
 - IDENTITY value, 383
 - javax.persistence.EmbeddedId annotation, 383
 - javax.persistence.IdClass annotation, 383
 - local JDBC database, 381
 - nonembedded technique, 390–391
 - persistent fields/property, 390
 - @SequenceGenerator annotation, 382
 - SEQUENCE value, 383
 - TABLE value, 383
 - unique identifier, 389
- obtainCustomList method, 432
- onDataAvailable method, 51
- @OneToOne annotation, 391

onMessage method, 445
 onWritePossible method, 51
 Opening/closing template clients, 171–172
 Optional transaction life-cycle callbacks
 AcmeFacade bean, 441
 container-managed transaction demarcation, 442
 TransactionAttributeType.REQUIRES_NEW
 attribute, 441
 Oracle Glassfish application server
 administrative console
 administrator password, 478
 Common Tasks panel, 478
 domain panel, 478
 General Information panel, 479
 login, 477
 main screen, 477
 Resources menu, 480
 “server” menu option, 479
 administrative user password
 default administrator password, 481
 domain menu, 481
 New Administrator Password form, 480
 database resource
 additional JDBC connection properties, 486
 connection pool, 484
 JDBC Resources, 485
 new JDBC connection pool, 485
 opening and closing connections, 487
 RDBMS, 484
 types, 487
 forms-based authentication
 database table, 488
 JDBC Security Realm Class, 490
 login form, 494
 new security realm, 490
 realm listing, 491
 security configuration, 493
 sun-web.xml configuration file, 492
 installation
 Config Results screen, 475
 directory screen, 473
 Glassfish v4 ZIP archive, 471
 graphical user interface, 476
 Java EE 7 development, 471
 typical Installation, 472
 update tool screen, 474
 ZIP archive, 475
 JavaMail, 494
 WAR archive
 Applications panel, 482
 autodeployment option, 483
 deploy applications/modules, 483
 deployed applications, 482
 EAR files, 484
 Java web application deployment, 484

oracle.jdbc.pool.OracleDataSource, 320
 OrderFacade class, 417
 org.javaeerecipes.chapter07, 326
 org.javaeerecipes.jpa.session.AuthorWork
 Facade class, 458
 org.javaeerecipes.jpa.session.BookFacade class, 451
 ourExecutorServiceName, 650
 OUTER join, 469
 out.println method, 14
 outputFormat component, 218
 outputLabel components, 218
 outputLink component, 217–218
 outputText component, 218

■ P

Page flow development
 definition, 312
 EL variable, 314
 exampleFlow, 309
 faces-config.xml file, 308
 FlowBean, 309
 <flow-definition> section, 308
 flowname, 308
 flow technology, 308
 managed bean, 312
 navigational rules, 313
 web flow, 311
 panelGrid and panelGroup components, 238
 passivationCapable attribute, 442–443
 PasswordLength, 300
 @Path annotation, 584
 p:dataTable component, 307
 performCreate method, 334
 performDelete method, 335
 performFind method, 339
 performUpdate method, 334
 Persistence.createEntityManagerFactory
 method, 379
 PersistenceException, 429
 Persistence.generateSchema method, 405
 PersistenceProvider.generateSchema
 method, 405
 @PersistenceUnit annotation, 416
 persistence.xml configuration file, 679
 persist method, 429
 PhaseListener, 296
 Plain old Java object (POJO), 104, 404
 poll component, 291
 populateNotificationTypes method, 235
 @POST, 584
 @PostActivate annotation, 422, 443
 @PostConstruct annotation, 415, 422
 PostConstructApplicationEvent, 285
 @PreDestroy annotation, 415, 422

PreDestroyApplicationEvent, 285
 PreparedStatement object, 335, 338–341, 353
 @PrePassivate annotation, 422, 443
 PrimeFaces autoComplete component, 299
 PrimeFaces component library, 252
 PrimeFaces dataTables, 306
 PrintWriter object, 7
 PrintWriter println method, 12
 processEvent method, 284
 processEvent(SystemEvent event) method, 283–284
 processItem method, 669
 <processor> element, 668
 processRequest method, 7, 14, 44
 <properties> element, 668
 p:rowEditor component, 307
 publish method, 572
 @PUT, 584

