Testing the Methods of a Web Service Endpoint To Test the Service without a Client

To Test the Service without a Client
A Simple JAVWS Application Client
Coding the Application Client
Coding the Application Client
Running the Application Client
A Simple JAV-WS Web Client
Coding the Service
Running the Web Client
Types Supported by JAV-WS
Scheme 4-0 Jawa Mapping
Java-to-Scheme Mapping

Web Services Interoperability and JAX-WS Further Information about JAX-WS

Building RESTful Web Services with JAX-RS
 JAX-RS: Advanced Topics and Example

21. JAVAS Advanced Topics and Example PART V Enterprise Bares 22. Elementer Bares 23. Elementer Bares 24. Randrug Franch Bares 24. Randrug Franch Bares 25. Allessephone Bares Examples 26. Using patient Demon Bare Examples 26. Using patient Bares Container 27. Using Apporturous Barel Container 27. Using Apporturous Markel of woodcotin in Beases Part V Contasts and Dependency lipidion for the Javas EE Platform 28. Randrug Bares 29. Individual for Contests and Dependency lipidion Examples 20. Contests and Dependency lipidion for Bares 20. Contests and Dependency lipidion for Bares 21. Randrug Bares 22. Landrug Bares 23. Randrug Bares 24. Randrug Bares 25. Contests and Dependency lipidion for Examples 26. Landrug Bares 26. Landrug Bares 27. Landrug Bares 28. Landrug Bares 29. Landrug Bares 29. Landrug Bares 29. Landrug Bares 29. Landrug Bares 20. Contests and Dependency lipidion for Examples 20. Landrug Bares 20. Landru

- 31. Narrang the Abranced Cortests and Lependaricy Injection Exem
 Part V Plensistence
 20. Introduction to the Java Pensistence API
 31. Ruring the Pensistence Examples
 34. The Java Pensistence Examples
 34. The Java Pensistence Examples
 35. Living the Cortest API So Created Countries
 36. Creating and Using String Based Cotteria Quaries
 37. Controlling Coccurrent Access to Estify data with Looking
 38. Using a Socond-Level Cache with Java Pensistence API Applicate

- Part VIII Security:
 39. Infocution to Security in the Java EP Platform
 40. Getting Started Securing Web Applications
 41. Getting Started Securing Enterprise Applications
 42. Java EE Security; Advanced Topics

- 42. Java EE Sacurity Advanced Topics
 Part VIII Java EE Supporting Technologies
 43. Introduction to Java EE Supporting Tech
 44. Transactions
 45. Resources and Resource Adapters
 45. The Resource Adapter Example
 47. Java Message Service Concepts
 48. Java Message Service Concepts
 49. Base Villation: Advanced Topics
 50. Using Java EE Interceptors
 50. Using Java EE Interceptors

- Part IX Case Studies
 51. Duke's Bookstore Case Study Example
 52. Duke's Tutoring Case Study Example
 53. Duke's Forest Case Study Example

The Java EE 6 Tutorial

Home | Download | PDF | FAQ | Feedback



In JAX-WS, a web service operation invocation is represented by an XML-based protocol, such as SOAP. The SOAP specification defines the envelope structure, encoding rules, and conventions for representing web service invocations and responses. These calls and responses are transmitted as SOAP messages (XML files) over HTTP.

Although SOAP messages are complex, the JAX-WS API hides this complexity from the application developer. On the server side, the developer specifies the web service operations by defining methods in an interface written in the Java programming language. The developer also codes one or more classes that implement those methods. Client programs are also easy to code. A client creates a proxy (a local object representing the service) and then simply invokes methods on the proxy. With JAX-WS, the developer does not generate or parse SOAP messages. It is the JAX-WS nutrinue system that converts the API calls and responses to and from SOAP messages.

With JAX-WS, clients and web services have a big advantage: the platform independence of the Java programming language. In addition, JAX-WS is not restrictive: A JAX-WS client can access a web service that is not running on the Java platform, and vice versa. This flexibility is possible because JAX-WS uses technologies defined by the W3C: HTTP, SOAP, and WSDL. WSDL specifies an XML format for describing a service as a set of endpoints operating on messages.

Note - Several files in the JAX-WS examples depend on the port that you specified when you installed the GlassFish Server. These tutorial examples assume that the server runs on the default port, 8080. They do not run with a nondefault port setting.

The following topics are addressed here:

- Types Supported by JAX-WS
- Further Information about JAX-WS



Copyright © 2013, Oracle and/or its affiliates. All rights reserved. Legal Notice:

