书中所有示例的源代码
软件的安装程序和升级补丁

JSP网络编程
工具书

新一代

JSP 网络编程 入门与实践

彭超　马丁　编著

255个
应用示例

- **针对性：** 融入作者多年计算机教育经验，针对初学者的特点设计实例，讲解技术
- **全面性：** 通过255个实例，讲解JSP开发技术的方方面面，既可快速入门，也可用作参考手册
- **专业性：** 结合网络编程典型应用，介绍XML、Struts、JSTL、国际化等流行技术

清华大学出版社

新一代

JSP 网络编程 入门与实践

彭 超 马 丁 编著

清华大学出版社

内 容 提 要

本书以初学者为出发点，循序渐进的介绍了 JSP 的相关技术架构及其 Web 应用的开发过程。本书以实例为主线，为读者提供了学习捷径，降低了学习成本。

本书内容包括基础、进阶、应用三部分，共分为 15 章。基础部分从第 1 章到第 4 章，是面向 JSP 初学者的，包括 JSP 概述、JSP 开发环境配置、Java 语法基础和 JSP 语法基础。进阶部分从第 5 章到第 14 章，包括 JDBC 与数据库操作、JSP 文件操作、JSP 与 JavaBean、Java Servlet、JSP 的 XML 文件操作、JSP 的身份验证、JSP 的国际化、JSP 的标签扩展、表达式语言及 JSTL 标签库和 Struts 应用基础。最后的应用部分是一个完整的开发实例——库存管理信息系统的设计和实现。

本书内容紧凑、实例丰富、结构严谨、深入浅出，使JSP初学者能快速上手，也能对具有一定经验的读者提供有益的参考。

图书在版编目（CIP）数据

新一代 JSP 网络编程入门与实践/彭超；马丁编著. —北京：清华大学出版社，2006

ISBN 978-7-302-14276-8

I. 新… II. ①彭… ②马… III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字（2006）第 147782 号

出 版 者：清华大学出版社　　**地　　址**：北京清华大学学研大厦

http://www.tup.com.cn　　**邮　　编**：100084

社总机：010-62770175　　**客户服务**：010-82896445

组稿编辑：科 海

文稿编辑：张 楠

封面设计：林 陶

版式设计：科 海

印 刷 者：北京市耀华印刷有限公司

发 行 者：新华书店总店北京发行所

开　　本：787×1092　1/16　　**印张**：29.5　**字数**：718 千字

版　　次：2007 年 1 月第 1 版　2007 年 1 月第 1 次印刷

书　　号：ISBN 978-7-302-14276-8

印　　数：1～4000

定　　价：48.00 元（1CD）

前　言

关于本书

JSP技术是Java Web技术的基础。它是基于Java Servlet以及Java平台的Web开发技术，具有动态页面与静态页面分离、脱离硬件平台束缚“一次编写，各处运行”等优点。利用这一技术可以建立安全、跨平台的先进动态网站。深刻理解JSP的技术和架构，是深入开发J2EE技术的必经之路。牢固掌握JSP技术，是架构高性能Web应用的基础。

JSP技术已经成为当前的主流Web开发技术之一。从某种程度上说，JSP技术已经相当成熟，且已有大量的资料和经验可以参考。这给初学者提供了丰富的学习资料，但同时纷繁复杂的技术和彼此矛盾的说法也让初学者无所适从，增加了学习的难度和成本。

本书就从初学者的角度入手，逐步展开对JSP技术的系统讨论，以清晰的知识结构和详尽的实用示例使读者快速上手，降低读者的学习成本。

另外本书提供了大量实际项目中具体问题的解决方案，例如JDBC中持久化JavaBean的实现、Web应用的中文化问题、Web应用的登录模块设计等。这些具体实例来源于笔者参与的实际工程项目，也是每个Web应用开发几乎都会遇到的问题。本书希望与读者分享这些经验，具有一定开发基础的读者也可以参考这些解决方案，并将其应用于自己的开发项目中。

JSP技术不是单独的技术，它以整个Java技术体系为后盾，与其他技术结合起来，从而充分发挥JSP的巨大优势。

总体来说，本书以实例为主线，突出实用性，使读者能以较小的学习成本快速上手。同时本书从纷繁复杂的技术中为读者指明最有效的学习捷径，并且提供了大量实际项目中具体问题的解决方案，从而便于读者在项目开发中随时参考。本书一方面面向初级读者的入门提高，另一方面也可以作为中级读者的参考手册，本书有以下特点：

