

Beginning Ruby on Rails E-Commerce

From Novice to Professional



Christian Hellsten and Jarkko Laine

Beginning Ruby on Rails E-Commerce: From Novice to Professional

Copyright © 2006 by Christian Hellsten and Jarkko Laine

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-736-1

ISBN-10 (pbk): 1-59059-736-2

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

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

Lead Editor: Keir Thomas

Technical Reviewer: Peter Marklund

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick,

Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade

Project Manager: Beth Christmas

Copy Edit Manager: Nicole Flores

Copy Editor: Marilyn Smith

Assistant Production Director: Kari Brooks-Copony

Production Editor: Kelly Winqvist

Composer: Pat Christenson

Proofreader: Dan Shaw

Indexer: Broccoli Information Management

Artist: Kinetic Publishing Services, LLC

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 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 <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Source Code/Download section.

Contents at a Glance

About the Authors	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
Introduction	xix
CHAPTER 1 Project Setup and Proof of Concept	1
CHAPTER 2 Author Management	29
CHAPTER 3 Book Inventory Management	59
CHAPTER 4 Book Catalog Browsing	113
CHAPTER 5 Shopping Cart Implementation	141
CHAPTER 6 Forum Implementation	169
CHAPTER 7 Tagging Support	197
CHAPTER 8 Security	223
CHAPTER 9 Checkout and Order Processing	251
CHAPTER 10 Multiple Language Support	297
CHAPTER 11 Acceptance Testing	327
CHAPTER 12 Application Deployment	351
CHAPTER 13 Performance Optimization	381
INDEX	403

Contents

About the Authors	xiii
About the Technical Reviewer.....	xv
Acknowledgments.....	xvii
Introduction	xix
CHAPTER 1 Project Setup and Proof of Concept	1
Introducing the Emporium Project.....	1
Installing the Software.....	2
Installing Ruby.....	4
Installing RubyGems.....	5
Installing Ruby on Rails.....	6
Installing MySQL.....	8
Installing the MySQL Driver.....	9
Introducing Scrum.....	10
Creating the Emporium Application.....	12
Creating the Skeleton Application.....	12
Creating the Emporium Database.....	14
Starting Emporium for the First Time.....	18
How Does Ruby on Rails Work?.....	20
Implementing the About Emporium User Story.....	20
Running the Generate Script.....	21
Modifying the Generated View.....	22
Creating the Layout.....	23
Modifying the Generated Controller.....	27
Summary.....	28

CHAPTER 2	Author Management	29
	Using Test-Driven Development	29
	Testing in Rails	30
	Unit Testing	30
	Functional Testing	31
	Integration Testing	31
	Creating the ActiveRecord Model	31
	Using ActiveRecord Migrations	32
	Running Unit Tests	36
	Creating the Controller	37
	Implementing the User Stories	39
	Adding an Author	39
	Listing Authors	48
	Viewing an Author	50
	Editing an Author	52
	Deleting an Author	54
	Adjusting the Flash Notifications	55
	Summary	57
CHAPTER 3	Book Inventory Management	59
	Getting the Requirements	59
	Using Scaffolding	60
	Implementing the Publisher Administration Interface	61
	Updating the Schema with the Publishers Table	61
	Generating Publisher Code with the Scaffolding Script	62
	Completing the Add Publisher User Story	64
	Completing the View Publisher User Story	66
	Completing the Edit Publisher User Story	68

Implementing the Book Administration Interface	69
Updating the Schema with the Books Table	69
Creating the Book Model	73
ActiveRecord Mapping	73
Modifying the Generated Models	77
Cloning the Database	80
Unit Testing Validations	81
Unit Testing the ActiveRecord Mappings	82
Generating Book Administration Code with the Scaffolding Script	88
Integration Testing	90
Completing the Add Book User Story	91
Completing the Upload Book Cover User Story	102
Completing the List Books User Story	104
Completing the View Book User Story	107
Completing the Edit Book User Story	110
Testing the Delete Book User Story	112
Summary	112
CHAPTER 4 Book Catalog Browsing	113
Getting the Book Catalog Requirements	113
Implementing the Book Catalog Interface	114
Implementing the Browse Books User Story	116
Implementing the View Book Details User Story	120
Implementing the Search Books User Story	125
Implementing the Get Latest Books User Story	133
Creating an RSS Feed	136
Summary	139

