

使用新一代框架技术构建功能强大、性能稳健的PHP站群

- ◎ 全面认识MVC开发带来的高效率
- ◎ 帮助读者编写优雅、高质量的PHP代码
- ◎ 使用PHP开发性能超强的网站
- ◎ 解决PHP团队开发难点
- ◎ 轻松构建分布式的PHP网站群

PHP MVC 开发实战

李开涌 ◎ 编著



机械工业出版社
CHINA MACHINE PRESS

PHP MVC 开发实战

李开涌 编著



机械工业出版社

MVC 是一种先进的开发模式,能够解决团队开发之间协同配合的问题,使得网站各部件以更高的效率运行。MVC 模式将网站分为 3 大部件,分别为模型、视图、控制器。这 3 大部件各自分离,但又相互依存,最终形成了一个容易维护、容易扩展、高效运行的网站平台。对于后台程序员,借助于 MVC 模式就可以更加专注于功能的实现,而不需要太多地涉及页面与前端。这种分工协作的最终目的是提高开发效率及项目质量。对于个人项目,也许在其他编程技术(例如 Java、Python)中,MVC 模式并没有优势,但在 PHP 中,由于支持混合编程,所以使用 MVC 模式进行编程,能显著提高工作效率。

本书是国内第一本专门介绍 PHP MVC 开发模式的图书,全书围绕 MVC 实现思路进行细致的讲解。通过 MVC 编程模式,以点带面,全面深入探讨 PHP 核心技术。同时,本书也是一本深入介绍利用 PHP 构建高性能网站的图书,通过 MVC 的数据库中间件,可以轻松实现网站群体、读写分离等高级应用,本书在此基础上还会进一步介绍当前流行的 NoSQL 应用、全文搜索应用等。最后,作者通过一个自行编写的 MVC 框架,引导读者开发属于自己的 PHP MVC 框架。

本书内容通俗易懂、示例形象,适合广大的 Web 从业人员阅读。由于 PHP 非常简单、易用,所以就算是未接触过 PHP 的读者或者初学者,只要掌握了基础的面向对象编程思想就可以轻松上手。

图书在版编目(CIP)数据

PHP MVC 开发实战 / 李开涌编著. —北京:机械工业出版社, 2013.6

ISBN 978-7-111-42852-7

I. ①P… II. ①李… III. ①PHP 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 125415 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:时 静

责任编辑:时 静

责任印制:乔 宇

三河市宏达印刷有限公司印刷

2013 年 7 月第 1 版 · 第 1 次印刷

184mm × 260mm · 38.75 印张 · 961 千字

0001 ~ 3000 册

标准书号:ISBN 978-7-111-42852-7

定价:98.80 元



凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

教材网:<http://www.cmpedu.com>

销售一部:(010) 68326294

机工官网:<http://www.cmpbook.com>

销售二部:(010) 88379649

机工官博:<http://weibo.com/cmp1952>

读者购书热线:(010) 88379203

封面无防伪标均为盗版

前言

我的编程之路

我本身是学习动画设计的，一次偶然的机会，接触到了计算机编程。记得那是 2003 年的时候，我刚拥有自己的第一台计算机，通过老师的介绍，学会了上网。那年暑假，在广州的计算机城购买了一套洪恩教育软件，认识了网页编程。那时的网页编程技术主流的是 ASP，由于 ASP 简单、易学，很快我就使用 ASP 技术构建了我的第一个网站（名称叫木棉休闲站，现已关闭），我的网站主要介绍天文知识，由于那时的 ASP 空间很贵，而且多数都不支持 FSO 组件，所以在实现图片上传时都是原图上传的（没有压缩等前期处理），随着访问人气的越来越高（大约 30 个请求量），很快网站就挂了。

