

Symbols

#springFormInput macro 563
#springFormTextarea macro 563
<@spring.formInput> macro 568
<@spring.formTextarea> macro 568
<[CDATA...]> construct 113
@AspectJ annotation 141, 144
 @Around annotation 144
@Cacheable annotation 217
@CacheFlush annotation 217
@Configurable annotation 87
@Pointcut annotation 142
@Transactional annotation 244

A

AbstractCommandController 508–510
AbstractController 508–509
AbstractDependencyInjectionSpringContext
 Tests 694, 696, 705
AbstractExcelView 570–571
AbstractFormController 508–509
abstracting base bean type 74, 76
 common properties 76, 78
 overriding inherited properties 76
 ProxyFactoryBean 138–139
AbstractJDomPayloadEndpoint 355
 invokeInternal() method 357
AbstractJmsMessageDrivenBean 431
AbstractMarshallingPayloadEndpoint 358
 invokeInternal() 360
AbstractMessageDrivenBean 431
AbstractModelAndViewTests class 690, 705
 assertion methods 690

AbstractPdfView 574–575
AbstractStatefulSessionBean 431
AbstractStatelessSessionBean 431–432
AbstractTransactionalDataSourceSpringContext-
 Tests 700–701, 705
AbstractTransactionalSpringContextTests 697,
 699
AbstractTransactionSpringContextTests 705
AbstractWizardFormController 508, 521, 524, 584
 getTargetPage() 524–525
 processCancel() 526
 processFinish() 525–526
access decisions managers 251, 271–275
 voter abstinance 275
 voting 272–274
AccessDecisionManager 271–272
 allowIfAllAbstain property 275
 decide() 272
 supports() 272
AccessDecisionVoter interface 272–273
 See also RoleVoter
AccessDeniedHandlerImpl 292
Acegi Security System 248–249
 See also Spring Security
<action> 607
<action-state> 597, 607
ActionSupport 627, 629
ActiveMQ message broker
 setting up 392–393
 creating connection factories 392
 declaring destinations 392–393
ActiveMQConnectionFactory 392–393
 brokerURL property 392
Adobe's Portable Document Format (PDF)
 569–575

- advice 119–120, 127, 132
 - AfterReturningAdvice 129–130
 - MethodBeforeAdvice interface 129
 - MethodInterceptor interface 131–132
 - Spring AOP 127
 - ThrowsAdvice interface 130–131
- advisors 132–136
 - combining pointcuts with 134
- after-invocation managers 251–252
- AfterReturningAdvice 129–130
- Ajax 663–664
 - accessing Spring-managed beans DWR 659, 661–663
 - Direct Web Remoting (DWR) 648, 650–651, 654, 656–658
- allowIfAllAbstain property 275
- annotated domain objects 187–188
- AnnotationAwareAspectJAutoProxyCreator 143
- annotation-driven caching 217–218
- annotation-driven transactions 243, 245
- AnnotationJmxAttributeSource 474
- annotations
 - declaring interceptors using Pitchfork 438–439
 - injecting resources by Pitchfork 437–438
 - supported by Pitchfork 435
- AnnotationSessionFactoryBean 187–188
- Ant 675–676
- aop namespace 243
- AOP. *See* aspect-oriented programming (AOP)
- <aop:after-throwing> 147
- <aop:aspect> 28
- <aop:aspectj-autoproxy> 143
- <aop:before> 29, 147
- <aop:config> 28, 146
- <aop:pointcut> 29, 148
- <aop:spring-configured> 87
- Apache Ant 675–676
- application context
 - exposing in JSF 648
 - splitting up 493–494
- application context module 8
- application events 101–103
- application lifecycle events 8
- application objects, associations between 32
- ApplicationContext interface 35, 40
 - lifecycle 37, 40
 - publishEvent() 102
- ApplicationContextAware interface 104
 - setApplicationContext() 105
- ApplicationContextAwareProcessor 95
- ApplicationContexts 33
 - compared to BeanFactories 36
 - loading 36–37
 - retrieving beans from 37
- application-managed Java Persistence API (JPA) 198–199
- approveCreditCard() 606
- AspectJ 122
- AspectJ 5 141
- AspectJ aspects 88
- AspectJ pointcuts 135–136
- aspectj/ directory 668
- AspectJExpressionPointcut 135
- AspectJExpressionPointcutAdvisor 135
- aspect-oriented programming (AOP) 8, 24, 30, 118, 125
 - autoproxying 139–144
 - compared to dependency injection (DI) 117
 - configuration elements 145
 - creating classic aspects 125–139
 - declaring pure-POJO aspects 145, 149
 - description 24–25
 - example 26–27, 30
 - module 8
 - overview 4, 6, 118–119
 - Spring support for 122, 125
 - terminology 119–120
- aspects 120, 125, 139
 - @AspectJ 141, 144
 - advice, creating 127–132
 - advisors 132–136
 - AspectJ 149, 152
 - autoproxies, creating 140–141
 - compared to inheritance and delegation 119
 - implementing 27, 30
 - pointcuts, defining 132, 136
 - AspectJ 135–136
 - combining with advisors 134
 - regular expression 133–134
 - ProxyFactoryBean 136, 139
 - abstracting 138–139
 - pure-POJO 145, 149
- asynchronous messaging 385–386, 422
 - Java Message Service (JMS) 386, 393, 407
 - architecture 387, 389
 - benefits 390–391
 - converting messages 402, 405
 - gateway support classes 405, 407
 - runaway code 393, 395
 - setting up ActiveMQ message broker 392–393

- asynchronous messaging, Java Message Service (JMS) (*continued*)
 - templates 395, 402
 - message-based RPC 416, 422
 - exporting services 418–419
 - Lingo 417–418
 - proxying Java Message Service (JMS) 420, 422
 - message-driven POJOs (MDPs) 407, 416
 - creating message listeners 408, 412
 - writing 412, 416
 - atomic, consistent, isolated, durable (ACID) 223–224
 - attributes property,
 - MethodDefinitionAttributes 303
 - attributes, MBean
 - transactions 233, 237
 - isolation levels 235–236
 - propagation behavior 233, 235
 - read-only 236–237
 - rollback rules 237
 - transaction timeout 237
 - using interfaces to define 472–473
 - AttributesJmxAttributeSource 474
 - AuthByAuthenticationProvider 254
 - See also* ProviderManager
 - authenticating users 252, 271
 - against databases 256, 264
 - caching user information 263–264
 - encrypted passwords 260, 262
 - InMemoryDaoImpl 257, 259
 - JdbcDaoImpl 259–260
 - against LDAP repositories 264, 271
 - BindAuthenticator 265, 267
 - comparing passwords 267, 269
 - DefaultLdapAuthoritiesPopulator 269, 271
 - ProviderManager 253, 256
 - authentication 287–291
 - authentication manager 250
 - authentication providers 254
 - AuthenticationDao 257
 - See also* DaoAuthenticationProvider
 - authenticationFailureUrl property 291
 - AuthenticationManager 252
 - authenticate() 253
 - interface 252
 - authenticationManager property 291
 - See also* ProviderManager
 - authentication-processing filters 277
 - AuthenticationProcessingFilter 290–291
 - AuthenticationProcessingFilterEntryPoint 289
 - AuthenticationProvider interface 256
 - authoritiesByUsernameQuery property 260
 - authorization exceptions 292–293
 - <authz:authentication> 300
 - <authz:authorize> 298
 - <authz:ifAllGranted> 298
 - <authz:ifAnyGranted> 298
 - <authz:ifNotGranted> 299
 - <authz:operation> 300
 - autoboxing 182
 - autodetect autowiring 62
 - autowire attribute 62
 - autowire property 58
 - autowiring 58–64
 - mixing with explicit wiring 63
 - shortcomings 63–64
 - types 59, 62
 - awareness 103–106
 - BeanNameAware interface 104–105
 - AxisBeanMappingServicePostProcessor 337
- ## B
-
- base bean types, abstracting 74, 76
 - BaseCommandController 508–509
 - BasicDataSource 167–168
 - BasicProcessingFilter 288
 - BasicProcessingFilterEntryPoint bean 286
 - <bean> 41
 - abstract attribute 74–75
 - factory-method attribute 67
 - parent attribute 74
 - BeanCounter 96
 - BeanDeserializer class 337
 - BeanFactories 33
 - retrieving beans from 35
 - BeanFactory class 13
 - BeanFactory container 7
 - BeanFactory interface 34–35
 - BeanFactoryAware interface 104–105
 - BeanFactoryPostProcessor interface 95–96
 - BeanNameAutoProxyCreator 301
 - BeanNameAware interface 104–105
 - BeanNameUrlHandlerMapping 499, 501, 503
 - BeanNameViewResolver 538, 540–541
 - BeanPostProcessor interface 93, 95
 - <beans> 12
 - beans
 - controller 498–499
 - exposing as HTTP services 324, 326

