开发大规模*Web*应用（影印版）

Developing

Large Web Applications

O'REILLY®
YAHOO! PRESS
東南大學出版社

Kyle Loudon 著
Nate Koechley 序

开发大规模 Web 应用（影印版）

Development Large Web Applications

Kyle Loudon

Foreword by Nate Koechley

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

图书在版编目（CIP）数据

开发大规模Web应用：英文/（美）劳登（Loudon, K.）著. —影印本. —南京：东南大学出版社，2011.1

书名原文：Developing Large Web Applications

ISBN 978-7-5641-2495-3

Ⅰ.①开… Ⅱ.①劳… Ⅲ.①主页制作－程序设计－英文 Ⅳ.①TP393.092

中国版本图书馆CIP数据核字（2010）第209442号

江苏省版权局著作权合同登记

图字：10-2010-284号

开发大规模Web应用（影印版）

出版发行：东南大学出版社
地　　址：南京四牌楼2号　　邮编：210096
出 版 人：江建中
网　　址：http://www.seupress.com
电子邮件：press@seu.edu.cn
印　　刷：扬中市印刷有限公司
开　　本：787毫米×980毫米　16开本
印　　张：19印张
字　　数：372千字
版　　次：2011年1月第1版
印　　次：2011年1月第1次印刷
书　　号：ISBN 978-7-5641-2495-3
印　　数：1~1700册
定　　价：52.00元（册）

本社图书若有印装质量问题，请直接与读者服务部联系。电话（传真）：025-83792328

Foreword

As a little kid, I wondered if I would be big and strong when I grew up. There were a lot of aspects to growing well. Would I be healthy? Useful? Productive? Successful?

Websites start out small, too. But these humble sites share my childhood dreams. They want to help more people in more ways; they want to be durable and reliable; they want to be indispensable and to live forever. In short: they want to be large and successful.

But growing up is hard to do. Challenges accumulate and complexity snowballs.

> Expansion means complexity and complexity decay.
>
> —C. Northcote Parkinson

I've seen it. The inevitable challenges of growth in websites—data management, performance—become crippling if mishandled. Things you thought were straightforward, like HTML, start giving you headaches. From front to back, JavaScript to PHP, harmony is displaced by dissonance.

> Fools ignore complexity. Pragmatists suffer it. Some can avoid it. Geniuses remove it.
>
> —Alan Perlis

I've worked hand-in-hand with Kyle on some of the Web's largest applications. I've watched him craft CSS systems to make sprawling sites skinable and design Ajax architectures that adapt to and enhance the sites. He emerges from the trenches on top every time. He's a perpetual teacher, and, like the best in any discipline, also a perpetual student. We all benefit from his expertise.

Kyle shares his genius and hard-won expertise in this valuable book that will prepare you and your application for scale and success. The book is well structured and readable, with memorable tenets supported by savvy insights, sound philosophy, and fully functioning code examples. Complexity is inevitable, but success rewards the prepared.

> The way to build a complex system that works is to build it from very simple systems that work.
>
> —Kevin Kelly

During this book's deft tour of the complete web application stack, Kyle, the perfect guide, converts lines of explanatory code from one context into insightful tips in another. Build big by thinking small. Build new by thinking old. Manage scope. Boost signal and reduce noise. Resist breakage...these things are easy to rattle off, but it takes an author like Kyle, and a book like this, to make them practical and real.

If you're ready to build a finely crafted large site, this is the book for you. Learn what it takes, because today's compromise is tomorrow's constraint. Start today, because the world is waiting for your application.

Grow large and prosper.

—Nate Koechley
San Francisco, January 2010

Preface

It's been a while since I first worked on a book with O'Reilly in 1997. That book was a practical guide to data structures and algorithms, a subject that, for the most part, had been defined many years before by some of the early giants of computer science (Dijkstra, Hoare, Knuth, to name a few). By comparison, I've been able to witness the rapid evolution of the subject of this book from the front lines, and I have had the good fortune to help refine it myself while working as a web developer at one of the largest web applications in the world, Yahoo!.

Web developers have a fascinating role. We work just as closely with user experience designers as with engineers, and sometimes we're the designers, too. In many ways, we are guardians of the user experience as a web design goes from its mockup to its implementation. But we also have to write exceptionally good code that performs well in the challenging environment of web browsers. Today, more than ever, engineers recognize that web development must be carried out with the same rigor as other types of software development.

