SOAP with Attachments API for Java

CSE 486/586
October 20, 2004

SAAJ & JAXM

- **SAAJ** was previously part of JAXM (Java API for XML Messaging) package
- JAXM and SAAJ conforms to SOAP 1.1 specification and SOAP with attachments specification
- SAAJ 1.1.1 - *javax.xml.soap* package provides API for creating, sending request-response SOAP messages
- JAXM 1.1.1 - *javax.xml.messaging* package provides API needed for using a Messaging Provider to send one-way messages

SAAJ - Package

- Two perspectives:
  - **Messages** - SAAJ helps create XML messages conforming to SOAP 1.1 by making Java API calls
  - **Connections** - SOAP Connection object represents *point-to-point* connection for sending and receiving *request-response* messages using SAAJ APIs
- SAAJ Classes and Interfaces represent XML elements
- Interface **Node** - base class for all interfaces, classes representing XML elements in SOAP message

SOAP Message with No Attachments
Creating simple SOAP Message

- Create a message using a **MessageFactory** object.
- SAAJ returns a new **SOAPMessage** object that already contains
  - A **SOAPPart** object that contains
    - A **SOAPEnvelope** object that contains
      - An empty **SOAPHeader** object
      - An empty **SOAPBody** object
- Access message elements using getter methods
- Create **SOAPBodyElement** object to hold the content
- Create an associated **Name** object for unique identification

Sending messages on SOAP connection

- Create a connection using a **SOAPConnectionFactory** object.
- Invoke the method **call** on the **SOAPConnection** object to send the SOAP message. Call method:
  - takes two arguments, the message being sent and the destination (usually identified by URL)
  - blocks until it receives the response.
  - returns the response SOAPMessage received from the remote party
- Close the connection

SOAP Connection

- Point-to-Point connection
- Request-Response messages
- Blocking Send

Getting the Content of a Message

- **getSOAPBody** on SOAPMessage gives a SOAPBody object
- **getFirstChild** on SOAPBody object returns an Iterator object that contains all of the child elements (as Java Object) identified by the Name object passed as the argument.
- Need to cast the Objects returned by getFirstChild to SOAPBodyElement objects.
- **getValue** on SOAPBodyElement returns the value of the immediate child of the element that calls the method.
Adding Attachments to SOAP Message

- Create a `AttachmentPart` object.
- Add content to the `AttachmentPart` using the `setContent` method.
- Set `AttachmentPart` Headers (helpful in accessing a particular attachment when a message has multiple attachments):
  - Content-Type: required
  - Content-ID: optional
  - Content-Location: optional
- Add `AttachmentPart` to SOAP message using the `addAttachmentPart` method.

Accessing Attachments in SOAP Message

- SOAPMessage class provides two versions of the method `getAttachments` for retrieving its `AttachmentPart` objects depending upon the argument passed.
  - **No argument**: returns a `java.util.Iterator` object over all the `AttachmentPart` objects in a message.
  - **Mime-Headers object (a list of MIME headers)**: returns an iterator over the `AttachmentPart` objects that have a header that matches one of the headers in the list.

SAAJ – HTTP Mapping

- SOAPMessage
- Content-Type
- Content-Length
- SOAPAction
- SOAPVersion
- SOAPMethod
- SOAPBody
- SOAPBody/Element
- SOAPBody/Text
- HTTPMessage
- Content-Type
- Content-Length
- SOAPAction
- SOAPVersion
- SOAPMethod
- SOAPBody
- SOAPBody/Element
- SOAPBody/Text
SOAP Fault

- **SOAP Fault**: specific type of XML element placed in the body of a reply SOAP message to convey status information; to report a fault while processing the original request.
- SOAP specification requires that there be no more than one SOAP fault in the message body.
- Attributes (get and set methods):
  - Fault code (e.g. VersionMismatch, MustUnderstand, Client, Server)
  - Fault string
  - Fault actor
- Creating Fault Elements
  - addFault() method on SOAP Body
  - setFaultCode, setFaultString, setFaultActor
  - addDetail( ) method for a detailed entry
- HTTP response code 500 for a SOAP Message with fault

More on SAAJ and related technologies

- **JWSDP Tutorial Document**
  - Chapter 13 SAAJ
  - [http://java.sun.com/j2ee/1.4/docs/tutorial/doc/SAAJ.html#wp69380](http://java.sun.com/j2ee/1.4/docs/tutorial/doc/SAAJ.html#wp69380)
- **Java Web Services in a Nutshell**
  - Chapter 3 SAAJ
  - [http://www.oreilly.com/catalog/javawsian/chapter/ch03.pdf](http://www.oreilly.com/catalog/javawsian/chapter/ch03.pdf)

Receiving SOAP Message: Servlets

- No SAAJ API for of receiving a SOAP message (corresponding to call() on sender's side)
- HTTP Servlets can be used to receive the messages:
  - Servlet API contains support to handle a payload delivered over an HTTP connection, and
  - SOAPConnection class uses HTTP as the default protocol when transmitting messages.
- Extend the HTTP Servlet to handle the SOAP messages
- Inside doPost or doGet:
  - Extract request from SOAP Message
  - Prepare response SOAP Message
  - Write it to HTTPServletResponse Object