ПWS

ПОИТ-4,

лабораторная 4

4 часа

**Разработка и исследование простейшего SOAP-сервиса**

1. Разработайте ASMX-сервис с именем **Simplex** помощью технологии ASP.NET.
2. Найдите на сайте W3C описание стандарта WSDL.
3. Найдите на сайте W3C описание протоколов SOAP 1.1, SOAP 1.2.
4. Установите для ASMX-сервиса собственное пространство имен http://XYZ/, где XYZ – ваши инициалы.
5. ASMX-сервис **Simplex** включает в себя следующие три метода.

|  |  |  |
| --- | --- | --- |
| Имя метода и его назначение | Параметры метода | Возвращаемый тип |
| **Add**  Возвращает значение суммы двух параметров | int x,  int y | int |
| **Concat**  Возвращает конкатенацию первого и второго параметров | string s  double d, | string |
| **Sum**  Возвращает объект A:  поле s – конкатенация a1.s и a2.s;  поле к – сумма a1.k и a2.k;  поле f – сумма a1.f и a2.f; | A a1,  A a2 | A |

1. Для класса ASM-сервиса **Simplex** определите параметр **Description** атрибута **WebService**. Поясните его назначение.
2. Для каждого метода ASM-сервиса определите параметр **Description** атрибута **WebMethod**. Поясните его назначение.
3. Для каждого метода ASM-сервиса **Simplex** определите параметр **MessageName** атрибута **WebMethod**. Поясните его назначение.
4. Подключитесь с помощью браузера к ASMX-сервису **Simplex**.
5. Проверьте работоспособность методов **Add** и **Concat** c

помощью стандартной web-формы описания ASMX-сервиса **Simplex**.

1. Для проверки работоспособности метода **Sum** ASMX-сервиса **Simplex** разработайте WinForm-приложение. Используя объект **Request** в методе **Sum** ASMX-сервиса, продемонстрируйте и исследуйте содержимое тела запроса, пришедшего от клиента.
2. Исследуйте WSDL-описание (отображается в браузере), ASMX-сервиса **Simplex** и укажите:

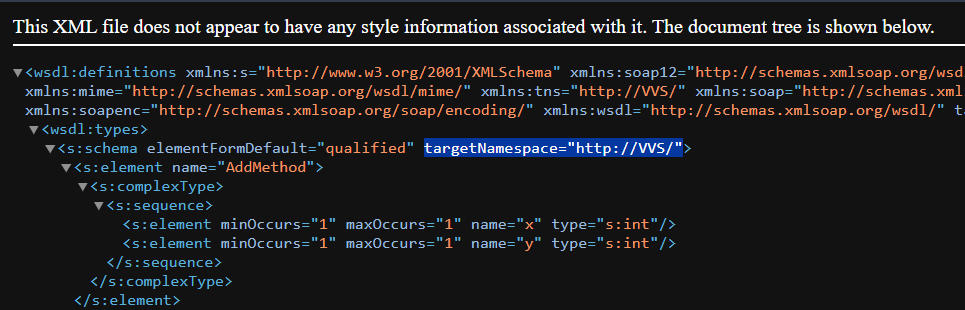
- где применяется собственное пространство имен;

* описание типов данных для всех методов сервиса;
* описание входных и выходных сообщений всех методов сервиса;
* описание методов сервиса;
* описание протоколов обмена данных;
* описание сервиса.

