

Index

■ A

- abandon parameter, 121
- AbstractJpaQueryProvider base class, 217
- AbstractTransactionalJUnit4SpringContextTests class, 459
- acceptance criteria, 31
- Account class, 339–340
- Account Detail, 38
- AccountDaoJdbc, 349
- AccountExecutive object, 244, 246–249, 252, 257
- AccountExecutiveDaoImpl, 246–247, 252
- accountExecutiveItemProcessor, 255
- accountNumber, 149
- AccountSummary class, 149–150
- accountSummaryReader, 146
- accountSummaryUpdater, 146
- AccountTransaction, 370
- accountTransactionQtyItemReader, 367
- AccountTransactionQuantity, 365, 367
- AccountTransactionQuantityRowMapper, 366
- AccountTransactionRowMapper, 371
- activeMg script, ActiveMQ server, 434
- ActiveMQ JMS queues, 445
- ActiveMQ server, 434
- ActiveMQ XSD, 445
- Address class, 339–340
- Address Number field, 168
- address.city property, 355
- address.state property, 355
- adhocLoopJob, 17
- Admin project, Spring Batch, 9
- Admin tool
 - external stoppage of jobs via, 155–157
 - launching jobs with, 135–141
- afterChunk, 81
- afterJob method, 59
- afterPropertiesSet method, 238, 344
- agile development, 29–32
 - source-control systems, 32
 - TDD, 32

- true development environments, 32
- user stories, 30–32
- algorithmic traders, 365
- allow-start-if-complete attribute, 145, 162
- Alternative output destination ItemWriters, 259
- AOP (aspect-oriented programming), 10
- appendAllowed option, 262
- appendAllowed parameter, 272
- application data model, 42
- application layer, Spring Batch, 7
- ApplicationContext, 111, 120
- applicationContext.xml file, 335
- applications, sample. *See* sample jobs
- applyFeeWriter, 375
- applyTransactionStep, 145
- aspect-oriented programming (AOP), 10
- assert methods, in test cases, 449
- AssertFile class, 469
- assertUpdates option, 279
- AutomaticJobRegistrar bean, 113, 122, 133–134
- availability, 3

■ B

- BackoffPolicy interface, 96
- basicJob.xml file, 49, 51
- Batch applications, profiling, 387–399
 - CPU profiling, 393–396
 - memory profiling, 397–399
 - and VisualVM, 388–392
- Batch class, 111
- batch processing
 - challenges of, 3–4
 - history of, 2
 - in Java language, 4–5
 - scaling processes, 4
 - with Spring Batch
 - framework of, 7–10
 - uses for, 6–7
- batch profile, 429
- batch_job_execution, 87

BATCH_JOB_EXECUTION table, 26–27
 batch_job_execution table, 48
 batch_job_execution_context, 48
 BATCH_JOB_EXECUTION_CONTEXT table, 27–28
 batch_job_instance, 48
 BATCH_JOB_INSTANCE table, 25
 batch_job_params, 48, 53
 BATCH_JOB_PARAMS table, 25–26
 batch_job_params table, 58
 batch_step_execution, 87
 BATCH_STEP_EXECUTION table, 26–27
 BATCH_STEP_EXECUTION_CONTEXT table, 27–28
 batch.address, 57
 batch.jdbc.driver, 334
 batch.jdbc.password, 334
 batch.jdbc.url, 334
 batch.jdbc.user, 334
 batch.name, 57
 batch.properties file, 50, 101, 143, 334, 336
 batch.runDate, 57
 batch.schema, 334
 BatchStatus, 86
 batchStep, 93
 bean namespace, 116
 bean tag, 101–103
 BeanFactoryStepLocator object, 445
 BeanPropertyItemSqlParameterSourceProvider, 285, 355
 beans tag, 49
 BeanWrapperFieldExtractor, 266, 269
 BeanWrapperFieldSetMapper, 362
 beanWrapperMapperSampleJob, 17
 beforeChunk, 81
 BEGIN_DATE, 38
 BigDecimal, 176
 boolean stop(long executionId) method, 110

■ C

Calculate Pricing Tiers, 37
 calculateTiers, 366, 369
 calculateTransactionFees, 371–372
 call() method, 68
 Callable object, 68
 CallableTaskletAdapter, 68–69
 carStep, 75
 cashBalanceUpdateWriter, 374–375
 cd ~/spring-batch, 72

chunk processing, of steps
 chunk-size configuration, 76–81
 vs. item processing, 65–66
 chunk tag, 97
 ChunkContext, 54
 chunkHandler bean, 431
 ChunkHandler object, 422, 433
 chunking, remote, 16
 ChunkMessageChannellItemWriter object, 422
 ChunkResponse object, 431
 chunkWriter bean, 431
 ChunkWriter object, 433
 City field, 168
 ClassifierCompositeItemWriter, 326, 329–330, 353–354
 classifierFormatJob, 328
 close method, 320
 COBOL (Common Business Oriented Language), 2, 6
 ColumnRangePartitioner object, 441
 CommandLineJobRunner runner, 47, 110, 120–122, 132, 155, 157, 268, 429
 Comments option, 167
 COMMIT_COUNT field, 27
 commit-interval attribute, 68, 78
 commitIntervalStep, 76
 Common Business Oriented Language (COBOL), 2, 6
 completeItemProcessor, 255
 CompletionPolicy interface, 76, 78, 80
 CompositeCompletionPolicy, 78
 CompositeItemProcessor implementation, of ItemProcessor interface, 248–256
 CompositeItemWriter, 322–326, 374–375
 compositeItemWriterSampleJob, 17
 com.thoughtworks.xstream.converters.basic.DateConverter, 201
 conditional logic, of steps, 83–86
 conditionalStepLogicJob, 87–89
 configurationClass, 288
 containers, running jobs in, 132–135
 ContextLoaderListener, 137–138
 copyFileStep, 255
 copyJob.xml file, 169, 213, 240
 copyStep, 76
 core layer, Spring Batch, 8
 CPU profiling, 393–396
 CREATE_TIME field, 26
 createJobExecution method, 103
 createQuery() method, 217

