

Symbols

<< operator 146
! method 298
? form=post attribute 116
? method 288
?? method 288
?~ method 390
#> symbol 115

A

abstractions
 HTTP 161
 JavaScript language 189–193
 basic commands 190–192
 JSON handling 192–193
AcceptedResponse type 177
Accept-Language header 284
Access control 140
ACID (atomicity, consistency, isolation, durability) 274
Active Record pattern 46
actor usage 209
ActorPing function 210, 217
ActorRef 307
actors 74, 208–209
administration section, for auction application 41–43
Advanced Message Queue Protocol. *See* AMQP
afterCreate 90
afterSave 246–248
afterSpec function 321, 334

AJAX (Asynchronous Javascript and XML) 188–203
 basic 194–197
 JavaScript language
 abstractions 189–193
 basic commands 190–192
 JSON handling 192–193
 sophisticated 197–199
 testing 341–346
 using JSON forms
 with 200–202
 with LiftScreen
 mechanism 202–203
AJAX bidding interface, for auction details page 68–74
ajaxEditable 195
ajaxForm 72, 215
Akka actors 301
Akka toolkit, messaging
 with 300–309
 fault tolerance through supervisors 301–304
 Lift project interaction with Akka toolkit 304–307
AkkaCometActor 307–308
AlertJsCmd 190
allTemplateNodeSeq 133
AMQP (Advanced Message Queue Protocol) module
 example of 297–300
 support for 295–297
AMQP terms 296
AMQPAddListener 299–300
AMQPMessage 298–299
AMQPSEnder 297–298
annotations 311

AnyVar 129–130, 361
application lifecycle 162–166
 bootloader class 164
 ResourceServer
 object 163–164
 shutdown process 165–166
application models, Comet.
 See Comet application model
argument groups 379, 383
asDouble 205–206, 306
AsInt helper 174
AskName 213, 215
Asynchronous Javascript and XML. *See* AJAX
atomicity, consistency, isolation, durability. *See* ACID
attributes, of snippets 120–121
auction application 37–103
 auction catalog for 61–65
 auction details page 66–82
 AJAX bidding
 interface 68–74
 Comet
 implementation 74–82
 URLs for 66–68
basket 91–93
checkout 93–98
data models 46–50
 connecting to
 database 49–50
 schema definition 46–49
order creation 84–91
 attributing auctions to customers 87–91
 data models for 84–87

auction application (*continued*)
 page flow for 40–43
 administration 41–43
 auction detail and bidding 41
 checkout 41
 global elements 40
 home page 40
 PayPal payments 99–102
 Buy Now button for 102
 environment setup 99–102
 scaffolding 50–57
 CRUD generation 53–57
 prototype traits 50–53
 templates 43–46
 design workflow for 43
 structure of 43–46
 validation 57–59
 definitions 58
 displaying field errors 58–59
 Auction class 48–49, 57–58, 68, 72, 75, 89
 Auction model 61, 75
 auction/supplier relationship 57
 AuctionHelpers 64, 67–68, 75–76, 91–92
 AuctionInstanceHelpers 75–76, 79, 81
 AuctionMachine 88–90
 AuctionServer object 76–77
 AuctionUpdater 78–81
 authentication
 parameters 150–153
 external security models 152–153
 HttpAuthProtected 151–152
 Test and Testaccess 150–151
 Author class 279
 AuthorBooks 234
 AutoComplete widget 137–138

B

backslash method 193
 BadResponse type 177
 bang (!) method 209, 294
 BASE (Basically Available, Soft-state, Eventual consistency) 274
 BaseField 263
 BaseForeignKey 233

Basically Available, Soft-state, Eventual consistency.
 See BASE
 BDD (behavior-driven development) 318–319
 before_filter 148
 beforeSave 246–248, 253
 beforeSpec function 321, 334
 beginServicing 371
 behavior-driven development.
 See BDD
 BigDecimal 263
 BinaryField class 262–263
 BinaryTypedField 262
 binding snippet 115, 191
 bookstore
 with MongoDB data store 278–280
 with Squeryl project 268–272
 CRUD statements 270–271
 querying statements 271–272
 Bookstore Schema 271
 BooleanField 263
 Boot class 45, 66, 101–102, 111–113, 128, 164–165, 237, 267–268, 270, 275–276, 301, 334
 booting application 29–30
 Box type 70, 388–391
 browser.click method 344–345
 browser.isTextPresent method 344–345
 browser.open method 344
 build tool 20
 bulkDelete_!! method 245
 Buy Now button, for PayPal payments 102
 BuyNowSnippet 96–97, 102
 By class 239
 By parameter 239
 ByList type 239
 ByRef type 240
 BySql type 240

