

## <<Apache模块开发指南>>

### 图书基本信息

书名：<<Apache模块开发指南>>

13位ISBN编号：9787121059681

10位ISBN编号：7121059681

出版时间：2008-3

出版时间：电子工业出版社

作者：丘

页数：558

字数：661000

译者：张立强,徐建明

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<Apache模块开发指南>>

### 内容概要

本书由ApacheDBD框架的主架构师Nick Kew编写，是Apache模块开发的指南。

本书详细介绍了Apache的架构和模块，并对Apache的发展历史、平台和架构、MPM、APR、编程技巧和规范、模块的配置、模块的扩展、模块的调试等方面进行了介绍，包括大量的示例代码，均以C语言进行描述。

本书适用于使用Apache web服务器的软件开发人员阅读使用。

## <<Apache模块开发指南>>

### 作者简介

NICK KEW是Apache应用程序的主要开发者，你可以在他的主页（[apache.webthing.com](http://apache.webthing.com)）中找到这些应用程序。

他是ADache网络服务器内核开发团队和Apache软件基金会的成员，活跃在用户和开发者中间，并为他们提供支持。

他还在相关的会议（如ApacheCon）上教授和展示Apache应用程序，创建并维护着[www.apachetutor.org](http://www.apachetutor.org)，同时在很多主流网站上撰写关于Apache的文章。

