

# The Definitive Guide to symfony



François Zaninotto and  
Fabien Potencier

## **The Definitive Guide to symfony**

**Copyright © 2007 by Sensio SA**

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and one Back-Cover Text: "Apress (<http://www.apress.com/>) and the authors ask for your support by buying the print edition through any online or retail outlet." A copy of the license is included in the section entitled "GNU Free Documentation License."

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): 978-1-59059-786-6

ISBN-10 (pbk): 1-59059-786-9

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: Jason Gilmore

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

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

Project Manager: Kylie Johnston

Copy Edit Manager: Nicole Flores

Copy Editors: Marilyn Smith and Ami Knox

Assistant Production Director: Kari Brooks-Copony

Production Editor: Katie Stence

Compositor: Susan Glinert

Proofreaders: Linda Marousek and April Eddy

Indexer: Toma Mulligan

Artist: April Milne

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 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail [info@apress.com](mailto: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.

*To Anne-Marie.*  
—François Zaninotto

*For Thomas and Hélène, with love.*  
—Fabien Potencier



# Contents at a Glance

About the Authors .....	xxi
About Sensio Labs .....	xxiii
Acknowledgments .....	xxv
Introduction .....	xxvii
License .....	xxix

## PART 1 ■■■ The Basics

■ CHAPTER 1	Introducing Symfony .....	3
■ CHAPTER 2	Exploring Symfony's Code .....	13
■ CHAPTER 3	Running Symfony .....	35
■ CHAPTER 4	The Basics of Page Creation .....	49
■ CHAPTER 5	Configuring Symfony .....	61

## PART 2 ■■■ The Core Architecture

■ CHAPTER 6	Inside the Controller Layer .....	83
■ CHAPTER 7	Inside the View Layer .....	113
■ CHAPTER 8	Inside the Model Layer .....	141

## PART 3 ■■■ Special Features

■ CHAPTER 9	Links and the Routing System .....	171
■ CHAPTER 10	Forms .....	191
■ CHAPTER 11	Ajax Integration .....	221
■ CHAPTER 12	Caching .....	245
■ CHAPTER 13	I18N and L10N .....	265

## PART 4 ■■■ Development Tools

■ CHAPTER 14	Generators .....	281
■ CHAPTER 15	Unit and Functional Testing .....	317
■ CHAPTER 16	Application Management Tools .....	345
■ CHAPTER 17	Extending Symfony .....	367

## PART 5 ■■■ Becoming a Symfony Expert

■ CHAPTER 18	Performance .....	397
■ CHAPTER 19	Mastering Symfony's Configuration Files .....	417
■ APPENDIX	GNU Free Documentation License .....	437
■ INDEX	.....	445

# Contents

About the Authors .....	xxi
About Sensio Labs .....	xxiii
Acknowledgments .....	xxv
Introduction .....	xxvii
License .....	xxix

## PART 1 ■■■ The Basics

■ CHAPTER 1	<b>Introducing Symfony</b> .....	3
	Symfony in Brief .....	3
	Symfony Features .....	3
	Who Made Symfony and Why? .....	5
	The Symfony Community .....	5
	Is Symfony for Me? .....	6
	Fundamental Concepts .....	6
	PHP 5 .....	7
	Object-Oriented Programming (OOP) .....	7
	Magic Methods .....	7
	PHP Extension and Application Repository (PEAR) .....	7
	Object-Relational Mapping (ORM) .....	8
	Rapid Application Development (RAD) .....	9
	YAML .....	10
	Summary .....	11
■ CHAPTER 2	<b>Exploring Symfony's Code</b> .....	13
	The MVC Pattern .....	13
	MVC Layering .....	14
	Layer Separation Beyond MVC .....	18
	Symfony's MVC Implementation .....	22
	Symfony Core Classes .....	25

