SilverStripe

The Complete Guide to CMS Development

Ingo Schommer and Steven Broschart

Translated by

Dr Julian Seidenberg and Ingo Schommer

WILEY

A John Wiley and Sons, Ltd., Publication

SilverStripe

SilverStripe

The Complete Guide to CMS Development

Ingo Schommer and Steven Broschart

Translated by

Dr Julian Seidenberg and Ingo Schommer

WILEY

A John Wiley and Sons, Ltd., Publication

Galileo Computing is an imprint of Galileo Press,
Bonn (Germany), Boston (USA).
German Edition first published 2008 by Galileo Press.

This edition first published 2009

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

ISBN 978-0-470-68183-1

A catalogue record for this book is available from the British Library.

Typeset in 10/12 Optima by Laserwords Private Limited, Chennai, India
Printed in USA

‘‘I feel SilverStripe is a great example of a well-constructed open source project.’’

Chris DiBona, Open Source Programs Manager, Google Inc.

Publisher's Acknowledgements

Some of the people who helped bring this book to market include the following:

Editorial and Production

VP Consumer and Technology Publishing Director: Michelle Leete

Associate Director - Book Content Management: Martin Tribe

Associate Publisher: Chris Webb

Executive Commissioning Editor: Birgit Gruber

Assistant Editor: Colleen Goldring

Content Editor: Claire Spinks

Publishing Assistant: Ellie Scott

Copy Editor: Andrew Finch

Composition: Laserwords Private Ltd

Marketing

Senior Marketing Manager: Louise Breinholt

Marketing Executive: Kate Batchelor

Contents

About the Authors

Steven Broschart has been active for 6 years as an advisor and developer at one of Germany's leading online marketing agencies, cyberpromote GmbH. Aside from developing PHP and Ruby on Rails business applications and coordinating key accounts for SEO, he advises his clients on selecting appropriate open source software. Steven is a regular author in relevant industry publications.

Ingo Schommer freelanced as a PHP and Flash developer for several years prior to joining SilverStripe in 2006 as a senior developer. At SilverStripe, Ingo analyses and builds modern web applications, making sure that they work well in a browser and not just on paper. He has a key role architecting and implementing core functionality in the SilverStripe platform, and facilitates involvement of the open-source community.

About the Translator

Julian Seidenberg has a background in developing web applications using PHP and Java and has worked in a variety of software engineering roles. He is also a developer at SilverStripe and holds a PhD in Semantic Web technology from the University of Manchester.

Foreword

Sigurd Magnusson

Co-founder and Chief Marketing Officer, SilverStripe Ltd

Thanks for buying the book! After countless long days and nights, it's a real joy to see it come to fruition.

In 2000, the dream of the *SilverStripe* project began: to make a fantastic software platform to build and manage websites. Our company *SilverStripe Ltd*, which had its fair share of geeks, was in the business of building websites for customers. We found the tools out there didn't suit us, and certainly didn't suit our customers. Tools were either too technical, which suited us fine but confused our clients, or were simple enough for our non-technical customers, but insulted us with poorly constructed HTML and limited us from building sophisticated website features. So, we committed to make a system that let professional, experienced web developers build innovative websites which can then be passed, in confidence, to non-technical people to manage.

Several years passed, and by 2006 we had a few hundred customers. We had always sought to make the product global, but because we were the only one developing SilverStripe, adoption was limited to our home country, New Zealand. We were working hard on an entirely rewritten version of SilverStripe, v2.0, and agreed that if we were to make it successful, it had to be released as free software, with a loyal community of developers around it.

In late 2008, we celebrated two years of being open source. It was inspiring to see how quickly we'd moved forward. We had surpassed 100,000 downloads, been invited into the *Google Summer of Code*, had a growing community with external contributors, and our software was powering some impressive websites. For example, a SilverStripe-powered website, `http://demconvention.com/`, served millions of visitors in a 100 hour span as Barack Obama accepted his nomination as the presidential candidate for the United States Democratic Party.

