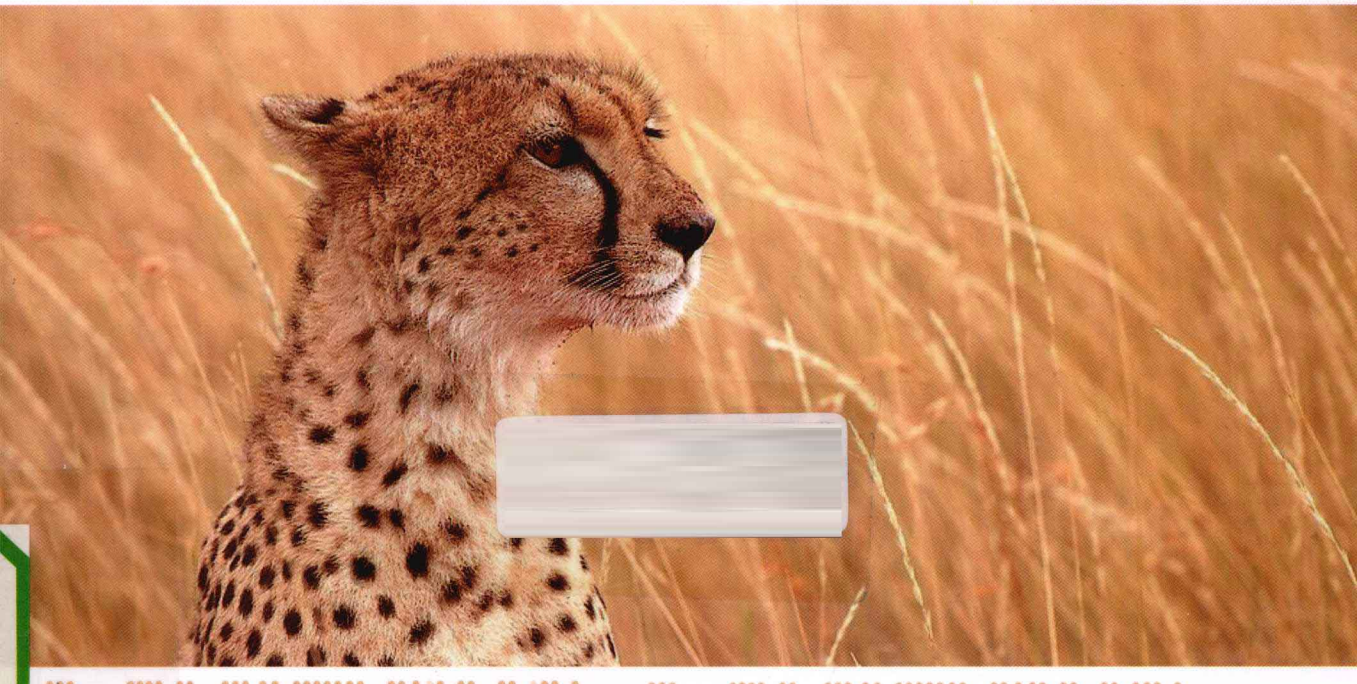
这是一本充满了各种程序代码实现方式的工具书籍
采用案例源码+解说形式全面介绍Apache CXF框架的功能

Broadview
www.broadview.com.cn

基于Apache CXF构建SOA应用

任钢 编著



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

基于Apache CXF构建SOA应用

任钢 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Apache CXF 框架是一个比较有前途的开源 Web Services 框架，也是构建 SOA 架构应用的利器。本书采用案例源码和解说形式全面介绍 Apache CXF 框架的功能。

本书共 15 章，大致分为三个部分。第一部分介绍关于 SOA 和 Web Services 的初步知识，第二部分介绍 Apache CXF 框架的一些基础知识，第三部分重点介绍 Apache CXF 框架的应用，包括 Apache CXF 框架的前端（Frontends）应用、数据绑定（DataBindings）应用、传输协议（Transports）应用，并隆重推出了 Apache CXF 框架如何实现 RESTful 服务、如何支持动态语言和 WS-* 规范等，另外，还包括 Apache CXF 框架一些高级功能的用法。最后，本书还描述了 Apache CXF 的工具、配置、调试、日志、部署和发布等使用的相关内容。