- beans (*continued*)
 - exposing functionality with Burlap
 - exporting services 321–322
 - exposing functionality with Burlap and Hessian 318, 322
 - exposing functionality with Hessian
 - controllers 320–321
 - HessianServiceExporter 319–320
 - lifecycle 34
 - steps 37
 - proxied inner 216–217
 - proxying for caching 215, 217
 - flushing 215–216
 - retrieving from ApplicationContexts 37
 - <beans> element 12
 - BeanSerializer class 337
 - BeanShell, scripting beans in 110–111
 - BindAuthenticator 265–267
 - boilerplate code, JDBC 172
 - brokerURL property 392
 - buildExcelDocument() 571
 - building
 - classpaths 669, 671
 - controllers 496–497
 - homepages 495–502
 - orders 601, 605
 - wizard controllers 521, 524
 - buildPdfDocument() 575
 - Burlap 316, 322
 - accessing services 317–318
 - exposing bean functionality 318, 322
 - exporting services 321–322
 - BurlapProxyFactoryBean 317
 - BurlapServiceExporter 321
- C**
-
- <cache> 213
 - cache property, JndiObjectFactoryBean 448
 - caching data 208, 218
 - annotation-driven 217–218
 - caching solutions 210, 215
 - EHCache 213, 215
 - proxying beans for 215, 217
 - flushing 215–216
 - proxied inner beans 216–217
 - callback classes 162
 - CasAuthenticationProvider 254
 - See also* ProviderManager
 - Cascading Style Sheets (CSS) 573
 - case normalization 270
 - CastorMarshaller 366
 - catch blocks 159–161
 - Caucho Technology 316
 - cellFuncs 658
 - ChannelDecisionManagerImpl 297
 - ChannelProcessingFilter 295
 - filterInvocationDefinitionSource 296
 - channels 294–297
 - child beans
 - declaring 73, 78
 - abstracting base bean type 74, 76
 - abstracting common properties 76, 78
 - class attribute 41
 - class inheritance 75
 - ClassEditor 89
 - classload time 121
 - classpaths, building 669, 671
 - ClassPathXmlApplicationContext 36–37
 - collaboration 19–20
 - collection configuration elements 52–53, 58
 - arrays 53–54
 - lists 53–54
 - maps 55–56
 - properties 56, 58
 - sets 54–55
 - commands, processing 509, 511
 - Commons Attributes compiler 505
 - Commons Logging 671
 - Commons Validator 517, 520
 - validation rules 519
 - CommonsHttpMessageSender 376
 - CommonsPathMapHandlerMapping 503, 505
 - compile time 121
 - ComponentControllerSupport 555
 - conditionally rendering content 298–299
 - configLocation property 205
 - configuration attributes, EHCache 214
 - configuration elements
 - aspect-oriented programming (AOP) 145
 - Spring Modules 212
 - configuring
 - context loaders 494–495
 - controller beans 498–499
 - data sources 96, 165, 169
 - Direct Web Remoting (DWR) 650–651
 - DispatcherServlet 492–495
 - engines
 - FreeMarker 565–566
 - Velocity 558–559
 - flow executors 586–587

- configuring (*continued*)
 - JDBC driver-based 168–169
 - JNDI 165, 167
 - Spring 2.0 166–167
 - mail senders 451, 453
 - Java Naming and Directory Interface (JNDI) 452
 - wiring into service beans 453
 - pooled 167–168
 - Remote Method Invocation (RMI) services 313, 316
 - Tiles 552
- connection factories, Java Message Service (JMS) 392
- ConnectorServerFactoryBean 478
- constructor autowiring 61–62
- constructor injection versus setter injection 44
- <constructor-arg> 13, 42
 - autowiring 63
 - ref attribute 44
 - value attribute 44
- constructors, injecting through 42, 45
 - object references 43, 45
- container 6
- container-managed Java Persistence API (JPA) 200, 202
- containers 6, 33
 - Bean factories 33
- containing beans 33–40
 - ApplicationContext interface 35, 40
 - lifecycle 37, 40
 - BeanFactory interface 34–35
- context loaders, configuring 494–495
- context reload, forcing 695–696
- contextConfigLocation parameter 495
- ContextLoaderListener 494, 661
- ContextLoaderServlet 495
- contextual sessions, Hibernate 3 192, 194
- contract-first web services 344–347, 373, 382–383
 - creating sample XML messages 348, 353
 - messages with service endpoints 353, 360
 - JDOM-based message endpoints 355, 357
 - marshaling message payloads 358, 360
 - web service gateway support 381–382
 - web service templates 374, 380
 - marshalers on client side 379–380
 - sending messages 377, 379
 - wiring 361, 373
 - configuration 361, 363
 - deploying 373
 - endpoint exceptions 367, 369
 - mapping messages to endpoints 363–364
 - message marshalers 364, 367
 - service endpoints 364
 - WSDL files 369, 373
- contract-last web services 345
- Controller 508
- controller beans, configuring 498–499
- Controller interface 507
- ControllerClassNameHandlerMapping 503–505, 586
- controllers
 - building 496–497
 - classes 508
 - creating Tile 554, 556
 - form 512
 - Hessian 320–321
 - hierarchy 507
 - mapping requests to 502–506
 - Spring MVC 491–492, 685, 692
 - throwaway 528, 531
 - wizard 521, 524
- conversation-style navigation 583
- convertAndSend() 404
- converted messages 402–405
- convertToUpperCase 270
- core container 7–8
- coupling 17, 19
- creating beans 40, 45
 - controlling 64, 71
 - creating from factory methods 66, 68
 - destroying 68, 71
 - initializing 68, 71
 - scoping 66
 - declaring 40–41
 - injecting through constructors 42, 45
 - object references 43, 45
- cron expressions 464
- cron jobs, scheduling 462–463
- CronTriggerBean 104, 462–463
- cross-cutting concerns 24, 117–118
- custom editors 89–92
- CustomDateEditor 89
- CustomEditorConfigurer 91
- customer information 594, 601
 - adding new customers 598, 601
 - creating data flow 615, 617
 - looking up data 597–598
 - phone numbers 595–596

D

- DAO. *See* data access objects (DAOs)
- DaoAuthenticationProvider 256–260
 - authenticationDao 257
 - diagram of 256
 - MD5 encoding 261
 - userDetailsService property 257
 - wiring 257
 - See also* ProviderManager
- data access 156, 219
 - caching 208, 218
 - annotation-driven 217–218
 - caching solutions 210, 215
 - proxying beans for 215, 217
 - data sources 165, 169
 - JDBC driver-based 168–169
 - JNDI 165, 167
 - pooled 167–168
 - Hibernate 183, 194
 - data access objects (DAOs) 190–191
 - Hibernate 3 contextual sessions 192, 194
 - templates 186, 190
 - versions 185–186
- iBatis 203, 208
 - data access objects (DAOs) 207–208
 - templates 204, 207
- Java Persistence API (JPA) 194, 203
 - data access objects (DAOs) 202–203
 - entity manager factories 197, 202
 - templates 194, 197
- JDBC 170, 183
 - data access object (DAO) support classes 180, 183
 - runaway code 170, 173
 - templates 173, 180
- overview 156–157
- tiers 157, 165
 - data access object (DAO) support classes 163, 165
 - exceptions 158, 161
 - templating 161, 163
- data access exceptions 160
- data access objects (DAOs) 157
 - Hibernate 190–191
 - iBatis 207–208
 - Java Persistence API (JPA) 202–203
 - JDBC 180, 183
 - named parameters 181–182
 - simplified in Java 5 182–183
 - module 8
 - service objects access 157
 - support classes 163, 165
- data access templates 163
 - HibernateTemplate 162
 - JdbcTemplate 162
 - JpaTemplate 162
- data contracts 349, 353
- data sources 165, 169
 - configuring 96
 - JDBC driver-based 168–169
 - JNDI 165, 167
 - Spring 2.0 166–167
 - pooled 167–168
- Database Connection Pools (DBCP) 167
- databases
 - authenticating users against 256, 264
 - caching user information 263–264
 - encrypted passwords 260, 262
 - InMemoryDaoImpl 257, 259
 - JdbcDaoImpl 259–260
 - integration-testing against 699, 701–702
- datacentric messaging 390
- DataSource 443
 - JdbcTemplate class 174
 - retrieving from Java Naming and Directory Interface (JNDI) 443, 445
- dataSource property
 - AnnotationSessionFactoryBean 188
 - LocalSessionFactoryBean 187
- DataSourceTransactionManager 226
- dates, formatting in Velocity 560–561
- dateToolAttribute property
 - VelocityViewResolver bean 560
- DBCP. *See* Database Connection Pools (DBCP)
- decide() 272
- decision states 612, 614
- <decision-state> 612–613, 616
- declarative transaction management 224
- decoupling 101–103
- default destinations, setting 399–400
- DefaultAdvisorAutoProxyCreator 95, 140
- DefaultBeanValidator 518
- <defaultCache> 213
- default-destroy-method 69
- DefaultInitialDirContextFactory 266, 269
- default-init-method attribute 69
- DefaultLdapAuthoritiesPopulator 269, 271
 - constructor arguments 269
 - convertToUpperCase property 270
 - groupRoleAttributes property 269
 - groupSearchFilter property 271

- DefaultMessageListenerContainer 411
 - transactionManager property 412
 - defaultObject property,
 - JndiObjectFactoryBean 449
 - DefaultPointcutAdvisor class 134
 - defaultTargetUrl property 291
 - DefaultValidatorFactory 518
 - DelegatingRequestProcessor 629, 631
 - DelegatingTilesRequestProcessor 630
 - DelegatingVariableResolver 621, 646, 648, 664
 - delegation, compared to aspects 119
 - dependency injected tests
 - setting up 696
 - tearing down 696
 - dependency injection (DI) 14–23
 - compared to aspect-oriented programming (AOP) 117
 - enterprise applications 21, 23
 - example 15, 21
 - coordinating collaboration 19–20
 - coupling 17, 19
 - unit testing 16–17
 - wiring 20–21
 - method injection 79, 85
 - getter injection 83, 85
 - method replacement 80, 83
 - overview 4, 6, 14
 - destinations
 - default Java Message Service (JMS) 399–400
 - defined 388
 - destinations, ActiveMQ declaring 392–393
 - destroy() 70
 - destroy-method attribute 68
 - DHTML. *See* Dynamic HTML (DHTML)
 - DI. *See* dependency injection (DI)
 - Diaphragma 641
 - Direct Web Remoting (DWR) 648, 650, 658, 664
 - calling remote methods from JavaScript 656–657
 - configuration 650–651
 - defining remote objects 651, 654
 - displaying results 657–658
 - exporting remote objects to JavaScript 654, 656
 - Spring configuration namespace 662–663
 - Spring creators 659, 661
 - dirty reads 235
 - DispatcherServlet 329, 491–492, 495, 501
 - configuring context loaders 494–495
 - splitting up application context 493–494
 - DisposableBean interface 69–71
 - docs/ directory 669
 - doInTransaction() 232
 - domain objects 86
 - driverClassName property 167
 - DriverManagerDataSource class 169
 - DWR. *See* Direct Web Remoting (DWR)
 - dwr.xml file 654
 - DwrServlet 650, 656
 - DwrSpringServlet 662
 - DWRUtil object 658
 - Dynamic HTML (DHTML) 649
 - dynamic MBeans 467
 - DynamicWsd111Definition 369
- ## E
-
- EasyMock 688–689
 - EHCACHE 213–217
 - <ehcache:annotations> 217
 - <ehcache:caching> 215
 - <ehcache:flushing> 215
 - <ehcache:proxy> 215
 - EhCacheBasedUserCache 263
 - EhCacheFactoryBean 264
 - <ehcache:when> 218
 - EJB. *See* Enterprise JavaBeans (EJBs)
 - EJB 2.x
 - compared to EJB 3 425
 - complexities 434
 - message-driven beans (MDBs) 408
 - session beans, proxying 426, 430
 - Spring-enabled Enterprise JavaBeans 431, 434
 - EJB 3 434, 439
 - compared to EJB 2.x 425
 - message-driven beans (MDBs) 408
 - Pitchfork 435–437
 - declaring interceptors using annotations 438–439
 - injecting resources by annotation 437–438
 - session beans, declaring 429–430
 - specification 194
 - EJB specification 194
 - email
 - constructing 453, 456
 - sending 450, 456–458
 - configuring mail senders 451, 453
 - embark() 17–18
 - embarkOnQuest() 29
 - encrypted passwords 260, 262
 - end states 593–594
 - endpoint exceptions 367, 369
 - endpoints, mapping messages to 363–364