However, writing the software is only half the battle: you also need documentation that teaches the world on how to take advantage of it. If there was one thing I wanted to improve, it was our documentation. So I'm very humbled by the months of work that Ingo and Steven have put into writing a book while holding down their 9 to 5 jobs. Their effort provided the German text that has now been translated into the English edition you're holding. It was very rewarding to see the book come out initially in Germany, because it showed just how international SilverStripe had become, and that a country renown for high quality engineering had given the thumbs up to SilverStripe! With this book now in English, we shift SilverStripe into a new gear, which is very exciting.

The concept of SilverStripe relies on us building up a community of developers, and so we hope that over time you will join us by emailing feedback, submitting code, and fixing bugs. We also hope to see your websites uploaded to the community showcase at `http://silverstripe.org/showcase/`!

In the years since we started SilverStripe, the concept of elegant, web-standard HTML and CSS has shifted from obscurity to being the norm. The idea of writing PHP in an elegant maintainable fashion, complete with object-oriented code, unit tests, and following the MVC pattern is still new to lots of people, but growing in popularity. However, having read the book, we hope that you can appreciate the other evolution of the web we've been anticipating for years: that the combination of a framework, a CMS product, and extension modules can together save developers a huge amount of development time by offering what they need out-of-the-box, while offering a huge potential for extension and innovation.

Welcome to the SilverStripe community!

Preface

Steven Broschart

Co-author of this book, and early member of the SilverStripe open source community.

It wasn't too long ago when a fellow developer at work asked me to look at his screen. He had stumbled across some new technology, and asked: 'Hey, have you heard about this?' All too often such questions are answered with a 'Yeah. Completely useless!', but sometimes the answer is 'No'. It's in those times that a feeling of anxiety creeps in – did I miss a potentially groundbreaking development on the Internet? In this case I decided to defer my answer, and stopped what I was doing to investigate further.

It's a good thing to keep in mind that in our current working environment every person is basically a human content-filter who must grapple with the daily excess of digital information. New products and ideas confront us so often, they don't get the attention they deserve. Hence, a presentation of a 'cool new thing' has only a few seconds of your attention in which to grab your interest, as every marketing guy will tell you.

As you might have guessed, my colleague was showing me SilverStripe for the first time, and he succeeded in quickly gathering my interest. The fact that you're reading this book probably means that you're starting with a certain eagerness towards this 'new thing', too.

For several years now, I've been a consultant and developer in an online marketing agency that specializes in search engine optimization and usability. This means that I've developed a number of internal systems and a fair share of websites. A major internal focus was ensuring a quick and flexible development process. In this context we trialled a bunch of content management systems – scoring them against our set of requirements. Having reviewed several systems so far, I couldn't wait to see what SilverStripe had in store. It seemed to be sufficiently different from the 'big players' to capture our interest. The ease with which usually complex tasks were solved in SilverStripe was promising and exciting. We liked what we saw and our agency began adopting SilverStripe in a big way. Since then, our experience has supported our positive first impression.

During the last couple of years, certain programming concepts have become very popular in the website and web-application development space. In particular, people have become far more interested in rapid development, technical elegance, robustness, and building increasingly interactive websites. Although these concepts aren't necessarily new, most web developers have a fairly informal development methodology – especially those writing in PHP. Writing code in a well-structured way was not common place. One event that influenced this trend was a framework called *Ruby on Rails* (`http://www.rubyonrails.com/`), which was increasing in popularity. This raised the game for everyone: The carefully structured architecture, methodology, and principles of that framework rubbed off and became best practices, which the whole web industry began to discuss, embrace, and challenge.

SilverStripe is written in PHP and has an architecture very much modeled on these contemporary best practices. SilverStripe combines the advantages of a CMS application with a powerful framework 'under the hood'. Time-intensive programming sessions for conceptually simple features should be a thing of the past, which shouldn't only delight the programmer, but also the manager who has to justify budgets and timeframes. After all, time and money are some of the primary factors determining the success of a project. After reading through the first tutorials it became clear to me that in this case these weren't just empty marketing phrases. Because care was put into its architecture, this product was able to give me a glance at advanced functionality without knowing its inner workings. Its shallow learning curve dwarfed many other competitors in the same space.