## &lt;&lt;Apache模块开发指南&gt;&gt;

## 书籍目录

第1章 使用Apache进行应用程序开发	1.1 Apache Web服务器简史	1.1.1 Apache 1	1.1.2 Apache 2
1.2 Apache软件基金会	1.2.1 功绩组织模式 ( Meritocracy )	1.2.2 角色	1.2.3 哲学理念
1.3 Apache开发流程	1.3.1 Apache代码仓库	1.3.2 开发者论坛	1.3.3 开发人员
参与Apache工作	1.4 Apache和知识产权	1.4.1 Apache许可证	1.4.2 第三方知识产权
1.5 进一步阅读资料	1.5.1 交互式在线论坛	1.5.2 会议	1.5.3 网站
1.6 小结	第2章 Apache平台和架构		
2.1 纵览	2.2 Apache运行的两个阶段	2.2.1 启动阶段	2.2.2 运行阶段
2.2.3 停止阶段	2.3 多处理模块MPM	2.3.1 为什么需要MPM	2.3.2 UNIX类的MPM模块
2.3.3 MPM模块和操作系统	2.4 基本概念和数据结构	2.4.1 request_rec	2.4.2 server_rec
2.4.3 conn_rec	2.4.4 process_rec	2.5 其他的关键API组件	2.6 Apache配置基础
2.7 Apache的请求处理	2.7.1 内容生成	2.7.2 请求处理阶段	2.7.3 处理钩子
2.7.4 数据轴和过滤器	2.7.5 处理的顺序	2.7.6 处理钩子	2.8 小结
第3章 Apache可移植运行时库			
3.1 APR	3.2 APR实用库	3.3 基本的约定	3.3.1 参考手册：API文档和Doxygen
3.3.2 命名空间	3.3.3 声明的宏	3.3.4 apr_status_t和返回值	3.3.5 条件编译
3.4 资源管理：APR池	3.4.1 资源管理的问题	3.4.2 APR池	3.4.3 资源的生命周期
3.4.4 池的局限性	3.5 精选的APR主题	3.5.1 字符串和格式	3.5.2 国际化
3.5.3 时间和日期	3.5.4 数据结构	3.5.5 Bucket和Brigade	3.5.6 文件系统
3.5.7 网络	3.5.8 编码和密码	3.5.9 URI处理	3.5.10 进程和线程
3.5.11 资源池	3.5.12 API扩展	3.6 APR/Apache中的数据库	3.6.1 DMB和apr_dbm模块
3.6.2 SQL数据库和apr_dbd	3.7 小结	第4章 编程技巧和忠告	
4.1 Apache编程约定	4.1.1 代码行	4.1.2 函数	4.1.3 代码块
4.1.4 流控制	4.1.5 声明	4.1.6 注释	4.2 管理模块数据
4.2.1 配置向量	4.2.2 生命周期域	4.3 模块之间的通讯	4.4 线程安全的编程问题
4.5 管理持久数据	4.5.1 线程安全	4.5.2 内存/资源管理	4.6 跨平台编程的问题
4.6.1 示例：创建一个临时文件	4.7 跨MPM编程问题	4.7.1 进程和全局锁	4.7.2 内存共享
4.8 安全编程问题	4.8.1 预防性原则：不要相信任何事情	4.8.2 拒绝服务攻击：限制破坏	4.8.3 采用操作系统来帮助你
4.9 外部的依赖和库	4.9.1 第三方的库	4.9.2 库的最佳实践	4.9.3 使用库构建模块
4.10 使用其他语言编写和编译模块	4.11 小结	第5章 开发内容生成器	
5.1 HelloWorld模块	5.1.1 模块构架	5.1.2 返回值	5.1.3 处理器的字段
5.1.4 完整的模块	5.1.5 使用request_rec对象	5.2 请求、响应和环境	5.2.1 I/O模块
5.2.2 读取表单数据	5.3 默认的处理程序	5.4 小结	第6章 请求处理周期和元数据处理
6.1 HTTP超文本传输协议	6.1.1 HTTP协议	6.1.2 解构HTTP请求	6.2 Apache的请求处理过程
6.2.1 映射至文件系统	6.2.2 内容协商	6.2.3 安全性	6.2.4 操作缓存
6.2.5 私有元数据	6.2.6 记录日志	6.3 转移请求：内部重定向	6.3.1 错误文档
6.3.2 处理格式错误的请求和恶意请求	6.4 采集信息：子请求	6.4.1 示例	6.5 开发模块
6.5.1 选择文档中不同的变量	6.5.2 错误处理和复用性	6.6 小结	第7章 AAA：访问、认证和授权
7.1 安全	7.1.1 认证：安全的层次	7.1.2 登录Web	7.2 AAA的概览
7.3 Apache1.x和2.0中的AAA	7.4 Apache2.1/2.2中的AAA	7.4.1 基于主机的访问控制	7.4.2 认证：check_user_id
7.4.3 密码查询 ( password lookup )	7.4.4 授权	7.5 AAA逻辑	7.5.1 认证和Require指令
7.5.2 拒绝访问	7.5.3 认证方法	7.6 编写AAA模块	7.6.1 一个基本认证提供者
7.6.2 一个授权函数	7.6.3 配置	7.6.4 基本认证提供者和摘要认证提供者	7.7 实现一个定制的登录机制
7.7.1 使用SQL进行会话管理	7.7.2 在没有浏览器认证对话框时进行认证	7.8 小结	第8章 过滤模块
8.1 输入过滤器和输出过滤器	8.2 内容过滤器、协议过滤器和连接过滤器	8.3 剖析过滤器	8.3.1 回调函数
8.3.2 流水线	8.4 过滤器API和对象	8.4.1 输出过滤器	8.4.2 输入过滤器
8.5 过滤器对象	8.6 过滤器输入/输出	8.7 巧妙的Apache 2.2过滤机制	8.7.1 预处理和后处理
8.7.2 mod_filter	8.7.3 过滤器的自配置	8.7.4 协议处理	8.8 示例：通过直接操作Bucket来过滤文本
8.8.1 Bucket函数	8.8.2 过滤器	8.9 复杂解析	8.10 使用现有的解析器进行过滤
8.11 类似stdio的过滤器输入/输出	8.12 输入过滤器和Pull API	8.12.1 模式	8.12.2 阻塞 ( Block )
8.12.3 readbytes	8.12.4 输入过滤器示例	8.13 小结	第9章 模块配置
9.1 配置基			