- cron jobs, 2
- cronTrigger bean, 132
- currentBalance, 149
- cursor processing, 202–206, 209–213
- custom processing, 5
- Customer class, 170, 189, 193, 210, 218, 236, 257, 287, 339–340
- customer data, in statement job example, 336–355
 - creating transaction reader, 338–346
 - ItemProcessor for, 346–353
 - writing data, 353–355
- Customer items, 237, 248–249, 257, 294
- Customer objects, 236, 238, 241, 244, 247, 249–251, 256
- customer record fields, 337
- Customer table design, 278
- CustomerClassifier, 327
- customerCompositeWriter, 325
- customer.csv file, 278, 299
- CustomerDao Retry, 95
- customerDaoImpl, 252
- CustomerDaoJdbc, 348–349
- CustomerEmailConverter, 309
- CustomerFieldSetMapper, 341
- customerFile bean, 254, 265
- customerFile1.csv, customerFile2.csv file, 197
- customerFile3.csv file, 197
- CustomerFileReader layer, 339
- customerFilterJob, 17
- customerItemIdProcessor, 255
- customerImportWriter, 354
- CustomerItemPreparedStatementSetter, 280–281, 305, 307
- Customer.java, 286, 304
- CustomerLookupItemProcessor, 338, 346, 353–354
- CustomerOutputFileSuffixCreator, 314
- CustomerRowMapper implementation, 305
- customers
 - generating monthly statements, 41–42
 - importing transaction data, 40
- CustomerServiceImpl, 295–296
- CustomerStatementReader, 376
- customerStatementReader, 380
- customerTransaction.csv file, 337, 346
- customerTransactionItemWriter, 354
- customerTransactionReader bean, 338
- customerTransactionWriter bean, 338
- customerWithEmail.csv file, 322
- CustomerXmlHeaderCallback, 316

D

- DailyJobTimestamper, 58
- DAO (data access objects), 348
- data models, 42–43
- Database ItemWriters, 259
- database-based ItemWriters, 278–293
 - HibernateItemWriter, 285–290
 - JdbcBatchItemWriter, 278–285
 - JpaItemWriter, 290–293
- databases
 - importing stock closing prices into, 41
 - input, 201–218
 - Hibernate project, 208–213
 - JDBC API, 202–208
 - JPA API, 214–218
 - job repository in, 101–104
 - schema configuration for, 102
 - transaction configuration for, 103–104
 - paged access, with Hibernate project, 213
 - test configuration, 42–43
- dataSource, 101
- Datasource bean, 336
- dataSource option, 280, 307
- DataSourceInitializer class, 20, 333
- DataSourceTransactionManager, 101
- DATE_VAL field, 26
- DefaultJobParameterExtractor, 92
- DefaultJobParametersValidator, 57
- DefaultLineMapper, 362
- delegate option, 311
- delegatingJob, 17
- deleteFiles application, 135
- deleteFilesJob, 125, 130, 132, 137, 140–141
- DeleteFilesTasklet, 130
- delimited files, 173–179, 269–270
- DelimitedLineAggregator, 270
- DelimitedLineTokenizer, 265, 346, 362
- development environments, 32
- documentation, 17–18
- DOUBLE_VAL field, 26
- doWithRetry method, 94
- doWithRetry(RetryContext context), 94

E

- Eclipse, and VisualVM, 388
- encoding option, 167, 262, 273
- END_DATE field, 38
- END_TIME field, 26–27

- EntityManagerFactory, 291–293
- environmentParams, 72
- error handling, 157–159, 226–231
 - no input, 230–231
 - records
 - logging invalid, 228–230
 - skipping, 226–228
 - for steps, 93–98
 - item retry, 93–97
 - item skip, 97
- execute method, 73, 130
- executed from start to finish, 46
- executeInternal method, 128
- execution of jobs, 119–120
- ExecutionContext state, of jobs
 - manipulating, 62–64
 - overview, 61–62
 - persistence of, 64
- ExecutionContexts, 156
- EXIT_CODE field, 26–27
- EXIT_MESSAGE field, 26–27
- exit-code, 87
- ExitStatus, 70, 81, 84, 86, 145, 147, 151–152, 159, 469
- ExitStatus.FAILURE, 88
- ExitStatus.COMPLETED, 70
- ExitStatus.FINISHED, 72
- ExponentialBackOffPolicy, 96
- extensibility, 3
- Extensible Markup Language (XML) files, 9, 198–201
- external stoppage of jobs, 155–159
 - using CommandLineJobRunner runner, 157
 - via Admin tool, 155–157

■ F

- FAILURE path, 89
- Fee Per Transaction, 37
- FeesItemProcessor, 372–373
- FieldExtractor interface, 263
- FieldSetMapper, 344
- FieldSets, 343
- file-based ItemWriters, 261–278
 - FlatFileItemWriter, 261–273
 - delimited files, 269–270
 - file creation options for, 270–273
 - formatted text files for, 263–269
 - StaxEventItemWriter, 273–278

- files, input, 166–201
 - flat files, 166–198
 - XML files, 198–201
- FileSystemResource, 346
- FILTER_COUNT field, 27
- filtering items, 257–258
- findCustomerByTaxId method, 349
- First Name field, 168
- fixed-width files, 168–173
- flat files, 166–198
 - delimited, 173–179
 - fixed-width, 168–173
 - multiple sources, 193–198
 - records
 - custom parsing, 179–181
 - multiline, 188–193
 - multiple formats, 182–188
- FlatFileHeaderCallback, 381–382
- FlatFileItemReader, 145, 264, 339, 362, 374
- FlatFileItemWriter, 261–273
 - delimited files, 269–270
 - file creation options for, 270–273
 - formatted text files for, 263–269
- flatFileOutputWriter, 274
- flexibility, 4, 13
- flow attribute, 91
- flow tag, 89, 91
- flows. *See also* jobs
 - externalizing, 89–92
 - parallelization of, 92–93
 - of steps, 83–93
 - conditional logic, 83–86
 - ending job, 86–89
- footballJob, 17
- footerCallback option, 262, 273
- formatJob, 271, 283, 286–287
- formatStep, 290
- formatted text files, for FlatFileItemWriter, 263–269
- FormatterLineAggregator, 266, 269
- framework, Spring Batch, 19–20
 - configuring Maven repository, 20
 - downloading zip distribution, 19
 - from Github repository, 19–20
 - from SpringSource tool suite, 19

■ G

- generateAccountSummaryStep, 146
- generateMonthlyStatement, 379

GET request, 357
 getAccountExecutiveForCustomer method, 247
 getCustomer() method, 220–221
 getFirstName() method, 172
 getJobParameters() function, 55
 getNextJobParameters method, 128–129
 getStocksForCustomer method, 377
 getSuffix, 314
 getTickersPaged method, 357
 getTotalValueForCustomer method, 377
 getValue method, 370
 Github repository, obtaining Spring Batch
 framework from, 19–20
 groovyJob, 17
 GUI-based programming, 3

