

# Practical Django Projects



James Bennett

## **Practical Django Projects**

**Copyright © 2008 by James Bennett**

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): 978-1-59059-996-9

ISBN-10 (pbk): 1-59059-996-9

ISBN-13 (electronic): 978-1-4302-0868-6

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.

Java™ and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc., in the US and other countries. Apress, Inc., is not affiliated with Sun Microsystems, Inc., and this book was written without endorsement from Sun Microsystems, Inc.

Lead Editors: Steve Anglin, Tom Welsh

Technical Reviewer: Russell Keith-Magee

Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell,

Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Richard Dal Porto

Copy Editors: Kim Benbow, Nicole Abramowitz

Associate Production Director: Kari Brooks-Copony

Production Editor: Kelly Gunther

Compositor: Dina Quan

Proofreader: Nancy Sixsmith

Indexer: Carol Burbo

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](mailto:orders-ny@springer-sbm.com), or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto:info@apress.com), or visit <http://www.apress.com>.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at <http://www.apress.com/info/bulksales>.

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>. You may need to answer questions pertaining to this book in order to successfully download the code.

*This book would not have been possible without the huge and supportive community that has grown up around Django in the past three years. The willingness of people all around the world to freely contribute their code, their ideas, and their time to improving the state of our art never ceases to amaze me.*

*This book also would not have been possible without Mr. Morgan, who instilled in me both the craft and the joy of writing. For that he has my deepest thanks.*



# Contents at a Glance

About the Author .....	xiii
About the Technical Reviewer.....	xv
Introduction .....	xvii
<b>CHAPTER 1</b> Welcome to Django .....	1
<b>CHAPTER 2</b> Your First Django Site: A Simple CMS .....	9
<b>CHAPTER 3</b> Customizing the Simple CMS.....	23
<b>CHAPTER 4</b> A Django-Powered Weblog.....	43
<b>CHAPTER 5</b> Expanding the Weblog .....	77
<b>CHAPTER 6</b> Templates for the Weblog.....	97
<b>CHAPTER 7</b> Finishing the Weblog .....	123
<b>CHAPTER 8</b> A Social Code-Sharing Site .....	147
<b>CHAPTER 9</b> Form Processing in the Code-Sharing Application .....	165
<b>CHAPTER 10</b> Finishing the Code-Sharing Application .....	187
<b>CHAPTER 11</b> Writing Reusable Django Applications .....	205
<b>INDEX</b> .....	225



# Contents

About the Author .....	xiii
About the Technical Reviewer.....	xv
Introduction .....	xvii
■ <b>CHAPTER 1</b>	<b>Welcome to Django</b> .....
	1
What's a Web Framework and Why Should I Want One?.....	1
Say Hello to Django.....	2
Say Hello to Python.....	3
Installing Django .....	4
Your First Steps with Django.....	5
Exploring Your Django Project.....	7
Looking Ahead.....	8
■ <b>CHAPTER 2</b>	<b>Your First Django Site: A Simple CMS</b> .....
	9
Configuring Your First Django Project .....	9
Putting Together the CMS .....	12
A Quick Introduction to the Django Template System.....	18
Looking Ahead.....	21
■ <b>CHAPTER 3</b>	<b>Customizing the Simple CMS</b> .....
	23
Adding Rich-Text Editing .....	23
Adding a Search System to the CMS.....	26
Improving the Search View .....	31
Improving the Search Function with Keywords .....	33
Looking Ahead.....	40
■ <b>CHAPTER 4</b>	<b>A Django-Powered Weblog</b> .....
	43
Feature Checklist.....	43
Writing a Django Application .....	44
Projects vs. Applications.....	44
Standalone and Coupled Applications .....	45

