**Вопросы 💎**

1. **Напишите JSON к вашим ответам на вопрос 2 из прошлой недели. Например, если вы ответили *Книга (название, автор, год выпуска),* то JSON может выглядеть так (минимум 3 примера):**

[{

"title": "Властелин Колец",

"author": "Дж. Р. Р. Толкин",

"year": 1954

},{

"title": "Алиса в Зазеркалье",

"author": "Льюис Кэрролл",

"year": 1871

},{

"title": "Совершенный код",

"author": "Стив Макконнелл",

"year": 1993

}]

**Мои примеры:**

// пример json Мои домашние животные

[{

    "animal": "Котик",

    "name": "Гасик",

    "breed": "шотландский",

    "years": 9

}, {

    "animal": "Кошка",

    "name": "Нелли",

    "breed": "абиссинская",

    "years": 5

}, {

    "animal": "Собака",

    "name": "Шелби",

    "breed": "Американский питбуль",

    "years": 4

}]

/// пример более сложного объекта json

[{

    "firstName": "Michail",

    "lastName": "Kalinin",

    "age": 27,

    "address": {

        "streetAddress": "Plot-6, Mohan Nagar",

        "city": "Ghaziabad",

        "state": "UP",

        "postalCode": "201007"

    }

} {

    "firstName": "Anna",

    "lastName": "Jaiswal",

    "age": 37,

    "address": {

        "streetAddress": "25, Pushkin",

        "city": "Erevan",

        "state": "Armavir",

        "postalCode": "1107"

    }

} {

    "firstName": "Evgeniy",

    "lastName": "Jarov",

    "age": 57,

    "address": {

        "streetAddress": "Solnechnaya, 34",

        "city": "Sebezh",

        "state": "Pskov",

        "postalCode": "217587"

    }

}]

// пример с булевым значением

[{

    "first": true,

    "second": false

}]

1. **Самостоятельно разберитесь, что за формат данных XML и чем он отличается от JSON? Приведите пример, как один и тот же объект *собачки с картинки ниже* будет выглядеть в JSON и в XML?**

**Для наглядности сравним:**

|  |  |
| --- | --- |
| JSON — это сокращение от JavaScript Object Notation. JSON использует читаемый текст для хранения и передачи данных, содержащих массивы и значения пар атрибутов. Текст JSON можно легко преобразовать в объект JavaScript внутри JSON, а затем отправить на сервер. Он основан на JavaScript и эффективно используется с множеством языков программирования | XML — это обширный язык разметки, созданный для переноса данных. Он определяет некоторый стандартный набор правил для кодирования файлов в читаемом формате. Цель разработки этого XML — сосредоточиться на простоте и удобстве использования в Интернете. Это сильно поддерживается через Unicode. Это один из наиболее широко используемых языков для представления произвольных структур данных. |
| Возможности:  -Легко писать и манипулировать.  - Поддерживает все фреймворки JavaScript, а также все браузеры.  - Потребляет меньше памяти.  - Открытый исходный код и бесплатное использование | Возможности:   * - Он более эффективно обрабатывает сложные структуры данных. * - XML описывает данные на языке разметки. * - Он имеет возможность управлять данными в древовидной структуре, имеющей только один корневой элемент. |
| «student» :   [{      «id»:«01»,      «firstName»:«Shazim»,      «lastName»:«Khan»,      «rollNum»:«Fa12»,      «mobileNum»:«03007652334»      },      {      «id»:«02»,      «firstName»:«Alizah»,      «lastName»:«Shah»,      «rollNum»:«Fa20»,      «mobileNum»:«03217652334»      }] | <?xml version=«1.0» encoding=«UTF-8»?>  <students>  <student>  <*firstName*>Shazim</*firstName*>  <*lastName*>Khan</*lastName*>  <rollnumber>Fa12</rollnumber>  </student>  <student>  <*firstName*>Alizah</*firstName*>  <*lastName*>Shah</*lastName*>  <rollnumber>Fa20</rollnumber>  </student>  <student>  <*firstName*>Shoib</*firstName*>  <*lastName*>Ahmad</*lastName*>  <rollnumber>Fa21</rollnumber>  </student>  <student>  <*firstName*>Maliha</*firstName*>  <*lastName*>Ali</*lastName*>  <rollnumber>Fa18</rollnumber>  </student>  </students> |

JSON имеет меньший размер файла и эффективно передает данные в Интернет по сравнению с XML. JSON обращается к данным через объекты JSON, тогда как XML требует анализа данных. JSON легко читается, поскольку имеет более организованную структуру кода. С другой стороны, XML трудно интерпретировать из-за его сложной структуры.