- <end-state> 593–594
 - engines, configuring
 - FreeMarker 565–566
 - Velocity 558–559
 - enterprise applications 21, 23
 - Enterprise JavaBeans (EJB) 424, 440
 - complexities 434
 - specification 4–5
 - enterprise services 442, 485
 - Java Management Extensions (JMX) 466, 484
 - exporting Spring beans as MBeans 467, 477
 - handling notifications 482, 484
 - remoting MBeans 477, 482
 - Java Naming and Directory Interface (JNDI)
 - conventional 443, 446
 - in Spring 2 449–450
 - injecting objects 446, 449
 - wiring objects from 442, 450
 - scheduling tasks 456, 466
 - invoking methods on schedule 464, 466
 - Java Timer class 457, 460
 - Quartz scheduler 460, 464
 - sending email 450, 456
 - configuring mail senders 451, 453
 - constructing email 453, 456
 - entity beans 194
 - entity manager factories
 - Java Persistence API (JPA) 197, 202
 - application-managed 198–199
 - container-managed 200, 202
 - EntityManager interface 194–197
 - entityManagerFactory property 195
 - <entry> 56
 - errors, displaying in JavaServer Pages (JSPs)
 - 547, 549
 - events 590
 - listening for 102–103
 - publishing 101–102
 - Excel spreadsheets 570, 573
 - exception translation filters 277
 - exceptionMappings property 368
 - exceptions
 - authorization 292–293
 - handling 531–532
 - security 291, 293
 - ExceptionHandlerFilter 292
 - execute() 231
 - explicit wiring, mixing with autowiring 63
 - exporting
 - Burlap services 321–322
 - Lingo services 418–419
 - methods by name 471–472
 - remote MBeans 478
 - Remote Method Invocation (RMI) services
 - 312, 316
 - Spring beans as MBeans 467–477
 - exposeRequestAttributes property
 - FreeMarkerViewResolver 566
 - VelocityViewResolver bean 561
 - exposeSessionAttributes property
 - FreeMarkerViewResolver 566
 - VelocityViewResolver bean 561
 - exposeSpringMacroHelpers property
 - VelocityViewResolver bean 563
 - exposing bean functionality
 - Burlap 318, 322
 - exporting services 321–322
 - Hessian 318, 322
 - controllers 320–321
 - HessianServiceExporter 319–320
 - HTTP services 324, 326
 - externalized messages, rendering in JavaServer
 - Pages (JSPs) 544, 547
 - externalizing configuration properties 96, 99
 - extract methods 614
-
- ## F
- factory methods, creating beans from 66, 68
 - factory-method attribute element 67
 - fallback objects, Java Naming and Directory
 - Interface (JNDI) 448–449
 - FileEditor 89
 - FileSystemXmlApplicationContext 36
 - <filter> 278
 - filter security interceptors 278
 - FilterChainProxy 281, 283
 - filterInvocationDefinitionSource property
 - 282, 296
 - <filter-mapping> 278
 - filterProcessesUrl property 290
 - filters, proxying 279–281
 - FilterSecurityInterceptor 293
 - objectDefinitionSource property 294
 - FilterToBeanProxy 279, 629
 - web.xml files 283
 - final methods 124
 - find() 197
 - flow definitions, registering 588
 - flow executors, configuring 586–587
 - flow variables 591, 593
 - scopes 593

- <flow:executor> 587
- <flow:location> 588
- <flow:registry> 588
- FlowAction 599–601
- FlowController 585–586, 596
- FlowExecutorFactoryBean 587
- FlowNavigationHandler 621
- FlowPhaseListener 621
- flows, creating 591, 611
 - building orders 601, 605
 - completing orders 605, 608
 - flow variables 591, 593
 - gathering customer information 594, 601
 - start states 593–594
- flushing caches 215–216
- forceHttps property 289
- foreign languages 540, 547, 549
- form controllers 510
- form data binding in JavaServer Pages (JSPs) 542, 544
- form fields, binding
 - in FreeMarker 567, 569
 - in Velocity 561, 564
- <form:errors> 547–548
- <form:form> 544
- <form:input> 544
- <form:textarea> 544
- formatting, in Velocity 560–561
- form-based authentication 289, 291
- form-binding JavaServer Page (JSP) tags 543
- formObjectClass property, FlowAction 600
- formObjectName property, FlowAction 600
- formObjectScope property, FlowAction 600
- forms
 - pages 524–525
 - processing with wizards 520–521, 524–526, 528
 - submissions 512, 515, 517, 520
 - validating input 515, 517, 520
- Fowler, Martin 14
- free-flow navigation 582
- FreeMarker 564, 569
 - binding form fields 567, 569
 - configuring engines 565–566
 - defining views 564–565
 - exposing attributes 566–567
 - macros 566
 - resolving views 566
- FreeMarkerConfigurer 565–566
- FreeMarkerViewResolver 566–567
- front controllers 491

G

- gateway support classes for Java Message Service (JMS) 405, 407
 - web services 381–382
- getAsText() 89
- getBean() 13, 35, 37
- getConnection() 164
- getContents() 81
- getHibernateTemplate() 190
- getInstance() 67
- getJdbcTemplate() 181
- getNamedParameterJdbcTemplate() 182
- getTargetPage() 524–525
- getter injection 83, 85
- gienah-testing 702, 704
 - SpringRunner 704
- global object 637, 639
- global-session scoping option 65
- greeting property 12–13
- GreetingService interface 11
- GreetingServiceImpl class 11–12
- Groovy, scripting beans in 110
- groupRoleAttributes property 269
- groupSearchFilter property 271

H

- handler mappings 505–506
- HandlerMapping
 - implementations 503
 - order property 506
- HeaderTileController 554, 556
- hello.xml file 12
- Hessian 316, 322
 - accessing services 317–318
 - controllers 320–321
 - exposing bean functionality 318, 322
- HessianProxyFactoryBean 317
- HessianServiceExporter 319–320
- Hibernate 86, 159, 183, 194
 - catch blocks 159
 - contextual sessions
 - advantages/disadvantages 193
 - data access objects (DAOs) 190–191
 - exception heirarchy 159
 - Hibernate 3 contextual sessions 192, 194
 - package structure 185
 - Session interface 186
 - SessionFactory interface 186
 - templates 186, 190

Hibernate, templates (*continued*)
 annotated domain objects 187–188
 classic mapping files 186–187
 transaction managers 227
 versions 185–186
 Hibernate 3 192–194
 HibernateDaoSupport class 190
 HibernateException class 186
 hibernateProperties property 187–188
 HibernateTemplate class 162, 186, 192
 load() 190
 saveOrUpdate() 189
 HibernateTemplate-based DAO 189
 HibernateTransactionManager 227
 HolyGrailQuest class 16–18
 HomeController 496, 501
 name attribute 498
 homepages 495, 502
 building controllers 496–497
 configuring controller beans 498–499
 creating JSP 500–501
 declaring view resolvers 499–500
 ModelAndView 497–498
 HTML content 493
 HttpInvoker 322, 326
 accessing services 323–324
 exposing bean functionality 324, 326
 HttpInvokerProxyFactoryBean 323
 HttpInvokerServiceExporter 324
 HttpSessionContextIntegrationFilter 286
 HttpURLConnectionMessageSender 376

I

iBatis 86, 203, 208
 data access objects (DAOs) 207–208
 templates 204, 207
 SQL maps 205–206
 SqlMapClientTemplate 204–205
 id attribute 41
 ifAllGranted 298
 ifAnyGranted 298
 ifNotGranted 299
 implementing aspects 27, 30
 indirection 387
 inheritance 75
 compared to aspects 119
 initialization on demand holder 67
 initializing beans 68, 71
 default-init-method attribute 69
 init-method attribute 68

<inject> 642
 InjectObject annotation 642
 InMemoryDaoImpl 257–259
 See also AuthenticationDao
 inner beans 51–52
 proxied 216–217
 integration filters 277
 integration-testing 684–704
 AbstractDependencyInjectionSpringContext-
 Tests 705
 AbstractTransactionalDataSourceSpring-
 ContextTests 705
 AbstractTransactionSpringContextTests 705
 against database 699, 701–702
 in JUnit 4 with gienah-testing 702, 704
 transactional objects 696, 698–699
 wired objects 694–696
 interceptors, in Pitchfork 438–439
 InterfaceBasedMBeanInfoAssembler 472–473
 interfaces
 defining MBean operations/attributes 472–473
 mocking with EasyMock 688–689
 template methods 161
 InternalResourceViewResolver 499–502, 536–537,
 540–541, 553
 viewClass property 537
 internationalization 8, 99, 547–549
 introductions 121
 invokeInternal()
 AbstractJDomPayloadEndpoint 357
 AbstractMarshallingPayloadEndpoint 360
 IoC. *See* inversion of control (IoC)
 isolation levels 235–236

J

JaasAuthenticationProvider 254
 See also ProviderManager
 Jakarta Commons Database Connection Pools
 (DBCP) 167
 Jakarta Commons Logging 671
 Jakarta Commons Validator 517–520
 Jakarta Struts, integrating Web Flow with 619–620
 Jakarta Tiles 549, 556
 creating controllers 554, 556
 views 550–554
 JAR files 670
 Java
 aspect-oriented programming (AOP) 123
 SimpleJdbcTemplate class 174
 Timer class 457–460