H

Hadoop framework, 6
 handleChunk method, ChunkHandler object,
 422
 handling errors, 226–231
 no input, 230–231
 records, 226–230
 headerCallback option, 262, 273
 headerExtractor bean, 431, 433
 headerFooterSample, 17
 headers and footers, for
 MultiResourceItemWriter
 in flat file, 317–322
 XML fragments, 315–317
 Hello World! job, 20–22
 helloWorld bean, 74
 HelloWorld class, 55
 HelloWorld tasklet, 54, 73, 106, 129
 helloWorld.jar, 121
 HelloWorldJob, 73–74
 helloWorldJob, 107, 118
 helloWorldStep, 74
 helloWorld.xml file, 114, 122
 Hibernate project, 208–213
 cursor processing with, 209–213
 paged database access with, 213
 hibernate.cfg.xml file, 211, 214, 287, 411
 HibernateCursorItemReader object, 418
 HibernateItemWriter, 285–290, 412
 hibernateJob, 17
 hiddenHttpMethodFilter, 138
 HTTP GET request, 426
 HttpSession, 62

I

id attribute, split tag, 415
 Import Stock Prices, 37
 Import Transactions, 35
 importCustomerAndTransaction, 353
 importCustomerAndTransactionData bean,
 338
 importStockPrices, 362
 inbound-requests channel, 442
 inbound-staging channel, 442
 incoming channel, 431, 433
 incrementing, parameters for jobs, 57–59
 infiniteLoopJob, 17
 infrastructure layer, Spring Batch, 8
 inheritance
 in jobs, 51–52
 in steps, 74–76
 in-memory, job repository, 99–100
 input
 custom, 221–226
 database, 201–218
 Hibernate project, 208–213
 JDBC API, 202–208
 JPA API, 214–218
 flat files, 166–198
 delimited, 173–179
 fixed-width, 168–173
 multiple sources, 193–198
 records, 179–193
 handling no input errors, 230–231
 XML files, 198–201
 input and output options, 29
 input validation, 235–242
 input/output (I/O), standardizing, 9
 insert statement, 354
 integration tests, 456–470
 defined, 456
 with Spring, 456–461
 configuring testing environment, 457–
 459
 writing integration test, 459–461
 for Spring Batch, 461–470
 jobs, 469–470
 steps, 466–469
 step-scoped Beans, 461–465
 interruptOnCancel attribute, 71
 Introduction to ItemWriters, 259
 invalid records, logging, 228–230
 I/O (input/output), standardizing, 9

- ioSampleJob, 18
- isComplete methods, 80
- isolation-level-for-create attribute, 103
- item processing, vs. chunk processing, 65–66
- Item-based processing, 65
- itemCountLimitPerResource, 384
- ItemPreparedStatementSetter interface, 280, 285, 305, 325
- ItemProcessor interface, 233–258
 - CompositeItemProcessor implementation, 248–256
 - custom, 256–258
 - introduction to, 233–234
 - ItemProcessorAdapter implementation, 243–248
 - for statement job example, 346–353
 - ValidatingItemProcessor implementation, 235–243
 - input validation, 235–242
 - subclassing, 242–243
- ItemProcessorAdapter implementation, of
 - ItemProcessor interface, 243–248
- ItemReader interface, 68, 76, 93, 165–166
- ItemReader object, 400, 412
- itemSqlParameterSourceProvider option, 280, 284
- ItemStream interface,
 - ClassifierCompositeItemWriter, 329–330
- ItemStreamException, 271
- ItemWriteListener object, 401–402
- ItemWriter object, 412
- ItemWriterAdapter, 294–297
- ItemWriters, 259–330
 - alternative output destination, 294–310
 - ItemWriterAdapter, 294–296
 - JmsItemWriter, 298–302
 - PropertyExtractingDelegatingItemWriter, 296–298
 - SimpleMailMessageItemWriter, 302–310
 - database-based, 278–293
 - HibernateItemWriter, 285–290
 - JdbcBatchItemWriter, 278–285
 - JpaItemWriter, 290–293
 - file-based, 261–278
 - FlatFileItemWriter, 261–273
 - StaxEventItemWriter, 273–278
 - multipart, 310–330
 - ClassifierCompositeItemWriter, 326–330
 - CompositeItemWriter, 322–326
 - MultiResourceItemWriter, 311–322
- overview, 260

■ J

- Java database connectivity API, processing. *See* JDBC API, processing
- java -jar metadata-0.0.1-SNAPSHOT.jar, 114
- Java language, batch processing in, 4–5
- Java Persistence Architecture (JPA) API, 214–218
- java.io.FileNotFoundException, 227
- java.lang.Long restart(long executionId) method, 109
- java.lang.Long start(java.lang.String jobName, java.lang.String parameters) method, 109
- java.lang.Long startNextInstance(java.lang.String jobName) method, 109
- java.lang.NullPointerException, 97
- java.lang.Runnable, 68
- java.lang.String getParameters(long executionId) method, 109
- java.lang.String getSummary(long executionId) method, 109
- java.util.Date, 53, 149, 176
- java.util.List<java.lang.Long> getExecutions(long instanceId) method, 109
- java.util.List<java.lang.Long> getJobInstances(java.lang.String jobName, int start, int count) method, 109
- java.util.List<java.lang.String> getJobNames() method, 105
- java.util.List<JobExecution> getJobExecutions(JobInstance instance) method, 105
- java.util.List<JobInstance> getJobInstances(java.lang.String jobName, int start, int count) method, 105
- java.util.Map<java.lang.Long, java.lang.String> getStepExecutionSummaries(long executionId) method, 109
- java.util.MapString, JobParameter object, 53
- java.util.Set<java.lang.Long> getRunningExecutions(java.lang.String jobName) method, 109
- java.util.Set<java.lang.String> getJobNames() method, 109
- java.util.Set<JobExecution> findRunningjobExecutions(java.lang.String jobName) method, 105
- javax.validation.Validator, 237
- JConsole main screen, 115
- JDBC (Java database connectivity) API, processing, 202–208

