## Учреждение образования

## Белорусский государственный технологический университет

Кафедра полиграфического оборудования и

системы обработки информации

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7**

по дисциплине «Стандартизация и сертификация

информационных систем и технологий»

Тема

«Ознакомление со стандартами обмена

сообщениями внутри сети *Internet*»

Выполнил студент:

Миневич Кристина

ф-та ИТ 4 к. 4 гр.

Проверил

кандидат технических наук

Сулим П.Е.

Отчет по лабораторной работе

защищен с отметкой баллов

Минск 2024

**Лабораторная работа № 7**

**ОЗНАКОМЛЕНИЕ СО СТАНДАРТАМИ ОБМЕНА**

**СООБЩЕНИЯМИ ВНУТРИ СЕТИ INTERNET**

***Цель* *работы*:**

Ознакомление со стандартами обмена сообщениями внутри сети *Internet.*

**Описание работы:**

1. **Стандарты доступа к сервисам.**

В основе Web-сервисов лежат Internet-стандарты. Эти стандарты определяют протоколы, а не способы их реализации. Такое утверждение является залогом успеха Internet — ни одна компания не может влиять на Internet-стандарты и задавать собственные правила игры. Например, стандарты Web-сервисов разрабатываются совместно такими компаниями, как IBM, Microsoft, Ariba и некоторыми другими, и обсуждаются комитетом World Wide Web Consortium (W3C).

Web-сервисы базируются на трех основных Web-стандартах:

* SOAP (Simple Object Access Protocol) — на протоколе для посылки сообщений по протоколу HTTP и другим Internet-протоколам;
* WSDL (Web Services Description Language) — на языке для описания программных интерфейсов Web-сервисов;
* UDDI (Universal Description, Discovery and Integration) — на стандарте для индексации Web-сервисов.

Источники:

<http://kunegin.com/ref6/web/4.htm>

1. **Стандарт *SOAP*, протокол *SOAP*.**

**SOAP** (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Simple Object Access Protocol* — простой протокол доступа к объектам) — [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) обмена структурированными сообщениями в распределённой вычислительной среде. Первоначально SOAP предназначался в основном для реализации удалённого вызова процедур ([RPC](https://ru.wikipedia.org/wiki/Remote_Procedure_Call)). Сейчас протокол используется для обмена произвольными сообщениями в формате [XML](https://ru.wikipedia.org/wiki/XML), а не только для вызова процедур. Официальная спецификация последней версии 1.2 протокола никак не расшифровывает название SOAP.

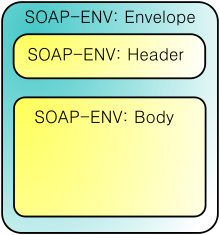
SOAP является расширением протокола [XML-RPC](https://ru.wikipedia.org/wiki/XML-RPC).

SOAP может использоваться с любым протоколом прикладного уровня: [SMTP](https://ru.wikipedia.org/wiki/SMTP), [FTP](https://ru.wikipedia.org/wiki/FTP), [HTTP](https://ru.wikipedia.org/wiki/HTTP), [HTTPS](https://ru.wikipedia.org/wiki/HTTPS) и др. Однако его взаимодействие с каждым из этих протоколов имеет свои особенности, которые должны быть определены отдельно. Чаще всего SOAP используется поверх HTTP.

SOAP является одним из стандартов, на которых базируются технологии [веб-служб](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0).

Сообщение SOAP выглядит так:

* **Envelope** — корневой элемент, который определяет сообщение и пространство имен, использованное в документе.
* **Header** — содержит атрибуты сообщения, например: информация о безопасности или о сетевой маршрутизации.
* **Body** — содержит сообщение, которым обмениваются приложения.
* **Fault** — необязательный элемент, который предоставляет информацию об ошибках, которые произошли при обработке сообщений.





Источники:

<https://ru.wikipedia.org/wiki/SOAP>

1. **Форматы *XML* и *JSON*.**

**XML** – расширяемый [язык разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8). Рекомендован [Консорциумом Всемирной паутины](https://ru.wikipedia.org/wiki/W3C) (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным [синтаксисом](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), удобный для создания и обработки документов как программами, так и человеком, с акцентом на использование в Интернете.

Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. *Расширение XML* – это конкретная [грамматика](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0), созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих, какие атрибуты и элементы могут входить в состав других элементов. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках [Юникод](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) для представления содержания документов привело к широкому использованию как, собственно, XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

XML является подмножеством [SGML](https://ru.wikipedia.org/wiki/SGML).

Спецификация XML описывает язык и ряд вопросов, касающихся кодировки и обработки документов. Материал этой секции представляет собой сокращённое изложение описания языка в Спецификации XML, адаптированное для настоящей статьи.