Anyway, back to the story. Apart from playing around with new shiny web toys, I'm a freelance author in Germany on subjects from search engine optimization to usability to web technologies. Driven by my first SilverStripe experiences, in 2008 I published an article on SilverStripe in a popular German industry magazine called *PHP Magazin* (see announcement at `http://silverstripe.com/silverstripe-main-feature-of-german-php-magazin/`).

While writing this article, I made contact with the company behind the software to check a few facts – the aptly named web development shop *SilverStripe Ltd*, based in Wellington, New Zealand. In particular I reached out to Ingo Schommer, a senior developer at the company. Ingo is a fellow German, who became attracted to SilverStripe after battling other open source CMS products – software too often plagued by complex configuration and the feeling of spending more time disabling features than actually implementing stuff. Emigrating from Germany to New Zealand in order to change this situation might seem like a desperate measure, but it worked out well.

The magazine article was so well received that it supported the idea of writing the book you're now reading. There was no print material to show interested developers a straightforward learning path to implement their own solutions in SilverStripe. The magazine article demonstrated demand, and so I asked Ingo to be a co-author, and sought a German publisher, *Galileo Computing*. Both promptly agreed to produce the book.

Writing the German book took several months of intense collaboration across continents (from Wellington to Munich and back), by which time the CMS had become increasing popular in the German market – which geographically couldn't be much further from New Zealand! The German book was released in early 2009, and sold sufficiently quickly that it had to be restocked on *Amazon.de* days after the book launched.

Of course, there's a larger audience of English readers for a SilverStripe book, and plans by SilverStripe Ltd to have an English book predated me even learning about their software. Translating the German book provided a sensible way to finally achieve this goal, and what you're reading now is the work of several SilverStripe Ltd staff members adapting the work into English.

Now please follow us on a quick tour through SilverStripe, and we'll show you what's behind the buzz. We hope that you have as much fun reading this book as we had writing it. All right, that was quite a stereotypical way to finish the preface of a technical book, but anyway: Enjoy!

So, to answer my colleague: Thanks for your discovery, Robert. No, didn't know this system!

Acknowledgments

Although the names of the authors are prominently placed on the cover of a book, the reality is that such an exhaustive guide couldn't happen without further support. Special thanks go towards our original publisher in Germany, Galileo Computing, especially to our editor Stephan Mattescheck who supported us throughout what happened to be the first book publication for the two of us. And consequently, a big thank you to Wiley Publishing for picking up this work and agreeing to make it available to a much larger English-speaking community, particularly to Martin Tribe and Colleen Goldring, our editors at Wiley, for their patience and giving us a chance to polish content rather than translate word-for-word. 'Dankeschön' to Sigurd Magnusson, who helped us navigate the unfamiliar waters of book publications and spent long nights reviewing content – the book got a lot more 'digestible' with your input! Thanks to Andy Finch for copy-editing our 'Germenglish', and to Sibylle Schwarz for keeping us focused. We'd like to thank SilverStripe Ltd, who 'lent' us considerable resources for the translation effort. Massive thanks to Julian Seidenberg for unexpectedly spending his first weeks as a developer with SilverStripe Ltd. translating the book rather than writing code. Kudos to our 'beta'-readers Andy Adiwidjaja, Claudia Liersch, Joon Choi, Tobias Hoderlein, Sam Minnée; without your comments and ideas the book would've been only half as good. And 'last but not least' thanks go to our families and friends, who lost us to text processing and late night Skype meetings for several months – you're the best!

Target Audience

This book mainly targets PHP developers who're looking for new solutions to quickly build websites, big and small. Because SilverStripe uses some techniques and paradigms that aren't necessarily widespread knowledge to PHP developers, we make some suggestions here for recommended further reading.

