

Úvod do SOAP protokolu

(zdroj: <http://w3schools.com>)

SOAP je jednoduchý, na XML založený protokol, ktorý umožňuje aplikáciám vzájomne si vymieňať informácie cez HTTP protokol. Jednoduchšie povedané, SOAP je protokol pre prístup k webovej službe.

Čo je SOAP?

- SOAP - Simple Object Access Protocol
 - SOAP je komunikačný protokol určený pre **komunikovanie medzi aplikáciami**
 - SOAP je **formát pre posielanie správ**
 - SOAP je navrhnutý pre komunikovanie cez internet
 - SOAP je platformovo nezávislý
 - SOAP je jazykovo nezávislý
 - SOAP je **založený na XML**
 - SOAP je jednoduchý a rozšíriteľný
 - SOAP umožňuje obísť firewall-y
 - SOAP je W3C štandardom od 24. Júna 2003
-

Prečo používať SOAP?

Pre vytvárané aplikácie je dôležité aby mali umožnenú vzájomnú internetovú komunikáciu. Objekty viacerých dnešných aplikácií vzájomne komunikuje pomocou vzdialeného volania procedúr (RPC - Remote Procedure Calls), ako napr. DCOM alebo CORBA aplikácie, ale HTTP nebol navrhnutý pre takúto komunikáciu. RPC často prináša bezpečnostné problémy a problémy s kompatibilitou, pretože firewall-y a proxy servre štandardne blokujú takýto druh komunikácie.

Lepší spôsob komunikácie medzi aplikáciami je komunikácie cez HTTP protokol, pretože ten je podporovaný všetkými webovými prehliadačmi a servermi. SOAP bol vytvorený na uskutočnenie takejto komunikácie.

SOAP poskytuje spôsob komunikácie medzi aplikáciami spustenými v rôznych operačných systémoch, vyvinutými pomocou rôznych technológií a v rôznych programovacích jazykoch.

SOAP Syntax

SOAP stavebné bloky

SOAP správa je obyčajný XML obsahujúci nasledovné elementy:

- **Envelope** element, ktorý identifikuje XML dokument ako SOAP správu
- **Header** element, ktorý obsahuje hlavičkové informácie
- **Body** element, ktorý obsahuje volacie a odpovedacie informácie
- **Fault** element obsahujúci chyby a informácie o statusu správy

Všetky tieto elementy sú deklarované v defaultovom **priestore názvov** pre **SOAP obálku**:

<http://www.w3.org/2001/12/soap-envelope> (SOAP 1.1)

<http://www.w3.org/2003/05/soap-envelope> (SOAP 1.2)

a v defaultovom **priestore názvov** pre **SOAP kódovanie a dátové typy**

<http://www.w3.org/2001/12/soap-encoding>

SOAP syntaktické pravidlá

- SOAP správa musí byť kódovaná pomocou XML
 - SOAP správa musí používať SOAP obáľkový priestor názvov
 - SOAP správa musí používať priestore názvov pre SOAP kódovanie
 - SOAP správa nesmie obsahovať DTD referenciu
 - SOAP správa nesmie obsahovať XML spracovávajúce inštrukcie
-

Skelet SOAP správy

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>
    ...
    <soap:Fault>
    ...
    </soap:Fault>
  </soap:Body>

</soap:Envelope>
```

SOAP Envelope element

Povinný **SOAP Envelope element** je koreňovým elementom SOAP správy. Tento element **definuje XML dokument ako SOAP správu**.

Príklad SOAP Envelope elementu

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  ...
  Informácie o správe budú tu
  ...
</soap:Envelope>
```

Priestor názvov *xmlns:soap*

by mal mať vždy hodnotu (pre SOAP 1.1):

"http://www.w3.org/2001/12/soap-envelope".

Tento priestor názvov definuje Envelope element ako **SOAP Envelope**. Ak je použitý iný priestor názvov, potom aplikácia generuje chybu a správu zahodí.