Code Organization . . . . .	25
Project Structure: Applications, Modules, and Actions . . . . .	25
File Tree Structure . . . . .	26
Common Instruments . . . . .	31
Parameter Holders . . . . .	31
Constants . . . . .	33
Class Autoloading . . . . .	33
Summary . . . . .	34
<b>CHAPTER 3 Running Symfony . . . . .</b>	<b>35</b>
Installing the Sandbox . . . . .	35
Installing the Symfony Libraries . . . . .	37
Installing the Symfony PEAR Package . . . . .	37
Checking Out Symfony from the SVN Repository . . . . .	38
Setting Up an Application . . . . .	39
Creating the Project . . . . .	39
Creating the Application . . . . .	40
Configuring the Web Server . . . . .	41
Setting Up a Virtual Host . . . . .	41
Configuring a Shared-Host Server . . . . .	42
Troubleshooting . . . . .	44
Typical Problems . . . . .	44
Symfony Resources . . . . .	45
Source Versioning . . . . .	45
Summary . . . . .	47
<b>CHAPTER 4 The Basics of Page Creation . . . . .</b>	<b>49</b>
Creating a Module Skeleton . . . . .	49
Adding a Page . . . . .	51
Adding an Action . . . . .	51
Adding a Template . . . . .	53
Passing Information from the Action to the Template . . . . .	54
Gathering Information from the User with Forms . . . . .	54
Linking to Another Action . . . . .	56
Getting Information from the Request . . . . .	58
Summary . . . . .	59



<b>CHAPTER 5</b>	<b>Configuring Symfony</b> .....	61
	The Configuration System .....	61
	YAML Syntax and Symfony Conventions .....	62
	Help, a YAML File Killed My App! .....	65
	Overview of the Configuration Files .....	66
	Project Configuration .....	66
	Application Configuration .....	67
	Module Configuration .....	70
	Environments .....	71
	What Is an Environment? .....	71
	Configuration Cascade .....	73
	The Configuration Cache .....	75
	Accessing the Configuration from Code .....	75
	The sfConfig Class .....	76
	Custom Application Settings and app.yml .....	77
	Tips for Getting More from Configuration Files .....	78
	Using Constants in YAML Configuration Files .....	78
	Using Scriptable Configuration .....	79
	Browsing Your Own YAML File .....	79
	Summary .....	80

## PART 2 ■■■ The Core Architecture

<b>CHAPTER 6</b>	<b>Inside the Controller Layer</b> .....	83
	The Front Controller .....	83
	The Front Controller's Job in Detail .....	84
	The Default Front Controller .....	84
	Calling Another Front Controller to Switch the Environment .....	85
	Batch Files .....	85
	Actions .....	86
	The Action Class .....	86
	Alternative Action Class Syntax .....	88
	Retrieving Information in the Action .....	88
	Action Termination .....	89
	Skipping to Another Action .....	91
	Repeating Code for Several Actions of a Module .....	93
	Accessing the Request .....	94

User Session .....	96
Accessing the User Session.....	96
Flash Attributes.....	98
Session Management.....	98
Action Security .....	100
Access Restriction .....	100
Granting Access .....	101
Complex Credentials.....	103
Validation and Error-Handling Methods .....	104
Filters .....	105
The Filter Chain.....	106
Building Your Own Filter .....	108
Filter Activation and Parameters.....	109
Sample Filters .....	110
Module Configuration .....	111
Summary .....	112

<b>CHAPTER 7</b>	<b>Inside the View Layer .....</b>	<b>113</b>
	Templating .....	113
	Helpers.....	114
	Page Layout.....	117
	Template Shortcuts .....	119
	Code Fragments .....	119
	Partials.....	120
	Components .....	122
	Slots .....	124
	View Configuration .....	126
	The view.yml File .....	127
	The Response Object .....	129
	View Configuration Settings.....	130
	Component Slots .....	135
	Output Escaping .....	137
	Activating Output Escaping .....	137
	Escaping Strategy.....	138
	Escaping Helpers .....	139
	Escaping Arrays and Objects.....	139
	Summary .....	140

<b>CHAPTER 8</b>	<b>Inside the Model Layer</b>	<b>141</b>
	Why Use an ORM and an Abstraction Layer?	141
	Symfony's Database Schema	143
	Schema Example	143
	Basic Schema Syntax	144
	Model Classes	145
	Base and Custom Classes	146
	Object and Peer Classes	146
	Accessing Data	147
	Retrieving the Column Value	148
	Retrieving Related Records	148
	Saving and Deleting Data	150
	Retrieving Records by Primary Key	150
	Retrieving Records with Criteria	151
	Using Raw SQL Queries	154
	Using Special Date Columns	155
	Database Connections	156
	Extending the Model	157
	Adding New Methods	157
	Overriding Existing Methods	158
	Using Model Behaviors	159
	Extended Schema Syntax	159
	Attributes	160
	Column Details	161
	Foreign Keys	162
	Indexes	163
	Empty Columns	164
	1:n Tables	164
	Beyond the schema.yml: The schema.xml	165
	Don't Create the Model Twice	166
	Building a SQL Database Structure Based on an Existing Schema	166
	Generating a YAML Data Model from an Existing Database	167
	Summary	168