Creating the Weblog Application . . . . .	45
Designing the Models . . . . .	47
The Entry Model . . . . .	52
Basic Fields . . . . .	53
Slugs, Useful Defaults, and Uniqueness Constraints . . . . .	54
Authors, Comments, and Featured Entries . . . . .	55
Different Types of Entries . . . . .	56
Categorizing and Tagging Entries . . . . .	58
Writing Entries Without Writing HTML . . . . .	59
Finishing Touches . . . . .	61
The Weblog Models So Far . . . . .	62
Writing the First Views . . . . .	65
Using Django's Generic Views . . . . .	69
How Did Django Do That? . . . . .	70
Decoupling the URLs . . . . .	72
Looking Ahead . . . . .	75

## ■ CHAPTER 5    **Expanding the Weblog** . . . . . 77

Writing the Link Model . . . . .	77
Views for the Link Model . . . . .	83
Setting Up Views for Categories . . . . .	84
Using Generic Views (Again) . . . . .	86
Views for Tags . . . . .	87
Cleaning Up the URLConf . . . . .	89
Handling Live Entries . . . . .	93
Looking Ahead . . . . .	95

## ■ CHAPTER 6    **Templates for the Weblog** . . . . . 97

Dealing with Repetitive Elements: The Power of Inheritance . . . . .	97
How Template Inheritance Works . . . . .	99
Limits of Template Inheritance . . . . .	100
Defining the Base Template for the Blog . . . . .	100
Section Templates . . . . .	103
Archives of Entries . . . . .	104
Entry Index . . . . .	104
Yearly Archive . . . . .	105
Monthly and Daily Archives . . . . .	106
Entry Detail . . . . .	107



Templates for Other Types of Content .....	110
Extending the Template System with Custom Tags .....	111
How a Django Template Works .....	112
A Simple Custom Tag .....	113
Writing a More Flexible Tag with Arguments .....	116
Writing the Compilation Function .....	116
Writing the LatestContentNode .....	119
Registering and Using the New Tag .....	120
Looking Ahead .....	122
<b>■ CHAPTER 7   Finishing the Weblog .....</b>	<b>123</b>
Comments and django.contrib.comments .....	123
Installing the Comments Application .....	123
Basic Setup .....	124
Retrieving Lists of Comments for Display .....	128
Comment Moderation .....	129
Signals and the Django Dispatcher .....	130
Building the Automatic Comment Moderator .....	131
Adding Akismet Support .....	132
E-mail Notification of Comments .....	135
Dealing with Moderated Comments in Public-Facing Templates .....	137
Adding Feeds .....	138
LatestEntriesFeed .....	139
Entries by Category: A More Complex Feed Example .....	142
Looking Ahead .....	146
<b>■ CHAPTER 8   A Social Code-Sharing Site .....</b>	<b>147</b>
Feature Checklist .....	147
Setting Up the Application .....	148
Building the Initial Models .....	148
The Language Model .....	149
The Snippet Model .....	151
Testing the Snippets Application .....	154

Initial Views for Snippets and Languages . . . . .	155
CSS for pygments Syntax Highlighting . . . . .	156
Views for Languages . . . . .	157
An Advanced View: Top Authors . . . . .	158
Improving the View of Top Authors . . . . .	159
Adding a top_languages View . . . . .	162
Looking Ahead . . . . .	163
 <b>■ CHAPTER 9 Form Processing in the Code-Sharing Application . . . . .</b>	<b>165</b>
A Brief Tour of Django's Form System . . . . .	165
A Simple Example . . . . .	166
Validating the Username . . . . .	168
Validating the Password . . . . .	169
Creating the New User . . . . .	169
How Form Validation Works . . . . .	171
Processing the Form . . . . .	173
A Form for Adding Code Snippets . . . . .	175
Writing a View to Process the Form . . . . .	178
Automatically Generating a Form for Adding Snippets . . . . .	180
Simplifying Templates That Display Forms . . . . .	183
Editing Snippets . . . . .	184
Looking Ahead . . . . .	186
 <b>■ CHAPTER 10 Finishing the Code-Sharing Application . . . . .</b>	<b>187</b>
Bookmarking Snippets . . . . .	187
Basic Bookmark Views . . . . .	188
A New Template Tag: {% if_bookmarked %} . . . . .	192
Parsing Ahead in a Django Template . . . . .	193
Resolving Variables Inside a Template Node . . . . .	194
Using RequestContext to Automatically Populate Template Variables . . . . .	196
Adding the User Rating System . . . . .	198
Rating Snippets . . . . .	201
Adding an {% if_rated %} Template Tag . . . . .	202
Retrieving a User's Rating . . . . .	203
Looking Ahead . . . . .	204

