

Symbols

<head> 179
<property> 306
@Override 14

Numerics

404 error code 342

A

a href 113
AbortWithWebErrorCode-
Exception 342
AbstractAjaxBehavior 245, 250,
260
AbstractDefaultAjaxBehavior
245, 255
AbstractReadOnlyModel 249
AbstractTree 248
AbstractValidator 165
onValidate() 165
resourceKey() 166
access-denied page 342
action 280
form tag 142
active session 347
addComponent() 244, 253
addComponentInstantiation-
Listener() 276
advise 316
Ajax 19, 239–247
advantages of using behav-
iors 258

behaviors 245–247
components 243–245
debug window 242
Debugger 338, 340
disadvantages 241
DOM id attribute 244
engine 242–243
engine, third-party 258–261
example 240
focus form component 160
gotchas 262–263
in-place editing 251
JavaScript event handlers
245
request/response cycle 240
requirements for repainting
components 244
sending request asynchro-
nously 122
support in Wicket 242
swapping 195
updating components 253
URLs 335
visibility 132, 177
AjaxCheckBox 156, 243
AjaxEditableLabel 243, 254
AjaxEventBehavior 255
AjaxFallbackLink 19, 177, 325
Ajax and non-Ajax requests
19
onClick() 20
usage 123
vs. AjaxLink 122
AjaxFormComponentUpdating-
Behavior 246
AjaxFormSubmitBehavior 246

AjaxFormValidatingBehavior
246
AjaxLink 243–244
submit form 159
vs. AjaxFallbackLink 122
AjaxRequestTarget 20, 123, 244,
253, 256
focusComponent() 160
AjaxSelfUpdatingTimer-
Behavior 246
AjaxSubmitButton 243, 246
AjaxSubmitLink 159, 243, 246
alternative feedback message
167
annotatedClasses 315
annotation 279, 309
processor 310
anonymous
inner class 14
subclass 18
Ant 45
AOP 316
Apache
Struts 5
Velocity 12–13
API 27, 200
appendJavascript() 245, 256
application 25–26, 270
configuring for production
336–344
feedback override 166
home page 272
init() 49, 333, 339
monitoring 344–349
mountBookmarkablePage()
333

application (*continued*)
 multithreading 46
 newConverterLocator()
 297
 scope 229
 testing 322–330
 Wicket, layering with Spring
 304–313
 Application Programmer Inter-
 face. *See* API
 applicationContext.xml 311
 applicationContextAware 312
 architectural pattern 300
 ASCII 290
 Aspect Oriented Programming.
See AOP
 assertComponentOnAjax-
 Response() 325
 assertContains() 324
 assertErrorMessages() 328
 assertLabel() 322
 assertListView() 329
 assertModelValue() 324
 assertNoErrorMessage() 328
 assertNoInfoMessage() 328
 assertRenderedPage() 326
 associated markup 35–36
 Asynchronous JavaScript and
 XML. *See* Ajax
 attribute modifier 134
 AttributeModifier 43, 147
 usage 134
 authentication 269–274
 example 270–272
 authorization 132, 274–280
 actions 279
 strategy 274–276
 AutoCompleteTextField 244
 auto-component 286
 auto-detecting child compo-
 nents 190
 auto-link 119, 179, 226, 251

B

back button 10
 base page 187
 bean 306
 behavior 134
 listener 44
 binding to form components
 90–91
 blockquote 175
 bookmark, creating 195
 bookmarkability 193

bookmarkable
 page 117
 session-relative URL 335
 URL 331
 vs. session-relative requests
 330–332
 BookmarkableLink
 example 272
 BookmarkablePageLink 115
 example 116
 BookmarkablePageRequest-
 TargetUrlCodingStrategy
 334
 boolean form component
 155
 bootstrapping Spring 310
 border 171
 bot 232
 browser, caching 229
 BrowserInfoPage 261
 browsing history 27
 buffer, redirect to 143
 Burbeck, Steve 39
 business
 layer 301, 305
 logic 300
 tier 300
 button
 back 10
 forward 10
 internationalization 157
 internationalizing text 157
 markup 22, 68
 onSubmit() 23
 radio 151
 setDefaultFormProcessing()
 161
 usage 70
 bypassing form processing 161

C

cache 101
 Captcha 232–236
 Carnegie Mellon 232
 cascading style sheets. *See* CSS
 central feedback message 166
 chain of models 102
 change
 locale 324, 328
 requirements 5
 submit method 142
 tag 134
 CheckBox 155
 CheckBoxMultipleChoice 153

cheese store
 adding the address form 67–
 72
 adding the buttons 70
 adding the Check Out but-
 ton 62
 adding the shopping cart 58–
 62, 64
 adding the text fields 69–70
 class diagram 47
 creating a common layout
 51–53
 creating a reusable shopping
 cart 74–78
 creating the checkout page
 66–78
 creating the home page 53–
 66
 designing the UI 51
 displaying feedback 73
 displaying the cheeses 54–58
 hiding the Check Out button
 63
 implementing the applica-
 tion 48–49
 implementing the base page
 50–51
 implementing the session 49–
 50
 introduction 46–53
 making the form fields
 required 72–73
 setting up shop 46
 choice in a list 150
 ChoiceRenderer 154, 285
 cleaning up markup 136–137
 clickLink 325, 329
 cluster 88, 97
 code
 duplication, avoiding 200,
 278
 hard-coded 303
 maintainability 83
 collection, large 130
 Column annotation 315
 comma-separated values. *See* CSV
 common layout 183
 Compagner, Johan 5
 Component 33
 error() 170
 getModelObject() 89
 getModelObjectAsString() 89
 getString() 170
 info() 170
 isVisible() 63, 131