## PART 3 ■■■ Special Features

■ CHAPTER 9	<b>Links and the Routing System</b> .....	171
	What Is Routing? .....	171
	URLs As Server Instructions .....	171
	URLs As Part of the Interface .....	172
	How It Works .....	174
	URL Rewriting .....	176
	Link Helpers .....	177
	Hyperlinks, Buttons, and Forms .....	177
	Link Helper Options .....	178
	Fake GET and POST Options .....	179
	Forcing Request Parameters As GET Variables .....	180
	Using Absolute Paths .....	181
	Routing Configuration .....	182
	Rules and Patterns .....	183
	Pattern Constraints .....	184
	Setting Default Values .....	186
	Speeding Up Routing by Using the Rule Name .....	186
	Adding an .html Extension .....	187
	Creating Rules Without routing.yml .....	188
	Dealing with Routes in Actions .....	189
	Summary .....	190
■ CHAPTER 10	<b>Forms</b> .....	191
	Form Helpers .....	191
	Main Form Tag .....	191
	Standard Form Elements .....	192
	Date Input Widgets .....	195
	Rich Text Editing .....	197
	Country and Language Selection .....	198
	Form Helpers for Objects .....	199
	Populating Drop-Down Lists with Objects .....	200
	Creating a Drop-Down List Based on a Foreign Key Column .....	201
	Updating Objects .....	202

Form Validation .....	203
Validators .....	203
Validation File .....	205
Redisplaying the Form .....	206
Displaying the Error Messages in the Form .....	207
Repopulating the Form .....	209
Standard Symfony Validators .....	210
Named Validators .....	213
Restricting the Validation to a Method .....	214
What Does a Validation File Look Like? .....	214
Complex Validation .....	215
Creating a Custom Validator .....	215
Using Array Syntax for Form Fields .....	217
Executing a Validator on an Empty Field .....	217
Summary .....	219
<b>CHAPTER 11 Ajax Integration .....</b>	<b>221</b>
Basic JavaScript Helpers .....	221
JavaScript in Templates .....	222
Updating a DOM Element .....	223
Graceful Degradation .....	224
Prototype .....	224
Ajax Helpers .....	226
Ajax Link .....	228
Ajax-Driven Forms .....	228
Periodically Calling Remote Functions .....	231
Remote Call Parameters .....	231
Updating Distinct Elements According to the Response Status .....	231
Updating an Element According to Position .....	232
Updating an Element According to a Condition .....	232
Determining the Ajax Request Method .....	233
Authorizing Script Execution .....	233
Creating Callbacks .....	234
Creating Visual Effects .....	235
JSON .....	236

Performing Complex Interactions with Ajax .....	238
Autocompletion .....	239
Drag-and-Drop .....	240
Sortable Lists .....	241
Edit in Place .....	242
Summary .....	243
<b>CHAPTER 12 Caching .....</b>	<b>245</b>
Caching the Response .....	245
Global Cache Settings .....	246
Caching an Action .....	246
Caching a Partial, Component, or Component Slot .....	248
Caching a Template Fragment .....	249
Configuring the Cache Dynamically .....	251
Using the Super Fast Cache .....	253
Removing Items from the Cache .....	254
Clearing the Entire Cache .....	254
Clearing Selective Parts of the Cache .....	255
Cache Directory Structure .....	257
Clearing the Cache Manually .....	258
Testing and Monitoring Caching .....	258
Building a Staging Environment .....	258
Monitoring Performance .....	259
Benchmarking .....	260
Identifying Cache Parts .....	260
HTTP 1.1 and Client-Side Caching .....	260
Adding an ETag Header to Avoid Sending Unchanged Content .....	261
Adding a Last-Modified Header to Avoid Sending Still Valid Content .....	261
Adding Vary Headers to Allow Several Cached Versions of a Page .....	262
Adding a Cache-Control Header to Allow Client-Side Caching .....	262
Summary .....	263

