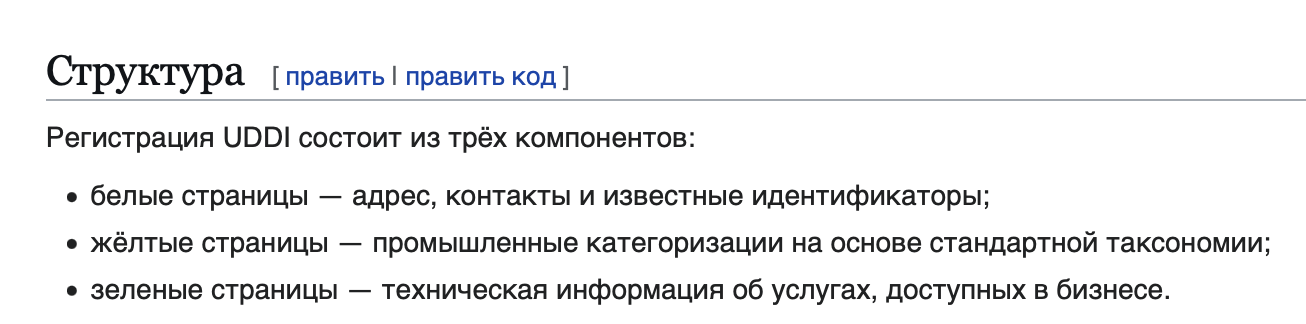
**SOAP –** протокол обмена произвольными сообщениями в формате **XML.  
  
Последняя версия спецификации 1.2**Является расширением протокола XML-RPC  
  
Может использоваться с любым протоколом прикладного уровня SMTP, FTP, HTTP, HTTPS

**SOAP-сервисы можно реализовать с использованием различных технологий, включая:**

- ASMX: это устаревший ASP.NET Web Services, который позволяет создавать и потреблять SOAP-веб-службы. ASMX-сервисы позволяют просто объявлять методы и автоматически генерируют SOAP-сообщения.

- WCF (Windows Communication Foundation): более современная платформа для создания сервисов в .NET, которая поддерживает как SOAP, так и другие протоколы  
  
**Структура SOAP:**  
1) Envelope – корневой элемент, определяет сообщение и пространство имен  
2) Header – содержит атрибуты сообщения  
3) Body – содержит сообщение, которыми обмениваются приложения  
4) Fault – необязательный элемент, который хранит информацию об ошибках  
  
Пример запроса на сервер  
  
  
Пример ответа  
  
  
Недостатки:   
1) Увеличенный объем сообщения  
2) Снижение скорости обработки

**WSDL –** (Web service description language) – язык описания веб-сервисов и доступа к ним, основанный на языке xml.  
  
Последняя спецификация 2.0, версия 1.2 имеет статус заметки  
  
**Документ WSDL можно разбить на следующие части:**  
1) Types Определение типов данных  
2) Message Элементы данных – сообщения используемые web-сервисом  
3) PortType Абстрактные операции – операции которые могут быть выполнены с сообщением  
4) Binding Связывание сервисов – способ доставления сообщения  
5) Service Точка вызова сервиса  
  
Пример:  


UDDI – (Universal Description Discovery & Integration) – инструмент для расположения описаний веб-сервисов (WSDL) для последующего их поиска другими организациями и интеграции в свои системы  
  
UDDI это [кроссплатформенное программное обеспечение](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5" \o "Кроссплатформенное программное обеспечение), основанное на [XML](https://ru.wikipedia.org/wiki/XML" \o "XML). UDDI является открытым проектом, спонсируемым [OASIS](https://ru.wikipedia.org/wiki/OASIS" \o "OASIS), который позволяет организациям публиковать описания [веб-сервисов](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81" \o "Веб-сервис) ([WSDL](https://ru.wikipedia.org/wiki/WSDL" \o "WSDL)) для последующего их поиска другими организациями и интеграции в свои системы, а также определять, как сервисы или приложения взаимодействуют через Internet.  
  
Он предназначен для опроса [SOAP](https://ru.wikipedia.org/wiki/SOAP" \o "SOAP) сообщениями и для обеспечения доступа к [Web Services Description Language](https://ru.wikipedia.org/wiki/Web_Services_Description_Language" \o "Web Services Description Language) (WSDL) документам, описывающим привязки протоколов и форматов сообщений, необходимых для взаимодействия с веб-услугами, перечисленными в его каталоге.  
  
  
  
  
UDDI

UDDI описывается с помощью структуры XML и включает в себя несколько ключевых тегов:

- <businessEntity> – Определяет бизнес-сущность, которая предоставляет услуги.

