

Web Services Soap

Vladyslav Kutsenko
Seminar Web services
Vortrag zum Thema Soap
uv08@stud.uni-karlsruhe.de

18. 05. 2004

Abstract

Teleseminar Web Services
Vortrag zum Thema Soap

Inhalt	1
Definition 1	2
Definition 2	3
Motivation für SOAP	4
Vorteile von SOAP	5
Nachteile von SOAP	6
Grundlagen	7
Elemente einer Soap-Nachricht	8
soap/Body	9
soap/Body Child elements	10
soap/Envelope	11
soap:Fault	12
Fehlercodes	13
soap:header	14
soap:header child elements	15
Actor Attribute	16
soap:EncodingStyle attribute	17
Mustunderstand attribute	18
Serialisationsregeln 1	19
Serialisationsregeln 2	20
Serialisationsregeln 3	21
Serialisierungsregeln: Beispiel	22
HTTP Binding 1	23
HTTP Binding 2	24
HTTP Binding: Beispiel	25

Definition 1

SOAP Version 1.2 (SOAP) is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. It uses XML technologies to define an extensible messaging framework providing a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

Quelle: W3C

Definition 2

The Simple Object Access Protocol (SOAP) is an XML messaging specification that describes a message format along with a set of serialization rules for datatypes including structured types and arrays. In addition, it describes how to use the Hypertext Transfer Protocol (HTTP) as a transport for such messages.

Quelle:

Essential XML

Aaron Skonnard

Martin Gudgin

Addison-Wessley

Motivation für SOAP

- Zusammenfügen von XML und Web.
- Erzeugung eines effektiven Protokols für die Kommunikation im Netzwerk.
- Interoperabilität verschiedener Systeme

Vorteile von SOAP

- Allgemein akzeptierte Standardisierung (W3C)
- Plattformunabhängigkeit (Voraussetzung für die Benutzung von SOAP: HTTP + XML)
- Offenheit (Offene Schnittstellen)
- Robustheit (HTTP-Protokol)

Nachteile von SOAP

- Overhead verursacht durch Verarbeitung von XML
- Performance Nachteile durch Benutzung von HTTP in Szenarien, die kurzen Verbindungszeiten erfordern
- Neue Sicherheitsherausforderung (Firewall Durchdringung)

Grundlagen

Die SOAP Spezifikation definiert:

- Das Root-Element Envelope, das ein XML-Dokument einschließt
- Beschreibt die Repräsentation der zusätzlichen Informationen wie z. B. Verweise auf die Schemata für das Dokument, die Transaktionsverwaltung, Sicherheit oder Applikationskoordination.
- Beschreibung der Regeln für die Serialisierung von Datentypen.
- Beschreibung der Bindung an HTTP-Protokol, damit die Daten fehlerfrei durchs Netzwerk zum Zielknoten und zum Soap-Prozessor übermittelt werden.

Elemente einer Soap-Nachricht

- Envelope (root element)
- Header (optional)
- Body

Diese Elemente sind im Namespace:

`http://schemas.xmlsoap.org/soap/envelope` definiert

soap/Body

```
<soap:Body xmlns:soap='http://sch... >
```

Nutzlast der Nachricht

```
</soap:Body>
```

soap/Body Child elements

- Ein oder mehr Namespace-qualified Elemente, die nicht in `http://schemas.xmlsoap.org/soap/envelope` definiert sind Bsp.:
...
`<m:getOrders
 xmlns:m='http://www.mywebshop.com//myschema' >`
...
● Fehlerelement im Falle eines Fehlers aus Namespace:
`http://schemas.xmlsoap.org/soap/envelope`

soap/Envelope

```
<soap:Envelope  
  xmlns:soap='http://schemas.xmlsoa... >
```

Optionale Kopfdaten (header)

Body

```
</soap:Envelope>
```

Durch soap/Envelope wird die XML-Nachricht als eine Soap-Nachricht identifiziert.

Child elements:

Optional Header und Body

soap:Fault

Zeigt an, daß bei der Anfragebearbeitung ein Fehler aufgetreten ist. Kommt nur in den Antwortnachrichten vor.

Child elements:

- Faultcode
definiert in:
`http://schemas.xmlsoap.org/soap/envelope/`
- Faultstring
Type: String. Enthält eine von Menschen lesbare Meldung
- Faultactor (optional)
Type: uri-reference. Zeigt die Fehlerquelle an. Z. B. Empfänger der Nachricht
- Detail (optional)
Type: Anwendungsspezifische Informationen zum Fehler. Kann beliebige Namespace-qualified Attribute enthalten.

Fehlercodes

Hier sind einige Beispiele der Fehler, die durch Faultcode angezeigt werden:

- VersionMismatch: Der Namespace des Envelope wird von Empfänger nicht akzeptiert. Die Nachricht soll korregiert wiederholt werden.
- Server: Serverseitige Probleme
- MustUnderstand: Ein Child Element des haeaders hat ein soap:mustunderstand Atribut, wird jedoch durch den Empfänger nicht verstanden
- client: Die Nachricht enthält nicht alle Elemente, die notwendig sind, damit die Verarbeitung durch den Empfänger erfolgen kann.
-

soap:header

Aufgabe: Container für die Benutzerdefinierten Erweiterungen.

