

Symbols

- _diffs 132
 - subdirectories 132
- _styled theme 132
- _unstyled theme 132
- action, ActionCommand
 - suffix 271
- BaseImpl 82
- Impl classes 79
- LocalServiceImpl 82
 - making customizations in 83
- .aol CSS selector 146
- .browser CSS selector 147
- .camino CSS selector 146
- .css file 134
- .firefox CSS selector 147
- .flock CSS selector 147
- .gecko CSS selector 146
- .icab CSS selector 147
- .ie CSS selector 147
- .iphone CSS selector 148
- .konqueror CSS selector 147
- .mac CSS selector 148
- .mozilla CSS selector 147
- .netscape CSS selector 147
- .opera CSS selector 147
- .os CSS selector 148
- .safari CSS selector 147
- .sun CSS selector 148
- .webkit CSS selector 146
- .win CSS selector 148
- \$browserSniffer 141

Numerics

1970 122

A

- account object 101
- action
 - firing with events 216
 - splitting into separate classes 270
- action attribute 104
- action URL 116
 - defining 114
- action.package.prefix 271
- ActionCommand 270–271
 - implementing 271–272
- ActionURL 52, 54, 315
- ActionUtil 104
- Activiti 180, 188
- Activities application 12
- Activities portlet
 - and social networking 159
 - interpreting activities 172–175
 - social requests and 168
- activity, social 167
- addCommunityPermissions 182
- addGuestPermissions 182
- aggregation 157
- Alloy UI 107, 136
 - as interface metaframe-work 102
 - as JavaScript library 136–140
 - components 136–137
 - JavaScript enabled 136
 - well tested 138
- CSS3 styling 136
- data, selecting 122
- design 137–138
- forms, consistency of 138
- generating markup with 103
- global AUI() object 140
- introduction to 102–105
- layouts, consistency of 138
- lazy loading 136
- option tag 122
- tag libraries, consistency of 138
- taglibs
 - creating form with 102–110
 - generating field types 120–123
 - use for forms 123
- tags 136
- Amazon.com 7
- ANT_HOME 38
- ANT_OPTS 38
- Ant, installing 38–39
- application, large,
 - organizing 270–272
- application.css 134
- Applications window 11
- Asset Publisher 180, 184–188
 - dynamically selecting content 198
- Asset Publisher XE 200
- asset renderer 184–188
 - factory 184
- asset, adding with
 - entities 180–184
- assetCategoryIds 182
- assets 180

assetTagNames 182
 attributes attribute 182
 AUI() 140
 Author tag 76

B

Banner section, default 133
 base.css 134
 BaseFriendlyURLMapper 267
 BaseIndexer 284
 BaseModelListener 217
 best practices 257–261
 deciding if you need
 Ext 259–261
 for customizing
 Liferay 258–259
 for developing applica-
 tions 257–258
 Blogs Aggregator 15
 branding, enforcing 141
 browser, specifying for a
 theme 146
 build-service 77
 build.properties 39–40
 customizing 41
 build.xml 46

C

Calendar 12
 case study 65
 categories.jspf 223, 230, 236
 category
 custom 60
 vs. tag 196
 Chan, Brian 331
 class 105
 Classic theme 132
 classNamedId 203
 code, organizing 33
 collaboration tools 14–15
 collaborative application
 adding assets to 179–188
 designing 178
 discussions and
 ratings 199–200
 workflow 188
 collection types 18
 collection, pages 20
 color_schemes folder 144
 colorScheme object 101
 column, defining 76
 command attribute 182

community, definition of 19
 communityPermissions 182
 company object 101
 companyId 182, 233
 configuration action 119
 Configuration button 119
 configuration, reusing 96
 configurator 278
 Constants.CMD 182
 contact object 101
 content
 filtering 272
 tagging and
 categorizing 196–199
 Content section, default 133
 contributing to Liferay 333–334
 bug reports and feature
 suggestions 334
 code 333
 community programs 333
 documentation 334
 Liferay IDE 334
 Control Panel 24
 adding portlet to 98
 as Edit mode alternative 94
 sections 98
 workflow configuration 195
 controller layer 92
 Counter utility 82
 createDate 183
 CSS
 in themes 134–135
 selectors
 and styling conven-
 tions 146–148
 coding conventions 148–149
 custom query 200
 custom.css 134
 customization, with Ext
 plugins 243–261
 CustomSQLUtil 204