后来我应聘到一家楼盘做网络管理员，并且负责文案录入，期间接触到了 Apache 服务器。我发现其网站后台也是使用 Web 设计的，可以对图片进行前期的压缩、裁剪、加水印等操作，而且单一服务器竟然轻松应对 1000 个以上请求量，而对外提供查询服务的正是 PHP 脚本。于是每到周末，我就到广州购书中心苦寻有关 PHP 的书籍，但是那时的 PHP 资料非常少，多数都是关于 JSP 以及 ASP 的。

2004 年初，随着国内网络的普及，很快网络上就活跃了一批开源爱好者，他们乐意将自己的学习成果分享到网络上，也就从那时起，我接触到了 PHP。由于早期的 PHP 在语法上与 C 很相似，所以我很快就上手了（在学校时我业余学习过 C 语言）。大约学习了 3 个月，PHP 的常见功能几乎能够运用自如了，与很多初学者一样，感觉 PHP 原来这么简单。

随着 Web 2.0 的到来，那时大兴 Flash 动画之风，我花了 9 个月时间使用 PHP+Flash 技术重写了木棉网（即前木棉休闲站），提供了在线制作大头贴的功能，并且允许将大头贴发布到论坛及 QQ 空间上，从而有效地积累了一批用户。也正因为这些经历，让我意识到了设计模式的重要性。

2006 年，为了给木棉网添加一个讨论区，我花了 3000 元钱购买了一套基于 ASP.NET 的论坛程序。论坛是搭建起来了，但是因为 PHP 与 C# 之间的差异性，论坛并没有很好地为用户提供服务（例如账号不同步，帖子内容不能与文章内容交互等）。于是那年 5 月，我决定推倒重来，苦心学习 C#，而这样做的目的仅仅是为了让系统账号同步。

由于专注于学习，并没有多余的精力管理网站，所以很快网站人气就下来了，直到 2007 年，遗憾地关闭了网站。在学习 ASP.NET 编程的这段日子里，我重新审视了 PHP 开发模式，并且将 ASP.NET 中广泛应用的类工厂设计模式引入到了 PHP 开发中。结合 Smarty 模板引擎，能够很好地实现分层设计思路，这成为我后来开发 PHP 产品的主要设计模式。

2009 年暑假，我当时参与设计的产品是基于 Windows CE 的图书管理系统，终端界面使用 C# 技术，而后台就是使用熟悉的 LAMP 组合。其中开发模式使用 Zend Framework 编程框架，我主要的工作就是编写 SOAP 服务，当时首选的技术方案是 ASP.NET，因为在 ASP.NET 中，创建 SOAP 服务是最简单及快速的方式，但考虑到成本及后期扩展等原因，最

PHP MVC 开发实战

终放弃了该方案，而是使用了 Zend Framework 的 SOAP 功能组件。也就从那时起，我就对 PHP MVC 开发模式产生了浓厚的兴趣，我发现 PHP MVC 模式与之前经常采用的类工厂模式非常类似，但 MVC 分层更加彻底及明确，而且 MVC 框架都是单一入口的，所以无论是在调试还是后期维护方面，都为程序员提供了极高的灵活性。为此我试图寻找一款更加适合在中文环境下使用的 MVC 框架（Zend Framework 1.x 对中文支持不好），所以认识了国内比较早的 FCS（即 ThinkPHP 早期版本）。

一转眼就过去几年了，相信阅读本书的读者与我一样，都是出于对编程的热爱才自学 PHP 的。每个人从自学到所谓的高手，都会经历着一翻波折，有些是必要的，有些则是完全没必要的。为此我将个人的编程经历写出来与读者分享，希望能够帮助读者从中找到适合自己的编程道路。总而言之，编程是枯燥的，但也是快乐的，当你在编程中把原本复杂的问题一一解开，喜悦的心情不用我多言，相信读者也能够感受得到。

目前我主要从事移动开发及云技术开发，参与开发的商业产品有“丽物收购站管理系统”、“海晨酒店管理系统”，这些产品都是终端采用 C#，后台使用 LAMP 组合。此外还开发了一个 Windows Phone 休闲游戏。虽然工作辛苦，但能够看着自己的代码服务于社会，这是莫大的鼓励。

