

CH4. MESSAGES D'ERRERES SOAP :

1- introduction

2- message d'erreur soap

3- message avec pièces jointes

1- INTRODUCTION :

Message d'erreur :

Un message d'erreur est un message SOAP généré par un nœud intermédiaire ou par le récepteur final suite à une erreur dans le traitement d'un message de requête et envoyé à l'expéditeur initial, afin d'expliquer la nature de l'erreur rencontrée.

Le corps d'un message d'erreur ne contient que l'entrée "*Fault*". Cette entrée sert à définir le type du message d'erreur, et à donner plus de détails sur cette erreur.

Selon le type d'erreur signalée, l'entête <env:Header> du message peut exister ou non. S'il existe il doit contenir des entrées spécifiques (obligatoires) définies par le protocole SOAP.

2- MESSAGE D'ERREUR SOAP :

Structure de l'entrée Fault:

L'entrée « *fault* » peut contenir les cinq éléments suivant dans l'ordre (4 pour soap 1.1) , les deux 1^{er} éléments sont obligatoires et les autres sont facultatifs :

```
<env:Body>
  <env:Fault>
    <env:Code>
      <env:Value> code attribué à l'erreur </env:Value>
      <env:Subcode> <env:Value> autre code d'erreur </env:Value>
      </env:Subcode>
    </env:Code>
    <env:Reason>
      la nature de l'erreur
    </env:Reason>
    <env:Node> ...</env:Node>
    <env:Role> ...</env:Role>
    <env:Detail> ... </env:Detail>
  </env:Fault>
</env:Body>
```

2- MESSAGE D'ERREUR SOAP :

Structure de l'entrée Fault:

1- L'élément « Code » (« faultcode » pour soap 1.1) :

- C'est un élément obligatoire, utilisé pour donner le code soap attribué à l'erreur signalée.
- Dans la version 1.2 cet élément contient un éléments fils « Value » (obligatoire) et un autre « Subcode » (optionnel).

- Le code soap attribué à l'erreur est reporté dans le sous-élément « value ».

- S'il y a un autre code d'erreur généré par l'application qui a provoqué l'erreur, celui-ci sera reporté dans le sous-élément « subcode ».
- Ce dernier s'il existe sera structure de la même manière de « code ».

- Pour la version 1.1 un seul code d'erreur est reporté (celui attribué par soap) directement dans l'élément « faultcode ».

2- MESSAGE D'ERREUR SOAP :

Structure de l'entrée Fault:

1. L'élément « Code » (« faultcode » pour soap 1.1) :

Selon les codes pouvant être signalés par cet élément, on distingue cinq types de messages d'erreur dans SOAP 1.2 et quatre dans SOAP 1.1 :

Code d'erreur		Explication
Version 1.2	Version 1.1	
VersionMismatch	VersionMismatch	Version non supportée (pb dans l'espace de nom indiquant la version)
MustUnderstand	MustUnderstand	Une ou plusieurs entrées dans l'entête non comprises.
DataEncodingUnknown	-	Codage inconnu
Sender	Client	Msg incorrect (structure), pas assez d'informations.
Receiver	Server	Erreur de traitement (provoquée par le nœud actuel).

2- MESSAGE D'ERREUR SOAP :

Structure de l'entrée Fault:

2- L'élément « *reason* » (« *faultstring* » pour soap 1.1) :

- C'est un élément obligatoire, qui sert à expliquer la nature de l'erreur à l'utilisateur.
- La version 1.2 du SOAP est doté d'un support multilingue afin d'expliquer l'erreur dans différentes langue.

```
<env:Body>
  <env:Fault>
    <env:Code>
      <env:Value> env:VersionMismatch </env:Value>
    </env:Code>
    <env:Reason>
      <env:Text xml:lang="en"> Version Mismatch </env:Text>
    </env:Reason>
  </env:Fault>
</env:Body>
```

Elle peut contenir plusieurs éléments textes (*text element*), chacun doit contenir la chaîne de caractère écrite dans une langue à part.

