Web-сервисы

**Веб-сервис**  - это программно-аппаратная система, доступная посредством сети и использующая стандартизованный протокол сообщений

**Веб-сервис**   – отдельные независимые приложения многократного использования, которые представляют свои функции через Web-интерфейс. Коммуникации с веб-сервисами может выполняться с помощью различных транспортных протоколов, таких как HTTP, HTTPS, FTP, SMTP.

Любой клиент и любой сервер (потребители) могут использовать службы независимо от языка их реализации и устройства на котором они установлены. Web-службы основаны на открытых стандартах (используется XML), ими легко овладеть, и эти стандарты широко поддерживаются на всех платформах Unix и Windows.

Web-службы позволяют приложениям или другим Web-службам совместно использовать данные и функции таким способом, при котором не имеет значения, как именно эти приложения выполняются, какую платформу, операционную систему или устройство они используют.

Архитектура Web-сервисов

**XML** (eXtensible Markup Language) – язык разметки документов, обеспечивающий текстовый формат хранения данных.

**SOAP** (Simple Object Access Protocol) – простой протокол доступа к объектам. Основан на XML для дистанционного вызова процедур по Intranet и Internet. Определяет формат запроса и параметров, передаваемых в запросе.

 **WSDL** (Web Service Description Language) – протокол описания Web-служб. Он позволяет предоставить описание и расположение всех методов Web-службы, а также их параметры на XML.

 **UDDI** (Universal Description, Discovery, and Integration) – универсальное описание, обнаружение и интеграция. Это открытый системный реестр, предназначенный для хранения информации о Web-службах. UDDI доступен по адресу www.uddi.org



Все спецификации, используемые в технологии, основаны на [XML](http://www.4stud.info/web-programming/xml.html) и, соответственно, наследуют его преимущества (структурированность, гибкость и т.д.) и недостатки (громоздкость, медлительность).

SOAP 1.2

SOAP – это протокол обмена сообщениями XML-формата в распределенной среде. Протокол SOAP является соглашением о структуре XML-документов, представляющих участвующие в обмене сообщения, которое позволяет передавать как символьную, так и двоичную информацию. Символьная информация может передаваться в теле SOAP-сообщения, а двоичная информация может быть прикреплена к SOAP-сообщению.

SOAP является самой главной частью технологии Web-сервисов. Он осуществляет перенос данных по сети из одного места в другое.

SOAP обеспечивает доставку данных веб-сервисов. Он позволяет отправителю и получателю XML-документов поддерживать общий протокол передачи данных, что обеспечивает эффективность сетевой связи.

SOAP – это базовая однонаправленная модель соединения, обеспечивающая согласованную передачу сообщения от отправителя к получателю, потенциально допускающая наличие посредников, которые могут обрабатывать часть сообщения или добавлять к нему дополнительные элементы. Спецификация SOAP содержит соглашения по преобразованию однонаправленного обмена сообщениями в соответствии с принципом «запрос/ответ», а также определяет как осуществлять передачу всего XML-документа.

Как видно из рис.1 SOAP предназначен для поддержания независимого абстрактного протокола связи, обеспечивающего коммуникацию двух и более приложений, сайтов, предприятий и т.п., реализованных на разных технологиях и аппаратных средств.



Общая структура SOAP-сообщения

SOAP-сообщение представляет собой XML-документ; сообщение состоит из трех основных элементов: конверт (SOAP Envelope), заголовок (SOAP Header) и тело (SOAP Body).



**SOAP-ENV:Envelope**

**SOAP-ENV:Header**

**SOAP-ENV:Body**

Пример SOAP сообщения:

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://www.w3.org/2003/05/soap-envelope" xmlns:t="www.example.com">

 <SOAP-ENV:Header>

 </SOAP-ENV:Header>

 <SOAP-ENV:Body>

 <t:CurrentDate>

 <Year>2011</Year>

 <Month>February</Month>

 <Day>12</Day>

 <Time>18:02:00</Time>

 </t:CurrentDate>

 </SOAP-ENV:Body>

</SOAP-ENV:Envelope>

Конверт (SOAP Envelope)

Является самым «верхним» элементом SOAP сообщения. Содержит корневой элемент XML-документа. Описывается с помощью элемента Envelope с обязательным пространством имен http://www.w3.org/2003/05/soap-envelope.

