

## Symbols

@After 31, 38, 370, 415  
 @AfterClass 31, 38, 92, 416  
 @Before 31, 38, 122, 370, 415  
 @BeforeClass 31, 38, 92, 416  
 @DataSet 350, 398  
 @Dummy 394  
 @ExpectedDataSet 398–399  
 @ForeignKey 385  
 @GeneratedValue 373  
 @Id 373  
 @Ignore 47, 417  
 @InterceptorClasses 428  
 @Mock 393, 396  
 @Parameters 18, 192–193, 217–218, 420  
 @RunWith 18, 429, 431  
 @SuiteClasses 22  
 @Test 11, 15–16, 31, 34, 44, 122, 418  
 @TestDataSource 399  
 \$CC\_HOME content table 173  
 \$CC\_HOME start command 176

## A

AbstractDbUnitELTemplate-TestCases 354  
 AbstractDbUnitTemplate-TestCases 351  
 AbstractDbUnitTestCase 340, 344  
 AbstractJpaDbUnitTestCase 370

AbstractJpaTestCase 370, 387  
 Account.java 101  
 AccountService 100  
 AccountService.java 102  
 ActiveXObject, Ajax 230  
 add 6  
 addAccount 102  
 Administration application  
   Cactus 266  
   testing JSPs 273  
 Administrator application  
   architecture 277  
 AdminServlet, doGet 271  
 agile methodologies 55  
 Ajax 306  
   architecture 225  
   challenges 225  
   CSS 225  
   experimental 234  
   functional testing 227, 229–234  
   HttpClient 247  
   interaction 225  
   JDOM 249  
   NicelyResynchronizingAjax-Controller 234  
   synchronizing 233  
   testing 225–227  
   testing patterns 227–229  
   timeout 232  
   waitForBackgroundJavaScript 234  
 Ajax4jsf 306  
 AjaxController 233  
 album\_details.jsp 298, 305–306  
 Album.java 294

AlbumDetailsBean 297  
   JMock testing 302  
 AlbumManager 295  
 AlertHandler 204  
 alerts  
   asserting 203  
   asserting none 204  
   custom handler 205  
   JavaScript 203, 220  
 allOf matcher 49  
 annotations  
   custom 350  
   EL test 355  
   UserDaoJdbcImplTest 351  
 Ant 137, 391  
   build file 139–140  
   build script 236  
   default target 141  
   Eclipse integration 444  
   installing 139–140  
   Java apps, and 138  
   JUnit 242  
   NetBeans integration 451  
   property elements 140–143  
   running build file 144  
   running tests from 138–139  
   targets 140  
*Ant in Action* 141  
 anyOf matcher 49  
 Anything matcher 49  
 Apache Ant. *See* Ant  
 Apache Commons 250  
   HttpClient 247  
   IO 247  
 Apache Felix. *See* Felix  
 Apache Ivy. *See* Ivy

API contract 5  
 Application Controller 26  
 applications, multilayered 366  
 archives, Cactus and 280  
 artifactId 161  
 assert 15, 17  
   cumbrous 47  
   sample table 15  
   simplify declaration 48  
 Assert class 15  
   HtmlUnit 191  
 assert statements 38  
 assertEquals 16  
 assertEqualsNotEquals 15  
 assertEquals(List, Object) 402  
 assertEquals 11, 16, 38  
 assertGreater 402  
 assertions  
   BigDecimal 405  
   BufferedImages 405  
   clarity in testing 401  
   collections 405  
   custom extensions 407  
   entry point 405  
   errors 423  
   exceptions 405  
   files 405  
   Hamcrest 421  
   Mycila 407  
   new JUnit 422  
   RhinoUnit 235  
   testing 401–407  
   third-party 401  
   Unitils 403  
 assertNotEquals 401  
 assertNotNull 16  
 assertNotSame 15  
 assertNotTrue 15  
 assertReflectionEquals 404  
 assertSame 16, 38  
 assertStartsWith 403  
 assertThat 48, 405  
 assertTrue 16  
 assumptions 422  
 automatic test  
   creating 30, 40  
   suite 38

## B

---

backwards compatible 72  
 base class  
   implementation 28  
   table of components 30  
 batchtest 149

Bean Scripting Framework,  
   Ajax 234  
 BeanUtils 269  
 beginXXX 262  
 best practices 8  
   always specify a reason for  
   skipping a test 47  
   always verify that the test fails  
   when it should fail 268  
 assertions helper 336  
 choose meaningful test  
   method names 34  
 continuous regression  
   testing 83  
 create complementary  
   tests 357  
 database testing 356–358  
 database tests 334  
 don't hardcode values 342  
 don't write business logic in  
   mock objects 103  
 EasyMock object creation 119  
 explain the failure reason in  
   assert calls 36  
 JSLint 245  
 let the test improve the  
   code 43, 45  
 make exception tests easy to  
   read 45  
 make sure target database is  
   tested 356  
 one database per  
   developer 356  
 one unit test equals one @Test  
   method 38  
 plan your dataset usage 357  
 refactor 55  
 same package, separate  
   directories 51  
 test anything that could possi-  
   bly fail 40  
 test cleanup 358  
 test only what can possibly  
   break 104  
 throw an exception for meth-  
   ods that aren't  
   implemented 272  
 unit test one object at a  
   time 33  
 use TDD XP principles 269  
 when writing load test  
   cases 357  
 write failing tests first 80  
 black box testing 62  
 approaches 62