If you're not a developer but are evaluating SilverStripe as a platform, you should still find many parts of this book useful. The book doesn't delve deep into code until Chapter 5, 'Development', and even then, there are plenty of explanations, screenshots, and summaries that will give less technical people meaningful insight into SilverStripe.

Of course we also address the actual users of the system, the content authors who manage websites with this tool. If this is you, Chapter 4, 'First Steps', is the one to focus on, because this is where we provide a summary of the CMS content editing interface.

For the more technical chapters of the book, we assume that you have some experience with PHP and that you're comfortable with object-oriented programming. Perhaps you're already familiar with Java, in which case you should feel right at home with the object-oriented capabilities of PHP. If you feel a bit lost with terms such as *objects, classes, interfaces*, or *inheritance*, we'd like to recommend a book covering these basics: *PHP 5 For Dummies* by Janet Valade (published by Wiley; ISBN: 978-0-7645-4166-7). The official PHP website at `http://php.net/`can of course also be used as a primer.

SilverStripe is based on PHP version 5, released in 2004. Because it uses the full capabilities of this version, much of its architecture is impossible to achieve in the outdated PHP4. We'll describe some of these capabilities as we go along, but developers who are used to PHP4 might need to brush up their knowledge about the object-oriented coding to take advantage of this book.

Because this book is aimed at developers, we also assume a basic familiarity with standard web technologies such as HTML, JavaScript, and CSS. We don't ask you to write anything advanced with them in this book – although once you've finished the book, having strong skills in those languages will enable you to make great SilverStripe websites!

Book Structure

This book begins with a mostly non-technical review that describes the underlying principles, technologies, and architecture, as well as the installation of SilverStripe. After everything is set up both conceptually and on your own webserver environment, we dive into code with an example website and application. Building this example project will span multiple chapters and provide a coherent learning path. The later chapters become more independent of this example and can be worked through in isolation if you wish – they relate to topics such as internationalization, security, and performance. The book ends with a set of code recipes that demonstrate useful techniques, and a summary of the most useful extension modules.

Book website: Updates and source code

We've tried our best to provide a perfect book without any mistakes. If that's not the case every now and then, you can find errata and corrections as well as up-to-date information on a website dedicated to the book: `www.wiley.com/go/silverstripe`.

The website is also where you can download the source code mentioned throughout the book.

Terminology: CMS, frontend and backend

- *Content Management System (CMS)*: Although we sometimes refer to the entire software as 'the CMS', in the book this term normally refers to the graphical user interface that the software provides to manage your website content.
- *Backend*: The generic name for an authenticated interface, with the CMS being one prime example.
- *Frontend*: The public-facing website.

Versions, Cutting Edge, and Bleeding Edge

What might sound like the title of a horror film on first reading is actually a way to describe the status of a technology project. If you hear the term *Cutting Edge* in relation to new features in software, you're dealing with fairly new developments, which most likely aren't yet contained in a stable release. *Bleeding Edge* is even further ahead of this, meaning that features and implementations are subject to change, and probably need some further testing – you can metaphorically cut yourself if you're not careful.

We raise this topic because toward the end of the book, you come into contact with some of the fresher and younger features of SilverStripe. Due to the software's modular architecture, it's easy to update individual components later as such features mature.

In our code examples we're referring to SilverStripe 2.3.3, which was released mid-2009. If you're using a newer version to follow our examples, please make sure to read the update notes at `http://doc.silverstripe.com/doku.php?id=upgrading`. If you need more details, please have a look at the changelog at `http://open.silverstripe.com/wiki/ChangeLog/`.

Coding Conventions

Code shown is this book is typically standard PHP, which follows the official SilverStripe coding conventions that you can read online at `http://doc.silverstripe.com/doku.php?id=coding-conventions`. It should therefore need no further explanation. Something we do want to point out is how we treat updating files you've already worked on – we generally show the new code without repeating the whole listing. The existing (but not shown) code is noted by an inline comment. Leave your current code as is, and write around it:

```
// ... PHP/JavaScript Omission
/* ... */ CSS Omission
<!-- ... --> SilverStripe Template Omission
```

But enough introduction, let's get on with the tour!

