

Design patterns using Spring and Guice

Dependency Injection

Dhanji R. Prasanna

Praise from the Creator of Guice





Dependency Injection

by Dhanji R. Prasanna

[www.manning.com/Dependency in Action](http://www.manning.com/Dependency%20in%20Action)

Copyright 2009 Manning Publications

Praise from the Creator of Guice

Dhanji lives on the bleeding edge. He's been around Google Guice since the beginning. Less than a week after we released Guice 1.0, Dhanji refactored a non-trivial Swing application to use Guice. A ThoughtWorks consultant at the time, he measured dramatic design improvements and performance and reliability gains. As a result, Dhanji sent the Guice team some of our first kudos, not to mention one of our first external feature requests.

Building on his extensive experience developing enterprise Java standards, Dhanji went on to receive his own praise when he released Warp Persist and Warp Servlets, two essential Guice extensions that respectively integrate with standard Java persistence and web APIs. His Warp frameworks seamlessly extend Guice's fluent, plain-Java configuration style to the enterprise.

Warp Persist provides JPA integration and declarative transaction support, two indispensable components of a modern enterprise Java stack. Warp Persist has no doubt been key to Guice's success in this space.

Dhanji became a key Guice team member when, in Guice 2, Warp Servlets supplanted Guice's default servlet module. Nowadays, Dhanji works with us at Google on one of the most notable Guice-based applications: Google Wave.

JavaOne attendees voted Dhanji a Rock Star speaker and his popular presentation style translates well to print. Dhanji's unique combination of experience as both a framework designer and an in-the-trenches dependency injection user results in deep yet accessible explanations. Advice like his comes only from years of eating one's own dog food.

This book is not a reference. While most dependency injection documentation focuses on the mechanics of individual frameworks, this book covers customary and effective use of dependency injection, especially with respect to application design. It fills a much needed gap.

While I'm obviously biased toward Guice, I'm happy to see that this book transcends implementation details and covers what will no doubt continue to become an essential part of day-to-day Java development. One day, we'll look at dependency injection support as just another language feature, a construct for importing instances. This book prepares you for that day.

BOB LEE
SOFTWARE ENGINEER, GOOGLE INC.
AND CREATOR OF GUICE

Dependency Injection DESIGN PATTERNS USING SPRING AND GUICE

Dhanji R. Prasanna

In object-oriented programming, a central program normally controls other objects in a module, library, or framework. With dependency injection, this pattern is inverted—a reference to a service is placed directly into the object, which eases testing and modularity. Spring or Google Guice use dependency injection so you can focus on your core application and let the framework handle infrastructural concerns.

Dependency Injection explores the DI idiom in fine detail, with numerous practical examples that show you the payoffs. You'll apply key techniques in Spring and Guice and learn important pitfalls, corner-cases, and design patterns. Readers need a working knowledge of Java, but no prior experience with DI is assumed.

What's Inside

- How to apply it (understand it first!)
- Design patterns and nuances
- Spring, Google Guice, PicoContainer ...
- How to integrate DI with Java frameworks

Dhanji R. Prasanna is a Google software engineer who works on Google Wave and represents Google on several Java expert groups. He contributes to Guice, MVEL, and other open source projects.

For online access to the author, code samples, and a free ebook for owners of this book, go to manning.com/DependencyInjection



“Transcends implementation details ... covers customary and effective use of DI.”

—Bob Lee
Software Engineer, Google Inc.
and Creator of Guice

“Most comprehensive coverage of DI I have seen.”

—Frank Wang, Chief Software
Architect, DigitalVelocity LLC

“A handy manual for better programs with less code.”

—Jesse Wilson
Guice 2.0 Lead, Google Inc.

“Dependency injection is not just for gurus—this book explains all.”

—Paul King, Director, ASERT

“A fantastic book ... makes writing great software much easier.”

—Rick Wagner, Enterprise Architect
Axiom Data Products

ISBN 13: 978-1-933988-55-9
ISBN 10: 1-933988-55-X