本书最大的特点是实用性。对于 SOA 和 Web Services 的基本概念只是初步介绍，主要内容是基于 Apache CXF 框架的 Web Services 应用案例。对于每一个 Apache CXF 框架的功能主题，都通过一个或多个实际的案例场景来进行阐述。对于每一个案例场景，都有源代码程序例子、架构描绘和程序实现说明。笔者可以负责任地说每一个例子都经过调试并能够运行。实践也是编写本书的一个重要目的，最终目的就是让读者全方位地了解 Apache CXF 框架能实现的功能，一方面让读者理解开发者的思路，另一方面帮助读者在实际工作中应用这些方法和编程。

本书适用于软件设计师、软件开发工程师和一些正在进行 SOA 开发的开发人员，既可以作为 Apache CXF 框架的学习指南，也可以提供给软件开发工程师在设计方面进行参考。

图书在版编目（CIP）数据

基于 Apache CXF 构建 SOA 应用 / 任钢编著. — 北京：电子工业出版社，2013.3
ISBN 978-7-121-19460-3

Ⅰ. ①基… Ⅱ. ①任… Ⅲ. ①互联网络－网络服务器 Ⅳ. ①TP368.5

中国版本图书馆 CIP 数据核字（2013）第 013726 号

策划编辑：张月萍
责任编辑：贾　莉
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×1092　　1/16　　印张：32.75　　字数：890 千字
印　　次：2013 年 3 月第 1 次印刷
印　　数：3000 册　　定价：76.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件到 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

软件架构的实现模式是一个发展的过程。从以前的面向过程、面向对象，到后来的面向构件、面向整合和面向集成，接着又进化到现在的面向服务模式。这时候，一个非常时髦的词——SOA就出现在我们的面前。

SOA是一个沉重的话题。我很早就接触了这个概念。那时既年轻也轻狂，觉得SOA无非就是那么几个已经耳熟能详的单词组合。SOA的确出现得很早，但具体落地非常艰难。仅有一些空洞的解决方案和让人发炫的理想场景。在这样高不可及的光芒下，我们只是空喊一些口号，创造着各种新鲜、时髦和美好的词汇与概念，可没有一个明确可以下手的地方。如何让SOA进入百姓家，这似乎成为了一个不可能完成的任务。

Web Services的出现，似乎给了我们黑的眼睛，让我们有了寻找光明的希望。同时，Java平台对Web Services的支持，也给了我们实现SOA的利器。

基于Java规范的开源Web Services框架，我最早接触的是Apache Axis，当时还编写了基于Axis框架的一个扩展框架。后来与一些公司的开发团队接触，才知道Apache还有一个开源Web Services框架，即Apache CXF。这是一个合并过来的产品。这样，国内开发人员又多了一个开源Web Services框架选择，而且这个开源框架整合了ESB和Web Services，所以我对Apache CXF未来在中国的发展前景还是充满着信心的。对于Java支持SOA，也有很长的时间了，但国内关于SOA的方案都是一些大企业的平台，有点阳春白雪的感觉。对于一些小企业，采用一种重量级的工具似乎有一些不堪重负或者得不偿失。而Apache CXF框架则是一个轻量级的开源Web Services框架，在这个框架上完全可以构筑一个企业级的SOA平台。正是在这个理念的基础上，我认真地学习了Apache CXF框架，并把在这个学习过程中的体会、经验和一些应用例子贡献给读者。

在本书的编写中，主要参考了Apache CXF官方网站的信息。毕竟这是关于Apache CXF最权威的官方信息来源。在参考材料中，我比较喜欢IBM的相关网站，IBM的技术网站一般都站在比较前沿的领域来讲解、讨论和分析问题，所以，书中有一些内容也是摘取或参考了IBM技术网站的思想和内容。

