

Developing Web Services Using Java 2 Enterprise Edition (J2EE)

Mike Yawn

Hewlett-Packard

mike.yawn@hp.com

Agenda

- What are web services?
- Web Services Data: XML and SOAP
- Web Services APIs: JAX*
- Web Services Registries
- Web Services Products

About the Speaker

- Author of 'J2EE and JAX: Developing Web Applications and Web Services'
- iPDL Java Customer Satisfaction Team
- SIG JAVA co-chair



J2EE & JAX Topics

- **XML**, XSLT
- Servlets
- JavaServer Pages
- Struts
- Wireless access
- Java Message Service
- Java RMI
- JNDI
- **SOAP**
- **Web Services**
- JAX* APIs
 - JAXP, JAXB, **JAXM**, **JAX-RPC**, JAXR
- Enterprise JavaBeans
 - 3 types: Session, Entity, Message-Driven

Web Services Defined

- A2A
- Coarse grained, loosely coupled
- May be document-centric or RPC-based
- XML data
- Passed using standard internet protocols
- Description published in registry

Web Services Data: XML

- eXtensible Markup Language
- Markup tags that look similar to HTML
 - Latest revision of HTML is an XML language
- Described via a Document Type Definition (older) or XML Schema (new)
- Well-formed documents vs. valid documents

XML Document & Schema

```
<?xml version="1.0" ?>
<game week="01" played="yes">
  <home city="Atlanta"
        name="Falcons" score="14" />
  <visitor city="San Francisco"
           name="49ers" score="17" />
</game>
```

```
<xsd:schema>
  <xsd:element name="game">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="nfl:home" />
        <xsd:element ref="nfl:visitor" />
      </xsd:sequence>
      <xsd:attribute name="week" use="required"
                    type="xsd:int"/>
      <xsd:attribute name="played" use="optional"
                    type="xsd:boolean" default="true" />
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="home">
    <xsd:complexType>
      <xsd:attribute name="city" use="required"
                    type="xsd:string" />
      <xsd:attribute name="name" use="required"
                    type="xsd:string" />
      <xsd:attribute name="score" use="required"
                    type="xsd:int" />
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

SOAP

- Simple Object Access Protocol
- Uses XML for document content (payload), but also for envelope, headers, and faults.

Sample SOAP Message

```
<soap-env:Envelope
  xmlns:soap-env=
    "http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header/>
  <soap-env:Body>
    <nfl:game
      xmlns:nfl="http://nfl.mydomain.com"
      week="1" played="true">
      <home team="Atlanta Falcons" score="14"/>
      <visitor team="San Francisco 49ers" score="17"/>
    </nfl:game>
  </soap-env:Body>
</soap-env:Envelope>
```

Java XML APIs

JAXP	Java API for XML Parsing
JAXB	Java API for XML Binding
JAXM	Java API for XML Messaging
JAX-RPC	Java API for XML-based Remote Procedure Calls
JAXR	Java API for XML Registries

XML Parsing: JAXP

- Java implementation of DOM, SAX, XSLT APIs.
- Apache Xerces parser is reference implementation of parsing
- Apache Xalan processor is reference implementation for transformations

JAXP: sample code for DOM

```
public Source generateXMLDocument() {
    DocumentBuilderFactory df = null;
    DocumentBuilder builder = null;

    df = DocumentBuilderFactory.newInstance();
    try {
        builder = df.newDocumentBuilder();
    } catch (ParserConfigurationException pce) {
        pce.printStackTrace();
        return null;
    }

    doc = builder.newDocument();
    Element stElement = doc.createElement("standings");
    doc.appendChild(stElement);
    Element dsElement = getDivisionStandings();
    Element csElement = getConferenceStandings();
    Element teamsElement = getTeams();
    stElement.appendChild(dsElement);
    stElement.appendChild(csElement);
    stElement.appendChild(teamsElement);
    return new DOMSource(doc);
}
```

JAXP: sample code for SAX