Mögliche sinnvolle benutzerdefinierte Erweiterungen:

- Transaktionsverwaltung
- Authentifizierung
- Kryptologie

soap:header child elements

Beliebige Namespace-qualified Elemente, die nicht in

`http://schemas.xmlsoap.org/soap/envelope/`
definiert sind.

Beispiel:

```
...  
<header>  
<x:sprache xmlns:  
  x='http://www.example.com/extension'>  
<x:codepage>latin1</x:codepage>  
</x:sprache>  
</header>  
...
```

Actor Attribute

Syntax: `soap:actor='anyuri'`

Spezifiziert den URI des Zwischenknoten, der die mit diesem Attribut markierten Erweiterung im Header interpretieren soll.

- Wenn `uri=`
`http://schemas.xmlsoap.org/soap/actor/next` wird die Headererweiterung vom nächsten in der Kette Knoten interpretiert.
- wenn `soap:actor` nicht angegeben ist, wird die die Headererweiterung vom Zielknoten verarbeitet

soap:EncodingStyle attribute

Syntax:

```
soap:encodingStyle='List of anyuri'
```

Zeigt dem Empfänger einer Soap-Nachricht an, welches Serialisierungsformat benutzt wurde, um das Element sowie die zugehörigen Unterelemente zu serialisieren.

Kann an jedem Element einer Soap-Nachricht angewandt werden.

Die Unterelemente können diese Attribute überschreiben.

Mustunderstand attribute

Syntax:

`mustunderstand='1'` oder `mustunderstand='0'`

Funktion:

Zeigt an, ob eine Header-Erweiterung verarbeitet werden soll.

Per Default ist `mustunderstand='0'`. Wenn dieses Attribut auf `'1'` gesetzt ist, muß die Header-erweiterung verarbeitet werden. Oder ein Fault zurückgegeben werden.

Serialisationsregeln 1

Objekte und Strukturen der Programmiersprachen werden auf die SOAP Data Structs abgebildet, und es werden remote calls erzeugt, die die Aufrufe der Methoden der entsprechenden Objekte der höheren Programmiersprachen in ihren Signaturen abbilden

Die einfachen Datentypen werden den entsprechenden XML-Datentypen zugeordnet.

Serialisationsregeln 2

Die strukturierten Daten werden als XML-Elemente dargestellt.

Metadaten z. B. Referenzen als Attribute

Datentypen wie Structs und Arrays werden in:
<http://schemas.xmlsoap.org/soap/encoding> definiert.

Serialisationsregeln 3

Datentypen wie Strukturen oder Klassen:

Jedes Element wird serialisiert durch ein xml-Element mit dem gleichen Namen

Verschachtelung ist zugelassen. Z. B. Elemente eines Arrays sind nicht primitiv.

Wurzelemente der strukturierten Datentypen sind Namespace-qualified, die darunterliegenden dagegen nicht.

Die Serialisationsregeln betreffen die Unterelemente des Header- und Bodyelements

Serialisationsregeln: Beispiel

```
package example.org.People;
// Java class definition
class Person
{
    String name;
    float alter;
}
```

Serialisierung in SOAP:

```
...
<p:Person
  xmlns:p='http://www.mywebshop.com/staff'>
  <name>Martin</name>
  <alter>30</alter>
</p:Person>
...
```

HTTP Binding 1

Die SOAP Spezifikation beschreibt das Verhältnis zwischen SOAP Request/Response und HTTP-Headers. (HTTP-Binding)

Content-type:

Content-Type: text/xml; charset=?

Festlegung der Mime type der Nachricht und optional der Zeichenkodierung. Z. B. utf-8

HTTP Binding 2

content-length:

Zeigt die Anzahl der Bytes in Body der Request oder Response Nachricht.

SOAPACTION zeigt dem Server an, daß die vorliegende Nachricht ein SOAP-Request ist. Der Wert ist URI

HTTP Binding: Beispiel

```
POST /endpoint HTTP/1.1
Content-Type: text/xml;
    charset=UTF-16
Content-Length: 167
SOAPAction: http://www.mywebshop.com#Method
<env:Envelope
xmlns:
    env='http://schemas.xmlsoap.org/soap/envelope/'>
<env:Body>
<m:Method
    xmlns:m='http://www.mywebshop.com' />
</env:Body>
</env:Envelope>
```

Zusammenfassung

Wie in diesem Vortrag beschrieben wurde, spezifiziert SOAP mit SOAP envelope, Headers, Body und Fault Elementen spezielle XML-Dokumente für ein nachrichtenorientiertes Übertragungsmechanismus. Es wird außerdem ein XML-basiertes Typsystem definiert, das zur Serialisation der Datentypen benutzt wird. Anschließend wird die Bindung des SOAP-Protokolls an HTTP behandelt.