- <businessKey> – Уникальный идентификатор бизнес-сущности.

- <name> – Имя бизнес-сущности.

- <description> – Описание бизнес-сущности.

- <businessService> – Определяет предоставляемые услуги.

- <serviceKey> – Уникальный идентификатор услуги.

- <name> – Имя услуги.

- <bindingTemplate> – Шаблон привязки (описывает, как к услуге получить доступ).

- <bindingKey> – Уникальный ключ привязки.

- <accessPoint> – Местоположение сервиса (URL).

- <tModelInstanceDetails> – Ссылки на модели типов (технические описания).

- <tModel> – Определяет модель для описания типовых данных.

- <tModelKey> – Уникальный идентификатор модели.

- <name> – Имя модели.

- <description> – Описание модели.

- <categoryBag> – Категории, связанные с моделью.

- <discoveryURLs> – URL-адреса для поиска услуг.

WSDL

WSDL также использует XML и состоит из множества элементов, которые делятся на определенные секции:

- <definitions> – Корневой элемент WSDL-файла.

- targetNamespace – Пространство имён, к которому принадлежит сервис.  
- xmlns:wsdl – Пространство имен для WSDL.

- <types> – Описание типов данных, используемых веб-сервисом (с использованием XML Schema).

- <message> – Определяет сообщения, которые обмениваются между клиентом и сервисом.

- <part> – Определяет часть сообщения (например, параметры, возвращаемые значения).

- <portType> – Определяет операции, доступные в веб-сервисе.

- <operation> – Операции, определенные в portType.

- <input> – Определяет входное сообщение для операции.

- <output> – Определяет выходное сообщение для операции.

- <fault> – Определяет возможные ошибки (исключения).

- <binding> – Определяет связывание (как взаимодействовать с операциями).

- <soap:binding> – Определяет, что используется SOAP.

- <operation> – Связывает операции с конкретным протоколом.

- <input>, <output>, <fault> – Соответствуют операциям в portType.

- <service> – Определяет сам сервис и его доступ.

- <port> – Указывает конкретный порт и связывание.

- <address> – URL-адрес, по которому доступен сервис.  
  
  
**ASMX** — это спецификация для создания веб-сервисов на платформе .NET. Она позволяет разрабатывать и развертывать веб-сервисы с использованием протокола SOAP (Simple Object Access Protocol) и HTTP.

Веб-сервисы ASMX можно реализовать двумя основными способами:

1. **Создание ASMX веб-сервиса через Visual Studio**:

- **Шаги**:

1. Откройте Visual Studio и создайте новый проект.

2. Выберите шаблон "Веб-приложение" (или "Web Application") на платформе ASP.NET.

3. Добавьте новый элемент (Add New Item) и выберите "Web Service" (ASMX).

4. Определите методы в вашем веб-сервисе, пометив их атрибутом [WebMethod]. Эти методы будут доступны для вызова через SOAP.

5. Опубликуйте веб-сервис на сервере или хостинге.

- **Пример**:

using System.Web.Services;

[WebService(Namespace = "http://example.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

public class MyService : WebService

{

[WebMethod]

public string HelloWorld()

{

return "Hello, World!";

}

}

2. **Использование WSDL для создания клиентской части**:

- **Шаги**:

1. После создания ASMX веб-сервиса, вы можете получить его WSDL (Web Services Description Language) по URL-адресу http://yourserver/yourservice.asmx?WSDL.

2. С помощью WSDL можно сгенерировать клиентский код, который может вызывать методы веб-сервиса. Это можно сделать с помощью инструмента wsdl.exe, который входит в поставку .NET Framework, или с использованием "Add Service Reference" в Visual Studio.

- **Пример**:

wsdl http://yourserver/yourservice.asmx?WSDL

- После выполнения этой команды у вас будет сгенерирован код клиента, который можно использовать для обращения к веб-сервису.

**Параметр Description атрибута WebService используется для предоставления описания самого веб-сервиса. Основная цель описания — помочь пользователям понять, что делает ваш веб-сервис, и какие функции он предлагает.**

**Параметр Description атрибута WebMethod используется для предоставления информации о каждом отдельном методе веб-сервиса. Помогает пользователям веб-сервиса понять функциональность каждого метода, делая API более самодокументируемым.  
  
MessageName атрибута WebMethod**

**Параметр MessageName атрибута WebMethod позволяет задать имя сообщения для SOAP. Полезно для создания более понятных и семантически значимых имен для операций, особенно когда несколько методов имеют схожие названия.**