Нормативным считается английский вариант документа, поэтому основные термины приводятся с их английскими оригиналами.

Перевод основных терминов в основном следует доступному в интернете переводу Спецификации на русский язык, исключение составляют термины *tag* и *declaration*. Для термина tag здесь используется перевод *тег*. Для термина declaration отдано предпочтение распространённому переводу *объявление* (против также распространённой кальки *декларация*).

В литературе и интернете могут встречаться и иные переводы основных терминов.

С физической точки зрения документ состоит из [*сущностей*](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D1%89%D0%BD%D0%BE%D1%81%D1%82%D1%8C_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), из которых каждая может ссылаться на другую сущность. Единственный ***корневой элемент*** — **документная сущность**. Содержание сущностей — символы.

С логической точки зрения документ состоит из *комментариев*, *объявлений*, *элементов*, *ссылок* на сущности и *инструкций обработки*. Всё это в документе структурируется *разметкой*.

**Сущность** — мельчайшая часть в документе. Все сущности что-нибудь содержат, и у всех них есть имя (существуют исключения, напр. **документная сущность**). Проще говоря, термин «сущность» описывает «сущую вещь», «**что-то**».

Документ состоит из сущностей, содержание которых — символы. Все символы разделены на два типа: символы данных и символы разметки. К разметке относятся:

1. Теги <- обозначают границы элементов
2. объявления и инструкции обработки, включая их *атрибуты* ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *attributes*)
3. ссылки на сущности
4. комментарии
5. а также последовательности символов, обрамляющие секции «[CDATA](https://ru.wikipedia.org/wiki/CDATA)»

Часть документа, не принадлежащая разметке, составляет символьные данные документа.

Все составляющие части документа обобщаются в *пролог* и **корневой элемент**. **Корневой элемент** — обязательная часть документа, составляющая всю его суть (пролог, вообще говоря, может отсутствовать). Корневой элемент может включать (а может не включать) вложенные в него элементы, символьные данные и комментарии. Вложенные в корневой элемент элементы, в свою очередь, могут включать вложенные в них элементы, символьные данные и комментарии, и так далее. **Пролог** может включать **объявления**, **инструкции обработки**, **комментарии**. Его следует начинать с *объявления XML*, хотя в определённой ситуации допускается отсутствие этого объявления.

Элементы документа должны быть *правильно вложены*: любой элемент, начинающийся внутри другого элемента (то есть любой элемент документа, кроме корневого), должен заканчиваться внутри элемента, в котором он начался. Символьные данные могут встречаться внутри элементов как непосредственно так и в специальных *секциях «CDATA»*. Объявления, инструкции обработки и элементы могут иметь связанные с ними атрибуты. Атрибуты используются для связывания с логической единицей текста пар имя-значение.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <!DOCTYPE recipe>  **<recipe** name="хлеб" preptime="5min" cooktime="180min"**>**  **<title>**  Простой хлеб  **</title>**  **<composition>**  **<ingredient** amount="3" unit="стакан"**>**Мука**</ingredient>**  **<ingredient** amount="0.25" unit="грамм"**>**Дрожжи**</ingredient>**  **<ingredient** amount="1.5" unit="стакан"**>**Тёплая вода**</ingredient>**  **</composition>**  **<instructions>**  **<step>**  Смешать все ингредиенты и тщательно замесить.  **</step>**  **<step>**  Закрыть тканью и оставить на один час в тёплом помещении.  **</step>**  *<!--*  *<step>*  *Почитать вчерашнюю газету.*  *</step>*  *- это сомнительный шаг...*  *-->*  **<step>**  Замесить ещё раз, положить на противень и поставить в духовку.  **</step>**  **</instructions>**  **</recipe>** |

**JSON** — [текстовый формат](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82) [обмена данными](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D0%B5%D0%BD_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8), основанный на [JavaScript](https://ru.wikipedia.org/wiki/JavaScript). Как и многие другие текстовые форматы, JSON легко читается людьми. Формат JSON был разработан [Дугласом Крокфордом](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D0%BA%D1%84%D0%BE%D1%80%D0%B4,_%D0%94%D1%83%D0%B3%D0%BB%D0%B0%D1%81).

Несмотря на происхождение от JavaScript (точнее, от подмножества языка стандарта [ECMA-262](https://ru.wikipedia.org/wiki/ECMA-262) [1999 года](https://ru.wikipedia.org/wiki/1999_%D0%B3%D0%BE%D0%B4)), формат считается независимым от языка и может использоваться практически с любым [языком программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Для многих языков существует готовый код для создания и обработки данных в формате JSON.