- cursor, 202–206
 - paged, 206–208
- JdbcBatchItemWriter, 278–286, 290, 354–355, 363, 369, 375
- jdbcBatchWriter, 282, 285
- JdbcCursorItemReader option, 312, 365
- JdbcItemWriter, 41
- JdbcTemplate, 284
- JDK version, and VisualVM, 388
- JMockit, 452
- JMS outbound channel adapter, 431
- JMS processing, 7
- JMS queues, 424, 433
- jmsFormatJob, 299
- JMSItemReader object, 412
- JmsItemWriter, 298–302
- JmsTemplate, 299, 301–302
- JMXJobLauncher, 113, 126
- JMXJobRunner class, 110–111, 117
- JMXJobRunner.runJob method, 117–118
- job components, 14
- Job Execution Context table, 27–28
- job repository, 99–104
 - database, 101–104
 - schema configuration for, 102
 - transaction configuration for, 103–104
 - in-memory, 99–100
- job results, 11
- job tag, 74
- JOB_EXECUTION_ID / STEP_EXECUTION_ID field, 28
- JOB_EXECUTION_ID field, 26–27
- JOB_INSTANCE_ID field, 25–26
- JOB_KEY field, 25
- JOB_NAME field, 25
- jobClass, 131
- job-context.xml file, 457
- job.customerFile, 92
- jobDataAsMap, 132
- JobDataMap, 128
- JobDetailBean, 125, 131
- JobDetails object, 125
- JobExecution getJobExecution(java.lang.Long executionId) method, 105
- JobExecution object, 48, 52, 59, 62, 84, 469
- JobExecutionContext, 91
- JobExecutionDecider, 84–85
- JobExecutionListener interface, 60
- jobExplorer bean, 335
- JobExplorer interface, 104–108
- JobExplorer object, 442
- jobIdentifier parameter, 121
- JobInstance, 48, 86, 103
- JobInstance getJobInstance(java.lang.Long instanceId) method, 105
- JobInterruptedException, 154, 156
- JobLauncher bean, 47, 52, 120, 335
- JobLauncherTestUtils class, 467, 469
- JobLocator, 128
- JobOperator interface, 108–118
- JobOperator.start method, 111
- JobParameter class, 53, 55
- JobParameters class, 54
- JobParameters code, 55
- JobParameters instance, 54
- JobParameters object, 120–121, 128, 467, 469
- jobParametersExtractor bean, 92
- JobParametersIncrementers, 122, 140–142
- JobParametersInvalidException, 56–57
- jobPath parameter, 121
- JobRegistryBackgroundJobRunner runner, 47, 120, 122–124, 130, 132
- JobRepository bean, 335
- JobRepository datastore, 23–28
 - configuration of, 23–24
 - tables in, 24–28
 - BATCH_JOB_EXECUTION and BATCH_STEP_EXECUTION, 26–27
 - BATCH_JOB_INSTANCE, 25
 - BATCH_JOB_PARAMS, 25–26
 - Job and Step Execution Context, 27–28
- job-repository tag, 101–103
- JobRepositoryFactoryBean, 102
- JobRunner interface, 47
- jobs, 45–64, 119–163. *See also* sample jobs
 - configuring, 49–51
 - defining, 8
 - description of, 40–42
 - calculating transaction fees, 41
 - generating customer monthly statements, 41–42
 - importing customer transaction data, 40
 - stock closing prices, 40–41
 - ending, 86–89
 - execution of, 13–14
 - ExecutionContext state of
 - manipulating, 62–64
 - overview, 61–62
 - persistence of, 64
 - Hello, World!, 20–22
 - inheritance in, 51–52
 - integration tests for, 469–470

jobs (*cont.*)

- lifecycle of, 47–48
- listeners for, 59–61
- managing, 8
- metadata for, 104–118
 - using JobExplorer interface, 104–108
 - using JobOperator interface, 108–118
- parameters for, 52–59
 - incrementing, 57–59
 - validating, 56–57
- restarting, 160–163
 - configuring number of restarts, 161–162
 - preventing, 160
 - rerunning complete steps, 162–163
- running, 22–23
- scaling, 399–445
 - multithreaded steps, 399–404
 - parallel steps, 404–421
 - partitioning, 435–445
 - remote chunking, 421–435
- starting, 119–141
 - job execution, 119–120
 - job runners, 120–124
 - third-party integration, 124–141
- statement, requirements of, 33–39
 - and steps, 12–13
- stopping, 142–159
 - error handling, 157–159
 - external stoppage, 155–159
 - natural ending, 142
 - programmatically ending, 142–155
- jobSampleJob, 18
- jobs/helloWorldJob.xml file, 113
- job.stockFile, 92
- JPA (Java Persistence Architecture) API, 214–218
- jpaDialect, 292
- JpaItemWriter, 290–293
- jpaVendorAdapter, 292
- JUnit, 447–451
- JVM
 - multiple, 434
 - single, 421
 - slaves, 434

■ **K**

- KEY_NAME field, 26

■ **L**

- Last Name field, 168
- LAST_UPDATED field, 26–27
- latFileItemReader, 307
- launch-context.xml file, 49–50, 100, 105, 131, 211, 254, 291, 408, 445
- launchJob() method, JobLauncherTestUtils class, 469
- lifecycle of jobs, 47–48
- LineAggregator interface, 261, 265, 269, 381
- LineMapper interface, 167, 261, 263, 344
- lineSeparator option, 262
- linesToSkip, 167
- LineTokenizer, 344
- listener profile, 429
- listenerContainer bean, 431
- listenerJob.xml, 61
- listeners
 - for jobs, 59–61
 - for steps, 81–83
- local parallelization, 9
- logAddress method, 296–297
- logging invalid records, 228–230
- LoggingStepStartStopListener, 82
- LONG_VAL field, 26
- loopFlowSample, 18

