



SOAP 教程

SOAP 是一种简单的基于 XML 的协议，它使应用程序通过 HTTP 来交换信息。

在我们的 SOAP 教程中，你将了解到什么是 SOAP，以及它如何在应用程序之间交换信息。

[现在开始学习 SOAP!](#)

内容列表

[SOAP 简介](#)

本章讲解 SOAP 的概念和作用。

[SOAP 语法](#)

本章讲解 SOAP 的构建模块和 SOAP 的语法规则。

[SOAP Envelope 元素](#)

本章讲解 SOAP 的 Envelope 元素。

[SOAP Header 元素](#)

本章讲解 SOAP 的 Header 元素。

[SOAP Body 元素](#)

本章讲解 SOAP 的 Body 元素。

[SOAP Fault 元素](#)

本章讲解 SOAP 的 Fault 元素。

[SOAP HTTP Binding](#)

本章是对 SOAP HTTP 绑定的简介。

[SOAP 实例](#)

本章介绍了一个 SOAP 实例。

[SOAP 总结](#)

本节是对您在本教程所学内容的一个总结，并为您推荐了下一步应该学习的内容。

[点我分享笔记](#)

反馈/建议



SOAP 简介

SOAP 是基于 XML 的简易协议，可使应用程序在 HTTP 之上进行信息交换。

或者更简单地说：SOAP 是用于访问网络服务的协议。

您应当具备的基础知识

在继续学习之前，您需要对下面的知识有基本的了解：

XML

XML 命名空间

如果您希望首先学习这些项目，请访问我们的《[XML 教程](#)》。

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 将被作为 *W3C 标准*来发展

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。

目前的应用程序通过使用远程过程调用（RPC）在诸如 DCOM 与 CORBA 等对象之间进行通信，但是 HTTP 不是为此设计的。RPC 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。

Microsoft 和 SOAP

SOAP 是微软 .net 架构的关键元素，用于未来的因特网应用程序开发。

SOAP 1.1 被提交到 W3C

在 2000 年 5 月，UserLand、Ariba、Commerce One、Compaq、Developmentor、HP、IBM、IONA、Lotus、Microsoft 以及 SAP 向 W3C 提交了 SOAP 因特网协议，这些公司期望此协议能够通过使用因特网标准（HTTP 以及 XML）把图形用户界面桌面应用程序连接到强大的因特网服务器，以此来彻底变革应用程序的开发。

W3C 正在发展 SOAP 1.2

首个关于 SOAP 的公共工作草案由 W3C 在 2001 年 12 月发布。如需阅读更多有关在 W3C 的 SOAP 活动的内容，请访问我们的《[W3C 教程](#)》。

□ SOAP 语法

SOAP Envelope 元素 □

□ [点我分享笔记](#)

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)[□ SOAP 教程](#)[SOAP 简介 □](#)

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

所有以上的元素均被声明于针对 SOAP 封装的默认命名空间中：

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

以及针对 SOAP 编码和数据类型的默认命名空间：

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

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 消息必须使用 SOAP Encoding 命名空间

SOAP 消息不能包含 DTD 引用

SOAP 消息不能包含 XML 处理指令

SOAP 消息的基本结构

```
<?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 教程](#)[SOAP 简介 □](#)

□ 点我分享笔记

反馈/建议

Copyright © 2013-2018 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1



[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)

□ SOAP 简介

SOAP Header 元素 □

SOAP Envelope 元素

强制使用的 SOAP 的 Envelope 元素是 SOAP 消息的根元素。

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。

实例

```
<?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 命名空间

SOAP 消息必须拥有与命名空间 "http://www.w3.org/2001/12/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 消息没有默认的编码方式。

语法

soap:encodingStyle="URI"

实例

```
<?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>
```

❏ 点我分享笔记

反馈/建议



SOAP Header 元素

可选的 SOAP Header 元素包含头部信息。

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"
```

实例

```
<?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 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"
```

实例

```
<?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>
```

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 消息没有默认的编码方式。

语法

```
soap:encodingStyle="URI"
```

SOAP Body 元素

强制使用的 SOAP Body 元素包含实际的 SOAP 消息。

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。

实例

```
<?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。

而一个 SOAP 响应应该类似这样：

```
<?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>
```

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)[□ SOAP Body 元素](#)[SOAP HTTP Binding □](#)

SOAP Fault 元素

SOAP Fault 元素用于存留 SOAP 消息的错误和状态信息。

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

在下面定义的 faultcode 值必须用于描述错误时的 faultcode 元素中：

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 Body 元素](#)[SOAP HTTP Binding □](#)[□ 点我分享笔记](#)

SOAP HTTP 协议

HTTP 协议

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 方法指的是遵守 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: 250
```



SOAP 实例

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"`

SOAP 请求:

```
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>
```

SOAP 响应:

```
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>
```

[☐ 点我分享笔记](#)

反馈/建议



您已经学习了 **SOAP**，下一步呢？

此教程已向您讲解了如何透过 HTTP 使用 SOAP 在应用程序之间交换信息。

您已经学习了有关 SOAP 消息中不同元素和属性的知识。

您也学习了如何把 SOAP 作为一种协议来使用以访问 web service。

您已经学习了 **SOAP**，下一步呢？

下一步应该学习 WSDL 和 Web Services。

WSDL

WSDL 是基于 XML 的语言，用于描述 Web services 以及如何访问它们。

WSDL 可描述某个 web service，连同用于此 web service 的消息格式和协议细节。

如果您需要学习更多有关 WSDL 的知识，请访问我们的 [WSDL 教程](#)。

Web Services

Web services 可把应用程序转换为网络应用程序（web-applications）。

通过使用 XML，消息可在应用程序之间进行传送。

如果您需要学习更多有关 web services 的知识，请访问我们的 [Web Services 教程](#)。

[☐ SOAP 实例](#)

[☐ 点我分享笔记](#)

反馈/建议