

The simply functional web framework for Scala

Lift IN ACTION

Covers Lift 2.x

Timothy Perrett

 MANNING



Lift in Action

Lift in Action

THE SIMPLY FUNCTIONAL
WEB FRAMEWORK FOR SCALA

TIMOTHY PERRETT



MANNING
SHELTER ISLAND

*To my Dad
for teaching me that hard work and dedication
can triumph over any problem*

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact


Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964
Email: orders@manning.com

©2012 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

© Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

 Manning Publications Co.
20 Baldwin Road
PO Box 261
Shelter Island, NY 11964

Development editors: Katharine Osborne
Copyeditor: Andy Carroll
Typesetter: Dennis Dalinnik
Cover designer: Marija Tudor

ISBN: 9781935182801

Printed in the United States of America

1 2 3 4 5 6 7 8 9 10 – MAL – 17 16 15 14 13 12 11

brief contents

PART 1	GETTING STARTED	1
	1 ■ Introducing Lift	3
	2 ■ Hello Lift	20
PART 2	APPLICATION TUTORIAL	37
	3 ■ The auction application	39
	4 ■ Customers, auctions, and bidding	60
	5 ■ Shopping basket and checkout	83
PART 3	LIFT IN DETAIL.....	105
	6 ■ Common tasks with Lift WebKit	107
	7 ■ SiteMap and access control	140
	8 ■ HTTP in Lift	160
	9 ■ AJAX, wiring, and Comet	187
	10 ■ Persistence with Mapper	223
	11 ■ Persistence with Record	259

- 12 ■ Localization 282
- 13 ■ Distributed messaging and
Java enterprise integration 293
- 14 ■ Application testing 317
- 15 ■ Deployment and scaling 347

contents

preface xiii
acknowledgments xv
about this book xvii
about the author xxi
about the cover illustration xxii

PART 1 GETTING STARTED.....1

1 Introducing Lift 3

1.1 What is Scala? 4

1.2 What is Lift? 6

Lift design goals 7 ▪ *View-first design* 8

Community and team 10

1.3 Lift features 11

Lift Core and Lift Web 12 ▪ *Lift Persistence* 15

Lift Modules 17

1.4 Summary 18

2 *Hello Lift* 20

- 2.1 Getting started with SBT 21
- 2.2 Your first Lift application 23
 - Creating the project* 23
 - *Inspecting the project* 26
 - Booting the application* 29
- 2.3 Snippets and templating overview 31
 - Snippets* 31
 - *Templating overview* 33
- 2.4 Summary 35

PART 2 APPLICATION TUTORIAL37

3 *The auction application* 39

- 3.1 Application requirements 40
 - Frontend* 40
 - *Administration* 41
- 3.2 Template structure 43
 - Design workflow* 43
 - *Template setup* 43
- 3.3 Data models 46
 - Schema definition* 46
 - *Connecting to the database* 49
- 3.4 Scaffolding 50
 - Prototype traits* 50
 - *CRUD generation* 53
- 3.5 Validation 57
 - Definitions* 58
 - *Displaying field errors* 58
- 3.6 Summary 59

4 *Customers, auctions, and bidding* 60

- 4.1 Building an auction catalog 61
 - Listing auctions* 61
 - *Adding to SiteMap* 65
- 4.2 Displaying auctions 66
 - Auction detail URLs* 66
 - *The AJAX bidding interface* 68
 - *Real-time bidding* 74
- 4.3 Summary 82

5 *Shopping basket and checkout* 83

- 5.1 Order creation 84
 - Order models* 84 ▪ *Attributing auctions to customers* 87
- 5.2 Implementing the basket and checkout process 91
 - Implementing the basket* 91 ▪ *Implementing the checkout* 93
- 5.3 Collecting payment with PayPal 99
 - Environment setup* 99 ▪ *The Buy Now button* 102
- 5.4 Summary 103

PART 3 LIFT IN DETAIL 105

6 *Common tasks with Lift WebKit* 107

- 6.1 Templates, snippets, and views 108
 - Templates* 108 ▪ *Snippets* 114 ▪ *Views* 125
- 6.2 Managing state 128
 - Request and session state* 128 ▪ *Cookies* 130
- 6.3 Forms with LiftScreen and Wizard 131
 - LiftScreen* 132 ▪ *Wizard* 135
- 6.4 Widgets 137
 - AutoComplete widget* 137 ▪ *Gravatar widget* 138
- 6.5 Summary 139