C

caching templates 365–366
 Cake pattern 329–331
 calculators, locale 284–286
 Call expression 192
 callbacks, lifecycle 247–248
 CDATA block 191
 Checkout screen 97

checkout, for auction application 41, 93–98
 class and companion object idiom 48
 classes, Scala language 376–378
 clients, content expiry 366
 Cobertura 322
 code coverage reports, and testing 322–323
 code, localized 288–289
 collections, Scala language 380
 Comet 14, 74, 207–208
 actors 208–209
 basic 209–211
 for auction details page 74–82
 support for 212–221
 testing 341–346
 CometActor 74, 77–80, 210–211, 342
 community 10
 compact method 193
 compile scope 28
 configurations 361–364
 multiple servers 363–364
 single server 362–363
 confirmScreen_? method 136
 connecting, to database 49–50
 connectivity, database 224–227
 console type 237
 consumer 296
 containers, servlet 348–351
 ContainerSerializer 361
 ContainerVar 358–359, 361, 373
 content embedding, and partial templates 110
 content expiry, client 366
 contents value 92
 continuous compilation mode 30
 control-affecting parameters 148
 Cookies 130–131, 334–335
 CouchDB 260, 274–276
 CouchMetaRecord types 275
 CouchRecord 275, 277
 CPU hot spots 302
 CreatedUpdated 47–48, 85–86, 229–230, 245–246
 critical state 356–361
 extended sessions 356–358
 Terracotta Web Sessions 358–361
 CRUD generation 53–57
 CRUD interface 60–61

CRUD statements 270–271
 CRUDify 37, 39, 42, 50, 53–57, 60, 90
 CSRF (cross-site request forgery) 342
 CSS class 114, 118
 CSS file 164
 CSS styling 56
 CSS-style transformers 68, 97, 114–120
 Customer class 51–52, 80, 86–87
 Customer model 86
 Customer.currentUser 70, 79, 85, 91, 94, 96

D

data 237–238
 querying 238–244
 updating and deleting 244–245
 data definition language. *See* DDL
 data models 46–50
 connecting to database 49–50
 for order creation 84–87
 schema definition 46–49
 data stores
 CouchDB 274–276
 MongoDB 276–278
 databases
 connecting to 49–50
 relational 266–272
 setting up 224–237
 defining Mapper instance 227–232
 installation and connectivity 224–227
 relationships 232–235
 schema creation and control 235–237
 DateTimeField 263
 DB object 243–244
 DB.addLogFunc method 251–252
 DB.defineConnectionManager 49–50, 227, 267
 DB.runQuery method 244, 258
 DB.use 252–254, 268, 270
 dbColumnName 57, 85–86, 88, 236
 DBLogEntry 251–252
 DBVendor class 49–50, 226–227
 DDL (data definition language) 267
 DecimalField 263
 declarations, DocType 111–112
 DefaultConnectionIdentifier 49–50, 253–254, 267–268, 270
 DefaultWebProject 27, 323
 delete_! method 244
 deleteCookie 131
 dependency injection. *See* DI
 deployment 347–374
 case studies 371–373
 Foursquare application 372
 Novell Vibe environment 372–373
 configurations 361–364
 multiple servers 363–364
 single server 362–363
 handling state 351–361
 distributing
 critical 356–361
 sticky session strategies 352–356
 servlet containers 348–351
 tools and techniques for 364–371
 built-in assistance 364–366
 monitoring 367–371
 details page, for auctions 66–82
 AJAX bidding interface 68–74
 Comet implementation 74–82
 URLs for 66–68
 Details snippet 68, 76, 78
 DI (dependency injection), and testing 327–334
 and Factory type 332–334
 and Injector type 332–334
 Cake pattern 329–331
 using functions 331–332
 dispatch guards 181–182
 dispatch method 64
 dispatch snippets 121–123
 dispatching
 HTTP, and web services 174–185
 methods 121–123
 DispatchLocSnippets parameter 149
 DispatchSnippet 63–64, 122–124, 129–130, 361
 distributed
 programming 293–316
 JEE integration 309–316
 messaging with Akka toolkit 300–309

with AMQP module 295–300
 DocType declarations, and markup validation 111–112
 domain-specific language. *See* DSL
 double percent symbol 27
 Double value 264
 DoubleField 264
 Driver class 267
 DSL (domain-specific language) 142, 176–179

