



Messages dans les Services Web



SOAP: Un standard pour l' échange de messages

Simple Object Access Protocol

Un message SOAP:

- Permet la transmission d' un message au format XML
- d' un nœud émetteur vers un nœud receveur
- en passant éventuellement par un certain nombre de « noeuds » intermédiaires
 - Les noeuds intermédiaires, peuvent maintenir des logs, faire des vérifications ou bien encore modifier ou rediriger les messages

Standards utilisés par SOAP

SOAP s'appuie:

- Sur XML et XML Schema pour la représentation des messages
- Sur des protocoles internet classiques pour la transmission des messages:
 - HTTP(S), SMTP, JMS, FTP, ...
 - La liste des protocoles utilisables n'est pas fermée



Schémas d'interaction classiques

- Requête / réponse (~RPC)
- Réception (demande d'envoi)
- Envoi (sans attente de réponse)
- Communication bi-directionnelle
 - En particulier pour les transactions de longue durée
 - Ex: suivi de commande

Structure d'un document SOAP...



... et codage en XML

```
<?xml version='1.0' ?>
<env:Envelope
  xmlns:env=
    "http://www.w3.org/2003/05/soap-envelope">
<env:Header>
  ... <!-- optionnel -->
</env:Header>
<env:Body>
  ... <!-- Ici le contenu du message -->
</env:Body>
</env:Envelope>
```



SOAP: Body

- Le contenu est le message en soi
- Son interprétation est à faire par l'application
- Peut correspondre à un appel de fonction à distance: SOAP-RPC

Exemple de message SOAP

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:Header/>
<env:Body>
  <commande xmlns="http://www.glaces.com">
    <parfum>Fraise</parfum>
    <quantite>16</quantite>
    <livraison>
      <adresse>8 bvd N.Bohr Villeurbanne</adresse>
      <destinataire>M2TI</destinataire>
    </livraison>
  </commande>
</env:Body>
</env:Envelope>
```


Exemple de message SOAP suite

```
<?xml version='1.0' ?>  
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">  
  <env:Header/>  
  <env:Body>  
    <confirmation xmlns="http://www.glaces.com">  
      <heure>10h</heure>  
    </confirmation>  
  </env:Body>  
</env:Envelope>
```

SOAP: Fautes

- Utilisé lorsqu'une erreur survient
 - En général dans une réponse à un message reçu auparavant
- Se place dans le body
- Le contenu est une explication de l'erreur:
 - Une valeur
 - Une explication textuelle
 - Un morceau de document spécifique à l'application (opt.)
 - Le rôle du nœud à l'origine de l'erreur
 - Utile en présence de nœuds intermédiaires

Exemple de faute SOAP

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
<env:Header/>
<env:Body>
  <env:Code>
    <env:Value>env:Receiver</env:Value>
  </env:Code>
  <env:Reason>
    <env:Text xml:lang="fr-FR">Plus de glaces à la fraise</env:Text>
    <env:Text xml:lang="en-GB">Nomore strawberry ice-creams</env:Text>
  </env:Reason>
</env:Body>
</env:Envelope>
```



Types d'erreur SOAP

<env:Value>

- env:VersionMismatch
- env:MustUnderstand
- env:DataEncodingUnknown
- env:Sender
- env:Receiver



SOAP: Header

- Paramètres annexes
 - Propriétés du message
- Peut être utilisé pour des besoins annexes:
 - Corrélation de messages
 - Informations d' authentification
 - Informations liées aux transactions
- Chaque application peut décider de la signification des headers
 - Mais les nœuds intermédiaires peuvent ajouter/supprimer/traiter/transformer des headers



Traitement du message par les nœuds intermédiaires

- Un nœud intermédiaire peut traiter un ou plusieurs éléments du header
 - Sauf spécification contraire (via env:relay) cet élément est retiré du message
 - env:role permet de spécifier les nœuds destinés à traiter l'élément



Attributs particuliers dans les éléments du header

- **env:mustUnderstand**
 - Si la valeur est `true` cet élément doit être pris en compte (sauf cas particulier avec `env:role`)
- **env:role**
 - <http://www.w3.org/2003/05/soap-envelope/role/xxx>
 - `next`
 - `ultimateReceiver`
 - `none`
 - Autre
- **env:relay**
 - `true`: on doit transférer le header au nœud suivant, même s'il a été traité

Exemple de Header

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header g:xmlns= »http://www.glaces.com/">
    <g:client-id env:relay="true">1234-56789</g:client-id>
    <g:dateAndTime
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">2007-10-12 08:30:25</g:dateAndTime>
  </env:Header>
  <env:Body>
    <commande xmlns="http://www.glaces.com"> ... </commande>
  </env:Body>
</env:Envelope>
```




Liaison avec les protocoles

- Support de transmission entre deux nœuds adjacents
 - Le protocole peut varier le long du chemin du nœud de départ au nœud final
- Protocoles Internet:
 - HTTP(S), SMTP, JMS, ...
- SOAP Protocol Binding
 - Comment un protocole particulier peut être utiliser pour transférer un message SOAP
 - Quels schémas d'interaction sont autorisés



Binding HTTP(S)

- Le plus utilisé
 - Compatible avec les proxy
 - Les services peuvent être implémentés dans des serveurs Web
 - Connexion sécurisée avec HTTPS
- Utilisation avec GET
 - Interaction type réception
- Utilisation avec POST
 - Interaction type requête/réponse
 - On utilise souvent un paramètre SOAPAction pour accélérer le sous-traitement du message



Avec GET (source:W3C)

GET /travelcompany.example.org/reservations?code=FT35ZBQ HTTP/1.1

Host: travelcompany.example.org

Accept: text/html;q=0.5, application/soap+xml

Avec GET (source:W3C)

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset="utf-8"

Content-Length: nnnn

