

---

# Web Services Definition Language WSDL

Exemplos: SoapHeaders, SoapExceptions

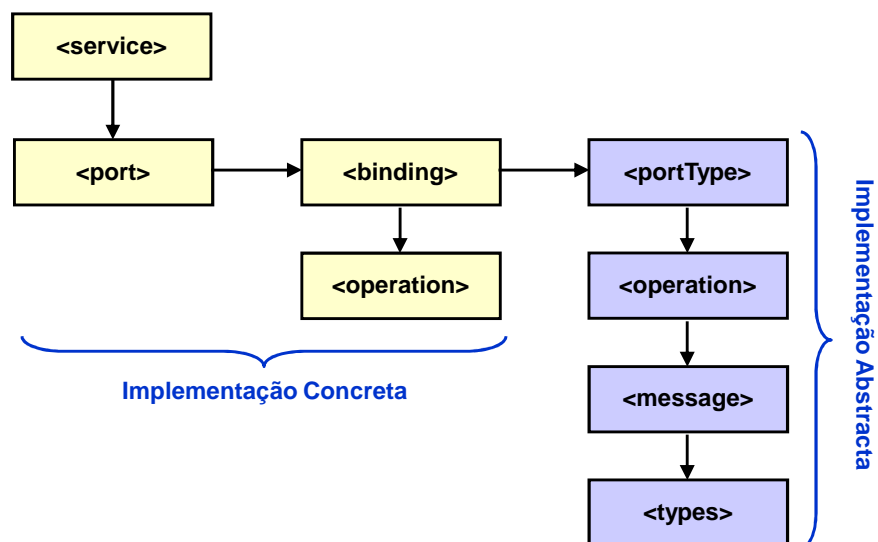
## Web Services Definition Language

- Formato de um documento para descrever com precisão *Web Services*;
- Especifica o formato exacto das mensagens, protocolo usado, endereço de localização, isto é, o WSDL define a Interface/contrato do Serviço;
- Por outras palavras o WSDL diz-nos como podemos usar o *Web Service*;
- O WSDL é normalmente gerado por *tools* a partir da implementação do *Web Service* (Em .NET: tool <disco> ou Add Web Reference no Visual Studio)
- Normalmente existem também *tools* que a partir do ficheiro WSDL permitem gerar *proxys* ou *stubs*, através dos quais o cliente acede ao serviço (Em .NET: tool <wsdl> ou Add Web Reference no Visual Studio);
- Um documento WSDL é um documento XML conforme com um *XML Schema standard*:
  - “<http://schemas.xmlsoap.org/wsdl/>”

## Elementos principais do WSDL Schema

- Elemento **<definitions>** - Elemento *xml root* do documento WSDL e onde são definidos os *namespaces xml* usados ao longo do documento;
- Elemento **<types>** - Utiliza *XML schema* para declarar tipos de dados e outros elementos usados no documento WSDL;
- Elemento **<message>** - Descreve as mensagens usando *XML schema* com os tipos de dados associados *built-in* ou *complex types* ou outros elementos definidos em **<types>**
- Elemento **<portType>** - Descreve as operações (interface do *Web Service*). Um elemento **<operation>** usa uma ou mais mensagens (definidas em **<message>**) para definir os parâmetros de *input* e *output*;
- Elemento **<binding>** - Associa o **<portType>** e respectivas operações com um protocolo, por exemplo, SOAP 1.1 e estilos de codificação (*Encoded* ou *Literal*) e, no caso em que existam, a especificação de *Soap Headers* nas operações;
- Elemento **<service>** - Associa um *Internet Address* para um **<binding>** específico

## Estrutura de um documento WSDL



## Exemplo de um Web Service – Tipos de dados Usados

**[System.Xml.Serialization.XmlInclude(typeof(PTAddress))]**

```
public class PTAddress {  
    public string nome;  
    public string rua;  
    public int[] numero;  
    public string cidade;  
    public string codigopostal;  
}
```

O atributo *xmlInclude* indica que estes tipos sejam considerados no documento WSDL

**[System.Xml.Serialization.XmlInclude(typeof(Book))]**

```
public class Book {  
    public string isbn;  
    public string title;  
    public int quantity;  
    public float price;  
}
```

