

Sistemas Web III

SOAP

Simple Object Access Protocol

Tiago Cruz de França

tcruz.franca@gmail.com

O que você precisa saber...

@ XML

@ Se não sabe, esta é uma boa referência:

▶ <http://www.w3schools.com/xml/default.asp>

@ SOAP independente de implementação

@ Apesar de implementarmos em Java e usarmos *frameworks* para fazer parte do trabalho chato

▶ Montar as mensagens contendo o que desejamos

▶ Descritor WSDL*

— Falaremos sobre WSDL neste material

▶ Possibilitar a criação de parte da aplicação cliente automaticamente

SOAP

@ Um protocolo para acesso a Web Services

- @ Baseado em XML

@ Troca de mensagens entre aplicações

- @ Independente de tecnologias usadas na implementação

- @ Mensagem padronizada em formato XML

@ Recomendação da W3C

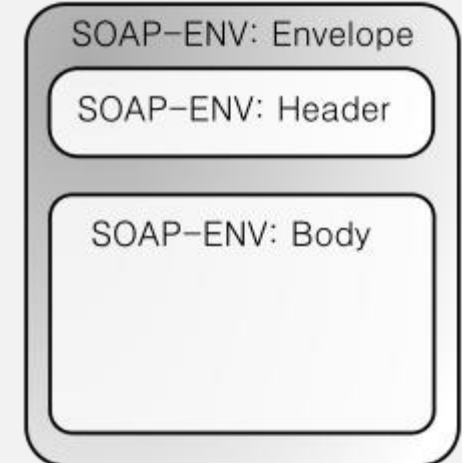
A Mensagem SOAP

@ A mensagem é um documento XML contendo:

- @ Uma tag de **envelope** que identifica a mensagem SOAP
- @ Um elemento **cabeçalho** com algumas informações
- @ Um **corpo** com informações de chamada e resposta
- @ Um elemento “**fault**” com informações de status e erro

Skeleton de uma Mensagem SOAP

```
<?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>
```



Principais Regras de Sintaxe

@ Uma mensagem SOAP deve:

- @ Ser codificada com XML
- @ Usar o Envelope *namespace* SOAP
 - ▶ Endereço: <http://www.w3.org/2001/12/soap-envelope>
- @ Usar o Encoding *namespace* SOAP
 - ▶ Endereço: <http://www.w3.org/2001/12/soap-encoding>

@ Uma mensagem SOAP Não deve:

- @ Ter uma referência a um DTD (*Document Type Definition*)
- @ Não deve possuir instruções de processamento XML

Entendendo a Mensagem SOAP

O Elemento Envelope

@ Elemento raiz da mensagem SOAP

```
<?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">
  ...
  Message information goes here
  ...
</soap:Envelope>
```

@ xmlns:soap

- ▶ Sempre indicar o mesmo *namespace*. Ele define o envelope
- ▶ Indicar um *namespace* diferente irá gerar um erro

O Elemento Envelope

@ Elemento raiz da mensagem SOAP²

```
<?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">
  ...
  Message information goes here
  ...
</soap:Envelope>
```

@ Atributo **encodingStyle**

- ▶ Define o tipo de dado usado no documento
 - Pode aparecer em qualquer elemento da mensagem SOAP e se aplica ao elemento e aos seus filhos
- ▶ Não existe codificação padrão

Elemento SOAP Header da mensagem SOAP

Ⓜ É opcional. Ele contém informações da aplicação

Ⓜ Se presente, deve ser o primeiro elemento do envelope

```
<?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">
<!-- os atributos definem como a mensagem soap deve ser processada -->
<soap:Header>
  <!-- elemento "Trans", um atributo o "mustUnderstand" e o valor "123"-->
  <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
soap:mustUnderstand="1">234
<!-- mustUnderstand define a entrada de cabeçalho é obrigatório ou opcional -->
  </m:Trans>
</soap:Header>
...
</soap:Envelope>
```

Elemento SOAP Header da mensagem SOAP

Ⓢ É opcional. Ele contém informações da aplicação

Ⓢ Se presente, deve ser o primeiro elemento do envelope

```
<?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>
    <!-- elemento "Trans", agora com atributo "actor" e valor "234"-->
    <soap:Header>
      <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
        soap:actor="http://www.w3schools.com/appml/">234
      </m:Trans>
    <!-- actor endereça o cabeçalho para um destino específico -->
    </soap:Header>
    ...
  </soap:Envelope>
```

