

Учебный курс
**Технологии и средства разработки
корпоративных систем**

Лекция 7
Создание веб-сервисов

Лекции читает
**кандидат технических наук, доцент
Зыков Сергей Викторович**

Содержание

1. Web-сервисы и их использование
2. Пример Web-сервиса
3. Реализация Web-сервисов в .NET
4. Обнаружение Web-сервисов
5. WSDL – язык описания Web-сервисов
6. Протокол SOAP и его поддержка
7. Использование Web-сервисов в приложениях .NET
8. Безопасность Web-сервисов

Что такое Web-сервисы?

Web-сервис - особый тип Web-приложения. Задачей Web-сервиса является не генерация html разметки для браузера пользователя, как у традиционных Web-приложений, а предоставление функций, называемых Web-методами, которые можно вызывать по Интернету.

Краеугольные камни Web-сервисов

- Web-сервисы исполняются на сервере (исполняющей средой в нашем случае служит ASP.NET)
- Web-сервисы публикуют методы, которые могут быть вызваны внешними клиентами
- Web-сервисы ожидают поступления http-запросов для вызова методов
- Web-сервисы исполняют запросы и возвращают http ответы

Использование Web-сервисов

- В2В решения
- Доступная клиенту функциональность
- Интеграция гетерогенных систем
- Быстрая разработка

Пример Web-сервиса

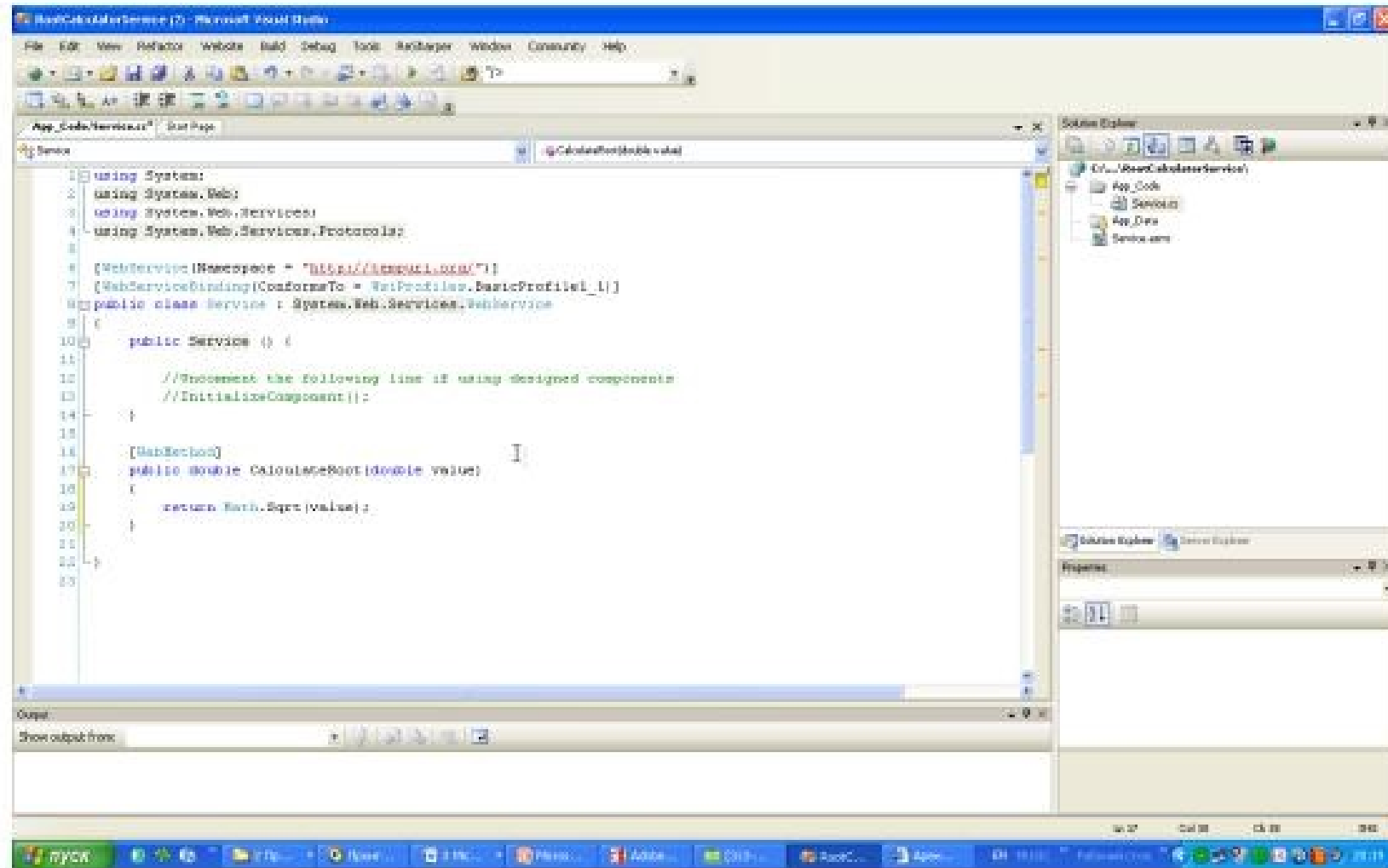
Рассмотрим пример Web-сервиса вычисляющий квадратный корень числа