**[System.Xml.Serialization.XmlInclude(typeof(PurchaseOrder))]**

```
public class PurchaseOrder {  
    public string accountName;  
    public int accountNumber;  
    public PTAddress paddress;  
    public Book itemBook;  
}
```

## Classe Web Service

**[WebService (Description="Exemplo de WS de cotação de Livros",  
Namespace="http://DEETC.SES.SD")]**

```
public class BookService : System.Web.Services.WebService {
```

**[WebMethod]**

```
    public string getISBN(string title) {  
        return "1-123-12345-9";  
    }
```

**[WebMethod]**

```
    public float getBookPrice(string isbn) {  
        return 24.9F;  
    }
```

**[WebMethod]**

```
    public string submitPOrder(PurchaseOrder po) {  
        return "OK: "+po.accountName+" : "+po.accountNumber;  
    }  
}
```

## Interface de um cliente

Cliente BookService WSDL

Title: Qualquer Titulo [getISBN] ISBN: 1-123-12345-9

ISBN: 1-123-12345-1 [getPrice] Price: 24,9

PO name: ISEL-DEETC [submitPOOrder] Resultado: OK: ISEL-DEETC : 123

PO Number: 1234 ( . . )

ISEL/DEETC - Sistemas Distribuídos

7

## Estrutura Básica do documento WSDL

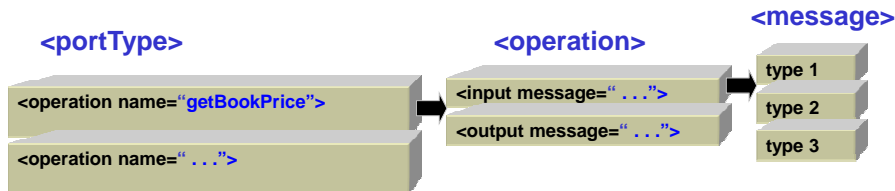
```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://DEETC.SES.SD"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  targetNamespace="http://DEETC.SES.SD"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified"
      targetNamespace="http://DEETC.SES.SD">
    </s:schema>
  </wsdl:types>
  <wsdl:message name="getBookPriceSoapIn"> ... </wsdl:message>
  <wsdl:portType name="BookServiceSoap"> ... </wsdl:portType>
  <wsdl:binding name="BookServiceSoap" type="tns:BookServiceSoap"></wsdl:binding>
  <wsdl:service name="BookService">
    <wsdl:port name="BookServiceSoap" binding="tns:BookServiceSoap">
      <soap:address location="http://localhost:1077/BookService/BookService.asmx" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

ISEL/DEETC - Sistemas Distribuídos

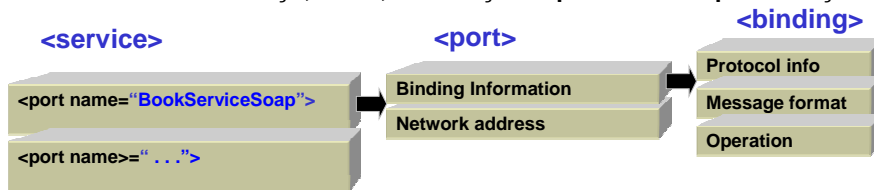
8

## Relações entre elementos em WSDL

**Service Interface Definition** – contém elementos abstractos que são usados para descrever o serviço de maneira geral

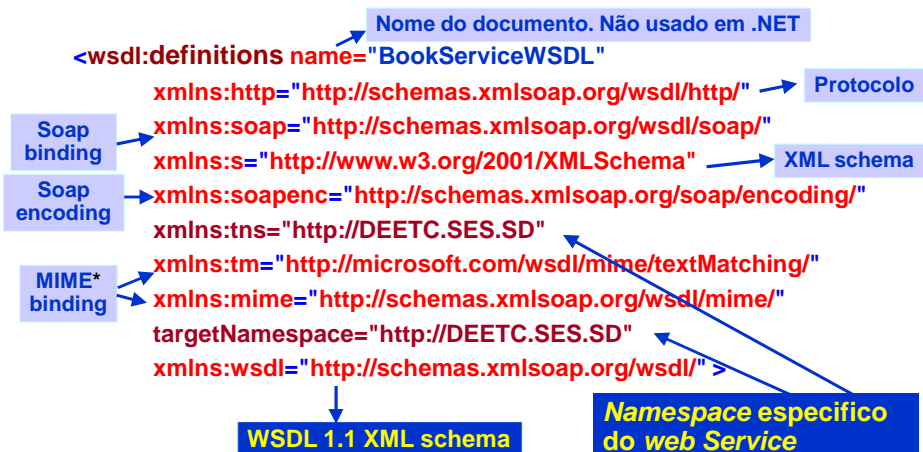


**Service Implementation Definition** – contém elementos usados para definir uma instância do serviço, isto é, informação específica de implementação.



## <definitions>

- Elemento *root* do documento WSDL, onde é definido o conjunto de *XML namespaces* necessários e opcionalmente um nome para o documento;



\*MIME - Multipurpose Internet Mail Extensions – protocolo que permite troca de mensagens com conteúdos de vários tipos (texto, áudio, vídeo, imagens, etc.).

## <types>

- Utiliza o **XML schema** para descrever os tipos de dados usados no **Web Service**;