■ **M**

- mailJob, 18
- main method, 110
- mainframe, Spring Batch, 5
- maintainability, 3–4, 13
- Map<String, Object>, 55, 63
- mapFieldSet, 175
- MapJobRegistry, 113
- mapLine method, 171, 344
- marshaller, 273, 276
- master job step, 436
- maven command, 115
- Maven repository, configuring, 20
- max-varchar-length, 102
- MBean browser, VisualVM, 392
- memory profiling, for Batch applications, 397–399
- merge attribute, 61
- MessageChannelPartitionHandler object, 438, 441–442

messageGateway bean, 431, 433
 MessageTemplate object, 433, 442
 metadata for jobs, 104–118
 using JobExplorer interface, 104–108
 using JobOperator interface, 108–118
 MetaDataInstanceFactory class, 464
 MethodInvokingTaskletAdapter, 69–70
 methods, in test cases, 448
 Middle Initial field, 168
 mock objects
 class remapping approach, 451
 defined, 451
 defining return values, 456
 proxy based approach, 451
 proxy object, 451
 returning values, 455
 and Stubs, 451
 for unit tests, 451–452
 Mockito, 447–448, 452–456
 Moore's Law, 6
 multicore processors, 6
 multiline records, 188–193
 multilineJob, 18
 multilineOrder, 18
 multipart ItemWriters, 310–330
 ClassifierCompositeItemWriter, 326, 329–330
 CompositeItemWriter, 322–326
 MultiResourceItemWriter, 311–322
 header and footer records in flat file for, 317–322
 header and footer XML fragments for, 315–317
 MultiResourceItemWriter, 311–322
 header and footer records in flat file for, 317–322
 header and footer XML fragments for, 315–317
 multithreaded steps, 15, 399–404
 mvn clean compile, 22, 24
 mvn clean install command, 138, 293, 295, 309, 355, 430, 445
 mvn clean install -P listener command, 430, 445
 myJobRunner, 116
 MySQL driver, 333

■ N

name parameter, 54, 64
 New York Stock Exchange (NYSE), 356

next() method, 202
 next attribute, 83, 415
 next parameter, 121
 next tag, 83
 NullPointerException, 97, 157
 NYSE (New York Stock Exchange), 356

■ O

obExecution, 48
 obRepository, 88
 open method, 97, 223, 225
 optionalKeys, 57
 optionalParamValidator, 57
 org.apache.commons.dbcp.BasicDataSource, 101
 org.codehaus.classworlds.Launcher "exec, java", 115
 org.hibernate.cfg.AnnotationConfiguration class, 211
 org.junit.Assert class, 449
 org.springframework.retry.RetryPolicy, 96
 org.springframework.batch.item.file.LineMapper interface, 166
 org.springframework.batch.core.BatchStatus, 84
 org.springframework.batch.core.ExitStatus, 70, 84
 org.springframework.batch.core.explore, 104
 org.springframework.batch.core.Job interface, 12
 org.springframework.batch.core.JobExecutionListener interface, 59
 org.springframework.batch.core.job.flow.JobExecutionDecider, 84
 org.springframework.batch.core.JobInstance, 48
 org.springframework.batch.core.JobParameters Builder, 128
 org.springframework.batch.core.JobParameters Validator interface, 56
 org.springframework.batch.core.launch.JobLauncher interface, 47, 119
 org.springframework.batch.core.launch.JobOperator interface, 108
 org.springframework.batch.core.launch.support package, 47
 org.springframework.batch.core.launch.support.CommandLineJobRunner, 120

- org.springframework.batch.core.launch.support.JobRegistryBackgroundJobRunner, 120
- org.springframework.batch.core.scope.context.StepContext, 54
- org.springframework.batch.core.StartLimitExceededException, 161
- org.springframework.batch.core.Step, 13
- org.springframework.batch.core.StepContribution, 54
- org.springframework.batch.core.step.item.ChunkOrientedTasklet<I>, 68
- org.springframework.batch.core.step.skip.SkipPolicy, 227
- org.springframework.batch.core.step.tasklet.Tasklet interface, 68
- org.springframework.batch.item.adapter.ItemReaderAdapter, 218
- org.springframework.batch.item.database.HibernateCursorItemReader, 212
- org.springframework.batch.item.database.ItemPreparedStatementSetter interface, 280
- org.springframework.batch.item.database.ItemSqlParameterSourceProvider interface, 285
- org.springframework.batch.item.database.JdbcCursorItemReader, 204
- org.springframework.batch.item.database.JpaPagingItemReader, 216
- org.springframework.batch.item.database.orm.AbstractJpaQueryProvider, 217
- org.springframework.batch.item.database.PagingQueryProvider, 207
- org.springframework.batch.item.file.DefaultLineMapper, 170
- org.springframework.batch.item.file.FieldSet, 168
- org.springframework.batch.item.file.FlatFileFooterCallback interface, 317
- org.springframework.batch.item.file.FlatFileItemReader consists, 166
- org.springframework.batch.item.file.mapping.BeanWrapperFieldSetMapper, 171
- org.springframework.batch.item.file.mapping.FieldSetMapper interface, 175
- org.springframework.batch.item.file.mapping.PatternMatchingCompositeLineMapper, 182
- org.springframework.batch.item.file.ResourceAwareItemWriterItemStream, 318
- org.springframework.batch.item.file.transform.DelimitedLineTokenizer, 174
- org.springframework.batch.item.file.transform.FieldExtractor interface, 266
- org.springframework.batch.item.file.transform.FixedLengthTokenizer, 171
- org.springframework.batch.item.file.transform.LineTokenizer interface, 179
- org.springframework.batch.item.ItemProcessor interface, 234
- org.springframework.batch.item.ItemProcessor<T>, 13
- org.springframework.batch.item.ItemReader<T>, 13
- org.springframework.batch.item.ItemReader<T> interface, 165
- org.springframework.batch.item.ItemStreamReader, 223
- org.springframework.batch.item.ItemWriter<T>, 13
- org.springframework.batch.item.jms.JmsItemWriter, 299
- org.springframework.batch.item.ParseException, 158, 226–227
- org.springframework.batch.item.support.CompositeItemProcessor, 248
- org.springframework.batch.item.validator.Validator interface, 237
- org.springframework.batch.item.xml.StaxEventItemReader, 200
- org.springframework.batch.item.xml.StaxEventItemWriter, 273
- org.springframework.batch.repeat.CompletionPolicy interface, 78
- org.springframework.batch.repeat.policy.SimpleCompletionPolicy, 78
- org.springframework.batch.repeat.RepeatStatus object, 73
- org.springframework.batch.retry.interceptor.RetryOperationsInterceptor, 95
- org.springframework.batch.retry.policy.TimeoutRetryPolicy, 95
- org.springframework.batch.retry.RetryCallback interface, 94
- org.springframework.batch.retry.RetryListener, 97
- org.springframework.batch.retry.RetryPolicy interface, 94
- org.springframework.batch.retry.RetryTemplate, 94
- org.springframework.beans.factory.InitializingBean interface, 238
- org.springframework.core.io.FileSystemResource beans, 169