E

eager evaluation, and attributes of snippets 120–121
 EAR (Enterprise Archives) 27
 EarlyResponse parameter 150
 eclipse command 387
 Eclipse IDE 386–387
 Either construct 127
 EJB3 (Enterprise JavaBeans 3) 309
 EmailField 264
 embedded documents 278
 embedding content, and partial templates 110
 Empty state 69
 endServicing 371
 Enterprise Archives. *See* EAR
 Enterprise JavaBeans 3. *See* EJB3
 EntityManager class 310
 entity-relationship diagram. *See* ERD
 EnumField 264
 ERD (entity-relationship diagram) 46, 83–84
 eventual consistency 274
 exception handlers 366
 exchange 296
 expiry, client content 366
 extended sessions 356–358
 Extensible Markup Language. *See* XML
 external security models 152–153

F

factories, custom resource 291
 Factory type 327, 332–333
 FactoryMaker 111, 333
 FadeIn 196
 Failure type 69, 72, 390–391

fault tolerance, through supervisors 301–304
 fictionalCount 121
 field method 132
 field types, custom 134–135
 fields
 mapped, custom 254–258
 Record system 262–265
 FIFO (first-in first-out) 296
 filter function 381
 filter predicate 181
 filtering resource 180
 FilterOrValidate 133
 find method 239
 findAll 61–62, 239–242, 277, 279
 findAllByInsecureSql 242–243
 findBooksByAuthor 313
 findComet 80
 first-in first-out. *See* FIFO
 flavor field 133
 for keyword 70
 ForbiddenResponse type 177
 for-comprehension 389–390
 FormBuilderLocator 133–134
 forms 131–136
 JSON, using AJAX
 with 200–202
 LiftScreen
 mechanism 132–135
 Wizard system 135–136
 Foursquare application, case study 372
 Foursquare engineering blog 280
 frameworks, for testing 318–323
 and code coverage reports 322–323
 Scala Specs 319–321
 ScalaCheck 321–322
 ScalaTest 318–319
 Full state 69
 Functional Programming 4
 functions, Scala language 380
 FunSuite type 319
 futures, Lift project interaction with Akka toolkit 307–309

G

Game actor 213, 217
 gauges 369–370
 getOrElse method 389–390
 getReference 314
 getSingleton 47, 51, 85–86, 229–230, 255, 357

global elements, for auction application 40
 graphical user interface. *See* GUI
 Gravatar widget 138–139
 green screen experiences 187
 guards, dispatch 181–182

H

H2 database 226, 267
 handlers, exception 366
 HAProxy software 352–354
 hashCode 286
 head element, resources of 110–111
 hello-service 301
 Helpers 63, 79, 206, 211, 263
 Helpers object 31–32, 115
 hidden fields 266
 Hidden parameter 65, 102, 146–148
 HideIfNoKids parameter 148
 higher-order functions 380
 highPriority 81, 210
 home page, for auction application 40
 HTML tags 93
 HTML5 standard, support for 112–113
 HTMLProperties 113
 HTTP (Hypertext Transfer Protocol)
 dispatching and web services 174–185
 HTTP dispatch DSL 176–179
 REST service 179–185
 lifecycles
 application 162–166
 request 166–170
 pipeline 161–170
 HTTP abstraction 161
 lifecycles 162–170
 URL rewriting 170–174
 advanced 173–174
 defining RewritePF partial function 171–173
 HTTP interface 368
 HTTP request pipeline 166
 HttpAuthProtected parameter 150–152
 HttpBasicAuthentication 152
 HTTPContext 130
 HTTPCookie type 130–131
 HttpDigestAuthentication 152

HttpRequest type 66, 130, 162, 167, 172, 285–286
 HTTPSession 358–359
 Hypertext Transfer Protocol. *See* HTTP