7 *SiteMap and access control* 140

- 7.1 Menus and locations 141
 - Understanding and implementing locations* 143
 - Rendering menus* 143
- 7.2 Location parameters 146
 - Default location parameters* 147
 - Authentication parameters* 150
- 7.3 Customizing SiteMap 153
 - Creating a custom Loc* 153 ▪ *When to customize SiteMap?* 158
- 7.4 Summary 159

8 *HTTP in Lift* 160

- 8.1 HTTP pipeline 161
 - HTTP abstraction* 161
 - Application lifecycle* 162
 - Request lifecycle* 166
- 8.2 URL rewriting 170
 - Defining a RewritePF* 171
 - Advanced rewriting* 173
- 8.3 Dispatching and web services 174
 - Using the HTTP dispatch DSL* 176
 - Basic REST service* 179
 - Advanced multiformat REST service* 182
- 8.4 Summary 185

9 *AJAX, wiring, and Comet* 187

- 9.1 AJAX 188
 - JavaScript abstractions* 189
 - AJAX 101* 194
 - Sophisticated AJAX* 197
 - Using JSON forms with AJAX* 200
 - AJAX with LiftScreen* 202
- 9.2 Wiring 203
 - Formula wiring* 203
- 9.3 Comet 207
 - What are actors?* 208
 - Basic Comet usage* 209
 - Comet-based rock-paper-scissors* 212
- 9.4 Summary 221

10 *Persistence with Mapper* 223

- 10.1 Setting up a database 224
 - Installation and connectivity* 224
 - Defining Mappers* 227
 - Relationships* 232
 - Schema creation and control* 235
- 10.2 Interacting with Mapper 237
 - Creating data* 237
 - Querying data* 238
 - Updating and deleting data* 244
 - Validation and lifecycle callbacks* 245
 - Display functionality* 248
- 10.3 Advanced Mapper 251
 - Query logging* 251
 - Transactions* 252
 - Custom mapped fields* 254
- 10.4 Summary 258

-
- 11 Persistence with Record 259**
- 11.1 Common Record functionality 260
 - Common Record fields* 262 ▪ *Integration with LiftScreen and Wizard* 265
 - 11.2 Record for relational databases 266
 - Connecting and querying with Squeryl* 266 ▪ *A bookstore with Squeryl* 268
 - 11.3 Record for NoSQL stores 273
 - NoSQL support in Lift* 273 ▪ *Bookstore with MongoDB* 278
 - 11.4 Summary 280
- 12 Localization 282**
- 12.1 Implementing localization 283
 - Implementing locale calculator* 284 ▪ *Localizing templates and code* 286
 - 12.2 Defining localized resources 289
 - Using XML resources* 289 ▪ *Using Java properties resources* 290
 - Using custom resource factories* 291
 - 12.3 Summary 292
- 13 Distributed messaging and Java enterprise integration 293**
- 13.1 Distributed programming 294
 - Messaging with AMQP* 295 ▪ *Messaging with Akka* 300
 - 13.2 Integrating Lift into existing Java infrastructure 309
 - JPA and Scala EntityManager* 309
 - 13.3 Summary 316
- 14 Application testing 317**
- 14.1 Scala testing frameworks 318
 - ScalaTest* 318 ▪ *Scala Specs* 319 ▪ *ScalaCheck* 321
 - Code coverage reports* 322
 - 14.2 Writing testable code 324
 - Complexities of testing state* 325 ▪ *Dependency injection* 327

- 14.3 Testing strategies 334
 - Testing snippets* 334
 - *Testing web services* 336
 - Testing with Mapper* 340
 - *Testing Comet and AJAX* 341
- 14.4 Summary 345

15 *Deployment and scaling* 347

- 15.1 Choosing a servlet container 348
 - 15.2 Handling state 351
 - Sticky session strategies* 352
 - *Distributing critical state* 356
 - 15.3 Choosing a configuration 361
 - Single server* 362
 - *Multiple servers* 363
 - 15.4 Deployment tools and techniques 364
 - Built-in assistance* 364
 - *Monitoring* 367
 - 15.5 Case studies 371
 - Foursquare* 372
 - *Novell Vibe* 372
 - 15.6 Summary 373
- appendix A Introduction to Scala* 375
- appendix B Configuring an IDE* 384
- appendix C Options and boxes* 388
- index* 393

preface

The web has completely revolutionized the way we live our lives—the average person in the UK now does an average of six Google searches a day. Within the lifetime of one generation, our entire society has changed, and it continues to be catalyzed by technology in a very fundamental way. For me, this is the most fascinating thing to observe and an even more interesting thing to be a part of.