2- MESSAGE D'ERREUR SOAP :

Structure de l'entrée Fault:

3- L'élément « *node* » :

Requis pour tous les nœuds SOAP, excepté l'ultime destinataire. Il contient un URI décrivant le nœud ayant entraîné l'erreur.

4- L'élément « *role* » (« *faultactor* » pour la version 1.1:

Requis pour tous les nœuds SOAP, excepté l'ultime destinataire. Il contient un URI décrivant le rôle (service) ayant entraîné l'erreur.

5- L'élément « *detail* » :

Cet élément est destiné à transporter des informations d'erreur spécifiques à aux éléments du corps du message (soap 1.1). Pour soap 1.2 il reporte différents détails sur l'erreur.

2- MESSAGE D'ERREUR SOAP :

Structure de l'entrée Fault:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:rpc='http://www.w3.org/2003/05/soap-rpc'>
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:Sender</env:Value>
        <env:Subcode>
          <env:Value>rpc:BadArguments</env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en-US">Processing error</env:Text>
        <env:Text xml:lang="cs">Chyba zpracování</env:Text>
      </env:Reason>
      <env:Detail>
        <e:myFaultDetails
          xmlns:e="http://travelcompany.example.org/faults">
            <e:message>Name does not match card number</e:message>
            <e:errorcode>999</e:errorcode>
          </e:myFaultDetails>
        </env:Detail>
      </env:Fault>
    </env:Body>
  </env:Envelope>
```

Msg incorrect (structure),

informations incorrectes
arguments incorrects lors de
l'appel de la fonction

explique la nature de l'erreur
En anglais, et En tchèque

2- MESSAGE D'ERREUR SOAP :

Structure de l'entrée Fault:

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <env:NotUnderstood qname="t:transaction"
                      xmlns:t="http://thirdparty.example.org/transaction"/>
  </env:Header>
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:MustUnderstand</env:Value>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en-US">Header not understood</env:Text>
        <env:Text xml:lang="fr">En-tête non compris</env:Text>
      </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

2- MESSAGE D'ERREUR SOAP :

Message d'erreur de type « VersionMismatch »:

Un message d'erreur de type "*VersionMismatch*" doit contenir obligatoirement une entrée entête appelée "*Upgrade*" contenant une liste ordonnée d'espaces de noms représentant les versions supportées.

Chaque version est reportée dans un sous-élément appelé « *SupportedEnvelope* ».

L'élément « Upgrade » ne doit pas contenir d'attribut « EncodingStyle ».

2- MESSAGE D'ERREUR SOAP :

Message d'erreur de type « VersionMismatch »:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  xmlns:xml="http://www.w3.org/XML/1998/namespace">

  <env:Header>
    <env:Upgrade>
      <SupportedEnvelope qname="ns1:Envelope"
        xmlns:ns1="http://www.w3.org/2003/05/soap-envelope">

      <SupportedEnvelope qname="ns2:Envelope"
        xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/">
    </env:Upgrade>
  </env:Header>

  <env:Body>
    <env:Fault>
      <env:Code>      <env:Value>      env:VersionMismatch      </env:Value>
      </env:Code>
      <env:Reason> <env:Text xml:lang="en">Version do not match
        </env:Text>
      </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

2- MESSAGE D'ERREUR SOAP :

Message d'erreur de type « MustUnderstand »:

Un message d'erreur de type « *MustUnderstand* » doit contenir obligatoirement des entrées entête appelée « *NotUnderstood* », chacune contient l'entrée entête du message de requête dont le traitement est impossible.

Dans la version 1.1 le mot utilisé est « MisUnderstood ».

Ces éléments ne doivent pas contenir d'attribut « EncodingStyle ».

2- MESSAGE D'ERREUR SOAP :

Message d'erreur de type « MustUnderstand »:

```
<env:Envelope xmlns:env='http://www.w3.org/2003/05/soap-envelope'>
  <env:Header>
    <abc:Extension1 xmlns:abc='http://example.org/2001/06/ext'
      env:mustUnderstand='true'> ... </abc:Extension1 >
    <def:Extension2 xmlns:def='http://example.com/stuff'
      env:mustUnderstand='true' > ... </def:Extension2 >
    <m:Extension3 xmlns:def='http://example.com/stuff'
      env:mustUnderstand='false' > ... </m:Extension3 >
  </env:Header>
  <env:Body>
    . . .
  </env:Body>
</env:Envelope>
```

2- MESSAGE D'ERREUR SOAP :

```
<env:Envelope xmlns:env='http://www.w3.org/2003/05/soap-envelope'>
  <env:Header>
    <env:NotUnderstood qname='abc:Extension1'
      xmlns:abc='http://example.org/2001/06/ext' >
    <env:NotUnderstood qname='def:Extension2'
      xmlns:def='http://example.com/stuff' >
  </env:Header>
  <env:Body>
    <env:Fault>
      <env:Code><env:Value>env:MustUnderstand</env:Value></env:Code>
      <env:Reason>
        <env:Text xml:lang='en'>One or more mandatory
          SOAP header blocks not understood
        </env:Text>
      </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

2- MESSAGE D'ERREUR SOAP :

TRANSITION DE VERSION SOAP:

Le problème d'interaction entre un nœud SOAP1.1 et un autre nœud SOAP1.2 est géré de la manière suivante :

- Lorsqu'un nœud SOAP supportant uniquement la version 1.1 reçoit un message SOAP écrit dans la version 1.2, il doit générer un message d'erreur de type « VersionMismatch » structuré selon la version 1.1.
- Lorsqu'un nœud SOAP supportant la version 1.2 reçoit un message SOAP écrit dans la version 1.1, il doit soit :
 - Traiter le message dans sa version (1.1) si c'est possible.
 - Générer un message d'erreur de type « VersionMismatch » structuré selon la version 1.1, mais en lui rajoutant l'entrée « UpGrade » indiquant le support de la version 1.2.

2- MESSAGE D'ERREUR SOAP :

Exemple de message d'erreur de type « MustUnderstand »:

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <env:NotUnderstood qname="t:transaction"
      xmlns:t="http://thirdparty.example.org/transaction"/>
  </env:Header>
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value>env:MustUnderstand</env:Value>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en-US">Header not understood</env:Text>
        <env:Text xml:lang="fr">En-tête non compris</env:Text>
      </env:Reason>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

2- MESSAGE D'ERREUR SOAP :