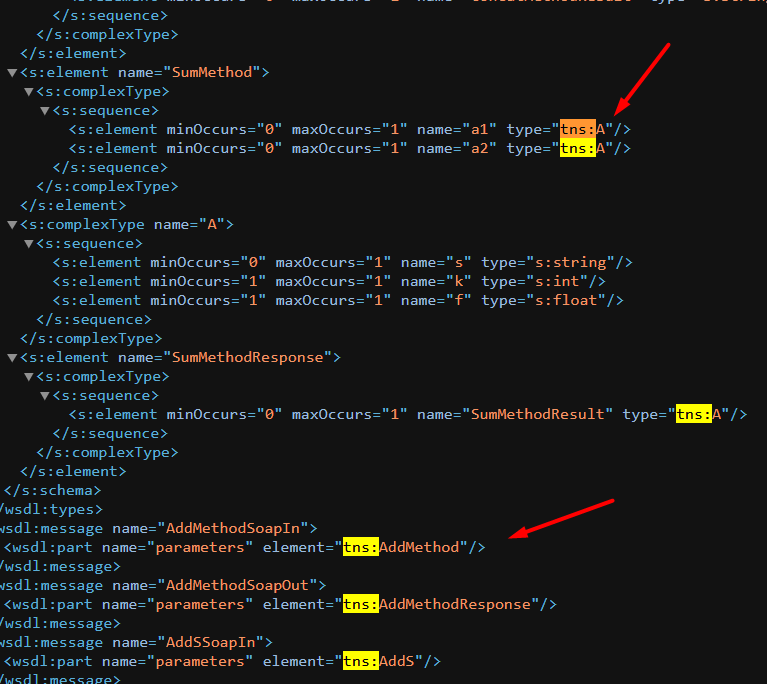
1. Ознакомьтесь со структурой http-запросов для доступа к ASMX-сервису **Simplex** (отображается на стандартной web-форме в браузере).
2. С помощью приложения **POSTMAN** выполните два типа запросов (SOAP 1.2, HTTP POST) к ASMX-сервису **Simplex** и убедитесь в их работоспособности.
3. Разработайте WinForm-приложение. С помощью утилиты **WSDL.EXE** сгенерируйте proxy-класс. Продемонстрируйте его работоспособность в отдельном приложении.
4. Разработайте WebForm-приложение. С помощью утилиты **WSDL.EXE** сгенерируйте класс для отдельного ASMX-сервиса. Продемонстрируйте его работоспособность.
5. Создайте в ASMX-сервисе **Simplex** дополнительный метод **AddS.** Метод **AddS** аналогичен методу **Add,** но предназначен для вызова с помощью AJAX-запроса, отправляющего и принимающего сообщения в формате JSON.
6. Разработайте ASP.NET-приложение, применяющее AJAX-запрос (jQuery) и демонстрирующее работоспособность метода **AddS** ASMX-сервиса **Simplex.**
7. Исследуйте WSDL-описание (отображается в браузере), ASMX-сервиса **Simplex** и укажите:

- где применяется собственное пространство имен;

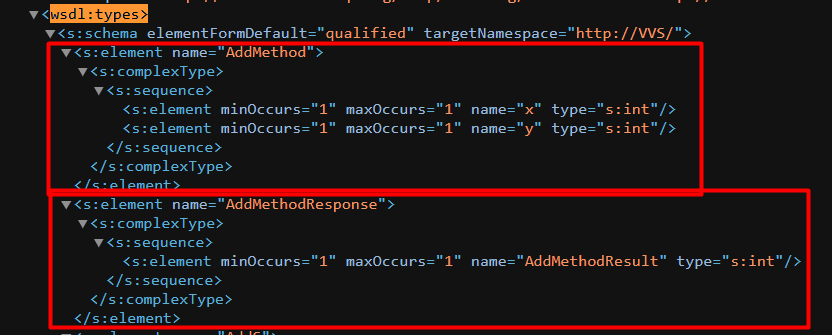
Определяется вот так:



Используется для определение пользовательских типов данных, идентификации сообщений и операций:



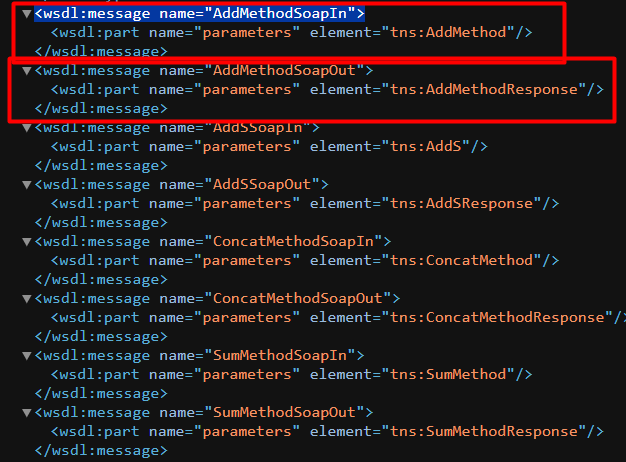
* описание типов данных для всех методов сервиса;



