物联网中间件

第六章 Web Services ——SOAP

不同的Web Services技术

- 两大类Web Services技术
  - Non-RESTful
  - RESTful

- 每一类Web Services技术之间是可以相互通信的，比如一个基于RESTful的技术可以调用另一个基于RESTful的技术。
- 不同类别的Web Services技术之间是不能相互通信的，比如一个基于non-RESTful的技术不可以调用一个基于RESTful的技术。

soap

- soap（Simple Object Access Protocol）即简单对象访问协议，是一种标准化的通讯规范，主要用于Web服务（web service）中。

- 它基于已经广泛使用的HTTP和XML协议，是第一个没有发明任何新技术的技术。

什么是 SOAP?

- SOAP 指*简易对象访问协议*
- SOAP 是一种*通信协议*
- SOAP 用于*应用程序之间*的通信
- SOAP 是一种用于*发送消息*的格式
- SOAP 被设计用来*通过因特网*进行通信
- SOAP *独立于平台*

- SOAP *独立于语言*
- SOAP *基于 XML*
- SOAP *很简单并可扩展*
- SOAP 允许您*绕过防火墙*
- SOAP 是*W3C 标准*
- 目前SOAP的版本主要有SOAP1.1和SOAP1.2（PPT上的例子为SOAP1.2版本）

为什么使用 SOAP?

- 对于应用程序开发来说，使程序之间进行因特网通信是很重要的。

- 传统的远程过程调用（RPC），诸如 JAVA RMI 与 CORBA 等，会产生兼容性以及安全问题；防火墙和代理服务器通常会阻止此类流量。

- 通过 HTTP 在应用程序间通信是更好的方法，因为 HTTP 得到了所有的因特网浏览器及服务器的支持。

- SOAP 就是被创造出来完成这个任务的。

- SOAP 提供了一种标准的方法，使得运行在不同的操作系统并使用不同的技术和编程语言的应用程序可以互相进行通信。

SOAP 语法

- **SOAP 构建模块**
- 一条 SOAP 消息就是一个普通的 XML 文档，包含下列元素:
- 必需的 Envelope 元素，可把此 XML 文档标识为一条 SOAP 消息
- 可选的 Header 元素，包含头部信息
- 必需的 Body 元素，包含所有的调用和响应信息
- 可选的 Fault 元素，提供有关在处理此消息所发生错误的信息

语法规则

- 这里是一些重要的语法规则:
- SOAP 消息必须用 XML 来编码
- SOAP 消息必须使用 SOAP Envelope 命名空间
- SOAP 消息必须设定 SOAP Encoding

```
<?xml version="1.0"?>
<soap:Envelope
        xmlns:soap="http://www.w3.org/2003/05/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 元素

- 必需的 SOAP 的 Envelope 元素是 SOAP 消息的根元素。它可把 XML 文档定义为 SOAP 消息。

```
<?xml version="1.0"?>
<soap:Envelope
        xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
        soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
    ...
    Message information goes here
    ...
</soap:Envelope>
```

- **xmlns:soap 命名空间**
- SOAP 消息必须拥有与命名空间 "http://www.w3.org/2003/05/soap-envelope" 相关联的一个 Envelope 元素。
- 如果使用了不同的命名空间，应用程序会发生错误，并抛弃此消息。

- **encodingStyle 属性**
- SOAP 的 encodingStyle 属性用于定义在文档中使用的数据类型。此属性可出现在任何 SOAP 元素中，并会被应用到元素的内容及元素的所有子元素上。

SOAP Header 元素

- 可选的 SOAP Header 元素可包含有关 SOAP 消息的应用程序专用信息（比如认证、支付等）。
- 如果 Header 元素被提供，则它必须是 Envelope 元素的第一个子元素。
- **注意：** 所有 Header 元素的直接子元素必须是合格的命名空间。
-

