

Сокеты.

В ситуациях где время соединения с сервером критично, необходимо использовать сокетное соединение. Пример работы сокетного соединения реализован на Node.js и подключаемых к нему модулях Express и socket.io. Программа представляет из себя, что-то наподобие чата, только вместо сообщений каждые 10 секунд генерируются целые число до 100 и отправляются с именем «пользователя» в чат. При вступлении в диалог нового «пользователя» ему предоставляется возможность представиться. Сам код не представляет ничего сложного и особенно в комментариях не нуждается.

Серверная часть

```
//сервер для socket

var socket = require('socket.io');
var express = require('express');
var app = express();
var io = socket.listen(app.listen(8080));

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

io.sockets.on('connection', function(client){
  console.log('connected...');
  client.on('message', function(data){
    client.emit('response', data);
    client.broadcast.emit('response', data);
  })
})
})
```

Клиентская часть (почти весь html и стили)

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="/socket.io/socket.io.js"></script>
  <style>
    #chat{
      border: 1px solid #757584;
      width: 150px;
      border-radius: 5px;
      margin: 10px 0px;
      padding: 2px;
    }
  </style>
  <title>Поток данных</title>
</head>
<body>
  <div id="chat"></div>
  <input type="button" value="ПУСК" id = 'start'>
  <input type="button" value="СТОП" id = 'stop'>
```

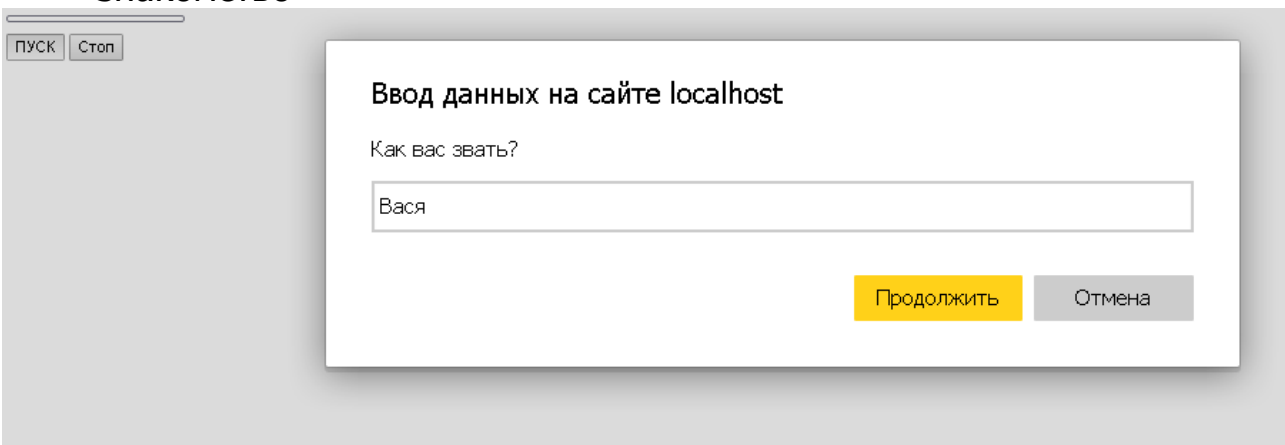
Клиентский скрипт

```
<script>
  var socket = io.connect('http://localhost:8080');
  //генерация массива случайных чисел через интервал времени их отправки через сокет
  function generate(){
    var rnd = Math.floor(Math.random() * Math.floor(100));
    socket.emit('message', {name: name, msg: rnd});
    time = setTimeout(generate, 10000);
  }
  //функция начала вещания чисел в чат
  function startSpeak(){
    //представьтесь
    name = prompt('Как вас звать?', 'Аноним');
    generate(name);
  }
  //функция прекращения вещания
  function stopSpeak(){
    clearTimeout(time)
  }
  socket.on('response', function(data){
    document.getElementById('chat').innerHTML += data.name + ' ' + data.msg + '<br>';
  })
  document.getElementById('start').onclick = startSpeak;
  document.getElementById('stop').onclick = stopSpeak;
</script>
</body>
</html>
```

Виды в браузере
начальный вид



ЗНАКОМСТВО



через несколько пользователей и времени

Вася 16
Петя 6
Петя 15
Петя 23
Петя 76
Петя 59
Петя 90
Света 12
Петя 80
Петя 20
Света 6
Петя 52
Петя 20
Света 8
Петя 60
Петя 44
Света 83
Петя 80
Петя 71
Петя 13
Петя 8

ПУСК

Стоп

Каждый участник может выйти в любой момент по кнопке Стоп

Как-то так можно применять сокеты, в приложениях где нужно постоянно (без переподключений) отслеживать состояние клиента и/или сервера