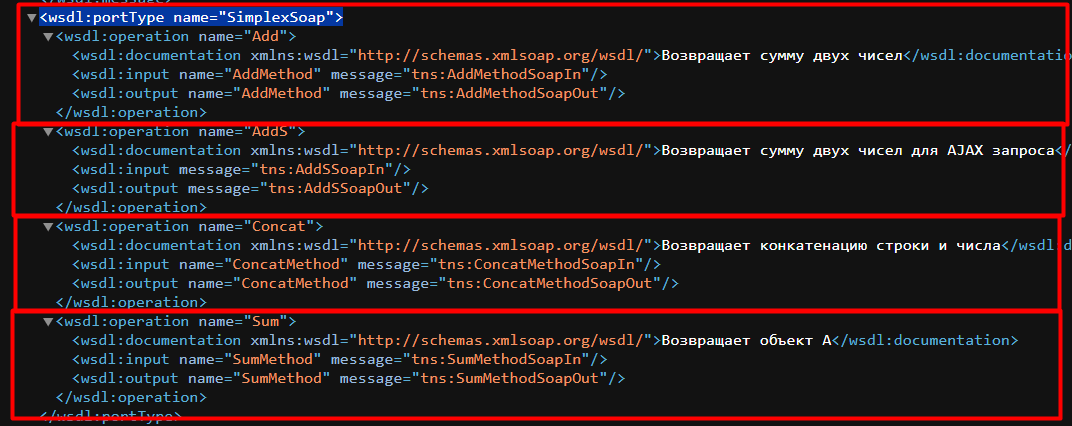
Сверху определяется тип входных параметров: обязательные поля (т.к. минимальное и максимальное количество появления этого параметра задано 1) x и y типа int.

Ниже определяется тип выходных параметров: обязательный AddMethodResult типа int

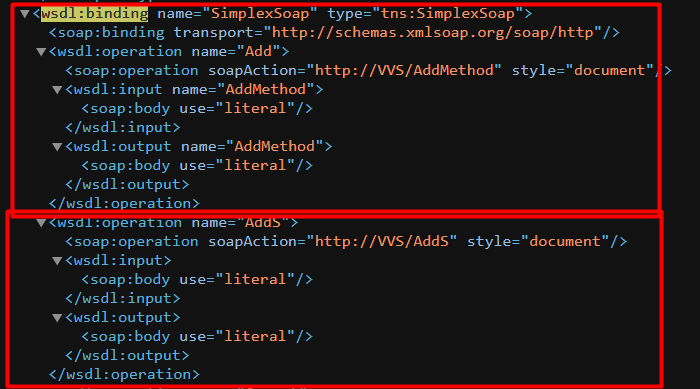
* описание входных и выходных сообщений всех методов сервиса;



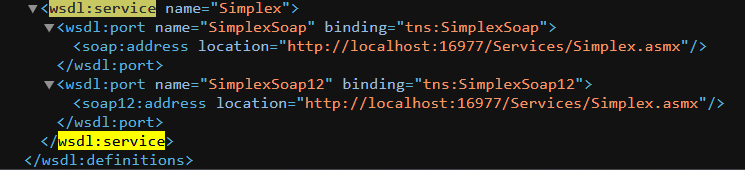
* описание методов сервиса;



* описание протоколов обмена данных;



* + <soap:binding> и <soap12:binding> задают HTTP как транспортный протокол.
  + Стиль обмена данными: document.
  + Тело сообщений кодируется как literal.
* описание сервиса.



### ****1. Разработка ASMX-сервиса с использованием ASP.NET****

#### ****Что такое ASMX-сервис?****

ASMX (Active Server Method Extensions) — это технология в ASP.NET для создания веб-сервисов на основе SOAP (Simple Object Access Protocol). Такие сервисы обрабатывают запросы и возвращают ответы в формате XML.

#### ****Шаги разработки ASMX-сервиса****

1. **Создание проекта**:
   * В Visual Studio выберите **ASP.NET Web Application** или **ASP.NET Empty Web Application**.
   * Добавьте новый файл **ASMX** (например, Simplex.asmx).
2. **Добавление методов**:
   * Методы сервиса описываются в классе, связанного с файлом .asmx.
   * Пример простого метода:

csharp

Копировать код

using System.Web.Services;

[WebService(Namespace = "http://XYZ/")] // Пространство имен

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)] // Поддержка SOAP 1.1

public class Simplex : WebService

{

[WebMethod]

public int Add(int a, int b)

{

return a + b;

}

[WebMethod]

public int Subtract(int a, int b)

{

return a - b;

}

[WebMethod]

public int Multiply(int a, int b)

{

return a \* b;

}

}

1. **Запуск сервиса**:
   * Запустите приложение и перейдите к http://<Ваш-сервер>/Simplex.asmx.
   * Вы увидите список доступных методов и сможете их протестировать.