JSON и XML нельзя полностью сопоставить.

JSON хорош для передачи данных, поскольку он не требует обработки, а XML может быть сложным, поскольку он позволяет не только передавать данные, но также обрабатывать и форматировать файлы.

JSON наименее безопасен в отличие от XML. JSON не может поддерживать пространства имен, но способен полностью поддерживать инструментарий Ajax.

XML поддерживает пространства имен, но не может поддерживать инструменты Ajax.

XML хранит данные иначе, чем JSON. Поскольку JSON хранит данные как карту, XML, напротив, хранит данные как древовидную структуру. Кроме того, JSON использует массивы, но не имеет закрывающих тегов.

***Оба они имеют разные аспекты с точки зрения представления, структуры данных, а также безопасности. Проанализировав ключевые различия между JSON и XML, приходим к выводу, что JSON — это гораздо более быстрый и простой подход для структурирования данных и их передачи. С другой стороны, XML также играет жизненно важную роль в хранении данных. Скорее JSON больше для людей, а XML для машин).***

1. **Что такое сериализация и десериализация (парсинг)? В каких ситуациях они нужны?**

Приведу пример на кошках

Вы пишете виртуального котика. У вас есть объект класс cat

class Cat {

private int age;

private int weight;

/ other cat logic

}

Мы хотим, чтобы при следующем запуске приложения этот же котик продолжал жить своей жизнью, а не пересоздался заново. Для этого мы должны реализовать сериализацию\десериализацию кота - то есть сохранение\загрузку.

Как - так как вам удобно. Можно сделать разными методами с помощью разных языков программирования, можно сохранять в текстовый файл как JSON (JavaScript Object Notation), можно в базе сохранить.

Главное - мы сохраняем каким-то образом его состояние (в данном случае - поля), и потом, когда нам это будет нужно, их загружаем.

Так же сериализованного кота можно будет, к примеру, передать по сети на сервер.

То есть вы передаете его состояние, а сервер у себя создаст новый объект класса Cat и установит у него это состояние (age и weight).

1. **Можно ли обработать ответ от сервера одновременно и как текст, и как JSON**?
2. **В чем особенность асинхронных запросов?**

**Совершенно непонятная история…** «асинхронный запрос» — это когда синхронно (т. е. одновременно) с ним может выполняться другой. А вот если синхронность в выполнении отсутствует, и запросы выполняются последовательно друг за другом, то их называют «синхронными». Если бы все запросы были синхронными, мы бы до пенсии ждали, пока на сайте все откроется и заработает) поэтому есть асинхронные запросы. Не уверена, что верно поняла.

1. **В чем преимущество AJAX-запросов перед старым способом работы с сервером через <form action="имя скрипта на сервере">?**

**Основные преимущества использования AJAX:**

* **снижение трафика** (из-за уменьшения объёма передаваемых данных между клиентом и сервером);
* **уменьшение нагрузки на сервер** (не нужно генерировать всю страницу, а только ту часть, которую нужно обновить);
* **увеличение быстродействия и отзывчивости** (нет необходимости в полной перезагрузки страницы, достаточно обновить содержимое только отдельных блоков);
* **повышение интерактивности** (с помощью AJAX можно сразу отображать результаты и сделать ресурс более удобным для пользования).

Взаимодействие с сервером через асинхронные запросы осуществляется посредством XHR или метода fetch().

1. **Напишите, как будет выглядеть fetch для получения данных вашего пользователя на github? Адрес URL для запроса должен выглядеть так:**

'<https://api.github.com/users/сюда> подставьте свой логин с github'

1. Самостоятельно разберитесь, что такое SPA?