Q

QName object, 575
 queryBooks method, 350

R

readEntity method, 589
 ReaderInterceptor, 592
 readFrom method, 592
 readItem method, 668
 ReadListener, 51–52
 readyState attribute, 609
 RecipeServlet, 84
 @Remote annotation, 415, 443–444
 @Remove method, 421
 Representational state transfer (REST) web services
 acme_bookstore_rest.fxml, 643
 clientLabel, 645
 createTask method, 645
 EventHandler<WorksrStateEvent>, 645
 javafx.application.Application class, 644
 obtainData method, 645
 RESTFulCientController, 643–644
 service client, 641–642
 user interface, 640
 request method, 589
 RequestScoped, 205
 response.getCookies(), 44
 response.getWriter method, 14
 ResultSet.CONCUR_READ_ONLY, 353
 ResultSet.CONCUR_UPDATABLE, 353
 ResultSet getInt method, 331
 ResultSet getString method, 331
 ResultSet objects, 331, 352
 @Retention annotation, 303
 reviewItem.xhtml, 280

RichFaces component library, 252
 RichFaces library, 306
 RowEditEvent, 307
 RowSet objects
 AUTHOR_WORK.AUTHOR_ID column, 367
 CachedRowSet, 367
 key-value relationships, 367
 run method, 660

S

scheduleAtFixedRate method, 662
 ScheduleExecutorService interface, 662
 Scriptlets, 53, 57–58, 94–95
 Scrollable ResultSets
 constants, 353
 org.javaeerecipes.chapter07.dao.ChapterDAO
 class, 351
 PreparedStatement, 352–353
 ResultSet.CONCUR_READ_ONLY, 353
 ResultSet.CONCUR_UPDATABLE, 353
 Secure Sockets Layer (SSL), 543
 Security
 authentication backend
 Enterprise JavaBean, 546
 HTTP request login, 556
 JSF Managed Bean, 551
 User entity, 553–556
 basic web application authentication
 @DeclareRoles and @RolesAllowed
 annotation, 541
 forms-based security, 540
 web-resource-collection element, 542
 web.xml deployment descriptor, 541
 XML configuration, 542
 custom security certificates
 CSR, 562
 Glassfish application server, 561
 Glassfish
 authentication and access purposes, 540
 edit realm form, 538
 file users form, 539
 file users list, 539
 New File Realm User form, 539
 realms form, 538
 JSF application
 Boolean, 557
 conditional logic, 557
 managed bean controller, 558
 PrimeFaces panel component, 558
 LDAP server
 form-login-page and form-error-page value, 561
 New LDAP security realm, 560
 security realm, Glassfish, 559
 programmatic login form

Security (*cont.*)

- Acme Bookstore template, 546
- creation, 544
- JSF and Facelets, 544
- selectBooleanCheckbox component, 234–235, 237
- selectManyCheckbox component, 234–235
- selectManyMenu component, 226
- selectOneListbox component, 226
- selectOneMenu component, 226
- selectOneRadio component, 236–237
- sendRedirect method, 40
- @ServerEndpoint annotation, 606
- Server-side POJO class, 605
- service method, 6
- ServiceUnavailableException, 595
- Servlet
 - attribute changes, 25–27
 - beingDestroyed value, 45
 - browser, 41–42, 44
 - compiling, 7–8
 - container events, 18–20
 - deploying, 7
 - destroy method, 45
 - develop
 - configuration and mapping, 4
 - doGet and doPost methods, 7
 - GenericServlet, 7
 - HttpServlet class, 6
 - HttpServletRequest object, 6
 - IDE, 5
 - init method, 6
 - javac command-line utility, 5
 - javax.servlet, 6
 - processRequest method, 6–7
 - request-response programming model, 5
 - service method, 6
 - ServletConfig interfaces, 7
 - simple servlet creation, 3
 - SimpleServlet.java file, 5
 - dispatching requests, 35–40
 - dynamic content, 12–14
 - file download, 31–33
 - filtering web requests, 23–24
 - handling requests and responses, 15–17
 - initialization parameters, 21–22
 - Java enterprise environment, 2
 - Java web applications, 1
 - JSF, 1
 - JSP, 1
 - non-blocking I/O
 - AcmeReaderServlet, 48
 - AcmeReadListenerImpl class, 47
 - AsyncContext.complete method, 51
 - onDataAvailable method, 51
 - ReadListener, 46, 51–52