D

DAO layer 79
 data
 displaying with search con-
 tainer 111–116
 editing and deleting 114
 selecting 123
 Data Access Object 74, 79
 data source, wiring up in
 Spring 275–276
 Data Transfer Object 79
 database
 accessing 273–276
 connecting to 37–38
 setting up 35–37
 date picker 121–122
 decorator pattern 220
 default.xml 117, 201
 portlet resource section 119
 DefaultFriendlyURLMapper 267
 defineObjects tag 51
 dependency injection (DI) 73
 discussions 199–200
 enabling ratings on 200
 DLApp 341
 DLAppService 341
 Dockbar 10, 21, 23
 Add menu 24
 default styling 134
 Go To menu 26
 Manage menu 24
 pin icon 24
 Toggle Edit Controls 26
 User Account link 26
 dockbar.css 134
 docroot 46
 Document 283
 converting entity into 285
 deleting 285
 document handling 335
 in Liferay 6.1 339–342
 with low-level API 337–338
 Document Library 14
 API 335
 document types 339
 handling multiple repositories
 in 6.1 341
 hooks 336
 metadata sets 339
 older API 337
 limitations 338
 Document Library Application
 API 341–342
 Documents API 335–342
 model objects 337
 repositories 341
 using 342
 doEdit() 49, 53
 doInterpret() 174
 doReceive() 281
 doView() 49
 DTO layer 78–79
 and service layer 79
 buffering business logic and
 database code 80
 implementing 80–84

E

-
- ECJ 41
 - Eclipse 296–303
 - and Liferay IDE 307
 - Ant integration 302
 - debugging 302
 - fixing project dependencies 302
 - plugin project, setting up 300–302
 - server runtime 298–300
 - working sets 298
 - workspaces 297–298
 - Edit mode 46
 - alternatives to 93–95
 - implementing 48–57
 - JSP 53
 - edit_category.jsp 223, 230
 - edit.jsp 53
 - EditUserAction 246
 - Enterprise Java Beans (EJBs) 73
 - entity
 - assets 181
 - defining 76
 - Entity tag 76
 - ERP, software and portlets 314
 - Expando 224
 - attributes 233
 - definition of 228
 - persistings 228
 - Expando API 228–234
 - token delimiters 233
 - tokens 232
 - and ServiceContext 233
 - Expando field,
 - customizing 232–233
 - ExpandoAttribute-- 233
 - ExpandoAttributeName-- 232
 - ExpandoBridge 227–228
 - ExpandoBridgeAttributes 183
 - Ext environment, old, migrating to Ext plugin 259
 - Ext plugin 242–261
 - and Liferay upgrades 249
 - as wrapper for core source 243
 - caveat 42
 - customizing Struts actions 246
 - deciding whether to use 259–261
 - decreasing need for 258
 - deploying 247–249
 - Great Power 243, 246
 - how to use 245–246
 - implementing listeners 253
 - introduction to 243–249
 - reverting plugins 248
 - strategy 246–247
 - structure 244–245
 - Struts and 249–253
 - undeploying 248
 - uses for 252–253
 - vs. hooks 213–214, 258
 - vs. other plugin types 243
 - vs. other plugins 248
 - ext-impl directory 244–245
 - ext-lib directory 244
 - ext-service directory 244
 - ext-spring.xml 275
 - ext-util-bridges directory 245
 - ext-util-java directory 245
 - ext-util-taglib directory 245
 - ext-web directory 245
 - Extension plugin. *See* Ext plugin

F

-
- Facebook 7–8, 155
 - adding portlet to 156
 - games 156
 - feedback, providing on
 - form 105–108
 - Field 283
 - field label 105
 - field types, generating 120
 - field-processing engine 233
 - field, repeatable 138
 - filtering content 273
 - finder method, defining 77
 - finder, custom query as 202
 - folksonomy 197
 - for 195
 - form
 - creating with AlloyUI 102–110
 - field validation 109–110
 - messages and feedback on 105
 - translating messages on 108
 - forms.css 134
 - FreeMarker
 - and Service Builder templates 142
 - developing themes 142
 - writing templates in 14
 - friendly URL 255, 265–270
 - benefits of 269
 - mapper, declaring 266–267
 - mapping 267–268
 - passing parameters with 268–270
 - Friendly URL servlet 254

