

```
1 // @author : bilibili:一只斌 / mail: tuduweb@qq.com
2 var express = require('express');
3 var app = express();
4 var http = require('http').createServer(app);
5
6 var fs = require('fs');
7 let sslOptions = {
8   key: fs.readFileSync('C:/privkey.key'), // 里面的文件替换成你生成的私钥
9   cert: fs.readFileSync('C:/cacert.pem') // 里面的文件替换成你生成的证书
10 };
11
12 const https = require('https').createServer(sslOptions, app);
13
14 var io = require('socket.io')(https);
15
16 var path = require('path');
17 app.use(express.static(path.join(__dirname, 'public')));
18
19
20 app.get('/', (req, res) => {
21   res.sendFile(__dirname + '/index.html');
22 });
23
24 app.get('/camera', (req, res) => {
25   res.sendFile(__dirname + '/camera.html');
26 });
27
28
29 io.on("connection", (socket) => {
30   // 连接加入子房间
31   socket.join( socket.id );
32
33   console.log("a user connected " + socket.id);
34
35   socket.on("disconnect", () => {
36     console.log("user disconnected: " + socket.id);
37     // 某个用户断开连接的时候, 我们需要告诉所有还在线的用户这个信息
38     socket.broadcast.emit('user disconnected', socket.id);
39   });
40
41   socket.on("chat message", (msg) => {
42     console.log(socket.id + " say: " + msg);
43     // io.emit("chat message", msg);
44     socket.broadcast.emit("chat message", msg);
45   });
46
47   // 当有新用户加入, 打招呼时, 需要转发消息到所有在线用户。
48   socket.on('new user greet', (data) => {
49     console.log(data);
50     console.log(socket.id + ' greet ' + data.msg);
51     socket.broadcast.emit('need connect', {sender: socket.id, msg: data.msg});
52   });
53   // 在线用户回应新用户消息的转发
54   socket.on('ok we connect', (data) => {
55     io.to(data.receiver).emit('ok we connect', {sender: data.sender});
56   });
57
58   // sdp 消息的转发
59   socket.on('sdp', (data) => {
60     console.log('sdp');
```

```
61     console.log(data.description);
62     //console.log('sdp: ' + data.sender + ' to: ' + data.to);
63     socket.to( data.to ).emit( 'sdp', {
64         description: data.description,
65         sender: data.sender
66     } );
67 } );
68
69 //candidates 消息的转发
70 socket.on( 'ice candidates', ( data ) => {
71     console.log('ice candidates: ');
72     console.log(data);
73     socket.to( data.to ).emit( 'ice candidates', {
74         candidate: data.candidate,
75         sender: data.sender
76     } );
77 } );
78
79 });
80
81
82 https.listen(443, () => {
83     console.log('https listening on *:443');
84 });
```