- ServletInputStream, 46–47, 51
- ServletOutputStream.canWrite method, 51
- WebSockets protocol, 50
- WriteListener, 51–52
- package, 7, 9
- redirect, 40–41
- session attributes, 29–31
- session listener, 27–29
- Sun Microsystems, 1
 - @WebServlet annotation, 9, 11–12
- ServletConfig interfaces, 7
- ServletContextEvent, 20
- ServletContextListener, 20
- ServletOutputStream.canWrite method, 51
- ServletRequest and ServletResponse objects, 24
- Session bean, 415
- SessionContext.getUserTransaction() method, 659
- sessionCreated method, 27, 29
- sessionDestroyed method, 27, 29
- @SessionScoped annotation, 282
- SessionScoped managed beans, 274
- SessionServlet, 30
- setAttribute() method, 29
- SET command, 8
- setMaxAge and setHttpOnly methods, 44
- setName and setValue methods, 44
- setParameter method, 450–451
- setProperty method, 587
- setTimeoutHandler, 595
- SimpleServlet.class file, 9
- SimpleServlet.java file, 5, 8
- Singleton bean
 - Acme Bookstore Facelets template, 436
 - BookstoreSessionController bean, 435
 - BookstoreSessionCounter bean, 434
 - @DependsOn annotation, 436
 - Java synchronization techniques, 437
 - javax.ejb.Singleton annotation, 436
 - @Lock annotation, 436
 - @PostConstruct callback annotation, 436
 - @Startup annotation, 436
 - thread-safe, 436
 - tracking number of visitors, 434
- @SOAPBinding interface, 569
- SQL injection
 - DAO (*see* Data access object (DAO))
 - executeQuery method, 340
 - executeUpdate method, 341
 - java.sql.PreparedStatement object, 335
 - performFind method, 339
 - PreparedStatement, 335, 338–341
 - String, 339
 - try-catch block, 339
 - try-with-resources clause, 340
- ResultSetMapping, 453–454

@ResultSetMapping annotation, 454, 457
 ResultSetMapping parameters, 456
 @Stateful annotation, 421, 442
 Stateful session beans, 409
 CartController code, 418
 client and bean one-to-one mapping, 421
 conversational state, 417
 life cycle, 421
 OrderFacade class, 417
 passivation, 422, 442
 @PostConstruct, @PrePassivate, @PostActivate, and
 @PreDestroy annotations, 422
 @Remove method, 421
 Serializable variables, 421
 @Stateful annotation, 421
 @StatefulTimeout annotation, 422
 Stateless session beans, 409
 abstract class, 413
 AbstractFacade class, 413, 416
 BOOK entity, 411
 BookFacade class, 412
 business interfaces, 415
 callback methods, 415
 class creation, 411
 EntityManager object, 415–416
 findByTitle method, 413, 416
 JavaEERecipes/JTA persistence unit, 415
 JDBC DataSource declaration, 415
 life cycle, 416
 no-argument constructor, 415
 @Stateless annotation, 415
 thread-safe, 415
 Statement executeQuery method, 334
 StreamMessage, 526
 Structured Query Language (SQL), 369
 Sun Microsystems, 1
 @Suspended annotation, 595
 SyncProviderException, 363
 Syntactic metadata, 302
 SystemEventListener class, 285
 SystemEventListener interface, 283–284
 System.out() method, 331

T

Tag library descriptor (TLD), 63
 @Target annotation, 302
 testAttr, 28
 TextMessage, 526
 @Timeout annotation, 440
 TimeoutHandler, 595
 @Timeout method, 441
 toUpperCase method, 272, 333
 @TransactionAttribute, 442
 TransactionRequiredException, 429–430

TREAT keyword, 467–468
 TREAT operator, 468
 try-catch block, 324, 339
 try-with-resources, 329, 340
 ttributeRemoved method, 27

U

updateRowData method, 307
 uploadFile method, 253
 UserTransaction begin method, 659
 UserTransaction commit method, 659

V

validatePassword method, 222–223
 validator attribute, 222
 valueChangeListener attribute, 235

W, X, Y, Z

Web archive (WAR), 378, 484
 @WebFilter annotation, 24
 WEB-INF directory, 8–9
 WEB-INF/lib directory, 252
 @WebInitParam annotation, 21–22
 @WebMethod annotation, 569–570, 577, 579
 @WebService annotation, 415, 577
 @WebServiceRef annotation, 576
 @WebServlet annotation, 9, 11–12
 WebSockets
 creation, 605
 full-duplex communication mechanism, 605
 sending messages
 alert dialog, 609
 bookChatRelay function, 606, 608
 close() method, 609
 if-statement, 608
 JavaScript code, 608
 JavaScript WebSocket object events, 609
 onclick attribute, 606
 onOpen handler, 609
 readyState values, 609
 send() method, 609
 source listing, 606
 WebTarget injection, 590
 web.xml deployment descriptor file, 22
 WHERE clause, 450
 WidgetOutputItem object, 666–667
 WidgetReportItem object, 665–667
 writeItems method, 669
 WriteListener implementation class, 51–52
 WriterInterceptor, 592
 writeTo method, 592
 wsimport tool, 573