У элемента Envelope могут быть атрибуты xmlns, определяющие пространства имен, и другие атрибуты, снабженные префиксами.

Envelope может иметь необязательный дочерний элемент Header c тем же пространством имен — заголовок. Если этот элемент присутствует, то он должен быть первым прямым дочерним элементом конверта.

Следующий дочерний элемент конверта должен иметь имя Body и то же самое пространство имен - тело. Это обязательный элемент и он должен быть вторым прямым дочерним элементом конверта, если есть заголовок, или первым — если заголовка нет.

Заголовок SOAP (SOAP Header)

Первый прямой дочерний элемент конверта. Не обязательный. Заголовок кроме атрибутов xmlns может содержать 0 или более стандартных атрибутов:

* encodingStyle
* role
* mustUnderstand
* relay

Атрибут encodingStyle

В SOAP-сообщениях могут передаваться данные различных типов (числа, даты, массивы, строки и т.п.). Определение этих типов данных выполняется в схемах XML (обычно — XSD). Типы, определенные в схеме, заносятся в пространство имен, идентификатор которого служит значением атрибута encodingStyle.

Стандартное пространство имен, в котором расположены имена типов данных SOAP <http://www.w3.org/2003/05/soap-encoding>

Атрибут mustUnderstand

Тип данных — boolean. Если он имеет значение true, обязывает обрабатывать данный блок заголовка. Если обработчик не понял данных блок заголовка, тогда он генерирует сообщение об ошибке и обработка SOAP-сообщения прерывается. Если значение атрибута false, тогда при возникновении ошибки обработка SOAP-сообщения не прерывается и блок заголовка игнорируется.

Атрибут relay

Тип данных — boolean. Показывает, что заголовочный блок, адресованный SOAP-посреднику, должен быть передан дальше, если он небыл обработан. Необходимо отметить, что если заголовочный блок обработан, правила обработки SOAP требуют, чтобы он был удален из уходящего сообщения. По умолчанию, необработанный заголовочный блок, предназначенный роли, которую исполняет SOAP-посредником, должен быть удален перед отправкой сообщения.

Тело SOAP (SOAP Body)

Элемент Body обязательно записывается сразу за элементом Header, если он есть в сообщении, или первым в SOAP-сообщении, если заголовок отсутствует. В элемент Body можно вложить произвольные элементы, спецификация никак не определяет их структуру.

Определен только один стандартный элемент, который может быть в теле сообщения -Fault, содержащий сообщение об ошибке.

Обработка ошибок в SOAP-сообщениях

Если SOAP-сервер, обрабатывая поступившее SOAP-сообщение, обнаружит ошибку, то он прекратит обработку и отправит клиенту SOAP-сообщение, содержащее один элемент Fault с сообщением об ошибке.

Элемент < Fault > содержит следующие дочерние элементы:

* Обязательный элемент <Code>. Он содержит обязательный вложенный элемент <Value> с кодом ошибки и необязательный вложенный элемент <Subcode>, также содержащий элемент <Value> с уточняющим кодом ошибки и элемент <Subcode>, и далее все повторяется рекурсивно.
* Обязательный элемент <Reason> – описывает возникшую ошибку с помощью дочерних элеменов <Text>, содержащих текстовую информацию об ошибки. Каждый элемент <Text> должен именть свое значение обязательного атрибута lang
* Необязательный элемент <Node>— адрес URI промежуточного SOAP-узла, заметившего ошибку.
* Необязательный элемент <Role> - указывает роль SOAP-узла, заметившего ошибку.
* Необязательный элемент <Detail> — содержит описание ошибки, замеченной при обработке тела Body послания, но не его заголовка.

В следующем примере показана ошибка, возникающая при попытке выполнения процедуры — имена аргументов процедуры неправильно записаны.