I

IBM DB2 database 267
 IDEs (Integrated Development Environments), configuring 384–387
 Eclipse IDE 386–387
 IntelliJ IDE 384–386
 IdPK 47–48, 85–86, 229–230, 245–246
 If / Unless parameter 150
 IHaveValidatedThisSQL type 240, 242–243
 immutability, Scala language 375–376
 Implicits, Scala language 32, 382–383
 In parameter 242
 infix notation 379
 infix operation pattern 179
 Injector type 327, 332–333
 InRaw parameter 242
 INSERT procedure 247
 Instant Payment Notification. *See* IPN
 Integrated Development Environments. *See* IDEs
 IntelliJ IDE 384–386
 interfaces, Mapper. *See* Mapper interface
 InternalServerErrorResponse type 177
 IntField 264
 IPN (Instant Payment Notification) 99
 isAsLocale method 264

J

JAR file 21, 25–26, 164, 359
 Java language, properties of 290–291
 Java Persistence API. *See* JPA
 JavaScript language, abstractions 189–193
 basic commands 190–192
 JSON handling 192–193

JavaScript Object Notation. *See* JSON
 javax.http.servlet.HttpSession 128
 javax.servlet.Filter 161
 javax.servlet.http.HttpServletRequest 358
 JE object 192
 JEE (Java Enterprise Edition), integration 309–316
 jettyPort 29
 jetty-run 53
 JettyStartupAndTearDown 344
 JettyTestServer 337–338, 342, 344
 jlift.js file 201
 JNDI connection 227
 JndiEMF 314
 JPA (Java Persistence API) module 309–316
 JqJsCmds.FadeOut 79, 221
 JRE (Java Runtime Edition) 21
 JRebel 30
 JSArtifacts 189–190, 196, 221
 JsCmds object 72–73, 189, 191, 198
 JSESSIONID 353–354
 JsExp 189
 JSON (JavaScript Object Notation)
 creation 193, 276–277
 forms, using AJAX with 200–202
 handling 192–193
 JsonResponse type 177
 JsRaw 192
 JValue 193, 256
 JVM property 328

K

KeyedRecord 269–270

L

LAFuture type 305
 lazy loading, and parallel execution 124–125
 LazyLoad 125
 LazyLoggable 251–252, 361
 level attribute 145
 li attribute 144
 li_item attribute 144
 lib_managed directory 26
 lifecycle callbacks 247–248

lifecycles, HTTP
 application lifecycle 162–166
 request lifecycle 166–170
 Lift WebKit
 forms 131–136
 LiftScreen mechanism 132, 134–135
 Wizard system 135–136
 managing state 128–131
 cookies 130–131
 request and session 128–130
 snippets 114–125
 CSS transformers 114–120
 eager evaluation and attributes of 120–121
 lazy loading and parallel execution 124–125
 method
 dispatching 121–123
 stateless vs. stateful 123–124
 templates 108–114
 content embedding and partial 110
 displaying messages 113
 DocType declarations and markup validation 111–112
 head element resources 110–111
 HTML5 standard support 112–113
 surrounds 108–110
 views 125–128
 widgets 137–139
 AutoComplete 137–138
 Gravatar 138–139
 Lift Webkit 14
 <lift:bind /> element 34
 <lift:tail /> snippet 35
 LiftActor 80, 209–210, 216–217, 299, 369–370
 LiftFilter 360, 365
 LiftResponse 150, 170, 174–178, 366
 LiftRules 127, 171, 175
 LiftRules.addToPackages 79, 121, 127–128, 170
 LiftRules.appendGlobalFormBuilder 133–135
 LiftRules.attachResourceId 111, 366
 LiftRules.cometCreation 211
 LiftRules.DispatchPF 177

LiftRules.docType 111
 LiftRules.exceptionHandler 366
 LiftRules.httpAuthProtectedResource 168
 LiftRules.jsArtifacts 190, 196–197
 LiftRules.localeCalculator 28 4–286
 LiftRules.onBeginProcessing 168
 LiftRules.templateCache 110, 366
 LiftRules.unloadHooks 49, 166, 227, 267, 334, 368
 LiftScreen mechanism 132–135 and Wizard system 265–266
 custom field types 134–135
 Mapper interface integration with 250–251
 using AJAX with 202–203
 LiftScreen object 133
 LiftServlet 165
 LiftSession 128, 325, 359, 371
 LiftView class 127–128, 170
 LiftView type 127
 lift-wizard 135–136
 Lifty project 25
 Like parameter 242
 linkToSelf attribute 145
 Listings snippet 61–62
 loading spinner 196
 Loc class 153
 Loc implementation 153–158
 Loc.Link 155
 locale calculator 284–286
 Locale.getDefault method 285–286
 LocaleField 264
 LocalEMF class 314
 localization 282–292
 defining localized resources 289–291
 custom resource factories 291
 Java language properties 290–291
 XML 289–290
 implementing 283–289
 locale calculator 284–286
 localized code 288–289
 localized templates 286–287
 localSetup 214–215, 307–308
 localShutdown 214–215, 308