В Microsoft Visual Studio .NET 2005 (2008) для создания Web-сервиса необходимо выбрать New Web Site - > ASP.NET Web Service. Назовем Web-сервис RootCalculatorService

Среда Microsoft Visual Studio .NET генерирует несколько файлов Service.asmx, Web.config и Service.cs

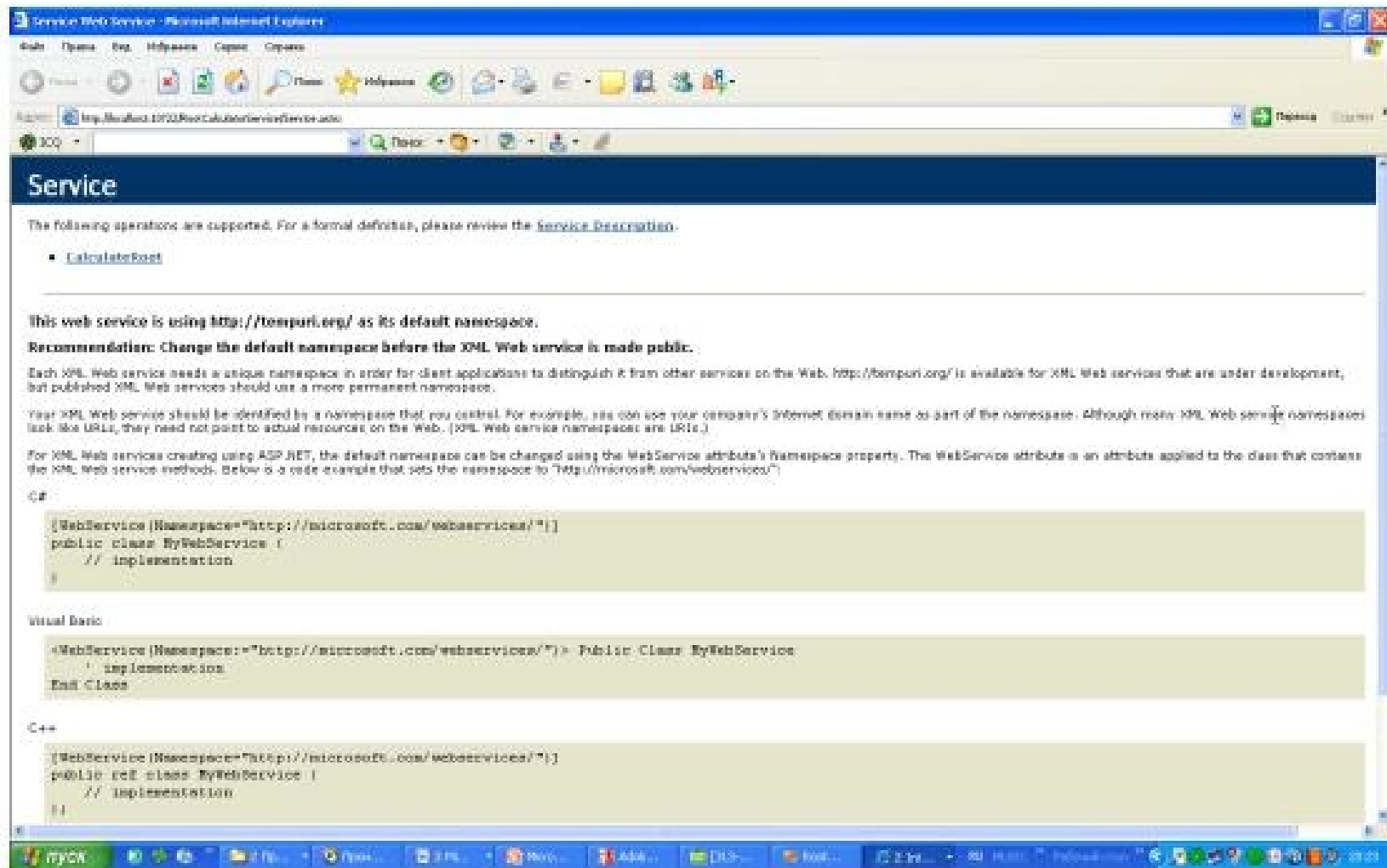
Корпоративные системы
Веб-сервисы и распределенные приложения