- 本书内容系统全面，缺乏经验的读者可以从中完整掌握相关技术要点。
- 以实例为基础，注重实用特性，使读者快速上手。
- 针对实际工作中的问题，以专题的形式提供解决方案。
- 通俗易懂的文字及由浅入深的讲解风格。

本书的内容

本书内容包括基础、进阶和应用三部分，由浅入深的系统介绍JSP技术的架构和应用。

基础部分是面向JSP初学者的，包括JSP技术概述、JSP开发环境配置、Java语法基础和JSP语法基础。通过本部分的学习，读者能掌握基本的JSP运行原理和开发方法，并且实现简单的JSP程序。

进阶部分是对基础部分的扩展。本部分中涵盖了JSP相关的若干技术和实际问题的解决方案。其中包括JDBC与数据库操作、JSP文件操作、JSP与JavaBean、Java Servlet、JSP的XML文件操作、JSP的身份验证、JSP的国际化、JSP的标签扩展、表达式语言及JSTL标签库和Struts应用基础。在本部分的实例中还提供了若干有趣的代码片断，包括封装数据库操作的持久化JavaBean、简单的连接池实现、发送邮件的JavaBean、绘图的JavaBean、树形标签、权限管理标签等。这些例子为读者提供了参考，希望可以为读者深入理解Java Web技术提供帮助。

本书最后的应用部分是一个完整的Web应用实例，综合使用了本书介绍的若干技术。提供了从需求分析、架构设计到最后的界面设计和编码实现的实例。为读者介绍了Web应用的完整开发流程，以及具体项目中技术的应用。

本书是以实例为主线，因为代码是最好、最精确的语言，读者可以从中了解技术细节的信息。

适合的读者

- Java Web应用开发人员
- 网站维护人员
- 网页制作爱好者
- 大学/大专/中专的学生
- 参加社会培训人员
- 毕业设计的学生

编者

2006年10月

目　录

第1章

JSP概述

随着Internet和电子商务应用的不断发展，动态网页技术日益流行。JSP作为主流的动态网页技术之一，其优势日益显著。本章将从整体上介绍JSP技术，旨在向读者展示JSP的清晰轮廓。第一节逐一阐述JSP的特点、知识体系以及学习建议，第二节探讨了JSP与Servlet的关系以及Servlet的运行机制，第三节详细比较了JSP与当前其他主流的动态网页技术，最后一节结合实例重点分析了JSP的技术构架和运行原理。

1.1 JSP简介

JSP（Java Server Pages）是由Sun Microsystems公司倡导、许多公司参与建立的一种动态网页技术标准。该技术为创建显示动态生成内容的Web页面提供了一个简捷而快速的方法。在目前流行的3P技术中（3P技术分别是：ASP，Active Server Pages；PHP，Personal HomePage；JSP，Java Server Pages），JSP已经逐渐成为Internet上的主流开发工具。JSP是基于Java Servlet以及整个Java体系的Web开发技术，具有动态页面与静态页面分离、能够脱离硬件平台束缚、“一次编写，各处运行”等优点。利用这一技术可以建立安全、跨平台的先进动态网站。自从1998年初，Sun公司发布了第一个公开的JSP规范草稿，JSP技术就不断更新、完善，目前已发展到较为成熟的JSP 2.0版。本书介绍的技术就是基于JSP 2.0规范的。

1.1.1 JSP特点

JSP主要有如下5个方面的特点：

1. 内容的生成和显示相分离

使用JSP技术，Web页面开发人员可以使用HTML或者XML来设计和格式化最终页面，使用JSP标签或者脚本来生成页面上的动态内容。生成内容的逻辑被封装在标签和JavaBean组件中，并且捆绑在脚本中，所有的脚本在服务器端运行。如果核心逻辑被封装在标签和

JavaBean组件中，那么其他人员，如页面设计者或是Web管理人员，就能够编辑和使用JSP页面而不影响内容的生成。

在服务器端，JSP引擎解释JSP标签和脚本，生成所请求的内容，并且将结果以HTML或XML页面的形式发送回浏览器。这有助于作者保护自己的代码，并且保证了任何基于HTML的Web浏览器的完全可用性。

