

# SAAJ: SOAP Attachment API for Java

## Based on [Ref1](#), [Ref2](#)

B. Ramamurthy

10/1/2004

1

## Introduction

- SAAJ stands for **SOAP with Attachments API for Java**.
- SAAJ needs to construct SOAP messages piece by piece and extract information from response messages.
- SAAJ provides a convenient library to construct and read SOAP messages.
- Lets you send and receive SOAP messages across the network.
- We will look at SOAP, attachments and an example for constructing and sending a SOAP with attachments.

10/1/2004

2

## JAX-RPC – SAAJ – JAXM

- JAX-RPC is concerned with allowing a client program to make a remote procedure call to a web service without exposing underlying XML-based protocols.
- SAAJ is a more low-level API concerned with messages (attachments) exchanged between the web service and its clients.
- SAAJ defines the namespace `javax.xml.soap`. The classes that reside in that package initially formed part of the Java API for XML Messaging (JAXM), but were recently separated into their own API.
- JAXM relies on SAAJ for SOAP message construction and manipulation, and adds message reliability and other features specific to XML messaging.
- Whereas SAAJ is a required component of J2EE (Java 2 Platform, Enterprise Edition) 1.4, JAXM is not.

JAXRPC → SAAJ → JAXM  
RPC → Attachments → Messaging  
Synchronous -----> asynchronous

10/1/2004

3

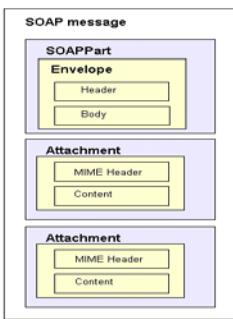
## SOAP

- Defines a way to wrap information represented in XML.
- Wrapped information is exchanged between peer entities.
- Entities interpret the information.
- Act on the interpreted information to provide a service.
- SOAP's attachment feature extends a SOAP message to include, in addition to the regular SOAP part, zero or more attachments.

10/1/2004

4

## SOAP Message with Attachments [\(ref\)](#)



10/1/2004

5

## SOAP Message

- Envelop is the container for header and body.
- Payload is the useful content of the message.
- Header carries information about processing and routing of payload.
  - Ex: digital signature to guarantee integrity of payload.
  - Authentication information
- Body is the only mandatory part of the envelop.
  - Contains the payload
- Everything in SOAP envelop must be encoded in XML.
- Each attachment is defined by a [MIME](#) (Multipurpose Internet Mail Extension) type and can assume any content represented as a byte stream.

10/1/2004

6

## Example

- Consider a real estate agency network that needs to distribute descriptions and photographs of homes for sale to a centralized real estate search portal. If the portal operates a servlet allowing the posting of SOAP messages with attachments, a real estate agency could update its listings with a few SOAP messages, including photos of those homes. The SOAP message body might embed the property description, and SOAP attachments could carry the image files.
- For this problem we will look at:
  - SOAP with attachments
  - Sending
  - Receiving

10/1/2004

7

## SOAP Message for Real estate Problem



10/1/2004

8

## Sending a Message

- For creating and editing SOAP messages, you can initialize a MessageFactory and a SOAPFactory. MessageFactory lets you create SOAP messages, and SOAPFactory provides the methods to create individual parts of a SOAP message:

```
SOAPConnectionFactory spConFactory =  
    SOAPConnectionFactory.newInstance();  
  
SOAPConnection con = spConFactory.createConnection();  
  
SOAPFactory soapFactory = SOAPFactory.newInstance();
```

10/1/2004

9

## Constructing the Message

- The following example shows the simplest method that creates an empty SOAP message with an envelope, and header and body in that envelope. Since you don't need a SOAP header in this message, you can remove that element from the message:

```
SOAPMessage message = factory.createMessage();  
SOAPHeader header = message.getSOAPHeader();  
header.detachNode();
```

- Adding the XML structure to the message body proves straightforward:

```
SOAPBody body = message.getSOAPBody();
```

[Construct the message piece by piece.](#)

10/1/2004

10

## Sending the message

- SOAPMessage response = connection.call(message, end point);
- The SOAP connection's call() method is synchronous; it blocks until it receives a reply.
- Reply is a SOAP message.

10/1/2004

11

## Summary

- We looked at SOAP with attachments and support provided by SAAJ to process this.
- We looked at constructing the SOAP message and code for sending it.
- Homework: study the receiving of the SOAP message and processing it using SAAJ. (exam question?)

10/1/2004

12