locations
 menus and 141–146
 parameters of 146–153
 authentication 150–153
 default 147–150

LocGroup 55, 65, 102, 150

LocParam class 55

LoggedIn 151

loggedInTest 73–74

logging queries 251–252

long polling 208

LongField 264

LongKeyedMapper 47–48,
 85–86, 229–230, 245–246

LongKeyedMetaMapper 85–86,
 229–230, 234, 248, 253, 341

LongMappedMapper 85–86,
 233–234, 236, 238, 241, 249

lowPriority 79–80, 210–211,
 307–308

M

makeFormsAjax 72, 119, 198,
 307–308

many-to-many relationship 232,
 234–235, 238, 246, 258

mapped fields, custom 254–258

MappedBirthYear type 231

MappedBoolean 57–58

MappedCountry type 231

MappedDate 230

MappedDateTime type 230

MappedEmail 47, 58, 230–231,
 246, 249

MappedEnum 85, 230–231, 249

MappedField type 48, 58, 231,
 236, 241, 255–256, 261

MappedForeign type 233

MappedForeignKey 233

MappedGender type 231

MappedLocale type 231

MappedLongForeignKey 88

MappedManyToMany 234–235,
 258

MappedOneToMany 47–49, 85,
 233, 241, 245

MappedPassword type 231

MappedPostalCode 58, 93, 232

MappedPostalCode type 232

MappedSplitString 255

MappedString 47–48, 57–58,
 229–231, 246–247

MappedText 47, 57, 229–230

MappedTimeZone type 232

Mapper class 48, 53, 56–57, 64,
 86

Mapper instance,
 defining 227–232

Mapper interface 223–258
 advanced 251–258
 custom mapped
 fields 254–258
 query logging 251–252
 transactions 252–254
 interacting with 237–251
 data 237–238
 display
 functionality 248–251
 lifecycle callbacks 247–248
 validation 246–247
 setting up database 224–237
 defining Mapper
 instance 227–232
 installation and
 connectivity 224–227
 relationships 232–235
 schema creation and
 control 235–237

Mapper models 93, 116

Mapper, testing with 340–341

MapperPaginatorSnippet
 class 63–64

MapperRules 236

MapReduce interface 274

markup validation, DocType
 declarations and 111–112

MaxRows parameter 242

mediumPriority 210, 219

Memoization object 286

Menu object 288

menus, and locations 141–146

message broker 295

messageHandler 77, 209–211,
 216–217, 299, 370

messages, displaying 113

messaging
 with Akka toolkit 300–309
 fault tolerance through
 supervisors 301–304
 Lift project interaction
 with 304–307
 with AMQP module 295–300
 example of 297–300
 support for 295–297

Meta object 229

MetaMapper 228, 234

MetaMegaProtoUser 51, 73, 86

MetaProtoExtendedSession 357

MetaProtoStateMachine 87–88

MetaPublisher 229

MetaRecord[T] 261

methods
 dispatching 121–123
 Scala language 379

Metric type 367

metrics 370–371

Microsoft SQL Server
 database 267

Model.createNamedQuery
 method 315

Model.mergeAndFlush(author)
 315–316

modeling 182

models 53

Model-View-Controller 6

MongoDB data store 276–280

MongoJsonObjectListField 279

MongoMetaRecord 277–279

MongoRecord 277–279

monitoring 367–371
 gauges 369–370
 metrics 370–371

multiple servers
 configuration 363–364

MySQL database 227, 267

N

net.liftweb.http.Bootable 164

net.liftweb.http.Generic-
 Validator 112

NGINX software 354–356

NodeResponse 176

nonrelational systems 260

NoSQL database 276

NoSQL movement 260

NoSQL stores, Record system
 for 273–280
 bookstore with MongoDB
 data store 278–280
 support for 273–278

