Spring Persistence
with Hibernate

Second Edition

Paul Fisher
Brian D. Murphy
# Contents at a Glance

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About the Authors</td>
<td>xi</td>
</tr>
<tr>
<td>About the Technical Reviewer</td>
<td>xiii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>xv</td>
</tr>
<tr>
<td>Preface</td>
<td>xvii</td>
</tr>
<tr>
<td>Chapter 1: Architecting Your Application with Spring, Hibernate,</td>
<td>1</td>
</tr>
<tr>
<td>and Patterns</td>
<td></td>
</tr>
<tr>
<td>Chapter 2: Spring Basics</td>
<td>17</td>
</tr>
<tr>
<td>Chapter 3: Basic Application Setup</td>
<td>35</td>
</tr>
<tr>
<td>Chapter 4: Persistence with Hibernate</td>
<td>55</td>
</tr>
<tr>
<td>Chapter 5: Domain Model Fundamentals</td>
<td>85</td>
</tr>
<tr>
<td>Chapter 6: Transaction Management</td>
<td>109</td>
</tr>
<tr>
<td>Chapter 7: Effective Testing</td>
<td>127</td>
</tr>
<tr>
<td>Chapter 8: Best Practices and Advanced Techniques</td>
<td>141</td>
</tr>
<tr>
<td>Index</td>
<td>161</td>
</tr>
</tbody>
</table>
Dependency Injection and Inversion of Control .............................................................. 24
  Setter-Based Dependency Injection .............................................................................. 24
  Constructor-Based Dependency Injection ...................................................................... 25
  Instance Collaboration .................................................................................................. 26
  Coding to Interfaces ..................................................................................................... 27
  Dependency Injection via Autowiring .......................................................................... 29
  Code-Based Dependency Injection ................................................................................ 29
  Set It and Forget It! ....................................................................................................... 32
Injecting Code Using AOP and Interceptors ................................................................. 33
Summary ......................................................................................................................... 34

Chapter 3: Basic Application Setup ......................................................................... 35

Application Management with Maven ........................................................................... 35
  Managed Dependencies ................................................................................................. 35
  Standard Directory Structure ...................................................................................... 37
  POM Deconstruction .................................................................................................... 37

Spring Configuration ..................................................................................................... 41
  Namespace Support ...................................................................................................... 43
  Externalizing Property Configurations ......................................................................... 44
  Component Scanning ..................................................................................................... 44
  Import Statements ........................................................................................................ 45

Database Integration ....................................................................................................... 45
  JDBC Support ............................................................................................................... 46
  Integration with JNDI .................................................................................................... 47

Web Application Configuration ...................................................................................... 49
  Servlet Definition ......................................................................................................... 51
  Spring MVC .................................................................................................................. 52

Summary ......................................................................................................................... 54
Chapter 4: Persistence with Hibernate
The Evolution of Database Persistence in Java
  EJB, JDO, and JPA
  How Hibernate Fits In
JPA Interface Hierarchy
The Audio Manager Domain Model and DAO Structure
  An @Entity-Annotated POJO
  Simplified DAO Pattern with Generics
  The Life Cycle of a JPA Entity
JPA Configuration
  Bare-Bones JPA Setup
  Spring Integration
  Querying and DAO Strategies
  Looking at the JPA Criteria API
  Using the JPA 2.0 Criteria API
  Using QueryDSL
  Integrating QueryDSL with Spring
Summary

Chapter 5: Domain Model Fundamentals
Understanding Associations
Building the Domain Model
  Polymorphism in JPA
  Convention over Configuration
  Managing Entity Identifiers
  Using Cascading Options to Establish Data Relationships
  Adding Second-Level Caching
  Using Polymorphism with Hibernate
Summary
### Chapter 6: Transaction Management

- The Joy of ACID ................................................................. 110
- Understanding Isolation Levels .................................................. 110
  - Serializable ............................................................................. 112
  - Repeatable Read ................................................................. 112
  - Read Committed ................................................................. 112
  - Read Uncommitted ............................................................... 112
- Controlling ACID Reflux............................................................. 113
  - Platform Transaction Management ........................................ 113
  - Declarative Transaction Management ..................................... 114
  - Programmatic Transaction Management ................................. 122
- Transactional Examples ............................................................. 123
  - Creating a Batch Application ............................................... 123
  - Using Two Datasources ....................................................... 124
- Summary .................................................................................. 125