Пример Web-сервиса (1)



```
1 using System;
2 using System.Web;
3 using System.Web.Services;
4 using System.Web.Services.Protocols;
5
6 [WebService(Namespace = "http://tempuri.org/")]
7 [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
8 public class Service : System.Web.Services.WebService
9 {
10     public Service () {
11         //To prevent the following line if using designed components
12         //InitializeComponent();
13     }
14
15     [WebMethod]
16     public double CalculateRoot(double value)
17     {
18         return Math.Sqrt(value);
19     }
20 }
21
22
23
```

Пример Web-сервиса (2)



Service

The following operations are supported. For a formal definition, please review the [Service Description](#).

- CalculatorExt

This web service is using <http://tempuri.org/> as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. <http://tempuri.org/> is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URIs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services created using ASP.NET, the default namespace can be changed using the `WebService` attribute's `Namespace` property. The `WebService` attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to `"http://microsoft.com/webservices/"`.

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // Implementation
}
```

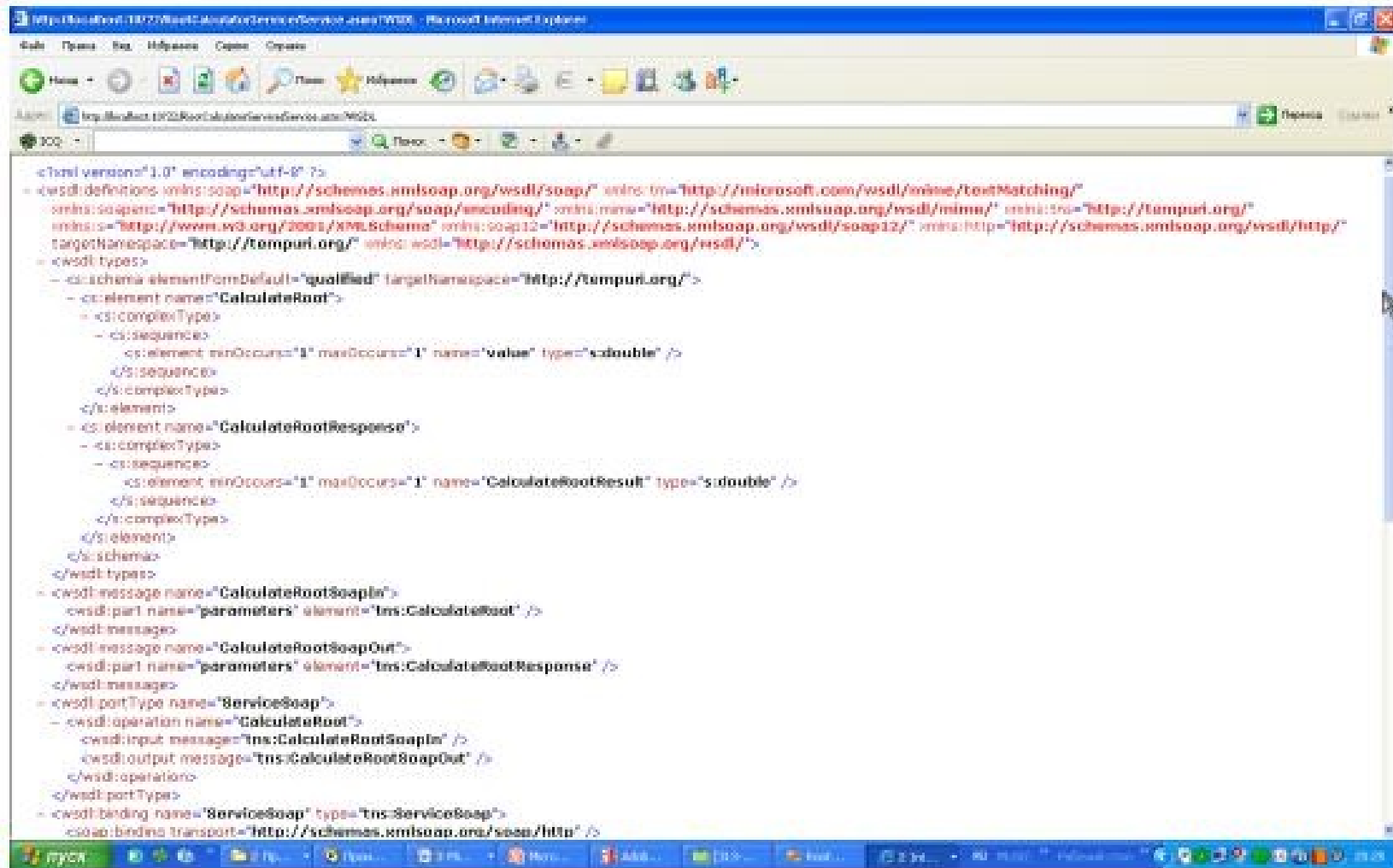
Visual Basic

