



Java EE 8 Recipes

A Problem-Solution Approach

Proven solutions for Java Enterprise
Edition 8 Development

Second Edition

Josh Juneau

Apress®

Java EE 8 Recipes

A Problem-Solution Approach

Second Edition



Josh Juneau

Apress®

Java EE 8 Recipes

Josh Juneau
Hinckley, Illinois, USA

ISBN-13 (pbk): 978-1-4842-3593-5
<https://doi.org/10.1007/978-1-4842-3594-2>

ISBN-13 (electronic): 978-1-4842-3594-2

Library of Congress Control Number: 2018946699

Copyright © 2018 by Josh Juneau

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484235935. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

This book is dedicated to my wife Angela, my five children: Kaitlyn, Jacob, Matthew, Zachary, and Lucas. You are my joy and inspiration. It is also dedicated to the many Java developers worldwide. I hope that these recipes can lead you to developing the sophisticated solutions of tomorrow.

—Josh Juneau

Contents

- About the Authorxxxix
- About the Technical Reviewer xli
- Acknowledgments xliii
- Introductionxlv
- Chapter 1: Working with Servlets..... 1
 - 1-1. Setting Up a Java Enterprise Environment 2
 - Problem2
 - Solution #1.....2
 - Solution #2.....3
 - How It Works.....3
 - 1-2. Developing a Servlet..... 4
 - Problem4
 - Solution.....4
 - How It Works.....7
 - 1-3. Packaging, Compiling, and Deploying a Servlet..... 9
 - Problem9
 - Solution.....9
 - How It Works.....10
 - 1-4. Registering Servlets Without WEB-XML..... 11
 - Problem11
 - Solution.....11
 - How It Works.....13

- 1-5. Displaying Dynamic Content with a Servlet..... 14
 - Problem 14
 - Solution..... 14
 - How It Works..... 16
- 1-6. Handling Requests and Responses..... 17
 - Problem 17
 - Solution..... 17
 - How It Works..... 19
- 1-7. Listening for Servlet Container Events..... 20
 - Problem 20
 - Solution..... 20
 - How It Works..... 22
- 1-8. Setting Initialization Parameters..... 23
 - Problem 23
 - Solution #1..... 23
 - Solution #2..... 24
 - How It Works..... 24
- 1-9. Filtering Web Requests 25
 - Problem 25
 - Solution..... 25
 - How It Works..... 26
- 1-10. Listening for Attribute Changes 27
 - Problem 27
 - Solution..... 27
 - How It Works..... 29
- 1-11. Applying a Listener to a Session..... 30
 - Problem 30
 - Solution..... 30
 - How It Works..... 31

1-12. Managing Session Attributes	32
Problem	32
Solution.....	32
How It Works.....	34
1-13. Downloading a File	34
Problem	34
Solution.....	34
How It Works.....	37
1-14. Dispatching Requests	38
Problem	38
Solution.....	38
How It Works.....	42
1-15. Redirecting to a Different Site	43
Problem	43
Solution.....	43
How It Works.....	43
1-16. Securely Maintaining State Within the Browser	44
Problem	44
Solution.....	44
How It Works.....	47
1-17. Finalizing Servlet Tasks	48
Problem	48
Solution.....	48
How It Works.....	49
1-18. Reading and Writing with Nonblocking I/O	49
Problem	49
Solution.....	49
How It Works.....	54

1-19. Pushing Resources from a Server to a Client	56
Problem	56
Solution.....	56
How It Works.....	57
■ Chapter 2: JavaServer Pages	59
2-1. Creating a Simple JSP Page	60
Problem	60
Solution.....	60
How It Works.....	61
2-2. Embedding Java into a JSP Page	62
Problem	62
Solution.....	62
How It Works.....	63
2-3. Separating Business Logic from View Code	64
Problem	64
Solution.....	64
How It Works.....	66
2-4. Yielding or Setting Values	67
Problem	67
Solution.....	67
How It Works.....	68
2-5. Invoking a Function in a Conditional Expression	70
Problem	70
Solution.....	70
How It Works.....	72
2-6. Creating a JSP Document	74
Problem	74
Solution.....	74
How It Works.....	75

