



Capstone Courseware, LLC

33 Boylston Street
Jamaica Plain, MA 02130

877-227-2477
capstonecourseware.com

561. Developing Java Web Services

Version 2.1

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.

(For training within the J2EE 1.4 environment, and a concentration on JAX-RPC and SAAJ, please see version 1.5.3 of this course.)

Prerequisites

- Strong Java programming skills are essential -- Course 103 is excellent preparation.
- Students must be able to read XML documents and to write well-formed XML by hand -- consider Course 501. 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.



Learning Objectives

- 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.

Timeline: 5 days.

IDE Support: Eclipse Galileo

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse Galileo. Students can code and compile all exercises from within Eclipse; deployment to the server is Ant-driven and can be triggered from the IDE as well. See also our orientation to Using Capstone's Eclipse Overlays, and please be advised that this is an optional feature; it is not a separate version of the course, and the course itself does not contain explicit Eclipse-specific lab instructions.





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
- Throwing Exceptions

Chapter 8. Client-Side Development

- Stubs and Proxies
- Generated Code
- Locating a Service
- Invoking a Service





Exception Handling

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

Appendix C. Quick Reference: Web-Services Acronyms

System Requirements

Hardware Requirements (Minimum)	1 GHz, 512 meg RAM, 1 gig disk space.
Hardware Requirements (Recommended)	2 GHz, 1 gig RAM, 1 gig disk space.
Operating System	Tested on Windows XP Professional. Course software should be viable on all systems which support the Java EE 5.0 SDK.





561. Developing Java Web Services

Outline

Network and Security

Limited privileges required -- please see our standard security requirements at <http://capcourse.com/Guides/Security.html>.

Software Requirements

All free downloadable tools.

