

PHP核心技术与最佳实践（第2版）



畅销书重磅升级，PHP领域公认标杆之作，为PHP程序员提供进阶指南，高瞻远瞩，系统性地深刻解读PHP开发中的编程思想、最佳实践、核心技术、开发技巧

實戰
第2版



列旭松 陈文 著

PHP Core Technology and Best Practice, Second Edition

PHP核心技术与最佳实践

机械工业出版社
China Machine Press

[PHP核心技术与最佳实践（第2版）_下载链接1](#)

著者:列旭松

出版者:机械工业出版社

出版时间:2018-9-13

装帧:平装

isbn:9787111608417

内容简介

PHP领域公认的标杆之作，被誉为“PHP开发工程师能力进阶的必读书”，致力于为希望成为中高级PHP程序员的读者提供全面、高效的指导。

本书第1版出版于2012年，累计印刷超过10次，6年后全新升级，一方面根据PHP技术最新的发展删除和更新了一些过时的内容，另一方面对原来的部分内容作了深度优化。

全书共13章，逻辑上分为5个部分：

第一部分（1～2章）：从不同的角度阐述了面向对象软件设计思想的核心概念、技术和原则，分析了面向对象的特性、设计模式的理念，指出了如何设计低耦合、高可扩展性的软件，等等。

第二部分（3～6章）：详细讲解了PHP网络编程的原理、方法、技巧和一些重要的操作，PDO、数据库应用优化，数据库设计和MySQL的高级应用，PHP框架的实现原理。

第三部分（第7章）：围绕PHP扩展开发进行了细致而深入的探讨，解析了PHP的底层实现和Zend虚拟机API，并用PHP扩展开发的实例带领读者走进PHP的底层世界，旨在让读者对PHP性能优化、底层原理进行深入的理解。

第四部分（8～10章）：重点讨论了缓存的设计、Memcached的原理与实践、NoSQL数据库Redis源码分析与应用实践等内容。

第五部分（12～13章）详细讲解了PHP代码的调试和测试、Hash算法和数据库的实现。

作者介绍:

目录: 前言

第1章 面向对象思想的核心概念1

1.1 面向对象的“形”与“本”2

1.1.1 对象的“形”2

1.1.2 对象的“本”4

1.1.3 对象与数组6

1.1.4 对象与类6

1.2 魔术方法的应用8

1.2.1 set和get方法8

1.2.2 call和callStatic方法…11

1.2.3 toString方法和debugInfo方法13

1.3 命名空间与自动加载18

1.3.1 理解命名空间18

1.3.2 自动加载21

1.4 继承与多态21

1.4.1 类的组合与继承22

1.4.2 各种语言中的多态26

1.5 面向接口编程30

1.5.1 接口的作用30

1.5.2 对PHP接口的思考32

1.6 反射35

1.6.1 如何使用反射API35

1.6.2 反射有什么作用37

1.7 异常和错误处理39

1.7.1 如何使用异常处理机制39

1.7.2 怎样看PHP的异常45

1.7.3 PHP中的错误级别45

1.7.4 PHP中的错误处理机制47

1.7.5 PHP7对异常机制的改进	49
1.8 本章小结	53
第2章 面向对象的设计原则	54
2.1 面向对象设计的五大原则	55
2.1.1 单一职责原则	55
2.1.2 接口隔离原则	63
2.1.3 开放-封闭原则	66
2.1.4 替换原则	69
2.1.5 依赖倒置原则	71
2.2 一个面向对象留言本的实例	73
2.3 面向对象的思考	77
2.4 本章小结	78
第3章 PHP网络技术及应用	79
3.1 HTTP协议详解	79
3.1.1 HTTP协议与HTTP/2协议…	79
3.1.2 HTTP协议如何工作	81
3.1.3 HTTP应用：模拟灌水机器人	86
3.1.4 垃圾信息防御措施	92
3.2 抓包工具	96
3.2.1 抓包工具分类	96
3.2.2 Fiddler功能与原理	96
3.2.3 安装Fiddler	97
3.2.4 Fiddler基本界面	98
3.2.5 使用Fiddler进行HTTP断点调试	101
3.3 Socket进程通信机制及应用	104
3.3.1 进程通信相关概念	104
3.3.2 Socket演示：实现服务器端与客户端的交互	105
3.3.3 使用Wireshark分析TCP/IP数据包	108
3.3.4 Socket函数原型	114
3.3.5 PHP中的Socket函数	114
3.3.6 Socket交互应用：使用Socket抓取数据	117
3.4 cURL工具及应用	119
3.4.1 建立cURL请求的基本步骤	119
3.4.2 检查cURL错误和获取返回信息	120
3.4.3 在cURL中伪造头信息	122
3.4.4 在cURL中用POST方法发送数据	124
3.4.5 使用cURL实现简单的文件上传服务	125
3.4.6 cURL批处理	129
3.4.7 cURL设置项	130
3.5 简单邮件传输协议SMTP	131
3.5.1 SMTP协议如何工作	132
3.5.2 SMTP协议常用命令	133
3.5.3 SMTP协议应用：使用Socket发送邮件	133
3.6 WebService的前世今生	138
3.6.1 WebService简介	138
3.6.2 Web服务的实现模式	138
3.6.3 简单对象访问协议SOAP…	140
3.6.4 调试工具soapUI	145
3.7 Cookie详解	148
3.7.1 Cookie的基本概念及设置…	148
3.7.2 PHP和JavaScript对Cookie的操作	149
3.7.3 Cookie存储机制及应用	151
3.7.4 Cookie跨域与P3P、CORS协议	153
3.7.5 本地存储localStorage	156

- 3.8 Session详解158
 - 3.8.1 Session的基本概念及设置158
 - 3.8.2 Session的工作原理159
 - 3.8.3 Session入库161
 - 3.8.4 集群环境下的Session处理165
 - 3.8.5 Cookie与Session问答166
- 3.9 本章小结167
- 第4章 RPC与服务治理的概念169
 - 4.1 理解RPC170
 - 4.1.1 RPC的原理171
 - 4.1.2 RPC的组成元素172
 - 4.2 HPROSE的介绍174
 - 4.3 何为服务治理176
 - 4.4 服务发现与服务注册177
 - 4.4.1 Paxos协议178
 - 4.4.2 ZooKeeper的介绍与安装…179
 - 4.4.3 深入了解ZooKeeper180
 - 4.4.4 ZooKeeper的应用183
 - 4.4.5 etcd的介绍185
 - 4.4.6 在PHP中使用etcd参与服务治理187
 - 4.5 高性能网络库193
 - 4.6 微服务概念简介194
 - 4.7 本章小结196
- 第5章 PHP与数据库基础197
 - 5.1 什么是PDO197
 - 5.1.1 PDO预定义类198
 - 5.1.2 如何使用PDO199
 - 5.1.3 PDO参数绑定与预编译200
 - 5.1.4 PDO事务处理205
 - 5.1.5 PDO的效率问题206
 - 5.2 数据库应用优化206
 - 5.2.1 索引与性能分析206
 - 5.2.2 服务器和配置的优化209
 - 5.2.3 MySQL瓶颈及应对措施211
 - 5.3 数据库设计212
 - 5.3.1 范式与反范式212
 - 5.3.2 数据库分区213
 - 5.3.3 分表的应用215
 - 5.4 MySQL的高级应用217
 - 5.4.1 MySQL视图217
 - 5.4.2 MySQL存储过程和事件调度220
 - 5.4.3 用MySQL模拟消息队列223
 - 5.4.4 SQL注入漏洞与防范226
 - 5.4.5 MySQL全文搜索与Elasticsearch231
 - 5.5 数据库锁与事务239
 - 5.5.1 数据库锁239
 - 5.5.2 数据库事务246
 - 5.5.3 分布式事务254
 - 5.6 数据库中间件257
 - 5.6.1 中间件的介绍257
 - 5.6.2 数据库中间件的原理与实现257
 - 5.7 本章小结263
- 第6章 PHP框架实现原理264
 - 6.1 MVC模式264

