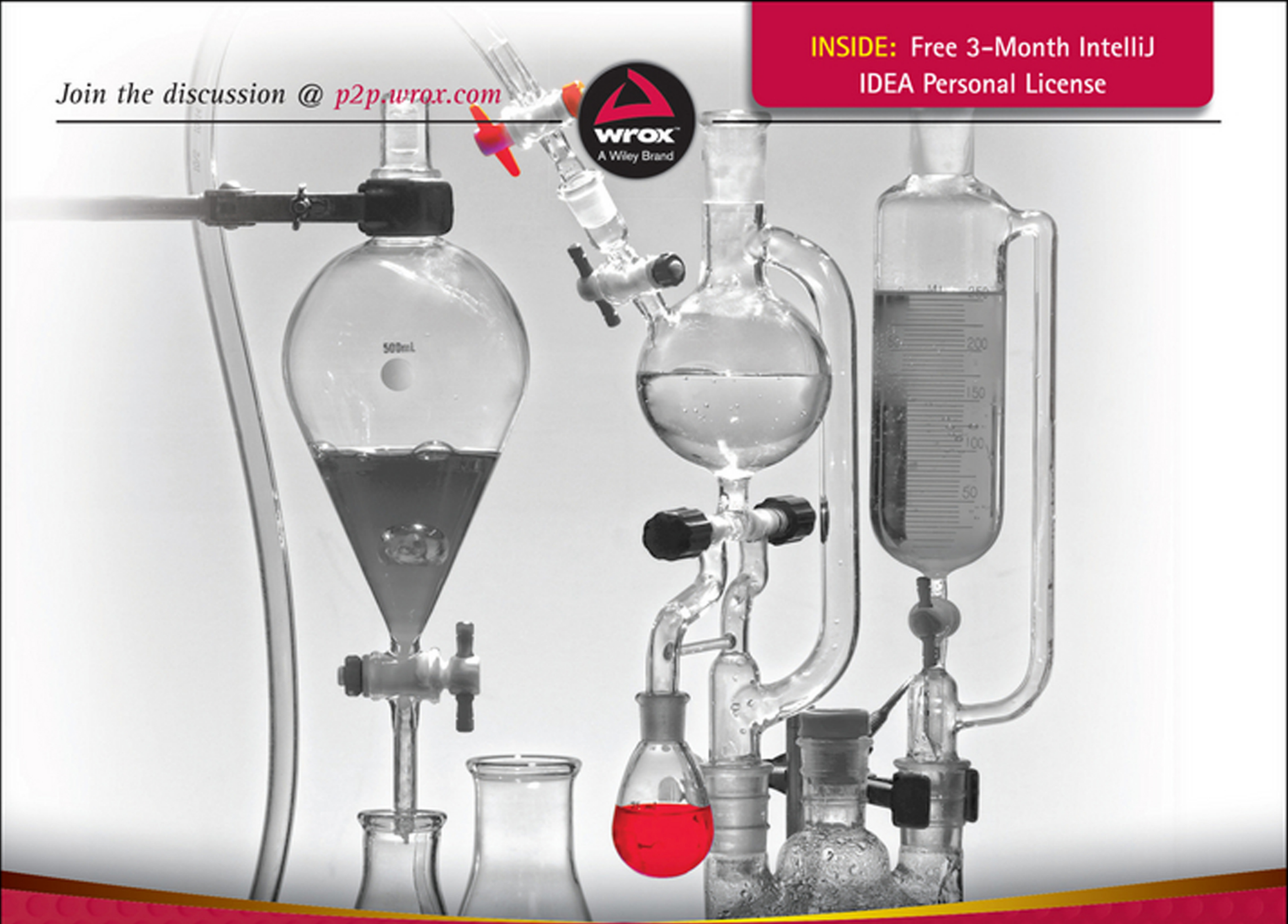

Join the discussion @ p2p.wrox.com

wrox
A Wiley Brand

INSIDE: Free 3-Month IntelliJ IDEA Personal License

Professional Java® for Web Applications

Featuring WebSockets, Spring Framework, JPA Hibernate, and Spring Security

Nicholas S. Williams

PROFESSIONAL

Java® for Web Applications

Nicholas S. Williams

wrox™
A Wiley Brand

Professional Java® for Web Applications

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Copyright © 2014 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-65646-4
ISBN: 978-1-118-65651-8 (ebk)
ISBN: 978-1-118-90931-7 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permissions.