1

Introduction

At it's heart, the role of a content management system (CMS) is to make it easier to publish content to websites and intranets. It may be used to allow the owner of a one-person company to update his or her website without needing a website developer or learning HTML skills. Or, in a multinational company, it might allow a skilled web team across various cities to manage a complex assortment of updates to products, services, promotions, and news in multiple languages. Either way, it automates various tasks, and makes building a website more efficient and reliable.

Countless products are available based on such a promise, however, all of varying sophistication, histories, programming languages, and geographical dominance. The decision-makers and developers involved in selecting a CMS work in fast-paced environments and are creatures of habit: They have personal favorites to solve the challenges their job provides. Why go through the trouble of trying a new CMS?

1.1 Why SilverStripe?

One thing up front: SilverStripe is not the answer to all problems that you might face throughout the development of a website or web application. No software is ever a silver bullet for *every* situation. You must determine selection criteria based on your needs, skills, budgets, and timeframes, and select the most relevant tool based on those criteria. That said, SilverStripe is *intended* to be both used out-of-the-box and for major customization. This makes it particularly interesting when building websites – because everyone seems to relish having a unique set of needs in terms of design, information architecture, content, features, integration with other systems, business policies, and so on.

The following pages list some major benefits of SilverStripe, and describe how it stands out in a crowded CMS market. We acknowledge that the information here is concise, but this will help you refer back to an overview even when you're familiar with SilverStripe. We try to keep things brief here; the rest of the book will substantiate the claims made, as you begin to build stuff using the software.

1.1.1 An Application and a Framework

SilverStripe is really two products in one: The *SilverStripe CMS* and the *Sapphire Framework*. There are plenty of CMS applications and quite a few programming frameworks in the marketplace; however, SilverStripe is very rare in that it *tightly* weaves the two concepts together.

You will need to read the whole book to understand entirely the value of this unity, but the value is much greater than the sum of its parts. In other words, the CMS has plenty of good things about it, as does *Sapphire.* However, with them joined so closely, their collective value is multiplied. The closeness of the two concepts is possible because *Sapphire* and the CMS were created in tandem, by the same people.

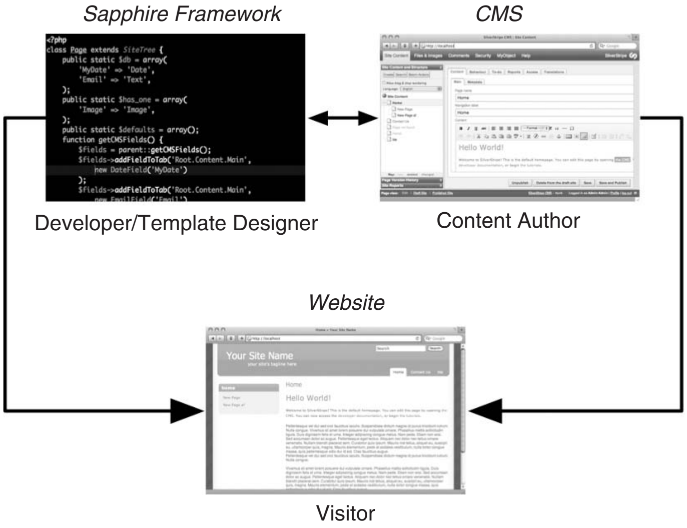
1.1.2 CMS for Content Authors and Framework for Developers

One fundamental reason for SilverStripe being divided into two sections is to honor the needs of two quite separate types of people who use the software. The CMS backend is designed to allow non-technical users to update web pages, images, links, moderate blog comments, and so on. This is the area where content authors do their content management, and anything technical is strictly kept out of sight for the sake of good usability.

The framework is used to build the underlying website, and isn't for content authors. You 'use' the framework by actually writing HTML, CSS, JavaScript, and object-oriented PHP code. The framework enables beginning PHP programmers to achieve quite powerful things, by following recipes and tutorials as shown in this book and elsewhere. However, the framework is targeted at savvy PHP programmers. For them, *Sapphire* allows much more creative freedom in building complex websites quickly and robustly.