本书的写作背景

当前市面上很多 PHP 图书，都是以介绍 PHP 基础为重点的，例如 PHP 语法、函数等。我身边的一些朋友，在阅读 PHP 图书时，都会发出“书怎么那么厚？”的疑问。这直接给读者 PHP 很高深、很难学的印象。事实上，PHP 是一门简单易懂的语言，它的函数命名方式非常形象化，只需要具备一点英文基础，就能够读懂其含义。借助于主流的 IDE 工具，就算没有计算机编程基础的读者也能够迅速上手，所以说到底 PHP 没有那么复杂。

当然，并不是说 PHP 的基础不用学，事实上任何的计算机编程技术，入门的课程还是需要牢固掌握的，例如语法、结构、运行原理等。但是，由于 PHP 开源及自由的特性，很多开发者，特别是初学者都没有开发模式的概念，导致编写的代码在后期维护时连自己都不认识。造成这一结果除了前面提到的因素外，还由于现在的 PHP 图书极少介绍 PHP 的开发模式。本书内容由始至终，一直围绕着 PHP MVC 展开，利用高效的 MVC 开发模式，彻底告别过去“面条式”的代码编写形式。

由于 PHP 全面开源的特性，所以现在国内外已经出现了很多针对 MVC 编程的 PHP 框架，选择哪一套框架作为开发平台，这对于 PHP 开发人员来讲又是一个极具考验的问题。在 PHP 中，虽然也有官方的 MVC 框架（Zend Framework），但其影响力远没有微软的 ASP.NET MVC 以及 Java 的 SSH（Struts+Spring+Hibernate）那么大。所以在 PHP 中实现 MVC 编程，小项目通常使用现有的开源框架（包括 Zend Framework）实现；而大型项目通常采用自行编写 MVC 框架的方式实现（或者在现在的开源框架上进行二次开发），本书为了兼顾这两种情况，在内容安排上均有涉及，相信读者能够从中找到适合自己的 MVC 框架。

Zend Framework 虽然功能强大，是 PHP MVC 框架中最具代表的框架之一，很多大型的网站都是基于 Zend Framework 构建的，但是由于种种原因，Zend Framework 在国内应用得并不广泛，加上 Zend Framework 的扩展是以组件化的方式提供的，而一些第三方功能组件在国内并不能够正常使用，所以本书并没有以 Zend Framework 作为 PHP MVC 平台，而是使用国内人气比较高的 ThinkPHP 作为 MVC 平台，读者在通过系统的学习后，不仅可以对现

有的开源 MVC 框架轻松上手，而且还能动手编写一个自己的 MVC 框架。

ThinkPHP 是国内比较早也是比较成熟的 PHP MVC 框架，在实现 MVC 编程方面，借鉴了大量先进的开源实现库思路，例如 Struts、JSP Tag、Smarty 等。其清晰的文件结构、高效及敏捷的编程方式，能够让开发者迅速上手。同时基于其灵活的扩展机制，使得项目的功能得到无限扩展。本书将重点对扩展的实现机制进行介绍，例如驱动扩展、类库扩展等，这些扩展的实现思路不仅适用于 ThinkPHP，同样适用于其他 PHP MVC 框架。

本书适合读者群

- 各类型的 PHP 程序员。
- 系统架构师。
- 项目架构师。
- 开源技术爱好者。
- 广大的 Web 开发从业人员。
- 计算机专业的学生。

本书主要内容

本书共分为 3 大部分，分别为基础篇、实战篇、项目篇。在内容组织上尽量以循序渐进的方式深入地讲解每个知识要领。初级的 PHP 程序员在阅读本书内容时，由于引用示例形象不会感觉生搬硬套、敷衍应付；高级的 PHP 程序员在阅读本书内容时，也会感受到作者清晰的实现思路，从中获益。本书的内容组织如下。