```
<?xml version='1.0' ?>
```

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-  
envelope">
```

```
<env:Header>
```

```
...
```

```
</env:Header>
```

```
<env:Body>
```

```
....
```

```
</env:Body>
```

```
</env:Envelope>
```

Avec POST (requête)

POST /Commandes HTTP/1.1

Host: www.glaces.com

Content-Type: application/soap+xml;charset="utf-8"

Content-Length: nnnn

```
<?xml version='1.0' ?>
```

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

```
<env:Header/>
```

```
<env:Body>
```

```
  <commande xmlns="http://www.glaces.com">
```

```
    <parfum>Fraise</parfum>
```

```
    <quantite>16</quantite>
```

```
    <livraison>
```

```
      <adresse>8 bvd N.Bohr Villeurbanne</adresse>
```

```
      <destinataire>M2TI</destinataire>
```

```
    </livraison>
```

```
  </commande>
```

```
</env:Body>
```

```
</env:Envelope>
```

Avec POST (réponse)

HTTP/1.1 200 OK

Content-Type: application/soap+xml; charset="utf-8"

Content-Length: nnnn

```
<?xml version='1.0' ?>
```

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

```
<env:Header/>
```

```
<env:Body>
```

```
  <confirmation xmlns="http://www.glaces.com">
```

```
    <heure>10h</heure>
```

```
  </confirmation>
```

```
</env:Body>
```

```
</env:Envelope>
```

Avec POST (faute)

HTTP/1.1 500 Internal Server Error

Content-Type: application/soap+xml; charset="utf-8"

Content-Length: nnnn

```
<?xml version='1.0' ?>
```

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

```
<env:Header/>
```

```
<env:Body>
```

```
  <env:Code>
```

```
    <env:Value>env:Receiver</env:Value>
```

```
  </env:Code>
```

```
  <env:Reason>
```

```
    <env:Text xml:lang="fr-FR">Plus de glaces à la fraise</env:Text>
```

```
    <env:Text xml:lang="en-GB">Nomore strawberry ice-creams</env:Text>
```

```
  </env:Reason>
```

```
</env:Body>
```

```
</env:Envelope>
```



Données binaires

- Encodage par défaut:
 - Utilisation de l' encodage des données binaire Base64 [a-z,A-Z,0-9,+,/]
 - Correspond au type XML Schema prédéfini `base64Binary`
 - Utilise (beaucoup trop) de place

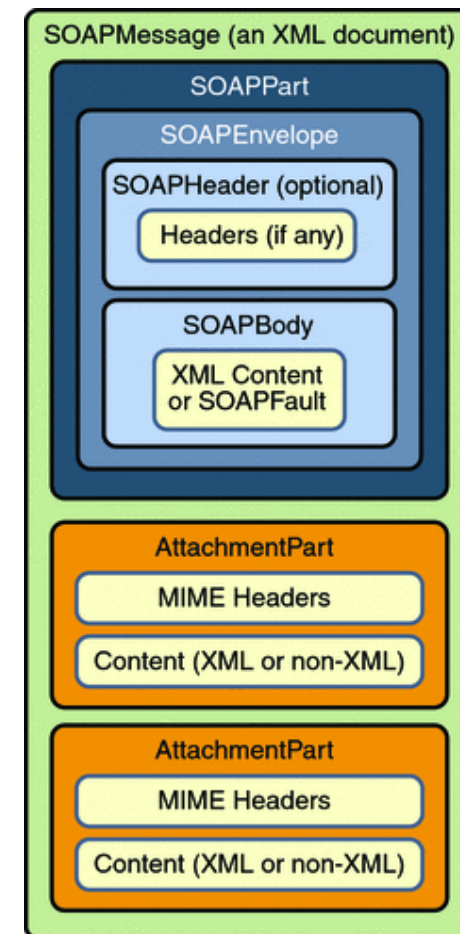


MTOM et XOP

- MTOM: Message Transmission Optimization Mechanism
- XOP: XML-binary Optimized Packaging
- Inclusion de données binaires fournies en attachement du message
 - Nécessite un protocole de transfert compatible (HTTP(S), SMTP)

SOAP with attachment

- Ajouter les données binaires comme des parties supplémentaires dans un paquet MIME/Multipart
- Indiquer le contenu binaire d'un élément par une URI pointant vers la partie concernée du paquet MIME/Multipart



(Source Java Tutorial)

Mise en oeuvre

...

```
<monElement xmime:contentType"typeMIMEDesDonneesBinaires">  
  <xop:Include  
    xmlns:xop='http://www.w3.org/2003/12/xop/include'  
    href="cid:nomDeLAttachement"/>  
</monElement>
```

...

