JAVASCRIPT: ИНСТРУМЕНТЫ РАБОТЫ С ДАННЫМИ

KEYWORDS: JAVASCRIPT, EVENT BUBBLING, COOKIES, WEBSTORAGE, WEBSOCKETS, PLUGINS

СОДЕРЖАНИЕ

- **Работа с данными**
  - Передача данных
    - Ajax
    - WebSocket
  - Пребразование данных
    - JSON
    - XML
  - Хранение данных
    - Cookies
    - Web Storage
    - Web SQL Storage

ПЕРЕДАЧА ДАННЫХ

ПЕРЕДАЧА ДАННЫХ В СТРАНИЦУ

- **Статичная** – данные извлекаются на сервере и используются при генерации HTML-документа на сервере
- **Асинхронная** по запросу (ajax) – JavaScript-код генерирует запрос к серверу и интерпретирует полученные данные
- **Дуплексная** передача (WebSocket) – инициатором передачи может быть как страница, так и сервер

AJAX = ASYNCHRONOUS JAVASCRIPT AND XML

В основе технологии лежит класс XMLHttpRequest.

```
var request = new XMLHttpRequest();
```

Обращение к серверу состоит из 3* частей:

1) Инициализация callback-функции

```
request.onreadystatechange = function () {
    if (request.readyState == 4)
            if (request.status == 200)
                    console.log(request.responseText);
}
```

2) Инициация запроса

```
request.open('GET', url, true);          // "POST"
request.setRequestHeader("Header-Name", "Header val");
```

3) Отправка запроса

```
request.send(requestBody);               // .send()
```

4) Ожидание ответа от сервера

WEBSOCKETS

- **Дуплексный подход**
- **Нешифрованный и шифрованный трафик (ws://, wss://)**
- **WebSocket class**

```
var websocket = new WebSocket(“ws://server:90/”);
```

- **Обработчик onmessage**

```
websocket.onmessage = function(evt) {
        console.log("message rcv: " + evt.data);
}
```

SERVER SENT EVENTS

```
var source = new EventSource("demo_sse.php");


source.onmessage =
       function(event) {
    document
       .getElementById("result")
       .innerHTML += event.data + "<br/>";
};
```

ПРЕОБРАЗОВАНИЕ ДАННЫХ

ОБРАБОТКА JSON

Как из текста получить JavaScript Object (Array)?

1) `[Obsolete] var obj = eval('(' + text + ')');`
   `var obj = eval('({"a" : 123 })');`
2) `JSON.parse(text);`

   `JSON.parse(text, function(key, val) { });`

А наоборот?

1) `JSON.stringify(obj);`
2) `JSON.stringify(obj, ['name', 'value'])`
   `(obj, function(key,val) { ... })`
3) Внутри stringify() для каждогого объекта вызывается метод **toJSON()** (если есть)

ОБРАБОТКА XML

1) **Как из текста получить DOM-дерево?**
   1) `xmlHttpRequest.responseXML`
   2) `new DOMParser().parseFromString(text, “text/xml”)`

2) **Как сформировать XML из объекта?**
   1) `new XMLSerializer().`
      `serializeToString(document.body)`

ХРАНЕНИЕ ДАННЫХ

COOKIES

1) **Set Cookie**

```
document.cookie = “key1=val1;key2=val2”
document.cookie =
“key1=val1;key2=val2;expires=Thu, 18 Dec 2013 12:00:00 UTC”
// new Date().toUTCString()

document.cookie =
“key1=val1;key2=val2;path=/”
```

2) Delete Cookie

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

2) Read Cookie

```
       var coo = document.cookie; //coo == “key1=val1;key2=val2”
```

WEBSTORAGE

Web Storage

Очень просто:

```
localStorage.newItem = “new value”;
localStorage.setItem(“newItem”, “new value”);
sessionStorage
```

Минусы:

- **Работает в рамках строгой песочницы (protocol, server, port)**
- **медленное (сериализация-десериализация)**
- **в основных браузерах поддержка только строк**

INDEXEDDB, FILESYSTEM

Web SQL Database – obsolete (пример в reader)

IndexedDB – объектное хранилище

FileSystem – файловая система с ограниченной квотой