combining with white box  
   testing 71  
 coverage 68  
 JSF 301  
   vs. white box 62–63  
 bootstrappers 175  
 branches 45  
 browserbot 232  
 browsers 225  
   Ajax testing challenges 227  
 BSF. *See* Bean Scripting  
   Framework  
 build descriptor,  
   namespaces 161  
 build file 139–140  
   Ant 139  
   properties 141  
 build lifecycles 155  
 build phases 155  
 build.dir 154  
 build.xml 139  
   cactifyear 282  
   cactifywar 282  
   Cactus 280  
 buildresultsurl 178  
 BundleActivator 314  
 BundleContext 314  
 BundleID 316  
 business layer  
   interface 362  
   testing 361  
   unit testing 366

## C

---

cactification 280  
 cactifyear 282  
 cactifywar 281  
 cactifyXXX 281  
 parameters 282  
 Cactus  
   Administration  
     application 266  
   AdminServlet 266  
   Ant integration 280–284  
   browser integration 290  
   Cactus tests vs. JUnit tests 261  
   cactus-report.xml 290  
   callView 269, 271  
   Cargo and 279  
   client-side logic 263  
   configuring with  
     HtmlUnit 454  
   doGet 271  
   getCommand 267–268

- Cactus (*continued*)
    - HtmlPage, getting 208
    - HtmlUnit 206
    - HtmlUnit code
      - integration 223
      - introduction 260
    - isAuthenticated 261
    - Java component testing 260
    - JSP tests 273, 277
    - matching begin to end
      - method 207
    - Maven integration 285–289
    - plug-ins for Maven2 285
    - pom.xml 285
    - SampleServlet 262
    - servlet tests 266
    - ServletException 271
    - ServletTestCase 262, 275
    - ServletTestRunner 290
    - simple servlet 207
    - test case requirements 262
    - test lifecycle 263
    - TestAdminServlet 268
    - testCallView 270
    - testing 260–265
    - testing EJBs 277–279
    - testing servlets and
      - filters 265–273
    - unit testing JSPs 273
    - writing tests in 206
  - cactus task 283
  - Calculator 18
  - Calculator class 11, 312
  - calculator class creation 6–8
  - CalculatorBundleActivator 314
  - CalculatorService
    - interface 313
    - mock test 321
    - OSGi test 323
    - sample client app 316
  - CalculatorTest 22
    - test case 15
  - CalculatorTest program 7, 11
  - Cargo 279
    - Ant tasks 284
  - cargo task 284
  - central repository 154
  - CI. *See* continuous integration
  - Class Extension 119
  - class factory 111
  - clean 150
  - ClickableElement 200
  - ClientBundleActivator 316, 322
    - refactored 319
  - closeTo matcher 50
  - Cobertura 69–70
    - cobertura.ser 70
    - cobertura-report script 70
    - code
      - change resistant 104
      - flexibility 104–105
      - scaffolding 8
      - source code for book 439
      - testing 7
    - code coverage metrics 56
    - code smell 45
    - coding strategies 42
    - Collection 18
    - combining test methods 37–38
    - Common Public License 4
    - ComparableAssert 401
    - compile
      - Maven build phase 155
      - targets 142
    - compile.java 142
    - compile.test 142
    - Compiler plug-in 164
    - components 127
      - testing with Cactus 260
    - composition vs. inheritance 77
    - conditional statement 78
    - conditionals vs.
      - polymorphism 77
    - config.xml 174
    - connectfour 172
    - ConnectionFactory 112
    - console test runner 12
    - constructors
      - Car 74
      - simple, creating 73
      - working in 74
    - containers 127
      - continuous integration 279
      - executing test from 265
      - Java EE 279
      - management 279
      - transaction management 366
      - types of 127
    - containsString matcher 50
    - continuous integration 169–172
      - benefits 184–185
      - containers 279
      - definition 170
      - module-first testing 171
      - modules and 170
      - scheme 171
      - software tool goals 171
      - vs. human execution 171
    - contracts 40, 72, 79
  - Controller 26–27
    - controller 26–30
      - Ajax 233
    - Controller pattern 25
    - convention over
      - configuration 153
  - CruiseControl 172–179
    - ad hoc user 172
    - build execution
      - frequency 176
    - build system 172
    - config file 173
    - control panel 177
    - email notification 178
    - execution result delivery 177
    - htmlmail notification 177
    - installation 172
    - notification aliases 178
    - sample project 173
    - website URL 172
  - CSS
    - Ajax and 225
    - testing 205
- 
- D**
- DAOs 399
    - debugging 380
    - implementation
      - approaches 361
    - JPA-based 379
    - lazy initialization 382
    - negative scenarios tests 381
    - testing 335
      - testing exceptions 382
  - data transfer objects 362
  - database
    - constraint 385
    - embedded 328
    - transactions 333
  - database container 130
  - database testing
    - best practices 356, 358
    - challenges 327–328
    - DAOs 327
      - integration tests 327
      - upper layers 327
  - datasets, asserting database
    - state 335–340
  - DbUnit 370
    - addUser 335
    - advanced techniques 347–356
    - annotations 397
    - Assertion 336