6.2 PHP框架Lonicera 0.1265	
6.2.1 实现bootstrap	267
6.2.2 路由器层的实现	269
6.2.3 数据模型	274
6.2.4 视图层实现	282
6.2.5 对框架的初步改进	286
6.3 Lonicera 0.2框架增强	290
6.3.1 引入异常机制	290
6.3.2 拦截器和插件的原理与实现	293
6.3.3 Request增强与安全防御	296
6.4 Lonicera框架0.3版实现	297
6.4.1 composer的介绍与安装	297
6.4.2 composer的类加载机制	301
6.4.3 使用composer来重构框架	303
6.4.4 基于composer的Model增强	307
6.4.5 控制反转与依赖注入	309
6.5 本章小结	318
第7章 PHP扩展开发	319
7.1 为什么要开发PHP扩展	319
7.2 搭建PHP扩展框架	319
7.2.1 PHP源代码目录	320
7.2.2 ext_skel工具	320
7.2.3 Windows平台环境配置	320
7.2.4 Linux平台环境配置	323
7.2.5 PHP的生命周期	325
7.3 PHP内核中的变量	327
7.3.1 PHP变量在内核中的存储方式	327
7.3.2 PHP内核变量访问宏	329
7.3.3 引用计数器与写时复制	330
7.4 PHP内核中的HashTable分析…	332
7.4.1 PHP内核HashTable的数据结构	332
7.4.2 HashTable的代码实现	337
7.5 Zend API详解与扩展编写	341
7.5.1 什么是Zend引擎	342
7.5.2 Zend引擎内存管理	342
7.5.3 PHP扩展的架构	342
7.5.4 接收用户传递的参数	350
7.5.5 在PHP扩展中创建变量	357
7.5.6 在PHP扩展中为变量赋值…	360
7.5.7 错误和输出API	371
7.5.8 运行时信息函数	374
7.5.9 调用用户自定义函数	374
7.5.10 PHP配置项	376
7.5.11 创建常量的宏	379
7.5.12 对象创建与操作	380
7.6 唯一ID生成扩展实现	383
7.6.1 snowflake算法	384
7.6.2 自旋锁原理	385
7.6.3 扩展的实现	387
7.7 本章小结	389
第8章 缓存详解	390
8.1 认识缓存	390
8.1.1 为什么使用缓存	390
8.1.2 命中率	391

- 8.1.3 缓存更新策略392
- 8.1.4 缓存最大数据量394
- 8.2 文件缓存394
 - 8.2.1 文件缓存机制394
 - 8.2.2 文件缓存开源产品Secache396
- 8.3 Opcode缓存398
- 8.4 客户端缓存404
 - 8.4.1 客户端缓存规则405
 - 8.4.2 HTTP协议中的缓存使用…405
 - 8.4.3 HTTP缓存实例408
 - 8.4.4 HTML 5中的Application Cache410
- 8.5 Web服务器缓存411
 - 8.5.1 Apache缓存412
 - 8.5.2 Nginx缓存412
- 8.6 本章小结414
- 第9章 Memcached使用与实践415
 - 9.1 为什么要用Memcached415
 - 9.2 Memcached的安装及使用415
 - 9.2.1 安装Memcached服务器416
 - 9.2.2 安装Memcached客户端417
 - 9.2.3 使用memcache扩展访问Memcached服务器418
 - 9.2.4 使用Memcached加速Web应用421
 - 9.3 深入了解Memcached422
 - 9.3.1 Memcached如何支持高并发422
 - 9.3.2 使用Slab分配算法保存数据422
 - 9.3.3 删除过期item426
 - 9.3.4 使用LRU算法淘汰数据427
 - 9.3.5 Memcached多线程模型428
 - 9.4 Memcached分布式布置方案431
 - 9.4.1 普通Hash分布431
 - 9.4.2 一致性Hash分布432
 - 9.4.3 一致性Hash分布算法实例434
 - 9.5 本章小结436
- 第10章 Redis使用与实践437
 - 10.1 Redis的安装及使用437
 - 10.1.1 Redis安装步骤437
 - 10.1.2 修改Redis配置文件438
 - 10.1.3 运行Redis服务器439
 - 10.1.4 key相关命令439
 - 10.1.5 Redis支持的数据类型…441
 - 10.1.6 Redis排序命令详解447
 - 10.2 事务处理449
 - 10.2.1 事务处理原理449
 - 10.2.2 事务处理实现450
 - 10.3 持久化452
 - 10.3.1 内存快照452
 - 10.3.2 日志追加452
 - 10.4 主从同步453
 - 10.4.1 Redis主从同步原理453
 - 10.4.2 Slave端的工作流程454
 - 10.4.3 Master端的工作流程457
 - 10.5 Redis集群实战462
 - 10.5.1 Redis集群搭建462
 - 10.5.2 Redis集群原理467