2-7. Embedding Expressions in EL	76
Problem	76
Solution.....	76
How It Works.....	78
2-8. Accessing Parameters in Multiple Pages.....	81
Problem	81
Solution.....	81
How It Works.....	82
2-9. Creating a Custom JSP Tag	83
Problem	83
Solution.....	83
How It Works.....	86
2-10. Including Other JSPs into a Page.....	87
Problem	87
Solution.....	87
How It Works.....	88
2-11. Creating an Input Form for a Database Record.....	89
Problem	89
Solution.....	89
How It Works.....	92
2-12. Looping Through Database Records Within a Page	94
Problem	94
Solution.....	94
How It Works.....	97
2-13. Handling JSP Errors.....	98
Problem	98
Solution.....	98
How It Works.....	99

2-14. Disabling Scriptlets in Pages	100
Problem	100
Solution.....	100
How It Works.....	101
2-15. Ignoring EL in Pages	101
Problem	101
Solution #1.....	101
Solution #2.....	101
Solution #3.....	101
How It Works.....	102
■ Chapter 3: The Basics of JavaServer Faces	103
3-1. Writing a Simple JSF Application	104
Problem	104
Solution #1.....	104
Solution #2.....	107
How It Works.....	108
3-2. Writing a Controller Class	110
Problem	110
Solution.....	110
How It Works.....	115
3-3. Building Sophisticated JSF Views with Components.....	117
Problem	117
Solution.....	117
How It Works.....	123
3-4. Displaying Messages in JSF Pages	125
Problem	125
Solution.....	125
How It Works.....	127

3-5. Updating Messages Without Recompiling	129
Problem	129
Solution.....	129
How It Works.....	131
3-6. Navigating Based Upon Conditions.....	131
Problem	131
Solution.....	131
How It Works.....	136
3-7. Validating User Input.....	138
Problem	138
Solution.....	138
How It Works.....	142
3-8. Evaluating Page Expressions Immediately	144
Problem	144
Solution.....	144
How It Works.....	146
3-9. Passing Page Parameters to Methods	146
Problem	146
Solution.....	146
How It Works.....	150
3-10. Using Operators and Reserved Words in Expressions	150
Problem	150
Solution.....	150
How It Works.....	152
3-11. Creating Bookmarkable URLs	154
Problem	154
Solution.....	154
How It Works.....	155

3-12. Displaying Lists of Objects.....	156
Problem	156
Solution.....	156
How It Works.....	160
3-13. Developing with HTML5	161
Problem	161
Solution.....	161
How It Works.....	162
3-14. Creating Page Templates	163
Problem	163
Solution.....	163
How It Works.....	164
3-15. Applying Templates	168
Problem	168
Solution.....	168
How It Works.....	175
3-16. Adding Resources into the Mix	177
Problem	177
Solution.....	178
How It Works.....	180
3-17. Handling Variable-Length Data	181
Problem	181
Solution.....	181
How It Works.....	183
3-18. Invoking Controller Class Actions on Lifecycle Phase Events	188
Problem	188
Solution.....	188
How It Works.....	188

■ Chapter 4: JavaServer Faces Standard Components	191
Component and Tag Primer	191
Common Component Tag Attributes	194
Common JavaScript Component Tags	194
Binding Components to Properties	195
4-1. Creating an Input Form	196
Problem	196
Solution.....	196
How It Works.....	199
4-2. Invoking Actions from Within a Page	201
Problem	201
Solution.....	201
How It Works.....	204
4-3. Displaying Output.....	206
Problem	206
Solution.....	206
How It Works.....	209
4-4. Adding Form Validation	213
Problem	213
Solution #1.....	213
Solution #2.....	214
Solution #3.....	214
How It Works.....	216
4-5. Adding Select Lists to Pages.....	219
Problem	219
Solution.....	219
How It Works.....	221
4-6. Adding Graphics to Your Pages	223
Problem	223
Solution.....	223
How It Works.....	223

4-7. Adding Check Boxes to a View	224
Problem	224
Solution.....	224
How It Works.....	227
4-8. Adding Radio Buttons to a View	229
Problem	229
Solution.....	229
How It Works.....	230
4-9. Displaying a Collection of Data	231
Problem	231
Solution.....	231
How It Works.....	236
4-10. Utilizing Custom JSF Component Libraries.....	239
Problem	239
Solution.....	239
How It Works.....	239
4-11. Implementing File Uploading	240
Problem	240
Solution.....	240
How It Works.....	240
■ Chapter 5: Advanced JavaServer Faces and Ajax	243
5-1. Validating Input with Ajax.....	244
Problem	244
Solution.....	244
How It Works.....	248
5-2. Submitting Pages Without Page Reloads.....	251
Problem	251
Solution.....	251
How It Works.....	251

5-3. Making Partial-Page Updates	252
Problem	252
Solution.....	252
How It Works.....	253
5-4. Applying Ajax Functionality to a Group of Components	253
Problem	253
Solution.....	253
How It Works.....	257
5-5. Custom Processing of Ajax Functionality	258
Problem	258
Solution.....	258
How It Works.....	260
5-6. Custom Conversion of Input Values	260
Problem	260
Solution.....	261
How It Works.....	262
5-7. Maintaining Managed Bean Scopes for a Session.....	264
Problem	264
Solution.....	264
How It Works.....	273
5-8. Listening for System-Level Events	274
Problem	274
Solution.....	274
How It Works.....	276
5-9. Listening for Component Events	276
Problem	276
Solution.....	276
How It Works.....	277

- 5-10. Invoking a Managed Bean Action on Render 278
 - Problem 278
 - Solution..... 278
 - How It Works..... 279
- 5-11. Asynchronously Updating Components 280
 - Problem 280
 - Solution..... 280
 - How It Works..... 283
- 5-12. Developing JSF Components Containing HTML5..... 283
 - Problem 283
 - Solution..... 283
 - How It Works..... 285
- 5-13. Listening to JSF Phases..... 286
 - Problem 286
 - Solution..... 286
 - How It Works..... 288
- 5-14. Adding Auto-Completion to Text Fields 288
 - Problem 288
 - Solution..... 288
 - How It Works..... 290
- 5-15. Developing Custom Constraint Annotations..... 291
 - Problem 291
 - Solution..... 291
 - How It Works..... 293
- 5-16. Developing a Page Flow..... 295
 - Problem 295
 - Solution..... 295
 - How It Works..... 298

5-17. Constructing a JSF View in Pure HTML5.....	301
Problem	301
Solution.....	301
How It Works.....	302
5-18. Invoking Server-Side Methods via Ajax	302
Problem	302
Solution.....	303
How It Works.....	305
5-19. Broadcasting Messages from the Server to All Clients.....	305
Problem	305
Solution.....	306
How It Works.....	307
5-20. Programmatically Searching for Components	308
Problem	308
Solution #1.....	308
Solution #2.....	309
How It Works.....	310
■ Chapter 6: The MVC Framework	313
6-1. Configuring an Application for the MVC Framework.....	314
Problem	314
Solution.....	314
How It Works.....	316
6-2. Making Data Available for the Application	317
Problem	317
Solution #1.....	317
Solution #2.....	321
How It Works.....	326
6-3. Writing a Controller Class	327
Problem	327
Solution.....	327
How It Works.....	328

6-4. Using a Model to Expose Data to a View.....	330
Problem	330
Solution.....	330
How It Works.....	332
6-5. Utilizing CDI for Exposing Data	332
Problem	332
Solution.....	332
How It Works.....	334
6-6. Supplying Message Feedback to the User.....	335
Problem	335
Solution.....	335
How It Works.....	337
6-7. Inserting and Updating Data	338
Problem	338
Solution.....	338
How It Works.....	339
6-8. Applying a Different View Engine.....	340
Problem	340
Solution #1.....	341
Solution #2.....	341
How It Works.....	343
■ Chapter 7: JDBC.....	345
7-1. Obtaining Database Drivers and Adding Them to the CLASSPATH.....	346
Problem	346
Solution.....	346
How It Works.....	346
7-2. Connecting to a Database	347
Problem	347
Solution #1.....	347
Solution #2.....	348
How It Works.....	351

7-3. Handling Database Connection Exceptions	352
Problem	352
Solution.....	353
How It Works.....	353
7-4. Simplifying Connection Management.....	353
Problem	353
Solution.....	354
How It Works.....	357
7-5. Querying a Database.....	358
Problem	358
Solution.....	358
How It Works.....	359
7-6. Performing CRUD Operations.....	360
Problem	360
Solution.....	360
How It Works.....	362
7-7. Preventing SQL Injection.....	363
Problem	363
Solution.....	364
How It Works.....	367
7-8. Utilizing Java Objects for Database Access	370
Problem	370
Solution.....	370
How It Works.....	376
7-9. Navigating Data with Scrollable ResultSets	376
Problem	376
Solution.....	377
How It Works.....	378