***encodingStyle* atribút**

sa používa na definovanie dátových typov použitých v dokumente. Tento atribút sa **môže** objaviť v ktoromkoľvek SOAP elemente, potom sa vzťahuje na obsah tohto elementu a na všetky jeho detské elementy.

SOAP správa nemá defaultové kódovanie.

Syntax

soap:encodingStyle="URI"

Príklad

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  ...
  Informácie o správe budú tu
  ...
</soap:Envelope>
```

SOAP Header element

Voliteľný SOAP Header element obsahuje aplikačne špecifické informácie (ako napr. autentifikácia, platba atď.) o SOAP správe.

Ak je Header element uvedený, musí byť prvým detským elementom Envelope elementu.

Všetky priame detské elementy Header elementu musia byť kvalifikované priestorom názvov.

Príklad

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
    <m:Trans xmlns:m="http://www.w3schools.com/transaction/" soap:mustUnderstand="1">234
    </m:Trans>
  </soap:Header>
  ...
  ...
</soap:Envelope>
```

Príklad obsahuje hlavičku s "Trans" elementom s hodnotou 234, ktorý má atribút "mustUnderstand" s hodnotou 1.

SOAP definuje 3 atribúty v default-ovom priestore názvov "http://www.w3.org/2001/12/soap-envelope" - **mustUnderstand**, **actor** a **encodingStyle**.

Atribúty definované v SOAP Header elemente definujú ako by mal príjemca spracovať SOAP správu.

mustUnderstand atribút

môže byť použitý na indikovanie toho, či hlavičkový dáta sú povinné alebo nepovinné na spracovanie pre príjemcu.

Ak pridáme atribút mustUnderstand="1" do detského elementu Header element, tým indikujeme, že príjemcove spracovanie Header elementu musí rozpoznať tento detský element. Ak príjemca nerozpozna tento detský element, tak potom zlyhá spracovanie Header elementu.

Syntax

soap:mustUnderstand="0|1"

Príklad

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
    <m:Trans xmlns:m="http://www.w3schools.com/transaction/" soap:mustUnderstand="1">234
    </m:Trans>
  </soap:Header>
  ...
  ...
</soap:Envelope>
```

actor atribút

SOAP správa môže počas "cestovania" od odosielateľa k príjemcovi prechádzať rôznymi koncovými bodmi. Nie všetky časti SOAP správy môžu byť určené pre konečný koncový bod. Niektoré časti tejto SOAP správy môžu byť určené pre jeden alebo viacero koncových bodov nachádzajúcich sa na ceste správy. SOAP **actor** atribút sa používa na adresovanie Header elementu pre špecifický koncový bod.

Syntax

soap:actor="URI"

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
    <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
      soap:actor="http://www.w3schools.com/appml/">234
    </m:Trans>
  </soap:Header>
  ...
  ...
</soap:Envelope>
```

SOAP Body element

Povinný SOAP Body element obsahuje samotnú SOAP správu určenú pre konečný koncový bod tejto správy.

Priamo podriadené detské elementy SOAP Body elementu môžu byť kvalifikované priestorom názvov.

Príklad

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body>
    <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>
```

</soap:Envelope>

Tento príklad SOAP správy požaduje od metódy webovej služby cenu jablka. Elementy **m:GetPrice** a **m:Item** sú aplikačne špecifické elementy. Nie sú súčasťou SOAP priestoru názvov. SOAP odpoveď na túto požiadavku môže vyzeráť nasledovne:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body>
    <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
      <m:Price>1.90</m:Price>
    </m:GetPriceResponse>
  </soap:Body>

</soap:Envelope>
```

SOAP Fault element

obsahuje chyby a informácie o statuse SOAP správy.

Voliteľný SOAP Fault element sa používa na indikovanie chybových správ.

Ak je Fault element uvedený, tak sa musí objaviť ako detský element Body elementu. Fault element sa môže **objaviť len 1-krát** v SOAP správe.

