

WEB SERVICES

Revised 11/10/2011

/training/etc

The Art of Knowledge.

This Page Intentionally Left Blank

Table of Contents

Web Services Using Java.....	1
Web Services using WebLogic.....	2
Developing Web Services on WebSphere®.....	3
Developing RESTful Services in Java v1.1.....	4

This Page Intentionally Left Blank

Course Description: This course is a comprehensive look at the state of the art in developing interoperable web services on the Java EE platform. Students learn the key standards -- SOAP, WSDL, and the WS-I Basic Profile -- and the Java architecture that has evolved to build interoperable services and clients. JAX-WS is central to the course, and we cover both WSDL-driven and Java-driven development paths, as well as message handlers and attachment support. With the new Provider and Dispatch APIs, it's now much easier to integrate SAAJ, JAXB, and JAXP code into services and clients, and we explore these strategies in depth as well.

Who Should Attend: Java developers who want to build Web services or clients, either from scratch or based on existing J2EE systems will benefit from this course.

Prerequisites: Students must be able to read XML documents and to write well-formed XML by hand. Knowledge of XML Schema will be helpful, too, but is not a strict prerequisite. Experience with other Java EE standards, especially servlets and JSP, will be very helpful in class, but is not strictly required.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Describe the interoperable web services architecture, including the roles of SOAP and WSDL.
- Understand the importance of the WS-I Basic Profile for interoperable web services.
- Build JAX-WS services and clients that take full advantage of the automated data binding of JAXB.
- Use lower-level SOAP and XML APIs for services and/or clients.
- Customize data binding by specifying specific type mappings or altering method or parameter names.
- Expose session beans as web services.
- Incorporate binary data, such as images, into service and client code.

Course Outline:

Chapter 1. Overview of Web Services

Why Web Services?
Service-Oriented Architecture
HTTP and XML
Simple Object Access Protocol (SOAP)
Web Service Description Language (WSDL)
Universal Description, Discovery and Integration (UDDI)
The WS-I Basic and Related Profiles
REST

Chapter 2. Web Services for Java EE

Hosting Web Services: Scenarios
Invoking Web Services: Scenarios
Web Services for Java EE (WS4JEE)
The Automated Approach: JAX-WS and JAXB
Manual Options: SAAJ and JAXP
Portable Web-Services Metadata
Service Registries: JAXR

Chapter 3. The Simple Object Access Protocol

Messaging Model
Namespaces
SOAP over HTTP
The SOAP Envelope
The Message Header
The Message Body
SOAP Faults
Attachments

Chapter 4. The Java API for XML Binding

The Need for Data Binding
XML Schema
Two Paths
JAXB Compilation
Mapping Schema Types to Java
Java-to-XML Mapping Using Annotations
Marshaling and Unmarshaling
Working with JAXB Object Models
In-Memory Validation

Chapter 5. Web Services Description Language

Web Services as Component-Based Software
The Need for an IDL
Web Services Description Language
WSDL Information Model
The Abstract Model -- Service Semantics
Message Description
Messaging Styles
The Concrete Model -- Ports, Services, Locations

Extending WSDL -- Bindings
Service Description

Chapter 6. The Java API for XML-Based Web Services

Two Paths
How It Works: Build Time and Runtime
The Service Endpoint Interface
Working from WSDL
Working from Java
RPC and Document Styles
One-Way Messaging
Binary Protocols

Chapter 7. WSDL-to-Java Development

The @WebService Annotation
Generated Code
Compilation and Assembly
Deployment
Runtime Behavior
Scope of Code Generation
More JAXB: Mapping Collections
More JAXB: Mapping Enumerations

Chapter 8. Client-Side Development

Stubs and Proxies
Generated Code
Locating a Service
Invoking a Service

Chapter 9. Java-to-WSDL Development

The @WebMethod, @XmlParam, and Related Annotations
Scope of Code Generation
More JAXB: Mapping Inheritance
Controlling the XML Model
Controlling the WSDL Description

Chapter 10. JAX-WS Best Practices

Which Way to Go?
Interoperability Impact
Portability Impact
Polymorphism in Web Services
Web Services as Java EE Components
Lifecycle Annotations
Context Interfaces
The @WebServiceRef Annotation

Chapter 11. Provider and Dispatch APIs

Stepping Down
The Provider<T> Interface
Implementing a Provider

JAXB Without WSDL
Integrating JAXP
The Dispatch<T> Interface
Building Clients

Chapter 12. The SOAP with Attachments API for Java

The SAAJ Object Model
Parsing a SOAP Message
Reading Message Content
Working with Namespaces
Creating a Message
Setting Message Content

Chapter 13. Message Handlers

Handling SOAP Headers
Servlet Endpoint Context
MessageContext and SOAPMessageContext
Message Handlers and Handler Chains
Processing Model and Patterns
Client-Side Handlers

Chapter 14. EJBs as Web Services

Enterprise JavaBeans
Three Tiers for Java EE
EJB3 and JAX-WS
Session Beans as Web Service Endpoints
The Bean's Service Endpoint Interface
SOAP as an EJB Protocol
Pitfalls

