



Einführung in SOAP

Seminar E-Services

von Christoph Kurek

Was ist SOAP?

- SOAP steht für Simple Object Access Protokoll
- SOAP ist ein Standardisiertes Verpackungsprotokoll für Nachrichten
- SOAP ist eine Anwendung der XML-Spezifikation
- Es ist **keine** komplette verteilte Architektur



Vorteile von SOAP

- + XML Basierend
- ⇨ Plattform-, Betriebssystem-, und Programmiersprachenunabhängig
- + Einfach und Erweiterbar
- + Baut auf Standardprotokollen auf, wie HTTP, SMTP, FTP
- + Firewall unabhängig (Vorteil?)

Nachteile von SOAP

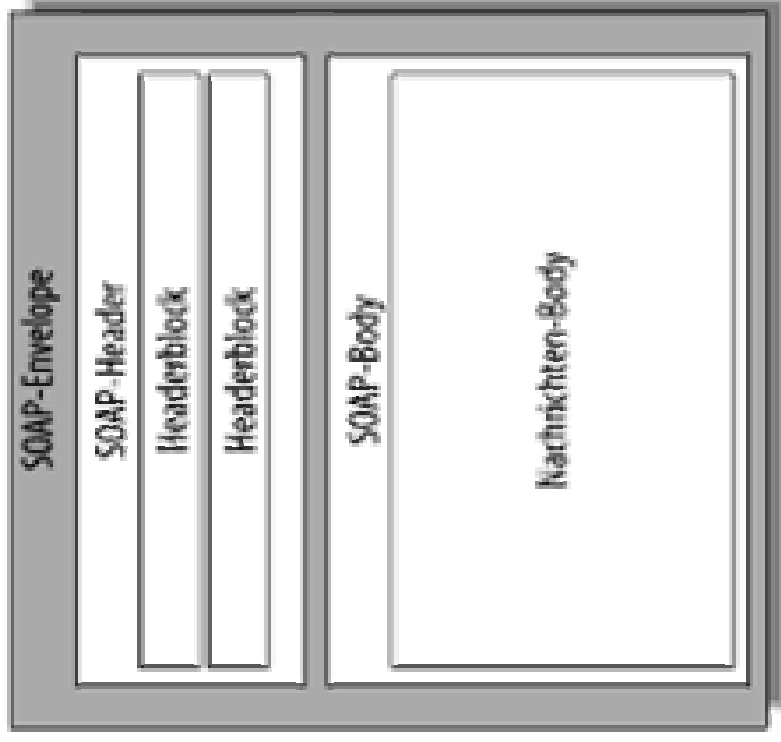
- Sicherheitsstandards noch nicht Standardisiert (Signatur, Verschlüsselung)
- Nachrichtengröße
- Performance

SOAP Nachrichten

- Besteht aus einem Envelope, der
 - Optional einen Header und
 - danach zwingend einen Body enthält.
- Alles, was der XML Syntax entspricht, kann im Body und Header stehen.
- XML Syntax einer SOAP Nachricht beruht auf dem Namensraum

<http://www.w3.org/2001/06/soap-envelope>

SOAP Nachrichten

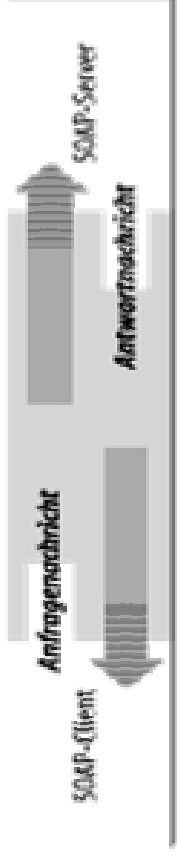


SOAP Nachrichten

```
<env:Envelope xmlns:env="http://www.w3.org/2002/06/soap-
  envelope">
  <env:Header>
    <n:alertcontrol
      xmlns:n="http://example.org/alertcontrol">
      <n:priority>1</n:priority>
      <n:expires>2001-06-22T14:00:00-05:00</n:expires>
    </n:alertcontrol>
  </env:Header>
  <env:Body>
    <m:alert xmlns:m="http://example.org/alert">
      <m:msg>Pick up Mary at school at 2pm</m:msg>
    </m:alert>
  </env:Body>
</env:Envelope>
```

SOAP als RCP

- RCP steht für Remote Procedure Call
- Nachrichten treten paarweise auf
 - Client fragt, Server antwortet (in SOAP Spezifikation nicht vorgeschrieben)