2. 可移植性

JSP的重要特点之一就是它由Java语言构建，是Java应用程序的一种。Java技术最鲜明的特点之一就是工作平台具有独立性。如果学习过Java语言，就一定听说过“Write Once，Run Anywhere”这句名言。与之相同，JSP也不必考虑在Web服务器环境的操作系统相关性。

不管JSP在何种平台中编写，只要服务器中有JSP Container就可以使用原先编写的程序来运行。正是因为它是由Java语言编写的程序，所以以JSP为基础编写的Web应用程序可以在其他Web服务器中运行，无需更改。这是Java系列产品共同的优点。所以，JSP开发者只需编写JSP程序，无需考虑其硬件是怎样构成的、运行体系是怎样的等问题。

在数据库连接上也有相同的优势。JSP与数据库连接时，只需使用由Java提供的JDBC（Java Data Base Connectivity）。JDBC也独立于平台工作，所以不必担心使用JDBC而使平台变更。由于Java的这种特性，使得基于JSP开发的Web应用程序可以很简单地在新开发的软件中重用。

3. 采用标签简化页面开发

Web页面开发人员不一定都是熟悉脚本语言的编程人员。JSP技术封装了许多功能，这些功能是在XML标签中生成动态内容所需要的。标准的JSP标签能够访问和实例化JavaBean组件、设置或者检索组件属性、下载Applet等。

通过开发自定义的标签库，可以扩展JSP技术，用户可以为常用功能创建自己的标签库。Web页面开发人员能够使用这些工具简化页面开发。

4. 使用可重用的组件

绝大多数JSP页面依赖于可重用的跨平台组件（JavaBean 或者 EJB：Enterprise JavaBean）来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，或者使这些组件为更多的用户所使用。基于组件的开发方法加速了总体的开发过程。

5. 完善的存储管理和安全性

由于JSP页面的内置脚本语言是基于Java语言的，而且所有的JSP页面都被编译成为Java Servlet，所以JSP页面就具有Java技术的所有优点，包括完善的存储管理和安全性。

1.1.2 JSP知识体系和学习建议

JSP技术并非单纯的JSP语法和几个JSP页面，而是一种涉及其他多种技术的综合技术，包括HTML、Java、JavaScript、Servlet、JDBC等，它是一个较为庞大的知识体系，各种知

识之间有一定的层次关系，如图1.1所示。

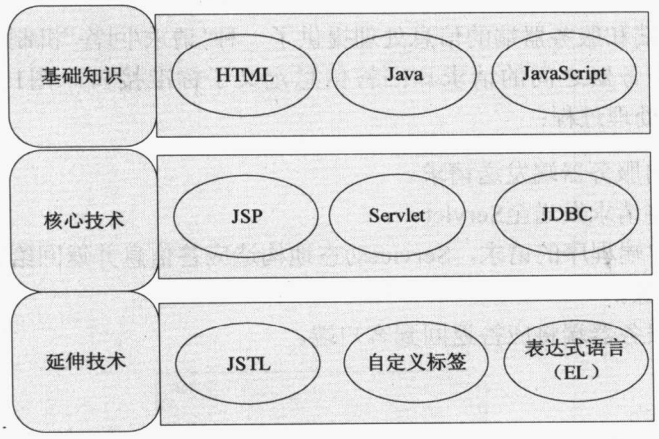

图 1.1　JSP 知识体系

面对如此复杂的知识体系，建议读者首先必须很好地掌握HTML和Java两门技术，对JavaScript有一定的了解，能够读懂一般的JavaScript代码，然后要深入学习Servlet，透彻地理解Servlet运行机制，在此基础上再熟练掌握JSP基本语法和JDBC数据库标准接口的使用。最后，为了进一步提高JSP开发效率，读者还可以学习自定义标签库、JSTL和表达式语言（EL）等较为高级的技术。

通过以上介绍，读者应该对JSP技术有了一个总体性的了解，这对于下一步的深入学习是大有好处的。初学者应该根据自身已有的知识结构，合理地安排学习顺序，可以在很大程度上提高学习效率并增加学习的兴趣和热情。有了一个好的开始，事情便成功了一半。