NotByRef type 240

NotFoundResponse type 177

notification utility method 71

Novell Vibe environment, case
 study 372–373

O

object keyword 377

object-oriented. *See* OO

OkResponse type 177

onComplete 306–307

OneToMany 47–48, 85, 93, 233, 238, 258, 311
 one-to-many
 relationship 232–233
 Option type 388–391
 OptionalStringField 263
 OptionalTypedField 281
 Oracle database 267
 Order class 84–85
 order creation 84–91
 attributing auctions to customers 87–91
 data models for 84–87
 Order model 93–94
 OrderAuction class 49, 84–86, 90, 92
 OrderBy 47, 49, 61–63, 154–155, 241–242
 OrderBySql parameter 242
 OrderStatus 85–86, 90, 100–101
 OrderSummary 95–96, 102
 ORM (object-relational mapping) 39, 224
 orm.xml file 312
 outer_tag attribute 145
 OwnedField 263

P

page flow, for auction
 application 40–43
 administration 41–43
 auction detail and bidding 41
 checkout 41
 global elements 40
 home page 40
 parallel execution, lazy loading
 and 124–125
 param method 285
 parameters, of
 locations 146–153
 authentication
 parameters 150–153
 default parameters 147–150
 ParsePath method 66, 154, 171–174
 partial templates, content
 embedding and 110
 partialUpdate 79, 81, 210–211, 219–220, 307–308
 participatingIn 79–80
 PasswordField 264
 pattern matching, Scala
 language 382
 Payment Data Transfer. *See* PDT

PayPal integration 99–102
 PayPal payments 99–102
 Buy Now button for 102
 environment setup 99–102
 PayPal sandbox 101
 PDD (property-driven development) 318
 PDT (Payment Data Transfer) 99
 PermRedirectResponse
 type 177
 persistence, with Mapper interface. *See* Mapper interface
 persistence, with Record system.
 See Record system
 persistence.xml file 312
 PlainTextResponse type 177
 PostalCodeField 264
 PostgreSQL database 226
 postStop function 304
 PreCache parameter 243
 predicate function 381
 processing chain 167
 programming, distributed. *See*
 distributed messaging
 project directory, in SBT 27–28
 Project.scala file 25
 Prop object 322
 property-driven development.
 See PDD
 Props object, and run
 modes 365
 Props.RunModes 328, 365–366
 ProtoExtendedSession 357
 ProtoStateMachine 50, 82, 87–89
 prototype traits,
 scaffolding 50–53
 ProtoUser 50–53, 55, 87, 356–357
 provided scope 28
 push data 304

Q

queries
 logging 251–252
 of data 238–244
 Query parameter 241–242
 querying
 statements 271–272
 with Squeryl project 266–268
 QueryParam 49, 241–242
 queues 296

R

RabbitMQ 295
 RDBMS (relational database management system) 223
 Record class 263, 275
 Record instance 261
 Record system 259
 for NoSQL stores 273–280
 bookstore with MongoDB
 data store 278–280
 support for 273–278
 for relational
 databases 266–272
 functionality 260–266
 common Record system
 fields 262–265
 integration with LiftScreen
 mechanism and Wizard
 system 265–266
 RedirectResponse 147, 150–151, 366
 registerInjection 332
 relational database management system. *See* RDBMS
 relational databases, Record system for 266–272
 relationships 232–235
 many-to-many 234–235
 one-to-many 233
 reload command 26, 387
 rendering, menus 143–146
 replaceWith 116, 210, 219
 Representational State Transfer.
 See REST
 Req instance 179
 request lifecycle 166–170
 early functions 167–168
 template resolution 170
 validation and
 execution 168–169
 RequestMemoize type 285–286
 requests, and session
 state 128–130
 RequestVar object 128–129
 RequestVarEM 314
 reRender 215, 219–220
 resource factories, custom 291
 Resource Orientated Architecture. *See* ROA
 ResourceBundle 283, 291
 resources, localized 289–291
 ResourceServer object 163–164
 response matchers 339
 Response type 177