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

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

 xmlns:rpc="http://www.w3.org/2002/06/soap-rpc1">

 <env:Body>

 <env:Fault>

 <env:Code>

 <env:Value>env:Sender</env:Value>

 <env:Subcode>

 <env:Value>rpc:BadArguments</env:Value>

 </env:Subcode>

 </env:Code>

 <env:Reason>Processing Error</env:Reason>

 <env:Detail>

 <e:myfaultdetails

 xmlns:e="http://www.example.org/faults">

 <message>Name does not match</message>

 <errorcode>999</errorcode>

 </e:myfaultdetails>

 </env:Detail>

 </env:Fault>

 </env:Body>

</env:Envelope>

WSDL

Спецификация WSDL определяет язык Web Services Description Language – XML-язык описания Web-сервисов. WSDL-описание Web-сервиса содержит имя и адрес сервиса, указывает способ и протокол взаимодействия с сервисом, а так же информацию о политиках сервиса, определяющих условия взаимодействия с ними.

Язык WSDL описывает Web-сервисы распределенной среды как набор конечных точек, обменивающихся сообщениями. WSDL-описаниеWeb-сервисов документирует такую распределенную среду и автоматизирует процессы коммуникации в ней.

Язык WSDL описывает Web-сервис на друх уровнях:

Абстрактный. На этом уровне Web-сервис описывается в терминах сообщений, которые он получает и посылает. Абстактный уровень содержит WSDL-элемент <interface>, определяющий операции и сообщения Web-сервиса.

Детальный (конкретный). На этом уровне определяются детали протокола обмена сообщениями и детали формата сообщений, адреса сервиса и его функциональность. Детальный уровень содержит WSDL- элементы <binding>, <endpoint> и <service>



Концептуальная модель WSDL



| **Элемент WSDL 1.1** | **Элемент WSDL 2.0** | **Краткое описание** |
| --- | --- | --- |
| PortType | Interface | Представляет описание интерфейса веб-сервиса (список операций и их параметров). |
| Service | Service | Список системных функций |
| Binding | Binding | Специфицирует интерфейсы и задает параметры связывания с протоколом SOAP: стиль связывания (RPC/Document) и транспорт (SOAP). Эта секция доступна и для каждой из операций |
| Operation | Operation | Определяет операцию, представляемую веб-сервером. WSDL-операция — это аналог традиционным функциям и процедурам. |
| Message | не использ. | Сообщение, связанное с определенной операцией. Содержит информацию, необходимую для выполнения данной операции. Каждое сообщение может состоять из нескольких логических частей, описывающих типы данных и имена атрибутов. В версии 2.0 было исключено, т.к. была внедрена поддержка XML Schema для всех элементов. |
| Types | Types | Описание данных в соответствии с XML Schema. |

definitions

Элемент **definitions** является корневым элементов любого документа WSDL. Он используется в качестве контейнера, в котором содержится вся необходимая информация о данной услуге и ее атрибутах. На рисунке 2 приведена схема элемента **definitions**. Атрибут **targetNamespace** этого элемента является обязательным атрибутом типа **anyURI**. Это пространство имен может напрямую или косвенно определять семантику WSDL. Кроме того у элемента **definitions** могут быть другие необязательные атрибуты, соответствующие различным пространствам имен, которые могут использоваться в документе WSDL.


**Рис. 2.**Схема элемента definitions

include

Элемент **include** предназначен для разделения описаний Web-сервиса на модули - различные компоненты описаний сервисов из одного и того же пространства имен могут находиться в другом документе WSDL, которой можно использовать в описаниях Web-сервисов. Атрибут location является обязательным, он задает нахождение этих документов WSDL. Фактическое значение пространства имен добавляемого документа WSDL должно соответствовать целевому пространству имен элемента **definitions** в документе WSDL, в который добавляется первый документ. На рисунке 3 приведена схема элемента **include**.


**Рис. 3.**Схема элемента include

import

По своему назначению элемент **import** очень похож на элемент **include**, с тем исключением, что импортированный документ WSDL может относиться к другому целевому пространству имен. Атрибут**namespace** этого элемента является обязательным, а атрибут **location** - необязательным. На рисунке 4 приведена схема элемента **import**.


