

Основы **WSDL**

Лекция № 6

В ходе лекции изучаются следующие темы:

- основные положения WSDL;
- структура WSDL-документа;
- элементы Types и Message;

Основные положения **WSDL**

Тема № 1

Web Services Description Language – XML-

ориентированный язык описания **web**-сервисов и
доступа к ним

Определение WSDL

В **WSDL**, абстрактное определение конечных точек и сообщений сервиса отделено от конкретных особенностей развертывания сети или привязок к формату данных. Это дает возможность использовать абстрактные определения сообщений (абстрактными описания данных в процессе обмена) и портов (абстрактные наборы операций)

Основная особенность WSDL

WSDL признает необходимость множества типов данных для описания форматов сообщений, поддерживает спецификацию **XML-схемы (XSD)** в качестве канонической системы типов. Однако, **WSDL** позволяет использовать и другие языки определения типов как расширения

Поддержка типов данных

WSDL определяет общий механизм связывания, позволяющий прикрепить конкретный формат данных, протокол или структуру данных к абстрактному сообщению, операции или конечной точке. Кроме того, **WSDL** позволяет специфические расширения связывания для протоколов **SOAP 1.1, HTTP GET / POST** и **MIME**

Назначение WSDL

Ключевые слова **WSDL** (описаны в **RFS-2119**):

- MUST;
- MUST NOT;
- REQUIRED;
- SHALL;
- SHALL NOT;
- SHOULD;
- SHOULD NOT;
- RECOMMENDED;
- MAY;
- OPTIONAL.

*Структура **WSDL**-документа*

Тема № 2

WSDL-документ представляет собой упорядоченный набор определений

Структуризация WSDL-документа

Основные элементы **WSDL**-документа:

- Documents – контейнер для определения документов, доступных для прочтения человеком;
- Types - контейнер для определения типов данных, использующих некоторый тип системы (например, XSD);
- Message-абстрактное, описываемое в том числе через введенные типы определение передаваемых данных;
- Operation - абстрактное описание какого-либо действия, поддерживаемого сервисом;
- Port Type - абстрактный набор операций, поддерживаемый одной или несколькими конечными точками;
- Binding – описание конкретного протокола и спецификаций формата данных для определенного Port Type;
- Port – описание одной конечной точки, которое определяется как сочетание элемента Binding и сетевого адреса;
- Service – описание сервиса как набора взаимосвязанных конечных точек (портов).

Пример структуры **WSDL**-документа:

```
<wsdl:definitions name="nmtoken"? targetNamespace="uri"?>
  <import namespace="uri" location="uri"/>*
  <wsdl:documentation .... /> ?
  <wsdl:types> ?
    <wsdl:documentation .... />?
    <xsd:schema .... />*
    <!-- extensibility element --> *
  </wsdl:types>
  <wsdl:message name="nmtoken"> *
    <wsdl:documentation .... />?
    <part name="nmtoken" element="qname"? type="qname"?/> *
  </wsdl:message>
```

Пример структуры **WSDL**-документа:

```
<wsdl:portType name="nmtoken">*
  <wsdl:documentation .... />?
  <wsdl:operation name="nmtoken">*
    <wsdl:documentation .... /> ?
    <wsdl:input name="nmtoken"? message="qname">?
      <wsdl:documentation .... /> ?
    </wsdl:input>
    <wsdl:output name="nmtoken"? message="qname">?
      <wsdl:documentation .... /> ?
    </wsdl:output>
    <wsdl:fault name="nmtoken" message="qname"> *
      <wsdl:documentation .... /> ?
    </wsdl:fault>
  </wsdl:operation>
</wsdl:portType>
```

Пример структуры **WSDL**-документа:

```
<wsdl:binding name="nmtoken" type="qname">*
  <wsdl:documentation .... />?
  <!-- extensibility element --> *
  <wsdl:operation name="nmtoken">*
    <wsdl:documentation .... /> ?
    <!-- extensibility element --> *
    <wsdl:input> ?
      <wsdl:documentation .... /> ?
      <!-- extensibility element -->
    </wsdl:input>
    <wsdl:output> ?
      <wsdl:documentation .... /> ?
      <!-- extensibility element --> *
    </wsdl:output>
    <wsdl:fault name="nmtoken"> *
      <wsdl:documentation .... /> ?
      <!-- extensibility element --> *
    </wsdl:fault>
  </wsdl:operation>
</wsdl:binding>
```

Пример структуры **WSDL**-документа:

```
<wsdl:service name="nmtoken"> *
  <wsdl:documentation .... />?
  <wsdl:port name="nmtoken" binding="qname"> *
    <wsdl:documentation .... /> ?
    <-- extensibility element -->
  </wsdl:port>
  <-- extensibility element -->
</wsdl:service>

<-- extensibility element --> *

</wsdl:definitions>
```

Элементы **Types** и **Message**

Тема № 3

Элемент **Types** содержит определения типов данных, которые используются для описания передаваемых сообщений. **WSDL** предпочитает использование **XSD** как канонической системы типов

Назначение элемента
Types

Рекомендации по кодированию абстрактных типов через **XSD** для **WSDL**:

- используйте формы элементов, а не атрибуты;
- не применять атрибуты или элементы, присущие общему кодированию и не имеющие ничего общего с абстрактным содержанием сообщения (некоторые примеры: soap:root, soap:encodingStyle, xmi:id, xmi:name);
- типы массивов должны основываться на типах массивов, определенных в схеме кодирования SOAP v1.1 (<http://schemas.xmlsoap.org/soap/encoding/>) независимо от того, использует ли на самом деле форма элементов кодировку, указанную в разделе 5 описания SOAP v1.1;

Рекомендации по кодированию абстрактных типов через **XSD** для **WSDL**:

- используйте имя `ArrayOfXXX` для типов `Array` (где `XXX` является тип элементов в массиве);
- тип элементов массива и размерность массива задаются с помощью значений по умолчанию для `soapenc:arrayType`;
- используйте тип `xsd:anyType` для представления поля / параметра, который может иметь любой тип.