本书覆盖的内容较多，可以毫不夸张地说，书中的很多章节都可以独立地撰写出一本很厚的书籍。正是出于这样的考虑，笔者不能对一些技术做详细描述，有的内容也只是蜻蜓点水地简单说明一下。本书最大的特点是实用性。对于SOA和Web Services的概念，以及其中的方方面面的内容，都有很多相关的书籍来进行阐述。作为本书的基本概念，我也介绍了一些关于这方面的内容。对于每一个例子，都进行了不止一次的编写、调试和测试。我可以负责任地说每一个例子都是可以运行的。对于我每一个涉及的主题，最终都是通过一个实际的案例（包括源程序代码）

进行阐述的。这本书不是一本介绍理论的书，而是充满了各种程序代码实现方式的工具书籍。

当然，阅读本书也要具备一定的基础知识，否则，有些术语和解释还是比较难以理解的。

本书从第一次编写到最后实现出版总共花费了两年半的时间。在这个过程中我耗费了大量的时间和精力。并且这些工作都是在业余时间内完成，白天还要照常上班，只有到了晚上或者节假日才有闲暇时光。我一般难得有闲暇时间，如果有，也是打算去休息或放松，毕竟平时的工作还是比较劳累的。但我还是硬着皮头坐下来写书，写书是一件非常枯燥的任务。对于枯燥，这还不是最大的障碍。我认为最困难的是一个接着一个的技术难题。很多时候有些难题没有办法一下子解决，于是就做了一个例子又一个例子，编写了一个测试接着又一个测试，可还是不能达到自己理想的结果，沮丧、失败、自责、怀疑、困惑等等都涌上头来。在这段时间中我有几次都考虑放弃，心里总是在继续写和终止写之间徘徊不定。但总是觉得已经走了这么长的路程，不能因为一时的挫折而终止多年的辛劳，于是还是像一个孤独的苦行僧执着地编写和测试下去。很高兴的是我还是坚持下来了，在经历了无数个寂寞和孤单的夜晚，把这本书赶写出来。

在这期间，我要感谢我的家人对我的理解和支持。我的妻子和女儿总是抱怨我一天到晚总是待在电脑旁边。我要感谢我的父亲和母亲，我远离家乡，没有时间照顾他们，但他们总是支持我现在做的一切。在这期间，我的母亲永远地离开了我，我只能用这本书来纪念她。我要感谢我的好朋友江愿兵、徐宾和卢建平，他们在很多方面给了我无尽和无私的支持与鼓励。我把这里的一切都献给他们并祝愿他们好人一生平安。

由于笔者水平有限，书中覆盖的范围又比较广，涉及的概念也比较多，所以书中的错误和缺点在所难免，希望读者能给予批评和指正。我的联系方式是：rengang66@sina.com。

目　录

第 1 章 Apache CXF 概述

本章内容：

1. 简介 Apache CXF。
2. Apache CXF 基本特征。
3. Apache CXF 功能特性。
4. Apache CXF 历史。

1.1 Apache CXF 框架简介

Apache CXF 框架是一个开源的 Web Services 框架。它来源于两个著名的开源项目——ObjectWeb Celtix 和 Codehaus XFire。其中，ObjectWeb Celtix 是一款著名的开源 ESB 产品，Codehaus XFire 则是闻名遐迩的 SOAP 堆栈软件。这两个开源项目在 2006 年合并后诞生了 Apache CXF 开源项目。强强联手之后生成的 Apache CXF 自然集两者的长处于一身。Apache CXF 的诞生给开源 SOA 工具领域带来很大的冲击。Apache CXF 是开源社区中继 JAX-WS R1 之后第一个通过 JAX-WS TCK 的 SOAP 堆栈，同时它还支持众多的标准、数据绑定、传输协议。

