



Symfony

The Best Practices Book

Version: 3.4

generated on November 13, 2018

The Best Practices Book (3.4)

This work is licensed under the “Attribution-Share Alike 3.0 Unported” license (<http://creativecommons.org/licenses/by-sa/3.0/>).

You are free **to share** (to copy, distribute and transmit the work), and **to remix** (to adapt the work) under the following conditions:

- **Attribution:** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- **Share Alike:** If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license. For any reuse or distribution, you must make clear to others the license terms of this work.

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 SensioLabs 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.

If you find typos or errors, feel free to report them by creating a ticket on the Symfony ticketing system (<http://github.com/symfony/symfony-docs/issues>). Based on tickets and users feedback, this book is continuously updated.

Contents at a Glance

- The Symfony Framework Best Practices 4
- Creating the Project 6
- Configuration 9
- Organizing Your Business Logic 13
- Controllers 19
- Templates..... 23
- Forms 26
- Internationalization..... 30
- Security 32
- Web Assets 38
- Tests..... 40



Chapter 1

The Symfony Framework Best Practices

The Symfony Framework is well-known for being *really* flexible and is used to build micro-sites, enterprise applications that handle billions of connections and even as the basis for *other* frameworks. Since its release in July 2011, the community has learned a lot about what's possible and how to do things *best*.

These community resources - like blog posts or presentations - have created an unofficial set of recommendations for developing Symfony applications. Unfortunately, a lot of these recommendations are unneeded for web applications. Much of the time, they unnecessarily overcomplicate things and don't follow the original pragmatic philosophy of Symfony.

What is this Guide About?

This guide aims to fix that by describing the **best practices for developing web apps with the Symfony full-stack Framework**. These are best practices that fit the philosophy of the framework as envisioned by its original creator *Fabien Potencier*¹.



Best practice is a noun that means "*a well defined procedure that is known to produce near-optimum results*". And that's exactly what this guide aims to provide. Even if you don't agree with every recommendation, we believe these will help you build great applications with less complexity.

This guide is **specially suited** for:

- Websites and web applications developed with the full-stack Symfony Framework.

For other situations, this guide might be a good **starting point** that you can then **extend and fit to your specific needs**:

- Bundles shared publicly to the Symfony community;
- Advanced developers or teams who have created their own standards;
- Some complex applications that have highly customized requirements;
- Bundles that may be shared internally within a company.

1. <https://connect.sensiolabs.com/profile/fabpot>

We know that old habits die hard and some of you will be shocked by some of these best practices. But by following these, you'll be able to develop apps faster, with less complexity and with the same or even higher quality. It's also a moving target that will continue to improve.

Keep in mind that these are **optional recommendations** that you and your team may or may not follow to develop Symfony applications. If you want to continue using your own best practices and methodologies, you can of course do it. Symfony is flexible enough to adapt to your needs. That will never change.

Who this Book Is for (Hint: It's not a Tutorial)

Any Symfony developer, whether you are an expert or a newcomer, can read this guide. But since this isn't a tutorial, you'll need some basic knowledge of Symfony to follow everything. If you are totally new to Symfony, welcome! Start with *The Quick Tour* tutorial first.

We've deliberately kept this guide short. We won't repeat explanations that you can find in the vast Symfony documentation, like discussions about Dependency Injection or front controllers. We'll solely focus on explaining how to do what you already know.

The Application

In addition to this guide, a sample application has been developed with all these best practices in mind. This project, called the Symfony Demo application, can be obtained through the Symfony Installer. First, *download and install*² the installer and then execute this command to download the demo application:

Listing 1-1 1 \$ symfony demo

The demo application is a simple blog engine, because that will allow us to focus on the Symfony concepts and features without getting buried in difficult implementation details. Instead of developing the application step by step in this guide, you'll find selected snippets of code through the chapters.

Don't Update Your Existing Applications

After reading this handbook, some of you may be considering refactoring your existing Symfony applications. Our recommendation is sound and clear: **you should not refactor your existing applications to comply with these best practices**. The reasons for not doing it are various:

- Your existing applications are not wrong, they just follow another set of guidelines;
- A full codebase refactorization is prone to introduce errors in your applications;
- The amount of work spent on this could be better dedicated to improving your tests or adding features that provide real value to the end users.

Next: *Creating the Project*

2. <https://symfony.com/download>

[Click here to download full PDF material](#)