Пример заголовка **WSDL**-документа:

```
<?xml version="1.0"?>
<definitions name="StockQuote"
  targetNamespace="http://example.com/stockquote.wsdl"
  xmlns:tns="http://example.com/stockquote.wsdl"
  xmlns:xsd="http://example.com/stockquote.xsd"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">
```

Пример описания элемента **Types WSDL-** документа:

```
<types>
  <schema targetNamespace="http://example.com/stockquote.xsd"
    xmlns="http://www.w3.org/2000/10/XMLSchema">
    <element name="TradePriceRequest">
      <complexType>
        <all>
          <element name="tickerSymbol" type="string"/>
        </all>
      </complexType>
    </element>
    <element name="TradePrice">
      <complexType>
        <all>
          <element name="price" type="float"/>
        </all>
      </complexType>
    </element>
  </schema>
</types>
```

Синтаксис описания элемента **Message:**

```
<definitions .... >  
  <message name="nmtoken" > *  
    <part name="nmtoken" element="qname"? type="qname"?/> *  
  </message>  
</definitions>
```

Назначение атрибутов описания **Message**:

- `message name` – используется для обозначения наименования сообщения, уникального среди наименований всех сообщений данного WSDL-документа;
- `part name` – используется для обозначения наименования части сообщения, уникального среди наименований всех частей данного сообщения WSDL-документа.

Пример элемента **Message WSDL-** документа:

```
<message name="GetLastTradePriceInput">  
  <part name="body" element="xsd1:TradePriceRequest"/>  
</message>
```

```
<message name="GetLastTradePriceOutput">  
  <part name="body" element="xsd1:TradePrice"/>  
</message>
```


Если содержание сообщения является достаточно сложным, такое сообщение можно задать через создание соответствующего композитного типа

**Альтернативный
синтаксис задания типов
для Message**

Пример альтернативного описания **Message**:

```
<complexType name="Composite">
  <choice>
    <element name="PO" minOccurs="1" maxOccurs="1" type="tns:POType"/>
    <element name="Invoice" minOccurs="0" maxOccurs="unbounded" type="tns:InvoiceType"/>
  </choice>
</complexType>
</schema>
</types>

<message name="PO">
  <part name="composite" type="tns:Composite"/>
</message>
</definitions>
```

Элемент **Port Types**

Тема № 4

Пример синтаксиса элемента **Port** **Types WSDL**-документа:

```
<wsdl:definitions .... >  
  <wsdl:portType name="nmtoken">  
    <wsdl:operation name="nmtoken" .... /> *  
  </wsdl:portType>  
</wsdl:definitions>
```

Основные шаблоны операций элемента

Port Types:

- One-way - конечная точка получает сообщение;
- Request-response - конечная точка получает сообщение и отправляет коррелированное сообщение;
- Solicit-response - конечная точка посылает сообщение, и получает коррелированное сообщение;
- Notification - конечная точка посылает сообщение.

Пример описания **One-way** операции:

```
<wsdl:definitions .... > <wsdl:portType .... > *  
  <wsdl:operation name="nmtoken">  
    <wsdl:input name="nmtoken"? message="qname"/>  
  </wsdl:operation>  
</wsdl:portType >  
</wsdl:definitions>
```

Пример описания **Request-response** операции:

```
<wsdl:definitions .... >
  <wsdl:portType .... > *
    <wsdl:operation name="nmtoken" parameterOrder="nmtokens">
      <wsdl:input name="nmtoken"? message="qname"/>
      <wsdl:output name="nmtoken"? message="qname"/>
      <wsdl:fault name="nmtoken" message="qname"/>*
    </wsdl:operation>
  </wsdl:portType >
</wsdl:definitions>
```

Элементы описания

Request-response операции:

- `input` – описание входного message;
- `output` - описание выходного message;
- `fault` – описание сообщения об ошибке.

Пример описания **Solicit-response** операции:

```
<wsdl:definitions .... >
  <wsdl:portType .... > *
    <wsdl:operation name="nmtoken" parameterOrder="nmtokens">
      <wsdl:output name="nmtoken"? message="qname"/>
      <wsdl:input name="nmtoken"? message="qname"/>
      <wsdl:fault name="nmtoken" message="qname"/>*
    </wsdl:operation>
  </wsdl:portType >
</wsdl:definitions>
```

Пример описания **Notification** операции:

```
<wsdl:definitions .... >
  <wsdl:portType .... > *
    <wsdl:operation name="nmtoken">
      <wsdl:output name="nmtoken"? message="qname"/>
    </wsdl:operation>
  </wsdl:portType >
</wsdl:definitions>
```

Перечень ссылок

- Web Services Description Language (WSDL) 1.1 [Электронный ресурс] // Сайт W3C Note. – Режим доступа: <http://www.w3.org/TR/wsdl>. - Заголовок с экрана.