За счёт своей лаконичности по сравнению с [XML](https://ru.wikipedia.org/wiki/XML) формат JSON может быть более подходящим для [сериализации](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) сложных структур. Применяется в веб-приложениях как для обмена данными между [браузером](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) и сервером ([AJAX](https://ru.wikipedia.org/wiki/AJAX)), так и между серверами (программные [HTTP](https://ru.wikipedia.org/wiki/HTTP)-сопряжения).

Поскольку формат JSON является подмножеством синтаксиса языка JavaScript, то он может быть быстро десериализован встроенной функцией eval().

JSON-текст представляет собой (в закодированном виде) одну из двух структур:

* Набор пар *ключ: значение*. В различных языках это реализовано как [запись](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%B8%D1%81%D1%8C_(%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)), [структура](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), [словарь](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2), [хеш-таблица](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88-%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0), [список](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) с ключом или [ассоциативный массив](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2). Ключом может быть только строка ([регистрозависимость](https://ru.wikipedia.org/wiki/%D0%A7%D1%83%D0%B2%D1%81%D1%82%D0%B2%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%BA_%D1%80%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80%D1%83_%D1%81%D0%B8%D0%BC%D0%B2%D0%BE%D0%BB%D0%BE%D0%B2) не регулируется стандартом, это остаётся на усмотрение программного обеспечения. Как правило, регистр учитывается программами — имена с буквами в разных регистрах считаются разными, например), значением — любая форма. Повторяющиеся имена ключей допустимы, но не рекомендуются стандартом; обработка таких ситуаций происходит на усмотрение программного обеспечения, возможные варианты — учитывать только первый такой ключ, учитывать только последний такой ключ, генерировать ошибку.
* Упорядоченный набор *значений*. Во многих языках это реализовано как [массив](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%81%D0%B8%D0%B2_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [вектор](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D0%B5%D0%BA%D1%81%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2), список или [последовательность](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C).

Структуры данных, используемые JSON, поддерживаются любым современным языком программирования, что и позволяет применять JSON для обмена данными между различными языками программирования и программными системами.

В качестве значений в JSON могут быть использованы:

* **запись** — это неупорядоченное множество пар **ключ**:**значение**, заключённое в фигурные скобки *«{ }»*. Ключ описывается **строкой**, между ним и значением стоит символ *«:»*. Пары *ключ-значение* отделяются друг от друга запятыми.
* **массив** (одномерный) — это упорядоченное множество **значений**. Массив заключается в квадратные скобки *«[ ]»*. Значения разделяются запятыми. Массив может быть пустым, то есть не содержать ни одного значения. Значения в пределах одного массива могут иметь разный тип.
* **число** (целое или вещественное).
* **литералы** *true* ([логическое значение](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF) «истина»), *false* ([логическое значение](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF) «ложь») и *null*.
* **строка** — это упорядоченное множество из нуля или более символов [юникода](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4), заключённое в двойные кавычки. Символы могут быть указаны с использованием [escape-последовательностей](https://ru.wikipedia.org/wiki/Escape-%D0%BF%D0%BE%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C), начинающихся с [обратной косой черты](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BE%D1%81%D0%B0%D1%8F_%D1%87%D0%B5%D1%80%D1%82%D0%B0) *«\»* (поддерживаются варианты \", \\, \/, \t, \n, \r, \f и \b), или записаны шестнадцатеричным кодом в кодировке [Unicode](https://ru.wikipedia.org/wiki/Unicode) в виде \uFFFF.

*Строка* очень похожа на [литерал](https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%82%D0%B5%D1%80%D0%B0%D0%BB_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) одноимённого типа данных в языке [Javascript](https://ru.wikipedia.org/wiki/Javascript). *Число* тоже очень похоже на Javascript-число, за исключением того, что используется только [десятичный формат](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%81%D1%8F%D1%82%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D0%B4%D1%80%D0%BE%D0%B1%D1%8C) (с точкой в качестве разделителя). Пробелы могут быть вставлены между любыми двумя синтаксическими элементами.

Следующий пример показывает JSON-представление данных об объекте, описывающем человека. В данных присутствуют *строковые* поля имени и фамилии, информация об адресе и массив, содержащий список телефонов. Как видно из примера, *значение* может представлять собой вложенную структуру.

|  |
| --- |
| {  **"firstName"**: "Иван",  **"lastName"**: "Иванов",  **"address"**: {  **"streetAddress"**: "Московское ш., 101, кв.101",  **"city"**: "Ленинград",  **"postalCode"**: 101101  },  **"phoneNumbers"**: [  "812 123-1234",  "916 123-4567"  ]  } |

Источники:

<https://www.w3.org/standards/xml/core#:~:text=What%20is%20XML%3F,more%20suitable%20for%20Web%20use>.