REST (Representational State Transfer) service
 advancedmultiformat 182
 basic 179–182
 RestHelper 176, 178–180, 185–186, 336
 RewritePF 171–173
 RewritePF partial
 function 171–173
 RewriteRequest object 66
 RewriteRequest type 172
 RewriteResponse object 66, 154, 171–174
 rewriting URLs 170–174
 advanced rewriting 173–174
 defining RewritePF partial
 function 171–173
 right-to-left. *See* RTL
 ROA (Resource Orientated Architecture) 175
 rock-paper-scissors
 game 212–216, 219–220
 RTL (right-to-left) 286
 Run command 220
 run modes 165, 365
 runtime scope 28

S

S.[?] method 288
 S.addAround(DB.buildLoan-Wrapper) 49, 267
 S.attr method 121
 S.deleteCookie(cookieName) 130–131
 S.error 71–72, 95, 113, 250, 315
 S.locale 286, 289
 S.notice 71–72, 113, 132–133, 136, 202, 250
 S.param method 68, 147, 172–173, 325, 390
 S.runTemplate 220, 335
 S.warning 71–72, 113
 saveMe method, Mapper 238
 SBT (Simple Build Tool) 20
 booting application 29–30
 creating project 23–26
 overview 21–23
 project directory 27–28
 src directory 28–29
 using snippets in 31–33
 sbt command 22–23
 SBT console shell 209

SBT downloads page 21
 SBT prompts 24
 scaffolding 50–57
 CRUD generation 53–57
 prototype traits 50–53
 Scala class 31, 123, 137, 210, 229, 311
 Scala Code Coverage Tool. *See* SCCT
 Scala JPA project 309–316
 Scala language 375–383
 classes 376–378
 collections 380
 functions 380
 implicits 382–383
 methods 379
 pattern matching 382
 traits 378
 variables, values, and immutability 375–376
 Scala Specs, frameworks for
 testing 319–321
 ScalaCheck, frameworks for
 testing 321–322
 ScalaTest, frameworks for
 testing 318–319
 scaling, deployment and. *See* deployment
 SCCT (Scala Code Coverage Tool) 323
 Schedule utility 80
 schema definition, for data
 models 46–49
 schemas, creation and control
 of 235–237
 Schemifier object 46, 49, 235, 237, 258, 340–341
 Screen type 136
 ScreenVar 94–95, 250, 265
 screenWrap, Mapper 52–53
 Script class 190
 security models,
 external 152–153
 Selenium 343–344
 self.reply method 306, 308
 server-push style
 programming 188
 servers
 multiple servers
 configuration 363–364
 single server
 configuration 362–363
 servlet containers 348–351
 ServletRequest 162
 session state, requests
 and 128–130
 SessionMaster 369–371
 SessionMonitor 369–370
 sessions, extended 356–358
 SessionVar 128–131, 358–359, 361–362
 SessionWatcherInfo 369–370
 SetHtml method 197–200, 210–211, 306–308
 shared state 158
 SHtml object 72, 114–115, 119, 156, 166, 195, 200
 SHtml.ajaxEditable
 control 195–196
 SHtml.ajaxSubmit 72, 198–199, 306, 308
 SHtml.ajaxText 205
 SHtml.button 220
 SHtml.password 134
 SHtml.select 134–135, 306
 SHtml.text 71–72, 115, 129, 196, 198, 215, 315
 shutdown process 165–166
 Simple Build Tool. *See* SBT
 SiteMap feature
 customising 153–159
 Loc implementation
 153–158
 use cases for 158–159
 locations
 menus and 141–146
 parameters of 146–153
 snippets 114–125
 CSS transformers 114–120
 eager evaluation and
 attributes of 120–121
 lazy loading and parallel
 execution 124–125
 method dispatching
 121–123
 stateless vs. stateful
 123–124
 splash.html 44
 SQL statement 223, 235, 237, 240, 242, 244, 266
 Squeryl project
 bookstore with 268–272
 CRUD statements 270–271
 querying statements
 271–272
 connecting and querying
 with 266–268
 StandardDBVendor 50, 225–226