O elemento Body

@ Requisição (subscribe)

@ O exemplo requer o preço (price) de maçãs (apples)

```
<?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">
<!-- o primeiro elemento do body deve apontar para o namespace da msg -->
<soap:Body>
  <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
    <m:Item>Apples</m:Item>
  </m:GetPrice>
</soap:Body>
<!-- m:GetPrice e o elemento Item são específicos da aplicação -->
</soap:Envelope>
```

O elemento Body

@ A reposta a requisição do exemplo anterior

```
<?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>
```

Informações sobre Erros e Status

Erros e Informações de Status das Mensagens SOAP

@ O elemento “fault” é opcional

- @ Usado para indicar erro de mensagem
- @ Quando um erro ocorre e o “fault” é apresentado, ele deve ser um elemento filho do Body
 - ▶ Só pode aparecer uma vez em uma mensagem SOAP

Sub Elemento	Descrição
<faultcode>	Código de identificação da falha
<faultstring>	Uma explicação de alto nível legível para humanos
<faultactor>	Informação sobre qual elemento causou a falha
<detail>	Contém informações específicas do erro relacionadas ao elemento Body

Códigos de Falha

@ Devem ser usados no elemento fault

Erro	Descrição
VersionMismatch	Namespace inválido para o Envelope da mensagem SOAP
MustUnderstand	Um elemento obrigatório do cabeçalho (mustUnderstand=1) não foi entendido
Client	Mensagem mal formada ou com erro em alguma informação
Server	Problemas no servidor impediram o processamento da mensagem

BIND

Antes do BINDING, vamos rever o HTTP

@ Por que?

- @ Vamos entender melhor nos próximos slides, por enquanto basta você saber que as mensagens SOAP são comumente enviadas dentro (no corpo) de mensagens HTTP
 - ▶ Mensagens de requisição e de resposta
- @ Veremos que outro protocolo poderia ser utilizado
 - ▶ Reforçando, o HTTP é comumente utilizado
 - Geralmente o argumento é que o HTTP (sua porta) permanece aberta na maioria dos firewalls

SOAP Binding com HTTP

@ Cliente estabelece conexão TCP para depois enviar mensagem HTTP

@ Cabeçalho da Requisição HTTP

```
POST /item HTTP/1.1
Host: 189.123.255.239
Content-Type: text/plain
Content-Length: 200
```

@ Resposta do servidor após processar a requisição

▶ Código de status
do processamento da
requisição

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

SOAP Binding com HTTP

@ Cliente estabelece conexão TCP para depois enviar mensagem HTTP

@ Cabeçalho da Requisição HTTP

```
POST /item HTTP/1.1  
Host: 189.123.255.239  
Content-Type: text/plain  
Content-Length: 200
```

@ Resposta do servidor após processar a requisição

▶ Código de status
do processamento da
requisição

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

SOAP Binding

- Ⓢ **A especificação SOAP não define como a mensagem vai ser trocada entre servidor e clientes**
 - Ⓢ SOAP Binding são os mecanismos para essa troca de mensagens
 - Ⓢ As mensagens podem ser enviadas em qualquer protocolo como o HTTP ou SMTP
 - ▶ **O HTTP é o mais usado (é síncrono)**
 - A requisição, como vimos no exemplo anterior, possui dois atributos: Content-Type e Content-Length
 - ▶ **O SMTP é uma opção menos utilizada (é assíncrono)**
 - Usada apenas em casos específicos

Conteúdo da Mensagem de Binding com mensagem SOAP

@ Content-Type

- Define o tipo MIME e, opcionalmente, a codificação da mensagem

▶ **Sintaxe:** `Content-Type: MIMEType; charset=character-encoding`

▶ **Exemplo:** `POST /item HTTP/1.1`
`Content-Type: application/soap+xml; charset=utf-8`

@ Content-Length

- Especifica o num. de bytes na requisição e resposta

▶ **Sintaxe:** `Content-Length: bytes`

▶ **Exemplo:** `POST /item HTTP/1.1`
`Content-Type: application/soap+xml; charset=utf-8`
`Content-Length: 250`

Exemplo Requisição SOAP

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>

</m:GetStockPrice>

</soap:Body>

</soap:Envelope>

Exemplo Reposta SOAP

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>

</m:GetStockPriceResponse>

</soap:Body>

</soap:Envelope>