

WebServices Using JAX-RPC

Based on the presentation in O'Reilly's Webservices in a NutShell by Kim Topley

9/26/2003

B.Ramamurthy

1

JAX-RPC

- JAX-RPC (The Java API for XML-based RPC) is designed to provide a simple way for developers to create Web services server and Web services client.
- Based on remote procedure calls; so the programming model is familiar to Java developers who have used RMI or CORBA.
- Major difference between RMI and JAX-RPC is that messages exchanged are encoded in XML based protocol and can be carried over a variety of transport protocols such as HTTP, SMTP etc.
- You can use JAX-RPC without having to be an expert in XML, SOAP, or HTTP.

9/26/2003

B.Ramamurthy

2

The JAX-RPC Programming Model

- Services, ports and bindings
- JAX-RPC web service servers and clients
- JAX-RPC service creation
- JAX-RPC client and server programming environments
- Stubs and ties
- Client invocation modes
- Static and dynamic stubs and invocation

9/26/2003

B.Ramamurthy

3

Services, ports and bindings

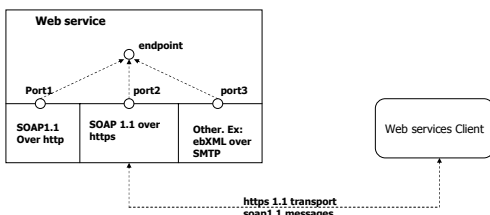
- Service endpoint interface or service endpoint that defines one or more operations that the web service offers.
- Access to an endpoint is provided by binding it to a protocol stack through a port.
 - A port has an address that the client can use to communicate with the service and invoke its operations.
- An endpoint can be bound to different ports each offering a different suite of protocols for interaction.

9/26/2003

B.Ramamurthy

4

Endpoint, Port and binding



9/26/2003

B.Ramamurthy

5

Web Service Clients and Servers

- JAX-RPC maps a
 - web service operation to a java method call.
 - service endpoint to a Java Interface.
- Thus one way to begin implementation of a web service in JAX-RPC is to define a Java interface with a method for each operation of the service along with a class that implements the interface. Of course, following the rules of remote invocation etc.
- Now visualize client/server invocation in the same address space and lets compare it with remote invocation.

9/26/2003

B.Ramamurthy

6

Local Date Service

```
//server
public class DataService {
    public Date getDate() {
        return new Date();}
}
//client
Public class Appln {
    public static void main (..) {
        DataService instance = new DataService();
        Date date = instance.getDate();
        System.out.println (" The date is" + date);
    }
}
```

- In the case of the remote call a layer of software is used to convey the method call from client to server. This layer of software is provided by JAX-RPC runtime.

9/26/2003

B.Ramamurthy

7

JAX-RPC service creation

- A service definition describes the operations that it provides and the data types that they require as argument and provide as return values.
- This definition can be made available as a document written in WSDL.
- From a WSDL document, JAX-RPC can generate the Java code required to connect a client to a server leaving one to write only the logic of the client application itself.
- Since WSDL is language independent the server can be in .net, Jax-rpc or any other compatible platform.

9/26/2003

B.Ramamurthy

8

JAX-RPB service creation (contd.)

- Define the service a Java interface.
- Generate WSDL using the tools provided with JAX-RPC package.
- Advertise it in a registry for the client to lookup and import it.
- For publication and lookup any other technology such as J2EE can be used.

9/26/2003

B.Ramamurthy

9

Client and Server Programming Environment

- JAX-RPC API is distributed over a set of packages:
 - javax.xml.rpc
 - javax.xml.rpc.encoding
 - javax.xml.rpc.handler
 - javax.xml.rpc.handler.soap
 - javax.xml.rpc.holders
 - javax.xml.rpc.server
 - javax.xml.rpc.soap

9/26/2003

B.Ramamurthy

10

Stubs and Ties

- Client Side: Stub object has the same methods as the service implementation class.
 - Client application is linked with the stub.
 - When it invokes a method stub delegates the call to the JAX-RPC runtime so that appropriate SOAP message can be sent to the server.
 - On completion the result return back in the reverse path as above.
- Server side:
 - Message received must be converted into a method call on actual service implementation. This functionality is provided by another piece of glue called tie.
 - Tie extracts method name and parameter from SOAP message.
 - Tie also converts the result of the method call back into a response message to be returned to client JAX-RPC runtime.
- Developer need not write these classes (tie and stub) since JAX-RPC comes with tools to generate them.

9/26/2003

B.Ramamurthy

11

Client Invocation Modes

- Synchronous request-response mode (tightly coupled).
- One-way RPC (loosely coupled): no value returned, no exception thrown, need to bypass stub layer, use Dynamic Invocation Interface (DII).

9/26/2003

B.Ramamurthy

12

Client Invocation Modes