1.2　JSP与Java Servlet技术

早期的动态Web技术是基于CGI（Common Gateway Interface）的，这种技术使用一些脚本类语言（例如Perl）来获取Web请求，完成后台动作，然后将结果返回给用户。它有两个严重的问题：其一是每个请求都会在系统中开启一个进程来运行脚本，这样在访问负载很重的情况下会消耗极大的系统资源，甚至导致崩溃；其二是脚本类语言与操作系统结合非常紧密，使得Web应用几乎完全跟操作系统和运行环境绑定在一起，移植和重用非常困难。

Servlet技术是Java动态Web技术的基础，它是用Java书写的一种规范，是与平台无关的服务器端构件，可以在支持Servlet的Web服务器或应用服务器上运行。Servlet技术将动态程序编译成字节码，然后在Web容器中运行。由于JSP与Servlet有着极为密切的联系，因此读者必须首先掌握好Servlet。

1.2.1 Servlet处理流程

Servlet为客户端和服务器端的信息处理提供了一种“请求/回答”机制。Java的Servlet API为处理客户端和服务器之间的请求和回答信息定义了标准接口。图1.2展示了客户端到Servlet再到后端的处理过程：

（1）客户端向服务器端发送请求。

（2）服务器将请求发送至Servlet。

（3）依据客户端程序的请求，Servlet动态地构造应答信息并返回给服务器，此时，也可能会利用外部资源。

（4）最后，服务器端将应答返回到客户端。

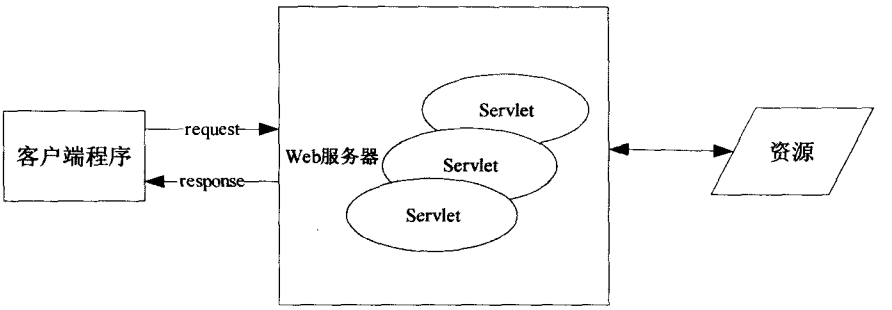

图 1.2 Servlet 处理流程

要理解上述的Servlet处理流程，必须深入探讨Servlet API。Servlet是实现复杂业务应用逻辑的一个强大工具。由于使用Java语言书写，Servlet可以访问整个Java API工具集。Servlet API是一个定义Web客户程序与Web服务器之间标准接口的Java类的集合。客户程序向Web服务器发出请求，Web服务器调用Servlet通过这种接口对请求提供服务。Servlet API是标准的Java API的扩充，它不是Java构架的核心部分，但它是一个有用的可不断扩充的程序包集。Servlet API是由以下两个包组成的：

- javax.servlet。
- javax.servlet.http。

javax.servlet包含有支持与普通协议无关的Servlet类。这意味着Servlet可用于许多协议，例如：HTTP和FTP。Javax.servlet.http包则扩充了基本包的功能，包含了对HTTP的特殊支持。

Servlet接口类是Servlet API的重要抽象。这个类定义了Servlet必须实现的方法，包括处理请求的service方法。GenericServlet类实现了这个接口，并且定义了类属的与协议无关的Servlet。为了编写一个用于Web的HTTP Servlet，必须使用GenericServlet的一个更专门的类——HttpServlet。HttpServlet提供了专门处理HTTP请求的方法，如GET和POST方法。虽然Servlet也可以实现service方法，但在大多数情况下会选择实现HTTP的特殊请求处理方法doGet和doPost。

1.2.2　Servlet生命周期

Servlet的运行机制体现在其生命周期中。基于Servlet应用的客户程序通常并不直接使用Servlet进行通信，而是通过Web服务器和应用服务器完成请求Servlet的装载和初始化、为请求提供服务、卸载和撤销Servlet。这些功能由应用服务器的Servlet管理功能模块提供。一般来说，服务器环境里每一时刻都有一个特定的Servlet对象的实例，这是Servlet持续性的基本原则。当Servlet首次被装载到服务器环境的时候，服务器负责对这个Servlet进行初始化。在这个环境中，Servlet在生命周期内保持活跃或持续的状态。

