

## Формат обмена XML и JSON

Люди для обмена информацией используют массу способов: голосом, текстом, а еще есть обширная область невербального общения, начать изучать которую можно на примере собственной супруги или кота.

Сама суть общения разная. Мы можем посыпать информацию окружающим, и нас не особо интересует, поняли тебя или нет. Те же собаки, когда рычат и скалят зубы. Или ваша благоверная, когда у нее руки в боки: ей совершенно без разницы, что сказал шеф на вашей работе. Другой тип передачи информации более кровавый. Это когда передающий пытается удостовериться, что принимающий понял. Начиная с сообщения во дворе со встроенным запросом валидации «че не понял, вали отсюда!», заканчивая чудной контрольной по русскому языку.

Если у Вас есть подчиненные на работе, ну или просто свои дети, то, скорее всего, Вы думали «как сделать так, чтобы он понял с первого раза». А что, если бы понималка работала сама, и достаточно было бы отправить информацию в нужном «формате», что получатель понял ее точно так, как задумывали Вы?

К счастью, в компьютерном мире пока еще нет приближения к понятию «понимание». Всё, что может компьютер, это передать без искажений цифровой сигнал и сложить единички друг с другом.

Единички, конечно, круто, но как быть с прокладкой между столом и монитором? Нам же хотелось бы не только общаться с компьютером понятным нам языком, но и писать свои программы так, чтобы в том месте, где мы собираемся что-то передать другому компьютеру или в другую часть программы, это всё было понятно человеку. Разумеется, как только появились компьютеры, сразу были придуманы файлы. И у файлов появилось сто триллионов форматов. Каждый автор программы придумывал его сам.

И сразу встал вопрос совместимости форматов. Те, кто постарше, помнят, что файлы с Mac не читались на MS Win 3.11 и далее в MS Win95, созданные даже одной версией программы. Был расцвет «конвертеров». Ну а файлы с других платформ мы даже не будем рассматривать. Для меня было чудо, как мой ZX Spectrum смог прочитать содержимое диска 5.25, записанного на x86.

Когда программ стало много, веб-технологии шагнули вперед, умные люди устали париться с форматами и сказали так. Ребята и девчонки! Давайте замутим универсальный формат обмена. Пусть он даже будет не оптимален с точки зрения размера (*f\*\*ck the fuel economy*), зато будет стандартизирован, всем понятен и будет куда меньше гемора. Ведь известно, что стоимость работы машины меньше стоимости работы человека, поэтому затраты на разбор формата и на хранения это ничто по сравнению с постоянными трудозатратами разработчиков на «совместимость».

### И вот, родился XML.

По мотивам HTML, который был разработан около 1986 года учеными в Швейцарии для обмена структурированной информацией.

Сам XML появился в 1998 году.

Спецификацию, разумеется, курим в Интернете.

Ключевые моменты такие:

1. Язык расширяемый. Т.е., можно самому придумывать грамматику. Т.е., всякие штуки, из которых состоит язык.

2. Основа — узлы (элементы), в которых можно делать другие узлы. Например, давайте опишем двух друзей:

```
<man>
  <name = «Вася» />
  <gender = «х3» />
  <isalco = «Да» />
</man>
```

```
<man>
  <name = «Петя» />
  <gender = «М» />
  <isalco = «Нет» />
</man>
```

Наполнив некий файл таким содержимым про всех нужных людей, мы точно и легко передадим все сведения туда, куда хотим. Такой формат понятен и человеку. Кто сейчас хмыкал, тот пишет элемент `<isalco></isalco>` про себя, только честно.

У всех есть Госуслуги. Зайдите туда, найдите свое последнее заявление, и качните его в XML формате. Зацените структуру.