- Component (*continued*)
 - onComponentTag() 133
 - setOutputMarkupId() 21, 124
 - setOutputMarkupPlaceHolderTag() 177
 - setOutputMarkupPlaceHolderTag() 132
 - setRedirect() 142
 - setRenderBodyOnly() 136
 - setResponsePage() 63
 - setVisible() 131, 176
 - warn() 170
- component
 - common tasks with 130–137
 - compound 199, 204–206
 - compound input 209–213
 - compound, synchronizing models 211
 - configuring 203
 - creating custom, with factory methods 210
 - creating your own 74
 - custom 177
 - date-time 208–213
 - definition 107–109
 - domain-specific 213
 - examples 32
 - extending with behaviors 42–44
 - generic 199
 - grouping 174, 177–183
 - grouping using a Fragment 182–183
 - grouping using a Panel 178–182
 - grouping using a WebMarkupContainer 175–178
 - hiding 63, 176, 278, 280
 - hiding/showing 131
 - hierarchy 174
 - identifier 16, 94, 323
 - identifier in markup and Java 16
 - identifier names 92
 - instantiation listener 275, 310
 - introduction 31–44
 - Java 33–34
 - list of 106
 - and markup 35
 - nested 39, 93, 174, 208
 - no grouping 174
 - page 34–35
 - path 35, 323, 329
 - previewing 206
 - programmatically disabling 278
 - replacing 181, 261–262
 - gotchas 215
 - requiring parameters 11
 - reusable 177, 199, 234–236, 258–262
 - reusable component 74
 - reusable custom 201–202
 - self-contained 224–226, 259
 - setEscapeModelStrings() 112
 - showing 63, 131
 - that selects current locate 200–208
 - triad 32–33, 108
 - use check 337
 - visibility 132
 - Wicket identifier vs. DOM identifier 19
 - wrapping around child markup 189–191
- component hierarchy 109, 176, 187
 - matching with markup structure 37–38
 - modifying 181
- component structure, navigating with compound Wicket identifiers 323
- ComponentFeedbackMessageFilter 171
- ComponentTag 206
- compound identifiers 323
- CompoundPropertyModel 92–96, 110
 - description 84
 - example 217
 - usage 93
- CompoundPropertyModel:bind 95
- compression 338–339
- concrete implementation 302
- conditional markup 63, 131
 - using Ajax 177
- configuration 143, 337
 - of container 302
 - custom error pages 341–344
 - for deployment mode 338
 - for development mode 337
 - default, modifying 339–341
 - development vs. deployment mode 336–341
 - parameters 337
- confirm 135
- link 134
- consistent layout
 - best way 193–195
 - bookmarkability 195
 - duplication 194
 - markup inheritance 187–191
 - navigation 195
 - panels 191–193
 - plain pages 184–187
 - previewability 193–194
- container 137, 294
- ContainerFeedbackMessageFilter 171
- context locator 308
- continueToOriginalDestination 272
- controller description 6
- conversion over configuration 38
- conversation defined 25
- conversion 110
 - customizing 295–298
 - localized 293–298
 - mechanism 83
 - overriding for an application 297–298
 - overriding for one component 296–297
 - using converters 294
 - using models 293
- converter 221, 293–295
 - configuring 283
 - custom 295–298
 - locator 297
 - resource key 163
- ConverterLocator 298
- converting 61
 - input 144
- convertInput() 211
- convertToObject() 296
- convertToString() 295–296
- cookies, storing values in 148
- copy/paste programming 185, 194
 - preventing 177
- covariance 270
- cross-site scripting. *See* XSS
- cross-site-request forgery 331
- CryptedUriWebRequestCodingStrategy 268, 331
- CSS 238, 250
 - declaring dependencies 247
 - file location 52
 - use in cheese store 51
- CSV 224, 226

custom
 application 328
 component 177
 request cycle 343
 resource loading 291–293
 resource locator 292
 session 328
 validator 165

D

DAO 26, 99, 300
 implementing with Hibernate 316–317
 data access layer 305, 313
 data access object. *See* DAO
 data tier, implementing with Hibernate 313–319
 database table, mapping domain class to 314
 database, abstracting details with Hibernate 314
 Databinder 320
 DataTable 130
 DataView 130
 date 293
 date-time component 208–213
 declarative programming model 309
 decode request 30
 default feedback message 167
 dependency injection. *See* DI
 deployment descriptor 337
 deployment mode 336
 configuration 338
 design pattern, template method 255
 designer versus programmer 53
 detachable model 318
 detaching 96
 determine request target 30
 development mode 336
 configuration 337
 DHTML 238
 DI 301, 303–304
 code without 302–303
 implementation 304
 open source frameworks 303
 Spring support for 305
 Direct Web Remoting. *See* DWR
 display text 16
 doctype declaration 291
 document object model. *See* DOM
 Dojo 21, 242, 258

DOM 19, 239, 244
 domain class, defining 314
 domain model 300
 domain object 305
 loading 312
 DomainObject 313
 domain-specific language. *See* DSL
 Don't Repeat Yourself. *See* DRY
 DropDownChoice 151
 example 202–204
 DRY 39, 64, 178
 DSL 32, 38
 duplication 193
 reducing 183
 duration of request 345
 Dutch 287
 DWR 242
 dynamic model 61
 vs. static 86–88
 dynamic text 109
 DynamicImageResource
 example 233
 DynamicWebResource 230

