Web Services 简介

* [WS 教程](http://www.w3school.com.cn/webservices/index.asp)
* [Why WS](http://www.w3school.com.cn/webservices/ws_why.asp)

**Web Services 可使您的应用程序成为 Web 应用程序。**

**Web Services 通过 Web 进行发布、查找和使用。**

**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：

* HTML
* XML

如果您希望首先学习这些项目，请在我们的[首页](http://www.w3school.com.cn/index.html" \o "W3School 在线教程)访问这些教程。

**什么是Web Services？**

* Web Services 是应用程序组件
* Web Services 使用开放协议进行通信
* Web Services 是独立的（self-contained）并可自我描述
* Web Services 可通过使用UDDI来发现
* Web Services 可被其他应用程序使用
* XML 是 Web Services 的基础

**它如何工作？**

基础的 Web Services 平台是 XML + HTTP。

HTTP 协议是最常用的因特网协议。

XML 提供了一种可用于不同的平台和编程语言之间的语言。

**Web services 平台的元素：**

* SOAP (简易对象访问协议)
* UDDI (通用描述、发现及整合)
* WSDL (Web services 描述语言)

我们会在本教程后面章节讲解这些主题。

**Web services 的未来**

请不要过早地期待太多！

Web services 平台是简单的可共同操作的消息收发框架。它仍然缺少许多诸如安全和路由等重要的特性。但是，一旦 SOAP 变得更加高级，这些事项就会得到解决。

Web services 有望使应用程序更加容易通信。

Why Web Services?

* [WS 简介](http://www.w3school.com.cn/webservices/ws_intro.asp)
* [WS 平台](http://www.w3school.com.cn/webservices/ws_platform.asp)

**几年前，Web services 的速度还没有快到让人们产生兴趣的程度。**

**感谢主要的 IT 开发商近年来的努力，大多数的人们和企业拥有了宽带连接，并越来越多地使用 Web。**

**最重要的事情是协同工作**

由于所有主要的平台均可通过 Web 浏览器来访问 Web，不同的平台可以借此进行交互。为了让这些平台协同工作，Web 应用程序被开发了出来。

Web 应用程序是运行在 Web 上的简易应用程序。它们围绕 Web 浏览器标准被进行构建，几乎可被任何平台之上的任何浏览器来使用。

**Web services 把 Web 应用程序提升到了另外一个层面**

通过使用 Web services，您的应用程序可向全世界发布功能或消息。

Web services 使用 XML 来编解码数据，并使用 SOAP 借由开放的协议来传输数据。

通过 Web services，您的会计部门的 Win 2k 服务器可与 IT 供应商的 UNIX 服务器进行连接。

**Web services 有两种类型的应用**

**可重复使用的应用程序组件**

有一些功能是不同的应用程序常常会用到的。那么为什么要周而复始地开发它们呢？

Web services 可以把应用程序组件作为服务来提供，比如汇率转换、天气预报或者甚至是语言翻译等等。

比较理想的情况是，每种应用程序组件只有一个最优秀的版本，这样任何人都可以在其应用程序中使用它。

**连接现有的软件**

通过为不同的应用程序提供一种链接其数据的途径，Web services有助于解决协同工作的问题。

通过使用 Web services，您可以在不同的应用程序与平台之间来交换数据。

Web Services 平台元素

* [Why WS](http://www.w3school.com.cn/webservices/ws_why.asp)
* [WS 实例](http://www.w3school.com.cn/webservices/ws_example.asp)

**Web Services 拥有三种基本的元素。**

**它们是：SOAP、WSDL 以及 UDDI。**

**什么是 SOAP？**

基本的 Web services 平台是 XML + HTTP。

* 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 将作为 W3C 标准来发展

如需更多有关 SOAP 的知识，请访问我们的《[SOAP 教程](http://www.w3school.com.cn/soap/index.asp" \o "SOAP教程)》

**什么是 WSDL?**

WSDL 是基于 XML 的用于描述 Web Services 以及如何访问 Web Services 的语言。

* WSDL 指网络服务描述语言
* WSDL 使用 XML 编写
* WSDL 是一种 XML 文档
* WSDL 用于描述网络服务
* WSDL 也可用于定位网络服务
* WSDL 还不是 W3C 标准

如需更多有关 WSDL 的知识，请访问我们的《[WSDL 教程](http://www.w3school.com.cn/wsdl/index.asp" \o "WSDL 教程)》

**什么是UDDI？**

UDDI 是一种目录服务，通过它，企业可注册并搜索 Web services。

* UDDI 指通用的描述、发现以及整合（Universal Description, Discovery and Integration）。
* UDDI 是一种用于存储有关 web services 的信息的目录。
* UDDI 是一种由 WSDL 描述的网络服务接口目录。
* UDDI 经由 SOAP 进行通迅。
* UDDI 被构建于 Microsoft .NET 平台之中。

Web Service 实例

* [WS 平台](http://www.w3school.com.cn/webservices/ws_platform.asp)
* [WS 使用](http://www.w3school.com.cn/webservices/ws_use.asp)

**任何应用程序都可拥有 Web Service 组件。**

**Web Services 的创建与编程语言的种类无关。**

**一个实例：ASP.NET Web Service**

在这个例子中，我们会使用 ASP.NET 来创建一个简单的 Web Service。

<%@ WebService Language="VB" Class="TempConvert" %>

Imports System

Imports System.Web.Services

Public Class TempConvert :Inherits WebService

<WebMethod()> Public Function FahrenheitToCelsius

(ByVal Fahrenheit As Int16) As Int16

Dim celsius As Int16

celsius = ((((Fahrenheit) - 32) / 9) \* 5)