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2013958292

Trademarks: Wiley, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Java is a registered trademark of Oracle America, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

ABOUT THE AUTHOR

NICK WILLIAMS is a Software Engineer for UL Workplace Health and Safety in Franklin, Tennessee. A computer science graduate from Belmont University, he has been active in commercial and open source software projects for more than 9 years. He is the founder of `DNSCrawler.com`, a site for free DNS and IP troubleshooting tools, and NWTS Java Code, an open source community that specializes in obscure Java libraries that meet niche needs. In 2010, the Nashville Technology Council named him the Software Engineer of the Year for Middle Tennessee. Nick is a committer for Apache Logging (including Log4j) and Jackson Data Processor JSR 310 Data Types. He has also contributed new features to Apache Tomcat 8.0, Spring Framework 4.0, Spring Security 3.2, Spring Data Commons 1.6, Spring Data JPA 1.4, and JBoss Logging 3.2; serves as a contributor on several other projects, including OpenJDK; and is a member of the Java Community Process (JCP).

Nick currently lives in Tennessee with his wife Allison. You can find him on Twitter @Java_Nick.

ABOUT THE TECHNICAL EDITORS

JAKE RADAKOVICH joined UL Workplace Health and Safety in 2009, and currently serves as Software Developer on the Occupational Health Manager product. Prior to that, he was a research assistant at Middle Tennessee State University working on AlgoTutor, a web-based algorithm development tutoring system. He holds a BS in Computer Science and Mathematics from Middle Tennessee State University. You can follow Jake on Twitter @JakeRadakovich.

MANUEL JORDAN ELERA is an autodidactic developer and researcher who enjoys learning new technologies for his own experiments and creating new integrations. He won the 2010 Springy Award and was a Community Champion and Spring Champion in 2013. In his little free time, he reads the Bible and composes music on his guitar. Manuel is a Senior Member in the Spring Community Forums known as dr_pompeii. You can read about him and contact him through his blog and you can follow him on his Twitter account, @dr_pompeii.

CREDITS

ACQUISITIONS EDITOR
Mary James

PROJECT EDITOR
Maureen Spears Tullis

TECHNICAL EDITORS
Michael Jordan Elera
Jake Radakovich

TECHNICAL PROOFREADER
Jonathan Giles

SENIOR PRODUCTION EDITOR
Kathleen Wisor

COPY EDITOR
Apostrophe Editing Services

EDITORIAL MANAGER
Mary Beth Wakefield

FREELANCER EDITORIAL MANAGER
Rosemarie Graham

ASSOCIATE DIRECTOR OF MARKETING
David Mayhew

MARKETING MANAGER
Ashley Zurcher

BUSINESS MANAGER
Amy Knies

VICE PRESIDENT AND EXECUTIVE GROUP PUBLISHER
Richard Swadley

ASSOCIATE PUBLISHER
Jim Minatel

PROJECT COORDINATOR, COVER
Todd Klemme

PROOFREADERS
Nancy Carrasco
Josh Chase, Word One

INDEXER
Robert Swanson

COVER DESIGNER
Wiley

COVER IMAGE
iStockphoto.com/ElementalImaging

ACKNOWLEDGMENTS

THANKS TO...

My wife Allison, whose unwavering support and persistent reminders about deadlines during this stressful year made this book possible.

My parents and siblings, who told me that I could do anything I put my mind to.

Drs. Joyce Blair Crowell and William Hooper, whose dedicated instruction and mentoring made my career possible.

Dr. Sarah Ann Stewart, who believed in me when I thought surely calculus and proofs spelled doom for my education.

Mrs. Lockhart, who inspired me to write.

Jay, for introducing me to Mary, and to Mary and Maureen for making this book a reality.

Jake, for being absurd. Oh, and for agreeing to be my technical editor.

CONTENTS

INTRODUCTION *xxiii*

PART I: CREATING ENTERPRISE APPLICATIONS

CHAPTER 1: INTRODUCING JAVA PLATFORM, ENTERPRISE EDITION 3

A Timeline of Java Platforms 3
In the Beginning 4
The Birth of Enterprise Java 5
Java SE and Java EE Evolving Together 6
Understanding the Most Recent Platform Features 9
A Continuing Evolution 13
Understanding the Basic Web Application Structure 13
Servlets, Filters, Listeners, and JSPs 13
Directory Structure and WAR Files 14
The Deployment Descriptor 15
Class Loader Architecture 16
Enterprise Archives 17
Summary 18

CHAPTER 2: USING WEB CONTAINERS 19

Choosing a Web Container 19
Apache Tomcat 20
GlassFish 21
JBoss and WildFly 22
Other Containers and Application Servers 22
Why You'll Use Tomcat in This Book 23
Installing Tomcat on Your Machine 23
Installing as a Windows Service 24
Installing as a Command-Line Application 24
Configuring a Custom JSP Compiler 26
Deploying and Undeploying Applications in Tomcat 27
Performing a Manual Deploy and Undeploy 28
Using the Tomcat Manager 28