- Java (*continued*)
 - creating timer tasks 458
 - starting timers 459
- Java 5
 - autoboxing 182
 - runtime targeting 187
 - simplified data access objects (DAOs) 182–183
 - simplifying JDBC 178, 180
 - varargs 182
- Java Data Objects (JDOs) 228
- Java EE Connector API (JCA) 9
- Java Management Extensions (JMX) 9
 - managing Spring beans with 466, 484
 - exporting as MBeans 467, 477
 - handling notifications 482, 484
 - remoting MBeans 477, 482
- Java Message Service (JMS) 10–11, 386–393, 407
 - ActiveMQ message broker 392–393
 - architecture 387–389
 - conventional (non-Spring)
 - sending/receiving 393–394
 - converting messages 402, 405
 - messageConverter property 405
 - sending/receiving 404–405
 - gateway support classes 405, 407
 - proxying 420–422
 - receiving messages 400, 402
 - runaway code 393, 395
 - sending messages 397, 399
 - setting default destinations 399–400
 - templates 395, 402
 - JmsTemplate 397
 - receiving messages 400, 402
 - sending messages 397, 399
 - setting default destinations 399–400
- Java Naming and Directory Interface (JNDI)
 - caching objects 447–448
 - conventional 443, 446
 - fallback objects 448–449
 - injecting objects 446, 449
 - lazily loading objects 448
 - mail sessions 452
 - retrieving DataSource from 443, 445
 - wiring objects from 442, 450
 - in Spring 2 449–450
- Java Persistence API (JPA) 194, 203
 - data access objects (DAOs) 202–203
 - entity manager factories 197–202
 - templates 194, 197
 - transaction managers 227–228
- Java Transaction API (JTA)
 - transaction managers 229
- java.beans.PropertyEditor interface 89
- java.util.Map collection 53
- java.util.Properties collection 53
- JavaBeans 1.00-A specification 4
- JavaDocs 669
- JavaMailSenderImpl 451–452
- JavaScript 652–655
- JavaServer Faces (JSF) 643, 648
 - exposing application context in 648
 - integrating Web Flow with 620, 622
 - resolving JSF-managed properties 644–645
 - resolving Spring beans 646
 - using Spring beans in pages 646, 648
- JavaServer Page (JSP) 500–501
 - templates 540, 542, 547, 549
- JaxRpcPortProxyFactoryBean 334, 337
- JaxRpcServicePostProcessor 338
- JBoss 165
- JBoss AOP 122
- JDBC 170–183
 - boilerplate code 172
 - compared to object-relational mapping (ORM) 203
 - data access object (DAO) support classes 180–183
 - exception heirarchy 159
 - inserting rows into databases 170
 - querying rows from databases 172
 - runaway code 170, 173
 - templates 173–180
 - JdbcTemplate 174, 176
 - transaction managers 226–227
 - updating rows in databases 171
- JDBC abstraction 8
- JDBC driver-based data source 168–169
- JdbcDaoImpl 259–260
 - authoritiesByUsernameQuery property 260
 - usersByUsernameQuery property 260
 - wiring 259
 - See also* AuthenticationDao
- JdbcDaoSupport base class 181
- JdbcDaoSupport class
 - getConnection() 164
- JdbcTemplate 174, 176
- JdbcTemplate class 162, 173
 - DataSource 174
 - named parameters 177
 - query() 176
 - querying data using 175
 - SQLException 175
 - update() 175

- jdbcTemplate instance variable 700
 - jdbcTemplate property 174
 - JDO. *See* Java Data Objects (JDOs)
 - JDOM-based message endpoints 355, 357
 - JdoTransactionManager 228
 - JEE namespace 166, 428–429
 - JEE specification 442
 - <jee:jndi-lookup> 166, 449–450, 452
 - <jee:local-slsb> 428
 - <jee:remote-slsb> 428
 - JeeBeanFactoryPostProcessor 436
 - JeeEjbBeanFactoryPostProcessor 436
 - JMS. *See* Java Message Service (JMS)
 - JMSException subclasses 395
 - JmsGatewaySupport class 405, 407
 - JmsProxyFactoryBean 420–421
 - JmsServiceExporter 418–419
 - JmsTemplate 395, 397, 402
 - convertAndSend() 404
 - receive() 401, 408
 - receiveAndConvert() 404
 - receiving messages 400, 402
 - send() 398
 - sending messages 397, 399
 - wiring 397
 - wiring default destinations 399–400
 - JmsTemplate102 396
 - JMX. *See* Java Management Extensions (JMX)
 - JMXConnectorServer 478
 - JNDI data sources 165, 167
 - JNDI. *See* Java Naming and Directory Interface (JNDI)
 - jndiName property 166–167
 - LocalStatelessSessionProxyFactoryBean 427
 - JndiObjectFactoryBean 166–167, 429–430, 445, 448–452
 - cache property 448
 - defaultObject property 449
 - JndiProxyFactoryBean 429
 - Johnson, Rod 5
 - joinpoints 120
 - method 124
 - JPA. *See* Java Persistence API (JPA)
 - JpaDaoSupport class
 - getJpaTemplate() 203
 - JpaDialect 228
 - JpaTemplate class 162, 194
 - entityManagerFactory property 195
 - find() 197
 - JpaTransactionManager 228
 - jpaVendorAdapter property 201
 - JPS. *See* Java Persistence API (JPA)
 - JSF. *See* JavaServer Faces (JSF)
 - JSP tags 86, 298–300
 - JSP. *See* JavaServer Page (JSP)
 - JSR-181 annotations 330–333
 - Jsr181HandlerMapping 332
 - Jsr181WebAnnotations 333
 - JTA. *See* Java Transaction API (JTA)
 - JtaTransactionManager 229
 - JUnit 680–684
 - setting up tests in 682–683
 - tearing down tests in 683–684
 - JUnit 4
 - gienah-testing 702, 704
 - junit.framework.TestCase class 681
- ## K
-
- key attribute
 - element 57
 - Knight interface 18
 - Knight object 21
 - KnightOfTheRoundTable class 19–20, 26, 29
- ## L
-
- <lang:bsh> 111
 - <lang:groovy> 110
 - <lang:inline-script> 113
 - <lang:jruby> 109
 - <lang:property> 112
 - lazy loading 201
 - objects 448
 - lazy-init attribute 427
 - LDAP repositories 264–271
 - LdapAuthenticationProvider 264–269
 - LdapShaPasswordEncoder 268
 - lib/ directory 671
 - Lingo 417–418
 - exporting services 418–419
 - <list> 52
 - values 54
 - load() 190
 - LocalContainerEntityManagerFactoryBean 198, 200
 - loadTimeWeaver property 201
 - LocaleEditor 89
 - LocalEntityManagerFactoryBean 198–199
 - LocalSessionFactoryBean 186
 - dataSource property 187
 - hibernateProperties property 187
 - mappingResources property 187

- LocalStatelessSessionProxyFactoryBean 426, 429
 - jndiName property 427
- location property 97
- Log4j 676–677
- loginFormUrl property 289
- <lookup-method> 84–85

- M**

- macros
 - FreeMarker 566
 - Velocity 562
- mail senders
 - configuring 451, 453
 - Java Naming and Directory Interface (JNDI)
 - mail sessions 452
 - wiring into service beans 453
 - mail sessions, Java Naming and Directory Interface (JNDI) 452
- MailSender interface 450, 452
- main() 13
- <map> 53, 56
- mapping
 - complex types 337–338
 - Hibernate files 186–187
 - messages to endpoints 363–364
 - requests to controllers 502–506
 - SQL queries 205
- mapping arrays 338, 340
- mapping requests
 - to JSR-181 annotated beans 332–333
 - to XFireExporter 329–330
- mapRow() 176, 179
- MarketingMdb class 409
- MarketingMdp class 409–410, 413–414, 416
 - onMessage() 412
- marshalers
 - message 364, 367
 - web service templates 379–380
- Marshaller interface 364
- marshalSendAndReceive() 379
- Maven 1 675
- Maven 2
 - adding Spring as dependency 672, 674
 - adding Spring Web Flow 584
 - Spring modules available in repository 672
- MBean servers (MBean agents) 467
- MBeanExporter 467, 470
 - registrationBehaviorName property 476–477
- MBeanProxyFactoryBean 480–481
- MBeans 472–473, 477
 - dynamic 467
 - exporting Spring beans as 467, 477
 - defining operations/attributes 472–473
 - exposing methods by name 471–472
 - metadata-driven 474, 476
 - object name collisions 476–477
 - open 467
 - proxying 480, 482
 - remoting 477–482
 - accessing 479–480
 - exposing 478
 - standard 467
- MBeanServerConnectionFactoryBean 479–480
- MD5 encoding 261
- MDBs. *See* message-driven beans (MDBs)
- MDPs. *See* message-driven POJOs (MDPs)
- message brokers 387
 - ActiveMQ 392–393
- message endpoints 353
 - options 354
- message listeners
 - containing 410–411
 - creating 408, 412
 - transactional message-driven POJOs (MDPs) 411–412
- message payloads, marshaling 358, 360
- MessageConverter interface 402, 404
- messageConverter property 405
- MessageDispatcherServlet 361, 363
- message-driven beans (MDBs) 408
- message-driven POJOs (MDPs) 407, 416
 - converting messages 415–416
 - creating message listeners 408, 412
 - containing message listeners 410–411
 - transactional (MDPs) 411–412
 - writing 412, 416
- MessageListenerAdapter 412–414
- messages
 - internationalization (I18N) of 99
 - mapping to endpoints 363–364
 - rendering externalized
 - in JSPs 544, 547
 - sample XML 348, 353
 - forging data contracts 349, 353
 - sending with web service templates 377, 379
 - with service endpoints 353, 360
 - JDOM-based message endpoints 355, 357
 - marshaling message payloads 358, 360
- MessageSource interface 99
- messaging 385–386, 422
 - asynchronous 385–386
 - datacentric 390