- FriendlyURLServlet 268
- FriendsRequestKeys.ADD_FRIEND 167

G

-
- Gecko rendering engine 146
 - GenericPortlet 48
 - MVCPortlet subclass 93
 - getExpandoBridgeAttributes() 228
 - getInitParameter() 49
 - getSetting() 142
 - GetterUtil 192
 - getTitle() 174
 - Go To menu 26
 - <group-limit> tag 140
 - group, type, checking for 165
 - guestPermissions 183

H

-
- Hello World portlet 43–45
 - Hello You portlet 45, 57–61
 - configuring 47
 - Help mode
 - alternatives to 94
 - Hibernate 73, 201
 - custom queries 201
 - database persistence 79
 - hook 212–239
 - and portlet actions 232
 - basics 213
 - configuration file 225–226
 - creating 214
 - customizing with 214–221
 - JSP files 218–219
 - language properties 217–218
 - listener properties 217
 - portal event properties 216
 - portal properties 215–216
 - services 219–221
 - definition of 212
 - generating service layer in 223–225
 - liferay-hook.xml 214
 - portal.properties 215
 - turning off terms of use feature 215
 - vs. Ext plugin 213–214, 258
 - writing with text editor 214
 - hot code replace 41
 - HSQL 34–35
 - default database, overriding 37

I

ICEfaces 99
 icon-delete tag 114
 implicit parameter 268
 include() 50
 indexer
 declaring 283
 getSummary() 284
 implementing 284
 methods 284
 indexing 282–292
 and message bus 283
 reindexing 286
 init_custom.vm 133
 init.jsp 99, 101, 214, 219
 init.vm 133
 init() 49
 initialization code, putting in
 one place 99
 initialization parameter 271
 Inkwell
 adding registered users and
 their products 88–89
 administrative portlet 95
 choice of Liferay Portal 67
 company profile 66–67
 directory structure 131
 ext plugin 244
 extranet 69
 form, creating 102–110
 internet 69
 intranet 69
 introduction 65–69
 portal
 phase 1 requirements 68–69
 portal, design 67–68
 product
 adding 80–83
 deleting 83
 querying for 84
 Product Registration
 portlet 69–72
 design 69–70
 portlet modes 72
 table design 70–72
 Product Registration portlet
 class 124
 profile pages 160–164
 sharing services 87–88
 shopping cart, customizing
 with hooks 221–239
 shopping category user
 interface 235–239
 theme 152
 website requirements 67

inter-portlet communica-
 tion 316–330
 example 318–328
 invokeTaglibDiscussion 200
 IPC 316
 isEmailAddress() 110
 isPhoneNumber() 110

J

January 1, 1970 122
 Java SDK, installing 32–33
 JAVA_HOME, setting 32–33
 JavaScript 136, 151
 and Alloy UI 136
 available functionality 138
 custom 138–140
 default file 135
 in themes 135
 objects, calling 139
 page-specific 135
 settings fields 135
 YUI 136
 jBPM 180, 188
 JCRHook 336
 JDBC driver, for MSQL 38
 JIRA 7
 Journal (Web Content) resource
 listeners, overriding 253
 JSP
 developing themes 142
 Liferay objects in 101
 JSP dependencies, having one
 location for 99–102
 JSP file
 customizing with
 hooks 218–219
 overriding 230–231
 jspPage 93, 115–116, 266, 268
 JSR-168 standard 53, 313, 317
 JSR-286 standard 53, 313