**Рис. 4.**Схема элемента import

types

Элемент **types** определяет типы данных, которые используются при обмене сообщениями. Язык WSDL использует W3C XML-схему в качестве предпочтительного языка схемы. WSDL может использовать и другие системы, например, DTD и RELAX NG. Чтобы воспользоваться схемами, их необходимо импортировать или внедрять в элементе **types** документа WSDL. Для импортирования используется конструкция **xs:import**, а для внедрения - **xs:schema**. Импортированные или внедренные компоненты схемы в документе WSDL указываются по **QName**. На рисунке 5 приведена схема элемента **types**.


**Рис. 5.**Схема элемента types

interface

Элемент **interface** содержит поименованный набор абстрактных операций и сообщений. При необходимости он может расширять один или несколько других элементов **interface**. Элементы **interface** в других компонентах, как, например, **binding**, указываются по **QName**. Элемент **interface** содержит элемент**operation**, атрибуты **name** и **pattern** которого является обязательными, а **style** - необязательным. На рисунке 5 приведена схема элемента **interface**. Элементы **feature** определяют функциональные возможности, связанные с обменом сообщениями между общающимися сторонами, что включает сведения о надежности, безопасности, зависимостях и маршрутизации. Элемент **property** используется для управления поведением элемента **feature**. Ряд возможных и допустимых значений этого элемента задается указаниями на описание схемы. Эти значения могут использоваться в нескольких элементах**feature**.


**Рис. 6.**Схема элемента interface

binding

Элемент **binding** определяет базовый транспорт и формат передачи для сообщений. Каждый элемент**binding** в документе WSDL указывает на элемент **interface**. Все элементы **operation**, определенные в элементе **interface**, должны быть связаны с этим **binding**. Элемент **endpoint** в компоненте **service**указывает элемент **binding**. И элементы **endpoint**, и элементы **binding** созданы для обеспечения гибкости и прозрачности местоположения. Многочисленные элементы **endpoint** с различным сетевым адресом могут использовать одну и ту же протокол, определенный в **binding**. В спецификации "WSDL 2.0, Связывания" определяются расширения элемента **binding** для таких протоколов и форматов сообщений, как SOAP, HTTP и MIME. На рисунке 7 приведена схема элемента **binding**.


**Рис. 7.**Схема элемента binding

service

Элемент **service** описывает набор элементов **endpoint**, которые указывают на одиночный сетевой адрес для элемента **binding**. Вся другая информация о протоколе сдержится в элементе **binding**. К элементу**service** можно обратиться по **Qname**. У этого элемента есть обязательные атрибуты **name** и **interface**. На рисунке 8 приведена схема элемента **service**.


**Рис. 8.**Схема элемента service

Шаблоны обмена сообщениями WSDL

Шаблоны обмена сообщениями определяют последовательность и число сообщений в одной операции (элементе **operation**). В спецификации Web Services Description Language (WSDL) Version 2.0 Part 2: Message Patterns ("Язык описания Web-сервисов (язык WSDL), версия 2.0, часть 2: Шаблоны сообщений") описывается несколько типов шаблонов сообщений. Шаблоны обмена сообщениями используют правила генерации неисправностей, предназначенные для оповещения о возникновении неисправностей. Обмен сообщениями может быть прерван, если генерация неисправностей происходит независимо от набора стандартных правил. Следующий набор стандартных правил кратко описывает поведение при возникновении ошибок.

* ошибка замещает сообщение (Fault Replaces Messages);
* сообщение вызывает ошибку (Message Triggers Fault);
* ошибок нет (No Faults).

На рисунке 9 показаны эти различные шаблоны обмена сообщениями со схемами обработки ошибок.


**Рис. 9.**Шаблоны обмена сообщениями WSDL

Описание сервиса, предназначенного для передачи информации о котировках акций, на WSDL 1.1 и WSDL 2.0

Источники литературы

<http://citforum.ru/internet/xml/wsdl2_0/>

<http://raleigh.ru/XML/w3schools/wsdl/>

<http://ru.wikipedia.org/wiki/WSDL>

<http://portal.odines.ws/?p=2292>

http://www.webmascon.com/topics/technologies/8a.asp