每个客户程序对Servlet的请求都是由相对于原对象实例的新线程来处理的。服务器负责创建新的线程来处理请求，还负责卸载和重新装入Servlet。这种情况可能出现在Web服务器关闭或Servlet的基础类文件发生变化的时候，具体取决于服务器的底层实现方式。图1.3展示了客户程序与Servlet的基本交互过程：

（1）Servlet 1是初始时由Web服务器装入的，实例变量初始化后，在Servlet的生命周期内仍然是活跃的和持续的。

（2）n个客户程序请求Servlet 1的服务，服务器为每个请求产生一个线程来进行处理。每个线程访问已装入的实例变量，这些变量在Servlet装入时已初始化。

（3）每个线程处理自己对应的请求，并且将回答发送给发出请求的客户程序。

（4）Servlet的生命周期主要表现为Servlet API中Servlet接口的init、service和destroy方法。

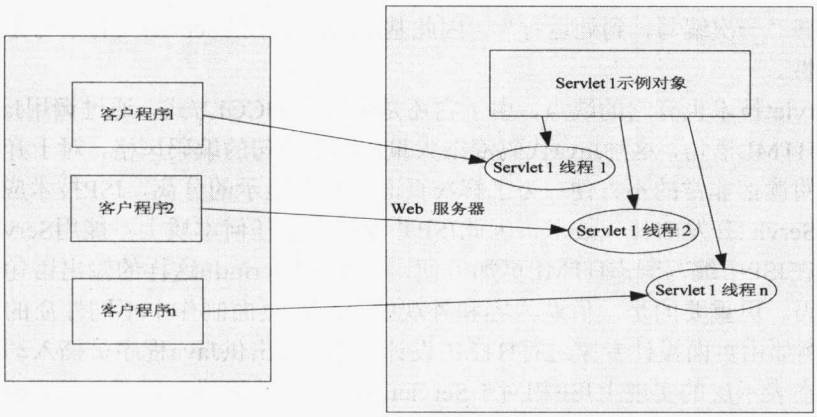

图 1.3　Servlet 与客户程序交互

当Servlet的服务首次被请求时，可以被动态地装载和实例化；当Web服务器启动时，Web服务器要进行部署，也可以使得特定的Servlet被装载和实例化。对于这两种情况，Servlet的init方法都要完成所有必要的初始化工作，并且在任何对Servlet的请求处理之前，保证每个Servlet的实例只被调用一次。

一旦Servlet进行了适当的初始化后，它就能够处理请求了。每个请求由一个ServletRequest对象表示，对应的应答由Servlet API中的ServletResponse对象表示。为了操作HttpServlet，就要操作HttpServletRequest和HttpServletResponse这些对象。HttpServletRequest

对象封装了客户程序请求信息，其中包括客户程序环境和从客户程序向Servlet发送的数据。HttpServletRequest类包含了从请求对象提取信息的方法。HttpServletResponse通常是动态地产生应答。

每当客户程序发出请求时，就会产生一个新的Servlet线程服务于这个请求。以这种方式，服务器可以处理对同一个Servlet发出的多个并发请求。对于每一个请求，通常是调用service、doGet或者doPost方法。这些方法传递HttpServletRequest和HttpServletResponse参数对象。最后，Web服务器将调用destroy方法卸载Servlet，至此，一个Servlet的完整生命周期就结束了。

1.2.3 Servlet总结

以上的分析表明，一个Servlet实质上是一个符合Servlet API规范的Java类。除了Java类一般的优点外，它还具有一些自己的特点，例如安全性、高效性、可扩展性（协议无关性）及与其他资源可交互性等。

Servlet解决了CGI技术的种种弊端。在传统的CGI中，每个请求都要启动一个新的进程，如果CGI程序本身的执行时间较短，启动进程所需要的开销很可能反而超过实际执行时间。而在Servlet中，每个请求由一个轻量级的Java线程处理（而不是重量级的操作系统进程），这样极大提高了运行效率。传统CGI是解释执行的，而Servlet是编译成字节码执行，因此Servlet能更加快速地运行。同时基于Java技术的Servlet对资源管理包括IO、数据库、内存性能有很大优化，例如缓冲以前的计算结果、保持数据库连接的活动等。由于Java是跨平台的语言，能够“一次编写，到处运行”，因此基于Java技术的Servlet具有了与生俱来的优秀的可移植特性。