--limite--

Content-Type: typeMIMEDesDonneesBinaires

Content-Transfer-Encoding: binary

Content-ID: <nomDeLAttachement>

Données binaires

--limite--

[Exemple W3C](#)



WS-Addressing

- Ensemble de propriétés
 - Identification de message
 - Adresse de départ et d'arrivée d'un message
 - Adresse de réponse (normale et fautive)
 - La signification du message
 - Liaison entre messages
- Les valeurs sont des URIs
- Se traduit par des éléments du header dans les messages SOAP

WS-Addressing: End Point Reference

- Adresse d' un nœud

```
<wsa:EndpointReference>  
  <wsa:Address>xs:anyURI</wsa:Address>  
  <wsa:ReferenceParameters>  
    xs:any*  
  </wsa:ReferenceParameters> ?  
  <wsa:Metadata>xs:any*</wsa:Metadata>?  
</wsa:EndpointReference>
```

WS-Addressing: propriétés

xmlns:wsa="http://www.w3.org/2005/08/addressing"

<wsa:To>xs:anyURI</wsa:To> ?

<wsa:From>wsa:EndpointReferenceType</wsa:From> ?

<wsa:ReplyTo>wsa:EndpointReferenceType</wsa:ReplyTo> ?

<wsa:FaultTo>wsa:EndpointReferenceType</wsa:FaultTo> ?

<wsa:Action>xs:anyURI</wsa:Action>

<wsa:MessageID>xs:anyURI</wsa:MessageID> ?

<wsa:RelatesTo RelationshipType="xs:anyURI"?>xs:anyURI</
wsa:RelatesTo> *

<wsa:ReferenceParameters>

xs:any*</wsa:ReferenceParameters> ?

WS-Addressing: exemple

(Source W3C)

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
              xmlns:wsa="http://www.w3.org/2005/08/addressing">
  <env:Header>
    <wsa:MessageID>http://example.com/6B29FC40-CA47-1067-
      B31D-00DD010662DA</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://example.com/business/client1</wsa:Address>
    </wsa:ReplyTo>
    <wsa:To>http://example.com/fabrikam/Purchasing</wsa:To>
    <wsa:Action>http://example.com/fabrikam/SubmitPO</wsa:Action>
  </env:Header>
  <env:Body>
    ...
  </env:Body>
</env:Envelope>
```



APIs Java

- SAAJ (Sun)
 - SOAP with Attachments API for Java
 - Inclus dans JAX-WS (J2EE/Metro)
- Axis 2 (apache.org)
 - AXIOM (DOM/StAX API)
 - Fourni également une implémentation de SAAJ



SAAJ

- Étend l' API DOM standard de Java
 - compatibilité avec le code DOM existant
- Fourni des classes et des méthodes
 - pour représenter des messages SOAP
 - avec attachement
 - pour envoyer des messages et recevoir une réponse

Code SAAJ Client

(Source: J2EE Tutorial)

```
import javax.xml.soap.*;

...

MessageFactory factory = MessageFactory.newInstance();
SOAPMessage message = factory.createMessage();
SOAPPart soapPart = message.getSOAPPart();
SOAPBody body = message.getSOAPBody();
QName bodyName = new QName("http://wombat.ztrade.com",
    "GetLastTradePrice", "m");
SOAPBodyElement bodyElement = body.addBodyElement
    (bodyName);
// bodyElement est utilisable comment un Element DOM
```



Code SAAJ Client - suite

(Source: J2EE Tutorial)

```
SOAPConnectionFactory soapConnectionFactory  
    = SOAPConnectionFactory.newInstance();  
SOAPConnection connection =  
    soapConnectionFactory.createConnection();  
java.net.URL endpoint = new URL("http://  
    wombat.ztrade.com/quotes");  
SOAPMessage response = connection.call  
    (message, endpoint);  
connection.close();
```

Axis 2

- Implémentation de la « pile » des services Web par apache.org
- Disponible en Java et en C
- Peut être installé dans un simple serveur Tomcat
 - Plus léger qu' un serveur J2EE complet
- Implémentation de JAX-WS (et donc de SAAJ)

Axis 2 Client

Packages

- org.apache.axiom ...
- org.apache.axis2 ...

```
OMFactory fac = OMAbstractFactory.getOMFactory();
```

```
OMNamespace omNs = fac.createOMNamespace("http://  
axiom.service.quickstart.samples/xsd", "tns");
```

```
OMEElement method = fac.createOMEElement("getPrice",  
omNs);
```

Axis 2 Client -suite

```
EndpointReference targetEPR = new EndpointReference  
("http://localhost:8080/axis2/services/  
StockQuoteService");
```

```
Options options = new Options();
```

```
options.setTo(targetEPR);
```

```
options.setTransportInProtocol  
(Constants.TRANSPORT_HTTP);
```

```
ServiceClient client = new ServiceClient();
```

```
client.setOptions(options);
```

```
OMElement answer = client.sendReceive(method);
```