

A client is to me a mere unit, a factor in a problem.

—Sir Arthur Conan Doyle

They also serve who only stand and wait.

—John Milton

...if the simplest things of nature have a message that you understand, rejoice, for your soul is alive.

—Eleonora Duse

Protocol is everything.

—Francoise Giuliani

Objectives

In this chapter you will learn:

- What a web service is.
- How to publish and consume web services in NetBeans.
- How XML, JSON, XML-Based Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) Architecture enable Java web services.
- How to create client desktop and web applications that consume web services.
- How to use session tracking in web services to maintain client state information.
- How to connect to databases from web services.
- How to pass objects of user-defined types to and return them from a web service.

Self-Review Exercises

31.1 State whether each of the following is *true* or *false*. If *false*, explain why.

a) All methods of a web service class can be invoked by clients of that web service.

ANS: False. Only methods declared with the `@WebMethod` annotation can be invoked by a web service's clients.

b) When consuming a web service in a client application created in NetBeans, you must create the proxy class that enables the client to communicate with the web service.

ANS: False. The proxy class is created by NetBeans when you add a web service client to the application.

c) A proxy class communicating with a web service normally uses SOAP to send and receive messages.

ANS: True.

d) Session tracking is automatically enabled in a client of a web service.

ANS: False. In the JAX-WS framework, the client must indicate whether it wants to allow the web service to maintain session information. First, you must cast the proxy object to interface type `BindingProvider`, then use the `BindingProvider`'s `getRequestContext` method to obtain the `RequestContext` object. Finally, you must use the `RequestContext`'s `put` method to set the property `BindingProvider.SESSION_MAINTAIN_PROPERTY` to `true`.

e) Web methods cannot be declared `static`.

ANS: True.

f) A user-defined type used in a web service must define both *get* and *set* methods for any property that will be serialized.

ANS: True.

g) Operations in a REST web service are defined by their own unique URLs.

ANS: True.

h) A SOAP-based web service can return data in JSON format.

ANS: False. A SOAP web service implicitly returns data in XML format.

31.2 Fill in the blanks for each of the following statements:

a) A key difference between SOAP and REST is that SOAP messages have data wrapped in a(n) _____.

ANS: SOAP message or SOAP envelope.

b) A web service in Java is a(n) _____—it does not need to implement any interfaces or extend any classes.

ANS: POJO (plain old Java object)

c) Web service requests are typically transported over the Internet via the _____ protocol.

ANS: HTTP.

d) To set the exposed name of a web method, use the _____ element of the `@WebMethod` annotation.

ANS: `operationName`.

e) _____ transforms an object into a format that can be sent between a web service and a client.

ANS: serialization.

f) To return data in JSON format from a method of a REST-based web service, the `@Produces` annotation is set to _____.

ANS: "application/json".

g) To return data in XML format from a method of a REST-based web service, the `@Produces` annotation is set to _____.

ANS: "application/xml".

Exercises

31.3 (*Phone Book Web Service*) PLEASE USE THE UPDATED VERSION OF THE EXERCISE POSTED AT: <http://bit.ly/hSLBon>

31.4 (*Phone Book Web Service Modification*) PLEASE USE THE UPDATED VERSION OF THE EXERCISE POSTED AT: <http://bit.ly/hSLBon>

31.5 (*Phone-Book Web Service with JSON*) Modify Exercise 31.4 so that the `PhoneBookEntry` class is passed to and from the web service as a JSON object. Use serialization to convert the JSON object into an object of type `PhoneBookEntry`.

ANS: NO SOLUTION PROVIDED

31.6 (*Blackjack Web Service Modification*) Modify the `Blackjack` web service example in Section 31.9 to include class `Card`. Modify web method `dealCard` so that it returns an object of type `Card` and modify web method `getHandValue` so that it receives an array of `Card` objects from the client. Also modify the client application to keep track of what cards have been dealt by using `ArrayLists` of `Card` objects. The proxy class created by NetBeans will treat a web method's array parameter as a `List`, so you can pass these `ArrayLists` of `Card` objects directly to the `getHandValue` method. Your `Card` class should include *set* and *get* methods for the face and suit of the card.

ANS: See solution on disk.

31.7 (*Airline Reservation Web-Service Modification*) Modify the airline reservation web service in Section 31.10 so that it contains two separate methods—one that allows users to view all available seats, and another that allows users to reserve a particular seat that is currently available. Use an object of type `Ticket` to pass information to and from the web service. The web service must be able to handle cases in which two users view available seats, one reserves a seat and the second user tries to reserve the same seat, not knowing that it is now taken. The names of the methods that execute should be `reserve` and `getAllAvailableSeats`.

ANS: NO SOLUTION PROVIDED

31.8 (*Morse Code Web Service*) In Exercise 16.22, you learned about Morse Code and wrote applications that could translate English phrases into Morse Code and vice versa. Create a SOAP-based web service that provides two methods—one that translates an English phrase into Morse Code and one that translates Morse Code into English. Next, build a Morse Code translator GUI application that invokes the web service to perform these translations.

ANS: NO SOLUTION PROVIDED

Making a Difference

31.9 (*Project: Spam Scanner Web Service*) In Exercise 16.27, you created a spam scanner application that scanned an e-mail and gave it a point rating based on the occurrence of certain words and phrases that commonly appear in spam e-mails and how many times the words and phrases occurred in the e-mail. Create a SOAP-based Spam scanner web service. Next, modify the GUI application you created in Exercise 16.27 to use the web service to scan an e-mail. Then display the point rating returned by the web service.

ANS: NO SOLUTION PROVIDED

31.10 (*SMS Web Service*) In Exercise 16.28, you created an SMS message-translator application. Create a SOAP-based web service with three methods:

- one that receives an SMS abbreviation and returns the corresponding English word or phrase,
- one that receives an entire SMS message and returns the corresponding English text, and
- one that translates English text into an SMS message.

4 Chapter 31 Web Services

Use the web service from a GUI application that displays the web service's responses.

ANS: NO SOLUTION PROVIDED

31.11 (*Gender-Neutrality Web Service*) In Exercise 1.12, you researched eliminating sexism in all forms of communication. You then described the algorithm you'd use to read through a paragraph of text and replace gender-specific words with gender-neutral equivalents. Create a SOAP-based web service that receives a paragraph of text, then replaces gender-specific words with gender-neutral ones. Use the web service from a GUI application that displays the resulting gender-neutral text.

ANS: NO SOLUTION PROVIDED