## &lt;&lt;Apache模块开发指南&gt;&gt;

基础 9.2 配置数据结构 9.3 管理模块配置 9.3.1 模块配置 9.3.2 服务器配置和目录配置 9.4 实现配置指令 9.4.1 配置函数 9.4.2 示例 9.4.3 配置函数中的用户数据 9.4.4 封装配置函数 9.4.5 配置的作用域 9.4.6 配置函数类型 9.5 配置层次结构 9.6 配置函数中的上下文 9.6.1 上下文检查 9.6.2 方法和 9.7 定制配置容器 9.8 可选的配置方法 9.9 小结 第10章 扩展API 10.1 在Apache中实现新的函数 10.1.1 导出函数 10.1.2 可选函数 10.2 钩子与可选钩子 10.2.1 进一步研究钩子程序 10.2.2 执行顺序 10.2.3 可选钩子示例: mod\_authz\_dbd 10.3 提供者API 10.3.1 实现 10.3.2 实现提供者 10.4 以服务方式提供API扩展 10.4.1 例子: mod\_dbd 10.4.2 实现reslist 10.5 跨平台API构建 10.5.1 使用预处理指令 10.5.2 声明模块API 10.6 小结 第11章 Apache数据库框架 11.1 对新框架的需求 11.1.1 Apache 1.x/2.0和Apache 2.2 11.1.2 连接池 11.2 DBD架构 11.3 apr\_dbd API 11.3.1 数据库操作 11.3.2 API函数 11.4 使用ap\_dbd API 11.5 一个示例应用模块: mod\_authn\_dbd 11.6 开发一个新的DBD驱动 11.6.1 apr\_dbd\_internal.h头文件 11.6.2 输出驱动 11.6.3 驱动函数 11.7 小结 第12章 模块调试 12.1 调试日志 12.1.1 错误日志 12.1.2 调试 12.2 在调试器中运行Apache 12.2.1 服务器的启动和调试 12.2.2 调试和多道处理模块MPM 12.2.3 追踪冲突 12.2.4 调试核心Dump 12.3 特殊用途的Hook和模块 12.3.1 标准模块 12.3.2 重大异常模块 12.3.3 处理反常运行的模块 12.4 过滤器调试 12.4.1 mod\_diagnostics 12.5 小结 附录A Apache许可证 附录B 贡献者许可证协议 附录C 超文本传送协议: HTTP/1.1 本备忘录状态 摘要 1 概述 1.1 目的 1.2 要求 1.3 术语 1.4 操作概述 2 符号惯例和一般语法 2.1 扩充BNF 2.2 基本规则 3 协议参数 3.1 HTTP版本 3.2 统一资源标识符 3.3 日期/时间格式 3.4 字符集 3.5 内容编码 3.6 传送编码 3.7 媒体类型 3.8 产品记号 3.9 质量值 3.10 语言标签 3.11 实体标签 3.12 范围单位 4 HTTP消息 4.1 消息类型 4.2 消息报头 4.3 消息主体 4.4 消息长度 4.5 通用报头字段 5 请求 5.1 请求行 (Request-Line) 5.2 请求所标识的资源 5.3 请求报头字段 6 应答 6.1 状态行 6.2 应答报头字段 7 实体 7.1 实体报头字段 7.2 实体主体 8 连接 8.1 持久连接 8.2 消息传送需求 9 方法定义 9.1 安全和幂等的方法 9.2 OPTIONS 9.3 GET 9.4 HEAD 9.5 POST 9.6 PUT 9.7 DELETE 9.8 TRACE 9.9 CONNECT 10 状态码定义 10.1 用于报告的1xx 10.2 成功的2xx 10.3 重定向的3xx 10.4 客户端错误的4xx 10.5 服务器错误的5xx 11 访问认证 12 内容协商 12.1 服务器驱动协商 12.2 代理驱动协商 12.3 透明协商 13 HTTP中的缓存 13.1 HTTP中缓存的基本设计理念 13.2 过期模型 13.3 验证模型 13.4 应答缓存能力 13.5 从缓存构造应答 13.6 缓存协商的应答 13.7 共享和非共享缓存 13.8 错误和不完全应答缓存行为 13.9 GET和HEAD的副作用 13.10 刷新或删除后的无效性 13.11 强制写通过 13.12 缓存替换 13.13 历史列表 14 报头字段定义 14.1 Accept (接受) 14.2 Accept-Charset (接受字符集) 14.3 Accept-Encoding (接受编码) 14.4 Accept-Language (接受语言) 14.5 Accept-Range (接受范围) 14.6 Age (年龄) 14.7 Allow (允许) 14.8 Authorization (授权) 14.9 Cache-control (缓存控制) 14.10 Connection (连接) 14.11 Content-Encoding (内容编码) 14.12 Content-Language (内容语言) 14.13 Content-Length (内容长度) 14.14 Content-Location (内容位置) 14.15 Content-MD5 (内容的MD5值) 14.16 Content-Range (内容范围) 14.17 Content-Type (内容类型) 14.18 Date (日期) 14.19 ETag (实体标签) 14.20 Expect (期望) 14.21 Expire (过期) 14.22 From (来自于) 14.23 Host (主机) 14.24 If-Match (如果匹配) 14.25 If-Modified-Since (如果自从.....被改变) 14.26 If-None-Match (如果没有一个匹配) 14.27 If-Range (如果有范围) 14.28 If-Unmodified-Since (如果自从.....未修改) 14.29 Last-Modified (最后一次修改) 14.30 Location (位置) 14.31 Max-Forwards (最大化向前传送) 14.32 Pragma (特殊指令) 14.33 Proxy-Authenticate (代理认证) 14.34 Proxy-Authorization (代理授权) 14.35 Range (范围) 14.36 Referer (参考者) 14.37 Retry-After (稍后重试) 14.38 Server (服务器) 14.39 TE (传送编码) 14.40 Trailer (跟踪器) 14.41 Transfer-Encoding (传送编码) 14.42 Upgrade (升级) 14.43 User-Agent (

