

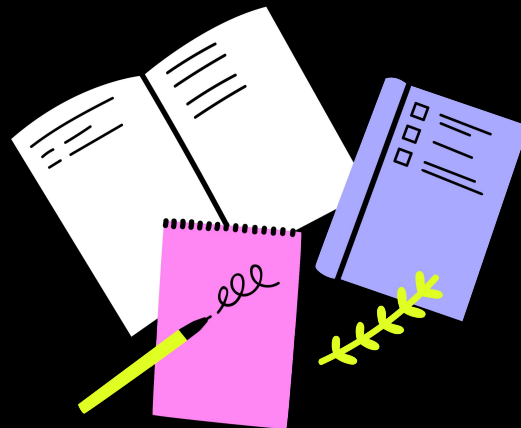
# JSON и XML — основные форматы передачи данных

Курс по знакомству  
с веб-технологиями

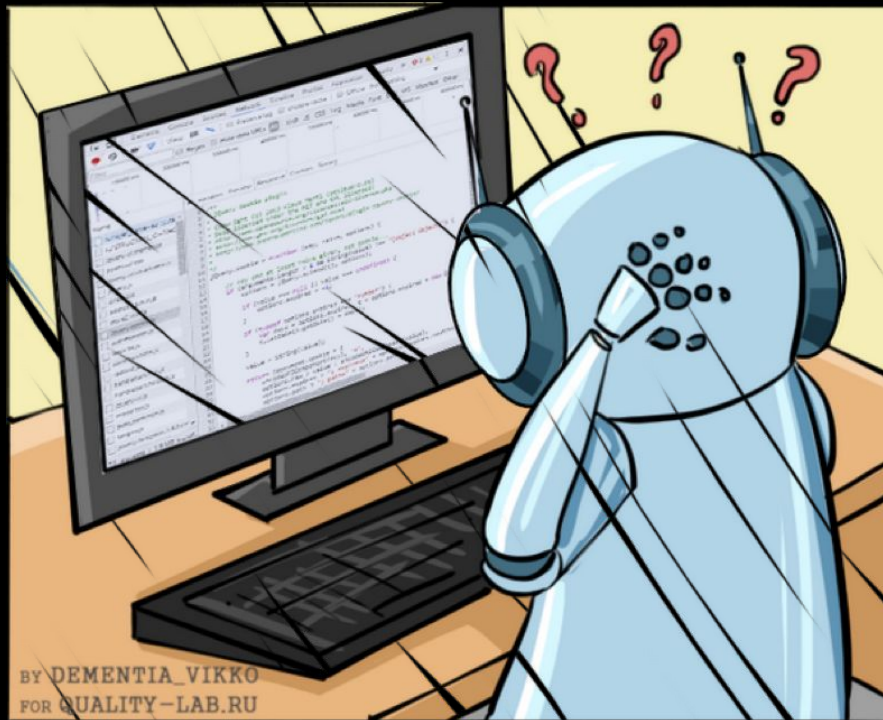


## План лекции:

- XML
- JSON



JSON и XML – основные форматы передачи данных





JSON и XML – основные форматы передачи данных

# Переписка с сервером





## **SOAP и REST — это два стиля API**

SOAP появился в 1998 году стараниями Microsoft. И продолжал развиваться, пока в 2003 году W3C не утвердила в качестве рекомендации SOAP 1.2, который и сейчас является последним.

В 2018 году большинство общедоступных веб-служб предоставили API-интерфейсы REST и передавали данные в компактном и простом в использовании формате обмена данными JSON.

Однако корпоративные пользователи все еще часто выбирают SOAP для своих веб-сервисов.