- DbUnit (*continued*)
    - avoiding repetition with
      - Template Method 347–352
    - best practices 332
    - CLEAN INSERT 335
    - CLOSE CONNECTION (operation) 335
    - connection 333
    - DAO interface 329
    - DatabaseOperation class 334
    - DatabaseSequenceFilter 347
    - datasets 330–335, 339, 374, 397
    - defining primary keys
      - dynamically 340
    - DELETE 334
    - DELETE ALL 335
    - dependencies 330
    - dynamic IDs 340
    - EL integration 353–356
    - EL support 354
    - empty.xml 350
    - example domain model 329
    - exporting dataset 347
    - FilteredDataSet 337, 347
    - handling NULL 342–345
    - ignoring columns 338
    - ignoring primary keys 339
    - importing from
      - database 346–347
    - INSERT 334
    - introduction 329–330
    - issues 331, 338, 342
    - logging 346
    - missing column issue 343
    - NONE 335
    - QueryDataSet 338
    - REFRESH 334
    - ReplacementDataSet 340, 344
    - sample application 329
    - SequenceTableFilter 338
    - Template Design Pattern 348
    - testAddUser 336, 339
    - testGetUserById 332
    - third-party tools 389, 397–401
    - trade-offs 345
    - TRANSACTION(operation) 335
    - transforming data 340–346
    - TRUNCATE 335
    - unitils.properties 400
    - UPDATE 334
    - user.dtd 345
    - user.xml 331
  - UserDaoJdbcImpl 331
  - user-EL.xml 356
  - user-ok.xml 342
  - user-reverted.xml 343, 345
  - users table 330–331
  - user-token.xml 340
  - using DTDs 345
  - Decorator pattern 338, 341
  - DefaultAccountManager 105
  - DefaultController 46
    - testing 30
  - DefaultSelenium 208
  - delta parameter 12
  - dependencies 161
    - reducing 72
    - separating code 73
  - dependency management 145, 154
    - repositories 145
  - deploy, Maven build phase 155
  - Design by Contract 5
  - design patterns
    - Controller 25
    - Decorator 338, 341
    - DTO 362
    - Facade 20
    - Inversion of Control 28, 106
    - Singleton 75
    - Template Method 348, 391, 395
  - development cycle 71, 169
  - development lifecycle
    - platforms 81–82
    - integration platform 81
  - development routine 138
  - directory structure
    - conventions for book 441
    - separate but equal 51
  - Document Object Model. *See* DOM
  - doGet, test code 272
  - DOM 225
    - Ajax 226, 229–230, 233, 249
    - scripts 228
  - domain object 32
    - passing 105
    - trapdoor for controlling 106
  - don't repeat yourself 336
  - doubles 6
  - DRY. *See* don't repeat yourself
  - DTOs. *See* data transfer objects
  - dummy test method 31
  - duplication, eliminating 79
  - DynaBeans, JSPs and 274
- ## E
- 
- EasyMock 117, 363
    - testing a servlet with 128
    - third-party tools 392
  - Eclipse 162, 212
    - Ant integration 444
    - code generation 256
    - Google Plug-in 251
    - installation 443
    - JUnit integration 442
    - JUnit test suite wizard 256
    - JUnitMAX 446
    - launch configuration 255
    - running JUnit tests from 443
    - testing with 256
  - EJB API 132
  - EJBs
    - AdministratorEJB 277
    - testing 277
  - EL. *See* Expression Language
  - ELContextImpl 354
  - ElementNotFoundException 198
  - embedded database 328, 367
    - cleanup 368
    - commit option 368
    - JavaScript 232
    - unit testing with 367
  - embedded web server 88
  - Jetty 89
  - encapsulation
    - bypassing in tests 407–411
    - reflection and 409
  - endsWith matcher 50
  - endXXX 274, 276
  - EntitiesHelper 336, 365, 373
  - EntityManager 366, 380, 399
  - EntityManagerFactory 370
  - environment 85
    - source-building 153
  - eq function 235
  - equals 204
  - equalToIgnoringCase
    - matcher 50
  - equalToIgnoringWhiteSpace
    - matcher 50
  - ErrorHandler, HtmlUnit 205
  - ErrorResponse 29
  - exception handling 40–45
  - exceptional conditions,
    - handling 41
  - exceptions
    - failing tests with 219
    - JavaScript 199

exceptions (*continued*)  
 testing and JUnit 417  
 testing for 198  
 expectations 114  
 with EasyMock 118  
 expected behavior 56  
 Expression Language 353  
 ELFunctionMapperImpl 375  
 ID solution 374  
 integrating with DbUnit 354–356  
 introduction 353  
 Extreme Programming rules 40, 43  
 The Simplest Thing That Could Possibly Work 269