Apache CXF 提供了对 JAX-WS 规范的全面支持，一方面提供了对多种绑定（Binding）、数据绑定（DataBinding）、传输协议（Transport）以及数据格式（Format）的支持，另一方面，可以根据实际项目的需要，采用代码优先（Code First）或者 WSDL 文档优先（WSDL First）来轻松地实现 Web Services 的发布和使用。Apache CXF 利用前端编程（Frontend API）模式来构建和开发 Services，这与 JAX-WS 类似。这些 Services 可以支持多种协议，如 SOAP、XML/HTTP、RESTful HTTP 或者 CORBA，并且可以在多种传输协议上运行，如 HTTP、JMS 或者 JBI。CXF 大大简化了 Services 的创建，同时它继承了 XFire 的传统，一样可以无缝地与 Spring 进行集成和整合。

选择 Apache CXF 框架的理由很多，主要体现在它良好的特性上。这些特性主要表现在：①简单易用，支持众多编程模型；②与 Spring 紧密集成；③提供了对 RESTful 服务的支持；④可插拔的框架结构设计，可自适应地支持其他多种数据绑定、协议标准和遗留系统；⑤具有轻量级 Web Services 框架；⑥友好的商业 Apache 许可证。

目前，Apache CXF 有很好的用户基础，这主要是因为 Apache CXF 传承了 XFire 和 Spring 的优良“基因”，这两个平台的使用者可继续沿用以前的知识。加上对 JAX-WS、数据绑定、多传输协议、多数据格式和 RESTful 的支持，Apache CXF 框架的未来一直都被看好，关注其进展是非常有意义的。

1.2 Apache CXF 的基本特征

Apache CXF 包含一个范围广泛、功能齐全的功能集合，该功能集合主要集中在以下几个方面：

- **Web Services 标准（WS-*规范）的支持**：Apache CXF 支持多种 Web Services 标准，包括 SOAP（SOAP1.1 和 SOAP1.2）规范、WSI Basic Profile、WSDL、WS-Addressing、WS-Policy、WS-ReliableMessaging、WS-Security、WS-SecurityPolicy 以及 WS-SecureConversation 等。而且现阶段对 Web Services 标准的支持还在继续扩大和完善中。
- **前端模式**：Apache CXF 支持多种前端编程模型。Apache CXF 实现了 JAX-WS 中的 API（TCK 的标准）。它还支持一种“简单前端编程模型”，即可以无注释创造客户和 SEI 终端。Apache CXF 同时支持 WSDL 优先和编码优先两种实现模式。针对 RESTful 规范，Apache CXF 还提供了对 JAX-RS（TCK 的标准）前端的支持。
- **易用性**：Apache CXF 设计明确，编码直观，易于使用。用简单的 API 调用和组合就能快速构建出编码优先的 Web Services，Maven 插件的引入使得集成工具更加容易，对 JAX-WS 的 API 支持、Spring 2.x 的 XML 支持，使配置变得更轻而易举，等等。
- **二进制和遗留协议支持**：Apache CXF 的设计采用了可插拔架构，故可以提供整合任何传输类型的架构容器，不仅支持 XML 类型的容器，也支持非 XML 类型的绑定，如 JSON 和 CORBA 等。

1.3 Apache CXF 的功能特性

Apache CXF 是一个开放源码的 Web 服务框架。该框架提供了一个基于 Web Services 标准并易于使用的编程模型。Web Services 可以使用多种不同的应用协议，诸如针对 HTTP 的应用协议就包含 SOAP、XML 和 JSON、REST 风格等。Apache CXF 框架也支持 JMS（Java 消息服务）的各种传输协议。

1.3.1 支持众多标准

1. 支持 JSR 的相关规范和标准

JAX-WS——Java API for XML-Based Web Services (JAX-WS) 2.0- JSR-224

Web Services Metadata for the Java Platform——JSR-181

JAX-RS——The Java API for RESTful Web Services - JSR-311

SAAJ——SOAP with Attachments API for Java (SAAJ) - JSR-67

2. 支持 WS-*及其相关规范

- 基本规范支持：WS-I Basic Profile 1.1。
- 服务质量规范：Web 服务可靠消息传输（WS-Reliable Messaging，WS-RM）。
- 元数据规范：WS-Policy、WSDL 1.1。
- 通信安全规范：WS-Security、WS-SecurityPolicy、WS-SecureConversation、WS-Trust（部