### Chapter 7: Effective Testing

- Unit, Integration, and Functional Testing ................................. 127
- Using JUnit for Effective Testing .............................................. 129
  - Unit Testing with Mocks ...................................................... 131
  - Spring Dependency Injection and Testing ............................ 133
  - Integration Testing with a Database ..................................... 135
  - Integration Testing for RESTful APIs ................................. 137
- Summary .................................................................................. 139

### Chapter 8: Best Practices and Advanced Techniques

- Lazy Loading Issues ............................................................... 141
  - The N+1 Selects Problem ...................................................... 142
  - Lazy Initialization Exceptions ............................................... 147
Paul Tepper Fisher first began working in technology at Johns Hopkins University, where he spent several years developing a distance learning platform, while completing graduate school there. Currently, Paul is the CTO at Recombine, a genetics testing company in New York City with a mission to expand the scope and impact of genomics in medicine.

Before joining Recombine, Paul was the CTO at Onswipe, a mobile publishing platform, which was acquired by Beanstack Media in 2014. Prior to Onswipe, Paul was the CTO at Sonar Media — one of the first mobile social discovery platforms, which provided real-time notifications about relevant people nearby. Before joining Sonar, Paul was the CTO at K2 MediaLabs, a mobile-focused venture fund where he oversaw K2’s profile companies — including Sonar, Marketsharing, and Tracks.

Prior to K2, Paul was the Director of Engineering at Lime Wire, a Peer-to-Peer file-sharing company, where he led seven engineering teams for the development of a cloud-based Streaming Music Service. Previously, Paul managed the technology division for Wired Digital (owned by Condé Nast Publications), growing and overseeing the co-located development teams in both New York and San Francisco for Wired.com, Webmonkey.com, and howto.wired.com, helping wired.com triple its traffic to 12 million users.

In 2004, Paul founded DialMercury.com, a real-time communications and telephony platform. In 1998, Paul co-founded SmartPants Media, Inc., a software development company focused on distance learning, video streaming, and interactive products, winning numerous awards, including a coveted Muse Award for the creation of an educational software application built for the Smithsonian Institution.

Paul has co-written two technology books, both published by Apress: Spring Persistence — A Running Start, and Spring Persistence with Hibernate.

Paul lives in Brooklyn, New York with his wife Melanie and daughter Madeleine.

Brian D. Murphy has been enamored with computers and programming since he got his first computer, an Apple IIc, in 1984. He graduated from Rutgers University with a BS in computer science. He has focused on web development in a variety of settings ranging from early-stage startups to large, multinational corporations in fields covering e-commerce, consulting, finance, and media. He was an early adopter of Spring and Hibernate, and he has used both frameworks on large production systems since 2003.

In his present role, Brian is the chief architect and director of engineering at Condé Nast, where he oversees the web and mobile presence for 25 award-winning brands, such as WIRED, The New Yorker, Epicurious, and Vanity Fair. He and his team leverage both Spring and Hibernate to power all of Condé Nast’s online products, drawing tens of millions of unique visitors each month. Brian deals with the challenges of building and operating scalable, distributed systems every single day.

Brian lives in Maplewood, New Jersey, with his wife, Dania, son, Liam, and their dog, Cooper.
Vinay Kumar is a technology evangelist. He has extensive, eight-plus years’ experience in designing and implementing large-scale enterprise technology projects in various consulting and system integration companies. His passion helped him achieve certifications in Oracle ADF, WebCenter Portal, and Java/Java EE. Experience and in-depth knowledge has helped him evolve into a focused domain expert and a well-known technical blogger. He loves to spend time mentoring, writing technical blogs, publishing white papers, and maintaining a dedicated education channel on YouTube about ADF/WebCenter. In addition to experience in Java/Java EE, he is versed in various OpenStack technologies as well.

Vinay has contributed to the Java/Oracle ADF/WebCenter community by publishing more than 300 technical articles on his personal blog at www.techartifact.com. He was awarded Oracle ACE in June 2014. You can follow him at @vinaykuma201 or in.linkedin.com/in/vinaykumar2.
Acknowledgments

Writing a book always ends up being more difficult than you initially imagined. Although the absurdly late nights and lost weekends prove difficult to the authors, it is often the people around them that end up suffering the most. To that end, I’d like to thank Melanie Colton for her endless patience and perseverance. She deserves more than a medal for putting up with the many 4AM nights and my noisy typing. This book would not have been possible without her support and understanding. I also want to thank my amazing daughter, Madeleine — although she’s too young to help copy-edit (or read) any chapters in this book, she inspires me every single day.

I would also like to acknowledge everyone I work with at Recombine for their continued trust and support. I consider myself lucky to have the opportunity to work with such a talented and dedicated team. I am grateful to be a part of such an important adventure.