```
<WebService(Namespace="http://microsoft.com/webservices/") > Public Class MyWebService
    ' Implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // Implementation
}
```


Пример Web-сервиса (3)



```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
targetNamespace="http://tempuri.org/" xmlns:wsd="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
<!-- schema: elementFormDefault="qualified" targetNamespace="http://tempuri.org/" -->
<!-- element name="CalculateRoot" -->
<!-- complexType -->
<!-- sequence -->
<!-- element minOccurs="1" maxOccurs="1" name="value" type="xsd:double" -->
</sequence>
</complexType>
</element>
<!-- element name="CalculateRootResponse" -->
<!-- complexType -->
<!-- sequence -->
<!-- element minOccurs="1" maxOccurs="1" name="CalculateRootResult" type="xsd:double" -->
</sequence>
</complexType>
</element>
</schema>
</wsdl:types>
<wsdl:message name="CalculateRootSoapIn">
<wsdl:part name="parameters" element="tns:CalculateRoot" />
</wsdl:message>
<wsdl:message name="CalculateRootSoapOut">
<wsdl:part name="parameters" element="tns:CalculateRootResponse" />
</wsdl:message>
<wsdl:portType name="ServiceSoap">
<wsdl:operation name="CalculateRoot">
<wsdl:input message="tns:CalculateRootSoapIn" />
<wsdl:output message="tns:CalculateRootSoapOut" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
</wsdl:binding>
</wsdl:definitions>
```

Пример Web-сервиса (4)

The screenshot displays two overlapping windows of Microsoft Internet Explorer. The background window shows a web service interface for 'CalculateRoot'. It includes a 'Test' section with a 'Parameter/Value' field containing '4' and a 'Break' button. Below this is a 'SOAP 1.1' section with a sample request and response. The foreground window shows the raw SOAP response in XML format.

Service

Click [here](#) for a complete list of operations.

CalculateRoot

Test

To test the operation using the HTTP POST protocol, click the 'Break' button.

Parameter/Value
value:

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholder `<value>` is used to represent the value returned by the service.

```
POST /RootCalculatorService/Service.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/CalculateRoot"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <CalculateRoot xmlns="http://tempuri.org/">
      <value><double/>
    </CalculateRoot>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8

```
<?xml version="1.0" encoding="utf-8"?>
<double xmlns="http://tempuri.org/">2</double>
```

Общие характеристики Web-сервисов:

- Web-сервисы реализуются как `asmx`-файлы (данное расширение зарегистрировано в файле `Machine.config`)
- Исполняемый код Web-сервиса может находиться как в самом файле `asmx`, так и в специальном файле с кодом
- `Asmx`-файлы начинаются директивой `@WebService`, эта директива содержит, по крайней мере, атрибут `class`, задающий класс, из которого состоит Web-сервис
- Классы Web-сервисов могут иметь необязательные атрибуты `WebService`
- Web-методы объявляются путем назначения открытым методам класса сервиса атрибута `WebMethod`.