7-10. Calling PL/SQL Stored Procedures.....	379
Problem	379
Solution.....	379
How It Works.....	380
7-11. Querying and Storing Large Objects	380
Problem	380
Solution.....	381
How It Works.....	383
7-12. Caching Data for Use When Disconnected.....	384
Problem	384
Solution.....	384
How It Works.....	387
7-13. Joining RowSet Objects When Not Connected to the Data Source.....	389
Problem	389
Solution.....	389
How It Works.....	392
7-14. Querying with a REF_CURSOR.....	393
Problem	393
Solution.....	394
How It Works.....	394
■ Chapter 8: Object-Relational Mapping.....	395
8-1. Creating an Entity	396
Problem	396
Solution.....	396
How It Works.....	399
8-2. Mapping Data Types.....	400
Problem	400
Solution.....	401
How It Works.....	402

8-3. Creating a Persistence Unit	403
Problem	403
Solution.....	403
How It Works.....	404
8-4. Using Database Sequences to Create Primary Key Values	406
Problem	406
Solution.....	406
How It Works.....	408
8-5. Generating Primary Keys Using More Than One Attribute.....	410
Problem	410
Solution #1.....	410
Solution #2.....	413
How It Works.....	416
8-6. Defining a One-to-One Relationship	418
Problem	418
Solution.....	418
How It Works.....	420
8-7. Defining One-to-Many and Many-to-One Relationships	420
Problem	420
Solution.....	421
How It Works.....	422
8-8. Defining a Many-to-Many Relationship	424
Problem	424
Solution.....	424
How It Works.....	426
8-9. Querying with Named Queries	428
Problem	428
Solution.....	428
How It Works.....	429

8-10. Performing Validation on Entity Fields	429
Problem	429
Solution.....	430
How It Works.....	431
8-11. Generating Database Schema Objects Automatically	432
Problem	432
Solution.....	432
How It Works.....	432
8-12. Mapping Date-Time Values	436
Problem	436
Solution.....	436
How It Works.....	437
8-13. Using the Same Annotation Many Times	437
Problem	437
Solution.....	438
How It Works.....	439
■ Chapter 9: Enterprise JavaBeans	441
9.1. Obtaining an Entity Manager	441
Problem	441
Solution #1.....	442
Solution #2.....	442
How It Works.....	442
9.2. Developing a Stateless Session Bean	443
Problem	443
Solution #1.....	443
Solution #2.....	444
How It Works.....	447
9.3. Developing a Stateful Session Bean.....	449
Problem	449
Solution.....	449
How It Works.....	453

9.4. Utilizing Session Beans with JSF	455
Problem	455
Solution.....	455
How It Works.....	457
9.5. Persisting an Object	459
Problem	459
Solution.....	459
How It Works.....	459
9.6. Updating an Object.....	460
Problem	460
Solution.....	460
How It Works.....	460
9.7. Returning Data to Display in a Table	460
Problem	460
Solution #1.....	461
Solution #2.....	462
How It Works.....	463
9.8. Creating a Singleton Bean.....	464
Problem	464
Solution.....	465
How It Works.....	467
9.9. Scheduling a Timer Service.....	468
Problem	468
Solution #1.....	468
Solution #2.....	468
How It Works.....	469
9.10. Performing Optional Transaction Lifecycle Callbacks	472
Problem	472
Solution.....	472
How It Works.....	473

9.11. Ensuring a Stateful Session Bean Is Not Passivated	473
Problem	473
Solution.....	473
How It Works.....	474
9.12. Denoting Local and Remote Interfaces	474
Problem	474
Solution.....	474
How It Works.....	474
9.13. Processing Messages Asynchronously from Enterprise Beans.....	476
Problem	476
Solution.....	476
How It Works.....	477
■ Chapter 10: The Query API and JPQL.....	479
10-1. Querying All Instances of an Entity	479
Problem	479
Solution #1.....	479
Solution #2.....	480
How It Works.....	480
10-2. Setting Parameters to Filter Query Results.....	482
Problem	482
Solution #1.....	482
Solution #2.....	482
How It Works.....	482
10-3. Returning a Single Object	484
Problem	484
Solution.....	484
How It Works.....	484
10-4. Creating Native Queries	484
Problem	484
Solution #1.....	485

Solution #2.....	485
How It Works.....	486
10-5. Querying More Than One Entity	487
Problem	487
Solution #1.....	487
Solution #2.....	488
How It Works.....	489
10-6. Calling JPQL Aggregate Functions	491
Problem	491
Solution.....	491
How It Works.....	492
10-7. Invoking Database Stored Procedures Natively	492
Problem	492
Solution.....	492
How It Works.....	493
10-8. Joining to Retrieve Instances Matching All Cases	493
Problem	493
Solution.....	493
How It Works.....	494
10-9. Joining to Retrieve All Rows Regardless of Match	494
Problem	494
Solution.....	495
How It Works.....	495
10-10. Applying JPQL Functional Expressions	496
Problem	496
Solution.....	496
How It Works.....	497
10-11. Forcing Query Execution Rather Than Cache Use.....	498
Problem	498
Solution.....	498
How It Works.....	498