<b>CHAPTER 13 I18N and L10N</b> .....	265
User Culture .....	265
Setting the Default Culture .....	266
Changing the Culture for a User .....	266
Determining the Culture Automatically .....	267
Standards and Formats .....	268
Outputting Data in the User's Culture .....	268
Getting Data from a Localized Input .....	269
Text Information in the Database .....	270
Creating Localized Schema .....	270
Using the Generated I18n Objects .....	271
Interface Translation .....	272
Configuring Translation .....	272
Using the Translation Helper .....	272
Using Dictionary Files .....	273
Managing Dictionaries .....	274
Handling Other Elements Requiring Translation .....	275
Handling Complex Translation Needs .....	275
Calling the Translation Helper Outside a Template .....	277
Summary .....	277

## PART 4 ■■■ Development Tools

<b>CHAPTER 14 Generators</b> .....	281
Code Generation Based on the Model .....	281
Scaffolding and Administration .....	282
Initiating or Generating Code .....	282
Example Data Model .....	282
Scaffolding .....	284
Generating a Scaffolding .....	284
Initiating a Scaffolding .....	286
Administration .....	286
Initiating an Administration Module .....	287
A Look at the Generated Code .....	288
Introducing the generator.yml Configuration File .....	289

Generator Configuration .....	290
Fields .....	291
View Customization .....	297
List View–Specific Customization .....	299
Edit View–Specific Customization .....	304
Dealing with Foreign Keys .....	307
Adding Interactions .....	308
Form Validation .....	311
Restricting User Actions Using Credentials .....	312
Modifying the Presentation of Generated Modules .....	312
Using a Custom Style Sheet .....	313
Creating a Custom Header and Footer .....	313
Customizing the Theme .....	314
Summary .....	316

## CHAPTER 15 Unit and Functional Testing .....

Automated Tests .....	317
Unit and Functional Tests .....	317
Test-Driven Development .....	318
The Lime Testing Framework .....	319
Unit Tests .....	319
What Do Unit Tests Look Like? .....	320
Unit Testing Methods .....	321
Testing Parameters .....	323
The test-unit Task .....	324
Stubs, Fixtures, and Autoloading .....	325
Functional Tests .....	328
What Do Functional Tests Look Like? .....	328
Browsing with the sfTestBrowser Object .....	330
Using Assertions .....	332
Using CSS Selectors .....	334
Working in the Test Environment .....	336
The test-functional Task .....	337
Test Naming Practices .....	338
Special Testing Needs .....	339
Executing Tests in a Test Harness .....	339
Accessing a Database .....	340
Testing the Cache .....	341
Testing Interactions on the Client .....	342
Summary .....	344



<b>CHAPTER 16 Application Management Tools</b> .....	345
Logging .....	345
PHP Logs .....	345
Symfony Logs .....	346
Debugging .....	349
Symfony Debug Mode .....	349
Symfony Exceptions .....	350
Xdebug Extension .....	351
Web Debug Toolbar .....	352
Manual Debugging .....	356
Populating a Database .....	358
Fixture File Syntax .....	358
Launching the Import .....	359
Using Linked Tables .....	359
Deploying Applications .....	360
Freezing a Project for FTP Transfer .....	360
Using rsync for Incremental File Transfer .....	361
Ignoring Irrelevant Files .....	363
Managing a Production Application .....	363
Summary .....	365
<b>CHAPTER 17 Extending Symfony</b> .....	367
Mixins .....	367
Understanding Multiple Inheritance .....	367
Mixing Classes .....	368
Declaring a Class As Extendable .....	370
Registering Extensions .....	372
Extending with More Precision .....	374
Factories .....	375
Bridges to Other Framework Components .....	377
Plug-Ins .....	378
Finding Symfony Plug-Ins .....	379
Installing a Plug-In .....	380
Anatomy of a Plug-In .....	383
How to Write a Plug-In .....	386
Summary .....	393

## PART 5 ■■■ Becoming a Symfony Expert

■ CHAPTER 18 Performance .....	397
Tweaking the Server .....	397
Tweaking the Model .....	398
Optimizing Propel Integration .....	398
Limiting the Number of Objects to Hydrate .....	399
Minimizing the Number of Queries with Joins.....	399
Avoid Using Temporary Arrays .....	402
Bypassing the ORM .....	403
Speeding Up the Database .....	404
Tweaking the View .....	406
Using the Fastest Code Fragment.....	406
Speeding Up the Routing Process.....	406
Skipping the Template.....	407
Restricting the Default Helpers .....	407
Compressing the Response.....	408
Tweaking the Cache .....	408
Clearing Selective Parts of the Cache .....	408
Generating Cached Pages .....	409
Using a Database Storage System for Caching.....	410
Bypassing Symfony .....	411
Caching the Result of a Function Call.....	411
Caching Data in the Server .....	412
Deactivating the Unused Features .....	413
Optimizing Your Code .....	414
Core Compilation .....	414
The sfOptimizer Plug-In .....	415
Summary .....	416