JSON и XML – основные форматы передачи данных

# Основные различия между SOAP и REST

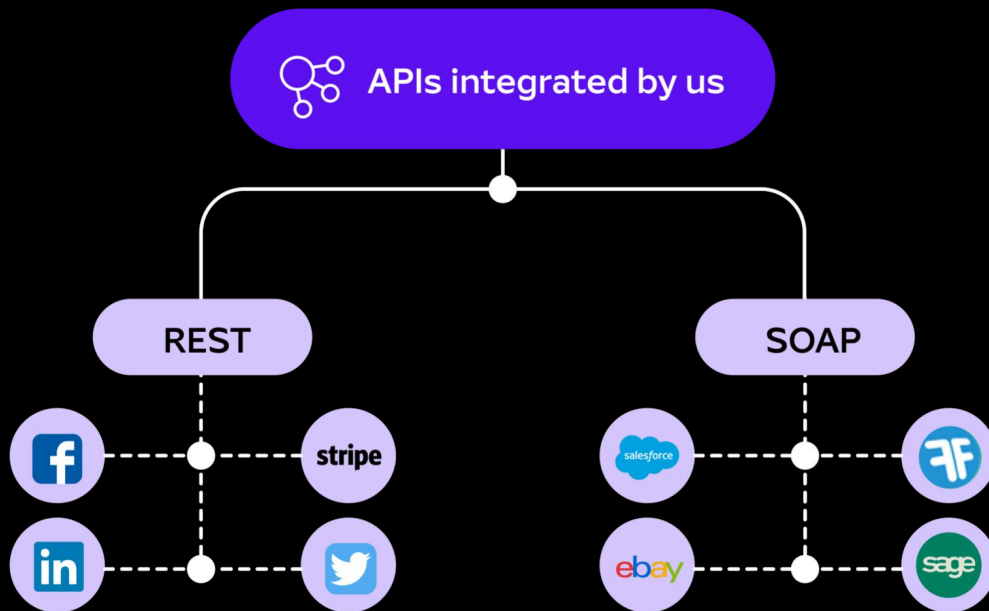




***WOULD YOU  
RECOMMEND  
REST OR SOAP,  
AMIGO?***



JSON и XML – основные форматы передачи данных







JSON и XML – основные форматы передачи данных

**SOAP** (Simple Object Access Protocol) —  
это стандартизированный протокол, который отправляет  
сообщения с использованием других протоколов, таких  
как HTTP и др.





**REST** (Representational State Transfer) —  
это не протокол, а архитектурный стиль.

Архитектура REST устанавливает набор рекомендаций,  
которым необходимо следовать, если вы хотите  
предоставить веб-службе RESTful, например,  
существование без сохранения состояния  
и использование кодов состояния HTTP.





JSON и XML – основные форматы передачи данных

Для SOAP и REST-запросов используется два наиболее популярных формата передачи данных:

- **XML** — используется в SOAP (всегда) и REST-запросах (реже);
- **JSON** — используется в REST-запросах.





JSON и XML – основные форматы передачи данных

# XML





JSON и XML – основные форматы передачи данных

**XML** (eXtensible Markup Language) — расширяемый язык разметки.

Используется для хранения и передачи данных. Так что увидеть его можно не только в API, но и в коде.



## Как устроен XML



## ◆ JSON и XML – основные форматы передачи данных

```
<req>  
<query>мск сухонская 11</query>  
<count>7</count>  
</req>
```





# Как это работает

[инструкция по настройке](#)