<b>CHAPTER 11</b>	<b>Writing Reusable Django Applications</b>	205
	One Thing at a Time	206
	Staying Focused	206
	Advantages of Tightly Focused Applications	207
	Developing Multiple Applications	208
	Drawing the Lines Between Applications	209
	Splitting Up the Snippets Application	210
	Building for Flexibility	210
	Flexible Form Handling	211
	Flexible Template Handling	212
	Flexible Post-Form Processing	213
	Flexible URL Handling	214
	Taking Advantage of Django's APIs	215
	Staying Generic	215
	Distributing Django Applications	217
	Python Packaging Tools	217
	Writing a setup.py Script with distutils	218
	Standard Files to Include in a Package	219
	Documenting an Application	220
	Looking Ahead	224
<b>INDEX</b>		225



# About the Author



**JAMES BENNETT** is a web developer at the *Lawrence Journal-World* in Lawrence, Kansas, where Django was originally developed. He is both a regular contributor to and the release manager for the open source Django project.



# About the Technical Reviewer



■ **DR. RUSSELL KEITH-MAGEE** has been a core developer on the Django project since January 2006. He is a cofounder of Django Evolution, a schema evolution framework for Django. He is an active participant on the Django Users and Django Developers mailing lists and is a mentor in the Google Summer of Code 2008.

In addition to his work with Django, Russell has worked at two startup companies—one very successful and one still in development. In those jobs, he has used his passion for good design, powerful tools, and automated testing to find elegant solutions to real-world problems faced by real-world users.

Russell lives with his wife, son, and two cats in Perth, Western Australia.





# Introduction

**T**he past few years have seen an explosion in the development of dynamic, database-driven web sites. Where many sites were once built using nothing but handwritten HTML, or a few CGI scripts or server-side includes, today database-backed web applications have become the norm for everything from personal blogs to online stores to the social networking sites that have revolutionized the way many people use the Web.

But this has come at a cost. Developing these applications, even for relatively simple uses, involves a significant amount of complex work, and much of that work ends up being repeated for each new application. Although web developers have always had access to libraries of code that could automate certain tasks, such as HTML templating or database querying, the process of bringing together all the necessary pieces for a fully polished application has largely remained difficult and tedious.

This has led to the recent development, and subsequent popularity, of “web frameworks,” reusable collections of components that handle many of the common and repetitive tasks of application development in an integrated fashion. Instead of requiring you to obtain disparate libraries of code and find ways to make them work together, web frameworks provide all the necessary components in a single package and take care of the integration work for you.

Django is one of the most recent crop of web frameworks, growing out of the needs of a fast-paced online news operation. Django's original developers needed a set of tools that would not only help them quickly develop new and highly dynamic web applications in response to the rapidly evolving requirements of the news industry, but would also let them save time and effort by reusing pieces of code, and even entire applications, whenever possible.

In this book, you'll see how Django can help you achieve both of these goals—rapid application development and flexible, reusable code—through the tools it provides to you directly and the development practices that it makes possible. I'll guide you through the development of several example applications and show you how the various components and applications bundled with Django can help you to write less code at each stage of the development process. You'll also see firsthand a number of best practices for reusable code and learn how you can apply them in your own applications, as well as see how to integrate existing third-party libraries into Django-powered applications to minimize the amount of code you'll need to write from scratch.

I've written this book from a pragmatic viewpoint. The sample applications are all intended to be useful in real-world situations, and once you've worked through them, you'll have more than just a technical understanding of Django and its components. You'll have a clear understanding of how Django can help you become a more productive and more effective developer.