CHAPTER 5	Shopping Cart Implementation	141
	Getting the Shopping Cart Requirements	141
	Setting Up the Shopping Cart	142
	Creating the Controller	142
	Adding a Functional Test	142
	Creating the Models	143
	Modifying the Controller	145
	Creating the Views	147
	Implementing the User Stories	152
	Implementing the Add Items to the Cart User Story	152
	Implementing the Remove Items from the Cart User Story	161
	Implementing the Clear the Cart User Story	166
	Summary	168
CHAPTER 6	Forum Implementation	169
	Getting the Forum Requirements	169
	Using the Threaded Forum Plugin	170
	Setting Up the Forum	171
	Updating the Database Schema	171
	Modifying the Model	175
	Unit Testing the Model	176
	Generating the Controller and View	177
	Implementing the User Stories	179
	Implementing the Post to Forum User Story	179
	Implementing the View Forum User Story	185
	Implementing the View Post User Story	190
	Implementing the Reply to Post User Story	192
	Summary	195
CHAPTER 7	Tagging Support	197
	Getting the Tagging Requirements	197
	Using the Tagging RubyGem	198
	Setting Up for Tagging	201
	Updating the Database Schema	201
	Preparing the Models	203
	Unit Testing the Model	204
	Using the Console to Test the Model	205

Implementing the User Stories	207
Implementing the Assign Tags User Story	207
Implementing the Edit Tags User Story	211
Implementing the List Tags and Show Tag User Stories	215
Implementing the Recommend Books User Story	218
Summary	221
CHAPTER 8 Security	223
Getting the Authentication Requirements	223
Using the Authentication Plugin	224
Implementing the User Stories	227
Implementing the Log In User Story	227
Implementing the Fail Log In User Story	233
Implementing the Reset Password User Story	238
Protecting Your Application	248
Cross-Site Scripting	248
URL and Form Manipulation	248
SQL Injection	249
Cross-Site Request Forgery	250
Summary	250
CHAPTER 9 Checkout and Order Processing	251
Getting the Checkout and Order-Processing Requirements	252
Implementing the Check Out User Story	252
Creating the Models	252
Adding Validations to the Model	257
Creating the Controller and Integration Test	259
Creating the View	262
Saving the Order Information	268
Integrating with Payment Gateways	271
Installing the Active Merchant Plugin	271
Integrating with PayPal	272
Integrating with Authorize.Net	280
Using the Payment Gem	284
Implementing the Administrator User Stories	286
Implementing the View Orders User Story	286
Implementing the View Order User Story	290
Implementing the Close Order User Story	292

Calculating Shipping Costs and Taxes	294
Using the Shipping Gem	294
Calculating Taxes	296
Summary	296
CHAPTER 10 Multiple Language Support	297
Getting the Localization Requirements	297
Using the Globalize Plugin	298
Localizing with Globalize	300
Setting Up Globalize	303
Implementing the User Stories	304
Implementing the Change Locale User Story	304
Implementing the Translation User Stories	306
Translating the View and the Book Model	313
Translating the View	313
Translating the Model	317
Localizing Dates, Numbers, and Currency	319
Localizing Dates	319
Localizing Numbers and Currencies	320
Adding Unicode (UTF-8) Support	322
Setting Character Encoding in HTML	323
Setting Character Encoding for the HTTP Response	324
Changing the Database to Use UTF-8	324
Summary	326
CHAPTER 11 Acceptance Testing	327
Using Selenium	327
Writing Selenium Tests	330
Selenium Commands	330
Selenium Test Formats	334
The First Acceptance Test	335
Recording Selenium Tests	337
Using the Selenium IDE	337
Recording the View Forum Acceptance Test	340
Recording the Post to Forum Acceptance Test	345
Recording the Show Post Acceptance Test	347
Recording the Reply to Post Acceptance Test	348
Summary	350

CHAPTER 12	Application Deployment	351
	Setting Up the Production Environment	351
	Connecting to the Production Server: SSH	352
	Installing the Web Server: LightTPD	353
	Installing the Application Server: Ruby on Rails and FastCGI	356
	Installing the Database Server (MySQL)	358
	Configuring LightTPD	358
	Creating the Production Database	365
	Deploying the Application Manually	366
	Copying the Application	367
	Creating Users and Groups	367
	Starting LightTPD	368
	Starting FastCGI Processes	369
	Automating Deployment	371
	Installing Capistrano	371
	Creating the Capistrano Deployment Recipe	371
	Running the Setup Task	375
	Deploying to Production	376
	Starting LightTPD	379
	Summary	380
CHAPTER 13	Performance Optimization	381
	Performance and Scaling	381
	Measuring Performance	382
	Checking the Log File	382
	Using Rails Analyzer	383
	Caching	388
	Page Caching	388
	Action Caching	390
	Fragment Caching	390
	Fragment Stores	392
	Caching ActiveRecord Objects	395
	Common Performance Problems in Rails	397
	Rendering Speed	397
	Database Access	399
	Summary	401
INDEX		403

About the Authors



■ **CHRISTIAN HELLSTEN** is the founder of Aktagon Ltd., a provider of consulting services and custom Internet software development, and CTO of Sanda Interactive Ltd. He has worked on large-scale e-business projects as a consultant for PricewaterhouseCoopers Consulting and IBM Business Consulting Services. Christian's background is in J2EE, but he fell in love with Ruby on Rails at first sight, and has been using it professionally ever since to build web applications. When he is not changing the diapers of his two young daughters at his home in Finland, Christian enjoys researching new and better ways of building software.