Войти

Адрес

Адрес по полям

Организация

Банк

ФИО

ФИО по полям

Email

Прочие

Город, улица, дом, квартира

↙ Введите адрес

мск сухонская 11

Выберите вариант или продолжите ввод

г Москва, ул [Сухонская](#), д 11

г Москва, ул [Сухонская](#), д 11А

г Москва, ул [Сухонская](#), д 11, кв 1

г Москва, ул [Сухонская](#), д 11, кв 2

г Москва, ул [Сухонская](#), д 11, кв 3

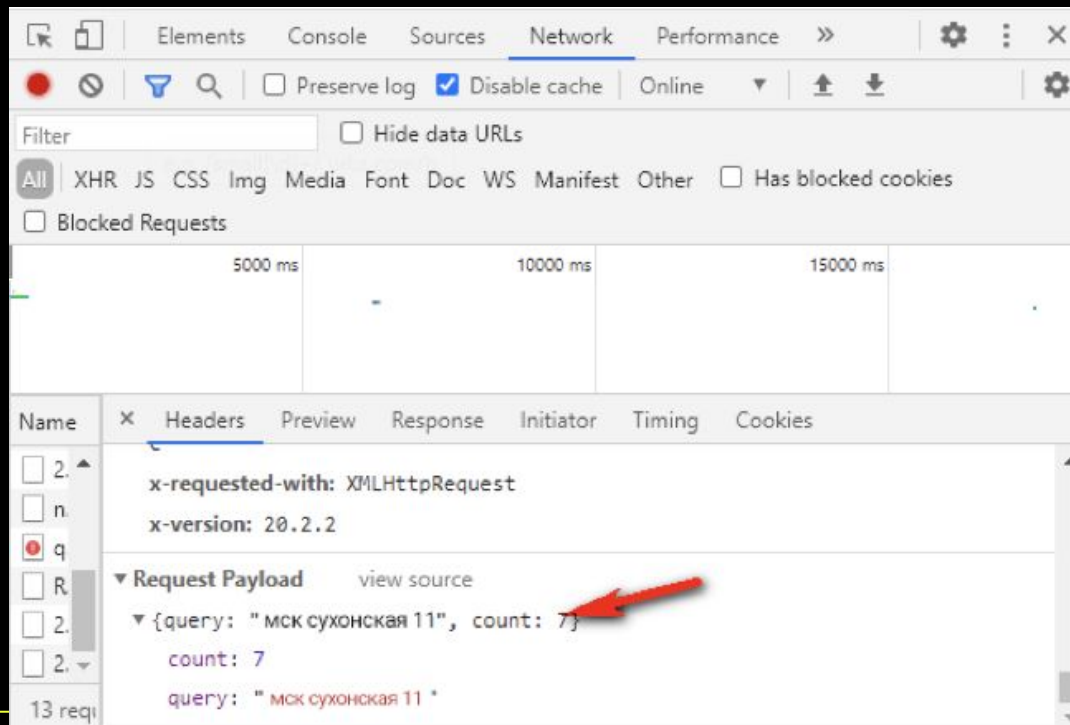
г Москва, ул [Сухонская](#), д 11, кв 4

г Москва, ул [Сухонская](#), д 11, кв 5





## JSON и XML – основные форматы передачи данных





```
<party type="PRIVATE" sourceSystem="AL" rawId="2">
  <field name="name">Ленинградский</field>
  <attribute type="NUMBER" rawId="AL.2.PH.1">
    <field name="type">NUMBER</field>
    <field name="house">43</field>
  </attribute>
</party>
```





JSON и XML – основные форматы передачи данных

# XSD-схема





JSON и XML – основные форматы передачи данных

**XSD** (XML Schema Definition) —  
это описание вашего XML.

Как он должен выглядеть, что в нем должно быть?  
Это ТЗ, написанное на языке машины.





Если мы создаем SOAP-метод,  
то указываем в схеме:

1. какие поля будут в запросе;
2. какие поля будут в ответе;
3. какие типы данных у каждого поля;
4. какие поля обязательны для заполнения, а какие нет;
5. есть ли у поля значение по умолчанию, и какое оно;
6. есть ли у поля ограничение по длине;
7. есть ли у поля другие параметры;
8. какая у запроса структура по вложенности элементов;
9. ...





## Пример

Метод doRegister в Users.

Должны передать email, name и password для запроса

```
<wrap:doRegister>  
  <email>olga@gmail.com</email>  
  <name>Ольга</name>  
  <password>1</password>  
</wrap:doRegister>
```





## Пример

### XSD-схема

```
<xs:element name="doRegister ">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="email" type="xs:string"/>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="password" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```





JSON и XML – основные форматы передачи данных

# Well Formed XML







## Правила well formed XML

1. Есть корневой элемент.
2. У каждого элемента есть закрывающийся тег.
3. Теги регистрозависимы!
4. Соблюдается правильная вложенность элементов.
5. Атрибуты оформлены в кавычках.





JSON и XML – основные форматы передачи данных

# JSON





**JSON** (англ. JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript. Но при этом формат независим от JS и может использоваться в любом языке программирования.



В SOAP API возможен только в формат XML, а вот REST API поддерживает как XML, так и JSON.

Разработчики предпочитают JSON — он легче читается человеком и меньше весит.





Объект заключен в фигурные скобки {}.

JSON-объект — это неупорядоченное множество пар «ключ:значение».

Ключ — это название параметра, который мы передаем серверу.

Он служит маркером для принимающей запрос системы: «смотри, здесь у меня значение такого-то параметра!».

```
{  
  "query": "мск сухонская 11",  
  "count": 7  
}
```





## Как это работает

[инструкция по настройке](#)

Войти

Адрес

Адрес по полям

Организация

Банк

ФИО

ФИО по полям

Email

Прочие

Город, улица, дом, квартира

↙ Введите адрес

мск сухонская 11

Выберите вариант или продолжите ввод

г Москва, ул Сухонская, д 11

г Москва, ул Сухонская, д 11А

г Москва, ул Сухонская, д 11, кв 1

г Москва, ул Сухонская, д 11, кв 2

г Москва, ул Сухонская, д 11, кв 3

г Москва, ул Сухонская, д 11, кв 4

г Москва, ул Сухонская, д 11, кв 5





JSON и XML – основные форматы передачи данных

# Well Formed JSON





## Правила well formed JSON

1. Данные написаны в виде пар «ключ:значение»
2. Данные разделены запятыми
3. Объект находится внутри фигурных скобок {}
4. Массив — внутри квадратных []





JSON и XML – основные форматы передачи данных

# Итоги





JSON и XML – основные форматы передачи данных





## XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<person>
  <name>Иван</name>
  <age>37</age>
  <mother>
    <name>Ольга</name>
    <age>58</age>
  </mother>
  <children>
    <child>Маша</child>
    <child>Игорь</child>
    <child>Таня</child>
  </children>
  <married>true</married>
  <dog null="true" />
</person>
```




## JSON

```
{
  "person":{
    "name":"Иван",
    "age":37,
    "mother":{
      "name":"Ольга",
      "age":58
    },
    "children":[
      "Маша",
      "Игорь",
      "Таня"
    ],
    "married":true,
    "dog":null
  }
}
```

JSON – это только формат данных, тогда как XML – это язык разметки.



**Спасибо  
за внимание**

A yellow smiley face is drawn over the text. It has two vertical lines for eyes and a wide, curved line for a mouth.