- messaging (*continued*)
 - destinations defined 388
 - indirection 387
 - Java Message Service (JMS) 386, 393, 407
 - architecture 387, 389
 - benefits 390–391
 - conventional (non-Spring) 393–394
 - converting messages 402, 405
 - gateway support classes 405, 407
 - runaway code 393, 395
 - setting up ActiveMQ message broker 392–393
 - templates 395, 402
 - message brokers 387
 - message-based RPC 416, 422
 - exporting services 418–419
 - Lingo 417–418
 - proxying JMS 420, 422
 - message-driven POJOs (MDPs) 407, 416
 - creating message listeners 408, 412
 - writing 412, 416
 - models 388–389
 - point-to-point model
 - location independence 391
 - publish-subscribe model
 - location independence 391
 - queues 388–389
 - synchronous 386, 390
 - topics 389
 - metadata 303–304
 - using to map controllers 505
 - MetadataMBeanInfoAssembler 474
 - method injection 79, 85
 - getter injection 80–85
 - method replacement 80–83
 - method invocations 300, 304
 - metadata 303–304
 - security aspects 301–302
 - method joinpoints 124
 - method replacement 80, 83
 - MethodBeforeAdvice interface 129
 - MethodDefinitionAttributes 303
 - MethodInterceptor interface 131–132
 - MethodInvokingJobDetailFactoryBean 465
 - MethodInvokingTimerTaskFactoryBean 464–465
 - MethodNameBasedMBeanInfoAssembler 471–472
 - MethodReplacer interface 83
 - methods
 - exposing by name 471–472
 - invoking on schedule 464, 466
 - MethodSecurityInterceptor 302
 - Minstrel class 26–29
 - mock/ directory 668
 - MockHttpServletRequest 687
 - MockHttpServletResponse 687
 - model MBeans 467
 - Model/View/Controller (MVC) framework 9
 - ModelAndView 497–498
 - asserting contents of 690, 692
 - ModelAndView object 492, 501, 534
 - modules 6–11
 - MultiActionController 508
 - MVC. *See* Model/View/Controller (MVC) framework
-
- N**
-
- name attribute, HomeController 498
 - named parameters
 - DAO support for 181–182
 - JDBC templates 176–178
 - NamedParameterJdbcDaoSupport 182
 - NamedParameterJdbcTemplate class 173, 177, 181
 - namingStrategy property 476
 - navigation 582–583
 - new creator 655
 - noncolonized names 351
 - nonrepeatable reads 235
 - non-Spring beans, injecting 85, 88
 - NotificationListener interface 484
 - NotificationPublisherAware interface 482–483
 - notifications
 - handling 482, 484
 - listening for 484
 - <null/> 58, 63
 - NullUserCache 263
 - numberToolAttribute property
 - VelocityViewResolver bean 560
-
- O**
-
- object name collisions
 - MBean 476–477
 - Object parameter 176
 - object references, injecting with constructors 43, 45
 - objectDefinitionSource property 294
 - object-relational mapping (ORM)
 - compared to JDBC 203
 - frameworks 184
 - integration module 9

- objects
 - caching 447–448
 - fallback 448–449
 - injecting 446, 449
 - lazily loading 448
 - mocking 687, 689
 - wiring 442, 450
 - wiring in Spring 2 449–450
 - onApplicationEvent() 102
 - open MBeans 467
 - operation attribute 300
 - operational contracts 349
 - operations, MBean, using interfaces to
 - define 472–473
 - orders
 - adding to 604–605
 - building 601, 605
 - completing 605, 608
 - taking payments 605, 607
 - displaying 602–603
 - submitting 607–608
 - OrderService interface 23
 - OrderServiceImpl class 22
 - org.springframework.beans.factory.BeanFactory 33–34
 - org.springframework.beans.propertyeditors.URLEditor 89
 - org.springframework.context.ApplicationContext interface 33
 - org.springframework.context.ApplicationEvent abstract class 102
 - org.springframework.context.ApplicationListener interface 103
 - ORM. *See* object-relational mapping (ORM)
 - <output-mapper> 617
- P**
-
- PagingNotificationListener 484
 - ParameterizableViewController 508
 - ParameterizedRowMapper class 179
 - parent attribute 74
 - parent beans 73–78
 - password encoders 261
 - password property 168
 - PasswordComparisonAuthenticator 267
 - PasswordDaoAuthenticationProvider 254
 - See also* ProviderManager
 - passwords
 - comparing 267, 269
 - encrypted 260, 262
 - PaymentProcessor class 606
 - payments 605, 607
 - persistence frameworks, Hibernate 159
 - persistence platform agnostic exceptions 159–160
 - persistence.xml file 198
 - persistence-layer code 494
 - PersistenceProvider interface
 - createContainerEntityManagerFactory() 197
 - createEntityManagerFactory() 197
 - persistenceUnitName property 199
 - persisting data 156, 219
 - caching 208, 218
 - annotation-driven 217–218
 - caching solutions 210, 215
 - proxying beans for 215, 217
 - data sources 165, 169
 - JDBC driver-based 168–169
 - JNDI 165, 167
 - pooled 167–168
 - Hibernate 183, 194
 - data access objects (DAOs) 190–191
 - Hibernate 3 contextual sessions 192, 194
 - templates 186, 190
 - versions 185–186
 - iBatis 203, 208
 - data access objects (DAOs) 207–208
 - templates 204, 207
 - Java Persistence API (JPA) 194, 203
 - data access objects (DAOs) 202–203
 - entity manager factories 197, 202
 - templates 194, 197
 - JDBC 170, 183
 - data access object (DAO) support classes 180, 183
 - runaway code 170, 173
 - templates 173, 180
 - overview 156–157
 - tiers 157, 165
 - data access object (DAO) support classes 163, 165
 - exceptions 158, 161
 - templating 161, 163
 - phantom reads 235
 - phone numbers 595–596
 - Pitchfork 435–437
 - annotations supported by 435
 - declaring interceptors using annotations 438–439
 - injecting resources by annotation 437–438
 - plain-old Java objects (POJOs) 4, 34
 - creating message listeners 408, 412
 - message-driven (MDPs) 407, 416
 - writing 412, 416

- PlatformTransactionManager 232
 - pointcuts 120, 132, 136
 - combining with advisors 134
 - declaring regular expression 133–134
 - defining AspectJ 135–136
 - point-to-point messaging model 388–389
 - location independence 391
 - POJOs. *See* plain-old Java objects (POJOs)
 - pool-configuration properties 168
 - pooled data sources 167–168
 - Portlet Model/View/Controller (MVC) 10
 - postProcessAfterInitialization() 93
 - postProcessBeanFactory() 95
 - postProcessBeforeInitialization() 93
 - postprocessing beans 93, 95
 - writing 93–94
 - processCancel() 526
 - processFinish() 525–526
 - processing
 - form submissions 517, 520
 - forms with wizards 524–526, 528
 - programmatic transaction management 224
 - programming transactions 229, 232
 - prompting users to log in 286, 291
 - <prop> 57
 - propagation behavior 233, 235
 - properties
 - abstracting common 76, 78
 - externalizing configuration 96, 99
 - injecting 46, 58
 - referencing other beans 48, 52
 - values 47–48
 - wiring nothing (null) 58
 - overriding inherited 76
 - Properties class 56
 - <property> 12, 47
 - autowiring 63
 - value attribute 48
 - property editors, registering custom 88, 92
 - property inheritance 75
 - PropertyEditorSupport class 89
 - PropertyPlaceholderConfigurer 96–97
 - <props> 53
 - prototype scoping option 65
 - ProviderManager 253–256
 - proxied classes 124
 - proxied inner beans 216–217
 - proxies 121
 - ProxyFactoryBean 136–139
 - proxying
 - beans for caching 215, 217
 - filters 278–284
 - Java Message Service (JMS) 420–422
 - MBeans 480, 482
 - session beans (EJB 2.x) 426–430
 - transactions 238–241
 - publish-subscribe messaging model 389
 - location independence 391
 - pure-POJO aspects 145, 149
-
- ## Q
- QA. *See* Quality Assurance (QA)
 - Quality Assurance (QA) 679
 - Quartz scheduler 460, 464
 - creating jobs 460–461
 - scheduling cron jobs 462–463
 - scheduling jobs 461–462
 - starting jobs 464
 - QuartzJobBean 460–461
 - query() 176
 - varargs 180
 - querying data
 - JdbcTemplate class 175, 179
 - SimpleJdbcTemplate class 179
 - querying rows from databases
 - JDBC 172
 - Quest interface 17–19
 - queues 388–389
-
- ## R
- read-only transactions 236–237
 - reads 235
 - receiveAndConvert() 404
 - receiving messages 404–405
 - conventional (non-Spring) Java Message Service (JMS) 394
 - JmsTemplate 400, 402
 - ref attribute element 44
 - reference document 669
 - referencing other beans 48, 52
 - injecting inner beans 51–52
 - refresh-check-delay attribute 113
 - refreshing scripted beans 112–113
 - RegexMethodPointcutAdvisor class 134
 - registerCustomEditor() method 91
 - registering
 - BeanPostProcessor interface 94–95
 - custom editors 92
 - custom property editors 88, 92
 - flow definitions 588
 - Spring plug-in with Struts 626–627