The web development industry has seen sweeping change over the past five or six years as it has attempted to cope with these new social habitats and behaviors. Probably one of the most notable changes was the way in which Ruby on Rails altered developers' outlook toward building applications and the manner in which they approached problems. Massive enterprise architecture was out the window and small, iterative, agile processes became all the rage. At the beginning of 2006, I had been coding Ruby on Rails for quite some time and had built several large systems with the Ruby stack. Although I was blown away by the productivity gains that Rails supplied, taking code to production was a comparative nightmare. I specifically recall Zed Shaw's "Rails is a Ghetto" rant and how that was very similar to my own views at the time. It was then that I started to look for something else, something new.

Before long, I came across Lift, which felt "right" from the very beginning. Scala and Lift's elegant fusion of the functional and object-oriented paradigms was a breath of fresh air when compared to other languages and frameworks. It was great to have all the security features baked right into a framework, and not have to worry about many things that typically cause a lot of headaches for developers. These kinds of choices make a great developer-oriented framework: focusing on removing work from the developer in a pragmatic and logical way while using as little runtime magic as possible.

Having been involved with Lift from an early stage, seeing it grow and evolve in an organic fashion has been very rewarding. Even with an intimate understanding of Lift, writing this book has been far more difficult than I could have ever anticipated. As a framework, Lift is growing at an exponential rate, and I've tried to cover as much of it as possible and keep it up-to-date with the latest advancements, all while providing you with a base from which to understand the *Lift way* of solving problems.

acknowledgments

Many people contributed to this book, both in the tangible sense of giving reviews and feedback, and also in a more intangible regard by giving me the encouragement and positive words to continue with the project, even when there was seemingly no end in sight.

Throughout the course of writing, I was fortunate enough to receive feedback from a wide range of sources, but there are several people that I specifically need to single out and thank. First, I would like to thank Jon-Anders Teigen and Ross Mellgren for being such amazing sounding boards for ideas, and for often providing a much-needed sanity check late at night. In addition, I would like to thank the following people from the Scala community who have had an influence on me during the writing of this book; your blogs, screencasts, and personal discussions have been a source of inspiration and always remind me there is so much more to learn: Martin Odersky, Debasish Ghosh, Tony Morris, Rúnar Bjarnason, Mark Harrah, Jeppe Nejsum Madsen, Jeppe Cramon, Vassil Dichev, Marius Danicu, Derek Chen-Becker, Jorge Ortiz, and Josh Suereth.

I would also like to thank the companies that use Scala commercially and who have constructively given their feedback; particular thanks go to Harry Heymann and all the Foursquare engineers, Daniel Spiewak and David LaPalomente at Novel, Steve Jenson and Robey Pointer at Twitter, and Jonas Bonér and Viktor Klang at Typesafe.

Writing *Lift in Action* has without doubt been one of the most difficult things I've ever done, and it's been a huge personal challenge. During the writing of this book, I've circumnavigated the globe nearly twice, severely broken my hand, learned Italian,

and still found time for a day job. None of those things would have been achievable without the support of my family and three best friends: Robert Dodge, Paul Dredge, and Michael Edwards. I simply couldn't wish for closer friends or a more supportive family. You guys are awesome.

I'd also like to say thank you to all the amazing people who have contributed to Lift over the years, and also to David Pollak for founding the project in the first place. Working on Lift and being a part of the community has truly been one of the highlights of my career to date.

The team at Manning has also been a huge, huge help. Working with such a professional group of people has been a joy end-to-end. I would specifically like to thank Michael Stephens for bringing me on board to write this book: his words from our first call together, "...writing a book is completely survivable," are something I have thought about often. Additionally, Katharine Osbourne has been a legendary development editor; without her support and consultation, this book would likely have never made it to completion. Thanks also to the production team of Andy Carroll, Melody Dolab, Dennis Dalinnik, and Mary Piergies; and to Jon Anders Teigen, Graham Tackley, and Phil Wells for their careful technical proofread of the manuscript, shortly before it went to press.

Finally, my thanks to the reviewers who read the manuscript numerous times during development and who provided invaluable feedback: Andy Dingley, Paul Stusiak, Guillaume Belrose, John Tyler, Ted Neward, Andrew Rhine, Jonas Bandi, Tom Jensen, Ross Mellgren, Richard Williams, Viktor Klang, and Dick Wall.

about this book

Lift is an advanced, next-generation framework for building highly interactive and intuitive web applications. Lift aims to give you a toolkit that scales with both your needs as a developer and the needs of your applications. Lift includes a range of features right out of the box that set it apart from other frameworks in the marketplace: namely security, statefulness, and performance.

Lift also includes a range of high-level abstractions that make day-to-day development easy and powerful. In fact, one of the main driving forces during Lift's evolution has been to include only features that have an actual production use. You, as the developer, can be sure that the features you find in Lift are distilled from real production code.

