Table of Contents

Java and SOAP By Robert Englander ISBN: 0-596-00175-4 Publisher: O'Reilly
Print Publication Date: 2002/05/01

Chapter 8. Alternative Techniques

Although SOAP is most commonly used for RPC, there are other possibilities as well. SOAP can also be used to create message-style services in which a SOAP envelope is sent to a server without the requirement that anything be returned to the caller. The message can contain any payload that is appropriate for the service — it doesn't have to look like a procedure call. This is similar to the way email systems work. When you send an email, you don't get anything back from the server, and there aren't any rules about what the email can contain. The server accepts your message and does its job without any more involvement from the sender. This is in sharp contrast to RPC services, where a return value is an integral part of the method invocation process.

Not all SOAP implementations support message-style services, but these services can be a useful model and are well worth investigating. GLUE doesn't support message services, but Apache SOAP does, so we'll use Apache SOAP to develop some examples.

8.1. SOAP Messaging

The format of the SOAP envelope for messaging services differs from the format used for RPC services. There is no concept of a method signature, method parameters, or return values in the messaging model. Basically, you're just sending some XML to a service. However, there are some requirements that allow you to direct the message to the correct service and service method. The name of the first child element of the SOAP `Body` must correspond to the name of the service method to be used, and the `xmlns` namespace attribute on that element specifies the name of the service. Here's a simple SOAP message that is directed to the `recordTemperature` method of the `urn:WeatherDiary` service for processing:

```
<SOAP_ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Body>
      <recordTemperature xmlns="urn:WeatherDiary"/>
      <temperature>75.5</temperature>
      <zipcode>50328</zipcode>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

8.2. Literal Encoding

We've seen how to use SOAP for RPC-type services, as well as how to pass around XML data in the less restrictive world of SOAP messaging services. But another possibility is to pass XML using RPC services. The problem is that RPC services rely on a type mapping

Java and SOAP By Robert Englander ISBN: 0-596-00175-4 Publisher: O'Reilly
Print Publication Date: 2002/05/01

mechanism to convert between XML elements and Java classes. It would be nice if you could use an RPC-style method invocation to pass parameters, with the result being standard XML. Well, you can. To do it, you need to use XML literal encoding, which tells the underlying SOAP framework that the data is XML and should remain that way.

Let's go back to the world of finance. We'll create a service that provides corporate information on companies in an XML format. The service is called `urn:CorporateDataService`, which has a method called `getDataForSymbol`. The service is an RPC service, and the method takes a stock symbol as the parameter. The data returned can include the company name, company address, year of incorporation, number of employees, and the high and low trading price for the past 52 weeks. This service will pull its data from another system that already keeps corporate data in XML format in a database. So it's not the responsibility of the service to format the data, only to pass it back to the caller. This scenario is perfect for using XML literal encoding. If the service were going to process the data in some way, it might just as well be designed to return a complex type that can be mapped to a Java class. Another reason for using literal encoding is that the client application may want to get the data in the XML format used on the back-end. It would be a waste of time to convert the XML on the server side, only to have the client put it back into XML. In this case, we're using SOAP to provide access to existing data, and then relying on existing software systems to do the processing.

Java and SOAP By Robert Englander ISBN: 0-596-00175-4 Publisher: O'Reilly
Print Publication Date: 2002/05/01