Debugging Tomcat from Your IDE 30
Using IntelliJ IDEA 30
Using Eclipse 35
Summary 39

CHAPTER 3: WRITING YOUR FIRST SERVLET 41

Creating a Servlet Class 42
What to Extend 42
Using the Initializer and Destroyer 45
Configuring a Servlet for Deployment 46
Adding the Servlet to the Descriptor 46
Mapping the Servlet to a URL 47
Running and Debugging Your Servlet 49
Understanding doGet(), doPost(), and Other Methods 51
What Should Happen during the service Method Execution? 51
Using HttpServletRequest 52
Using HttpServletResponse 55
Using Parameters and Accepting Form Submissions 56
Configuring your Application Using Init Parameters 61
Using Context Init Parameters 61
Using Servlet Init Parameters 62
Uploading Files from a Form 64
Introducing the Customer Support Project 65
Configuring the Servlet for File Uploads 65
Accepting a File Upload 68
Making Your Application Safe for Multithreading 69
Understanding Requests, Threads, and Method Execution 69
Protecting Shared Resources 70
Summary 71

CHAPTER 4: USING JSPS TO DISPLAY CONTENT 73

<br /> Is Easier Than output.println("<br />") 74
Why JSPs Are Better 75
What Happens to a JSP at Run Time 76
Creating Your First JSP 78
Understanding the File Structure 78
Directives, Declarations, Scriptlets, and Expressions 79
Commenting Your Code 81
Adding Imports to Your JSP 82

Using Directives 83
Using the <jsp> Tag 86
Using Java within a JSP (and Why You Shouldn't!) 88
Using the Implicit Variables in a JSP 88
Why You Shouldn't Use Java in a JSP 93
Combining Servlets and JSPs 94
Configuring JSP Properties in the Deployment Descriptor 94
Forwarding a Request from a Servlet to a JSP 97
A Note about JSP Documents (JSPX) 102
Summary 104

CHAPTER 5: MAINTAINING STATE USING SESSIONS 105

Understanding Why Sessions Are Necessary 106
Maintaining State 106
Remembering Users 107
Enabling Application Workflow 107
Using Session Cookies and URL Rewriting 107
Understanding the Session Cookie 108
Session IDs in the URL 110
Session Vulnerabilities 112
Storing Data in a Session 116
Configuring Sessions in the Deployment Descriptor 116
Storing and Retrieving Data 119
Removing Data 123
Storing More Complex Data in Sessions 125
Applying Sessions Usefully 129
Adding Login to the Customer Support Application 129
Detecting Changes to Sessions Using Listeners 133
Maintaining a List of Active Sessions 135
Clustering an Application That Uses Sessions 139
Using Session IDs in a Cluster 139
Understand Session Replication and Failover 141
Summary 142

CHAPTER 6: USING THE EXPRESSION LANGUAGE IN JSPS 143

Understanding Expression Language 144
What It's For 144
Understanding the Base Syntax 145
Placing EL Expressions 146
Writing with the EL Syntax 147

Reserved Keywords 148
Operator Precedence 148
Object Properties and Methods 154
EL Functions 155
Static Field and Method Access 156
Enums 157
Lambda Expressions 157
Collections 158
Using Scoped Variables in EL Expressions 160
Using the Implicit EL Scope 161
Using the Implicit EL Variables 165
Accessing Collections with the Stream API 167
Understanding Intermediate Operations 168
Using Terminal Operations 170
Putting the Stream API to Use 171
Replacing Java Code with Expression Language 172
Summary 175

CHAPTER 7: USING THE JAVA STANDARD TAG LIBRARY 177

Introducing JSP Tags and the JSTL 178
Working with Tags 178
Using the Core Tag Library (C Namespace) 182
<c:out> 182
<c:url> 183
<c:if> 184
<c:choose>, <c:when>, and <c:otherwise> 185
<c:forEach> 186
<c:forTokens> 187
<c:redirect> 188
<c:import> 188
<c:set> and <c:remove> 189
Putting Core Library Tags to Use 190
Using the Internationalization and Formatting Tag Library (FMT Namespace) 193
Internationalization and Localization Components 193
<fmt:message> 194
<fmt:setLocale> 196
<fmt:bundle> and <fmt:setBundle> 196
<fmt:requestEncoding> 197
<fmt:timeZone> and <fmt:setTimeZone> 197
<fmt:formatDate> and <fmt:parseDate> 198
<fmt:formatNumber> and <fmt:parseNumber> 199

Putting i18n and Formatting Library Tags to Use 200
Using the Database Access Tag Library (SQL Namespace) 203
Using the XML Processing Tag Library (X Namespace) 205
Replacing Java Code with JSP Tags 205
Summary 208