分支持）。

- 消息支持规范：WS-Addressing、SOAP 1.1、SOAP 1.2、MTOM（Message Transmission Optimization Mechanism）。

1.3.2　支持多种传输协议和协议绑定、数据绑定和数据格式

- 协议绑定：SOAP、REST/HTTP、纯 XML。
- 数据绑定：JAXB 2.x、Aegis、Apache XMLBeans、SDO（Service Data Objects）、JiBX（部分支持）。
- 格式：XML、JSON、FastInfoset。
- 传输协议：HTTP、Servlet、JMS 和 Local（即 JVM 内部消息通信机制），还有其他通过 Apache CXF 的 Camel 传输协议，如 SMTP/POP3、TCP 和 Jabber 等。

Apache CXF 可扩展的 API 允许开发者方便地对绑定和消息格式进行扩展，如 CORBA/IIOP 等格式支持。

1.3.3　灵活部署

- 轻量级容器：Jetty、Tomcat 或基于 Spring 的容器。
- JBI 集成：发布成为 JBI 容器中的服务引擎，这些 JBI 容器有 ServiceMix、OpenESB 或 Petals 等。
- Java EE 集成：可部署在 Java EE 应用服务器中，如 Tomcat、JBoss、Apache Geronimo、JOnAS、RedHat JBoss、OC4J、Oracle WebLogic 和 IBM WebSphere。
- 单独运行的客户机/服务器。

1.3.4　支持多种语言编程

- 完全支持 JAX-WS 2.X 客户机/服务器编程模型。
- JAX-WS 2.X 的同步、异步和单程 API。
- JAX-WS 2.X 动态调用接口（Dll）API。
- JAX-RS 的 RESTful 客户端。
- 支持包装（wrapped）和非包装（non-wrapped）风格。
- XML 消息传递 API。
- 客户端和服务器编程都支持使用 JavaScript 和 ECMAScript 4 XML(E4X)。
- 通过 Yoko 提供对 CORBA 的支持。
- 通过 ServiceMix 来支持 JBI。

1.3.5　支持的工具

- 代码生成工具：WSDL 到 Java，WSDL 到 JavaScript，Java 到 JavaScript。
- WSDL 生成工具：Java 到 WSDL，XSD 到 WSDL，IDL 到 WSDL，WSDL 到 XML。
- 增加服务端点：WSDL 到 IDL。

- 验证文件：WSDL 验证。

1.4 Apache CXF 的历史

Apache CXF 是两个著名的开源项目 ObjectWeb Celtix 和 Codehaus XFire 于 2006 年合并后的产物，Celtix 是一个开放源码并基于 Java 的企业服务总线（ESB）的项目，由 ObjectWeb 财团提供开放源码的中间件解决方案的产品。XFire 是一个基于 Java 的 SOAP 框架，是一个 Codehaus 项目的开放源码。Apache CXF 的名称来自于 Celtix 和 XFire 的组合，因此称为 Apache CXF 的产品。Apache CXF 集 ObjectWeb Celtix 和 Codehaus XFire 的长处于一身，给开源 SOA 工具领域带来不小的冲击。