```
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://DEETC.SES.SD">
    <s:element name="getISBN">...
    <s:element name="getISBNResponse">...
    <s:element name="getBookPrice">...
    <s:element name="getBookPriceResponse">...
    <s:element name="submitPOOrder">...
    <s:complexType name="PurchaseOrder">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="accountName" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="accountNumber" type="s:int" />
        <s:element minOccurs="0" maxOccurs="1" name="paddress" type="tns:PTAddress" />
        <s:element minOccurs="0" maxOccurs="1" name="itemBook" type="tns:Book" />
      </s:sequence>
    </s:complexType>
    <s:complexType name="PTAddress">...
    <s:complexType name="ArrayOfInt">...
    <s:complexType name="Book">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="isbn" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="title" type="s:string" />
        <s:element minOccurs="1" maxOccurs="1" name="quantity" type="s:int" />
        <s:element minOccurs="1" maxOccurs="1" name="price" type="s:float" />
      </s:sequence>
    </s:complexType>
    <s:element name="submitPOOrderResponse">...
  </s:schema>
</wsdl:types>
```

Tipos associados às mensagens

Tipos complexos:  
• PurchaseOrder;  
• PTAddress;  
• Book

ISEL/DEETC - Sistemas Distribuídos

11

## <message>

- Descreve as mensagens de **input** e **output** (parâmetros) das operações e eventuais **SOAP Header Blocks**;
- Os elementos **<message>**, **<portType>** e **<operations>** descrevem a interface abstracta do **Web Service**

```
<wsdl:types>...
<wsdl:message name="getISBNSoapIn">
  <wsdl:part name="parameters" element="tns:getISBN" />
</wsdl:message>
<wsdl:message name="getISBNSoapOut">
  <wsdl:part name="parameters" element="tns:getISBNResponse" />
</wsdl:message>
<wsdl:message name="getBookPriceSoapIn">
  <wsdl:part name="parameters" element="tns:getBookPrice" />
</wsdl:message>
<wsdl:message name="getBookPriceSoapOut">
  <wsdl:part name="parameters" element="tns:getBookPriceResponse" />
</wsdl:message>
<wsdl:message name="submitPOOrderSoapIn">
  <wsdl:part name="parameters" element="tns:submitPOOrder" />
</wsdl:message>
<wsdl:message name="submitPOOrderSoapOut">
  <wsdl:part name="parameters" element="tns:submitPOOrderResponse" />
</wsdl:message>
<wsdl:portType name="BookServiceSoap">...
<wsdl:binding name="BookServiceSoap" type="tns:BookServiceSoap">...
<wsdl:binding name="BookServiceSoap12" type="tns:BookServiceSoap">...
<wsdl:service name="BookService">...
```

ISEL/DEETC - Sistemas Distribuídos

12

## **<message> - Modo *RPC/Encoded***

- Tal como vimos em SOAP as mensagens *RPC Encoded*, têm um formato diferente;

- No caso de usarmos *RPC/Encoded* em .NET:

```
[WebService(Namespace="http://DEETC.SES.SD")]
[System.Web.Services.Protocols.SoapRpcServiceAttribute]
public class BookService : System.Web.Services.WebService { ....}