This book presents a number of techniques for applying established practices of good software engineering to web development—that is, development primarily using the disparate technologies of HTML, CSS, JavaScript, and server-side scripting languages. Whereas there are many books on how to use languages, how to use libraries, and how to approach software engineering, this is the first book to codify many of the techniques it presents. These techniques will make the components of your own web applications more reusable, maintainable, and reliable.

Audience

The primary audience for this book is software developers and managers interested in large web applications; however, you'll find that the techniques in this book are equally useful for web applications of any size. Although it's especially important to follow good development practices in large web applications, smaller web applications benefit from many of the same techniques, too.

To get the most out of this book, you should already be very familiar with HTML, CSS, and JavaScript; this book does not teach these languages, although it covers many interesting aspects about them. This book uses PHP as the scripting language for server-side examples. Many readers will have a good understanding of PHP as well, but even those who don't should find the examples easy to follow. PHP is known for its flexibility, ubiquity, and ease of use, so it works well. Most examples can be translated to other server-side scripting languages fairly easily, if you desire.

Organization of This Book

This book is organized into three types of material: background (e.g., Object Orientation in Chapter 2), techniques associated with specific languages (e.g., Large-Scale HTML in Chapter 3, Large-Scale CSS in Chapter 4, Large-Scale JavaScript in Chapter 5, and Large-Scale PHP in Chapter 7), and techniques related to other aspects of development (e.g., Data Management in Chapter 6, Large-Scale Ajax in Chapter 8, Performance in Chapter 9, and Application Architecture in Chapter 10). Each chapter begins with a tenet presented from Chapter 1. These tenets act as assertions about the topic for each chapter to provide a concisely articulated direction.

Throughout the book, there are numerous examples in real code to demonstrate many of the techniques presented. Some of the numbered examples work together to create larger, more complete examples that extend across multiple chapters. While the focus of this book is not on teaching the specific languages addressed, the examples do demonstrate a number of aspects of each language that will help make you more proficient with each as you master them.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic
: Indicates new terms, URLs, filenames, and Unix utilities.

`Constant width`
: Indicates command-line options, variables and other code elements, HTML tags, the contents of files, and the output from commands.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

`Constant width italic`
: Shows text that should be replaced with user-supplied values.

This icon signifies a tip, suggestion, or general note.

There are some other conventions to be aware of in this book:

`...`
: Indicates something that is missing (for you to fill in) in a line of code or a path (e.g., `require_once(.../navbar.inc);`).

`<?php ... ?>`
: Wraps PHP examples that contain the complete code for a file. Most PHP examples don't have this, because they show only a code snippet.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books *does* require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation *does* require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Developing Large Web Applications*, by Kyle Loudon. Copyright Yahoo!, Inc., 978-0-596-80302-5."

If you feel your use of code examples falls outside fair use or the permission given here, feel free to contact us at *permissions@oreilly.com*.

We'd Like to Hear From You

Every example has in this book has been tested on various platforms, but occasionally you may encounter problems. The information in this book has also been verified at each step of the production process. However, mistakes and oversights can occur and we will gratefully receive details of any you find, as well as any suggestions you would like to make for future editions. You can contact the author and editors at:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international or local)
(707) 829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

http://www.oreilly.com/catalog/9780596803025

You can find additional information about this book, including electronic versions of the examples at:

http://kyleloudon.com

To comment or ask technical questions about this book, send email to the following address, mentioning its ISBN number (978-0-596-80302-5):

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our website at:

http://www.oreilly.com

Safari® Books Online

Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at *http://my.safaribooksonline.com*.

Acknowledgments

This book is the result of having worked with outstanding people both at O'Reilly and in many projects leading up to the book. For this, I offer my heartfelt thanks.

First, I thank my editor at O'Reilly, Andy Oram. Andy and I worked together on my first book with O'Reilly, and I had hoped for a long time that working on another book together would not be a matter of if, but when. Having finished this book, I hope that another book will be a matter of when again. Andy inspires me with his ability to always find ways to make things better. His insights appear in one form or another on nearly every page of this book. Andy also kept our project moving along while being patient and understanding of the struggle that writers doing other jobs constantly face.

I also extend my sincere thanks to the entire production team at O'Reilly, who constantly impress me with their ability to handle the numerous aspects of production so smoothly. The ease with which it all seems to take place belies the work that it really requires. I would also like to thank Amy Thomson, my copyeditor, for having worked under such tight time constraints at the end of the book.