- registrationBehaviorName property 476–477
 - regular expression pointcuts 133–134
 - Perl5RegexpMethodPointcut class 133
 - reimplement() 83
 - remote exceptions 308
 - Remote Method Invocation (RMI) 309, 316
 - configuring 313, 316
 - exporting 312, 316
 - wiring 310, 312
 - remote methods, JavaScript and 654–657
 - remote objects, defining 651, 654
 - remote procedure calls (RPCs) 307–308
 - Lingo 418
 - message-based 416, 422
 - Lingo 417, 419
 - proxying Java Message Service (JMS) 420, 422
 - remote services 306–309, 342
 - Burlap 316–322
 - Hessian 316–322
 - HttpInvoker 322–326
 - Remote Method Invocation (RMI) 309–316
 - web services 326, 333, 340–341
 - JaxRpcPortProxyFactoryBean 334, 337
 - JSR-181 annotations 330, 333
 - mapping arrays 338, 340
 - mapping complex types 337–338
 - proxying with XFire client 340–341
 - XFire 326, 330
 - RemoteAuthenticationProvider 254
 - See also* ProviderManager
 - remoting 10
 - MBeans 477, 482
 - accessing 479–480
 - exposing 478
 - proxying 480, 482
 - <replaced-method> 82
 - request attributes, exposing
 - FreeMarker 566–567
 - Velocity 561
 - requests
 - handling with controllers 506, 509, 511–512, 520, 528, 531
 - lifecycles 491–492
 - mapping to controllers 502–506
 - resource bundles, resolving views from 539–540
 - Resource objects 34
 - ResourceBundleMessageSource 100, 546
 - ResourceBundleViewResolver 539–540
 - resource-ref attribute 167
 - resourceRef property 166–167
 - Rich Site Summary (RSS) 576, 578
 - RMI. *See* Remote Method Invocation (RMI)
 - RmiProxyFactoryBean 311
 - RmiServiceExporter 314
 - RoleVoter 274
 - rollback rules 237
 - RowMapper object 176
 - mapRow() 176, 179
 - RPC. *See* remote procedure calls (RPCs)
 - RSS. *See* Rich Site Summary (RSS)
 - Ruby
 - adding new methods 79
 - scripting beans in 109
 - run-as managers 251
 - RunAsImplAuthenticationProvider 254
 - See also* ProviderManager
 - runaway code 170, 173, 393–395
 - runtime 121
- ## S
-
- salt sources 261
 - saveOrUpdate() 189
 - sayGreeting() 12, 14
 - ScheduledTimerTask 458–459
 - SchedulerFactoryBean 464
 - scheduling tasks 456, 466
 - invoking methods on schedule 464, 466
 - Java Timer class 457, 460
 - creating timer tasks 458
 - delaying start of timers 459–460
 - scheduling timer tasks 458–459
 - starting timers 459
 - Quartz scheduler 460–464
 - scheduling cron jobs 462–463
 - scheduling jobs 461–462
 - starting jobs 464
 - timer tasks 458–459
 - schema declarations, Spring Modules 212
 - scope attribute 65
 - scopes 593
 - scoping beans 66
 - scoping options 65
 - scripting beans 106–114
 - BeanShell 110–111
 - Groovy 110
 - injecting properties 111–112
 - refreshing 112–113
 - Ruby 109
 - writing inline 113–114
 - secure channels 297
 - security context 285–286

- security exceptions 250, 291, 293
 - authorization exceptions 292–293
- security-layer code 494
- send() 398
- sendAndReceive() 377
- service beans, wiring mail senders into 453
- service endpoints 364
 - messages with 353, 360
 - JDOM-based message endpoints 355, 357
 - marshaling message payloads 358, 360
- service objects, accessing of DAOs 157
- service-layer code 494
- service-oriented architecture (SOA) 326
- servlet filters 276
 - authentication-processing 277
 - exception translation 277
 - filter security interceptors 278
 - integration 277
 - proxying 279, 281
 - with Acegi 275
- servlet listeners 494
- <servlet-name> 493
- session attributes 561–567
- session beans
 - EJB 2.x 426, 430
 - EJB 3 429–430
- session scoping option 65
- SessionFactory interface 192
- SessionFactory property 186
- <set> 52, 54
- setApplicationContext() 40, 105
- setAsText(String value) 89
- setBeanFactory() 105
- setBeanName() 104
- setComplete() 698
- setGreeting() 12–13
- setOrderService() 23
- setQuery() 20
- setRollbackOnly() 231
- setter injection
 - compared to constructor injection 45, 253
 - versus constructor injection 44
- setUp() 683
- SimpleFormController 508–509, 514, 520
- SimpleJdbcDaoSupport 182
- SimpleJdbcTemplate 174, 178
- SimpleJdbcTemplate class
 - Java 174
 - querying data using 179
 - RowMapper object
 - mapRow() 179
- SimpleMappingExceptionResolver 531–532
- SimpleMessageListenerContainer 411
- SimpleRemoteStatelessSessionProxyFactoryBean 426–427, 429
- SimpleTriggerBean 461–462
- SimpleUrlHandlerMapping 329, 503–504, 585
- SimpleWsd111Definition 371
- Since 448
- singAfter() 29
- singBefore() 29
- SingleConnectionDataSource 169
- singleton beans, loading 37
- singleton scoping option 65
- SOA. *See* service-oriented architecture (SOA)
- SoapFaultMappingExceptionResolver 367
 - exceptionMappings property 368
- SoapServiceExporter 326
- software testing, Quality Assurance (QA) 679
- special beans 92, 106
 - awareness 103, 106
 - BeanNameAware interface 104–106
 - BeanFactoryPostProcessor interface 95–96
 - decoupling with application events 101–103
 - externalizing configuration properties 96, 99
 - postprocessing beans 93–95
 - resolving text messages 99, 101
- Spring 4–6, 30
 - basic example 11, 14
 - data access exceptions 158, 161
 - catch blocks 160–161
 - persistence platform agnostic 159–160
 - data access object (DAO) support classes 163, 165
 - data access tiers 157, 165
 - DAO support classes 163, 165
 - exceptions 158, 161
 - templating 161, 163
 - directory structure 669
 - downloading 668–669, 671
 - integrating with 624, 632–633, 636, 642–643, 648, 663, 665
 - JavaDocs 669
 - mixing with Enterprise JavaBeans (EJBs) 424
 - modules available in Maven 2 repository 672
 - participation in testing 684–685
 - reference document 669
 - templating 161, 163
 - transaction management support 224–225
 - website 668
- Spring 2 449–450
- Spring 2.0
 - declaring transactions 241, 243
 - JNDI data sources 166–167

- Spring AOP 122, 125
 - advice 127
- Spring aspects 140–141
- Spring BeanDoc, autowiring 64
- Spring beans
 - exporting as MBeans 467, 477
 - defining operations/attributes 472–473
 - exposing methods by name 471–472
 - metadata-driven 474, 476
 - object name collisions 476–477
 - loading into Tapestry pages 639, 641
 - managing with Java Management Extensions (JMX) 466, 484
 - exporting as MBeans 467, 477
 - handling notifications 482, 484
 - remoting MBeans 477, 482
 - resolving 646
 - using in JavaServer Faces (JSF) pages 646, 648
 - wiring Enterprise JavaBeans (EJBs) into 430–431
- Spring IDE, autowiring 64
- Spring Model-View-Controller (MVC) 490, 502
 - building homepages 495–502
 - configuring DispatcherServlet 492–495
 - request lifecycles 491–492
- Spring Modules 210–212
- Spring MVC controllers 491–492
 - unit-testing 685, 687, 689–690, 692
- Spring MVC conversational applications 583
- spring object factory 635
- Spring Security 248, 304
 - access decisions managers 271–272, 275
 - voter abstinence 275
 - voting 272–274
 - authenticating users 252, 271
 - against databases 256, 264
 - against LDAP repositories 264, 271
 - ProviderManager 253, 256
 - authentication entry points 287
 - authentication providers 254
 - authentication-processing filters 287
 - credentials 250
 - elements 249
 - JSP tags 298
 - method invocations 300, 304
 - metadata 303–304
 - security aspects 301–302
 - password encoders 261
 - principals 250
 - servlet filters 276–278
 - order of configuration 284
 - view-layer security 297–300
 - web security 275, 297
 - enforcing 293–294
 - prompting users to log in 286, 291
 - proxying filters 278, 284
 - secure channels 294, 297
 - security context 285–286
 - security exceptions 291, 293
- Spring Web Flow. *See* Web Flow
- spring.jar file 671
- <spring:message> 545, 547
- SpringBindingActionForm 620
- Spring-enabled Enterprise JavaBeans (EJB 2.x) 431, 434
- Spring-managed beans
 - accessing 659, 661–663
- Spring-OXM
 - marshaling options 365
- SpringRunner 704
- SpringTapestryEngine 637, 639
- Spring-WS 344–348, 353, 373, 382–383
 - message factory implementations 375
 - message senders 375
 - messages with service endpoints 353, 360
 - JDOM-based message endpoints 355, 357
 - marshaling message payloads 358, 360
 - service configuration 362
 - web service gateway support 381–382
 - web service templates 374, 380
 - marshalers on client side 379–380
 - sending messages 377, 379
- wiring 361, 373
 - configuration 361, 363
 - deploying 373
 - endpoint exceptions 367, 369
 - mapping messages to endpoints 363–364
 - message marshalers 364, 367
 - service endpoints 364
 - WSDL files 369, 373
- SQL queries, mapping 205
- SQLException 175
- SqlMapClient interface 204
- SqlMapClientDaoSupport 207
- SqlMapClientFactoryBean 205
- SqlMapClientTemplate 204–205
- src/ directory 668
- standard MBeans 467
- start states 593–594
- starting
 - Quartz jobs 464
 - timers 459
- <start-state> 593–594