第 1 章主要介绍了 PHP 与 MVC 设计模式的关系，并且介绍了在实验环境下及生产环境下的 PHP MVC 运行环境搭建过程。由于 MVC 设计模式的本质是解决团队开发所带来的分工问题，所以本章后面还加入了团队开发的环境配置，帮助每位读者建立起良好的团队协作的概念。

第 2、3 章主要介绍了 PHP 的类结构。由于后面我们还会重点介绍 MVC 扩展的实现，以及编写自己的 MVC 框架，而要学习这些课程，深入掌握 PHP 的面向对象编程是必要的。所以这里安排了 2 章针对 PHP 类结构的内容，帮助读者加深对类的认识。

第 4 章主要介绍了在 PHP 中主流的 MVC 框架发展状况，并通过简单的示例分别演示了这些框架的简单使用，帮助读者初步认识 PHP MVC 编程状况。

第 5~9 章主要介绍 ThinkPHP MVC 框架的使用。这 5 章内容中，作者通过简单、形象的示例，充分地讲解了 ThinkPHP 在实现 MVC 编程中的灵活及高效，例如 MVC 项目部署、数据库 CURD 操作、功能类库的使用等。此外，为了增强应用的稳定性，还介绍了第三方高效的开源库，例如 Nginx 文件上传、HttpSqs 消息队列等。

第 10、11 章主要介绍了 ThinkPHP 的扩展机制。作者将会详细介绍 ThinkPHP 所支持的几种类型扩展，例如类库扩展、驱动扩展、模型扩展、视图扩展、行为扩展、缓存扩展等。

第 12 章将重点介绍 PHP 实现 SOAP 服务的过程，使用 SOAP 服务，就能够实现高效的业务整合。

第 13 章主要是为了兼顾传统的 PHP 程序员更好地利用 ThinkPHP 的模型与控制器机制，实现更灵活的 MVC 定制，从而将 Smarty 模板引擎无缝地移植到 ThinkPHP 中。

第 14~16 章是专门为构建高性能网站而撰写的。现在的网站对性能是非常苛刻的，如果你的网站正在被性能所困扰，建议阅读这 3 章。

PHP MVC 开发实战

第 17 章是本书内容的浓缩部分，作者通过一个论坛系统项目，详细回顾了本书前面所介绍的内容精华，帮助读者巩固所学的知识。

第 18 章再次突出“PHP MVC 开发”这个主题，通过本章内容读者将由 MVC 框架的使用者变成 MVC 框架的开发者。作者将通过一个简单的 MVC 框架，帮助读者了解开发一个 MVC 框架所需要的技术及思路，从而有效提高 PHP 开发技能及组织能力。

致谢

由于 PHP 开源的特性，尽管我使用 PHP MVC 开发网站已有多多年，但将庞大的碎片经验整合为一本厚达几百页的书，其中辛酸、个中滋味非三言两语能够道破的。最后能够面世，这里还要感谢一些人。

首先感谢女朋友逸韵的鼓励与支持，无数次的代码调试，烦躁孤寂的夜晚，QQ 的那头总能感受到关爱，这是我勇于克服障碍的原动力。其次还要多谢家里人，是他们的宽容，让我能够以安静的心调试每一行代码。感谢华软吴呈老师耐心的指导，感谢认识我的同学。最后感谢时静等编辑给予我的帮助。

勘误及支持

尽管我已经花了很多时间及精力，对书中的文字、代码、图片等进行了细致核对，但由于能力所限，书中难免会出现纰漏。在此，恳请读者批评指正，我的邮件地址是：kf@86055.com。