I send my heartfelt thanks to Nate Koechley for writing the foreword. Nate was one of my earliest colleagues at Yahoo! to turn me on to the truly awesome potential of web development. Much of what I've tried to capture in this book came from ideas that Nate worked passionately to instill at Yahoo! and across the Web. I couldn't have asked for a more fitting person to write the foreword.

I am grateful to have had outstanding technical reviewers for this book as well. Christoph Dorn, Steve Griffith, and Nate Koechley each provided an impressive level of detail and thought in their reviews. The book benefited greatly from their comments.

I would also like to acknowledge the influence of my many colleagues at Yahoo! and other projects before this. I especially thank Bryce Kujala and Vy Phan, who helped refine many of the ideas in the book by putting them to the test in practice early on. I'm also grateful to the exceptional user experience designers with whom I've had the honor to work closest: Veronica Gaspari, Cathy Tiritoglu, and Sasha Verhage.

Finally, I thank Shala, my wife, for her encouragement on another book project; my parents, Marc and Judy, for their support from afar; Shala's parents, Elias and Maria, for their frequent assistance at a moment's notice; and Julian, who has been my late-night companion—just too young to know it yet.

Table of Contents

CHAPTER 1
The Tenets

As applications on the Web become larger and larger, how can web developers manage the complexity? In many ways, we need to turn to some of the same good practices used in other types of software development. Generally speaking, these practices are not yet pervasive in web development—that is, in software development primarily using HTML, CSS, JavaScript, and various server-side scripting languages (we'll use PHP for the server-side scripting in this book, but the same principles apply to many other languages). Furthermore, the uniqueness of these technologies poses a challenge for developers trying to apply good practices in a cohesive way.

One of the themes that you'll see repeated in this book is the importance of extending modular development practices to web development. This book presents concrete, practical techniques to achieve modularity in large web applications. In the process, we'll explore many of the finer aspects of HTML, CSS, JavaScript, and PHP. You'll find that most of the techniques are relatively simple to apply, and none rely on the use of specific frameworks. That said, it's important to realize that they don't preclude you from using various frameworks, either; to the contrary, these techniques create a better landscape in which to use many frameworks. As a case in point, we'll look at several examples that utilize the Yahoo! User Interface (YUI) JavaScript library.

At the outset, it's important to establish why the techniques that we're going to explore in this book are especially useful for web developers working on large web applications. We'll begin by looking at some of the factors that contribute to the complexity of many large web applications. Then we'll explore how modularity plays an important role in managing this complexity. Last, we'll examine a list of tenets that will guide our discussions throughout the rest of the book.

Managing Complexity

If you consider how different the Internet is today from just 10 years ago, it's clear how complicated web applications have become and just how quickly the changes have taken place. Far too often, this complexity makes large web applications difficult to

maintain, less reliable, and more costly over their lifetimes. Let's examine some factors that contribute to the complexity of many large web applications. Typically, large web applications have the following characteristics:

Continuous availability
: Most large web applications must be running 24/7. In addition, response times have to be fast at any moment, even at peak load times. Web developers need to write code that is especially robust.

Large user base
: Large web applications usually have large numbers of users. This necessitates management of a large number of simultaneous connections or layers of caching. Web developers often need to write code to manage these situations.

Piece-by-piece delivery
: Whereas many types of software are distributed as complete units, web applications have many parts delivered page by page, or connection by connection via Ajax. As a result, large web applications operate within an environment effectively shared by a huge number of users.

Diversity
: It's hard to think of a business or service that doesn't have at least some sort of web interface. For example, we see financial applications, ticketing sites, applications that organize massive amounts of data (e.g., search engines), media systems (e.g., news sites), and the list goes on. Web developers need to write code that may be reused in unexpected places.

Longevity
: The largest web applications today, even those that have been around many years, are just at the beginning of their lifetimes. Web developers need to write code under the assumption that it will have to stand up to years of changes and maintenance.

Multiple environments
: The Web is a fast-changing landscape littered with old browsers and other devices that can be difficult to support. Users access large web applications from all types of environments and with screens of wildly different sizes (including mobile devices). Web developers must write code that can handle the numerous idiosyncrasies that result from this.

Real-time updates
: Large web applications are not static; they are constantly fluctuating applications for which changes are typically pushed to servers regularly. Web developers need to write code to address this potential for moving parts.

Over time, web developers often end up addressing complexity in large web applications via one-off fixes and tweaks as their applications reach various breaking points. But there is a better way. This book will show you how to address challenges like the ones above head-on from the start. Mitigating the complexity from these challenges can often be attributed to one or more byproducts of modularity that we'll examine in