Return celsius

End Function

<WebMethod()> Public Function CelsiusToFahrenheit

(ByVal Celsius As Int16) As Int16

Dim fahrenheit As Int16

fahrenheit = ((((Celsius) \* 9) / 5) + 32)

Return fahrenheit

End Function

End Class

此文档是一个 .asmx 文件。这是用于 XML Web Services 的 ASP.NET 文件扩展名。

**要运行这个例子，我们需要一个 .NET 服务器**

此文档中第一行表明这是一个 Web Service，由 VB 编写，其 class 名称是 "TempConvert"。

<%@ WebService Language="VB" Class="TempConvert" %>

接下来的代码行从 .NET 框架导入了命名空间 "System.Web.Services"。

Imports System

Imports System.Web.Services

下面这一行定义 "TempConvert" 类是一个 WebSerivce 类：

Public Class TempConvert :Inherits WebService

接下来的步骤是基础的 VB 编程。此应用程序有两个函数。一个把华氏度转换为摄氏度，而另一个把摄氏度转换为华氏度。

与普通的应用程序唯一的不同是，此函数被定义为 "WebMethod"。

请在您希望其成为 web services 的应用程序中使用 "WebMethod" 来标记函数。

<WebMethod()> Public Function FahrenheitToCelsius

(ByVal Fahrenheit As Int16) As Int16

Dim celsius As Int16

celsius = ((((Fahrenheit) - 32) / 9) \* 5)

Return celsius

End Function

<WebMethod()> Public Function CelsiusToFahrenheit

(ByVal Celsius As Int16) As Int16

Dim fahrenheit As Int16

fahrenheit = ((((Celsius) \* 9) / 5) + 32)

Return fahrenheit

End Function

最后要做的事情是终止函数和类：

End Function

End Class

假如您把此文件另存为 .asmx 文件，并发布于支持 .NET 的服务器上，那么您就拥有了第一个可工作的 Web Service。

**ASP.NET 的自动化处理**

通过 ASP.NET，你不必亲自编写 WSDL 和 SOAP 文档。

如果您仔细研究我们的这个例子，您会发现 ASP.NET 会自动创建 WSDL 和 SOAP 请求。

Web Service 使用

* [WS 实例](http://www.w3school.com.cn/webservices/ws_example.asp)
* [WS 总结](http://www.w3school.com.cn/webservices/ws_summary.asp)

**使用我们的 ASP.NET Web Service 例子**

在上一节，我们创建了一个 [Web Service 的例子](http://www.w3schools.com/webservices/tempconvert.asmx)。

请在此测试华氏度转换摄氏度函数：[华氏度转换为摄氏度](http://www.w3schools.com/webservices/tempconvert.asmx?op=FahrenheitToCelsius" \o "TempConvert Web Service)

请在此测试摄氏度转换华氏度函数：[摄氏度转换为华氏度](http://www.w3schools.com/webservices/tempconvert.asmx?op=CelsiusToFahrenheit" \o "TempConvert Web Service)

**这些函数会向您发送一个 XML 回答**

本测试使用 HTTP POST，会发送类似这样的 XML 响应：

<?xml version="1.0" encoding="utf-8" ?>

<short xmlns="http://tempuri.org/">38</short>

**使用表单来访问 Web Service**

通过使用表单和 HTTP POST，您可以把 web service 置于您的站点上，比如这样：

Top of Form

华氏度转换为摄氏度：

 

Bottom of Form

Top of Form

摄氏度转换为华氏度：

 

Bottom of Form

**您可以把 web service 置于您的站点上**

您可以使用这些代码把 web service 放置在您的站点上：

<form target="\_blank"

action='http://w3school.com.cn/webservices/tempconvert.asmx/FahrenheitToCelsius'

method="POST">

<label>华氏度转换为摄氏度：</label>

<p>

<span>

<input class="frmInput" type="text" size="30" name="Fahrenheit">

</span>

<span>

<input type="submit" value="提交" class="button">

</span>

</p>

</form>

<form target="\_blank"

action='http://w3school.com.cn/webservices/tempconvert.asmx/CelsiusToFahrenheit'

method="POST">

<label>摄氏度转换为华氏度：</label>

<p>

<span>

<input class="frmInput" type="text" size="30" name="Celsius">

</span>

<span>

<input type="submit" value="提交" class="button">

</span>

</p>

</form>

您已经学习了 Web Services,下一步学习什么内容呢？

* [WS 使用](http://www.w3school.com.cn/webservices/ws_use.asp)
* [WS 教程](http://www.w3school.com.cn/webservices/index.asp)

**Web Services 概要**

本教程已经向您讲解了如何把应用程序转换为网络应用程序.

您已经学习了如何使用 XML 在应用程序间发送消息。

您也学习了如何从应用程序导出某项功能（创建一个 web service）。

**您已经学习了 Web Services,下一步呢？**

下一步您应当学习 WSDL 和 SOAP。

**WSDL**

WSDL 是基于 XML 的用来描述 Web services 以及如何访问它们的一种语言。

WSDL 可描述 web service，连同用于 web service 的消息格式和协议的细节。

如果您希望学习更多有关 WSDL 的知识，请访问我们的《[WSDL 教程](http://www.w3school.com.cn/wsdl/index.asp)》。

**SOAP**

SOAP 是一种使应用程序有能力通过 HTTP 交换信息的基于 XML 的简易协议。

或者可以更简单地说：SOAP 是一种用于访问 web service 的协议。

如果您希望学习更多有关 SOAP 的知识，请访问我们的《[SOAP 教程](http://www.w3school.com.cn/soap/index.asp)》。