<b>CHAPTER 19</b>	<b>Mastering Symfony's Configuration Files</b> .....	417
	Symfony Settings .....	417
	Default Modules and Actions.....	417
	Optional Feature Activation .....	419
	Feature Configuration.....	421
	Extending the Autoloading Feature .....	424
	Custom File Structure .....	426
	The Basic File Structure.....	426
	Customizing the File Structure .....	428
	Modifying the Project Web Root .....	428
	Linking to Symfony Libraries.....	429
	Understanding Configuration Handlers .....	430
	Default Configuration Handlers .....	430
	Adding Your Own Handler .....	431
	Controlling PHP Settings .....	434
	Summary .....	435
<b>APPENDIX</b>	<b>GNU Free Documentation License</b> .....	437
<b>INDEX</b>	.....	445



# About the Authors

■ **FRANÇOIS ZANINOTTO** is a consultant and project manager for Internet application projects. He graduated from the French business school Ecole des Mines in 1997 with a specialization in computer science. He tried quite a few jobs before settling on the Internet business: social worker in a children's facility, manager of a bike rental shop, web project manager for a tire manufacturer, writer of a travel guide on Germany for the same tire manufacturer, logistician for Médecins Sans Frontières, and IT architect for a consumer credit company. He joined the Sensio web agency in 2003, and since then has managed many Internet and intranet web application projects, dealing with complex usability issues, agile development methodologies, and cutting-edge web techniques. When the symfony project started, he took responsibility for the documentation, and wrote the symfony online book and tutorials.

■ **FABIEN POTENCIER** is a serial entrepreneur. Since he was ten, he always dreamed of creating and running companies. He started his career with an engineering degree from the French business school Ecole des Mines and an MBA in entrepreneurship from HEC Paris. In 1998, right after graduation, Fabien founded his very first company with a fellow student. The company was a web agency focused on simplicity and open source technologies, and was called Sensio. His acute technical knowledge and his endless curiosity won him the confidence of many French big corporate companies. While Sensio kept growing (at the time of writing, it has more than 30 employees), Fabien started other businesses: an indoor go-kart circuit in Lille (France), an auto spare parts e-commerce shop, and an autopilot training business riding on the most famous French racetracks. Fabien is the main developer of the symfony framework and is responsible for 95% of its code. Today, Fabien spends most of his time as Sensio's CEO and as the symfony project leader.



# About Sensio Labs

**S**ensio is a French web agency well known for its innovative ideas on web development. Founded in 1998 by Fabien Potencier, Gregory Pascal, and Samuel Potencier, Sensio benefited from the Internet growth of the late 1990s and situated itself as a major player for building complex web applications. It survived the Internet bubble burst by applying professional and industrial methods to a business where most players seemed to reinvent the wheel for each project. Most of Sensio's clients are large French corporations, who hire its teams to deal with small- to middle-scale projects with strong time-to-market and innovation constraints.

Today, Sensio's activity is divided in two business lines:

- **Extreme Sensio** deals with the interactive marketing projects and provides consulting on Internet communication strategies. It builds online communication campaigns from early conception to final product (websites, mailings, videos, viral marketing, and so on).
- **Sensio Labs** develops interactive web applications, both for dot-com and traditional companies. This division also provides auditing, consulting, and training on Internet technologies and complex application deployment. It helps define the global Internet strategy of large-scale industrial players. Sensio Labs has projects in France and abroad.

For its own needs, Sensio Labs develops the symfony framework and sponsors its deployment as an open source project. This means that symfony is built from experience and is really employed in many web applications, including those of large corporations.

Since its beginnings nine years ago, Sensio has always based its strategy on strong technical expertise. The company focuses on open source technologies, and as for dynamic scripting languages, Sensio offers developments in all LAMP platforms (Perl, Python, PHP, and Ruby, even if the latter doesn't start with a *P*). Sensio acquired strong experience on the best frameworks using these languages, and often develops web applications in Django, Rails, and, of course, symfony.