E

Echo 5
 Eclipse 56, 74, 225
 effect, Web 2.0 124
 Ehcache 349
 EJB container 302
 email address validator 164
 emailing URLs to Wicket page 115
 empty fields, preventing 144
 empty input 144
 EmptyPanel 181
 encapsulation 178, 181
 enclosure 132
 encoding parameters 116
 encryption key
 default 331
 Entity annotation 315
 EqualPasswordInputValidator 164
 equals 219, 319
 error message, registering 165
 error page
 custom 341–344
 for specific errors 343
 error() 169
 ErrorLevelFeedbackMessageFilter 171
 escape model strings 112
 escaped unicode 286
 escaping 116
 event listener 71
 example
 adding to shopping cart using Ajax 123
 advanced search check box 155
 Ajax behavior 247
 Ajax comment form 159
 Ajax component 246
 AjaxFallbackLink 19
 BookmarkableLink 272
 BookmarkablePageLink 116
 changing style with AttributeModifier 134
 clock, dynamic 87
 clock, static 86
 CompoundPropertyModel 93
 confirmation link 134
 counting Ajax link 19–21
 counting link 18–19
 custom request target 29
 custom session 27
 custom validator DivisibleValidator 165
 date format label 202
 detachable cheese model 98
 echo application 21–23
 generating a menu with ListView 128
 generating a menu with RepeatingView 126
 get all cheeses with LoadableDetachabeModel 100
 having fun with links 17–21
 Hello World 15–17
 Label 36
 Link 201, 279
 loadable cheese model 101
 locale drop-down menu 202–204
 locale Reset link 207–208
 menu 185
 Model workaround for serialization 89
 modifying attributes in onComponentTag 133
 nesting a LoadableCheeseModel in a PropertyModel 103
 nesting PropertyModels 104
 omelette du fromage 174, 180
 Page 270
 PageParameters 117, 272

- example (*continued*)
 - PasswordTextField 270
 - PropertyModel 273
 - protecting pages 276
 - refreshing a ListView with
 - LoadableDetachableModel 128
 - RestartResponseAtInterceptPageException 272
 - reusable required text field 202
 - reusable ShoppingCartPanel 75
 - StatelessForm 270
 - TextField 201, 270
 - URL converter 296
 - using Ajax component 244
 - example markup 55, 194
 - exception
 - aborting, class hierarchy 277
 - cause of 343
 - logging 88, 98
 - showing in stack trace 338
 - uncaught 343
 - exploit 113
 - external link 115
 - ExternalLink
 - usage 114
- F**
-
- factory method, creating custom components 254, 256
 - failed validation 165
 - failover 88
 - favorite links 115
 - FKEditor 149
 - feedback filtering 171
 - feedback message 73
 - alternative 167
 - central 166
 - default 167
 - standard 166, 169
 - substituting a variable 166
 - FeedbackPanel
 - example 252
 - markup 73, 171
 - usage 73
 - Fielding, Roy T. 9
 - file location 15, 55
 - filter 317
 - configuring 315
 - filtering feedback message 171
 - final classes and methods 259
 - Firebug 243
 - Firefox 243
 - first-level cache 315
 - flash message 169
 - Flickr 239
 - focus form component 160
 - focusComponent() 160, 256
 - forCss() 248
 - forJavaScript() 248
 - form
 - binding 90–91
 - check box for boolean properties 155–157
 - client-side form processing 141–143
 - components for submitting forms 157–162
 - components for text input 145–149
 - converting user input 163
 - custom validators 165
 - displaying messages using a FeedbackPanel 170–171
 - form processing 141–145
 - getMethod() 142
 - input processing 23
 - mapping an object to a choice using ChoiceRenderer 153–154
 - markup 22, 68
 - multiline text with TextArea 148–149
 - onError() 141, 145
 - onSubmit() 141, 145
 - onSubmit() vs. button onSubmit() 158
 - processing 144
 - protecting text with PasswordTextField 147–148
 - providing feedback 166–171
 - providing non-validation-related messages 169–170
 - required validation 162–163
 - selecting a single value 149–152
 - selecting from a list of items 149–157
 - selecting multiple values 152–153
 - server-side form processing 144–145
 - server-side processing 69
 - single-line text with TextField 146–147
 - skip processing 70
 - standard feedback messages 166, 169
 - standard validators 163–164
 - submit form with buttons 157–158
 - submitting form with Ajax links 159–161
 - submitting form with links 158–159
 - submitting form without processing 161–162
 - submitting using get 142
 - submitting using post 142
 - submitting using redirect after post 143
 - usage 69, 140
 - validating user input 162–166
 - what are forms? 140–141
 - form component
 - AjaxSubmitLink 159
 - Button 157
 - CheckBox 155
 - CheckBoxMultipleChoice 153
 - CheckGroup 153
 - DropDownChoice 151
 - embedding 209–211
 - ListChoice 150
 - ListMultipleChoice 152
 - PasswordTextField 147
 - RadioChoice 151
 - SubmitLink 158
 - TextArea 148
 - TextField 146
 - form fields, setRequired() 72
 - form processing 209, 212
 - bypassing 161
 - formatted text 111–112
 - FormComponent
 - setPersistent() 148
 - setRequired() 162
 - FormComponentFeedbackBorder 171
 - FormComponentPanel 209
 - FormTester
 - setValue() 326
 - submit() 326
 - testing forms automatically 326
 - forward button 10
 - Fowler, Martin 301
 - Fragment 182–183
 - creating 182
 - markup 182
 - tags 182
 - usage 182

Frame 34
functional programming 14, 309

G

Garfield 96
Garret, Jesse James 239
general-purpose language 38
GeneratedValue annotation 315
generating markup id 21
generic model 104
get form submission 142
getCallbackUrl() 245
getChainedModel() 103
getConverter() 202, 221, 296
getConverterMethod() 253
getCryptFactory() 331
getItemModels() 217
getListItemModel() 130
getModelObject() 42
getModelObjectAsString() 295
getObject() 41–42
getParent() 218
getResourceStream() 227
getResponse() 250
getSessionInfo() 346
getString() 170
getTagById() 324
getting active session 347
getting input from users 140
Google Maps 239
Google Suggest 239, 244
Grails 320
grouping components 174–183
 no grouping 174
 using a Fragment 182–183
 using a Panel 178–182
 using a WebMarkupContainer 175–178
Guice 303
gzip 339

H

hard-coded 303
hashCode() 219, 319
head 180
header contributions 247–251
 behaviors 248
 performing with package resources 259
 using templates 249
HeaderContributor 248