- org.springframework.core.task.TaskExecutor, 399
- org.springframework.jdbc.core.PreparedStatementSetter, 205
- org.springframework.jdbc.support.JdbcTemplate, 278
- org.springframework.orm.jpa.JpaTransactionManager, 214
- org.springframework.xml.Marshaller, 274
- org.springframework.xml.Unmarshaller, 200
- org.springframework.xml.xstream.XStreamMarshaller, 200
- org.springframework.xml.xstream.XStreamMarshaller class, 276
- org.springframework.scheduling.quartz.CronTriggerBean, 132
- org.springframework.web.context.ContextLoaderListener, 133
- org.springframework.web.servlet.DispatcherServlet, 138
- org.springframework.core.task.SyncTaskExecutor, 47
- outbound-replies channel, 442
- outbound-requests channel, 442
- outbound-staging channel, 442
- OutOfMemoryException, 398
- outputFile, 255, 313–314
- outputWriter, 68, 255
- overwriteOutput, 273

■ P

- paged database access, with Hibernate project, 213
- paged processing, 206–208
- parallel processing, 6, 418
- parallel steps, 404–421
 - building picklists, 418–421
 - configuring, 415–418
 - loading orders into database, 412–415
 - preloading data for processing, 406–412
- parallelization, 14–16
 - local and remote, 9
 - multithreaded steps, 15
 - parallel steps, 15
 - partitioning, 16
 - remote chunking, 16
- parallelJob, 18
- parameters for jobs, 52–59
 - incrementing, 57–59

- validating, 56–57
- ParseException, 159, 345
- parsing, custom, 179–181
- partition method, 436
- partition step, 438
- Partitioner interface, 436, 441
- partitionFileJob, 18
- PartitionHandler object, 441
- partitioning, 16, 435–445
- partitionJdbcJob, 18
- PassThroughLineAggregator, 361
- PatternMatchingCompositeLineMapper, 343
- patternTokenizers Map, 344
- peak method, 190
- persistence, of ExecutionContext state of jobs, 64
- persistence.xml file, 214, 216, 290–291
- picklists, building, 418–421
- PlaceholderProperties bean, 336
- plain-text brokerage statement, 34
- POJO, 433, 435, 442
- POM file, 286, 333, 408, 423, 429
- pom.xml file, 125, 209
- PreparedStatement, 278, 280, 355
- preProcessingJob, 92
- pricing tiers, in statement job example, 364–370
 - calculating, 367–369
 - reading number of transactions for customer, 365–367
 - updating database with, 369–370
- PricingTier enum, 367–368, 370
- PricingTierItemProcessor, 368
- pricingTiersItemProcessor, 373
- Print Account Summary, 38
- process method, 247, 348
- PROCESS_SKIP_COUNT field, 27
- processing
 - cursor, 202–206, 209–213
 - paged, 206–208
- Product object, 248
- profiling
 - Batch applications, 387–399
 - CPU profiling, 393–396
 - memory profiling, 397–399
 - and VisualVM, 388–392
 - filtering packages, 396
- programmatic ending, of jobs, 142–155
 - StepExecution method, 152–155
 - stop Tag, 142–152
- <project_home>/src/main/java/example file, 333

- <project_home>/src/main/java/test file, 333
- <project_home>/src/main/resources directory, 130
- <project_home>/src/main/resources/jobs file, 333
- <project_home>/src/main/resources/launch-context.xml, 122
- <project_home>/src/main/resources/META-INF/spring file, 333
- <project_home>/src/main/webapp/WEB-INF directory, 133
- <project_home>/src/test/java/example file, 333
- <project_home>/src/test/resources/test-context.xml file, 333
- <project_home>/target directory, 121–122
- <PROJECT_ROOT>vsrsrc/main/resources directory, 138
- projects, 19–20
- PropertyExtractingDelegatingItemWriter, 296–298
- PropertyPlaceholderConfigurer, 50
- proxy objects, 451
- pseudostep, 415

■ Q

- Quartz scheduler, scheduling jobs with, 124–132
- QuartzJobBean, 125, 128
- queryName option, 213
- queryProvider option, 213
- queryString option, 213

■ R

- RandomDecider, 85
- read() method, 189–190, 204–205, 221
- READ_COUNT field, 27
- READ_SKIP_COUNT field, 27
- readDate method, 186
- readDouble method, 186
- readers, 165–231
 - custom input, 221–226
 - database input, 201–218
 - Hibernate project, 208–213
 - JDBC API, 202–208
 - JPA API, 214–218
 - error handling, 226–231
 - no input, 230–231

- records, 226–230
 - existing services, 218–220
 - file input, 166–201
 - flat, 166–198
 - XML, 198–201
 - ItemReader interface, 165–166
- records
 - custom parsing, 179–181
 - logging invalid, 228–230
 - multiline, 188–193
 - multiple formats, 182–188
 - skipping, 226–228
- recordSeparatorPolicy, 167
- RegularExpressionLineMapper, 343–344, 346
- reliability, 13
- remote chunking, 16
 - jobs, 421–435
 - and partitioning, 435
 - scaling, 421
 - sending data, 436
- remote parallelization, 9
- RemoteAccessException, 93–94
- RemoteChunkHandler object, 433
- RemoteChunkHandlerFactoryBean object, 433
- remoteStep, 93
- reparedStatement, 280
- RepeatContext, 80
- RepeatStatus object, 68–69, 73
- RepeatStatus.CONTINUABLE, 73
- RepeatStatus.FINISHED, 73, 130
- replies channel, 431
- reply queue, 438
- request queue, 438
- request/response processing, 45
- requests channel, 431, 433
- requiredKeys, 57
- requiredParamValidator, 57
- requirement, in test cases, 449
- resource option, 262, 273, 312
- ResourceAwareItemWriterItemStream interface, 318
- resourceSuffixCreator option, 312, 314
- restart parameter, 121
- restartable attribute, 160
- restarting jobs, 160–163
 - configuring number of restarts, 161–162
 - preventing, 160
 - rerunning complete steps, 162–163
- restartSampleJob, 19
- retrieveStockPrices, 360
- RetryCallback interface, 94

- RetryListener, 97
- RetryOperationsInterceptor, 96
- RetryPolicy, 94–96
- retrySample, 18
- RetryTemplate, 94–95
- ROLLBACK_COUNT field, 27
- rootElementAttributes, 274
- rootTagName, 274
- RowMapper, 305, 365, 370
- run.id parameter value, 106
- RunIdIncrementer, 106, 130
- Runnable interface, 68
- runners, job, 120–124
 - CommandLineJobRunner runner, 120–122
 - JobRegistryBackgroundJobRunner runner, 122–124

