

- Chapter 1: Getting Familiar with CXF
 - Web service technology standards
 - XML
 - SOAP (Simple Object Access Protocol)
 - WSDL (Web Services Description language)
 - REST (Representational State Transfer)
 - Service Registry
 - Introducing web services
 - Approaches for web service development
 - Web service SOAP communication styles
 - Apache CXF
 - History of CXF
 - Why CXF?
 - Support for web service standards
 - Support for POJO (Plain Old Java Object)
 - Frontend programming APIs
 - Tools support
 - Support for RESTful services
 - Support for different transports and bindings
 - Support for non-XML binding
 - Ease of use
 - Flexible deployment
 - Setting up the environment
 - For ANT users
 - For Maven users
 - Summary
- Chapter 2: Developing a Web Service with CXF
 - The Order Processing Application
 - Developing a service
 - Creating a Service Endpoint Interface (SEI)
 - Developing a service implementation class
 - Spring-based server bean
 - Developing a client
 - Developing a Spring-based client bean
 - Developing web service client code
 - Running the program
 - Building the code
 - Deploying the code
 - Executing the code
 - CXF architecture
 - Bus
 - Frontend
 - JAX-WS
 - Simple frontend
 - Messaging and Interceptors

- Service model
 - Data binding
 - Protocol binding
 - Transports
 - Summary
- Chapter 3: Working with CXF Frontends
 - JAX-WS frontend
 - Code-first development
 - Creating Service Endpoint Interface (SEI)
 - Adding Java annotations
 - Publishing the service
 - Developing a consumer
 - Running the Code-first example
 - Contract-first development
 - Generating service components
 - Implementing the service method
 - Publishing the web service
 - Invoking the web service
 - Using dynamic client
 - Creating a simple dynamic client
 - Running the dynamic client
 - Using the CXF service model for building dynamic client
 - Running the dynamic client which uses Service Model API
 - Provider and Dispatch services
 - Understanding messaging modes
 - Message mode
 - Payload mode
 - Understanding types of message objects
 - `javax.xml.transform.Source`
 - Implementing Provider service
 - Publishing the Provider service
 - Implementing the Dispatch service
 - Running the provider dispatch example
 - Web service context
 - Implementing Context in service
 - Running the web service context example
 - Simple frontend
 - Developing a simple frontend
 - Creating service implementation class and interface
 - Creating server implementation
 - Creating client
 - Running the simple frontend example
 - Summary
- Chapter 4: Learning about Service Transports

- Transport protocols in CXF
 - HTTP transport
 - SOAP over HTTP
 - HTTP only
 - HTTP Conduit
 - HTTP destination
 - HTTPs transport
 - Developing the service and implementation class
 - Generating crypto key
 - Creating client and server bean configuration
 - Configuring the server to support SSL
 - Developing the client component
 - Building and deploying
 - Configuring SSL for Jetty runtime
 - JMS transport
 - Developing the service and implementation class
 - Developing an embedded broker
 - Creating a server and client bean configuration
 - Developing a client component
 - Performing build and deployment
 - Local transport
 - Developing SEI and an implementation class
 - Developing a server
 - Creating client bean configuration
 - Developing a Client
 - Building and executing
 - Summary
- Chapter 5: Implementing Advanced Features
 - Understanding CXF interceptors
 - Understanding interceptor phase and chain
 - Overview of the interceptor API
 - Interceptor interface
 - The PhaseInterceptor interface
 - The AbstractPhaseInterceptor class
 - Developing the custom interceptor
 - Developing the server side interceptor
 - Adding a server side interceptor to the Order Process service
 - Developing the client side interceptor
 - Adding a client side interceptor to the client code
 - Developing the standalone server for publishing the Order Process web service
 - Building and running the Order Process web service and interceptor
 - Building the code
 - Executing the code

- Testing the custom interceptor for negative condition
 - Understanding CXF features
 - Applying the GZIP feature to the Order Process web service
 - Developing service and implementation class
 - Developing a server component
 - Creating the client bean configuration file
 - Creating a client component to consume the service
 - Building and executing the code
 - Understanding CXF Invoker
 - Overview of Invoker API
 - The Invoker interface
 - The AbstractInvoker class
 - Developing custom invoker
 - Summary
- Chapter 6: Developing RESTful Services with CXF
 - Overview of REST and RESTful services
 - Java API for RESTful services
 - CXF JAX-RS implementation
 - Developing end-to-end RESTful services using CXF JAX-RS implementation
 - Developing the RESTful service
 - Creating Java data objects for Request and Response
 - Providing binding for the Request and Response data objects
 - Developing the implementation class
 - Creating the client
 - Running the program
 - Building the code
 - Deploying the code
 - Executing the code
 - Adding exception handling to RESTful service
 - Summary
- Chapter 7: Deploying RESTful Services with CXF
 - Packaging the Book Shop application
 - Configuring CategoryService RESTful bean using Spring
 - Integrating Spring using web.xml
 - Building and deploying the WAR file
 - Building the code
 - Deploying the code
 - Invoking operations on the Book Shop RESTful application
 - Installing POSTER client
 - Invoking the Book Shop application using the the POSTER tool
 - Invoking the Get Category operation
 - Invoking the Add Category operation

- Invoking the Update Category operation
 - Invoking the Add Books operation
 - Invoking the Get Books operation
 - Invoking the Update Category operation with invalid request
 - Invoking the Get Category operation with invalid request
 - Invoking the Delete Category operation
 - Invoking the Book Shop application using CXF APIs
 - Configuring JSON support for the Book Shop application
 - Incorporating JSON message format for the Book Shop application
 - Invoking the Get Category operation with JSON as the message format
 - Invoking the Add Category operation with JSON as the message format
 - Invoking the Book Shop application with JSON as the message format using CXF APIs
 - Intercepting messages for the Book Shop application
 - Deploying the Book Shop application in the application servers
 - Summary
- Chapter 8: Working with CXF Tools
 - Invoking a web service using the Java client
 - Analyzing the service WSDL definition
 - Building and running the Java web service clients
 - Generate the web service clients
 - Analyzing the JAX-WS and client generated artifacts
 - Modifying the generated client
 - Building the client
 - Running the client
 - Invoking the web service using JavaScript
 - Building and running the Java web service clients
 - Generating the JavaScript client
 - Analyzing the generated artifacts
 - Creating the client
 - Running the client
 - Creating Service Implementation from the WSDL file
 - Generating and deploying the Service Implementation from the WSDL file
 - Generating the web Service Implementation
 - Analyzing the Service Implementation generated artifacts
 - Modifying the generated Service Implementation
 - Building the web service project
 - Deploying and publishing the web service
 - Invoking the web service
 - Validating WSDL files
 - Summary

- Appendix A: Getting Ready with Code Examples
 - Downloading the source code
 - Downloading the software required for the book
 - Setting up the environment
 - Using Maven for Build management
 - Building chapter source code using Maven

- Appendix B: Getting Started with Spring
 - Concept of POJO-based development
 - Understanding Inversion of Control
 - Overview of aspect-oriented programming
 - Introduction to Spring framework
 - The Spring IoC container
 - Creating a Spring IoC application
 - Creating the entity model
 - Creating services
 - Creating the application and wiring POJO
 - Creating the standalone client
 - Running the program
 - Building the code
 - Executing the code
 - Summary