为了方便读者更好地学习，我还创建了本书的专属网站，感兴趣的读者可去浏览，网址是：<http://beauty-soft.net/blog/ceiba>。

目 录

前言

基 础 篇

第 1 章 开发前准备	2
1.1 PHP 与 MVC 概述	3
1.2 开发环境搭建	7
1.2.1 在 Windows 下使用一键安装包	8
1.2.2 在 Linux 平台安装 LNMP	11
1.3 开发工具介绍	20
1.3.1 PHP Coder	20
1.3.2 PHP Editor	21
1.3.3 NetBeans IDE	21
1.3.4 Eclipse PDT	22
1.3.5 Zend Studio	24
1.3.6 Adobe Dreamweaver	25
1.3.7 VS.PHP	27
1.4 SVN 版本控制	28
1.4.1 SVN 介绍	28
1.4.2 SVN 的安装	29
1.4.3 TortoiseSVN 的简单使用	32
1.4.4 TortoiseSVN 文件管理	35
1.4.5 SVN 版本库权限配置	38
1.5 小结	40
第 2 章 面向对象基础	41
2.1 面向对象介绍	42
2.2 PHP 面向对象基础	43
2.2.1 class 关键字	43
2.2.2 类中的成员	44
2.2.3 实例化类	46
2.3 类中的方法	47
2.3.1 方法的参数	47
2.3.2 方法的返回值	48
2.3.3 访问修饰符	49

PHP MVC 开发实战

2.3.4	\$this 关键字	50
2.4	类的继承	50
2.4.1	子类继承父类	51
2.4.2	重写父类中的成员属性	52
2.4.3	final 关键字	53
2.5	静态类成员	54
2.5.1	static 关键字	54
2.5.2	访问静态类成员 (self::parent::)	55
2.5.3	类常量	57
2.6	小结	58
第 3 章	类的高级特性	59
3.1	类对象的克隆 (clone)	60
3.2	类接口 (interface)	60
3.2.1	接口的意义	60
3.2.2	定义接口	61
3.2.3	实现接口 (implements)	62
3.3	抽象类与抽象方法	64
3.3.1	理解抽象概念	65
3.3.2	定义抽象类和方法 (abstract)	65
3.3.3	使用抽象类	66
3.3.4	接口与抽象类的区别	67
3.4	类的异常	68
3.4.1	Exception 类	68
3.4.2	使用 try、catch、throw 语句	69
3.5	小结	70

实 战 篇

第 4 章	PHP MVC 发展状况	72
4.1	出身豪门的 Zend Framework	73
4.1.1	Zend Framework 简介	73
4.1.2	安装 Zend Framework	73
4.1.3	使用 Zend Framework 实现 MVC	75
4.1.4	Zend Framework 核心组件	76
4.2	功能强大的 Symfony	80
4.2.1	Symfony 简介	80
4.2.2	获得 Symfony	81
4.2.3	实现一个简单的 MVC	82
4.2.4	Symfony 的配置文件	85
4.3	灵活完善的 CakePHP	85

4.3.1	CakePHP 简介	85
4.3.2	下载安装 CakePHP	86
4.3.3	使用 CakePHP 构建 MVC 编程	86
4.3.4	好用的 CakePHP 视图助手	89
4.4	使用广泛的 CodeIgniter	89
4.4.1	CodeIgniter 简介	89
4.4.2	安装 CodeIgniter	90
4.4.3	使用 CodeIgniter 实现 MVC	91
4.5	高效便捷的 ThinkPHP	94
4.5.1	ThinkPHP 介绍	94
4.5.2	下载安装 ThinkPHP	95
4.5.3	使用 ThinkPHP 实现 MVC	96
4.5.4	高效的 ThinkPHP 视图引擎	98
4.6	小结	100
第 5 章	ThinkPHP 开发入门	101
5.1	大道至简、开发由我	102
5.1.1	入口文件	102
5.1.2	两种创建项目的方式	103
5.1.3	模块分组	104
5.2	ThinkPHP 目录	106
5.2.1	系统目录	106
5.2.2	项目目录	110
5.3	配置文件	111
5.3.1	选项配置	111
5.3.2	静态缓存配置	113
5.3.3	路由配置	115
5.4	小结	115
第 6 章	ThinkPHP 开发 MVC 应用	116
6.1	ThinkPHP 中的 URL	117
6.1.1	URL 模式	117
6.1.2	URL 路由	119
6.1.3	自动生成匹配的 URL	121
6.1.4	实现文件伪静态	122
6.2	模型 (Model)	123
6.2.1	创建模型	123
6.2.2	实例化模型	124
6.2.3	模型初始化 (_initialize())	125
6.3	控制器 (Controller)	126
6.3.1	创建控制器	126