- 10.6 扩展库phpredis安装及使用470
- 10.7 Redis应用实践470
 - 10.7.1 使用消息队列发布微博…471
 - 10.7.2 Redis替代文件存储Session472
- 10.8 深入了解Redis内核475
 - 10.8.1 内存淘汰475
 - 10.8.2 对象引用计数器477
 - 10.8.3 自动关闭超时连接479
 - 10.8.4 清除过期数据481
- 10.9 本章小结482
- 第11章 高性能网站架构方案483
 - 11.1 如何优化网站响应时间483
 - 11.1.1 吞吐率484
 - 11.1.2 压力测试485
 - 11.1.3 持久连接487
 - 11.2 MySQL响应速度提高方案：HandlerSocket491
 - 11.2.1 HandlerSocket工作原理491
 - 11.2.2 HandlerSocket安装和配置491
 - 11.2.3 PHP-HandlerSocket性能测试493
 - 11.3 MySQL稳定性提高方案：主从复制494
 - 11.3.1 主从复制工作原理494
 - 11.3.2 主从复制配置495
 - 11.3.3 连接主从服务器497
 - 11.4 Web应用加速方案：Varnish…499
 - 11.4.1 传统代理与反向代理499
 - 11.4.2 Varnish安装和配置500
 - 11.4.3 Varnish性能测试501
 - 11.4.4 修改缓存规则503
 - 11.4.5 监控Varnish运行状态…505
 - 11.5 异步计算方案：Gearman506
 - 11.5.1 Gearman工作原理506
 - 11.5.2 安装Gearman和PHP扩展508
 - 11.5.3 使用Gearman异步发送邮件508
 - 11.6 本章小结510
- 第12章 代码调试和测试511
 - 12.1 调试PHP代码511
 - 12.1.1 PHP调试函数511
 - 12.1.2 断点调试与变量跟踪工具Xdebug515
 - 12.1.3 phpdbg的使用介绍518
 - 12.1.4 使用strace来跟踪定位PHP程序520
 - 12.2 前端调试525
 - 12.2.1 Firebug调试API525
 - 12.2.2 使用Firebug调试DOM结构527
 - 12.2.3 使用Firebug进行远程调试528
 - 12.3 日志管理530
 - 12.3.1 PHP日志531
 - 12.3.2 Apache/Nginx服务器日志531
 - 12.3.3 MySQL日志534
 - 12.4 代码性能测试技术537
 - 12.4.1 时间点测试537
 - 12.4.2 文件查看工具WinCache-Grind538
 - 12.4.3 性能测试注意事项540
 - 12.5 单元测试541
 - 12.5.1 单元测试框架PHPUnit的安装541

- 12.5.2 初步使用PHPUnit进行单元测试542
- 12.5.3 PHPUnit中的断言函数…544
- 12.5.4 PHPUnit常用模板方法…548
- 12.5.5 PHPUnit常用注解549
- 12.6 压力测试551
 - 12.6.1 使用JMeter压力测试HTTP551
 - 12.6.2 压力测试MySQL554
 - 12.6.3 JMeter+Badboy组合测试557
- 12.7 本章小结559
- 第13章 Hash算法与数据库实现…560
 - 13.1 Hash函数560
 - 13.2 Hash算法560
 - 13.2.1 直接取余法560
 - 13.2.2 乘积取整法561
 - 13.2.3 经典Hash算法Times33…561
 - 13.3 Hash表561
 - 13.3.1 Hash表结构562
 - 13.3.2 使用PHP实现Hash表…562
 - 13.3.3 Hash表冲突564
 - 13.3.4 拉链法解决冲突564
 - 13.4 一个小型数据库的实现567
 - 13.4.1 pack函数的用法567
 - 13.4.2 unpack函数的用法569
 - 13.4.3 索引文件和数据文件570
 - 13.4.4 数据库接口方法571
 - 13.4.5 源代码解析572
 - 13.4.6 测试代码580
 - 13.5 本章小结581
 - • • • • ([收起](#))

[PHP核心技术与最佳实践（第2版）_下载链接1](#)

标签

PHP

计算机

●2019

php

★自学编程

评论

还可以 基本的都说到了

有讲面向对象的基础，介绍部分php源码和案例代码都有说明，不错的一本phper书

[PHP核心技术与最佳实践（第2版）_下载链接1](#)

书评

[PHP核心技术与最佳实践（第2版）_下载链接1](#)