10-12. Performing Bulk Updates and Deletes	499
Problem	499
Solution.....	499
How It Works.....	500
10-13. Retrieving Entity Subclasses	501
Problem	501
Solution.....	501
How It Works.....	502
10-14. Joining with ON Conditions.....	502
Problem	502
Solution.....	502
How It Works.....	503
10-15. Processing Query Results with Streams.....	504
Problem	504
Solution.....	504
How It Works.....	504
10-16. Converting Attribute Data Types.....	505
Problem	505
Solution.....	505
How It Works.....	506
■ Chapter 11: Bean Validation	507
11-1. Validating Fields with Built-In Constraints	508
Problem	508
Solution #1.....	508
Solution #2.....	508
How It Works.....	509
11-2. Writing Custom Constraint Validators	510
Problem	510
Solution.....	510
How It Works.....	512

11-3. Validating at the Class Level	512
Problem	512
Solution.....	512
How It Works.....	514
11-4. Validating Parameters.....	515
Problem	515
Solution.....	515
How It Works.....	515
11-5. Constructor Validation.....	516
Problem	516
Solution.....	516
How It Works.....	516
11-6. Validating Return Values	517
Problem	517
Solution.....	517
How It Works.....	517
11-7. Defining a Dynamic Validation Error Message	518
Problem	518
Solution.....	518
How It Works.....	518
11-8. Manually Invoking Validator Engine	519
Problem	519
Solution.....	519
How It Works.....	520
11-9. Grouping Validation Constraints.....	520
Problem	520
Solution.....	520
How It Works.....	522

- **Chapter 12: Java EE Containers 523**
 - 12.1. Installing GlassFish 5 or Payara 5 and Starting Up 523
 - Problem 523
 - Solution #1: GlassFish 523
 - Solution #2: Payara..... 524
 - How It Works..... 524
 - 12.2. Logging into the Administrative Console 524
 - Problem 524
 - Solution..... 525
 - How It Works..... 526
 - 12.3. Changing the Administrator User Password 530
 - Problem 530
 - Solution #1..... 530
 - Solution #2..... 530
 - How It Works..... 531
 - 12.4. Deploying a WAR File..... 531
 - Problem 531
 - Solution #1..... 531
 - Solution #2..... 533
 - How It Works..... 533
 - 12.5. Adding a Database Resource 534
 - Problem 534
 - Solution..... 534
 - How It Works..... 537
 - 12.6. Adding Forms-Based Authentication 539
 - Problem 539
 - Solution..... 539
 - How It Works..... 544

12.7. Deploying a Microservice to Payara Micro	545
Problem	545
Solution.....	545
How It Works.....	551
12.8. Packaging a Web Application with Payara Micro as an Executable JAR.....	553
Problem	553
Solution.....	553
How It Works.....	554
12.9. Deploying Payara Micro Apps on Docker.....	554
Problem	554
Solution.....	555
How It Works.....	556
■ Chapter 13: Contexts and Dependency Injection	559
13-1. Injecting a Contextual Bean or Other Object.....	560
Problem	560
Solution.....	560
How It Works.....	561
13-2. Binding a Bean to a Web View	562
Problem	562
Solution.....	562
How It Works.....	564
13-3. Allocating a Specific Bean for Injection	565
Problem	565
Solution.....	565
How It Works.....	567
13-4. Determining the Scope of a Bean	568
Problem	568
Solution.....	568
How It Works.....	570

13-5. Injecting Non-Bean Objects	571
Problem	571
Solution.....	571
How It Works.....	573
13-6. Ignoring Classes	574
Problem	574
Solution #1.....	574
Solution #2.....	574
How It Works.....	575
13-7. Disposing of Producer Fields	575
Problem	575
Solution.....	576
How It Works.....	576
13-8. Specifying an Alternative Implementation at Deployment Time	576
Problem	576
Solution.....	576
How It Works.....	577
13-9. Injecting a Bean and Obtaining Metadata.....	577
Problem	577
Solution.....	577
How It Works.....	578
13-10. Invoking and Processing Events	578
Problem	578
Solution.....	578
How It Works.....	581
13-11. Intercepting Method Invocations	582
Problem	582
Solution.....	582
How It Works.....	584