

Socket



ОПРЕДЕЛЕНИЕ

COMET – общий термин, описывающий различные техники получения данных по инициативе сервера.

AJAX – это «отправил запрос – получил результат», а **COMET** – это «непрерывный канал, по которому приходят данные».

Примеры COMET-приложений:

Чат – человек сидит и смотрит, что пишут другие. При этом новые сообщения приходят «сами по себе», он не должен нажимать на кнопку для обновления окна чата.

Аукцион – человек смотрит на экран и видит, как обновляется текущая ставка за товар.

Интерфейс редактирования – когда один редактор начинает изменять документ, другие видят информацию об этом.

Возможно и совместное редактирование, когда редакторы видят изменения друг друга.

ЧАСТЫЕ ОПРОСЫ

Первое решение для непрерывного получения событий с сервера – это «**частые опросы**» (**polling**), т.е. периодические запросы на сервер, например, раз в 10 секунд.

В ответ сервер во-первых помечает у себя, что клиент онлайн, а во-вторых посылает сообщение, в котором в специальном формате содержится весь пакет событий, накопившихся к данному моменту.

Недостатки

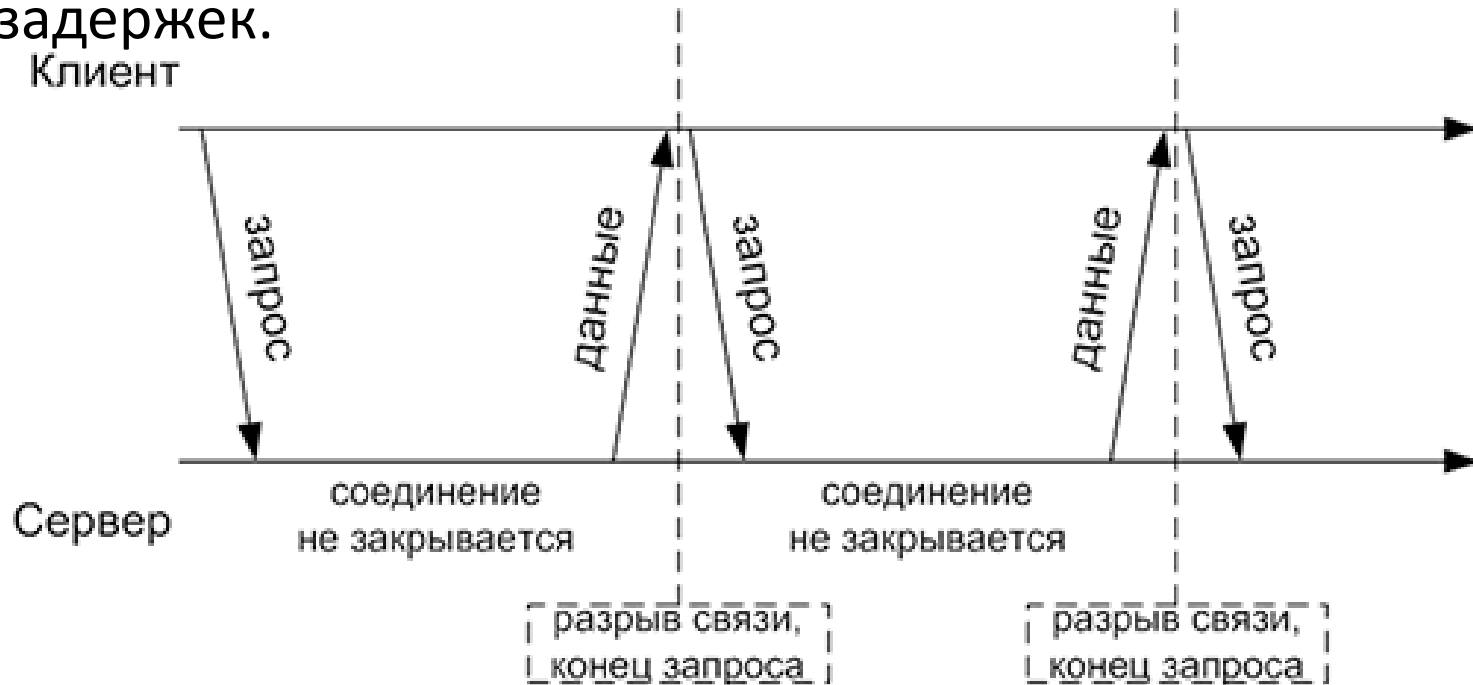
- Задержки между событием и уведомлением.
- Лишний трафик и запросы на сервер.