S

- sample jobs, 29–43. *See also* jobs
 - agile development, 29–32
 - source-control systems, 32
 - TDD, 32
 - true development environments, 32
 - user stories, 30–32
 - designing, 39–43
 - data model, 42–43
 - job description, 40–42
 - requirements of statement job, 33–39
- sampleJob, 51, 64
- saveState option, 262, 274, 312
- scalability, 3, 13, 29
- scaling, 387–445
 - batch processes, 4
 - jobs, 399–445
 - multithreaded steps, 399–404
 - parallel steps, 404–421
 - partitioning, 435–445
 - remote chunking, 421–435
 - profiling Batch applications, 387–399
 - CPU profiling, 393–396
 - memory profiling, 397–399
 - and VisualVM, 388–392
- SchedulerFactory, 125
- security, 3
- SERIALIZABLE, 103
- Serializable interface, 424
- SERIALIZED_CONTEXT field, 28, 64
- SessionFactory, 286–290

- setEntityManager(EntityManager em) method, 217
- setFirstName(String name), 172
- setResource method, 195
- SHORT_CONTEXT field, 28
- shouldDeleteIfEmpty option, 262, 271
- shouldDeleteIfExists flag, 272
- shouldDeleteIfExists option, 262
- shouldSkip, 227
- SimpleAsyncTaskExecutor object, 417
- simple-cli-archetype, 113
- SimpleCompletionPolicy, 80
- simpleJdbcTemplate option, 280
- SimpleJobLauncher, 119–120
- SimpleJobOperator, 113
- SimpleMailMessageItemWriter, 302–310
- SimpleMailMessages, 306
- skip-limit attribute, 97
- skippedLinesCallback, 167
- skipping records, 226–228
- skipSampleJob, 18
- slave nodes, 435
- slave workers, 435
- source-control systems, 32
- split tag, 415
- Spring Batch
 - @Autowired annotation, 459
 - @ContextConfiguration annotation, 459, 462
 - @RunWith annotation, 462
 - @TestExecutionListeners annotation, 462
 - Admin application, 16
 - Admin project, 9
 - architecture of, 11–18
 - documentation, 17–18
 - job execution, 13–14
 - jobs and steps, 12–13
 - parallelization, 14–16
 - Spring Batch Admin application, 16
 - batch processing with
 - framework, 7–10
 - other uses for, 6–7
 - Hello, World! job, 20–22
 - Integration project, 421–422
 - integration tests for, 461–470
 - jobs, 469–470
 - steps, 466–469
 - step-scoped Beans, 461–465

- Spring Batch (*cont.*)
 - integration tests with, 456–461
 - configuring testing environment, 457–459
 - writing integration test, 459–461
 - JobRepository datastore, 23–28
 - configuration of, 23–24
 - tables in, 24–28
 - obtaining framework, 19–20
 - configuring Maven repository, 20
 - downloading zip distribution, 19
 - from Github repository, 19–20
 - from SpringSource tool suite, 19
 - project setup, 19–20
 - running jobs, 22–23
- SpringBatchQuartzJobLauncher class, 126, 128–129, 131–132
- spring-batch-simple-cli-2.1.3.RELEASE.jar, 121
- spring-context-support, 125
- .springframework.batch.item.file.transform.LineAggregator interface, 261
- .springframework.batch.item.ItemStreamException, 271
- spring.framework.version property, 333
- Spring's
 - org.springframework.jdbc.core.JdbcTemplate, 150
- Spring's SyncTaskExecutor, 120
- SpringSource tool suite, obtaining Spring Batch framework from, 19
- SpringValidator class, 238
- sql option, 280
- src/main/resources directory, 49, 458
- src/test/resources directory, 458
- src/test/resources/data directory, 462
- staging
 - defined, 400
 - implemented, 401
 - managing flat files, 404
 - queue, 438
- start method, 80
- START_TIME field, 26–27
- State field, 169
- Statement class, 375–376
- Statement Header, 39
- statement job example, 331–385
 - customer data in, 336–355
 - creating transaction reader, 338–346
 - ItemProcessor for, 346–353
 - writing data, 353–355
 - generating monthly statement, 375–385
 - reading statement data, 375–380
 - writing statements, 381–385
 - new project for, 332–336
 - overview, 331–332
 - pricing tiers in, 364–370
 - calculating, 367–369
 - reading number of transactions for customer, 365–367
 - updating database with, 369–370
 - requirements of, 33–39
 - stock prices for, 355–364
 - reading stock file, 362–363
 - reading tickers, 356–360
 - writing stock file, 360–361
 - writing to database, 363–364
 - transaction fees in, 370–375
 - calculating prices, 372–374
 - saving to database, 374–375
- Statement writers, 259
- StatementFormatter, 381, 383–384
- StatementHeaderCallback, 383
- statementJob file, 338, 353, 457
- statements, generating monthly for customer, 41–42
- statement/src/main/resources/batch.properties file, 334
- statement/src/main/resources/jobs directory, 338
- statement/src/main/resources/launch-context.xml file, 334
- StatementSuffixGenerator, 383
- statementSuffixGenerator, 384
- STATUS field, 26–27
- StaxEventItemReader, 273
- StaxEventItemWriter option, 273–278, 312, 316
- StaxWriterCallback interface, 315
- step, defined, 466
- step element, 74
- Step Execution Context table, 27–28
- STEP_EXECUTION_ID field, 27
- STEP_NAME field, 27
- StepContext.getJobExecutionContext() method, 63
- StepExecution method, 62, 84, 105, 152–155
- StepExecution object, 435–436, 438, 442, 464
- StepExecutionContext, 91
- StepExecutionRequestHandler object, 442
- StepExecution.setTerminateOnly() method, 152
- StepListener object, 145, 400–402
- StepLocator object, 442