## F

---

façade 20  
 Façade pattern 20  
 faces-config.xml 296  
 FacesContext 298  
 factorial function 235  
 tests 238  
 failing automatic tests 79  
 failure conditions, testing for 94  
 feature  
 core 6  
 overlap 390  
 program 6  
 Felix  
 bundle installation 315  
 bundles 312  
 junit command 324  
 FEST  
 accessing private members 411  
 assertions 405–406  
 EasyMock support 394–395  
 Fluent Assertions Module 405  
 introduction 390  
 FEST-Assert. *See* FEST, Fluent Assertions Module  
 FEST-Mocks 394  
 FEST-Reflect 411  
 TestingHelper 411  
 FileAssert 403  
 fine-grained isolation 99  
 Firefox 211  
 JsUnit 244  
 JUnit tests 218  
 Selenium 208  
 foreign keys 385  
 forms  
 Ajax 231

example page 201  
 Selenium and 220  
 testing 200–201  
 validation 200  
 frames, testing 202  
 framework  
 Cactus 260  
 for Java 4  
 source-building 153  
 Front Controller 26  
 functional tests 54  
 Ajax 229, 233  
 as performance integration tests 132  
 Selenium 229  
 functional unit test 61

## G

---

generic methods 76  
 getAlert 220  
 getContent 86, 110  
 getHandler 32  
 getTestParameters 18  
 glass box testing 62  
 global state 74  
 Google Web Toolkit  
 asynchronous mode 255  
 callback 253  
 design 252  
 finish test 255  
 GWTTestCase 253–254  
 introduction 251–257  
 junitCreator 252  
 methods for overriding 256  
 remote procedure call 253  
 running a test suite 257  
 schedule 255  
 test case 253  
 test framework 251  
 test suite 256  
 TimeoutException 255  
 greaterThan matcher 50  
 greaterThanOrEqual matcher 50  
 green bar 20  
 test run 20  
 groupId 161  
 Guice 391  
 GWT. *See* Google Web Toolkit  
 GWTTestCase 254  
 code generation 255  
 restrictions 256

## H

---

haltonerror 143  
 haltonfailure 143  
 Hamcrest 47–48  
 assertions 421  
 extensible 50  
 matcher 432, 434  
 matchers 47–51  
 handler, adding 32  
 hasEntry matcher 50  
 hasItem matcher 50  
 hasItems matcher 50  
 hasKey matcher 50  
 hasProperty matcher 50  
 hasValue matcher 50  
 Heisenberg uncertainty principle 111  
 Hevery, Miško 75  
 hibernate.properties 370  
 hidden dependencies 74  
 Hollywood Principle 28  
 HTML  
 focus 200  
 id 196  
 page title 221  
 parsing 208  
 reports 147–149  
 HTMLAnchor 308  
 htmlemail 178  
 HtmlPage  
 content access 195  
 frame support 202  
 HtmlUnit  
 Assert 191  
 Cactus and 206–208  
 Cactus integration 454  
 description 190  
 Eclipse configuration 453  
 element name 195  
 element reference 196  
 element types 195  
 example 191  
 exceptions 198  
 form testing 233  
 introduction 190–191  
 JavaScript and 203  
 JavaScriptJobManager 234  
 JSFUnit and 307  
 logging 199  
 navigation 199  
 object model 195  
 recommendation 222  
 standalone tests 193  
 standard configuration 452

- HtmlUnit (*continued*)
    - synchronizing Ajax calls 234
    - testing for supported browsers 192
    - testing multiple browsers 192
    - trailing slash 195
    - UnknownHostException 198
    - vs. Selenium 222
    - web browsers 192
    - web client 233
    - writing tests 191–205
    - XPath and 197
  - HTTP connection
    - HtmlUnit 193
    - mocking 107–113
    - sample 107–108
    - stubbing example 87
  - HTTP container 130
  - HttpClient 247–251
  - HttpServletRequest,
    - mocking 127
  - HttpUnit 276
  - HttpURLConnection 97
  - HttpURLConnectionFactory 112
  - Hudson 179–184
    - build triggers 182
    - configuration screen 181
    - configuring 180
    - home directory 180
    - installation 179
    - job home page 183
    - job-configuration screen 182
    - jobs 181
    - post-build triggers 183
    - project configuration 182
    - startup screen 180
    - website URL 179
- I**
- 
- ID
    - ELContextImpl 375
    - generators, testing 373
    - synchronization problem 374
  - IDEs
    - Eclipse 212
    - Selenium 208
  - iframes 202
  - IIS 247
  - impedance mismatch 327
  - ImportPackage 317
  - in-container testing 126, 129–130
    - Cactus 260
    - comparison with other approaches 131–134
    - complex configuration 133
    - frameworks 130
    - implementation
      - strategies 129
    - longer execution time 133
    - mock objects solution 127–129
    - mock objects, pros and cons 131–132
    - no good IDE support 133
    - specific tools required 133
    - stubs, pros and cons 131
    - typical lifecycle 130
    - vs. out-of-container testing 134
  - inheritance vs. composition 77
  - InputStream
    - expectation for close 115
    - mocking 114
  - install, Maven build phase 155
  - instanceOf matcher 49
  - integration test
    - Maven build phase 155
    - post-test 288
    - pre-test 286
  - integration unit test 61, 126, 366
    - testing interval 170
  - IntelliJ IDEA 162
  - Interceptor pattern 425
  - InterceptorRunner 427
  - interceptors
    - SampleLoggingInterceptor 429
    - SampleTimingInterceptor 430
    - test case 431
  - InterceptorStatement 426
  - interface 5
    - fluent 390, 411
  - Internet Explorer, and JUnit tests 218
  - Internet Information Services. *See* IIS
  - invalid URL, test for 94
  - Inversion of Control 28
    - applying to class 106
  - invocation-count 122
  - IOC. *See* Inversion of Control
  - is matcher 49
  - isAuthenticated 127
  - isCompatibleType matcher 49
  - isFalse function 236
  - isGreaterThan 406
  - isLessThan 406
  - isNull function 236
  - isTrue function 236
  - Ivy
    - adding to build file 145
    - dependency management with 145–146
    - installation 145
  - ivy.xml 146
  - ivy-module 146
- J**
- 
- Jakarta Cactus. *See* Cactus
  - JARs, external versions 440
  - Java 5, annotations 350
  - Java 6 234
    - JUnit 242
  - Java Community Process. *See* JCP
  - Java EE
    - application server 100
    - bypassing encapsulation 407
    - Cactus in-container testing 261
    - containers 279
    - spec 292
  - Java EE container 366
    - for security 84
  - Java Persistence API 361, 385
    - commitment level 367
    - embedded database 367
    - entities mapping 371
    - ID integration 378
    - mapping, caveats 371
    - query, duplicated objects 383
    - Telephone class
      - definition 362
    - test case classes 378
    - testing DAOs 379
    - testing foreign keys 385
    - testing ID generators 373–379
    - testing infrastructure 368–371
    - testing strategies 367
    - testing transactions 373, 401
    - transaction management 366
    - transactions 367
    - User class definition 362
    - UserDao implementation 363
    - UserFacadeImpl 364
  - javac 141
  - Javadocs 140