SOAP als RPC

- Aufruf von `public Float getQuote(String symbol);`

```
<s:Envelope xmlns:s="http://www.w3.org/2001/06/soap-envelope">  
<s:Header>  
  <m:transaction xmlns:m="soap-transaction"  
    s:mustUnderstand="true">  
    <transactionID>1234</transactionID>  
  </m:transaction>  
</s:Header>  
<s:Body>  
  <n:getQuote xmlns:n="urn:QuoteService">  
    <symbol xsi:type="xsd:string"> IBM </symbol>  
  </n:getQuote>  
</s:Body>  
</s:Envelope>
```

SOAP als RPC

- Mögliche Antwort:

```
<s:Envelope xmlns:s="http://www.w3.org/2001/06/soap-  
envelope">  
<s:Body>  
<n:getQuoteResponse xmlns:n="urn:QuoteService">  
  <value xsi:type="xsd:float"> 98.06 </value>  
</n:getQuoteResponse>  
</s:Body>  
</s:Envelope>
```

Attribut mustUnderstand

- Headerblöcke werden „optional“ verarbeitet.
- Der mit Attribut `mustUnderstand="true"` muss verstanden werden.
- Falls nicht der Fall, muss eine Fehlermeldung erfolgen.
- Es ist unabhängig davon, ob die Nachricht im Body verarbeitet werden kann.

Datencodierung in SOAP

■ Von W3C wird ein Codierungsstil vorgeschlagen

```
<s:Body>  
  <n:getQuote xmlns:n="urn:QuoteService"  
    s:encodingStyle="http://www.w3.org/2001/06/soap-  
      encoding">  
    <symbol xsi:type="xsd:string">IBM</symbol>  
  </n:getQuote>  
</s:Body>
```

Datencodierung in SOAP

3 Mögliche Arten, den Datentyp auszudrücken

1. Verwendung des Attributes `xsi:type` und Bezug auf einen Datentyp des XML Schemas
2. Verweis auf ein XML-Schema-Dokument, der den `xsi:type` des Elementes festlegt
3. Verweis auf ein Schema Dokument, der den Datentyp des Elements festlegt

Datencodierung in SOAP

1.

```
<person>
  <name xsi:type="xsd:string">Erika Mustermann</name>
</person>
```
2.

```
<person xmlns="personschema.xsd">
  <name>Erika Mustermann</name>
</person>
<!-- wobei "personschema.xsd" das Element name als
type="xsd:string" definiert -->
```
3.

```
<person xmlns="urn:some_namespace">
  <name>Erika Mustermann</name>
</person>
<!-- wobei "urn:some_namespace" einen Namensraum
anzeigt, in dem die Werte von name-Elementen Strings
sind -->
```

SOAP Datentypen

2 Mögliche Schreibweisen für die Typisierung von Werten:

1. Anonym

```
<SOAP-ENC:int>36</SOAP-ENC:int>
```

2. Benannt

```
<value xsi:type="xsd:int">36</value>
```

SOAP Datentypen

■ Mehrfachreferenz

```
<value xsi:type="xsd:int" id="v1">42</value>  
<value href="#v1" />
```

■ Objektreferenz

- *Java*

```
Address address = new Address();  
Person person = new Person();  
person.setAddress(address);
```

- *SOAP*

```
<Person> <Address href="#address1" /> </Person>  
<Address id="address1" />
```


SOAP Datentypen

- Byte Arrays

```
<some_binary_data xsi:type="SOAP-ENC:base64">  
    aDF4JIK34KJjk3443kj1kj43SDF43==  
</some_binary_data>
```

SOAP Datentypen

■ Ein- und Zweidimensionale Arrays

```
<names xsi:type="SOAP-ENC:Array" SOAP-  
ENC:arrayType="xsd:string[2,2]">  
  <name xsi:type="xsd:string">a1d1</name>  
  <name xsi:type="xsd:string">a2d1</name>  
  <name xsi:type="xsd:string">a1d2</name>  
  <name xsi:type="xsd:string">a2d2</name>  
</names>
```

```
<names xsi:type="SOAP-ENC:Array" SOAP-  
ENC:arrayType="xsd:string[4]">  
  <name xsi:type="xsd:string">a1d1</name>  
  <name xsi:type="xsd:string">a2d1</name>  
  <name xsi:type="xsd:string">a3d1</name>  
  <name xsi:type="xsd:string">a4d1</name>  
</names>
```

SOAP Datentypen

■ Dünn besetzte Arrays

```
<names xsi:type="SOAP-ENC:Array" SOAP-ENC:arrayType="xsd:string[10,10]">  
  <name SOAP-ENC:position=" [ 2, 5 ] ">Wert</name>  
  <name SOAP-ENC:position=" [ 5, 2 ] ">Wert</name>  
</names>
```