Chapter 15. Handling Binary Content

The WS-I Attachments Profile
Using base64Binary
MIME Attachments
JAX-WS Support
MTOM and XOP
SAAJ Support

Appendix A. Learning Resources

Appendix B. Compatibility and Migration

JAX-RPC
Comparing JAX-RPC and JAX-WS
Using JAX-RPC and JAX-WS Together
SOAP "Section 5" Encoding

Course Description: This course prepares Java programmers to develop Web services and clients using the BEA WebLogic Platform(TM), in accordance with prevailing standards such as SOAP, WSDL, and JAX-RPC. Students get an overview of the interoperable and Java-specific Web services architectures and then learn the standard (J2EE 1.4) APIs for SOAP messaging and WSDL-driven, component-based service development.

Who Should Attend: Java developers interested in building Web services using BEA WebLogic should attend this course.

Prerequisites: Experience in Java Programming, including object-oriented Java and the Java streams model, is essential. Knowledge of XML and XML Schema will be helpful, but is not required.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Describe the interoperable Web services architecture (SOAP, WSDL, and UDDI).
- Describe the Java Web services architecture (JAXP, SAAJ, JAXM, JAX-RPC, JAXR).
- Read and write SOAP messages.
- Use SAAJ to manage SOAP message content as Java objects.
- Write WSDL descriptors: types, messages, interfaces and services.
- Map types between WSDL/XML Schema and Java using JAX-RPC.
- Build Web services from WSDL descriptors or based on existing Servlet, JSP, or EJB applications.
- Implement a JAX-RPC message handler chain to adapt an existing Web service.
- Create, send, receive, and read SOAP attachments using SAAJ.
- Build asynchronous Web services using JMS and JAX-RPC.
- Secure an existing Web service by requiring authentication and authorization at the service URI.
- Implement message security using JAX-RPC message handlers.

Course Outline:

The Web Services Architecture

WebLogic and Web Services

The Java Web Services Architecture

SOAP

SAAJ

SAAJ Web Services

WSDL

JAX-RPC

Generating Web Services From Java

Generating Java Web Services From WSDL

Web Services and EJB

Message Context And Message Handlers

Soap Attachments

Web Services and JMS

Security

Appendix A: Learning Resources

Appendix B: Quick Reference

Course Description: A comprehensive look at the state of the art in developing interoperable web services on the J2EE platform using IBM® WebSphere® Application Server. Students learn the key standards -- SOAP, WSDL, and the WS-I Basic Profile -- and the Java architecture that has evolved to build interoperable services and clients. JAX-WS is central to the course, and we cover both WSDL-driven and Java-driven development paths, as well as message handlers and attachment support. With the new Provider and Dispatch APIs, it's now much easier to integrate SAAJ, JAXB, and JAXP code into services and clients, and we explore these strategies in depth as well.

Who Should Attend: This course is for developers interested in building Web services using WebSphere.

Prerequisites: Students must have strong Java programming skills. Students must also be able to read XML documents and to write well-formed XML by hand. Knowledge of XML Schema will be helpful, too, but is not a strict prerequisite. Experience with other Java EE standards, especially servlets and JSP, will be very helpful in class, but is not strictly required.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Describe the interoperable web services architecture, including the roles of SOAP and WSDL.
- Understand the importance of the WS-I Basic Profile for interoperable web services.
- Build JAX-WS services and clients that take full advantage of the automated data binding of JAXB.
- Use lower-level SOAP and XML APIs for services and/or clients.
- Customize data binding by specifying specific type mappings or altering method or parameter names.

Course Outline:

Chapter 1. Overview of Web Services

Why Web Services?
Service-Oriented Architecture
HTTP and XML
Simple Object Access Protocol (SOAP)
Web Service Description Language (WSDL)
Universal Description, Discovery and Integration (UDDI)
The WS-I Basic and Related Profiles
REST

Chapter 2. Web Services for Java EE

Hosting Web Services: Scenarios
Invoking Web Services: Scenarios
Web Services for Java EE (WS4JEE)
The Automated Approach: JAX-WS and JAXB
Manual Options: SAAJ and JAXP
Portable Web-Services Metadata
Service Registries: JAXR

Chapter 3. The Simple Object Access Protocol

Messaging Model
Namespaces
SOAP over HTTP
The SOAP Envelope
The Message Header
The Message Body
SOAP Faults
Attachments

Chapter 4. The Java API for XML Binding

The Need for Data Binding
XML Schema
Two Paths
JAXB Compilation
Mapping Schema Types to Java
Java-to-XML Mapping Using Annotations
Marshaling and Unmarshaling
Working with JAXB Object Models
In-Memory Validation

Chapter 5. Web Services Description Language

Web Services as Component-Based Software
The Need for an IDL
Web Services Description Language
WSDL Information Model
The Abstract Model -- Service Semantics
Message Description
Messaging Styles
The Concrete Model -- Ports, Services, Locations