- JavaScript
    - Ajax and 225, 228
    - alerts 203, 220–221
    - embedded database 232
    - event handler 225
    - failure 199
    - JsUnit 237
    - RhinoUnit 234
    - test functions 239
    - testing 203, 234–244
  - JavaScript Object Notation. *See* JSON
  - JavaServer Faces. *See* JSF
  - JavaServer Pages. *See* JSPs
  - JCP 292
  - JDK, JUnit and 414
  - Jeffries, Ron 106
  - Jetty 89, 95, 177
    - Handler 91
  - JettySample 89
    - testing in browser 90
  - JMeter 59
  - JMock 117, 121, 396
    - mocking library 124
    - testing with 302
  - JPA. *See* Java Persistence API
  - JSF
    - application testing 301
    - black box testing 301
    - introduction 293–294
    - mocks and 302
    - performance testing 308
    - phases 309
    - sample application 294
    - static analysis 301
    - typical testing problems 300
  - JSFClientSession 304
  - JSFSession 304
  - JSFTimer 308–309
  - JSFUnit 301
    - Ajax testing 305
    - browser tests 305
    - HtmlUnit and 307
    - performance testing 308
    - RichFaces and 307
    - testing with 304
  - JSLint 245, 250
    - Ant 246
    - JSLintant 246
    - lint 251
    - RhinoUnit 235
  - JSLint4java, XML validation 250
  - JSON 247
    - JSLint 250
    - validating response 249
  - JSPs 273, 297
    - Cactus servlet test case 275
    - DynaBeans and 274
    - RichFaces and 299
    - unit testing 273
  - JsUnit 228
    - Aborted 242
    - addTestPage 240
    - addTestSuite 240
    - Ant 242
    - BUILD SUCCESSFUL 244
    - build.xml 242
    - console 244
    - Firefox tip 244
    - installation directory 243
    - installing 455
    - introduction 237
    - invocation order 239
    - jsunit\_self\_test 244
    - JsUnitTestSuite 240
    - jsUnitTestSuite.html 244
    - link href 238
    - log 244
    - recommendation 245
    - script src 238
    - server farm 244
    - setUp 239
    - setUpPage 239
    - setUpPageStatus 239
    - Status 242
    - suite 240
    - tearDown 239
    - test runner 241
    - test suites 239
    - testing manually 241
    - testRunner.html 240
    - tests 238
    - timeout 242
    - URL 244
    - vs. JUnit 239
    - vs. RhinoUnit 245
  - JsUnitTest 228
  - JsUnitTestSuite 240
  - JUnit 4–5
    - annotations 415
    - API changes 414
    - assertion errors 423
    - assumptions 422
    - benefits 13
    - Cactus 260
    - compatibility 414
    - constructors 414
    - custom matcher 432
    - design goals 9
    - download site 9
    - Eclipse integration 442
    - equals methods 204
    - exception testing 417
    - features 10–13
    - ignoring a test 417
    - jar 9
    - NetBeans integration 447
    - new assertions 421–422
    - overview 15–17
    - package structure
      - changes 414
    - reports with Maven 166
    - reports, future of 149
    - runners 419–421
    - setting up 9–10
    - static imports 415, 417
    - test method names 415
    - test runner 193, 419
    - test suites 419
    - third-party tools 390–391
    - version changes 413
    - vs. JsUnit 239
  - JUnit 3.x 413
  - JUnit 4.x 413
  - JUnit task 143
  - junit task, optional
    - component 144
  - junit.jar 145
  - junit-4.6 9
  - JUnit4OSGi 322
    - CalculatorService test 323
    - GUI runner 324
    - JUnit 3.x rules 324
  - JUnit4TestAdapter 414
  - JUnit-addons
    - accessing private members 410
    - assertions 401–403
    - introduction 390
    - list assertion 402
    - TestingHelper 410
  - junitCreator 255
  - JUnitMAX 446
  - junitreport 147
- K**
- 
- keyboard, navigation 200
- L**
- 
- Law of Demeter 74
  - lazy initialization
    - DAOs 382
    - exception 327