**Вот пример красивого XML для обмена данными о заказах между магазином и поставщиком (строка заказа):**

```
<lineItems>
  <currencyISOCode>RUB</currencyISOCode>
  <lineItem>
    <description>Баритовый Концентрат ст.
    Приобье</description>
    <internalBuyerCode>0039</internalBuyerCode>
    <onePlaceQuantity>4</onePlaceQuantity>
    <requestedQuantity
      unitOfMeasure="PCE">4</requestedQuantity>
      <vATAmount>0</vATAmount>
    </lineItem>

    <totalAmount>0</totalAmount>
    <totalSumExcludingTaxes>0</totalSumExcludingTaxes>
    <totalVATAmount>0</totalVATAmount>
  </lineItems>
```

Мне кажется, тут и описание не нужно, всё очевидно.

**А вот творчество нашей ФНС. Это формат документа УПД:**

```
<Документ ВремИнфПр="07.14.13" ДатаИнфПр="26.05.2021" КНД="1115131"
НаимДокОпр="Счет-фактура и документ об отгрузке товаров (выполнении работ),
передаче имущественных прав (документ об оказании услуг)"
НаимЭконСубСост="ООО "&quot;&quot;"
```

ПоФактХЖ="Документ об отгрузке товаров (выполнении работ), передаче имущественных прав (документ об оказании услуг)" Функция="СЧФДОП">  
<СвСчФакт ДатаСчФ="27.04.2021" КодОКВ="643" НомерСчФ="/">  
  <СвПрод ОКПО="81195149">  
    <ИдСв>  
      <СвЮЛУч ИННЮЛ="1903\*\*\*\*" КПП="19\*\*\*\*" НаимОрг="ООО  
      "Солнышко"/>  
    </ИдСв>  
    <Адрес>  
      <АдрИнф АдрТекст="г Черногрязкс, ул Петра Толстого д. 6А оф. " КодСтр="643"/>  
    </Адрес>  
  </СвПрод>  
  <ГрузОтр>  
    <ГрузОтпр>  
      <ИдСв>  
        <СвЮЛУч ИННЮЛ="770\*\*" КПП="720\*\*" НаимОрг="ООО  
      "ТЕХНОЛОГИЧЕСКАЯ КОМПАНИЯ ИП Карапетян"/>  
      </ИдСв>  
      <Адрес>  
        <АдрИнф АдрТекст="г Тюмень, ул 150 лет Октября д. 140 оф. 5" КодСтр="643"/>  
      </Адрес>  
    </ГрузОтпр>

Почувствуйте разницу. Как загнать канцелярит в XML. Да легко!

## JSON

XML отлично подходит для выгрузки данных в файл, передаче их через всякие каналы. Но для быстрого обмена часто хочется чего-то более интересного. В современных языках есть такая штука «**Словарь**». Устроено довольно просто. Это набор пар «ключ»: «значение». Типа можно «выкрикнуть» ключ, а программа ответит, что там в этом ключе. Иванов «Я», Петров «Я», Тртысячи тридцатый «Зозо моя фамилия».

Давайте перепишем формат сведений о людях так:

```
{  
  «name»: «Дмитрий»;  
  «gender»: «Ж»;  
  «isalco»: «Да»;  
}
```

Что, если обменяться таким лаконичным запросиком с какой-то штукой, которая нам сможет ответить в таком же формате? Это куда быстрее, чем формировать XML и пр.

Такой формат называется JSON, применяется повсеместно для связи веб-приложений, интеграции с другими системами и прочее.

Например, некоторое интеграционное решение имеет веб-контроллер для обработки запросов. И запросы к нему могут выглядеть вот так:

```
curl -X POST 'http://localhost:8069/partner/create' -H 'Content-Type:application/json' -d '{"token":"123456","client_guid_1s":"1234","name":"ИП Карапетян","inn":"7715","kpp":"1","email":"123@mail.ru"}'
```

## Задания

1. Прочитать про XML.
2. Прочитать про JSON.
3. Разобраться, что такое CURL.
4. Раздобыть пример УПД в формате XML, изучить.
5. Загрузить со своих Госуслуг любое заявление в формате XML и изучить.