

Fault Handling and Remote Procedure Call Use of SOAP over HTTP

CS 183 – Hypermedia and the Web

Fault Handling in SOAP

If an error occurs when processing a SOAP message, a fault message is generated. The fault information is placed in the SOAP body, in a `env:Fault` element (where `env` is the standard SOAP envelope XML namespace (<http://www.w3.org/2003/05/soap-envelope>)). The `Fault` element has two mandatory subelements, `env:Code` and `env:Reason`, and possibly optional `env:Detail` and `env:Node` elements.

env:Code: holds one or two child elements, `env:Value` and `env:Subcode`. `env:Value` contains a value indicating the kind of error that occurred:

- *env:Sender* – the sender of the message incorrectly formed the message (syntax error) or the message was lacking information (missing parameters or authentication information).
- *env:Receiver* – an error processing the message on the receiving side (e.g., the receiver had to communicate with some other information service, and was unable to)
- *env:MustUnderstand* – an element of the Header had the `MustUnderstand` attribute set to true, but the SOAP node didn't understand the message
- Also `VersionMismatch` and `DataEncodingUnknown`

env:Subcode can contain additional, more specific error information (for example, which specific kind of Sender error occurred?) When SOAP is being used for RPC style interaction, the Subcode can convey a specific RPC error message.

env:Reason contains a human readable description of the error condition, in one or more *env:Text* elements.

env:Detail when present, is a place where application-specific error information can be placed. This error information is not likely to be generally understood by all SOAP nodes, only nodes aware of the specific application that generated the error information.

Error example, from the SOAP 1.2 primer:

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

Use of SOAP over HTTP

SOAP supports two styles of interaction over HTTP:

1. Remote procedure call (request message in SOAP, response message in SOAP), transported using HTTP POST (this is called the “SOAP request-response message exchange pattern”)
2. Response in SOAP (request message as HTTP GET w/URL parameters, response message formatted using SOAP) – this is called the “SOAP response message exchange pattern.”

Example of GET use of SOAP (from SOAP primer):

Request message (typical HTTP):

```
GET /travelcompany.example.org/reservations?code=FT35ZBQ HTTP/1.1
Host: travelcompany.example.org
Accept: text/html;q=0.5, application/soap+xml
```

Note that Accept header is specifying that the preferred response is of MIME type application/soap+xml, i.e., a SOAP response.

Response message:

```
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>
    <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:mustUnderstand="true">
      <m:reference>uuid:093a2dal-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2001-11-30T16:25:00.000-05:00</m:dateAndTime>
    </m:reservation>
  </env:Header>
  <env:Body>
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:x="http://travelcompany.example.org/vocab#"
      env:encodingStyle="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
      <x:ReservationRequest
        rdf:about="http://travelcompany.example.org/reservations?code=FT35ZBQ">
        <x:passenger>Åke Jógvan Øyvind</x:passenger>
        <x:outbound>
          <x:TravelRequest>
            <x:to>LAX</x:to>
            <x:from>LGA</x:from>
            <x:date>2001-12-14</x:date>
          </x:TravelRequest>
        </x:outbound>
        <x:return>
          <x:TravelRequest>
            <x:to>JFK</x:to>
            <x:from>LAX</x:from>
            <x:date>2001-12-20</x:date>
          </x:TravelRequest>
        </x:return>
      </x:ReservationRequest>
    </rdf:RDF>
  </env:Body>
</env:Envelope>
```

Note use of RDF to represent data in the env:Body element. At present, there is no normative representation of data within a SOAP message, though the SOAP specifications provide a description of one that can be used.

Example of POST use of SOAP:

SOAP request:

```
POST /Reservations HTTP/1.1
Host: travelcompany.example.org
Content-Type: application/soap+xml; charset="utf-8"
Content-Length: nmn

<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" >
  <env:Header>
    <t:transaction
      xmlns:t="http://thirdparty.example.org/transaction"
      env:encodingStyle="http://example.com/encoding"
      env:mustUnderstand="true" >5</t:transaction>
  </env:Header>
  <env:Body>
    <m:chargeReservation
      env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
      xmlns:m="http://travelcompany.example.org/">
      <m:reservation
        xmlns:m="http://travelcompany.example.org/reservation">
        <m:code>FT35ZBQ</m:code>
        </m:reservation>
        <o:creditCard xmlns:o="http://mycompany.example.com/financial">
          <n:name xmlns:n="http://mycompany.example.com/employees">
            Åke Jógvan Øyvind
          </n:name>
          <o:number>123456789099999</o:number>
          <o:expiration>2005-02</o:expiration>
        </o:creditCard>
      </m:chargeReservation>
    </env:Body>
  </env:Envelope>
```

SOAP Response:

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>
    <t:transaction
      xmlns:t="http://thirdparty.example.org/transaction"
      env:encodingStyle="http://example.com/encoding"
      env:mustUnderstand="true">5</t:transaction>
  </env:Header>
  <env:Body>
    <m:chargeReservationResponse
      env:encodingStyle="http://www.w3.org/2003/05/soap-encoding"
      xmlns:m="http://travelcompany.example.org/">
      <m:code>FT35ZBQ</m:code>
      <m:viewAt>
        http://travelcompany.example.org/reservations?code=FT35ZBQ
      </m:viewAt>
    </m:chargeReservationResponse>
  </env:Body>
</env:Envelope>
```