LENIENT\_DATES 404  
 LENIENT\_ORDER 404  
 lessThan matcher 50  
 lessThanOrEqual matcher 50  
 libraries, repository 440  
 List 49  
 ListAssertions 402  
 ListAvailableAlbumsBean 295  
 listeners 175  
   ELPostInsertEventListener 377  
   Hibernate 374, 376  
   setEntityManager and 377  
 local repository 155  
 locators, Selenium 219  
 log 175  
 logic unit test 61

**M**

---

managed-bean-class 296  
 ManagedSeleniumServer 213  
 MANIFEST.MF  
   calculator 315  
   for client app 317  
 mapping, unit tests 372  
 matchers 47, 49  
   BaseMatcher 433  
   custom 432, 436  
   Hamcrest 432  
   IsNotNullOrEmpty 433  
   IsStrongPassword 434  
   TypeSafeMatcher 433  
 matches function 235  
 matchesSafely 435  
 Maven 391  
   build directory 163  
   build lifecycles 155  
   build phases 155  
   Cactus tests and 285  
   Compiler plug-in 164  
   constraint framework 168  
   convention over  
     configuration 153  
   dependency  
     management 154  
   features 153–159  
   HTML JUnit reports and 166  
   installation steps 159  
   key principles 153  
   plug-ins, introduction 163–167  
   plug-ins, list 156  
   Project Object Model 157  
   pros and cons 167

  setting up project 159–163  
   Surefire plug-in 165  
   website URL 153, 166  
 Maven2 285  
   cactifyear 289  
   cactifywar 285  
   Cactus plug-in 1 285  
   Cactus plug-in 2 286  
   Cactus plug-in 3 288  
   cargo-maven2-plugin 287  
   maven-surefire-plugin 288  
   pre-integration phase 289  
 maven-compiler-plugin 164  
   configuring 164  
 maven-eclipse-plugin 162  
 maven-sampling 159  
   pom.xml for 160  
 maven-surefire-plugin 165  
 maven-surefire-report-  
   plugin 166  
 method factory 111  
 methodInvokerStatement 428  
 Microsoft  
   Msxml2.XMLHTTP 247  
   XMLHTTP 230, 247  
 mock frameworks 117–124  
 mock objects 68, 71, 85, 100  
   as probes 114  
   as Trojan horses 113–117  
   defining 108  
   Driver 73  
   for refactoring 110  
   HTTP connection 107  
   isolation testing 100  
   JDBC logic 104  
   refactoring with 104–107  
   sample application 100  
   swapping into production  
     code 108  
     testing persistence layer 366  
     unit testing with 100–103  
 MockAccountManager 103  
 MockConnectionFactory 113  
 MockInputStream 115  
 mock-object 122  
 mocks  
   FEST-Mocks 394  
   JSF and 302  
   Mycila 395  
   pros and cons 397  
   test method structure 391  
   transparent usage 391–397  
 MockURL 108  
 MockWebConnection 194  
 modelVersion 161

modificationset 175  
 modules 158  
   exceptions 394  
   specific properties 394  
 MOJOs 285  
 monumental  
   methodologies 269  
 multilayered applications 361  
   testing negative scenarios 365  
   testing strategies 363  
   testing with mocks 363–365  
 MusicStore application 294  
 Mycila  
   assertions 407  
   EasyMock support 395–397  
   introduction 391  
 MyFaces 293

**N**

---

NamingAssert 403  
 navigation  
   click 200  
   frames 202  
   keyboard 200  
   page 199  
   Selenium 218  
 negative tests 383  
 NetBeans  
   Ant integration 451  
   JUnit integration 447  
 NicelyResynchronizingAjax-  
   Controller 233  
 not function 236  
 not matcher 49  
 notNullValue matcher 50  
 NULLTest 343  
 nullValue matcher 50

**O**

---

object model 195  
 ObjectAssert 403  
 object-relational mapping,  
   tools 361  
 objects  
   interaction testing 71  
   rules 74  
 onblur 200  
 onfocus 200  
 onload 221  
 optimization 59  
 ORM. *See* object-relational  
   mapping

## OSGi

- alliance 311
- bundle 311
- bundle states 312
- CalculatorService 312
- framework 311
- in-container testing 322
- introduction 311–312
- lifecycle of service 311
- mock object testing 319
- service interface 313
- service sample 312
- testing services 318

**P**

- 
- package, Maven build phase 155
  - packaging 161
  - page
    - HTML 199
    - navigation through frames 202
    - tests and 219
  - Page Controller 26
  - Parameterized 17
  - parameters, sets 17
  - patterns, Interceptor 425
  - performance tests 59
    - Selenium 211
  - persistence layer
    - issues with 326
    - testing 361
    - testing with mocks 366
  - persistence.xml 370
  - plug-ins
    - cactifyear 289
    - execution order 287
    - JMock 396
    - Maven, list of 156
  - polymorphism 76, 78
    - vs. conditionals 77
  - POM. *See* Project Object Model
  - pom.xml 157
    - child modules 158
    - description elements 162
    - inheritance features 157
    - metadata for 161
  - poms, elements inherited from parent 158
  - Principle of Least Knowledge 74
  - printsummary 143
  - PrivateAccessor 410
  - procedural programming 76
  - Project Object Model 157
  - property elements 140