CHAPTER 8: WRITING CUSTOM TAG AND FUNCTION LIBRARIES 209

Understanding TLDs, Tag Files, and Tag Handlers 210
Reading the Java Standard Tag Library TLD 211
Comparing JSP Directives and Tag File Directives 217
Creating Your First Tag File to Serve as an HTML Template 219
Creating a More Useful Date Formatting Tag Handler 221
Creating an EL Function to Abbreviate Strings 226
Replacing Java Code with Custom JSP Tags 227
Summary 232

CHAPTER 9: IMPROVING YOUR APPLICATION USING FILTERS 233

Understanding the Purpose of Filters 234
Logging Filters 234
Authentication Filters 234
Compression and Encryption Filters 234
Error Handling Filters 235
Creating, Declaring, and Mapping Filters 235
Understanding the Filter Chain 235
Mapping to URL Patterns and Servlet Names 236
Mapping to Different Request Dispatcher Types 236
Using the Deployment Descriptor 237
Using Annotations 238
Using Programmatic Configuration 238
Ordering Your Filters Properly 239
URL Pattern Mapping versus Servlet Name Mapping 239
Exploring Filter Order with a Simple Example 241
Using Filters with Asynchronous Request Handling 243
Investigating Practical Uses for Filters 247
Adding a Simple Logging Filter 248
Compressing Response Content Using a Filter 249
Simplifying Authentication with a Filter 254
Summary 255

CHAPTER 10: MAKING YOUR APPLICATION INTERACTIVE WITH WEBSOCKETS 257

Evolution: From Ajax to WebSockets 258
Problem: Getting New Data from the Server to the Browser 259
Solution 1: Frequent Polling 259
Solution 2: Long Polling 260
Solution 3: Chunked Encoding 262
Solution 4: Applets and Adobe Flash 263
WebSockets: The Solution Nobody Knew Kind of Already Existed 264
Understanding the WebSocket APIs 268
HTML5 (JavaScript) Client API 268
Java WebSocket APIs 270
Creating Multiplayer Games with WebSockets 273
Implementing the Basic Tic-Tac-Toe Algorithm 274
Creating the Server Endpoint 274
Writing the JavaScript Game Console 278
Playing WebSocket Tic-Tac-Toe 283
Using WebSockets to Communicate in a Cluster 284
Simulating a Simple Cluster Using Two Servlet Instances 284
Transmitting and Receiving Binary Messages 286
Testing the Simulated Cluster Application 287
Adding “Chat with Support” to the Customer Support Application 288
Using Encoders and Decoders to Translate Messages 289
Creating the Chat Server Endpoint 291
Writing the JavaScript Chat Application 294
Summary 296

CHAPTER 11: USING LOGGING TO MONITOR YOUR APPLICATION 297

Understanding the Concepts of Logging 298
Why You Should Log 298
What Content You Might Want to See in Logs 299
How Logs Are Written 301
Using Logging Levels and Categories 303
Why Are There Different Logging Levels? 303
Logging Levels Defined 303
How Logging Categories Work 304
How Log Sifting Works 305
Choosing a Logging Framework 305
API versus Implementation 305

Performance 306
A Quick Look at Apache Commons Logging and SLF4J 307
Introducing Log4j 2 307
Integrating Logging into Your Application 312
Creating the Log4j 2 Configuration Files 313
Utilizing Fish Tagging with a Web Filter 316
Writing Logging Statements in Java Code 317
Using the Log Tag Library in JSPs 319
Logging in the Customer Support Application 319
Summary 320

PART II: ADDING SPRING FRAMEWORK INTO THE MIX

CHAPTER 12: INTRODUCING SPRING FRAMEWORK 323

What Is Spring Framework? 324
Inversion of Control and Dependency Injection 325
Aspect-Oriented Programming 325
Data Access and Transaction Management 325
Application Messaging 326
Model-View-Controller Pattern for Web Applications 326
Why Spring Framework? 326
Logical Code Groupings 326
Multiple User Interfaces Utilizing One Code Base 327
Understanding Application Contexts 327
Bootstrapping Spring Framework 329
Using the Deployment Descriptor to Bootstrap Spring 330
Programmatically Bootstrapping Spring in an Initializer 332
Configuring Spring Framework 336
Creating an XML Configuration 338
Creating a Hybrid Configuration 340
Configuring Spring with Java Using @Configuration 345
Utilizing Bean Definition Profiles 349
Understanding How Profiles Work 350
Considering Antipatterns and Security Concerns 352
Summary 353

CHAPTER 13: REPLACING YOUR SERVLETS WITH CONTROLLERS 355

Understanding @RequestMapping 356
Using @RequestMapping Attributes to Narrow Request Matching 356
Specifying Controller Method Parameters 360
Selecting Valid Return Types for Controller Methods 368