<wsdl:message name="getISBNSoapIn">
  <wsdl:part name="title" type="s:string" />
</wsdl:message>
<wsdl:message name="getISBNSoapOut">
  <wsdl:part name="getISBNResult" type="s:string" />
</wsdl:message>
<wsdl:message name="getBookPriceSoapIn">
  <wsdl:part name="isbn" type="s:string" />
</wsdl:message>
<wsdl:message name="getBookPriceSoapOut">
  <wsdl:part name="getBookPriceResult" type="s:float" />
</wsdl:message>
<wsdl:message name="submitPOOrderSoapIn">
  <wsdl:part name="po" type="tns:PurchaseOrder" />
</wsdl:message>
<wsdl:message name="submitPOOrderSoapOut">
  <wsdl:part name="submitPOOrderResult" type="s:string" />
</wsdl:message>
```

indicação de tipos primitivos  
nas mensagens

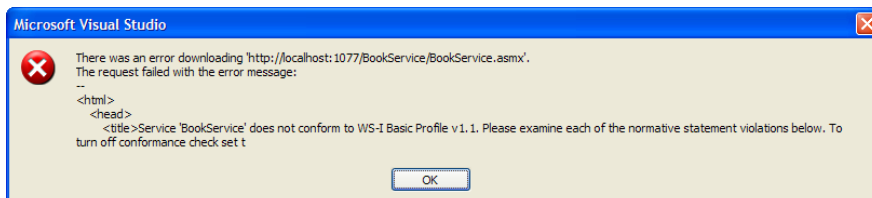
## ***RPC/Encoded e o WS-I Basic Profile 1.1***

- O Visual Studio 2005/2008/2010 cria *Web Services* por omissão conformes com as especificações *WS-I Basic Profile 1.1*;

```
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
```

```
public class BookService : System.Web.Services.WebService
```

- O *WS-I Basic Profile 1.1* não considera o modo *RPC/Encoded* pelo que deve ser evitado este modo;
- Ao realizar no Cliente <add Web Reference> obtemos a seguinte mensagem de erro



## <portType>

- O elemento **<portType>** define a interface do Serviço. Os elementos **<operation>** definem os métodos da interface.
- Um documento WSDL pode ter mais que um elemento **<portType>**, isto é, mais que uma interface;

```
<wsdl:message name="getISBNSoapIn">...</wsdl:message>
<wsdl:message name="getISBNSoapOut">...</wsdl:message>
<wsdl:message name="getBookPriceSoapIn">...</wsdl:message>
<wsdl:message name="getBookPriceSoapOut">...</wsdl:message>
<wsdl:message name="submitPOrderSoapIn">...</wsdl:message>
<wsdl:message name="submitPOrderSoapOut">...</wsdl:message>
<wsdl:portType name="BookServiceSoap">
  <wsdl:operation name="getISBN">
    <wsdl:input message="tns:getISBNSoapIn" />
    <wsdl:output message="tns:getISBNSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="getBookPrice">
    <wsdl:input message="tns:getBookPriceSoapIn" />
    <wsdl:output message="tns:getBookPriceSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="submitPOrder">
    <wsdl:input message="tns:submitPOrderSoapIn" />
    <wsdl:output message="tns:submitPOrderSoapOut" />
  </wsdl:operation>
</wsdl:portType>
```

O Web Service tem as operações:

- *getISBN*
- *getBookPrice*
- *submitPOrder*

Com mensagens de *input* e *output*

## Relação entre *operation* – *message* - *types*

```
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://DEETC.SES.SP">
    <s:element name="getISBN">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="title" type="s:string" />
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="getISBNResponse">...</s:element>
    <s:element name="getBookPrice">...</s:element>
    <s:element name="getBookPriceResponse">...</s:element>
    <s:element name="submitPOrder">...</s:element>
    <s:complexType name="PurchaseOrder">...</s:complexType>
    <s:complexType name="PTAddress">...</s:complexType>
    <s:complexType name="ArrayOfInt">...</s:complexType>
    <s:complexType name="Book">...</s:complexType>
    <s:element name="submitPOrderResponse">...</s:element>
  </s:schema>
</wsdl:types>
<wsdl:message name="getISBNSoapIn">
  <wsdl:part name="parameters" element="tns:getISBN" />
</wsdl:message>
<wsdl:message name="getISBNSoapOut">...</wsdl:message>
<wsdl:message name="getBookPriceSoapIn">...</wsdl:message>
<wsdl:message name="getBookPriceSoapOut">...</wsdl:message>
<wsdl:message name="submitPOrderSoapIn">...</wsdl:message>
<wsdl:message name="submitPOrderSoapOut">...</wsdl:message>
<wsdl:portType name="BookServiceSoap">
  <wsdl:operation name="getISBN">
    <wsdl:input message="tns:getISBNSoapIn" />
    <wsdl:output message="tns:getISBNSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="getBookPrice">...</wsdl:operation>
</wsdl:portType>
```