- ProtocolException 97
- proving component works 4–5
- public methods, signature changes 72
- publishers 175

**R**

- 
- red bar 20
  - redirector 263
    - application descriptors 279
    - cactification 282
    - servlet 264
  - Refactor, definition 37
  - refactoring 55, 67
    - example 105
    - improving implementation 55
    - method factory 111
  - reflection 9
    - encapsulation and 409
    - for bypassing encapsulation 407
  - reflection API, accessing private members 408
  - ReflectionAssert 403–405
  - releaseConnection 247
  - Request 26–27
  - RequestHandler 26–27, 32
    - testing 34
  - Response 26–27
  - RhinoUnit 228
    - fail 236
    - functions 235
    - installing 455
    - JavaScript testing 234
    - JSLint 235
    - mustCall 236
    - path 235, 237
    - recommendation 245
    - rhinoUnitAnt.js 237
    - testCases 235
    - vs. JsUnit 245
  - RichFaces
    - JSFUnit and 307
    - JSP and 299
  - Runner 16–17
  - runners
    - custom 425, 429
    - InterceptorRunner 427
    - JUnit4OSGi 324
  - runtime, compiler confirmation 6
  - RuntimeException, Selenium 220

**S**

- 
- sameInstance matcher 50
  - SampleExceptionHandler 41
  - SampleResponse, refactored 39
  - schedule 175
  - SchemaTest.java 385
  - screen shot 221
  - ScriptException, HtmlUnit 199
  - SecurityManager 409
  - SeleneseTestCase 210
    - setUp 216
    - subclassing 213
  - Selenium 228
    - description 190
    - example 210
    - forms 220
    - IDE 208
    - installing 455
    - introduction 208
    - JUnit 4 and 212
    - launcher strings 215
    - locator string 218
    - locators 219
    - managed 231
    - navigation 218
    - performance 211–212
    - performance drawback 213
    - proxy injection 216
    - recommendation 222
    - running tests 211–215
    - security 216
    - server management 211–212
    - Suite class and 214
    - supported browsers 215
    - test generation 210
    - testing Ajax 229
    - vs. HtmlUnit 222
    - waitForCondition 232
    - writing tests 215–222
  - SeleniumCommandTimedOutException 220
  - SeleniumException 220–221
  - separate-but-equal filing system 51
  - ServiceReference 324
  - services, independent validation 247
  - Servlet API 132
  - ServletConfig 264
  - ServletTestCase
    - HtmlUnit boilerplate 206
    - instance variables 207
  - setAjaxController 233
  - setUp 211

- setup logic, factorizing 36
  - signature 34
  - Singleton pattern 75
  - singletons, pros and cons 75
  - software tests, types 57
    - acceptance tests 60
    - functional tests 58
    - integration tests 57
    - stress tests 58
  - SourceForge 4
  - Spring 371
  - SSL, errors 205
  - standalone tests
    - configuring 194
    - HtmlUnit 193
  - startWith matcher 50
  - static import 11, 118
    - JUnit and 417
  - stepwise refinement 55
  - StringAssert 403
  - StrutsTestCase 62
  - stubbing 85
    - connection 95–98
    - drawbacks 88
    - HTTP connection 86–89
    - remote web resource 86
    - solutions 88
    - web server resources 90–95
  - StubHttpURLConnection 98
  - stubs 68, 85–86
    - cons 86
    - definition 85
    - setting up test 90
    - test review 95
  - Suite 16–17, 21, 214
    - composing from test classes 22
    - of suites 22
  - suite, composing tests with 21–23
  - Super POM 157
  - Surefire plug-in 165
- T**
- 
- Taglib API 134
  - target.dir 154
  - targets 139–140
    - compile 142
    - compile.java 142
    - compile.test 142
    - test 143
  - TDD. *See* test-driven development
  - tearDown 211
  - Template Method pattern 348
  - test calculator class 6
  - test case 16
  - test classes 11, 14, 16–17
    - placement 33
    - public 11
    - writing 92
  - test coverage 68
    - complete 68
    - measuring 67–72
    - reports, generating 70
    - with white box testing 68
  - Test Driven Development: By Example* 125
  - test objects 32
  - test runners 14, 16, 19–20
    - Cactus 263
    - custom 20
    - default 20
    - JUnit 241
    - JUnit 4, table 19
    - JUnit38ClassRunner 19
    - Parameterized 19
    - Suite 19
  - test suite 14, 16
    - benefits 100
    - Google Web Toolkit 256
    - JUnit 239
    - Selenium 230
  - test target 143
  - test types 57–62
  - test, Maven build phase 155
  - testable code, writing 72–78
  - TestAccountService
    - with EasyMock 117
    - with JMock 121
  - testAdd 8, 15
  - TestAdminServlet.java 275
  - TestCase 16, 415
  - TestCase class 11
  - TestDefaultController 33, 36
  - testDeleteUserWithTelephone 381
  - test-driven development 78–80
    - adapting cycle 78
    - Cactus and 266
    - definition 78
    - development cycle 78
    - development lifecycle 80
    - JUnit best practice 269
    - test, code, refactor 79
    - two-step 79
  - testform.html 230
  - testFormNo 231
  - testGetUserByIdWithTelephone 382
  - test-infected 4
  - testing
    - assertions 401
    - automatically 4
    - black box vs. white box 63
    - by hand 40
    - class factory 111
    - controller package 50
    - for exceptions 44–45
    - framework, choosing 190
    - HtmlUnit 233
    - ignoring in JUnit 417
    - in isolation 100, 102
    - JavaScript frameworks for 234
    - method refactoring 109
    - multiple classes 138
    - multiple web browsers 216
    - patterns, Ajax 227
    - repetitive 4
    - services 228
    - solution, elegant 107
    - strategies 366
    - three steps 103
    - timeout 418
    - tools 389
  - TestingHelper
    - accessing private members 408
    - FEST-Reflect 411
    - JUnit-addons 410
    - reflection API 408
  - TestingHelperFESTReflect 411
  - TestingHelperJUnitAddons 410
  - testProcessRequest 35
    - improving 38
  - testProcessRequestAnswers-ErrorResponse 42
  - testProcessRequestExceptionHandler 42
  - TestRunner 264
  - tests 3, 17
    - automated 6
    - batching 149–150
    - Cactus servlet tests 266
    - Cactus vs. JUnit 261
    - diagnostic 40
    - failure 7
    - fine-grained 134
    - incremental 5
    - isolation vs. performance 92
    - JUnit 238
    - JUnit, from NetBeans 449
    - negative 383
    - parameterized 17–19, 420