<https://ru.wikipedia.org/wiki/XML>

<https://ru.wikipedia.org/wiki/JSON>

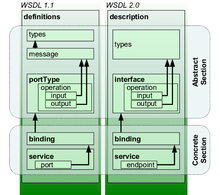
1. **Спецификация *WDDX*. Язык описания сервисов *WSDL*, *REST* и *RPC.***

**WDDX** (*Web Distributed Data eXchange*) — механизм обмена данными в [гетерогенной среде](https://ru.wikipedia.org/w/index.php?title=%D0%93%D0%B5%D1%82%D0%B5%D1%80%D0%BE%D0%B3%D0%B5%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0&action=edit&redlink=1), независящий от [языка программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), [платформы](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0) и способа транспортировки. Представляет собой [спецификацию](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B5%D1%86%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) определённого формата [XML](https://ru.wikipedia.org/wiki/XML) 1.0 [DTD](https://ru.wikipedia.org/wiki/DTD) и имеет множество интерфейсов (библиотек) в большинстве современных языков программирования.

WDDX поддерживает примитивные [типы данных](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), такие как целые и вещественные числа, строки и булевы значения, а также более сложные — структуры, множества и другие.

Формат был разработан в [1998 году](https://ru.wikipedia.org/wiki/1998_%D0%B3%D0%BE%D0%B4) Симеоном Симеоновым в фирме [Allaire Corporation](https://ru.wikipedia.org/wiki/Allaire_Corporation) для использования с [ColdFusion](https://ru.wikipedia.org/wiki/ColdFusion).

**WSDL (*Web Services Description Language*)**— язык 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) и доступа к ним, основанный на языке [XML](https://ru.wikipedia.org/wiki/XML).

[](https://commons.wikimedia.org/wiki/File:WSDL_11vs20.png?uselang=ru)

Последняя официальная спецификация на момент написания статьи версия 2.0 ([WSDL Version 2.0](http://www.w3.org/TR/wsdl20) от [26 июня](https://ru.wikipedia.org/wiki/26_%D0%B8%D1%8E%D0%BD%D1%8F) [2007 года](https://ru.wikipedia.org/wiki/2007_%D0%B3%D0%BE%D0%B4)), которая имеет статус рекомендации, и версия 1.1 ([WSDL Version 1.1](https://www.w3.org/TR/2001/NOTE-wsdl-20010315) от [15 марта](https://ru.wikipedia.org/wiki/15_%D0%BC%D0%B0%D1%80%D1%82%D0%B0) [2001 года](https://ru.wikipedia.org/wiki/2001_%D0%B3%D0%BE%D0%B4)), которая имеет статус заметки (note).

Каждый документ WSDL 1.1 можно разбить на следующие логические части:

1. определение типов данных (types) — определение вида отправляемых и получаемых сервисом XML-сообщений
2. элементы данных (message) — сообщения, используемые web-сервисом
3. абстрактные операции (portType) — список операций, которые могут быть выполнены с сообщениями
4. связывание сервисов (binding) — способ, которым сообщение будет доставлено

|  |
| --- |
| **<message** name="getTermRequest"**>**  **<part** name="term" type="xs:string"**/>**  **</message>**  **<message** name="getTermResponse"**>**  **<part** name="value" type="xs:string"**/>**  **</message>**  **<portType** name="glossaryTerms"**>**  **<operation** name="getTerm"**>**  **<input** message="getTermRequest"**/>**  **<output** message="getTermResponse"**/>**  **</operation>**  **</portType>** |

**REST (Representational state transfer)** – это стиль архитектуры программного обеспечения для распределенных систем, таких как World Wide Web, который, как правило, используется для построения веб-служб.

**RPC** - класс технологий, позволяющих компьютерным программам вызывать функции или процедуры в другом адресном пространстве (на удалённых компьютерах, либо в независимой сторонней системе на том же устройстве).

UDDI — Universal Description, Discovery and Integration

Задача UDDI — предоставить механизм для обнаружения веб-сервисов (индексации). UDDI задает бизнес-реестр, в котором провайдеры веб-сервисов могут регистрировать сервисы, а разработчики — искать необходимые им сервисы.

Источники:

<https://ru.wikipedia.org/wiki/WDDX>

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

<https://ru.wikipedia.org/wiki/REST>

<https://ru.wikipedia.org/wiki/UDDI>

**Вывод:** В данной работе приведены примеры реализации форматов сообщений JSON, XML, SOAP. Рассмотрены основные Web-стандарты на которых базируются Web-сервисы. Пояснено что представляет из себя WDDX. Изучены архитектурные подходы к проектированию веб приложений REST, RPC с моделью взаимодействия обеих архитектур.