

## **АРХІТЕКТУРА XML ВЕБ-СЕРВІСІВ. SOAP.**

Застосування сервісно-орієнтованих підходів дозволяє говорити про повторне використання (reuse) на макрорівні (рівні сервісів), на відміну від мікрорівня (рівня об'єктів). Сервісно-орієнтований підхід передбачає використання простих і загальноприйнятих стандартів, що дозволяє самим різним програмам використовувати функціональність один одного.

XML Web-сервіс - це особливий тип web-додатки, який:

- розгортається на web-сервері;
- публікує web-методи, які можуть бути викликані зовнішніми клієнтами;
- очікує надходження HTTP-запитів, які є командами викликів web-методів;
- виконує web-методи і повертає результати.

Web-сервіс може надавати в реальному часі інформацію по курсах акцій, перевіряти кредитні карти або повідомляти прогноз погоди. Web-сервіси настільки ж різноманітні, як і звичайні програми.

Існує чимало протоколів і технологій віддаленого виклику: Microsoft Distributed Component Object Model (DCOM), the Object Management Group's Common Object Request Broker Architecture (CORBA), Sun's Remote Method Invocation (RMI), .NET Remoting, XML Web Services.

Всі ці компонентно-орієнтовані технології (DCOM, CORBA і RMI) довгі роки успішно застосовувалися в Intranet-додатках. Вони забезпечують надійну, масштабовану архітектуру. Однак при використанні цих технологій в Internet виникають проблеми. Вони погано взаємодіють між собою. Всі технології оперують об'єктами, але істотно відрізняються деталями: управлінням життєвим циклом, підтримкою конструкторів і ступенем підтримки успадкування.

XML Web Service - компонент, що надає Internet-клієнтам набір функцій API або web-методів. XML входить в назву, оскільки web-сервіси і їх клієнти використовують його для обміну даними. В основі web-сервісів лежать відкриті стандарти, такі як HTTP, XML (Extensible Markup Language), SOAP (Simple Object Access Protocol - стандарт Internet, що описує, як додатки можуть взаємодіяти, тобто викликати методи один одного, за допомогою HTTP і інших протоколів).

Веб-служби можуть взаємодіяти один з одним і зі сторонніми додатками за допомогою повідомлень, заснованих на певних протоколах (SOAP, XML-RPC, REST і т. Д.). Веб-служба є одиницею модульності при використанні сервіс-орієнтованої архітектури додатку.

Веб-сервіси (або веб-служби) - це технологія. І як і будь-яка інша технологія, вони мають досить чітко окреслену середу застосування.

На сьогоднішній день найбільшого поширення набули наступні протоколи реалізації веб-сервісів:

- SOAP (Simple Object Access Protocol) - по суті це трійка стандартів SOAP / WSDL / UDDI
- REST (Representational State Transfer)
- XML-RPC (XML Remote Procedure Call)

Насправді, SOAP стався від XML-RPC і є наступним етапом його розвитку. У той час як REST - це концепція, в основі якої лежить швидше архітектурний стиль, ніж нова технологія, що базується на теорії маніпуляції об'єктами CRUD (Create Read Update Delete) в контексті концепцій WWW.

Якщо подивитися на веб-сервіси в розрізі стека мережевих протоколів, ми побачимо, що це, в класичному випадку, не що інше, як ще одна надбудова над протоколом HTTP.

WSDL (Web Services Description Language). Правила, за якими складаються повідомлення для веб-сервісу описуються так само за допомогою xml і також мають чітку структуру. Тобто якщо веб-сервіс надає можливість виклику якогось методу, він повинен дати можливість клієнтам дізнатися які параметри для даного методу використовуються. Якщо веб-сервіс чекає рядок для методу Method1 як параметр і рядок повинна мати ім'я Param1, то в описі веб-сервісу ці правила будуть вказані.

SOAP (від англ. Simple Object Access Protocol - простий протокол доступу до об'єктів; аж до специфікації 1.2) - протокол обміну структурованими повідомленнями в розподіленій обчислювальній середовищі. Спочатку SOAP призначався в основному для реалізації віддаленого виклику процедур (RPC). Зараз протокол використовується для обміну довільними повідомленнями в форматі XML, а не тільки для виклику процедур. Офіційна специфікація останньої версії 1.2 протоколу Ніяк не розшифровує назву SOAP. SOAP є розширенням протоколу XML-RPC.