Hello World 15
Hibernate 10, 39, 300
 abstracting database details 314
 configuring 314–316
 filters 317
 implementing DAOs 316–317
 implementing data tier with 313–319
 introducing 314
 MBeans 349
 pitfalls 317
 session 315
hiding
 component 63, 131
 part of page 131–132
 with Ajax 132
HiveMind 303
Hollywood Principle 301
HTML
 change tag 134
 disabled input fields 279
 file location 15, 55
 form 140
 img tag 224
 in model 112
 select tag 206
HtmlUnit 322
HTTP log 344
HTTP session 88
HTTP status code 342
HttpSession 10, 27, 49
HttpSessionStore 340
HybridUrlCodingStrategy 335

I

i18n 17, 282–283
IApplicationSettings 340
IAuthorizationStrategy 274, 277
IBehavior 258
IBehaviorListener 44, 245, 258, 260
IChainingModel 103
IChoiceRenderer 154
IComponentInstantiationListener 276
IConverter 202, 294
IConverterLocator 298
Id annotation 315
id attribute 19
IDE 5, 32
 configuring 225
IDebugSettings 340
identifier 108
IDetachable 41, 97
IDetachable:detach 97
IExceptionSettings 340
IFrameworkSettings 340
IHeaderContributor 245, 249, 258
IHeaderResponse 249
Initializer 230
ILinkListener 30
Image
 example 225, 233
 setImageResource() 233
IMarkupSettings 340
IModel 41, 83, 134, 217
 generic models 104
 setObject() 145, 207
IModel:detach() 97
IModel:getObject() 83
IModel:setObject() 83
impedance mismatch 5, 10
imperative programming 14
implementation details
 hiding 202
import 15
IndexedHybridUrlCodingStrategy 335
IndexedParamUrlCodingStrategy 334
indirection 33
 using in custom components 255
info() 169
inheritable model 93, 95
init 333, 339
initializer 230–231
injecting JavaScript 113
InjectorHolder 312
inline panel 182
input
 conversion 144
 converting 163
 empty 144
 getting from users 140
 required 144
 retaining and skipping processing 161
 validating 162–166
 validation 145
integrated development environment. *See* IDE
interception URL 272
interface 302
internal-error page 342
internationalization. *See* i18n

internationalizing button text 157
 Internet Explorer 262
 maximum URL length 117
 interpolating a variable 166
 invalidateNow 274
 inversion of control. *See* IoC
 InvocationTargetException 344
 IoC 301
 IPageSettings 340
 IRequestCycleProcessor 231
 IRequestCycleSettings 143, 340
 IRequestListener 71
 IRequestLoggerSettings 340
 IResourceFinder 293
 IResourceListener 228
 IResourceSettings 340
 IResourceStreamLocator 291
 isActionAuthorized() 279
 ISecuritySettings 331, 340
 ISessionLogInfo 346
 ISessionSettings 340
 isInstantiationAuthorized() 277
 ISO 8859-1 286
 isTransparentResolver() 189
 isValid() 213
 isVisible() 63, 131, 278
 IUnauthorizedComponent-InstantiationListener 277
 IValidatable 234
 IValidationError 166
 IValidator 74, 165
 IWebApplicationFactory 311

J

J2EE 39, 305
 JAAS 281
 Java 5 annotation 14
 Java Authentication and Authorization Service. *See* JAAS
 Java component 33–34
 Java Development Kit. *See* JDK
 Java Enterprise Edition. *See* J2EE
 Java Management Extensions. *See* JMX
 Java Persistence API. *See* JPA
 Java Virtual Machine. *See* JVM
 Javadoc 32
 JavaScript 238
 as Ajax enabler 239
 declaring dependencies 247
 injection 113
 stripping comments 338

JavaServer Faces. *See* JSF
 JavaServer Pages. *See* JSP
 JCapcha 232
 JConsole 347
 JDBC 300
 JDK 284
 Jetty 347
 jetty-management 348
 JMX 347–349
 enabling 347
 JNDI 302
 Johnson, Rod 305
 JPA 146
 JSF 5, 12, 39
 JSP 6, 12, 38
 request/response cycle 6
 JUnit 322
 just-Java 309
 JVM 5
 JWebUnit 322

K

key validation messages 167

L

110n 283
 Label 36
 as alternative to message tags 288
 constructor 85
 example 202
 introduction 32
 markup 18, 110, 175
 markup tags 59
 models 111
 usage 16
 label
 displaying formatted text
 using labels 112–113
 using the Label component 109–111
 using the MultiLineLabel component 111–112
 language
 general-purpose 38
 variant 289
 large collection 130
 lasagna code 81
 Latin 1 286
 layered architecture 300
 advantages 301
 layering 301

layout
 consistent, best way 193–195
 consistent, bookmarkability 195
 consistent, duplication 194
 consistent, with markup inheritance 187–191
 consistent, navigation 195
 consistent, previewability 193–194
 consistent, with panels 191–193
 consistent, with plain pages 184–187
 lazy collection 318
 lazy initialization 229
 LengthValidator 74
 lightweight container 302
 lines of code 83
 Link 120
 as a button 62
 example 201, 214, 279
 markup 18, 62
 onClick() 11, 18, 57, 60, 63, 86, 120
 skipping form processing 161
 submit form 158
 usage 120
 link
 click, redirect after 143
 from database 114
 download 228
 external 115
 to external site 114
 favorite 115
 navigating with 113–120
 navigation using
 BookmarkablePageLink 115–119
 navigation using static links 113–114
 navigation using wicket:link 119–120
 responding to client actions 120–124
 static 113
 testing 329
 using AjaxFallbackLink 122–124
 using ExternalLink 114–115
 using Link 120–122
 to Wicket page 115
 list
 choices 150
 working with 216–217
 working with, in forms 217–221