但是Servlet技术也有它的缺点，由于它还是采用老的CGI方式，通过调用Java的输出函数逐句输出HTML语句，这种Java代码混杂大量HTML语句的编码风格，对于开发人员的编写、修改和阅读都非常的不方便。为了解决页面逻辑和显示的分离，JSP技术应运而生。它是一种简化Servlet开发的替代技术，因此JSP并没有增加任何本质上不能用Servlet实现的功能。但是，在JSP中编写静态HTML更加方便，不必再用println这样的输出语句来输出每一行HTML代码。更重要的是，借助内容和外观的分离，页面制作中不同性质的任务可以方便地分开，例如由页面设计专家进行HTML设计，同时留出供Java程序员插入动态内容的空间。因此，在表示层的实现上JSP相对于Servlet具有更大的优势。但要开发Web应用中很复杂的控制逻辑时，则需要使用Servlet，因为它可以非常清晰方便地封装这些控制逻辑，典型的Web层应用框架Struts就是采用了Servlet来实现控制逻辑的。

1.3 JSP与PHP、ASP/ASP.NET、CGI的比较

目前主流的动态网页技术主要有ASP/ASP.NET、PHP、CGI、JSP等。它们技术上各有优缺点，不能简单地说某种技术优于另一种技术。但JSP以其功能强大、跨平台的特性，在众多技术中独树一帜，下面就将JSP与其他几种主流技术做一个简单的比较。

1.3.1 JSP与PHP的比较

PHP是一种能快速学习、跨平台的HTML内嵌式开发语言。它大量地借用了C、Java和Perl语言的语法，并结合自己的特性，提供了比CGI或者Perl更快速执行的动态网页，与HTML语言具有非常好的兼容性。

PHP还能够支持多种数据库，如Microsoft SQL Server、MySQL、Sybase、Oracle等，它与数据库连接方便、兼容性、扩展性强，并且可以进行面向对象编程。

PHP与Apache可以以静态编译的方式结合起来，而与其他的扩展库也可以同样的方式结合。这种方式最大的好处就是充分利用了CPU和内存，同时极为有效地利用了Apache高性能地的吐能力。由于与数据库的接口也使用了这样的方式，使用的是本地化调用，这也让数据库发挥了最佳性能。

另外，PHP还具有良好的安全性，由于PHP代码开放，所以它的代码在许多工程师手中进行了检测，同时它与Apache编译在一起的方式也可以让它具有灵活的安全设定。PHP的安全性能已经得到了公认。

PHP虽然具有以上优点，但是相对于JSP仍有以下劣势：

1. 缺少企业级的支持

PHP主要应用于一些中小型系统。因为它缺乏多层结构的支持，也缺乏组件的支持，所有的扩充只能依靠PHP开发组所给出的接口。事实上这样的接口还不够多，同时难以将集群、应用服务器这样的特性加入到系统中去，而一个大型的站点或是企业级的应用正需要这样的支持。

2. 没有统一的数据库操作接口

由于PHP的所有扩展接口都是独立团队开发完成的，并在开发时为了形成相应数据的个性化操作，因此针对每种数据库的开发语言和操作接口几乎完全不同。这就使得基于一种数据库的开发工作，在数据库进行升级后几乎需要对全部代码进行修改。而为了让应用支持多种数据库，就需要开发人员将同样的数据库操作使用不同的代码写出多种代码库来，这无疑使程序员的工作量大大增加。同时，PHP缺乏统一的数据库操作接口，这一弊端也使得它不适合应用于电子商务中。

3. 安装较为复杂

由于PHP的每一种扩充模块并不是完全由PHP本身来完成的，需要许多外部的应用库，例如图形需要GD库、LDAP需要LDAP库等。在安装完相应的应用后，再链接进PHP中来，只有在这些环境下才能方便地编译对应的扩展库。这些都是一般开发人员在使用PHP前首先要面对和解决的问题。

4. 缺少正规的商业支持，无法实现商品化应用的开发

由于PHP没有任何编译性的开发工作，所有的开发都是基于脚本技术来完成的，所以所有的源代码都无法编译，完成的应用只能是自己或是内部使用，难以实现商品化。