Достоинства

- Простота реализации на клиенте.

ДЛИННЫЕ ОПРОСЫ

Длинные опросы – отличная альтернатива частым опросам. Они также удобны в реализации, и при этом сообщения доставляются без задержек.



Длинные опросы отлично работают в тех случаях, когда сообщения приходят редко. При большом количестве частых сообщений график приёма-отправки, приведённый выше, превращается в «пилу».

WEBSOCKET

Протокол **WebSocket** (стандарт RFC 6455) предназначен для решения любых задач и снятия ограничений обмена данными между браузером и сервером.

Протокол WebSocket работает над TCP.

Это означает, что при соединении браузер отправляет по HTTP специальные заголовки, спрашивая: «поддерживает ли сервер WebSocket?».

Если сервер в ответных заголовках отвечает «да, поддерживаю», то дальше HTTP прекращается и общение идёт на специальном протоколе WebSocket, который уже не имеет с HTTP ничего общего.

WEBSOCKET (НА КЛИЕНТЕ)

```
let socket = new WebSocket("ws:localhost:8080");

socket.onopen = function() {
    console.log("Соединение установлено.");
};

socket.onclose = function(event) {
    if (event.wasClean) {
        console.log('Соединение закрыто чисто');
    } else {
        console.log('Обрыв соединения');
    }
    console.log('Код: ' + event.code );
};
```

WEBSOCKET (НА КЛИЕНТЕ)

```
socket.onmessage = function(event) {  
    console.log("Получены данные " + event.data);  
};
```

```
socket.onerror = function(error) {  
    console.log("Ошибка");  
};
```

WEBSOCKET (HA CEPBEPE)

```
const WebSocket = require('ws');
```

```
const wss = new WebSocket.Server({ port: 8080 });
```

```
wss.on('connection', function connection(socket, req) {  
    socket.on('message', function incoming(message) {  
        console.log('received: %s', message);  
    });  
});
```

```
let interval = setInterval(function(){  
    socket.send('something');  
}, 5000);
```


WEBSOCKET (HA CEPBEPE)

```
socket.on('close', function connection(code, reason) {  
    clearInterval(interval);  
    console.log('disconnect client Code: ' + code);  
});
```

```
socket.on('error', function connection(error) {  
    clearInterval(interval);  
    console.log('Error: ' + error);  
});
```

```
});
```

WEBSOCKET (HA CEPBEPE C EXPRESS)

```
const express = require('express');
const http = require('http');
const WebSocket = require('ws');

let app = express();

const server = http.createServer(app);
server.listen(8080);

app.use('/', function(req, res, next){
    res.sendFile(__dirname + '/index.html');
});

const wss = new WebSocket.Server({ server });
```

SOCKET.IO (НА КЛИЕНТЕ)

```
<script src="/socket.io/socket.io.js"></script>
<script>
  let socket = io.connect('http://localhost:8080');
  socket.on('news', function (data) {
    console.log(data);
    socket.emit('my other event', { my: 'data' });
  });
</script>
```

SOCKET.IO (HA CEPBEPE)

```
const express = require('express');
const app = express();
const server = require('http').Server(app);
const io = require('socket.io')(server);
server.listen(8080);

app.get('/', function (req, res) {
  res.sendFile(__dirname + '/index.html');
});
io.on('connection', function (socket) {
  socket.emit('news', { hello: 'world' });
  socket.on('my other event', function (data) {
    console.log(data);
  });
});
```

ПОЛЕЗНАЯ ССЫЛКА

<https://learn.javascript.ru/xhr-longpoll>

- COMET с XMLHttpRequest: длинные опросы

<https://learn.javascript.ru/websockets>

- про WebSocket

<https://github.com/websockets/ws>

- ws: a Node.js WebSocket library

<https://github.com/socketio/socket.io>

- socket.io