- ListChoice 150
 - setMaxRows() 151
 - listener 71
 - ListenerInterfaceRequest-Target 30
 - ListItem 56, 128
 - vs. WebMarkupContainer 128
 - ListMultipleChoice 152
 - ListView 55, 65
 - getItemModel 130
 - getting the selected item 57, 60
 - item with CompoundPropertyModel 129
 - large collections 130
 - limitations 262
 - markup 54
 - populateItem() 56, 60
 - refresh contents 100
 - usage 56, 128
 - using with database results 129
 - load 100
 - balancing 9, 88
 - LoadableDetachableModel 128, 273
 - description 84
 - load() 100
 - onAttach() 102
 - onDetach() 102
 - Lob annotation 315
 - Locale 200, 282
 - locale 324
 - selecting current 200–208
 - locale-dependent
 - part 290
 - text 284
 - localization 17, 85, 282
 - filename patterns 287
 - loading localized markup 289–291
 - looking up messages for components, algorithm 288–289
 - looking up messages for components, mechanism 286
 - multiple languages 284–291
 - resource bundles or loading localized markup 291
 - support for 283
 - Localization. *See* 110n
 - Localizer 283
 - localizing a message 170
 - log 336
 - log4j 346
 - example configuration 346
 - logging 88, 98, 345
 - logic
 - mixing in templates 13
 - separation from presentation 38–39
 - logical tier 300
 - loose coupling 301
- M**
-
- maintainability 83
 - maintenance nightmare 187
 - Managed Beans 347
 - management console 349
 - managing state
 - HTTP protocol 8–9
 - storing state in session 10
 - storing state in URLs 9–10
 - manipulating markup attributes 133–135
 - ManyToOne annotation 315
 - mapping object to choice 154
 - markup
 - cleaning up 136–137
 - conditional 63, 161, 177
 - file, localized 289–291
 - introduction 32
 - location 15, 55
 - loop 124
 - manipulating attributes 133–135
 - reloading 337
 - removing 136
 - tag, change 134
 - valid 137
 - markup hierarchy 187
 - modifying 181
 - markup inheritance 194
 - child 189
 - contributing to head 189
 - multiple layers 189
 - parent 188
 - MarkupContainer 218
 - isTransparentResolver() 189
 - master template 326
 - Matryoshka doll 102
 - Maven 45
 - maxlength 146
 - MBeans 347
 - memory 83
 - leak 307
 - usage 97, 101, 193
 - menu 185
 - message
 - displaying using a FeedbackPanel 170–171
 - localizing 170
 - metadata 207, 309
 - tagging with annotations 309
 - MetaDataKey 207
 - minifying 338
 - minimize
 - JavaScript 338
 - state 97
 - MixedParamUrlCodingStrategy 335
 - mixing logic in templates 13
 - mode 336
 - model 39–42
 - alternative property expressions 95–96
 - chain 102
 - CompoundPropertyModel 92–96
 - converting values 221
 - definition 82
 - description 84
 - detachable 96–102
 - doesn't update 60
 - generic 104
 - getObject() 61, 87
 - inheritable 93, 95
 - introduction 33
 - list of models 84
 - LoadableDetachableModel 100–102
 - nesting 102–104
 - PropertyModel 90–91
 - serializing 88
 - standard models 84–96
 - static vs. dynamic 61, 86–88
 - using detachable models to avoid serialization errors 98–100
 - using Model to avoid serialization errors 89–90
 - using the simple Model 84–90
 - using to avoid serialization errors 89–90
 - value 110
 - Model 2 5
 - difference from Wicket 5
 - framework 39, 268
 - request/response cycle 6
 - Model model 84
 - Model View Controller. *See* MVC
 - Modelibra 320
 - ModelIteratorAdapter 217

modifying tag attributes 147
 modularization 200
 monetary to String conversion
 61
 monitoring
 an application 344–349
 using JMX 347–349
 using request logger 344–
 347
 mountBookmarkablePage 333
 mounting 332
 MultiLineLabel, rendered out-
 put 111
 multiple-field validation 164
 multithreading 46
 MVC 39–40, 82, 108

N

nationality 293
 navigating using links 113–120
 navigation 193
 easy 195
 testing 325
 nested component 93, 174
 newChildId() 125
 newClientInfo() 261
 newConverterLocator() 297
 newline in text 111
 newSession() 270
 NotSerializableException 88
 null values 144
 NullPointerException 85
 number 293
 of sessions 345
 of visible rows 151
 NumberFormat 294
 NumberValidator 164

O

object orientation 187
 Object Relational Mapping. *See*
 ORM
 object-oriented design 26
 object-oriented programming.
See OOP
 Object-Relational Mapping. *See*
 ORM
 onAttach() 102
 onBeforeRender() 211
 onBlur() 257
 onClick() 120, 244
 onclick() 255

onComponentTag() 133, 175,
 206, 234, 257
 onComponentTagBody() 175,
 294
 onDetach() 102
 one pass render 143
 onError() 141, 145, 258
 onInstantiation() 276
 onPopulate() 127
 onRenderHead() 259
 onRequest() 260
 onResourceRequested() 228
 onRuntimeException() 343
 onSubmit() 141, 145, 219,
 258
 onValidate() 165
 OOP 10, 314
 ORM 10, 300, 314

P

package name 15
 package resource 224–226
 including with auto-linking
 226
 performing header contribu-
 tions 259
 PackageResource example 225
 Page 34–36, 75
 example 15, 270
 file location 15, 55
 introduction 32
 markup 16
 markup filename 15, 55,
 178
 partial update 19
 setHeaders 342
 page
 associated markup 35
 bookmarkable 117, 268
 hiding part of 131–132
 indexing 229
 layout 183–195
 map 28, 31
 navigation 215
 partial update 19
 protecting 274
 refreshing 86
 session-relative 268
 store 88
 Page Expired message 88
 PageableListView 65
 large collections 130
 page-expired error page 342
 PageMap 272
 PageParameters 117
 add() 117
 example 117, 272
 parsing 119
 PagingNavigator 64
 markup 65
 Panel 75
 empty panel 181
 example 204, 214
 inline panel 182
 markup 178
 markup filename 178
 replacing 181
 tags 181
 testing 329
 usage 179–180
 panel
 custom, testing 328–330
 swapping 191, 194
 using in a page 178
 parameter
 encoding 116
 parsing 119
 partial page update 19
 passing state 268
 password, reset 148
 PasswordTextField 147
 example 270
 setResetPassword() 148
 usage 148
 Path 293
 PatternValidator 164
 performance 83, 193
 implications of detaching 101
 monitoring 345
 persistence framework 313
 persistent object, loading 318
 PicoContainer 303
 Plain Old Java Object. *See* POJO
 POJO 39
 populateItem 217, 220
 popup list 151
 portlet 215
 post
 form submission 142
 redirect after 143
 POSTDATA popup, preventing
 142
 presentation
 layer 305
 separation from logic 38–39
 tier 300
 preview 55, 179
 previewability 193
 primary key 98, 315