SOAP Fault element má nasledujúce subelementy (tabuľka v angl. jazyku z <http://w3schools.com>):

Sub Element	Description
<faultcode>	A code for identifying the fault
<faultstring>	A human readable explanation of the fault
<faultactor>	Information about who caused the fault to happen
<detail>	Holds application specific error information related to the Body element

SOAP Fault kódy

Hodnoty SOAP Fault kódov musia byť použité vo faultcode elemente, pretože popisujú zlyhanie. Tabuľka hodnotami SOAP Fault kódov (v angl. jazyku z <http://w3schools.com>):

Error	Description
VersionMismatch	Found an invalid namespace for the SOAP Envelope element
MustUnderstand	An immediate child element of the Header element, with the mustUnderstand attribute set to "1", was not understood
Client	The message was incorrectly formed or contained incorrect information
Server	There was a problem with the server so the message could not proceed

SOAP HTTP väzba

HTTP protokol

HTTP protokol komunikuje cez TCP/IP. HTTP klient sa pripája na HTTP server pomocou TCP protokolu. Po vytvorení spojenia klient môže poslať HTTP požiadavkovú správu na server:

```
POST /item HTTP/1.1
Host: 189.123.345.239
Content-Type: text/plain
Content-Length: 200
```

Server potom spracuje požiadavku a pošle HTTP odpoveď späť klientovi. Odpoveď obsahuje kód stavu (status code), ktorý indikuje stav spracovania požiadavky:

```
200 OK
Content-Type: text/plain
Content-Length: 200
```

V tomto príklade server vrátil stavový kód 200, čo znamená, že HTTP požiadavka bola spracovaná úspešne.

Ak by server nemohol dekodovať požiadavku, potom by mohol vrátiť nasledovnú odpoveď:

```
400 Bad Request
Content-Length: 0
```

SOAP HTTP väzba

SOAP metóda je http požiadavka/odpoveď, ktorá dodržiava pravidlá SOAP kódovania.

HTTP + XML = SOAP

SOAP požiadavka môže byť typu HTTP POST alebo HTTP GET požiadavka.

HTTP POST požiadavka špecifikuje najmenej dve HTTP hlavičky: **Content-Type** a **Content-Length**

Content-Type

Content-Type hlavička definuje pre SOAP požiadavku a odpoveď MIME typ správy a kódovanie znakov (voliteľné) použité v XML tele požiadavky alebo odpovede.

Syntax

Content-Type: **MIMEType**; charset=**character-encoding**

Príklad

POST /item HTTP/1.1

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

Content-Length

Content-Length hlavička špecifikuje počet bytov v tele SOAP požiadavky alebo odpovede.

Syntax

Content-Length: **bytes**

Príklad

POST /item HTTP/1.1

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

Content-Length: **250**

Príklad SOAP požiadavky a odpovede

V nasledujúcom príklade je zaslaná požiadavka **GetStockPrice** na server. Požiadavka má **StockName** parameter a **Price** parameter, ktorý bude vrátený v odpovedi servera. Priestor názvov metódy je definovaný v "http://www.example.org/stock".

SOAP požiadavka:

POST /InStock HTTP/1.1

Host: www.example.org

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

Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope

xmlns:soap="http://www.w3.org/2001/12/soap-envelope"

soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">

<m:**GetStockPrice**>

<m:**StockName**>IBM</m:**StockName**> //parameter 'StockName' požiadavky

</m:**GetStockPrice**>

</soap:Body>

</soap:Envelope>

SOAP odpoveď

HTTP/1.1 200 OK

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

Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope

xmlns:soap="http://www.w3.org/2001/12/soap-envelope"

soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">

<m:GetStockPriceResponse>

<m:Price>34.5</m:Price> //návratový parameter 'Price' vratený v odpovedi

</m:GetStockPriceResponse>

</soap:Body>

</soap:Envelope>