```
public interface BookService {
  getISBNSoapOut getISBN(string title);
  // ...
}
```

Como veremos *BookService* está definido no elemento **<service>**



## <binding>

- Associa um <portType> para um conjunto de protocolos concretos, por exemplo, SOAP e HTTP, modo da mensagem (RPC ou Document) e estilo de codificação (Literal ou SOAP Encoding);

```
<wsdl:portType name="BookServiceSoap">
  <wsdl:operation name="getISBN">
    <wsdl:input message="tns:getISBNSoapIn" />
    <wsdl:output message="tns:getISBNSoapOut" />
  </wsdl:operation>
  <wsdl:operation name="getBookPrice">...
  <wsdl:operation name="submitPOrder">...
</wsdl:portType>
<wsdl:binding name="BookServiceSoap" type="tns:BookServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="getISBN">
    <soap:operation soapAction="http://DEETC.SES.SD/getISBN" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getBookPrice">...
  <wsdl:operation name="submitPOrder">...
</wsdl:binding>
```

A operação **getISBN** usa protocolo SOAP em HTTP em modo **document** e estilo **literal**

## <binding> RPC Encoding

```
<wsdl:binding name="BookServiceSoap" type="tns:BookServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="rpc" />
  <wsdl:operation name="getISBN">
    <soap:operation soapAction="http://DEETC.SES.SD/getISBN" style="rpc" />
    <wsdl:input>
      <soap:body use="encoded" namespace="http://DEETC.SES.SD"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="http://DEETC.SES.SD"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="getBookPrice">...
  <wsdl:operation name="submitPOrder">...
</wsdl:binding>
```

## <service>

- Contém um ou mais elementos <port> em cada um representa um *Web Service*;
- O elemento <port> atribui um URL a um <binding> específico;

```
<wsdl:portType name="BookServiceSoap">
  <wsdl:operation name="getISBN">...
  <wsdl:operation name="getBookPrice">...
  <wsdl:operation name="submitPOrder">...
</wsdl:portType>
<wsdl:binding name="BookServiceSoap" type="tns:BookServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="getISBN">...
  <wsdl:operation name="getBookPrice">...
  <wsdl:operation name="submitPOrder">...
</wsdl:binding>
<wsdl:binding name="BookServiceSoap12" type="tns:BookServiceSoap">...
<wsdl:service name="BookService">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Exemplo de WS de
  <wsdl:port name="BookServiceSoap" binding="tns:BookServiceSoap">
    <soap:address location="http://localhost:1077/BookService/BookService.asmx" />
  </wsdl:port>
  <wsdl:port name="BookServiceSoap12" binding="tns:BookServiceSoap12">
    <soap12:address location="http://localhost:1077/BookService/BookService.asmx" />
  </wsdl:port>
</wsdl:service>
```

O *Web Service* com nome "BookService" tem um *binding* "BookServiceSoap" e está localizado em "http://localhost:1077/BookService/BookService.asmx"

## <import>

- O *schema WSDL* pode ainda ter um elemento <import>, similar ao usado em *XML schema* e que é usado para importar definições a partir de outro documento WSDL

```
<definitions name="AllWebservices"
  xmlns="http://schemas.xmlsoap.org/wsdl/" >
  <import namespace="http://DEETC.SES.SD/BookService"
    location="http://localhost:1077/BookService/BookService.wsdl" />
  <import namespace="http://DEETC.SES.ASD/Soma"
    location="http://localhost/Soma/Soma.wsdl" />