process events 30
 production mode 336
 programmatically modifying tags 134
 programming
 functional 14
 imperative 14
 property
 expression 90
 file 166
 model 18
 PropertyListView 129
 PropertyModel 18, 90–91
 autocreate objects 90
 description 84
 example 90, 273
 usage 69
 Prototype 21
 proxy 310
 factory 308
 instead of direct reference 307
 object 317

Q

QueryStringUrlEncoding-
 Strategy 334
 Quickmodels 320

R

radio button 151
 RadioChoice 151
 setPrefix() 152
 setSuffix() 152
 RadioGroup 152
 redirection 142
 reducing duplication 183
 refactoring 38, 92, 95, 213
 not safe with wicket:link 120
 refreshing a page 86
 RefreshingView 319
 example 216, 252
 registering an error message 165
 regular expression 164
 relational model 314
 reloading
 markup 337
 resources 337
 remember me 50
 removing markup 136

render
 redirect to 143
 strategies 143
 renderHead() 249, 259
 renderString() 249
 repeater 216
 item-reuse strategies 219
 using ListView 128
 using RepeatingView 124–127
 repeating markup 54
 and components with repeaters 124–130
 RepeatingView
 example 124
 limitations 262
 newChildId 125
 onPopulate 127
 replace() 218
 replaceComponentTagBody() 295
 replaceWith() 218
 Representational State Transfer.
 See REST
 Request 28
 request 25
 cycle 231, 343
 listener 71
 page 345
 parameter 9, 119
 processing 25–31
 objects 25–31
 steps 29–31
 processing independently
 from pages 223
 request target, custom 231
 request/response cycle, traditional 239
 RequestCycle 28
 onRuntimeException() 343
 RequestCycleProcessor 28
 RequestLogger 344–347
 custom information 346
 enabling 346–347
 RequestLoggerSettings 346
 RequestParameters 29
 RequestTarget 28–29, 231
 required input 144
 requirement, changing 5
 Reset link, example 207–208
 reset password 148
 resolving component 190
 resource 223
 Captcha 232–236
 component-hosted 228
 dynamically generated 226–231
 lazy initialization 229
 not thread-safe 230
 packaging with components 247
 PDF, RTF, Excel, and so on 232–236
 reloading 337
 settings 340
 shared 229–230
 shared, initializing 230–231
 utilization 301
 resource bundle 17, 166, 286, 323
 support 283
 resource key 166–167
 converters 163
 IConverter 163
 IConverter.type 163
 Required 163
 resourceKey() 166
 ResourceLink 228
 resource-lookup mechanism 291
 ResourceModel 288
 description 84
 ResourceReference
 example 225
 ResourceStreamLocator 291
 ResourceStreamRequestTarget
 example 231
 respond 30
 Response 28
 response
 header setting 227
 page 345
 REST 9, 268
 RestartResponseAtIntercept-
 PageException 277
 example 272
 RESTful encoding 334
 retaining input and skipping
 processing 161
 reusable
 component 74, 177, 258–262
 software module 31
 reuse of template 194
 RFC-1738 116
 rich text editor 149
 Rich Text Format. *See* RTF
 row, visible, number of 151
 RTF 232
 runtime configuration 347

S

-
- scalability 97
 - screen, partial update 239
 - Scriptaculous 242, 258–259
 - script-injection attack 113
 - search engine 52
 - second-level cache 318
 - section 508 20
 - security 331
 - select tag 206
 - selected value 150
 - selection component 149
 - Selenium 322
 - separation
 - of concerns 38, 53
 - of layers 304
 - Serializable 88
 - serialization 88
 - problems with 98
 - service 305
 - layer 300
 - object 306
 - service locator pattern 302, 307, 309, 312
 - problem with 310
 - servlet
 - container 311
 - listener 310
 - Servlet API 10
 - web application directory 224
 - Session 26, 207, 269–270
 - creation 50
 - custom class 49
 - error() 169
 - factory method 50
 - info() 169
 - invalidating 273
 - Session.invalidate() 274
 - Session.invalidateNow() 274
 - setLocale() 324, 328
 - warn() 169
 - session 25
 - active 347
 - expiration 271
 - number of 345
 - per request 315
 - state 97
 - timeout 49
 - session-relative
 - page 268
 - request 330–331
 - URL 331
 - SessionStore 27
 - setAttachmentHeader() 227
 - setChainedModel() 103
 - setConvertedInput() 212
 - setCryptFactory() 331
 - setDefaultFormProcessing() 161
 - setHeaders() 227, 342
 - setItemReuseStrategy() 219
 - setLocale() 324, 328
 - setMaxRows() 151
 - setMetaData() 207
 - setObject() 41, 145
 - setOutputMarkupId() 124, 244, 253, 262
 - setOutputMarkupPlaceholderTag() 132, 177
 - setPersistent() 148
 - setPrefix() 152
 - setRedirect() 142
 - setRenderBodyOnly() 136
 - setRequestLoggerEnabled() 346
 - setRequestTarget() 231, 260
 - setRequired() 72, 162, 201
 - setResetPassword() 148
 - setResourcePollFrequency() 340
 - setResponsePage() 30, 63, 71, 120, 201
 - setSuffix() 152
 - setUpRequestAndResponse() 324
 - setVisible() 131, 176
 - ShoppingCartPanel 75
 - code 77
 - markup 76
 - usage 77
 - showing component 63, 131
 - SimpleAttributeModifier 43, 147
 - single-threaded programming
 - model 31
 - singleton 306
 - skipping form processing 70
 - Smalltalk 39
 - software module, reusable 31
 - spaghetti code 81
 - Spring 26, 39, 303
 - bootstrapping 310
 - framework 299, 305–306
 - layering Wicket applications 304–313
 - MVC 5
 - support for Hibernate 315
 - transaction annotations 316
 - Spring bean annotation 309–312
 - using with non-Wicket objects 312–313
 - SpringBean 309
 - annotation, example 236
 - SpringWebApplicationFactory 311
 - SQL
 - injection 147
 - shielding from, with Hibernate 314
 - stack trace 338, 341
 - standards compliance 137
 - Star Wars 3
 - startPage() 322
 - startPanel() 328
 - state
 - encoding in URLs 9–10
 - management 10
 - minimizing 97
 - passing 268
 - storing in component 18
 - storing in session 10
 - StatelessForm 271
 - example 270
 - static
 - link 113
 - model 61
 - model vs. dynamic 86–88
 - typing 11, 27, 38
 - storing values in cookies 148
 - strategy pattern 303
 - StringBuilder 249
 - StringHeaderContributor 248
 - StringRequestTarget 260
 - StringResourceModel
 - description 84
 - StringResourceStream 227
 - example 231
 - StringValidator 164
 - Stripes 5
 - stripping
 - comments 338
 - whitespace 338
 - Struts 39
 - stylesheet 180
 - submit button 140
 - SubmitLink 158
 - example 231
 - setDefaultFormProcessing() 161
 - substitution variable 166
 - Sun JCE encryption 331
 - swapping panels 191, 194