K

Kaleo 180, 188

L

language
 properties, customizing with
 hooks 217–218
 translating to 108–109
 language keys, getting with
 JavaScript 139

language properties 193–195
 Language-ext.properties 168, 244
 Language.properties 105
 activity message keys 172
 location in source 109
 languageId 183
 languages, multiple, support-
 ing 105
 LanguageUtil 192, 291
 layering
 and persistence 79–80
 importance of 78–79
 layout 254
 layout template 149–151
 anatomy of 150–151
 creating 149
 deploying 151
 layout.css 134
 layout.jsp 257
 LayoutAction 255
 execute() 255
 includeLayoutContent()
 255–257
 WebKeys.LAYOUT_
 CONTENT 257
 processLayout() 255
 processPortletRequest() 255
 layoutFullURL 183
 layouttpl folder 149
 layoutTypePortlet object 101
 layoutURL 183
 Lesser General Public License
 (LGPL) 9
 Liferay bundle 33
 JDBC driver 38
 sample web site 34
 Liferay Home 37
 Liferay IDE 303–307
 installing 303–304
 perspective 304
 projects, creating and import-
 ing 306–307
 runtimes 304
 Search Container snippet 304
 Liferay Portal 213, 215, 218, 232
 adherence to JSR-286 9
 and Web 2.0 8
 applications, default 11
 as a collaboration tool 14–15
 as a development platform 5
 as application aggrega-
 tor 10–12
 as CMS 12–14
 benefits of 8–9
 best practices 257

Liferay Portal (*continued*)
 bundle, choosing 33
 bundle, connecting to database 37
 company history 331–332
 configuration files 47
 connecting to database 37
 contributing to 331
 customizing 258–259
 documentation 195
 extensibility 15–16
 Extension plugin 243
 hooks 212
 installing 31–34
 introduction 9–17
 linking to software that isn't
 open source 9
 navigating 23–26
 open source 8
 organization 19
 role 19
 Service Builder 9
 size of 8
 Struts 1.x, reason for using 249
 templates 149
 user interface 9
 versions 9
 liferay-display.xml 47, 59–60, 98,
 327
 liferay-hook.xml 214, 225
 liferay-look-and-feel.xml 140–146
 and color schemes 145
 defining settings in 142–144
 liferay-plugin-
 package.properties 47, 328
 liferay-portlet-ext.xml 253
 liferay-portlet.xml 47, 59, 98,
 185, 266, 326–327
 declaring indexer in 283
 setting up schedule in 282
 <liferay-ui:custom-attribute-list>
 tag 183
 <liferay-ui:custom-attribute>
 tag 183
 <liferay-ui:input-permissions>
 tag 182
 <liferay-util:buffer /> 219
 Liferay.AutoFields 138
 Liferay.ColorPicker 139
 Liferay.Language 139
 Liferay.Notice 139
 Liferay.Panel 139
 Liferay.Upload 139
 Liferay.Util 139
 LiferayRepository 341

LinkedIn 155
 Linux 148
 MySQL and 36
 MySQL issues 37
 linux CSS selector 148
 listener 217
 and message bus 278
 custom 217
 customizing with Ext plug-
 ins 253
 declaring 282
 implementing 280
 properties, customizing with
 hooks 217–218
 locale object 101
 LocalRepository 342
 location 68
 logging, enabling 51
 look and feel, enforcing 148
 Lucene, search syntax 282, 288
 LUM
 acronym, explained 31
 setting JAVA_HOME 32

M

main.css 134
 Manage menu 24
 Manage Pages function 144
 Message 281
 Message Boards 11
 message bus 277, 281
 and indexing 283
 and Message Boards
 portlet 277
 and scheduler 282
 configuring 278–280
 Spring configuration 278
 uses for 277
 message, translating to multiple
 languages 108–109
 MessageListener 282
 messaging-spring.xml 279
 META-INF/custom_jsp 226
 method, giving same name as
 action URL 104
 [Model]LocalServiceImpl 220
 [Model]LocalServiceWrapper
 219
 model hints 122
 model layer 92
 model resource,
 defining 119–120
 Model-View-Controller (MVC).
 See MVC

ModelListener 217
 modifiedDate 183
 MVC 92–96
 according to Liferay
 approach 95
 alternative to 96
 implementing 93
 Liferay approach 96
 benefits of 124–126
 MVCPortlet 93
 as default portlet class 47
 benefits of using 124
 controlling page manage-
 ment with 96
 managing page flow 116
 render parameters 268
 MySpace 7, 155
 MySQL 37
 command-line utility 36
 connecting Liferay to 35
 installing 36
 JDBC driver 38
 Linux issues 37
 root password, setting 36
 mysql 37–38