### ****2. Описание стандарта WSDL****

#### ****Что такое WSDL?****

WSDL (Web Services Description Language) — это XML-документ, описывающий веб-сервис. Он содержит информацию о:

* **Методах сервиса** (операциях),
* **Типах данных**,
* **Точках доступа** (URL-адресах).

#### ****Основные разделы WSDL****:

1. **Types** — описание используемых типов данных.
2. **Messages** — описание входных и выходных данных для каждого метода.
3. **PortType** — набор доступных операций (методов).
4. **Binding** — описание протоколов (SOAP, HTTP и др.).
5. **Service** — конечная точка доступа (URL сервиса).

#### ****Пример структуры WSDL****:

xml

Копировать код

<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"

xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:tns="http://XYZ/"

targetNamespace="http://XYZ/">

<types>

<!-- Описание типов данных -->

</types>

<message name="AddRequest">

<part name="a" type="xsd:int"/>

<part name="b" type="xsd:int"/>

</message>

<message name="AddResponse">

<part name="result" type="xsd:int"/>

</message>

<portType name="SimplexPortType">

<operation name="Add">

<input message="tns:AddRequest"/>

<output message="tns:AddResponse"/>

</operation>

</portType>

<binding name="SimplexBinding" type="tns:SimplexPortType">

<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>

<operation name="Add">

<soap:operation soapAction="http://XYZ/Add"/>

</operation>

</binding>

<service name="SimplexService">

<port name="SimplexPort" binding="tns:SimplexBinding">

<soap:address location="http://localhost/Simplex.asmx"/>

</port>

</service>

</definitions>

#### ****Полезный источник WSDL****:

[W3C WSDL Documentation](https://www.w3.org/TR/wsdl)

### ****3. Протоколы SOAP 1.1 и SOAP 1.2****

#### ****SOAP**** — это протокол для обмена сообщениями между клиентом и сервером.

* Основан на XML.
* Использует стандартные транспортные протоколы (обычно HTTP или HTTPS).

#### ****SOAP 1.1****:

* Простая структура XML.
* Широкая поддержка в старых системах.
* Пространство имен: http://schemas.xmlsoap.org/soap/envelope/.

#### ****SOAP 1.2****:

* Более строгие правила обработки ошибок.
* Улучшенная поддержка различных транспортных протоколов.
* Пространство имен: http://www.w3.org/2003/05/soap-envelope.

#### ****Пример SOAP-запроса (1.2)****:

xml

Копировать код

<soap12:Envelope xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">

<soap12:Body>

<Add xmlns="http://XYZ/">

<a>5</a>

<b>10</b>

</Add>

</soap12:Body>

</soap12:Envelope>

#### ****Полезный источник SOAP****:

[W3C SOAP Documentation](https://www.w3.org/TR/soap/)

### ****4. Установка пространства имен****

Пространство имен — это уникальный URI, который идентифицирует ваш веб-сервис. Он указывается в атрибуте Namespace в классе ASMX-сервиса:

csharp

Копировать код

[WebService(Namespace = "http://XYZ/")]

Это пространство имен используется для генерации WSDL и SOAP-запросов.

### ****5. Методы ASMX-сервиса****

#### ****Задание требует реализации трех методов:****

1. **Add (Сложение)**:

csharp

Копировать код

[WebMethod]

public int Add(int a, int b)

{

return a + b;

}

1. **Subtract (Вычитание)**:

csharp

Копировать код

[WebMethod]

public int Subtract(int a, int b)

{

return a - b;

}

1. **Multiply (Умножение)**:

csharp

Копировать код

[WebMethod]

public int Multiply(int a, int b)

{

return a \* b;

}

Эти методы принимают два числа в качестве входных параметров и возвращают результат математической операции.

### ****Итог****

* Разработка ASMX-сервиса в ASP.NET включает создание методов, генерацию WSDL и поддержку SOAP-запросов.
* Пространство имен добавляет уникальность вашему сервису.
* SOAP 1.1 и 1.2 различаются по структуре, пространствам имен и способу обработки сообщений.
* WSDL служит описанием сервиса, включая методы, типы данных и точки доступа.

Если что-то из этой теории нужно подробнее — напишите! 😊