PHP MVC 开发实战

6.3.2	控制器中的动作 (Action)	126
6.3.3	控制器的调用	128
6.3.4	项目空控制器与控制器空动作	129
6.3.5	动作的前后操作	131
6.4	视图 (View)	132
6.4.1	创建和使用视图	132
6.4.2	系统变量与常量	133
6.4.3	在视图中使用函数	134
6.4.4	数据循环	135
6.4.5	条件判断	138
6.4.6	使用外部文件	141
6.4.7	导入标签库	142
6.4.8	使用布局	145
6.5	小结	147
第 7 章	ThinkPHP 的数据库操作	148
7.1	定义数据表模型	149
7.1.1	模型映射	149
7.1.2	自定义模型	149
7.1.3	create 方法	151
7.1.4	模型属性	154
7.2	基础模型 (Model)	158
7.2.1	连贯操作	159
7.2.2	CURD	159
7.2.3	查询语言	163
7.2.4	使用原生的 SQL 语言	166
7.3	关联模型 (RelationModel)	167
7.3.1	关联关系	167
7.3.2	关联定义	168
7.3.3	关联模型的 CURD	169
7.4	高级模型 (AdvModel)	174
7.4.1	定位查询	174
7.4.2	动态查询	175
7.4.3	内容存文本	176
7.5	大数据支持	176
7.5.1	分布式数据库	177
7.5.2	读写分离	179
7.5.3	数据表分区	183
7.6	小结	187
第 8 章	安全与调试	188

8.1	构建稳健及安全的 MVC	189
8.2	ThinkPHP 内置的安全机制	189
8.2.1	表单令牌	189
8.2.2	字段检测	190
8.2.3	数据验证	193
8.2.4	数据验证码	197
8.3	ThinkPHP 安全日志机制	201
8.3.1	记录方式	201
8.3.2	系统日志	202
8.3.3	应用日志	202
8.4	使用消息队列机制	204
8.4.1	HTTPSQS 基础	204
8.4.2	安装 HTTPSQS	205
8.4.3	测试 HTTPSQS	207
8.4.4	在 MVC 中使用 HTTPSQS	209
8.5	ThinkPHP 程序调试机制	212
8.5.1	开启调试功能	212
8.5.2	代码编译概念	214
8.5.3	异常定制	215
8.5.4	性能调试	216
8.6	小结	218
第 9 章	ThinkPHP 功能库	220
9.1	数据处理	221
9.1.1	XML 引擎	221
9.1.2	返回 XML	228
9.1.3	返回 Json	232
9.1.4	使用 Json 及 Jsonp	235
9.2	ThinkPHP 函数库	241
9.2.1	快捷方法	241
9.2.2	基础函数库	242
9.2.3	扩展函数库	245
9.3	ThinkPHP 多语言支持	248
9.3.1	部署多语言	248
9.3.2	实现多语言	249
9.3.3	多语言与客户端	250
9.4	客户端	252
9.4.1	封装的 Session	252
9.4.2	封装的 Cookie	256