Swing 41
 switch locale 324, 328
 synchronization 97
 methods 308
 system property 337

T

tag attribute, modifying 147
 tag library 13
 TagTester 324
 Tapestry 5
 Teachscape 7
 template
 loading 291
 method design pattern 255
 reuse 194
 templating 214
 testing
 Ajax links 325
 applications 322–330
 custom panels 328–330
 DOM tags 324
 feedback messages 327
 forms 326–327
 framework 322
 label values 322–324
 link testing 324–326
 links 329
 list view 329
 navigation 325
 rendered page 326
 submit 326
 template 326
 using custom application 328
 using custom session 328
 using DI 303
 validators 327–328
 TestPanelSource 329
 text
 displaying with label components 109–113
 dynamic 109
 formatted 111–112
 locale-dependent 284
 newline in 111
 TextArea 148
 TextField
 example 201, 234, 270
 markup 22, 68
 maxlength 146
 setRequired() 72
 update value 23
 usage 69, 146

TextTemplateHeader-
 Contributor 248–249
 Thai 202, 287
 thread-safety 31
 rules 31
 three-tiered service architecture 300–304
 TinyMCE 149
 toggling visibility using a check box 155
 TopLink 10
 toString() 110
 transaction 307
 annotation 316
 demarcation 301
 transient reference 99
 translation 323
 transparent resolving 190
 type conversion 9

U

ugly URL 117
 uncaught exception 343
 unicode 284, 287
 escaped 286
 unit testing 322
 unmanaged framework 309
 updating part of page 19
 UploadStatusResource 230
 URL 114
 Ajax and bookmarking 335
 bookmarkable 331
 clean 332
 emailing to Wicket page 115
 encryption 331
 escaping 116
 maximum length 117
 optimization 330–336
 session relative 331
 stable 229
 ugly 117
 validator 164
 URL encoding 333
 examples 333
 hybrid 335
 indexed parameters 335
 key/value pair 334
 mixed indexed and query string 335
 provided by Wicket 333
 query string 334
 strategies 332–336
 use check 337

user
 authorization level 132
 feedback 342
 user input
 converting 163
 validating 162–166
 User-Agent request header 261
 UserPanel, localizing 284–285
 UTF-8 287, 290

V

valid markup 137
 validating two fields 164
 validation 72, 145, 212
 failed 165
 regular expression 164
 validator
 base class 165
 custom 165
 EmailAddressValidator 164
 EqualPasswordInputValidator 164
 finding all 74
 NumberValidator 164
 PatternValidator 164
 standard 163–164
 StringValidator 164
 testing 327–328
 UrlValidator 164
 value
 object 319
 selected 150
 variable substitution 166
 Velocity Tools 13
 versioning 215
 ViewRoot 34
 visibility 131
 change using Ajax 177
 toggling using a check box 155

W

warn() 169
 Web 2.0 239
 effect 124
 web application 7
 factory interface 311
 web designer, working with 13
 web MVC 5
 web.xml 310, 337
 webapp directory 311
 WebClientInfo 261

