

# Node.js05

---



## 本节知识点

- 前端轮循获取数据
- SSE (server send event) 服务器推送数据;
- websocket协议
- nodejs中使用socket.io模块实现长连接;
- 网页版聊天系统;

## 课堂目标

- 理解数据推送含义
- 学会sse实现数据推送
- 了解websocket协议
- 会使用socket.io模块

## 前端轮循获取数据

- 前端轮循环, 获取数据;
  - 循环ajax请求, 获取数据
  - 消耗性能, 消耗资源, 不推荐;

## SSE (server send event) 服务器推送数据;

Server-sent events:使用server-sent 事件的方法,服务器可以在任何时刻向我们的web页面推送数据和信息.这些被推送进来的信息可以在这个页面上作为事件+data来处理.

- 服务端设置
  - 设置头部

```
"Content-type", "text/event-stream"
```
  - 返还数据格式

```
data: 声明数据开始
```

```
\r\n\r\n 标志数据结尾
```
- 前端获取

```
let source = new EventSource("/test");
source.onopen = function(){
    console.log("连接建立...",this.readyState);
}
// console.log(source)
source.onmessage = function(evnet){
    console.log("数据是: ",evnet.data)
}
source.error = function(err){
    return console.log('err');
}
```

◦ readyState 代表连接状态:

- 0 CONNECTING (0)。
- 1 OPEN (1),
- 2 CLOSED (2`)

## websocket

- WebSocket 是 HTML5 开始提供的一种在单个 TCP 连接上进行全双工通讯的协议;
- 创建websocket服务器;

```
var WebSocketServer = require('ws').Server,
wss = new WebSocketServer({ port: 8181 });
wss.on('connection', function (ws) {
    console.log('client connected');
    ws.on('message', function (message) {
        //监听接收的数据
        console.log(message);
    });
    setInterval(() => {
        let somedata = {
            name:"张三",
            age:20
        }
        ws.send(JSON.stringify(somedata));
    }, 1000);
});
```

- 客户端代码
  - 建立握手

```
var ws = new WebSocket("ws://localhost:8181");
```

打开协议

```
ws.onopen = function () {}
```

- 发送数据到服务端

```
ws.send("客户端数据");
```

- 关闭协议:关闭协议后不能发送数据

```
ws.close();
```

- 接收消息

```
ws.onmessage = function(e){  
    // console.log(e.data);  
}
```

## socket.io模块

- 服务端

```
const server = require('http').createServer(app.callback());  
const io = require('socket.io')(server);  
server.listen(3000);
```

- 客户端

```
let socket = io.connect();  
this.emit("clientfn","hello i am client");  
socket.on("message",function(data){})
```

## 课程总结

- 前端轮循获取数据
- SSE (server send event) 服务器推送数据;
- websocket协议
- nodejs中使用socket.io模块实现长连接;

## 下节预告

- 前后端交互01