PHP MVC 开发实战

9.5 小结	257
第 10 章 网站静态化	259
10.1 Cache 类	260
10.1.1 缓存的方式	260
10.1.2 开启缓存	260
10.1.3 安装 Memcached	261
10.1.4 使用 Memcached	265
10.2 Memcached 实战应用	267
10.2.1 页面局部缓存	267
10.2.2 缓存数据集	267
10.2.3 使用 Memcache 存放 Session	270
10.3 使用 Redis 缓存	274
10.3.1 Redis 的安装	274
10.3.2 安装 Redis 扩展	277
10.3.3 测试 Redis	279
10.3.4 Redis 缓存	279
10.4 静态缓存	282
10.4.1 静态缓存配置	282
10.4.2 静态缓存规则	283
10.5 小结	286
第 11 章 ThinkPHP 扩展	287
11.1 使用扩展	288
11.1.1 扩展的分类	288
11.1.2 模板引擎扩展	291
11.2 网络操作	293
11.2.1 文件上传	293
11.2.2 Nginx 文件上传进度	295
11.2.3 FTP 文件上传	304
11.2.4 下载文件	305
11.2.5 Socket 套接字编程	306
11.2.6 定位当前位置	308
11.2.7 发送电子邮件	309
11.3 数据处理	311
11.3.1 数据加密	311
11.3.2 数据编码转换	313
11.3.3 数据分页	314
11.3.4 日期数据	315
11.3.5 Input 类	317
11.3.6 GD 库绘制图形	318

11.4 行为扩展	320
11.4.1 行为概述	320
11.4.2 内置行为扩展	322
11.4.3 自定义行为扩展	323
11.5 小结	325
第 12 章 SOAP 分布式开发	326
12.1 分布式开发概念	327
12.1.1 SOA 与 SOAP	327
12.1.2 PHP 5 分布式开发	329
12.2 SOAP 消息体	331
12.2.1 WSDL 文件	331
12.2.2 定义根消息体 definitions	335
12.2.3 type 类型	335
12.2.4 portType 端口类型	336
12.2.5 message 消息列表	337
12.2.6 binding 服务绑定描述	337
12.2.7 service 服务描述	338
12.2.8 可视化创建 WSDL	338
12.2.9 使用 nusoap 创建 WSDL	341
12.2.10 ThinkPHP 生成 SOAP 服务	343
12.3 使用 soapUI 测试 WSDL	347
12.3.1 soapUI 简介	347
12.3.2 安装 soapUI	348
12.3.3 创建项目	349
12.3.4 测试项目	350
12.3.5 负载测试	351
12.4 小结	353
第 13 章 整合 Smarty 模板引擎	354
13.1 Smarty 模板引擎介绍	355
13.2 使用 Smarty	356
13.2.1 在 PHP 中使用 Smarty	356
13.2.2 开启 ThinkPHP 模板扩展	358
13.2.3 以扩展的方式使用全功能 Smarty	358
13.3 Smarty 模板函数和标签	360
13.3.1 include (包含文件)	361
13.3.2 capture (暂存数据)	361
13.3.3 include_php (包含 PHP 文件)	362
13.3.4 insert (插入函数)	362
13.3.5 literal (原文本输出)	363