Lift in Action is a step-by-step exploration of the Lift web framework, and it's split into two main parts: chapters 1 through 5 introduce Lift and walk you through building a small, sample application, and then chapters 6 through 15 take a deep dive into the various parts of Lift, providing you with a deep technical reference to help you get the best out of Lift.

Roadmap

Chapter 1 introduces Lift and sets the scene with regard to how it came into existence. It also covers the various modules of the framework to give you an appreciation for the bigger picture.

Chapter 2 shows you how to get up and running with the Scala build tool SBT and start making your first web application with Lift. This chapter focuses on small, incremental steps covering the concepts of development that you'll need in the rest of the book.

Chapter 3, 4, and 5 walk you through the construction of a real-time auction application to cover as many different parts of Lift as possible. This includes creating templates, connecting to a database, and implementing basic AJAX and Comet.

Chapter 6 takes a dive into the practical aspects of Lift WebKit, showing you how to work with the sophisticated templating system, snippets, and form building through `LiftScreen` and `Wizard`. Additionally, this chapter introduces Lift's own abstraction for handling application state in the form of `RequestVar` and `SessionVar`. This chapter concludes with an overview of some useful extension modules, known as widgets, that ship with the Lift distribution.

Chapter 7 focuses on Lift's SiteMap feature, which allows you to control access and security for particular resources.

Chapter 8 covers the internal working of Lift's HTTP pipeline, detailing the various hooks that are available and demonstrating several techniques for implementing HTTP services.

Chapter 9 explores Lift's sophisticated AJAX and Comet support, demonstrating these technologies in practice by assembling a *rock-paper-scissors* game. This chapter also covers Lift's AJAX abstraction called *wiring*, which allows you to build chains of AJAX interaction with ease.

Chapters 10 and 11 cover Lift's persistence systems, `Mapper` and `Record`. `Mapper` is an active-record style object-relational mapper (ORM) for interacting with SQL data stores, whereas `Record` is store-agnostic and can be used with any backend system from MySQL to modern NoSQL stores such as MongoDB.

Chapter 12 demonstrates Lift's localization toolkit for building applications that can work seamlessly in any language. This includes the various ways in which you can hook in your `ResourceBundles` to store localized content.

Chapter 13 is all about the enterprise aspects often associated with web application development. Technologies such as JPA are prevalent within the enterprise space, and companies often want to reuse them, so this chapter shows you how to implement JPA with Lift. Additionally, this chapter covers messaging using the Akka framework.

Chapter 14 covers testing with Lift and shows you some different strategies for testing snippets. More broadly, it demonstrates how to design code that has a higher degree of decoupling, so your general coding lends itself to testing.

Finally, chapter 15 consolidates all that you've read in the book and shows you how to take your application into production. This includes an overview of various servlet containers, a demonstration of implementing distributed state handling, and a guide to monitoring with Twitter Ostrich.

Who should read this book?

Primarily, this book is intended to demonstrate how to get things done using Lift. With this in mind, the book is largely slanted toward users who are new to Lift, but who have experience with other web development frameworks. Lift has its own unique way of doing things, so some of the concepts may seem foreign, but I make conceptual

comparisons to things you may be familiar with from other popular frameworks or libraries to smooth the transition.

If you're coming to Lift with little or no knowledge of Scala, you should know that Lift makes use of many Scala language features. This book includes a Scala rough guide to get you up and running within the context of Lift as quickly as possible.

The book largely assumes that you have familiarity with XML and HTML. Lift's templating mechanism is 100 percent based on XML, and although it's straightforward to use, it's useful to have an understanding of structured XML that makes use of namespaces.

Finally, because Lift is primarily a web framework designed for browser-based experiences, JavaScript is inevitably part of the application toolchain. Lift includes a high-level Scala abstraction for building JavaScript expressions, but having an understanding of JavaScript and client-side scripting can greatly improve your understanding of the client-server interactions supplied by Lift.

Code conventions and examples

This book includes a wide range of examples and code illustrations from Scala code and HTML templates, to plain text configurations for third-party products. Source code in the listings and in the text is presented in a fixed width font to separate it from ordinary text. Additionally, Scala types, methods, keywords, and XML-based markup elements in text are also presented using fixed width font. Where applicable, the code examples explicitly include import statements to clarify which types and members originate from which packages. In addition, functions and methods have explicitly annotated types where the result type is not clear.