- states 589–590
 - String parameter 176
 - StringArrayPropertyEditor 89
 - StringTrimmerEditor 89
 - Struts 624, 633
 - actions 627, 629–632
 - integrating Web Flow with 619–620
 - registering Spring plug-in with 626–627
 - Struts 2 632–633, 636
 - subbeaning 74, 78
 - subflows 614–618
 - <subflow-state> 617
 - substates 614, 618
 - supports() 272
 - synchronous messaging 386, 390
- T**
-
- tags, form-binding JavaServer Page (JSP) 543
 - Tapestry 636, 642
 - loading Spring beans into Tapestry pages 639, 641
 - Tapestry 3 637, 641
 - Tapestry 4 641–642
 - targets 121
 - tasks
 - scheduling 456, 466
 - invoking methods on schedule 464, 466
 - Java Timer class 457, 460
 - Quartz scheduler 460, 464
 - timer
 - creating 458
 - scheduling 458–459
 - tearDown() 684
 - tearing down tests in JUnit 684
 - template classes 162
 - HibernateTemplate 162
 - JdbcTemplate 162, 173
 - JpaTemplate class 162
 - NamedParameterJdbcTemplate 173
 - relationship to DAO support classes 164
 - SimpleJdbcTemplate 174
 - template methods 161
 - template views, resolving 534–535, 537
 - templates
 - Hibernate 186, 190
 - annotated domain objects 187–188
 - classic XML mapping files 186–187
 - iBATIS 204, 207
 - SQL maps 205–206
 - SqlMapClientTemplate 204–205
 - Java Message Service (JMS) 395, 402
 - JmsTemplate 397
 - receiving messages 400, 402
 - sending messages 397, 399
 - setting default destinations 399–400
 - Java Persistence API (JPA) 194, 197
 - JavaServer Page (JSP) 542, 549
 - binding form data 542, 544
 - displaying errors 547, 549
 - rendering externalized messages 544, 547
 - JDBC 173, 180
 - JdbcTemplate 174, 176
 - named parameters 176, 178
 - simplifying in Java 5 178, 180
 - transaction proxy 240–241
 - Velocity 557, 564
 - binding form fields 561, 564
 - configuring engines 558–559
 - defining views 557–558
 - exposing request/session attributes 561
 - formatting dates/numbers 560–561
 - resolving views 559–560
 - web services 374, 380
 - marshalers on client side 379–380
 - sending messages 377, 379
 - templating data access 161, 163
 - test methods 681
 - setUp() 683
 - tearDown() 684
 - testing 679, 705
 - integration-testing 692–694, 696, 699, 702, 704
 - JUnit 680, 682–684
 - Spring participation in 684–685
 - types of 680
 - unit-testing Spring MVC controllers 685, 687, 689–690, 692
 - TestingAuthenticationProvider 254
 - See also* ProviderManager
 - text messages, resolving 99, 101
 - ThrowawayController 508, 528, 531
 - ThrowawayControllerHandlerAdapter 530
 - ThrowsAdvice interface 130–131
 - afterThrowing() 130
 - tiger/ directory 668
 - <tile:insert> 550
 - Tiles 549, 556
 - creating controllers 554, 556
 - views 550, 554
 - configuring 552
 - resolving 553–554
 - TilesConfigurer 552
 - TilesJstView 553

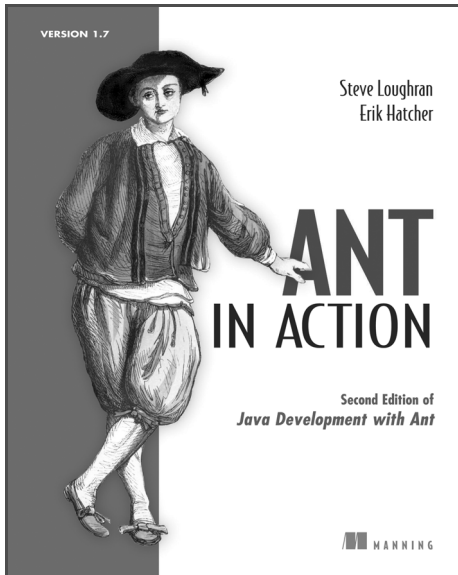
- timeouts 237
 - Timer class 457, 460
 - creating timer tasks 458
 - delaying start of timers 459–460
 - scheduling timer tasks 458–459
 - starting timers 459
 - timer tasks, creating 458
 - TimerFactoryBean 457, 459
 - timers, starting 459
 - Tomcat 165
 - TopLinkJpaVendorAdapter property 201
 - Trang 350
 - transaction managers 225, 229
 - Hibernate 227
 - Java Data Objects (JDOs) 228–229
 - Java Persistence API (JPA) 227–228
 - Java Transaction API (JTA) 229
 - JDBC 226–227
 - transactional message-driven POJOs (MDPs) 411–412
 - transactional objects, integration-testing 696, 698–699
 - transactional testing 684
 - transactionAttributes property 239
 - TransactionCallback interface 231
 - doInTransaction() 232
 - execute() 231
 - setRollbackOnly() 231
 - TransactionDefinition 233
 - transactionManager property 412
 - TransactionProxyFactoryBean 238
 - transactions 221–225, 246
 - atomicity 223
 - consistency 223
 - declaring 232, 245
 - annotation-driven transactions 243, 245
 - attributes 233, 237
 - proxying 238, 241
 - Spring 2.0 241, 243
 - declarative 224
 - described 221
 - durability 223
 - isolation 223
 - managers 225
 - programmatically adding 231
 - programming 229, 232
 - propagation behaviors 234
 - read-only, propagation behaviors 237
 - timeouts, propagation behaviors 237
 - transaction managers 225, 229
 - Hibernate 227
 - Java Data Objects (JDOs) 228–229
 - Java Persistence API (JPA) 227–228
 - Java Transaction API (JTA) 229
 - JDBC 226–227
 - <transition> 608
 - transitions 590
 - Trigger class 461
 - tx namespace 241
 - <tx:advice> 242
 - <tx:annotation-driven> 244
 - <tx:attribute> 242
 - <tx:method> 242
 - txProxyTemplate 240
-
- ## U
- UnanimousBased wiring 272
 - unit testing 16–17, 445
 - Unmarshaller interface 364
 - update() 175
 - updateTable 657
 - updating rows in databases, JDBC 171
 - url property 167
 - URLEditor 89
 - UrlFilenameViewController 508
 - UserCache interface 263
 - UserDetailsService interface 257
 - userDetailsService property 257
 - userDnPatterns property 266
 - username property 168
 - usersByUsernameQuery property 260
-
- ## V
- validate() 528
 - validatePage() 527
 - validating form input 517
 - validation 8
 - Validator interface 515, 517–518
 - Validator object
 - validate() 528
 - validatePage() 527
 - validatorFactory bean 518
 - value attribute element 44
 - values, injecting 47–48
 - varargs 178, 182
 - query() 180
 - Velocity macros 562
 - Velocity templates 557, 564
 - binding form fields 561, 564
 - configuring engines 558–559
 - defining views 557–558

- Velocity templates (*continued*)
 - exposing
 - request attributes 561
 - session attributes 561
 - formatting
 - dates 560–561
 - numbers 560–561
 - resolving views 559–560
 - VelocityConfigurer bean 558–559
 - VelocityViewResolver bean 559, 561
 - dateToolAttribute 560
 - exposeRequestAttributes 561
 - exposeSessionAttributes 561
 - exposeSpringMacroHelpers 563
 - numberToolAttribute 560
 - view beans
 - resolving 537, 540
 - declaring in separate XML files 538
 - from resource bundles 539–540
 - view resolvers
 - choosing 540–541
 - declaring 499–500
 - using multiple 540–541
 - viewClass property 537
 - view-layer security 297, 300
 - conditionally rendering content 298–299
 - displaying user authentication information 299–300
 - ViewResolver implementations 535
 - views 534, 556, 579
 - developing custom 576, 578
 - FreeMarker 564–569
 - binding form fields 567, 569
 - configuring engines 565–566
 - defining views 564–565
 - exposing request/session attributes 566–567
 - resolving views 566
 - JavaServer Page (JSP) templates 542, 549
 - binding form data 542, 544
 - displaying errors 547, 549
 - rendering externalized messages 544, 547
 - non-HTML output 569, 578
 - developing custom views 576, 578
 - Excel spreadsheets 570, 573
 - PDF documents 573, 575
 - resolving 534, 541
 - choosing view resolvers 540–541
 - FreeMarker 566
 - resolving view beans 537, 540
 - template views 535, 537
 - Velocity 559–560
 - Tiles 549–550, 554, 556
 - creating controllers 554, 556
 - Velocity templates 557, 564
 - binding form fields 561, 564
 - configuring engines 558–559
 - defining views 557–558
 - exposing request/session attributes 561
 - formatting dates/numbers 560–561
 - resolving views 559–560
 - <view-state> 595, 598, 602, 604–605
 - voting access decisions 272–274
-
- ## W
- weaving 121–122
 - web containers, initializing servlet listeners before servlets 494
 - Web Flow 581–622
 - advanced techniques 611, 618
 - decision states 612, 614
 - extracting subflows 614, 618
 - substates 614, 618
 - creating flows 591, 611
 - building orders 601, 605
 - completing orders 605, 608
 - end states 593–594
 - flow variables 591, 593
 - gathering customer information 594, 601
 - start states 593–594
 - installing 584, 588
 - configuring flow executors 586–587
 - FlowController 585–586
 - registering flow definitions 588
 - integrating with other frameworks 619, 622
 - Jakarta Struts 619–620
 - JavaServer Faces 620, 622
 - main elements 589–590
 - web module 10
 - web requests 490, 532
 - handling exceptions 531–532
 - handling requests with controllers 506, 509, 511–512, 520, 528, 531
 - mapping requests to controllers 502–506
 - Spring Model-View-Controller (MVC) 490–492, 495, 502
 - web security 275, 297
 - enforcing 293–294
 - prompting users to log in 286, 291
 - basic authentication 287, 289
 - form-based authentication 289, 291

- web security (*continued*)
 - proxying filters 278, 284
 - configuring proxies 283–284
 - multiple filters 281–282
 - servlet filters 279, 281
 - secure channels 294, 297
 - channel decisions 297
 - security context 285–286
 - security exceptions 291–293
- web service templates 374, 380
 - marshalers on client side 379–380
 - sending messages 377, 379
- web services 326, 333, 340–341
 - contract-first 344, 373, 382–383
 - defined 347, 353
 - messages with service endpoints 353, 360
 - overview 344–345, 347
 - web service gateway support 381–382
 - web service templates 374, 380
 - wiring 361, 373
 - JaxRpcPortProxyFactoryBean 334, 337
 - JSR-181 annotations 330, 333
 - mapping requests to JSR-181 annotated beans 332–333
 - mapping arrays 338, 340
 - mapping complex types 337–338
 - proxying with XFire client 340–341
 - XFire 326, 330
 - DispatcherServlet 329
 - mapping requests to XFireExporter 329–330
 - XFireExporter 328–329
- Web Services Metadata for the Java Platform (JSR-181), annotations 330
- web views 534, 556, 579
 - FreeMarker 564, 569
 - binding form fields 567, 569
 - configuring engines 565–566
 - defining views 564–565
 - exposing request/session attributes 566–567
 - resolving views 566
 - JavaServer Page (JSP) templates 542, 549
 - binding form data 542, 544
 - displaying errors 547, 549
 - rendering externalized messages 544, 547
 - non-HTML output 569, 578
 - developing custom views 576, 578
 - Excel spreadsheets 570, 573
 - PDF documents 573, 575
 - resolving 534, 541
 - choosing view resolvers 540–541
 - resolving view beans 537, 540
 - template views 535, 537
 - Tiles 549–550, 554, 556
 - creating controllers 554, 556
 - Velocity templates 557, 564
 - binding form fields 561, 564
 - configuring engines 558–559
 - defining views 557–558
 - exposing request/session attributes 561
 - formatting dates/numbers 560–561
 - resolving views 559–560
- web.xml files, FilterToBeanProxy 283
- webApplicationContext 648
- WebApplicationContextUtils 627
- WebApplicationContextVariableResolver 621, 648, 664
- web-layer code 494
- WebServiceTemplate 373
 - marshalSendAndReceive() 379
 - sendAndReceive() 377
- WebSphere 165
- WebWork 2 633, 636
- wired objects
 - integration-testing 694–696
- wiring 20–21
 - default destinations into JmsTemplate 399–400
 - defined 32
 - Enterprise JavaBeans 425, 431
 - into Spring beans 430–431
 - proxying session beans (EJB 2.x) 426, 430
 - Java Naming and Directory Interface (JNDI) objects 442, 450
 - in Spring 2 449–450
 - JmsTemplate 397
 - mail senders into service beans 453
 - nothing (null) 58
 - Remote Method Invocation (RMI) services 310, 312
 - Spring-WS services 361, 373
 - configuration 361, 363
 - deploying 373
 - endpoint exceptions 367, 369
 - mapping messages to endpoints 363–364
 - message marshalers 364, 367
 - service endpoints 364
 - WSDL files 369, 373
 - Struts actions in Spring 630–631
- wiring beans 32, 71, 73, 115
 - auto-wiring 58, 64
 - mixing with explicit wiring 63
 - shortcomings 63–64
 - types 59, 62