- steps, 65–98. *See also* flows; jobs
 - chunk vs. item processing of, 65–66
 - chunk-size configuration for, 76–81
 - configuring, 66–68
 - controlling flow of, 83–93
 - conditional logic, 83–86
 - ending job, 86–89
 - externalizing flows, 89–92
 - parallelization of flows, 92–93
 - error handling for, 93–98
 - item retry, 93–97
 - item skip, 97
 - inheritance in, 74–76
 - integration tests for, 466–469
 - jobs and, 12–13
 - listeners for, 81–83
 - multithreaded, 15
 - parallel, 15
 - rerunning complete, 162–163
 - Tasklet interface for, 68–83
 - CallableTaskletAdapter, 68–69
 - creating tasklet steps, 73–74
 - MethodInvokingTaskletAdapter, 69–70
 - SystemCommandTasklet, 71–72
- step-scoped Beans, integration tests for, 461–465
- StepScopeTestExecutionListener class, 461
- StepScopeTestUtils object, 465
- Stock Closing Price Input File, 40
- stock prices
 - importing into database, 41
 - retrieving, 40
 - for statement job example, 355, 361–364
 - reading stock file, 362–363
 - reading tickers, 356–360
 - writing stock file, 360–361
 - writing to database, 363–364
- stockFileWriter, 361
- Stocks Closing Price, 36
- stop parameter, 121
- <stop> tag, 142–152
- stoppage, external, 155–159
- STOPPED ExitStatus, 159
- Street field, 168
- String arrays, 57
- STRING_VAL field, 26
- String.format() method, 266
- Stubs, and mock objects, 451
- SubclassClassifier, 353

- SystemCommandTasklet, 71–72
- System.out.println(HELLO_WORLD), 26
- systemProcessExitCodeMapper, 72

T

- table-prefix, 102
- tables, in JobRepository datastore, 24–28
 - BATCH_JOB_EXECUTION and BATCH_STEP_EXECUTION tables, 26–27
 - BATCH_JOB_INSTANCE table, 25
 - BATCH_JOB_PARAMS table, 25–26
 - Job and Step Execution Context tables, 27–28
- targetMethod, 294
- targetObject, 294
- TaskExecutor, 47, 72, 120, 400, 415
- task-executor attribute, 415
- taskExecutor property, 120
- Tasklet interface, for steps, 68–83
 - CallableTaskletAdapter, 68–69
 - creating tasklet steps, 73–74
 - MethodInvokingTaskletAdapter, 69–70
 - SystemCommandTasklet, 71–72
- tasklet step, 438
- taskletJob, 18
- TDD (test-driven development), 32
- temCountLimitPerResource option, 311
- terminationCheckInterval, 72
- test cases
 - @After annotation, 449
 - @Before annotation, 449
 - @BeforeClass annotation, 451
 - @Ignore annotation, 451
 - @RunWith annotation, 451
 - @Test annotation, 449
 - assert methods, 449
 - common practices, 448
 - defined, 448
 - flow, 449
 - methods, 448
 - requirement, 449
 - test method, 449
- test configuration, database, 457
- test method, in test cases, 449
- test-batch.properties file, 459
- test-context.xml file, 458, 468
- test-driven development (TDD), 32
- TestExecutionListener, 461

testing, 447
 integration tests, 456–470
 with Spring, 456–461
 for Spring Batch, 461–470
 Spring Batch component
 listener approach, 464
 wrapping execution in a step, 464
 unit tests, 447–456
 with JUnit, 448–451
 mock objects for, 451–452
 with Mockito, 452–456
 third-party integration, 124–141
 launching jobs with Admin tool, 135–141
 running jobs in containers, 132–135
 scheduling jobs with Quartz scheduler, 124–132
 Thread Inspector plugin, VisualVM, 392
 threadpool, 400
 ThreadPoolTaskExecutor, 120
 Ticker class, 362
 ticker DAO, 351
 TickerDaoJdbc, 352, 356, 377, 379
 tickerUpdateItemWriter, 363
 tiers, transaction fee, 41
 tiersUpdateWriter, 369
 tier.value parameter, 370
 TimeoutTerminationPolicy, 78
 timestamp, 149
 tokenize method, 180
 <TOMCAT_HOME>/bin, 135
 <TOMCAT_HOME>/logs/catalina.out file, 135
 <TOMCAT_HOME>/webapps, 135
 toString() method, 192, 205, 261
 tradeJob, 18
 Transaction class, 211, 339–340
 transaction fees, in statement job example, 370–375
 calculating prices, 372–374
 saving to database, 374–375
 Transaction objects, 182, 185, 188
 transaction record fields, 337
 transactional option, 262, 274
 TransactionApplierProcessor, 146, 149
 TransactionDaoImpl, 149–150
 TransactionFieldSetMapper, 341
 transactionImportWriter, 354–355
 transactionJob, 145, 152, 155, 159–162
 transactionManager, 50–51, 101, 411
 TransactionManager bean, 336

transactionPricingItemReader, 372, 374
 transactions
 calculating fees, 41
 importing customer data, 40
 translateParams, 128
 truckStep, 75
 TYPE_CD field, 26

■ U

Unique, 46
 unit tests, 447–456
 defined, 447
 goal, 448
 with JUnit, 448–451
 mock objects for, 451–452
 with Mockito, 452–456
 UNIVAC, 2
 update method, 80, 223, 225
 update statement, 354
 UriReader class, 358, 360
 user stories, 30–32

■ V

validating parameters, for jobs, 56–57
 ValidatingItemProcessor implementation, of
 ItemProcessor interface, 235–243
 input validation, 235–242
 subclassing ValidatingItemProcessor
 implementation, 242–243
 ValidatingItemReader class, 234
 ValidationException, 238
 varchar columns, 102
 varchar data type, 102
 vehicleStep, 74
 VERSION field, 25–27
 Visual GC plugin, VisualVM, 392
 VisualVM
 capabilities, 388
 difference between Profiler and Sampler,
 392
 as Eclipse instance, 388
 and JKD version, 388
 Local section, 388
 MBean browser, 392
 Monitor tab, 389–390, 398
 Overview tab, 389
 Profiler tab, 389

- and profiling Batch applications, 388–392
- Remote section, 388
- Sampler tab, 389, 396
- snapshot, 399
- Snapshots, 388
- Thread Inspector plugin, 392
- Threads tab, 389–390
- URL download, 388
- Visual GC plugin, 392
- VM core dumps, 388

■ W

- webapps directory, 138
- wildcards, 343
- workingDirectory, 72
- write method, 315

- WRITE_COUNT field, 27
- WRITE_SKIP_COUNT field, 27

■ X, Y

- XML (Extensible Markup Language) files, 9, 198–201
- XMLEventFactory, 329
- xmlOutputWriter, 276, 313, 329

■ Z

- Zip Code field, 169
- zip distribution, downloading Spring Batch framework from, 19
- <ZIP_ROOT>/samples/spring-batch-simple-cli directory, 332