PHP MVC 开发实战

13.3.6 php (执行 PHP 语句块) 363

13.3.7 strip (保留空格和回车符) 364

13.4 Smarty 模板控制语句 364

13.4.1 if、elseif (判断语句) 364

13.4.2 foreach (循环数据) 365

13.4.3 section (遍历数组) 366

13.5 变量调节器 367

13.5.1 capitalize (首字母大写) 367

13.5.2 count_characters (统计字符) 367

13.5.3 count_paragraphs (统计段落) 367

13.5.4 count_sentences (统计句数) 368

13.5.5 count_words (统计单词) 368

13.5.6 date_format (格式化日期) 368

13.5.7 escape (字符转码) 369

13.5.8 replace (字符替换) 369

13.5.9 regex_replace (正则替换) 370

13.5.10 truncate (字符截取) 370

13.6 视图助手 370

13.6.1 html_image (生成图像) 370

13.6.2 html_options (生成表单选择组件) 371

13.6.3 html_radios (生成表单单选组件) 372

13.6.4 html_checkboxes (生成表单复选组件) 373

13.6.5 html_select_date (生成表单日期选择组件) 374

13.6.6 html_select_time (生成表单时间选择组件) 376

13.6.7 html_table (生成表格) 377

13.7 Smarty 对象方法 378

13.7.1 display (显示模板) 378

13.7.2 fetch (获取输出内容) 379

13.7.3 configLoad (加载配置信息) 379

13.7.4 registerPlugin (注册插件) 380

13.8 Smarty 缓存 383

13.8.1 开启缓存 383

13.8.2 局部缓存 384

13.9 小结 386

第 14 章 整合 Coreseek 全文搜索服务 387

14.1 全文索引概述 388

14.2 Coreseek 基础 389

14.2.1 Coreseek 概述 389

14.2.2 在 Windows 下安装 Coreseek 390

14.2.3	在 Linux 下安装 Coreseek	394
14.3	Coreseek 管理工具	397
14.3.1	indexer	397
14.3.2	searchd	400
14.3.3	search	402
14.4	创建索引	403
14.4.1	索引源分类	403
14.4.2	增量索引	407
14.4.3	实时索引	409
14.5	在 MVC 中搜索数据	412
14.5.1	安装 Sphinx 扩展模块	412
14.5.2	使用 PHP 接口	413
14.5.3	在 MVC 中搜索数据	414
14.5.4	实现关键词高亮显示	418
14.6	小结	420
第 15 章	使用 MongoDB	421
15.1	MongoDB 介绍	422
15.2	MongoDB 的安装	424
15.2.1	在 Windows 下安装 MongoDB	424
15.2.2	在 Linux 下安装 MongoDB	426
15.3	MongoDB 的使用	428
15.3.1	理解 MongoDB 的数据结构	428
15.3.2	数据库管理	429
15.3.3	文档数据类型	430
15.3.4	插入数据	432
15.3.5	查询数据	433
15.3.6	更新数据	436
15.3.7	删除数据	436
15.4	条件操作	437
15.4.1	条件判断语句	437
15.4.2	\$all 匹配全部	438
15.4.3	\$exists 检查字段	438
15.4.4	null 空值处理	439
15.4.5	\$ne 比较	439
15.4.6	\$mod 取模运算	440
15.4.7	\$in、\$nin 枚举查询	440
15.4.8	\$or、\$nor 判断查询	441
15.4.9	\$type 映射查询	442
15.4.10	使用正则表达式匹配	442

PHP MVC 开发实战

15.4.11 limit、skip 限制查询 443

15.4.12 count 查询记录条数 444

15.4.13 sort 查询结果排序 444

15.5 性能优化 445

15.5.1 使用索引 445

15.5.2 固定集合 448

15.5.3 GridFS 450

15.5.4 Profile 优化器 450

15.6 在 MVC 中使用 MongoDB 451

15.6.1 安装 PHP 扩展 451

15.6.2 开启 MongoDB 用户验证 453

15.6.3 ThinkPHP 操作 MongoDB 455

15.7 小结 459

第 16 章 Redis 实战 460

16.1 Redis 的使用 461

16.1.1 Redis 概述 461

16.1.2 常用管理命令 462

16.1.3 Redis 用户验证 464

16.2 Redis 数据类型 465

16.2.1 String 类型 465

16.2.2 Hash 类型 467

16.2.3 List 类型 469

16.2.4 Sets 类型 473

16.2.5 Zset 类型 477

16.2.6 使用 phpRedisAdmin 480

16.3 Redis 高级使用 483

16.3.1 数据持久化 483

16.3.2 虚拟内存 484

16.3.3 事务处理 484

16.3.4 主从同步 486

16.4 在 MVC 中使用 Redis 487

16.4.1 在 PHP 中使用 Redis 487

16.4.2 在 MVC 中进行 CURD 操作 491

16.4.3 数据分页 496

16.5 小结 498

项 目 篇

第 17 章 开发论坛系统 500

17.1 开发前准备 501