Атрибут `WebMethod`

Имя параметра	Значение
<code>bufferResponse</code>	Включение/отключение буферизации откликов
<code>cacheDuration</code>	Кэширует отклики данного метода на указанное число секунд
<code>Description</code>	Описание Web-метода
<code>enableSession</code>	Включает/отключает поддержку состояния сеанса
<code>messageName</code>	Название Web-метода
<code>transactionOption</code>	Параметры обработки транзакций

Наследование от класса `WebService`

- Наследование от данного класса позволяет получить прямой доступ к таким свойствам как `Application`, `Session`, `Context`, `Server`, `User`
- Наследование от этого класса позволяет в частности использовать технологию `.NET Remoting`

Обнаружение Web-сервисов

- Файловый механизм DISCO – для обнаружения локальных Web-сервисов
- Служба Universal Description, Discovery and Integration (UDDI) – для глобального поиска Web-сервисов

Язык описания Web-сервисов WSDL

- Язык WSDL - XML-диалект, позволяющий описывать Web-сервисы и их методы
- Язык WSDL использует несколько уровней абстракции для описания Web-сервисов в модульном ключе
- Описание WSDL помещается между тегами базового элемента `Definitions` и включает разделы:
 - `Types`,
 - `Operations`,
 - `Bindings`,
 - `Services`
 - `Messages`,
 - `Port Types`,
 - `Ports`,

SOAP – протокол взаимодействия Web-сервисов

- SOAP – протокол удаленного вызова, функционирующий поверх http (Simple Object Access Protocol)
- Каждое сообщение SOAP состоит из следующих частей:
 - Конверт
 - Заголовок
 - Тело
- SOAP основывается на XML

Настройка представления SOAP

- Среда .NET позволяет настраивать формат сообщений SOAP отправляемых Web-методом
- Настраивать это представление позволяют атрибуты:
 - SoapMethodAttribute
 - SoapRpcMethodAttribute

Конфигурация заголовков SOAP

- Заголовки SOAP настраиваются программно с помощью атрибута `SoapHeaderAttribute`
- Для определения своего собственного представления заголовка необходимо наследовать от класса `System.Web.Services.Protocols.SoapHeader`
- Для атрибута указывается имя переменной класса заголовка `[WebService()]`

```
public class Service1 : System.Web.Services.WebService {  
    public Header1 m_foo;  
    ...  
    [WebMethod]  
    [SoapHeader("m_foo")]  
    public string Bar() { m_foo.DidUnderstand = true;  
return m_foo.Header; }  
}
```

Расширения SOAP

- Расширения (Extensions) SOAP – предоставляют широкие возможности по настройке и обработке пакетов данных в формате SOAP
- Для использования расширений SOAP необходимо:
 - создать наследника класса `SoapExtension`,
 - использовать сопутствующий атрибут `SoapExtensionAttribute`

Прокси Web-сервисов

- Прокси Web-сервиса - локальное представление удаленного Web-сервиса в приложении
- Прокси является объектом класса, созданного .NET Framework SDK на основе WSDL-описания данного Web-сервиса
- Методы данного класса соответствуют методам Web-сервиса
- Генерация данных классов встроена в Microsoft Visual Studio .NET, однако их можно генерировать и специальной утилитой `wsdl.exe`

Асинхронный вызов Web-сервиса

- Прокси Web-сервисов позволяют осуществлять как синхронный, так и асинхронный вызов методов
- Асинхронные методы Web-сервиса помечены префиксами `Begin` и `End`
- Сигнализатором окончания вызова служит либо объект возвращаемый методом `Begin[MethodName]` и реализующий интерфейс `IAsyncResult`, либо подписка на уведомление о завершении метода путем передачи делегата

Web-сервисы и безопасность

Intranet

- IP Security
- Firewalls
- Virtual Private Networks
- Http Security
- ASP.NET Authentication

Internet

- SOAP Digital Signatures
- Application-Specific Authentication