Many other CMS products don't have such a separation. In other words, in many systems the CMS user interface is shared by content authors and developers, which typically makes the interface too complicated for it's primary users: the content authors. In some systems, developers perform their work mainly through a graphical configuration interface, which is usually more limiting than raw programming. A graphical interface to accommodate sophisticated needs for developers would require a bewildering set of options, which would add complexity and bloat the software, when the idea in question is likely to be far more efficiently expressed as a few lines of code.

So, this separation in SilverStripe means that the CMS backend can be designed for non-technical users, and the framework can confidently concentrate on being suitable to technically-minded web developers (see Figure 1.1 for an overview of the SilverStripe system).



```
<?php
class Page extends SiteTree {
    public static $db = array(
        'MyDate' => 'Date',
        'Email' => 'Text',
    );
    public static $has_one = array(
        'Image' => 'Image',
    );
    public static $defaults = array();
    function getCMSFields() {
        $fields = parent::getCMSFields();
        $fields->addFieldToTab('Root.Content.Main',
            new DateField('MyDate')
        );
        $fields->addFieldToTab('Root.Content.Main',
            new EmailField('Email')
```

Figure 1.1 SilverStripe system overview.

1.1.3 Clear Separation of Concerns

As we begin to suggest above, a core principle in SilverStripe is the *separation of concerns*. In this introduction, we use that term loosely to mean organizing the software into all sorts of logical divisions. You will be familiar with how HTML and CSS separate the concerns of markup and presentation, and that JavaScript is used to allow for the concern of behavior and interaction on a web page. This structure provides many benefits such as enabling reuse: change a single style in a CSS file, and so long as you've stuck to best practices in your HTML, styles throughout the whole website will update, saving time and maintaining consistency.

SilverStripe follows this principle: Update an isolated area of the system that deals with a particular function, and it produces logical, robust, and application-wide changes. Without this architectural principle, changing a feature of your website means changing a few lines of code here, a few lines of code there, and finding that each step of the way upsets unrelated parts of your website. This domino effect suddenly turns a small change into a major coding and testing fiasco. SilverStripe's architecture therefore helps to keep code changes proportionate to the effort needed to implement them. However, like the HTML, CSS, and JavaScript example, this reward is only given to those who carefully architect their SilverStripe application to its best practices.

Let's highlight a few examples relating to this principle:

- ***Avoid repitition with the Sapphire framework.*** *Sapphire* supports the concept of writing the logic of your website *once*. For instance, if your website deals with selling T-shirts, you define that a T-shirt has an attribute 'color' in one place in your code. You don't have to write redundant code elsewhere to explain that T-shirts have colors; the template, CMS, and even the database will automatically be updated based on your single declaration. This principle is commonly called *Don't repeat yourself* (*DRY*).
- **No need to change database schema manually.** This important point is hinted at above: SilverStripe will inspect your code for the tables and fields it needs, and update the schema accordingly. This makes keeping your database schema in sync with your PHP code a beautiful thing.
- **Separates the PHP logic into the Model and the Controller.** You're probably familiar with separating PHP logic from HTML templates, but SilverStripe goes one step further. This is a very integral part of *Sapphire*, so let's pause for a moment to investigate this in the following box.

Model View Controller as a central concept

The *Model View Controller* (MVC) concept is one of many *design patterns* that have been identified as good programming architecture. A design pattern is essentially a structured way to document the solution to a specific problem in computer science, a term made popular by the 'Gang of Four' in their book *Design Patterns: Elements of Reusable Object-Oriented Software* (Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, published by Addison-Wesley). The three parts, *Model, View*, and *Controller*, describe the main roles of any application built with SilverStripe:

- The *Model* is responsible for maintaining the state in the application. It's basically data: Some of it might be transient and only live within a web request, whereas other data might be permanently stored in a database. You can think of the *Model* as essentially the database