Although Scala code is typically quite concise, there are some listings that needed to be reformatted to fit in the available page space in the book. You are encouraged to download the source code from the online repository, in order to see the sample code in its original form (<https://github.com/timperrett/lift-in-action>). In addition to some reformatting, all the comments have been removed for brevity. You can also download the code for the examples in the book from the publisher's website at www.manning.com/LiftinAction.

Code annotations accompany many of the source code listings, highlighting important concepts. In some cases, numbered bullets link to explanations in the subsequent text.

Lift itself is released under the Apache Software License, version 2.0, and all the source code is available online at the official Github repository (<https://github.com/lift/framework/>). Reading Lift's source code can greatly speed your efforts at becoming productive in using Lift for your own applications.

Author Online

Purchase of *Lift in Action* includes free access to a private web forum run by Manning Publications where you can make comments about the book, ask technical

questions, and receive help from the author and from other users. To access the forum and subscribe to it, point your web browser to www.manning.com/LiftinAction or www.manning.com/perrett. This page provides information on how to get on the forum once you're registered, what kind of help is available, and the rules of conduct on the forum.

Manning's commitment to our readers is to provide a venue where a meaningful dialog between individual readers and between readers and the author can take place. It's not a commitment to any specific amount of participation on the part of the author, whose contribution to the AO remains voluntary (and unpaid). We suggest you try asking the author some challenging questions lest his interest stray!

The Author Online forum and the archives of previous discussions will be accessible from the publisher's website as long as the book is in print.

about the author

Timothy Perrett is a technical specialist at a business unit of Xerox Corporation and has been a member of the Lift core team since early 2008. He has a wealth of experience programming in different languages and platforms but has now settled on Scala as his language (and community) of choice for nearly all production activities. Timothy is a specialist in enterprise integration and automation systems for manufacturing and marketing workflows.

When not speaking at conferences or blogging about Scala and Lift, Timothy lives by the river in the beautiful city of Bath, England, where he enjoys socializing with friends and drinking the local ale.



about the cover illustration

The figure on the cover of *Lift in Action* is captioned “A Water Carrier.” The illustration is taken from a 19th-century edition of Sylvain Maréchal’s four-volume compendium of regional dress customs published in France. Each illustration is finely drawn and colored by hand. The rich variety of Maréchal’s collection reminds us vividly of how culturally apart the world’s towns and regions were just 200 years ago. Isolated from each other, people spoke different dialects and languages. In the streets or in the countryside, it was easy to identify where they lived and what their trade or station in life was just by their dress.

Dress codes have changed since then and the diversity by region, so rich at the time, has faded away. It is now hard to tell apart the inhabitants of different continents, let alone different towns or regions. Perhaps we have traded cultural diversity for a more varied personal life—certainly for a more varied and fast-paced technological life.

At a time when it is hard to tell one computer book from another, Manning celebrates the inventiveness and initiative of the computer business with book covers based on the rich diversity of regional life of two centuries ago, brought back to life by Maréchal’s pictures.

Part 1

Getting started

The first two chapters of this book introduce the Lift framework and demonstrate how you can get everything set up and ready for your first development.

Chapter 1 starts by introducing both Scala and Lift concepts, complete with high-level explanations and samples. The aim is to give you grounding in what is a fundamentally different way of thinking. In chapter 2, you'll be building upon the basis laid down in chapter 1 by constructing your very first Hello World application, which will involve the most basic Lift steps. In these chapters, you'll see first-hand how Lift leverages a *view-first architecture* and how easy it is to get up and running with the Lift web framework.

- [download online Making Sense of Behavior: The Meaning of Control pdf, azw \(kindle\), epub](#)
- [Listening to Music pdf, azw \(kindle\)](#)
- [On Aristotle On the Heavens 3.7-4.6 \(Ancient Commentators on Aristotle\) book](#)
- [download online I'm Just Here for the Food: Version 2.0 for free](#)
- [read **Fighting in the streets: A manual of urban guerrilla warfare** book](#)
- [click Convertibles pdf, azw \(kindle\), epub, doc, mobi](#)

- <http://betsy.wesleychapelcomputerrepair.com/library/Pilgrim-p---havet.pdf>
- <http://fortune-touko.com/library/Torchwood--Exodus-Code.pdf>
- <http://nexson.arzamaszev.com/library/Mini-Manual-of-the-Urban-Guerilla.pdf>
- <http://schrolf.de/books/I-m-Just-Here-for-the-Food--Version-2-0.pdf>
- <http://ramazotti.ru/library/Dragonflies-and-Damselflies-of-the-West--Princeton-Field-Guides-.pdf>
- <http://paulczajak.com/?library/The-Crime-of-My-Very-Existence--Nazism-and-the-Myth-of-Jewish-Criminality.pdf>