N

namespace tag 54
 naming conventions 270
 navigation.css 134
 navigation.vm 133
 NetBeans 307–311
 and Ant 307
 installing 307
 plugins 307
 project
 debugging 311
 setting up 309–311
 settings 311
 server runtime 307–308
 notification area,
 controlling 139

O

object, in JSPs 101
 online interaction 199
 open source 33
 and Liferay Portal 8
 as improvement to Portal API 7
 benefits of 7
 hooks and 212
 issues with 211

operating system agnosticism,
 rules of 31
 operating system, specifying for
 a theme 148
 organization 68
 definition of 19

P

page

Banner section 133
 Content section 133
 customizing 133
 default structure 133
 delivering 253–257
 JavaScript in 135
 plid 254
 processing logic 254–257
 template, processing 255–256

page management, controlling
 with MVCPortlet 96

page scope 22–23

pageContext 112

parameter

implicit 268
 initialization 271
 render 268

pathMain 183

pattern tag 268

permission checker, customiz-
 ing with Ext plugins 253

permission checking 273

permissionChecker object 101

permissions 116–120

configuration, pointing to 116
 configuring 117–120
 default 120

permissions action 119

permissions.checker 253

personal community 20

plid 183, 254

plugin

adding to page 44
 benefits of 42
 creating 43
 definition 42
 deploying 43–45

Plugins SDK 39, 42, 72, 214–215
 and Ant 38

and Liferay IDE 305–306

and PHP/Ruby

applications 314

automatic translation 109

build.properties 40

configuring 39–41

customizing

build.properties 41

installing 31, 39

layouttpl folder 149

packaging themes 131

setting up 38

portal

definitions of 4

designing 26–28

categories 28

content 28

questions to ask 27

disappointment with 6–8

event properties, customizing

with hooks 216

history of 4–9

properties

customizing with Ext

plugins 253

customizing with

hooks 215–216

reasons for lack of widespread

acceptance 7

structure of 18–23

Portal API 7

portal instance, creating 141

portal_normal.vm 133

portal_pop_up.vm 133

portal-ext.properties 37, 215,
 244, 276

creating profile pages 160

portal.layout tile 257

portalURL 183

PortalUtil 203, 256, 291

PortalUtil.renderPortlet() 256

portlet

adding 11

adding to Control Panel 98

adding to social networking

site 156

and ERP software 314

and frameworks 314

as fragment of web

page 312–313

attachments 342

core changes, customizing

with Ext plugins 253

defining in deployment

descriptors 97–99

deploying 57–61

deploying and testing 328

event 328–330

declaring 329

definition 321–322

distribution 330

implementing 322–325

order 330

receiver 321

sender 321

ID 161–162

mapped to human-read-
 able names 163

initializing 49–52

instanceable 22

inter-portlet communica-
 tion 315–316

introduction to 5

liferay-display.xml 327

liferay-portlet.xml 326–327

logic, implementing 48

modes 315

naming 43

non-instanceable 21–22

phases 315

PHP/Ruby application as 314

portlet.xml 314

preferences, modifying 253

renamed, telling Liferay

about 60–61

renaming 59

render phase 325

rendering 256

scope 20–23

security 116–120

specification, history
 of 316–318

structure of 314–315

URLs in, types 54

vs. servlet 315

weight 98

write as plugin 251

Portlet 1.0 specification 313

history of 317

limitations of 317

Portlet 2.0 specification 313

backward compatibility 315

history of 317–318

use of Java 5 322

portlet actions 45, 119, 232, 251

and processAction method 54

defining 52

portlet class

custom 47

default 47

portlet configuration 97

portlet modes 45–46, 256, 271

defining 72

lack of use 93

portlet name, vs. display
 name 61

portlet plugin,
 developing 42–45
 portlet preferences 45
 portlet project
 anatomy of 46–47
 configuring 96–102
 directory structure 46
 folders 46
 portlet resource, defining 119
 portlet-ext.xml 253
 portlet-model-hints.xml 122
 portlet.css 134
 portlet.jsp 255
 portlet.properties 116
 portlet.vm 133
 portlet.xml 47–48, 59, 97–98,
 105, 185, 266, 271, 314
 tag 48
 portletConfig 51
 portletDisplay object 101
 PortletPreferences 50
 portletPreferencesIds 183
 primary key, generating 81
 private layout 161
 private page 161
 processAction() 54
 processCommand() 272
 ProdRegValidator 104
 Product Registration 69, 87, 160
 ProductActivityKeys 172
 profile page, defining
 content 162–164
 public layout 161
 public page 161
 content, defining 162–164

