1. API - application programming interface(программный интерфейс приложения.)

JSON - JavaScript Object Notation

XML - Extensible Markup Language

AJAX - Asynchronous Javascript and XML

Back-end - Это фнутренняя часть сайта(приложения) скрытая от пользователя

Front-end - Это внешняя часть сайта(приложения) видная пользователю и с которой он взаимодействует

Все эти технологии используются при разработке сайтов

2. Для обмена данными между браузером и сервером (AJAX), так и между серверами (программные HTTP-сопряжения).

Применяется при приемущественно в REST API.

3. {"name": "Jon", "age": 14}

4. [{"name": "Jon", "age": 14}, {"name": "Max", "age": 13}]

5. Сохраняет текущее состояние объекта.

const userJson = JSON.stringify(userObject);

6. Для декодирования JSON объекта

const a = Json.parse(object);

7. XML:

+ Позволяет создавать пользовательские структуры данных.

- Плохо читаем и много весит

Json:

+ Легко читаем и мало весит

- Все значения в JSON считаются строками

8. JSON

2. Cookie

1. Сохранить состояние объекта, отслеживание действий пользователя

2. document.cookie = `${elementName}=${elementValue};`;

3.

1. Ключ

2. Значение

3. Определение URL пути на который будут отправлтся cookie

4. domain - браузер будт отправлять cookie только для запрсов, относящихся к этому домену или поддомену.

5. expires - срок хранения cookie

6. secure - cookie будут отправляться только к сайта поддерживающим HTTPS

7. samesite - используется для контроля как cookie будут отправляться с использование cross-site request

8. httpOnly - который предотвращает доступ скриптов на стороне клиента к данным.

4. const cookie = document.cookie;

5. Установит expires на прошедушую дату

document.cookie = "username=; expires=Thu, 01 Jan 1970 00:00:00"

6. устанавливаются доменом, отличным от того, который вы посещаете в данный момент

1. Атрибут SameSite: Браузеры теперь применяют атрибут SameSite для файлов cookie,

который контролирует их отправку при межсайтовых запросах.

2. Блокировка файлов cookie третьими лицами

3. local storage, session storage

1.

Local storage - хранилище в браузере которое сохраяет данные до их очистки

Session storage - сохраняет данные только на время сессии.

2.

Может хранить больше данных.

Не отправляются при каждом запросе на сервер.

3. document.cookie = `userJson=${userJson};`;

4. const data = localStorage.getItem(key);

5. localStorage.removeItem(key);

localStorage.clear();

6. var key = localStorage[0];

7. var length = localStorage.length