- wiring beans (*continued*)
 - containing 33, 40
 - ApplicationContext interface 35, 40
 - BeanFactory interface 34–35
 - controlling creation 64, 71
 - creating from factory methods 66, 68
 - destroying 68, 71
 - initializing 68, 71
 - scoping 66
 - creating 40, 45
 - declaring 40–41
 - injecting through constructors 42, 45
 - declaring child beans 73, 78
 - abstracting base bean type 74, 76
 - abstracting common properties 76, 78
 - declaring parent beans 73, 78
 - abstracting base bean type 74, 76
 - abstracting common properties 76, 78
 - injecting non-Spring beans 85, 88
 - injecting properties 46, 58
 - injecting simple values 47–48
 - referencing other beans 48, 52
 - wiring nothing (null) 58
 - method injection 79, 85
 - getter injection 83, 85
 - method replacement 80, 83
 - registering custom property editors 88, 92
 - scripting beans 106, 114
 - example 107–108
 - injecting properties 111–112
 - refreshing 112–113
 - scripting bean 108, 111
 - writing inline 113–114
 - special beans 92, 106
 - awareness 103, 106
 - BeanFactoryPostProcessor interface 95–96
 - BeanPostProcessor interface 93, 95
 - decoupling with application events 101, 103
 - externalizing configuration properties 96, 99
 - resolving text messages 99, 101
 - wizard controllers, building 521, 524
 - wizards
 - canceling 526
 - finishing 525–526
 - processing forms with 520–521, 524–526, 528
 - validating forms 526, 528
 - WSDL files 369, 373
 - predefined WSDL 371, 373
-
- ## X
- XFire 326, 330
 - DispatcherServlet 329
 - XFireClientFactoryBean 334, 340
 - XFireExporter 327–329
 - mapping requests to 329–330
 - XML files
 - declaring view beans in separate 538
 - splitting application context across multiple 493
 - XML mapping files, Hibernate 186–187
 - XML messages 348, 353
 - forging data contracts 349, 353
 - XML Schema (XSD) 349
 - XmlBeanFactory 21
 - creating 34
 - XmlFileViewResolver 538, 540–541
 - XmlFlowRegistryFactoryBean 588
 - XMLHttpRequest 649
 - XmlWebApplicationContext 36

MORE TITLES FROM MANNING



Ant in Action, Second Edition of Java Development with Ant

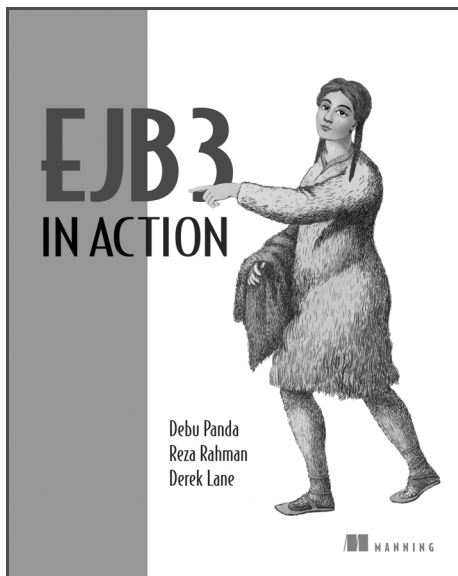
by Steve Loughran and Eric Hatcher

ISBN: 1-932394-80-X

600 pages

\$44.99

July 2007



EJB 3 in Action

by Debu Panda, Reza Rahman
and Derek Lane

ISBN: 1-933988-34-7

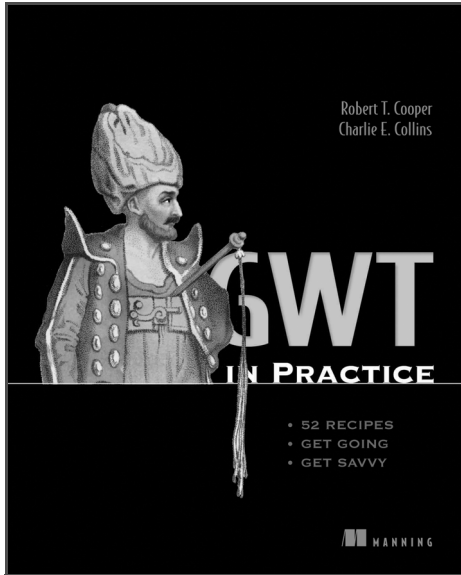
712 pages

\$49.99

April 2007

For ordering information go to www.manning.com

MORE TITLES FROM MANNING



GWT in Practice

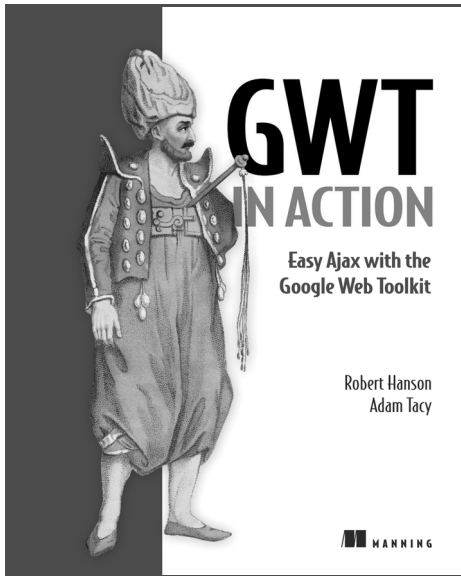
by Robert T. Cooper
and Charlie E. Collins

ISBN: 1-933988-29-0

450 pages

\$49.99

January 2008



GWT in Action

Easy Ajax with the Google Web Toolkit

by Robert Hanson and Adam Tacy

ISBN: 1-933988-23-1

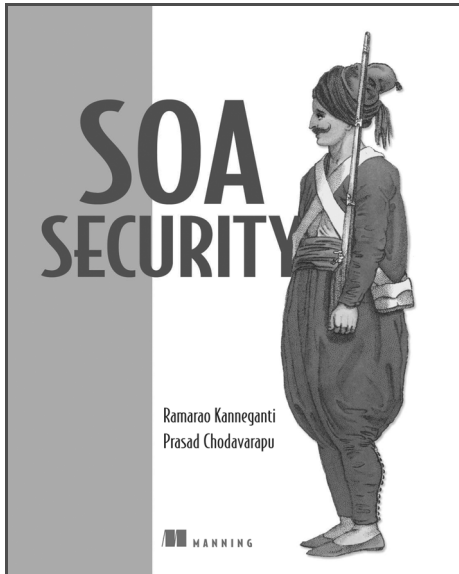
632 pages

\$49.99

June 2007

For ordering information go to www.manning.com

MORE TITLES FROM MANNING



SOA Security

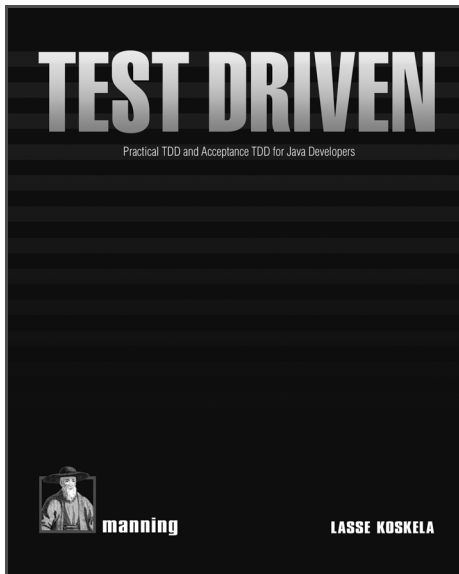
by Ramarao Kanneganti
and Prasad Chodavarapu

ISBN: 1-932394-68-0

350 pages

\$49.99

October 2007



Test Driven

*Practical TDD and Acceptance TDD
for Java Developers*

by Lasse Koskela

ISBN: 1-932394-85-0

470 pages

\$44.99

September 2007

For ordering information go to www.manning.com

MANNING EBOOK PROGRAM

All ebooks are 50% off the price of the print edition!

In the spring of 2000 Manning became the first publisher to offer ebook versions of all our new titles as a way to get customers the information they need quickly and easily. We continue to publish ebook versions of all our new releases, and every ebook is priced at 50% off the print version!

Go to www.manning.com/SpringinAction to download the ebook version of this book and have the information at your fingertips wherever you might be.

MANNING EARLY ACCESS PROGRAM



Get Early Chapters Now!

In 2003 we launched MEAP, our groundbreaking Early Access Program, to give customers who can't wait the opportunity to read chapters as they are written and receive the book when it is released. Because these are "early" chapters, your feedback will also help shape the final manuscript.

Our entire MEAP title list is always changing and you can find the current titles at www.manning.com