SOAP може використовуватися з будь-яким протоколом прикладного рівня: SMTP, FTP, HTTP, HTTPS і ін. Однак його взаємодія з кожним із цих протоколів має свої особливості, які повинні бути визначені окремо. Найчастіше SOAP використовується поверх HTTP.

Для використання SOAP необхідно знайти відповідний сервіс та закачати файл з розширенням .wsdl (рис.1)

```

<?xml version="1.0"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://litesms.net/sms_soap.php"
  xmlns:soap-env="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://litesms.net/sms_soap.php">
<types>
  <xsd:schema>
    <xsd:complexType name="array">
      <xsd:complexContent>
        <xsd:restriction base="soapenc:array">
          <xsd:attribute ref="soapenc:arrayType" arrayType="tns:mixed[]"/>
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="mixed">
      <all>
        <element name="varString" type="xsd:string"/><element name="varInt" type="xsd:int"/>
        <element name="varFloat" type="xsd:float"/><element name="varArray" type="tns:array"/>
      </all>
    </xsd:complexType>
  </xsd:schema>
</types>
<message name="sendSmsInput">
  <part name="message" type="xsd:string"/>
  <part name="to" type="xsd:string"/>
  <part name="from" type="xsd:string"/>
  <part name="translit" type="xsd:string"/>
</message>
<message name="getBalanceInput"/>
<message name="getStatusInput">
  <part name="message_id" type="xsd:int"/>
</message>
<message name="sendSmsOutput">
  <part name="return" type="xsd:string"/>
</message>

```

Рис. 1. Структура wsdl файлу

SOAP клієнт для обраного сервера має наступний вид (рис.2.)

```

<?php
require_once('creds.php');
ini_set("soap.wsdl_cache_enabled","1");
$client = new SoapClient("http://litesms.net/sms_soap.php?wsdl",
    array( 'login' =>$login,
          'password' =>$password,
          'trace' =>true
        ));

try{
    echo '<br/>';
    //Получаем статус сообщения
    echo $client->getStatus(50996). '<br />';
    echo $client->sendSMS("Privet, Webguru", "380504634646", "prog-school.ru"). "<br/>";
    echo $client->getBalance(). "<br/>";
    var_dump($client->__getLastRequest());
    var_dump($client->__getLastRequestHeaders());
    var_dump($client->__getLastResponse());
    var_dump($client->__getLastResponseHeaders());
    //Получение типов данных объявленных в wsdl
    var_dump($client->__getTypes());
    // Установка Cookie
    $client->__setCookie('cookie_name', 'cookie_value');
    var_dump($client->__soapCall("getStatus", array(50996)));
    //Установка SOAP заголовков
    $headers=array();
    $headers[]=new SoapHeader('http://soapinterop.org/echoheader/',
        'echoMeStringRequest', 'hello word');
    $headers[]=new SoapHeader('http://soapinterop.org/echoheader/',
        'echoMeStringRequest', 'hello word again');
    $client->__setSoapHeaders($headers);
    $client->getStatus(50996). "<br />";
    var_dump($client->__getLastRequest());
    //Вызов функции
    var_dump($client->__soapCall("getStatus", array(50996)));}

```

Рис. 2. SOAP клієнт

Також можна написати власний SOAP сервер. Типовий SOAP сервер пишімо на PHP (рис.3.)

```

<?php
class MyMix {
    public $varString;
    public $varFloat;
    public $varInt;
}

$server = new SoapServer( "sms_ex.wsdl",
    array( 'classmap' => array( 'mixed' => "MyMix" ) ) );
//Отключаем кеширование WSDL
ini_set("soap.wsdl_cache_enabled","0");
function test(MyMix $data) {
    return($data->varString);
}

$server->addFunction("test");
$server->handle();
?>

```

Рис. 3. Soap сервер

Актуальність веб-служб зростає з популярністю Progressive Web Apps (PWA). PWA можна сприймати як сайт, побудований з використанням веб-технологій, але який взаємодіє з користувачем як додаток.