Extending WSDL -- Bindings
Service Description

Chapter 6. The Java API for XML-Based Web Services

Two Paths
How It Works: Build Time and Runtime
The Service Endpoint Interface
Working from WSDL
Working from Java
RPC and Document Styles
One-Way Messaging
Binary Protocols

Chapter 7. WSDL-to-Java Development

The @WebService Annotation
Generated Code
Compilation and Assembly
Deployment
Runtime Behavior
Scope of Code Generation
More JAXB: Mapping Collections
More JAXB: Mapping Enumerations

Chapter 8. Client-Side Development

Stubs and Proxies
Generated Code
Locating a Service
Invoking a Service

Chapter 9. Java-to-WSDL Development

The @WebMethod, @XmlParam, and Related Annotations
Scope of Code Generation
More JAXB: Mapping Inheritance
Controlling the XML Model
Controlling the WSDL Description

Chapter 10. JAX-WS Best Practices

Which Way to Go?
Interoperability Impact
Portability Impact
Polymorphism in Web Services
Web Services as Java EE Components
Lifecycle Annotations
Context Interfaces

Chapter 11. Provider and Dispatch APIs

Stepping Down
The Provider<T> Interface
Implementing a Provider
JAXB Without WSDL
Integrating JAXP

The Dispatch<T> Interface
Building Clients

Chapter 12. The SOAP with Attachments API for Java

The SAAJ Object Model
Parsing a SOAP Message
Reading Message Content
Working with Namespaces
Creating a Message
Setting Message Content

Chapter 13. Message Handlers

Handling SOAP Headers
Servlet Endpoint Context
MessageContext and SOAPMessageContext
Message Handlers and Handler Chains
Processing Model and Patterns
Client-Side Handlers

Appendix A. Learning Resources

Appendix B. Compatibility and Migration

JAX-RPC
Comparing JAX-RPC and JAX-WS
Using JAX-RPC and JAX-WS Together
SOAP "Section 5" Encoding

Course Description: This course shows experienced Java programmers how to build RESTful web services using the Java API for RESTful Web Services, or JAX-RS. We begin with an overview of web services development in the Java EE platform, including SOAP, WSDL, REST; JAX-WS, JAXB, and JAX-RS. Then students learn to work with JAXB to bind Java object models to XML Schema. The bulk of the course is devoted to study of JAX-RS: students work with the Jersey implementation of JAX-RS to create RESTful services from simple single-value interactions to more sophisticated services that manage CRUD (create/retrieve/update/delete) operations on more complex data types, using JAXB to marshal and unmarshal data over the wire.

Who Should Attend: This course is for experienced Java programmers who want to learn how to build RESTful web services using the Java API for RESTful Web Services or JAX-RS.

Prerequisites: Strong Java programming skills are essential. Students must be able to read XML documents and to write well-formed XML by hand. Experience with other Java EE standards, especially servlets and JSP, will be very helpful in class, but is not strictly required.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Understand the advantages of the REST architecture for web services.
- Use JAX-RS to develop simple RESTful services.
- Control dispatching to service methods based on URL patterns and HTTP methods.
- Bind request values to method parameters when expressed as HTTP query parameters, form values, headers, cookies, and more.
- Manage XML content using XML Schema and JAXB.
- Incorporate XML entities into service input and output.
- Take advantage of lifecycle and context services available to JAX-RS services as Java EE components.

Course Outline:

Overview of Web Services

Why Web Services?
Service-Oriented Architecture
HTTP and XML
SOAP
WSDL
REST
UDDI
The WS-I Basic Profile
Security

Web Services for Java EE

Hosting Web Services: Scenarios
Web Services for Java EE
JAX-WS and JAXB
Web-Services Metadata
WSDL-to-Java and Java-to-WSDL Paths
Provider and Dispatch APIs
SAAJ and JAXP
JAX-RS for RESTful Services
JAXR

The Java API for XML Binding

The Need for Data Binding
XML Schema
Two Paths
JAXB Compilation
Mapping Schema Types to Java
Java-to-XML Mapping Using Annotations
Marshaling and Unmarshaling
Working with JAXB Object Models

The Java API for RESTful Services

Applications
Resources
Sub-Resources
Providers
Scanning and @ApplicationPath

Dispatching Requests to Methods

The Application Path
The @Path Annotation
The HTTP Method Annotations
Sub-Resource Locators
Annotation Inheritance and overriding

Parameter and Return Types

Simple Parameter Types
@Consumes and @Produces Annotations

@XXXParam Annotations
The @DefaultValue Annotation
Return Types
Binary Content
Delivering a File

Entities and Complex Content

Entity Providers
Built-In Entity Providers
Working with XML
Driving XML Representations from Schema

Context and Lifecycle

Reading Web Resources
Finding Java EE Components
Finding Databases
Security Contexts

Appendix A. Course Tools and Utilities

Appendix B. Handy Guide to Web-Services Acronyms