  <!-- outros Web Services -->
</definitions>
```

## <documentation>

- Em qualquer elemento do schema WSDL é possível colocar elementos **<documentation>** para comentar o documento WSDL;

```
<wsdl:service name="BookService">
  <documentation xmlns="http://schemas.xmlsoap.org/wsdl/">
    Exemplo de WS de cotação de Livros
  </documentation>

  <!-- ..... -->

</wsdl:service>
```

## Soap Headers em WSDL

```
namespace SoapHeaderLogin {
  public class TimeStamp {
    public DateTime dtIn;
    public DateTime dtOut;
  }
  public class SoapHdLogin : System.Web.Services.Protocols.SoapHeader {
    public string UserName;
    public string TokenKey;
    public TimeStamp ts;
  }
  public class SoapHdMoreInfo : System.Web.Services.Protocols.SoapHeader {
    public string info;
  }
}
```

## Web Service

```
namespace SoapHeaders {  
    [WebService(Description="SoapHeaders", Namespace="http://deetc.isel.ipl.pt/ASD/")]  
    public class WSLLogin : System.Web.Services.WebService {  
        public SoapHdLogin CurrentUser;  
        public SoapHdMoreInfo MoreInfo=new SoapHdMoreInfo();  
  
        [WebMethod, SoapHeader("CurrentUser", Direction=SoapHeaderDirection.InOut)]  
        [SoapHeader("MoreInfo", Direction=SoapHeaderDirection.Out)]  
        public string Login(string msg) {  
            CurrentUser.ts.dtOut=DateTime.Now;  
            MoreInfo.info="Informação out do WS";  
            if (CurrentUser.UserName == "ISEL" && CurrentUser.TokenKey == "12345")  
                return "Login Efectuado " + msg;  
            else {  
                CurrentUser.UserName="?????"; CurrentUser.TokenKey="-----";  
                return "Login não Efectuado " + msg;  
            }  
        }  
    }  
}
```

## Código do cliente

```
private void button1_Click(object sender, System.EventArgs e) {  
    WSLLogin wslprx = new WSLLogin();  
    SoapHdLogin soaphdl = new SoapHdLogin();  
    soaphdl.UserName=textBox1.Text; soaphdl.TokenKey=textBox2.Text;  
    soaphdl.ts=new Timestamp();  
    soaphdl.ts.dtIn=DateTime.Now;  
    wslprx.SoapHdLoginValue=soaphdl;  
    wslprx.SoapHdMoreInfoValue=new SoapHdMoreInfo(); // Não é necessário dado  
                                                    // que só no retorno temos valor do Header de output  
  
    try {  
        textBox3.Text=wslprx.Login(textBox1.Text);  
        textBox1.Text=wslprx.SoapHdLoginValue.UserName;  
        textBox2.Text=wslprx.SoapHdLoginValue.TokenKey;  
        textBox5.Text=wslprx.SoapHdLoginValue.ts.dtIn.ToString();  
        textBox6.Text=wslprx.SoapHdLoginValue.ts.dtOut.ToString();  
        // Note que o SoapHdMoreInfo só tem valor após a chamada  
        textBox4.Text=wslprx.SoapHdMoreInfoValue.info;  
    } catch (Exception ex) {  
        MessageBox.Show(ex.ToString());  
    }  
}
```

## WSDL com SOAP Headers

```
<wsdl:types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://deetc.isel.ipl.pt/SD/">
    <s:element name="Login">...
    <s:element name="LoginResponse">...
    <s:element name="SoapHdLogin" type="tns:SoapHdLogin" />
    <s:complexType name="SoapHdLogin">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="UserName" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="TokenKey" type="s:string" />
        <s:element minOccurs="0" maxOccurs="1" name="ts" type="tns:TimeStamp" />
      </s:sequence>
      <s:anyAttribute />
    </s:complexType>
    <s:complexType name="TimeStamp">
      <s:sequence>
        <s:element minOccurs="1" maxOccurs="1" name="dtIn" type="s:dateTime" />
        <s:element minOccurs="1" maxOccurs="1" name="dtOut" type="s:dateTime" />
      </s:sequence>
    </s:complexType>
    <s:element name="SoapHdMoreInfo" type="tns:SoapHdMoreInfo" />
    <s:complexType name="SoapHdMoreInfo">
      <s:sequence>
        <s:element minOccurs="0" maxOccurs="1" name="info" type="s:string" />
      </s:sequence>
      <s:anyAttribute />
    </s:complexType>
  </s:schema>
</wsdl:types>
```

xml schema dos tipos  
usados nos SOAP Headers

## Elementos <message> do WSDL