Q

QName 321
 query, custom 200
 as finder 202
 calling 204
 hooking to service 202

R

rating 180, 199
 and comments 200
 enabling on discussion 200
 granularity 200
 IDs 203
 ratings tag 200–201
 realUser object 102
 reference, defining 169

relationship, one to many 87,
 166
 remote service 76
 render parameter 93, 115, 126,
 268
 render URL 116
 render_portlet.jsp, you 256
 rendering engine, specifying for
 a theme 146
 renderRequest 51
 renderResponse 51
 RenderURL 54
 repeatable field 138
 Repository interface 341
 Repository table 342
 RepositoryServiceUtil 342
 request processing, forwarding
 to a JSP 50
 Requests portlet
 and social requests 160
 social requests 159
 resource
 permissions, defining 119
 persisting 82
 resource-actions folder 117
 resourcePrimKey 269
 role, definition of 19
 RuntimePortletUtil 255
 processTemplate() 256

S

scheduler 281–282
 and message bus 282
 <scheduler-entry> 282
 scopeGroupId 102, 165, 183,
 233
 search
 for portal capabilities 286
 implementing in UI 287–292
 Lucene used for 282
 search container 110–116, 134,
 222–223
 adding actions to 113
 custom columns 204–206
 editing and deleting
 data 114–116
 presenting data 111–114
 security, social relations and 164
 select box 122
 sender, implementing 280–281
 separation between site design-
 ers and site
 programmers 99
 separation of concerns 78
 server administration 35–38
 serveResource() 315
 service
 creating in hooks 221
 customizing with
 hooks 219–221
 overriding 227–230
 Service Builder 9, 77, 80, 84, 232
 accessing existing data
 with 273
 and custom queries 201
 and Hibernate 201
 and Hibernate and Spring 74
 as code generator 75
 automatic generation of DAO
 and DTO layers 79
 class inheritance 220
 creating Spring configura-
 tion 87
 DAO layer 78
 database persistence config-
 uration 77
 defining interface first 83
 DTO layer 78
 encouraging separation of
 concerns 79
 entity IDs 203
 generating DB code 72, 78
 generating interfaces 202
 generating Javadoc 76
 generating Spring configura-
 tion 169
 implementation layer 77
 interface layer 77
 introduction 72–78
 layering functionality 78–84
 needs met by 73–75
 running 77–78
 service, injecting 169
 service.properties 77
 service.xml 75–77
 sharing services 87–88
 Spring and Hibernate config-
 uration files 77
 workflow configuration 179
 service layer
 adding updateStatus 190
 and DTO layer 79
 generating in a hook 223
 service.properties 77
 Spring configuration files 281
 Spring configuration files
 in 77