Finally, my sincere appreciation goes to Brian Murphy for joining the project and keeping things rolling along. If it hadn’t been for Brian’s tenacity and motivation, this book would never have seen the light of day. It’s been an honor and privilege working with you again.

—Paul Tepper Fisher

We’d like to thank Apress for the opportunity to write this book. Special thanks to Steve Anglin for believing in us and letting us stretch the schedule to cover advanced topics in depth. We owe Mark Powers a special debt of gratitude for shepherding us through this process and ultimately dragging us across the finish line. Thanks to Matt Moodie, Marilyn Smith, and Sia Cyrus, who provided invaluable feedback, suggestions, and encouragement along the way. This is a much better book as a result of their wisdom and patience. Any issues or errors in this text are ours alone.

I would like to thank my wife, Dania, without whom this book wouldn’t be possible. She graciously took on the role of super mom while I devoted nights and weekends to writing for far longer than bargained for. I’d like to thank my son Liam for being the most terrific little kid. You provide me with more joy and a new appreciation for the world than you’ll ever know. I’d also like to acknowledge our second son, who is due shortly after this book will be published. I can’t wait to meet you!

Lastly, I’d like to thank Paul Fisher for sharing this experience with me. This book was Paul’s brainchild and I’m glad he invited me along for the ride. Writing this book has been both rewarding and challenging. I learned a ton and it’s been great to work with you again.

—Brian D. Murphy
Preface

Since its inception, the Spring Framework has gradually changed the rules of application development in the Java community. This book is the ideal guide and teaching companion for developers interested in learning about the Spring Framework and how it can be leveraged to build persistence-driven applications using Hibernate, one of the most popular Java persistence frameworks today. *Spring Persistence with Hibernate* gets you rolling with fundamental Spring concepts, as well as proven design patterns for integrating persistence into your applications.

Many of the lessons illustrated in this book were culled from years of practical experience building scalable, high-volume web applications using Spring and Hibernate. One of the details that stands out in our joint experience is the importance and benefit of learning through hands-on experience. To this end, we will build a real-world application that utilizes Spring 4, Hibernate 5, Spring-Data, JPA 2.1, and Query-DSL. We firmly believe that learning about Spring and Hibernate implies far more than simply understanding the respective APIs of each framework. To be able to effectively develop with these two amazing technologies, it is necessary to understand the design patterns and best practices for getting the most from these frameworks, and building on them in a consistent, proven manner. We hope that this book teaches you more than just how to use Spring and Hibernate together. Our goal is to channel the development experience, lessons, and best practices we’ve seen work successfully in our experience, so that you can apply these skills and tools in your own applications.

Throughout these pages, we introduce core Hibernate fundamentals, demonstrating how the framework can be best utilized within a Spring context. We start with foundational concepts, such as strategies for developing an effective domain model and DAO layer, and then move into querying techniques using HQL, JPQL, Spring-Data, and Query-DSL (a powerful framework that offers a flexible, generic, and type-safe query abstraction). After fundamental concepts are introduced, we move on to more advanced topics, such as fetching and caching strategies. We also illustrate several approaches for architecting a transactional service facade. Both programmatic and declarative transactions are examined, showcasing the benefits of using Spring for expressing transactional semantics.

*Spring Persistence with Hibernate* also introduces JPA, covering its history and the ways in which Hibernate influenced its development. We discuss the benefits of following the JPA standard, as well as when it makes sense to utilize Hibernate-specific features. The book also examines different strategies and best-practices for architecting your persistence tier, such as illustrating the differences between the DAO and Active Record patterns. Throughout this book, we explore topics related to concurrency/optimistic locking, Hibernate Session state, caching approaches, and transaction management.

The last part of the book introduces several advanced techniques, important for working with enterprise Spring/Hibernate applications. We illustrate some of the pitfalls with integrating legacy databases, as well as best practices for developing REST web services, handling Hibernate proxies and lazy collections, and proven patterns that will prove valuable for any database-driven project running on the JVM.
Here are some of the main topics that we discuss in this book:

- Basic Spring Framework features such as IoC and AOP
- Core concepts for architecting a well-layered persistence tier
- JPA concepts and steps for integrating JPA
- Foundational and advanced concepts for working with Hibernate
- Hibernate querying techniques
- DAO and Service Facade layer development
- Building a REST web service
- Understanding the DTO pattern
- Leveraging other frameworks and technologies, such as Query-DSL
- Advanced caching and integration strategies