```
<wsdl:message name="LoginSoapIn">
  <wsdl:part name="parameters" element="tns:Login" />
</wsdl:message>
<wsdl:message name="LoginSoapOut">
  <wsdl:part name="parameters" element="tns:LoginResponse" />
</wsdl:message>
<wsdl:message name="LoginSoapHdLogin">
  <wsdl:part name="SoapHdLogin" element="tns:SoapHdLogin" />
</wsdl:message>
<wsdl:message name="LoginSoapHdMoreInfo">
  <wsdl:part name="SoapHdMoreInfo" element="tns:SoapHdMoreInfo" />
</wsdl:message>
```

Operation message

Response message

SOAP Headers messages

## Elemento `<portType>` - Operações do Web Service

```
<wsdl:portType name="WSLoginSoap">
  <wsdl:operation name="Login">
    <wsdl:input message="tns:LoginSoapIn" />
    <wsdl:output message="tns:LoginSoapOut" />
  </wsdl:operation>
</wsdl:portType>
```

- Note que no elemento `<portType>` nada é indicado relativamente à existência de SOAP Headers.
- A associação dos SOAP Headers é realizada no elemento `<binding>` do WSDL (ver slide seguinte).

## Binding da operação Login() com SOAP Headers

```
<wsdl:binding name="WSLoginSoap" type="tns:WSLoginSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="Login">
    <soap:operation soapAction="http://deetc.isel.ipl.pt/SD/Login" style="document" />
    <wsdl:input>
      <soap:body use="literal" />
      <soap:header message="tns:LoginSoapHdLogin" part="SoapHdLogin" use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
      <soap:header message="tns:LoginSoapHdLogin" part="SoapHdLogin" use="literal" />
      <soap:header message="tns:LoginSoapHdMoreInfo" part="SoapHdMoreInfo" use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
```

} Header de input

} Headers de output

## Mensagens SOAP <fault>

- No elemento <message> podem ser definidas mensagens associadas a falhas;
- <message> name="ISBNnotValid">

```
<message name="ISBNnotValid">  
  <part name="error_message" element="tns:ISBNnotValidDetail" />  
</message>
```

Implica existir em <types> o elemento com name= "ISBNnotValidDetail"

- No entanto em .NET não está definido nenhuma forma de gerar um WSDL que contenha as mensagens SOAP envolvidas, quando o *Web Service* faz *throw* de uma exceção;
- No entanto as mensagens SOAP retornadas permitem passar os detalhes, sendo responsabilidade do cliente do *Web Service* obtê-los.

## fault detail numa SoapException em .NET

```
[WebMethod]  
public float getBookPrice(string isbn) {  
  if (isbnNotValid(isbn)) {  
    XmlDocument doc = new XmlDocument();  
    XmlNode node = doc.CreateNode(XmlNodeType.Element,  
                                   SoapException.DetailElementName.Name,  
                                   SoapException.DetailElementName.Namespace);  
    XmlNode child = doc.CreateNode(XmlNodeType.Element,  
                                    "error",  
                                    "http://DEETC.SES.SD");  
    child.InnerText="BookService:Not valid ISBN in getBookPrice operation.";  
    node.AppendChild(child);  
    throw new SoapException("ISBN not Valid",  
                             SoapException.ServerFaultCode,  
                             "ActorServer",  
                             node);  
  }  
  return 24.9F;  
}
```

```
// para simular a exceção  
private bool isbnNotValid(string isbn) {  
  return true;  
}
```

## Código do Cliente

```
private void BtgetPrice_Click(object sender, System.EventArgs e) {  
    try {  
        textBox3.Text=(bserv.getBookPrice(textBox4.Text)).ToString();  
    } catch (SoapException ex) {  
        MessageBox.Show(ex.Message+"\n"+  
            "Code:"+ex.Code+"\n"+  
            "Details:"+ex.Detail.OuterXml+"\n"+  
            "Actor role:"+ex.Actor);  
    }  
}
```