<<Apache模块开发指南>>

用户代理) 14.44 Vary (验证) 14.45 Via (路由) 14.46 Warning (警告) 14.47  
WWW-Authenticate (WWW-认证) 15 安全考虑 15.1 个人信息 15.2 基于文件和路径名称  
的攻击 15.3 DNS欺骗 15.4 Location报头和欺骗 15.5 Content-Disposition问题 15.6 认  
证证书和空闲的客户端 15.7 代理服务器和缓存 16 致谢 17 参考文献 18 作者地址 19 附  
录 19.1 互联网媒体类型message/http和application/http 19.2 互联网媒体类型multipart/byteranges  
19.3 可容忍的应用程序 19.4 HTTP实体和RFC 2045实体之间的区别 19.5 附加特性  
19.6 和以前版本的兼容 20 索引 21 全部版权声明 致谢 索引

章节摘录

第1章 使用Apache进行应用程序开发： 1.2 Apache软件基金会： Apache软件基金会（ASF）为一系列应用范围广泛的开源软件项目提供组织上、法律上和财政上的支持。Apache软件基金会建立了一整套框架来管理知识产权和财政捐献，同时，限制捐献者潜在的、合法的资源泄漏。

通过精诚合作的项目开发流程，Apache项目交付了企业级的、免费获取的软件产品，吸引大量的社区用户使用。

注重实效的Apache许可证（Apache License）使得无论是商业用户还是个人用户都易于部署Apache的产品。

Apache软件基金会的前身是Apache组织（Apache Group）。

Apache软件基金会是一个非盈利的组织，基于成员制，以保证Apache项目在个人志愿者参与之外能够继续存在。

如果个人承诺与开源软件开发项目合作，并不断地参与和贡献Apache软件基金会的项目，那么他（她）就有资格成为Apache软件基金会的成员。

通过Apache基金会现有大部分成员的赞同和任命，个人才能成为软件基金会的成员。

因此，Apache软件基金会为它所直接服务的社区所掌控，成员都为社区内的项目进行协作。

Apache软件基金会的成员按照基金会的规章制度，每隔一段时间选举一个董事会，管理基金会的组织事务，由董事会任命成员监管基金会的日常事务。

社区会以公开获取Apache基金会运作的记录。

&hellip;&hellip;

## <<Apache模块开发指南>>

### 编辑推荐

Apache不仅是世界上最流行的服务器，同时也是一个非常强大的可扩展的开发平台。现在ApacheTutor.org的Nick Kew撰写了《The Apache Modules Book》，本书从头到尾都以丰富的实例来指导每一个想要充分利用Apache的开发者。

Kew首先深入浅出地详细描述了Apache构架和它的API，然后讲述了你需要的所有技术——从需求处理到代码安全。

Perl和PHP之类的高级语言能提高软件开发的效率，而有些开发任务必须依靠基于C的技术来完成，Kew完美地结合了C语言和高级语言的优点。

利用本书中的代码样例，Kew还介绍了在其他书中甚至在其他任何地方也没有记录过的技术。



<<Apache模块开发指南>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>