```
public void startElement(String namespace,
                        String localname,
                        String qname,
                        Attributes attrs) {
    if (localname.equals("conference")) {
        inConfPart = true;
        // Check name attribute; match?
        if (attrs.getValue("name").equals(confName))
            interestingConference = true;
        else
            interestingConference = false;
    }
    if (localname.equals("teams")) {
        inTeamsPart = true;
    }
    if (localname.equals("team")) {
        if (interestingConference)
            interestingTeam = true;
    }
    if (localname.equals("teamDetails")) {
        // must defer decision until name parsed
        interestingTeam = true;
    }
    if (localname.equals("name")) {
        if (interestingTeam) {
            inNameElement = true;
            nameSB = new StringBuffer();
        }
    }
}
```

JAXP: sample code for XSLT

```
<!-- Template to match division -->
<xsl:template match="division">
  <table border="1">
    <thead>
      <tr>
        <th colspan="10">
          <h2>
            <xsl:value-of select="@name"/>
          </h2>
        </th>
      </tr>
      <tr>
        <th><xsl:text>Team</xsl:text></th>
        <th><xsl:text>Wins</xsl:text></th>
        <th><xsl:text>Losses</xsl:text></th>
<!-- other fields omitted due to space limitation on slide ->
      </tr>
    </thead>
    <tbody>
      <xsl:apply-templates select="team"/>
    </tbody>
  </table>
</xsl:template>

<!-- Template to match team in division -->
<xsl:template match="division/team">
  <tr>
    <td><xsl:value-of select="name"/></td>
    <xsl:variable name="currentTeam" select="name"/>
    <xsl:apply-templates
      select="teamDetails[name=$currentTeam]"/>
  </tr>
</xsl:template>
```

XML Binding: JAXB

- Java API for XML Binding
- Map XML documents to Java classes (1st release)
- Map Java classes to XML document (future)
- ‘Unmarshal’ a document into a tree of classes.
- Early version of code embedded in JAX-RPC.

JAXB sample code

```
// Build object tree from XML document
FileInputStream fis = null;
try {
    File stand = new File(
        "../../presentation/xml/standings.xml");
    fis = new FileInputStream(stand);
    standings = standings.unmarshal(fis);
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (UnmarshalException e) {
    e.printStackTrace();
} finally {
    fis.close();
}

// Traverse tree to find some data
// - W/L/T record for Denver Broncos
Teams teams = standings.getTeams();
java.util.List details = teams.getTeamDetails();
for (int i=0; i<details.size(); i++) {
    TeamDetails team = (TeamDetails) details.get(i);
    if (team.getName().equals("Denver Broncos")) {
        String wins = team.getWins();
        String ties = team.getTies();
        String losses = team.getLosses();
        System.out.println("Denver's record is " +
            wins + "-" + losses + "-" + ties + ".");
        break;
    }
}
}
```


XML Messaging: JAXM

- Key web service technology for building document-centric services.
- Programming style is to deal extensively with XML.
- Support for asynchronous services

JAXM sample code

```
try {
    MessageFactory mf = MessageFactory.newInstance();
    message = mf.createMessage();
    SOAPPart sp = message.getSOAPPart();
    envelope = sp.getEnvelope();
    header = envelope.getHeader();
    body = envelope.getBody();
    sef = SOAPElementFactory.newInstance();
} catch (SOAPException e) {
    e.printStackTrace();
}

for (int i=0; i<gameList.size(); i++) {
    game = (Game) gameList.get(i);

    try {
        SOAPBodyElement soapGame =
            body.addBodyElement(bodyName);
        soapGame.addAttribute(weekName,
            Integer.toString(game.getWeek()));

        SOAPElement soapHome = sef.create(homeName);
        soapHome.addAttribute(teamName, game.getHome());
        soapHome.addAttribute(scoreName,
            Integer.toString(game.getScore()[1]));
        soapGame.addChildElement(soapHome);