Sensio is always open to new business opportunities, so if you ever need help developing a web application, learning symfony, or evaluating a symfony development, feel free to contact us at [info@sensio.com](mailto:info@sensio.com). The consultants, project managers, web designers, and developers of Sensio can handle projects from A to Z.





# Acknowledgments

**T**he authors would like to thank the Apress team, including Jason, Kylie, Marilyn, Katie, Ami, and all the people who collaborated on the writing of this book.

Acknowledgments also go to the Sensio team, particularly to those who were willing to take their personal time to develop and write this book.

The symfony community, who asked thousands of questions about the framework, is also to be thanked, for they made us understand that this book should contain many practical tips.

And lastly, the authors would like to thank the reader of this book, who contributes to the development of the symfony project by this purchase, and would like to welcome every reader into the community.



# Introduction

**W**hen symfony first appeared in October 2005, many people heard about it mainly because of its extensive documentation. Unlike with other open source projects, you don't have to dig into the symfony code to understand how a method works or what a feature does. Documentation has always been a major concern among the symfony core team, and that's why we wrote this book: to leverage the adoption of the framework we initiated, to serve as a reference for the 1.0 release, and to allow enterprise use of a framework written primarily for professionals.

## Who This Book Is For

For this book, you need a basic understanding of PHP 5 and object-oriented programming. Of course, having already developed web applications in PHP before reading this book is a plus, for you will see this book as a collection of answers to the questions you regularly ask yourself.

## How This Book Is Structured

In this book, you will learn how to use symfony to build web applications. The chapters are grouped into five parts:

- *The Basics* covers all the general concepts and prerequisites for starting symfony.
- *The Core Architecture* describes how the three layers of the Model-View-Controller (MVC) architecture are implemented in symfony, and how to build pages and applications according to this separation.
- *Special Features* explains how to use symfony's mechanisms for shortening the development of smart URLs, forms and validation, Ajax interactions, caching, and internationalized applications.
- *Development Tools* covers the resources provided by symfony to make day-to-day development tasks easier: code generators, unit testing framework, command-line tasks, and plug-ins.
- *Becoming a Symfony Expert* reveals a few secret corners of the symfony code and shows you how to tweak the framework's behavior.

## Conventions Used in This Book

In the code examples, the names starting with `my` are just examples of actual names. For instance, you will see `myproject`, `myapp`, and `mymodule`. In your own code, replace these with the

real names of your project, application, and module. Not surprisingly, the words Foo and Bar (and sometimes FooBar) will be used as sample content for strings. Also, in syntax explanations, three Xs represent a sample name. For instance, validateXXX() is the syntax of a method that can be named validateUpdate(), validateEdit(), and so on.

Code lines longer than the book page width are indicated with a ➤ character. When you see this symbol at the end of a line, it means that you should read this line and the following as a single line. Here's an example:

```
<?php echo link_to('I never say my name', 'mymodule/myaction?name=anonymous', ➤
'class=foobar') ?>
```

Each command that should be typed in a \*nix shell or a Windows command line starts with a greater-than sign:

```
> symfony clear-cache
```

In some code examples, the output of a script when viewed by a browser is written directly after the script code, preceded by an arrow, as follows:

```
<?php echo "Hello, World!" ?>
=> Hello, World!
```

On the other hand, the output of a command appears between two horizontal separators, as follows:

```
> php symfony -V
```

---

```
symfony version 1.0.0
```

---

## Contacting the Authors

You can chat with François and Fabien on the #symfony IRC channel on freenode (irc://irc.freenode.net/symfony) or send them an e-mail at the following addresses:

```
francois.zaninotto@symfony-project.com
fabien.potencier@symfony-project.com
```

Be aware that they both live in France, so depending where you're writing from, there might be a delay in the response due to the difference in time zone.

# License

The symfony framework is available under the MIT license, reproduced hereafter.  
Copyright © 2004-2007 Fabien Potencier

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Symfony integrates and/or uses code from third-party projects: **Mojavi3** (MVC framework), **Propel** (object-relational mapping), **Creole** (database abstraction), **Phing** (CLI utility), **Unicode** (i18n), **Dynarch** (JavaScript calendar), **Prototype** (JavaScript library), **script.aculo.us** (JavaScript library), **famfamfam** (icons), **Prado** (i18n classes), **PHPMailer** (mail functions), **Spyc** (YAML parser), **pake** (CLI utility), and **lime** (testing framework). Refer to the online copyright file at <http://www.symfony-project.com/trac/browser/trunk/COPYRIGHT>.