```
<?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>
```

- 上面的例子包含了一个带有一个 "Trans" 元素的头部，它的值是 234，此元素的 "mustUnderstand" 属性的值是 "1"。

- SOAP 在默认的命名空间中 ("http://www.w3.org/2001/12/soap-envelope") 定义了三个属性。
- 这三个属性是：actor、 mustUnderstand 以及 encodingStyle。这些被定义在 SOAP 头部的属性可定义容器如何对 SOAP 消息进行处理。

- **mustUnderstand 属性**
- SOAP 的 mustUnderstand 属性可用于标识标题项对于要对其进行处理的接收者来说是强制的还是可选的。
- 假如您向 Header 元素的某个子元素添加了 "mustUnderstand="1"，则它可指示处理此头部的接收者必须认可此元素。假如此接收者无法认可此元素，则在处理此头部时必须失效。
- **语法**
- soap:mustUnderstand="0|1"

- **actor 属性**
- 通过沿着消息路径经过不同的端点，SOAP 消息可从某个发送者传播到某个接收者。并非 SOAP 消息的所有部分均打算传送到 SOAP 消息的最终端点，不过，另一个方面，也许打算传送给消息路径上的一个或多个端点。
- SOAP 的 actor 属性可被用于将 Header 元素寻址到一个特定的端点。
- **语法**
- 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 元素

- 必需的 SOAP Body 元素可包含打算传送到消息最终端点的实际 SOAP 消息。

- SOAP Body 元素的直接子元素可以是合格的命名空间。

```
<?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>
```

- 上面的例子请求苹果的价格。请注意，上面的 m:GetPrice 和 Item 元素是应用程序专用的元素。它们并不是 SOAP 标准的一部分。

```
<?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>
```

- 而一个 SOAP 响应如上所示

SOAP Fault 元素

- 可选的 SOAP Fault 元素用于指示错误消息。
- 如果已提供了 Fault 元素，则它必须是 Body 元素的子元素。在一条 SOAP 消息中，Fault 元素只能出现一次。
- SOAP 的 Fault 元素拥有下列子元素：

子元素	描述
<faultcode>	供识别故障的代码
<faultstring>	可供人阅读的有关故障的说明
<faultactor>	有关是谁引发故障的信息
<detail>	存留涉及 Body 元素的应用程序专用错误信息

SOAP Fault 代码

错误	描述
VersionMismatch	SOAP Envelope 元素的无效命名空间被发现
MustUnderstand	Header 元素的一个直接子元素（带有设置为 "1" 的 mustUnderstand 属性）无法被理解。
Client	消息被不正确地构成，或包含了不正确的信息。
Server	服务器有问题，因此无法处理进行下去。

SOAP HTTP Binding

- HTTP 在 TCP/IP 之上进行通信。HTTP 客户机使用 TCP 连接到 HTTP 服务器。在建立连接之后，客户机可向服务器发送 HTTP 请求消息：

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

- 随后服务器会处理此请求，然后向客户机发送一个 HTTP 响应。此响应包含了可指示请求状态的状态代码

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

- 在上面的例子中，服务器返回了一个 200 的状态代码。这是 HTTP 的标准成功代码。
- 假如服务器无法对请求进行解码，它可能会返回类似这样的信息

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

SOAP HTTP Binding

- **SOAP HTTP Binding** 指的是遵守 SOAP 编码规则的 HTTP 请求/响应。

- **HTTP + XML = SOAP**
- SOAP 请求可能是 HTTP POST 或 HTTP GET 请求。
- HTTP POST 请求规定至少两个 HTTP 头： Content-Type 和 Content-Length。

Content-Type

- SOAP 的请求和响应的 Content-Type 头可定义消息的 MIME 类型，以及用于请求或响应的 XML 主体的字符编码（可选）。

- 语法
- Content-Type: MIMEType; charset=character-encoding
- 实例
- POST /item HTTP/1.1
- Content-Type: application/soap+xml; charset=utf-8

Content-Length

- SOAP 的请求和响应的 Content-Length 头规定请求或响应主体的字节数。

- 语法
- Content-Length: bytes
- 实例
- POST /item HTTP/1.1
- Content-Type: application/soap+xml; charset=utf-8
- Content-Length: 300

一个 SOAP 实例

- 在下面的例子中，一个 GetStockPrice 请求被发送到了服务器。此请求有一个 StockName 参数，而在响应中则会返回一个 Price 参数。此功能的命名空间被定义在此地址中： "http://www.example.org/stock"

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 300

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/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>
```

```
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/2003/05/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>
```

答疑