Exemple de message d'erreur de type « Sender »:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
  xmlns:m="http://www.example.org/timeouts"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <env:Body>
    <env:Fault>
      <env:Code>
        <env:Value> env:Sender </env:Value>
        <env:Subcode>
          <env:Value> m:MessageTimeout </env:Value>
        </env:Subcode>
      </env:Code>
      <env:Reason>
        <env:Text xml:lang="en"> Sender Timeout
      </env:Text>
      </env:Reason>
      <env:Detail>
        <m:MaxTime> P5M </m:MaxTime>
      </env:Detail>
    </env:Fault>
  </env:Body>
</env:Envelope>
```

2- MESSAGE D'ERREUR SOAP :

Exemple de message d'erreur de type « Server »:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode> soap:Server
    </faultcode>
    <faultstring> Insufficient funds </faultstring>
    <detail>
      <x:TransferError xmlns:x="urn:examples-org:banking">
        <sourceAccount> 22-342439 </sourceAccount>
        <transferAmount> 100.00 </transferAmount>
        <currentBalance> 89.23 </currentBalance>
      </x:TransferError>
    </detail>
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```

3. MESSAGE AVEC PIÈCES JOINTES:

L'utilisation de SOAP avec des données non XML est parfois très utile, par exemple si on désire transmettre des images à partir d'une caméra à travers une liaison sans fil en utilisant SOAP vers un serveur client.

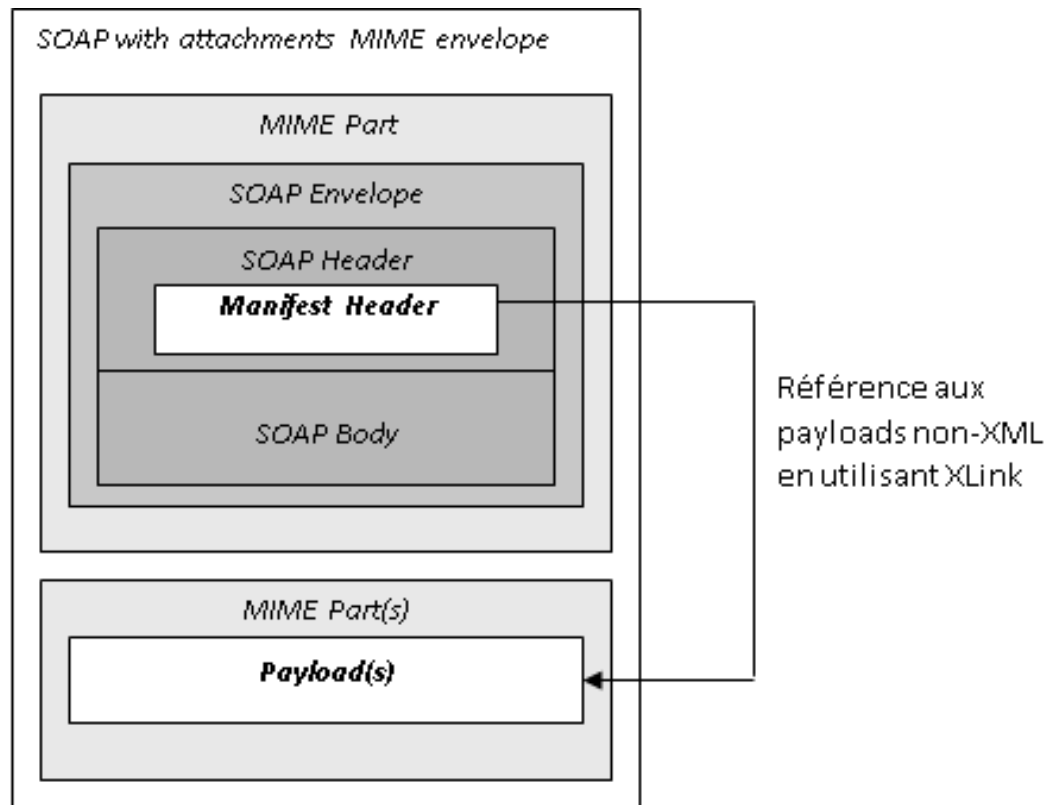
Le support des données non XML requiert un packaging complémentaire du message qui peut être fourni par une structure MIME multipart et impacte la liaison d'un message à son protocole de transport sous-jacent.

L'enveloppe MIME contient un ensemble de parties MIME individuelles, la 1ère contient le message SOAP qui doit comporter un bloc Manifest Header créé par le Handler Message Manifest. La 2ème partie MIME contient les payloads qui peuvent être des documents XML ou d'autres formats.

Un message avec attachement est structuré comme suit:

3. MESSAGE AVEC PIÈCES JOINTES:

Un message avec attachement est structuré comme suit :



Structure d'un message avec attachements

3. MESSAGE AVEC PIÈCES JOINTES:

```
<env:Envelope xmlns:env='http://www.w3.org/2001/12/soap-envelope'
  <env:Header>
    <n:Manifest xmlns:n="http://exampleorg/manifest">
      <n:Reference n:id="image01"
        Xlink:href="cid:payload-1"
        Xlink:role="http://example.org/image">
        <n.description> Photo numero 1 </n.description>
      </n:Reference>
      <n:Reference n:id="image02"
        Xlink:href="cid:payload-2"
        Xlink:role="http://example.org/image">
        <n.description> Photo numero 2 </n.description>
      </n:Reference>
    </n:Manifest>
  </env:Header>
  <env:Body>
    . . .
  </env:Body>
</env:Envelope>
```

FIN