官方网站：http://cxf.apache.org/index.html。

第 2 章　相关基础知识

本章内容：

1. SOA 基础知识。
2. Web Services 的相关规范。
3. Java 中关于 SOA 的相关规范。

2.1　SOA 基础知识

SOA 基础知识包括 SOA 的定义、基本特征、优点、参考架构、相关技术标准、设计原则等，也包括 SOA 和 Web Services 的关系。

2.1.1　SOA 的定义、基本特征和优点

SOA 有多种定义，这是一个没有正确答案的理论问题。但是其基本特征和优点基本上还是一致。

1. SOA 的定义

一般而言，SOA（Service-Oriented Architecture，面向服务架构）是一种架构模型。SOA 可以根据需求，通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。服务层是 SOA 的基石。服务可以直接被应用调用，这样 SOA 有效控制了系统中与软件代理交互的人为依赖性。关于 SOA 的定义存在着多种说法。下面是一些关于 SOA 比较典型的定义说明。

SOA 的关键是“服务”的概念，W3C 将服务定义为：“服务提供者完成一组工作，为服务使用者交付所需的最终结果。最终结果通常会使使用者的状态发生变化，但也可能使提供者的状态改变，或者双方都产生变化”。

Service-architecture.com 将 SOA 定义为：“本质上是服务的集合。服务间彼此通信，这种通信可能是简单的数据传送，也可能是两个或更多的服务协调进行某些活动。服务间需要某些方法进行连接。所谓服务就是精确定义、封装完善、独立于其他服务所处环境和状态的函数。”Looselycoupled.com 将 SOA 定义为：“按需连接资源的系统。在 SOA 中，资源被作为可通过标准方式访问的独立服务，提供给网络中的其他成员。与传统的系统结构相比，SOA 规定了资源间更为灵活的松散耦合关系。”

Gartner 则将 SOA 描述为：“客户端/服务器的软件设计方法，一项应用由软件服务和软件服务使用者组成……SOA 与大多数通用的客户端/服务器模型的不同之处在于，它着重强调软件组

件的松散耦合，并使用独立的标准接口。”

IBM 定义：面向服务的体系结构是一个组件模型，它将应用程序的不同功能单元称为服务。通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务可以以统一和通用的方式进行交互。

根据以上定义，我们可以初步对面向服务的体系结构有所理解，一般认为：面向服务体系（SOA）结构是一种软件工程方法学，同时也是一种新的软件开发范型。SOA 以松散耦合的粗粒度服务为基本单元，通过开放的、标准的并独立于硬件平台、操作系统和编程语言的接口来发布服务，并通过开放式和规范性的通信协议进行通信。SOA 是一种支持多种服务方式组合、构成和跨平台、并具有一定适应性和扩展性的软件系统。

2. SOA 的基本特征

SOA 具有几个鲜明的基本特征。这些特征表示如下：

- 标准化、规范化、开放性的服务接口；
- 可复用、可重用的服务；
- 以粗粒度为主，同时结合细粒度的服务接口；
- 分级分层访问模式，并可满足安全性要求；
- 松散耦合的组合模式；
- 使用的时间性，随时可用；
- 服务接口设计管理；
- 支持各种消息模式；
- 精确定义的服务契约；
- 通过外部访问来调用 SOA。

3. SOA 的优点

了解了 SOA 的定义和基本特征，下面介绍 SOA 的优点。

（1）灵活性、开放性和可扩展性

组件化和模块化的低层服务可以采用不同组合方式创建高层服务。也就是细粒度的小服务或中服务通过集成、合并和整合来构造粗粒度的大服务，从而实现服务的复用，这些都体现了 SOA 的灵活性和开放性。同时，由于服务消费者通过服务中介来访问服务生产者，这种服务的实现方式本身也体现了 SOA 的可扩展性。

（2）明确开发人员角色

所处理的服务可以区分开发人员角色。服务之间通过服务接口或者服务描述来进行访问和调用。不同的开发人员通过服务契约进行协同，实现服务功能。服务用于解决应用间的互操作问题以及用于组合新应用或应用系统。例如，熟悉 ESB 的开发人员可以集中精力在重用访问层，协调层开发人员则无须特别了解 ESB 的实现，而可将精力放在解决核心业务问题上。

（3）支持多种访问类型

借助对 XML、Web Services 标准的支持和准确的服务接口定义，SOA 可以支持多种访问类型，包括智能手机、智能终端等新型访问渠道。