■ Leere Variablen

```
<name xsi:type="xsd:string" xsi:nil="true" />
```

SOAP Nachrichtenaustausch

- Nachricht: eine Einwegübertragung
- Idee: Eine Nachricht durchläuft viele Zwischenverarbeitungsstellen (Pipeline)
- Nachrichtenpfad
- Akteur

SOAP Nachrichtenaustausch

- SOAP Targeting: Bestimmt, welcher Teil der Nachricht für wen
- Nur in Bezug auf Headerblöcke
- Attribut `actor`

SOAP Nachrichtenaustausch

```
<s:Envelope xmlns:s="...">  
  <s:Header>  
    <x:signature actor="uri:SignatureVerifier"> ...  
      </x:signature>  
    </s:Header>  
  <s:Body>  
    <abc:purchaseOrder>...</abc:purchaseOrder>  
  </s:Body>  
</s:Envelope>
```

SOAP Nachrichtenaustausch

- Konstruktion des Nachrichtenpfades nicht in der SOAP Spezifikation!
- Erweiterungen: z.B. SOAP Routing Protocol (WS-Routing von Microsoft)

SOAP Nachrichtenaustausch

```
<s:Envelope xmlns:s="...">
  <s:Header>
    <m:path xmlns:m="http://schemas.xmlsoap.org/rp/"
      s:mustUnderstand="true">
      <m:action>http://www.im.org/chat</m:action>
      <m:to>http://D.de/irgendein/endpoint</m:to>
      <m:fwd>
        <m:via>http://B.ch</m:via>
        <m:via>http://C.at</m:via>
      </m:fwd>
      <m:rev>
        <m:via/>
      </m:rev>
      <m:from>mailto:lieschen.mueller@spitze.de</m:from>
      <m:id> uuid:84b9f5d0-33fb-4a81-b02b-5b760641c1d6 </m:id>
    </m:path>
  </s:Header>
  <s:Body> ... </s:Body>
</s:Envelope>
```

SOAP Faults

Enthaltene Informationen:

- Fault-Code
Ein Wert, der den Fehler identifiziert
- Fault-String
Fehlererklärung
- Fault-Actor
Der Knoten, an den Fehler auftrat
- Fault-Details
Einzelheiten, nur falls der Fehler unmittelbar mit einem Problem in Body zusammenhängt.

SOAP Faults

```
<s:Envelope xmlns:s="...">
<s:Body>
  <s:Fault>
    <faultcode>Client.Authentication</faultcode>
    <faultstring> Ungültiger Identitätsnachweis
    </faultstring>
    <faultactor>http://spitze.de</faultactor>
    <details> <!--Anwendungsspezifische Einzelheiten-->
      </details>
    </s:Fault>
  </s:Body>
</s:Envelope>
```

SOAP Faults

Standardcodes für SOAP-Faults:

- Server
Fehler, der nicht direkt mit Nachrichtenverarbeitung zusammenhängt
- Client
Problem mit der Nachricht
- VersionMismatch
Falscher Namensraum für SOAP Envelope
- MustUnderstand
*Ein von Empfänger nicht verstandener
mustUnderstand Headerblock*

SOAP Faults

- MustUnderstand-Faults

```
<s:Envelope xmlns:s="...">
```

```
<s:Header>
```

```
  <f:Misunderstood QName="abc:transaction" xmlns:="soap-  
  transactions" />
```

```
</s:Header>
```

```
<s:Body>
```

```
  <s:Fault>
```

```
    <faultcode>MustUnderstand</faultcode>
```

```
    <faultstring>Header nicht verstanden </faultstring>
```

```
    <faultactor>http://spitze.de</faultactor>
```

```
  </s:Fault>
```

```
</s:Body>
```

```
</s:Envelope>
```

SOAP Faults

■ VersionMismatch-Fault

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <V:Upgrade xmlns:V="http://www.w3.org/2001/06/soap-upgrade">
      <envelope qname="ns1:Envelope"
        xmlns:ns1="http://www.w3.org/2001/06/soap-envelope"/>
    </V:Upgrade>
  </s:Header>
  <s:Body>
    <s:Fault>
      <faultcode>s:VersionMismatch</faultcode>
      <faultstring>Versionsfehler</faultstring>
    </s:Fault>
  </s:Body>
</s:Envelope>
```

Literatur

- <http://www.w3c.org/2000/xp/Group/>
- <http://www.oreilly.de/catalog/progwebservice/chapter/ch02.html>
- **Webservice-Programmierung mit SOAP**
James Snell, Doug Tidwell & Pavel Kulchenko
O'Reilly 1. Auflage, Juli 2002
ISBN 3-89721-159-9