    } catch (SOAPException e) {
        e.printStackTrace();
    }

    sendMessage(message);
}
```

XML RPC: JAX-RPC

- Service defined via Java interfaces; mapping to XML is done by tools.
- Usage is similar to Java RMI or CORBA.
- This is expected to be the most widely used web service model for Java.

JAX-RPC sample code

```
try {
    stub = (JaxRpcStandingsIF_Stub)(new NFLStandings_Impl().getJaxRpcStandingsIF());
} catch (JAXRPCException e) { e.printStackTrace(); }
/* code to initialize URL removed */
stub._setProperty(javax.xml.rpc.Stub.ENDPOINT_ADDRESS_PROPERTY, url);

SoapTeamImpl[] teams = null;
String[] divnames = null;

try {
    stub.init();
    divnames = stub.getDivisionNames();
} catch (RemoteException e) { e.printStackTrace(); System.exit(-1); }
/* code to initialize divnames removed */
try {
    teams = stub.getDivisionStandings(divnames[a]);
} catch (RemoteException e) {
    e.printStackTrace(); System.exit(-1);
} catch (NoSuchDivisionException e) {
    e.printStackTrace(); System.exit(-1);
}

for (int i=0; i<teams.length; i++) {
    System.out.print(i+1 + ". ");
    printDetail(teams[i]);
}
```

Web Service Registries

- A web service registry may be based on UDDI or ebXML.
- Stores or references a WSDL definition of service
- Accessed programmatically via JAXR protocol

WSDL

- WSDL is an XML language for describing Web Services.
- Incorporates XML Schema for description of specific messages.
- Associates 'requests' with 'responses' and 'faults' they may generate.
- Provides detail on reaching the service (URL, protocol, encoding)

Example WSDL Document

```
<definitions name="NFLStandingsService" >
  <types>
    <schema>
      <complexType name="SoapTeamImpl">
        <sequence>
          <element name="record" type="tns:ArrayOfint"/>
          <element name="name" type="string"/>
          <element name="winLossPct" type="float"/></sequence></complexType>
        </schema></types>

    <message name="getConferenceStandings">
      <part name="String_1" type="xsd:string"/></message>
    <message name="getConferenceStandingsResponse">
      <part name="result" type="ns2:ArrayOfSoapTeamImpl"/></message>

    <portType name="JaxRpcStandingsIF">
      <operation name="getConferenceStandings">
        <input message="tns:getConferenceStandings"/>
        <output message="tns:getConferenceStandingsResponse"/>
        <fault name="NoSuchConferenceException" message="tns:NoSuchConferenceException"/></operation>
      </portType>
    <binding name="JaxRpcStandingsIFBinding" type="tns:JaxRpcStandingsIF">
      <operation name="getConferenceStandings">
        <input>
          <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
            namespace="http://hp.com/wsdl"/></input>
          <output>
            <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
              namespace="http://hp.com/wsdl"/></output>
          <fault name="NoSuchConferenceException">
            <soap:fault encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" use="encoded"
              namespace="http://hp.com/wsdl"/></fault>
          <soap:operation soapAction=""/></operation>
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc"/></binding>
      <service name="NFLStandings">
        <port name="JaxRpcStandingsIFPort" binding="tns:JaxRpcStandingsIFBinding">
          <soap:address location="REPLACE_WITH_ACTUAL_URL"/></port></service></definitions>
```

UDDI

- Universal Description, Discovery, and Integration
- Publish link to your WSDL document; can be searched by company name (white pages), business type (yellow pages), or technical description (green pages)
- UDDI4j, jointly developed by IBM and HP, is functionally ahead of the JAXR implementation.

ebXML

- Electronic Business XML
- Includes Registry/Repository, but also other standards for messaging, collaboration agreements, document schemas

Products

- HP Web Services Platform
 - Registry Composer, Service Composer, UDDI registry
- Application Servers
 - BEA WebLogic, HP-AS
- Tomcat
- Java XML Pack, JWSDP