■ **JARKKO LAINE** is the owner and CEO of O'Design, a Rails-based web design shop. He has been using Ruby on Rails since its public launch in 2004. He has contributed patches to the core developer team, and has also contributed to several Rails plugins. Jarkko has provided Rails consultancy for a number of organizations, from nonprofits to Fortune 500 companies. He has also taught Rails at the university level and delivers lectures about Rails around the world. Currently, he works on dotherightthing.com, a project that will bring people a whole new way to rate, follow, and discuss the social responsibility of companies. Jarkko is a sports junkie, so if he isn't sitting in front of his computer, he is probably running around forests or kicking a ball on the nearest field. He lives in Tampere, Finland, with his fiancée Maria and a growing list of pending household chores.

About the Technical Reviewer



■ **PETER MARKLUND** has extensive experience with and expertise in object orientation, web development, relational databases, and testing, and has been doing web development with Java and Tcl since 2000. He was one of the core developers of the OpenACS open source web framework. In late 2004, he was introduced to Ruby on Rails and has since helped develop an online community and a CRM system with Rails. Peter is currently working as a Ruby on Rails freelancer and is also helping organize events for the Ruby on Rails developer community in Stockholm. Peter has a personal blog at <http://marklunds.com>, where he shares Rails tips with other developers.

Acknowledgments

First of all, I would like to thank my family for allowing me to take on such a time-consuming project as this in my spare time. Secondly, I would like to thank everyone involved in this project, including Keir Thomas, Jarkko Laine, Peter Marklund, Beth Christmas, Marilyn Smith, and Kelly Winquist. Last, but not least, I would like to thank my parents, for buying me a Commodore VIC-20, back in the early 1980s.

Christian Hellsten

I am eternally grateful to the following people: Yukihiro “Matz” Matsumoto and David Heinemeier Hansson for bringing passion and joy back to programming; my ex-girlfriend—now fiancée—Maria, for putting up with the innumerable nights spent married to the computer; my parents, for telling me to believe in and pursue my dreams, even if it was just “fooling around with computers”; the whole team at Apress, for towing me back on track in the moments of despair; and finally, Philip and Alex, for igniting the spark.

Jarkko Laine

Introduction

B*eginning Ruby on Rails E-Commerce* is for people who want to learn how to build real-world professional web applications using Rails best practices. We put a specific emphasis on e-commerce by showing you how to build an online bookstore, including a shopping cart, catalog, forum, and other functionality. On the front-end, we guide you through important technologies like Ajax, syndication, tagging, and internationalization. On the back-end, we show you how to integrate with payment gateways, use ActiveRecord and the Ferret search engine, and many other techniques.

This book is also targeted at people who already have written an application or two using Rails, but who want to learn more about how test-driven development (TDD) can improve the quality of their code, and how to go beyond the standard test features built inside Rails.

We will guide you through all the phases of a professional e-commerce project, from concept to production deployment and maintenance. In the first chapters, we show you how to jump-start your project and build a good, solid foundation for it, using agile practices like TDD. In later chapters, we dig deeper into Ruby on Rails, covering common requirements, such as translating your application into multiple languages and debugging production problems.

Beginning Ruby on Rails E-Commerce is not intended to be a reference manual for Ruby on Rails. You can find many online resources and other books that provide a complete reference to the Ruby on Rails API and features, and these are mentioned throughout this book.

What Is Ruby on Rails?

Ruby on Rails (<http://rubyonrails.org>) is a web application framework written using the Ruby programming language. It was originally created by David Heinemeier Hansson, a Danish hacker, during the development of an online project collaboration tool called Basecamp.

As with most great things, Ruby on Rails started as an itch. Hansson was not happy with the available web application frameworks at the time, so he decided to write his own. In the design of Ruby on Rails, David emphasized a couple of things like convention over configuration, less software, and that programmer happiness ultimately leads to better productivity.

Ruby on Rails was first released to the public in July 2004. Since then, it has seen an explosive growth in popularity. It is loved because of its simplicity and power, which allow

you to solve problems faster and with less code than, as David said, “most frameworks spend doing XML sit-ups.”

What Is Ruby?

Ruby (<http://ruby-lang.org>) is a dynamically typed programming language created by a Japanese Software Engineer called Yukihiro “Matz” Matsumoto in February 1993. Ruby is licensed under the GPL-like Ruby license and was released to the public in 1995, which is about one year later than Java. It is actively maintained by Matz and contributors from all over the world.

Unlike most other programming languages, Matz designed Ruby to increase programmer happiness, and to let programmers concentrate more on solving the task at hand than on language syntax. This is arguably the greatest strength of the Ruby programming language, when compared to other programming languages.

Ruby is a completely object-oriented language, unlike for instance Java, which has primitives. Everything in Ruby is an object, even `nil`. Ruby is also highly dynamic, allowing you to change classes and to introduce new methods at runtime. This allows the programmer to do things that aren't possible in languages like Java and C++.