- WebMacro 13
- WebMarkupContainer 94, 126, 175
 - group for Ajax updates 180
 - onComponentTag() 175
 - onComponentTagBody() 175
 - vs. ListItem 128
- WebRequestCycle 262
- WebResource 227
- WebResponse 227
- WebSession 270
- websites vs. web applications 7–8
- WebWork 5
- Wicket
 - abort exceptions, class hierarchy 277
 - AbortWithWebErrorCodeException 342
 - Application 26
 - application settings 261
 - authentication 269–274
 - authentication, example 270–272
 - authorization 274–280
 - authorization strategy 274–276
 - authorization, actions 279
 - auto link 251
 - avoiding code duplication 200, 278
 - behaviors 42–44
 - bookmarkable pages 268
 - browser caching 229
 - compound components 199, 204–206
 - compound components, synchronizing models 211
 - compound identifiers 323
 - compound input components 209–213
 - configuration 143, 337
 - configuring components 203
 - configuring to use Spring 306
 - conversion mechanism 83
 - conversions 293–298
 - conversions, customizing 295–298
 - conversions, overriding for an application 297–298
 - conversions, overriding for one component 296–297
 - conversions, using converters 294
 - conversions, using models 293
 - converters 221, 293–295
 - CSS 223
 - custom request targets 231
 - custom resource loading 291–293
 - definition 5
 - domain-specific abstractions 14
 - domain-specific components 213
 - dos and don'ts 49, 64, 121, 147, 194, 336, 339
 - download links 228
 - extending/configuring components, using inheritance or composition 42–44
 - filter 49
 - final classes and methods 221, 259
 - form processing 209, 212
 - generic components 199
 - header contributions 247–251
 - header contributions, behaviors 248
 - header contributions, performing with package resources 259
 - header contributions, using templates 249
 - hiding components 280
 - hiding implementation details 202
 - images 223
 - indexing pages 229
 - initializers 230–231
 - interception URLs 272
 - Java components 33–34
 - JavaScript 223
 - just HTML 12–13
 - just Java 11
 - layering applications with Spring 304–313
 - list of components 106
 - loading templates 291
 - logging facility 345
 - matching component hierarchy with markup structure 37–38
 - metadata 207
 - model 39–42
 - model, introduction 33
 - models, converting values 221
 - nested form components 208
 - no code in markup 12
 - package resources, performing header contributions 259
 - packaged resources 224–226
 - packaging resources with components 247
 - Page 34–35
 - page map 28
 - partial screen updates 239
 - passing state 268
 - pitfalls of using Hibernate with 317
 - previewing components 206
 - problems it solves 5–10
 - processing requests independently from pages 223
 - projects that implement authorization and authentication 281
 - protecting pages 274
 - protecting pages, example 276
 - repeaters 216
 - repeaters, item-reuse strategies 219
 - Request 28
 - request processing 25–31
 - request processing, objects 25–29
 - request processing, steps 29–31
 - RequestCycle 28
 - RequestCycleProcessor 28
 - RequestTarget 29
 - Response 28
 - reusable components 199, 234–236, 258–262
 - security 267
 - self-contained components 224–226, 259
 - servlet 49
 - Session 26
 - session-relative pages 268
 - sessions 269–270
 - sessions, expiration 271
 - SessionStore 27
 - setting response headers 227
 - settings 276
 - stable URLs 229
 - switching between development and deployment mode 337
 - templating 214
 - the right abstractions 13–14

- Wicket (*continued*)
 - traditional request/response cycle 239
 - transparent state management 10
 - URL encodings 333
 - using factory methods to create custom components 210, 254, 256
 - using indirection in custom components 255
 - validation 212
 - versioning 215
 - what it is 10–14
 - what it is not 3
 - why reusable components 200
 - why Wicket 4–10
 - WicketRuntimeException 343
 - working with lists 216–217
 - working with lists in forms 217–221
 - XML namespace 137
 - Wicket and Ajax 239–247
 - advantages of using behaviors 258
 - behaviors 245–247
 - components 243–245
 - debug window 242
 - engine 242–243
 - goals 242
 - gotchas 262–263
 - JavaScript event handlers 245
 - request/response cycle 240
 - requirements for repainting components 244
 - updating components 253
 - using third-party engines 258–261
 - Wicket components
 - hiding 278
 - hierarchy 13
 - identifier 19
 - instantiation listener 275
 - introduction 31–44
 - and markup 35
 - page 34–35
 - programmatically disabling 278
 - replacement 213–221, 261–262
 - replacement, gotchas 215
 - Wicket examples
 - Ajax behavior 247
 - Ajax component 246
 - custom request target 29
 - custom session 27
 - introduction 24
 - Label 36
 - URL converter 296
 - using Ajax component 244
 - Wicket filter 311
 - wicket id 19
 - Wicket identifier, adding to markup 58
 - Wicket localization
 - filename patterns 287
 - loading localized markup 289–291
 - looking up messages for components, algorithm 288–289
 - looking up messages for components, mechanism 286
 - multiple languages 284–291
 - resource bundles or loading localized markup 291
 - support 283
 - Wicket resources 223
 - Captcha 232–236
 - component-hosted 228
 - dynamically generated 226–231
 - lazy initialization 229
 - PDF, RTF, Excel, and so on 232–236
 - shared 229–230
 - Wicket Stuff 242, 248, 281
 - Wicket tag 338
 - removing 136
 - wicket:child 188
 - wicket:container 137
 - wicket:enclosure 132, 273
 - wicket:extend 189
 - wicket:fragment 182
 - wicket:head 179, 189, 250–251
 - wicket:id 32, 54, 58
 - wicket:link 251, 291
 - wicket:message 285–288
 - wicket:panel 76, 179, 250, 273
 - wicket:remove 55, 206
 - wicket.properties 230
 - wicket:link 119, 179, 185, 193
 - not refactoring safe 120
 - wicket-ajax.js 242, 255
 - wicket-ajax-debug.js 242
 - wicket-ajax-debug-drag.js 242
 - wicketAjaxGet 255
 - wicket-auth-roles 281
 - wicket-contrib-yui 248
 - wicket-ioc 308
 - wicket-jmx 347
 - WicketMessage 56
 - WicketRuntimeException 90, 94, 343
 - wicket-security-swarm 281
 - wicket-security-wasp 281
 - wicket-spring 309, 311
 - using proxies from 308–309
 - wicket-spring-annot 309
 - WicketTester 322
 - assertComponentOnAjaxResponse() 325
 - assertContains() 324
 - assertErrorMessages() 328
 - assertLabel() 322
 - assertListView() 329
 - assertModelValue() 324
 - assertNoErrorMessage() 328
 - assertNoInfoMessage() 328
 - assertRenderedPage() 326
 - clickLink() 325, 329
 - getTagById() 324
 - newFormTester() 326
 - setupRequestAndResponse() 324
 - startPage() 322
 - startPanel() 328
 - WicketWebBeans 320
 - widget, defined 32
 - wiki 262
 - Window 34
 - wrapping a component around child markup 189–191
- ## X
-
- XML
 - configuring Spring with 306
 - declaration 290
 - message bundle 286
 - XMLHttpRequest 239
 - XSS 113
- ## Y
-
- Yahoo User Interface. *See* YUI
 - YUI 242, 248
 - library 242