tests (*continued*)  
 patterns 35  
 programmer 5  
 with multiple page  
   fixtures 194  
   writing first 107  
 testSaveUserWithTelephone 373  
 TestWebClient  
   with close expectation 115  
   with JMock 123  
 TestWebClient1 98  
 test.XXX 273  
   six steps 264  
 test.XXX pattern 11  
 test.XXXYYY 34  
 text test runner 9  
 The Simplest Thing That Could  
   Possibly Work 55, 269  
 third-party tools 390–391  
 time barrier 46  
 timeout 46, 60  
   Ajax 232  
   testing 418  
 timeout tests 45–47  
 try/catch block 8

## U

---

undocumented exceptions 44  
 unit of work 5  
 unit test method 11  
 unit testing  
   Ajax 228  
   benefits 134  
   client-side scripts 228  
   impedance mismatch 327  
   in-container 126  
   JSPs 273  
   limitations of standard  
     tests 126–127  
   objective 32  
   with IoC 106  
 unit testing framework 8–9  
   rules 8–9  
 unit tests 5  
   as automatic  
     documentation 56  
     code in isolation 327  
     coverage improvement 54  
     ease of dev and use 328  
     executing 165  
     fast to run 328  
   productivity improvement 54  
   protect against regression 100  
   protected methods 51

refactoring with  
   confidence 55  
 regressions and debugging 54  
 success rate 150  
 two for each persistent  
   entity 372  
 types 60–61  
 vs. functional tests 54  
 when to execute 170  
 why necessary 54–56  
 Unitils 371  
   assertions 403–405  
   database testing modules 397  
   DbUnit integration 397–401  
   dummies 394  
   EasyMock support 392–394  
   introduction 390  
   modules 392–393  
   properties 392  
   UserFacadeImpl 392  
 unitils.properties 400  
 UNIX, cron tool 182  
 updateAccount 102

## URL

  HtmlUnit 193  
   protocol handler, custom 95  
 URLStreamHandler 95  
 user-centric approach 63  
 UserDaoJdbcImplAnnotation-  
   Test 355  
 UserDaoJdbcImplTemplate-  
   PatternTest 348  
 UserDaoJdbcImplTest 331, 335–  
   336, 339  
   Unitils and 398  
 UserDaoJpaImpl 366  
   final version 384  
 UserDaoJpaImplTest 379  
 UserFacade 382  
 UserFacadeImpl  
   FEST-Mocks and 394  
   JMock and 396  
   Mycila and 395  
   Unitils mock support 392  
 UserFacadeImplFESTTest 394  
 UserFacadeImplMycilaEasy-  
   MockTest 395  
 UserFacadeImplTest 391  
 user-with-telephone.xml 372,  
   374

## V

---

validate, Maven build phase 155  
 validation, form 200

verify, Maven build phase 155  
 version 161  
   reverting to last stable 447

## W

---

waitForCondition 232  
 web browsers  
   HtmlUnit 192  
   testing multiple 216  
 WebAssert 191, 198  
 Web-classic interaction 225  
 WebClient 195  
   closing input stream 116  
   refactoring 112–113  
   with EasyMock 119  
 white box testing 60, 62  
   combining with black box  
     testing 71  
   coverage 68  
   unit tests 68, 71  
 Winstone 179

## X

---

### XML

  response, validating 248  
   service, calling 247  
   validation 250  
   vocabulary 248  
   well formed 248  
 XML Schema 248  
 XMLHttpRequest 225  
   callback 226  
   form testing 233  
   scripts 228

### XPath

  accessing elements with 197  
   data gathering 197  
   HtmlUnit 197  
   query 197

xUnit 4

## Y

---

YAGNI. *See* You Ain't Gonna  
 Need It  
 You Ain't Gonna Need It 55  
 YYYTestCase 263  
   six steps 264

## Z

---

ZipURLInstaller 284