StartAt parameter 242
 startTime 371
 state 351–361
 distributing critical 356–361
 extended sessions 356–358
 Terracotta Web
 Sessions 358–361
 managing 128–131
 cookies 130–131
 request and session
 state 128–130
 sticky session
 strategies 352–356
 HAProxy software 352–354
 NGINX software 354–356
 state machine 87
 stateful components,
 testing 325–327
 stateful processes 166
 stateful snippets, stateless snippets vs. 123–124
 StatefulSnippet 67–68, 71,
 123–124, 129
 stateless processes 166
 stateless snippets, vs. stateful
 snippets 123–124
 Static file 350–351
 Stats.addMetric method 371
 sticky session strategies 352–356
 HAProxy software 352–354
 NGINX software 354–356
 stores, NoSQL. *See* NoSQL stores
 StringAMQPsender 298
 StringField 261, 263–264,
 269–270, 275, 277–278
 structure, of templates 43–46
 Submit button 72, 115, 307
 supervisors, fault tolerance
 through 301–304
 surround method 287
 surrounds 108–110
 symbols 147
 System.setProperty 365

T

table method 268
 TDD (test-driven
 development) 318
 Team 10
 template caching 365–366
 Template parameter 149
 template surround 33
 TemplateFinder 149, 170

templates 43–46, 108–114
 content embedding and
 partial 110
 design workflow for 43
 displaying messages 113
 DocType declarations and
 markup validation 111–112
 head element
 resources 110–111
 HTML5 standard
 support 112–113
 localized 286–287
 overview 33–35
 resolution of 170
 structure of 43–46
 surrounds 108–110
 Terracotta Web
 Sessions 358–361
 Test location
 parameter 150–151
 test.default.props 340
 Testaccess location
 parameter 150–151
 TestAccess parameter 150–151
 TestCond 73, 91, 93
 test-driven development. *See*
 TDD
 testing 317–346
 AJAX 341–346
 and dependency
 injection 327–334
 and Factory type 332–334
 and Injector type 332–334
 Cake pattern 329–331
 using functions 331–332
 Comet 341–346
 frameworks for 318–323
 and code coverage
 reports 322–323
 Scala Specs 319–321
 ScalaCheck 321–322
 ScalaTest 318–319
 snippets 334–336
 stateful components 325–327
 web services 336–340
 with Mapper 340–341
 TestKit 317–318, 325–326,
 336–340
 TextareaField 265
 TimeSpan 79–80, 90
 TimeZoneField 265
 Title parameter 149
 toForm 134, 155, 249–250, 256,
 265–266

top attribute 145
 traits, Scala language 378
 transactions, Mapper interface
 support for 252–254
 transformation modifiers 117
 Transformers, CSS 114–120
 TransientRequestVar 358
 try/catch statements 316
 TypedField 263

U

UI control 134
 unapply functionality 171,
 173
 UnauthorizedResponse
 type 177
 Uniform Resource Locators. *See*
 URLs
 Unless class 147–148
 update command 26
 UPDATE procedure 247
 Update state 247
 updatedAt 48, 229, 236–237
 updateOrder 100–101
 URL parameter 146, 172
 URLs (Uniform Resource
 Locators)
 for auction details page
 66–68
 rewriting 170–174
 advanced 173–174
 defining RewritePF partial
 function 171–173
 User type 357
 userRoles 152
 util package 328, 332

V

val keyword 376–377
 validation 57–59, 246–247,
 262
 and execution 168–169
 definitions 58
 displaying field errors
 58–59
 markup, DocType declara-
 tions and 111–112
 Validation state 247
 validSelectValues 57, 249
 valMinLength 246
 ValueCell 204–206

values, Scala language 375–376
VARCHAR type 256
variables, Scala language
 375–376
Vibe environment,
 Novell 372–373
View First 6, 8
view-affecting parameter 148
viewDispatch 127
views 125–128
virtual host 296

W

WAR (Web Application
 Archives) 27
WAR file 289, 319, 347
Web Application Archives. *See*
 WAR

web services
 HTTP dispatching and
 174–185
 DSL 176–179
 REST service 179–185
 testing 336–340
web.xml file 27
webapp directory 164
webapp/templates-hidden
 directory 133
WEB-INF directory 27
WebKit, Lift. *See* Lift WebKit
WebSpec 326, 334–335, 346
where() method 272
wide.html 44
widgets 137–139
 AutoComplete 137–138
 Gravatar 138–139
wiki 158
wiki.html template 157

WikiEntry class 153–156
wildcard syntax 142
wiring 203–207
WiringUI object 204–206
with-resource-id 111, 366
withSFor method 335
Wizard system, LiftScreen
 mechanism and 135–136,
 265–266

X

XHTML templates 43
XML (Extensible Markup
 Language) 289–290
XmlGet type 178, 180–181,
 185
XmlResponse type 177