- service.xml 75–77, 83–84, 223–224
 - <finder>tags 79
 - building 272–273
 - injecting service 169
 - wrapping database calls in a transaction 169
 - ServiceContext 181, 228
 - and Expando tokens 233
 - attributes 182
 - custom attributes 228
 - putting custom fields in 227
 - reason for creation of 233
 - storing custom attributes in 233
 - servlet filter, customizing with Ext plugins 253
 - SessionAction 216
 - SessionErrors 107
 - SessionMessages 107
 - SimpleAction 216
 - Slogan Contest 178–206
 - search feature 287
 - sending messages 279
 - URL 266
 - view URL 269
 - view-level permission filtering 273
 - social activity 167–175
 - adding in service layer 168–172
 - message keys in Language.properties 172
 - publishing 173
 - user ID and 172
 - Social API 12
 - social networking 117
 - connecting users 156
 - connecting your site 156
 - Liferay features 158
 - reasons for importance of 155–158
 - relationships 159
 - social activities 167
 - user experience 157–158
 - social networking API 156
 - Social Networking portlet plugin
 - features 158–160
 - publishing 159
 - social relations 159
 - social requests 159–160
 - installing 158
 - Request portlet 159
 - Summary portlet 159
 - social relation 159, 164–167
 - and Summary portlet 160
 - checking for 166
 - security and 164
 - types 166–167
 - requesting 167
 - social relationship, coding for 164–167
 - social request
 - and Social Requests portlet 160
 - sending 159–160
 - social-networking-portlet project 167
 - SocialActivity table 172
 - SocialActivityInterpreter 174
 - SocialRelationConstants.java 166
 - Spring 73
 - and message bus 278
 - configuration, generating 169
 - defining data source with 273–276
 - SQL, custom queries 200–206
 - startWorkflowInstance() 189
 - Struts 79, 250, 254, 258
 - 1.x 249, 255
 - actions 249–250
 - customizing with Ext plugins 246, 251–252
 - declaring 252
 - forwards 250
 - locating 251
 - and Ext plugin 249
 - introduction to 249–251
 - tiles 250–251
 - struts 237
 - struts_action 251
 - struts-config-ext.xml 246, 252
 - struts-config.xml 245, 249
 - StrutsPortlets 245
 - style
 - custom 134
 - default, overriding 134
 - Summary portlet 159
 - and social relations 159
 - social requests 159
 - system-ext.properties 244
 - folksonomy 197
 - vs. category 196–197
 - taxonomy 197
 - template 149
 - <template-extension> tag 142
 - TemplateProcessor 256
 - terms of use feature 215
 - text message, internationalized 244
 - theme 258
 - _diffs 132
 - _styled 132
 - _unstyled 132
 - color schemes 144–146
 - conditional settings 142–144
 - conventions 146–149
 - for coding 148
 - for styles 146–148
 - specifying browsers 146
 - specifying operating systems 148
 - specifying rendering engines 146
 - CSS in 134–135
 - default files 134
 - default (Classic) 132
 - default paths, modifying 141–142
 - generating 131
 - introduction to 129–132
 - JavaScript in 135
 - limiting by company 140–141
 - markup 132–135
 - security 144
 - structure of 131–132
- theme markup 133
- theme object 102
- themeDisplay 102, 165, 174
- tightly coupled design 78
- tiles-defs.xml 245, 251
- time, perception of 122
- timeZone object 102
- to 302
- translation, automatic 108–109
- U**
-
- updateStatus(), adding to service layer 190
 - URL
 - making friendly 265
 - parameters 266
 - URL object 52
 - useBean tag 54
- T**
-
- table relationships, defining 84–87
 - tag
 - enabling 197–199

- user group
 - definition of 19
 - page template 20
- user object 102
- user, gets a personal community 20
- userDisplayURL 183
- userId 183
- Using Liferay Portal 195
- uuid 183

V

- validation, and hooks 234
- Validator 109–110
 - methods 110
- Velocity
 - _unstyled theme 132
 - developing themes 142
 - resource listeners, overriding 253
 - variables
 - corresponding to Java objects 133
 - custom 133
 - modifying theme default paths 141
 - writing templates in 14

- VelocityVariables.java 133
- view action 119
- view layer 92
 - JSPs as 93
- View mode 45, 48
 - implementing 49
 - JSP 51
- view-level permission filtering 273
- view.jsp 51
 - search form 287

W

- Wall portlet
 - checking group type 165
 - checking if portlet is on page 165
- Web Content Display, introduction to 12–14
- web content, publishing 12
- web desktop 6
- web site, sample, removing 35
- WEB-INF 46
- WEB-INF/src 46
- WEB-INF/src/META-INF, and message bus 278

- WebKit rendering engine 146
- website, sample, removing 37
- Wiki application 11
- Windows
 - MySQL and 36
 - organizing code 33
 - path length limit 33
 - setting JAVA_HOME 32
- workflow 188–196
 - benefits 188
 - big picture 189–190
 - enabling 190–191
 - instance, calling 190
- workflow configuration 195
- workflowAction 183
- WorkflowHandler 189–190
 - implementing 191
 - updateStatus() 190
- WorkflowHandlerRegistryUtil 189

Y

- YUI 136

