1. 什么是websocket?

Websocket是html5新增的协议，通过websocket可以让客户端与服务端双向通信，也就是说除客户可以发请求给服务器外，服务器也可以主动推送消息给客户端。

1. Websocket特点？
2. 客户端与服务端一旦建立连接，该通道会一直保持；
3. 节省带宽与服务器资源
4. 使用socket.io第三包：

1、安装：

在node当中第三方有 socket.io包

npm i express socket.io -S // -S等价于--save

2、使用：

socket.emit( '事件','消息' )

socket.on( '事件',function(message){

} )

emit 发布 发消息

on 兼听事件

socket.emit 只让自己收到消息

socket.broadcast.emit 除了自己其它人会收到消息

io.emit 都会收到消息

1. Socket.io包基本用法：
2. 服务端：

const express = require('express')

const app = express();

const path = require('path');

const server = require('http').createServer(app);

const io = require('socket.io')(server);

//注意：服务一旦启动会创建隐藏文件： /socket.io/socket.io.js

//监听事件

io.on('connection', (socket) => {

    // console.log('ok...');

    //接收消息:

    socket.on('tv1', (msg) => {

        // console.log(msg);

        //发送消息：

        // socket.emit('cctv13', `Server:${msg}`);

        //发送广播消息：

        // socket.broadcast.emit('cctv13', `Server:${msg}`);

        //给所有客户端发送消息：

        io.emit('cctv13', `Server:${msg}`)

    });

});

//静态资源托管

app.use(express.static(path.resolve(\_\_dirname, 'views')));

server.listen(3000);

1. 客户端：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <script src="/socket.io/socket.io.js"></script>

    <script>

        let sockets = io();

        //sockets.emit('事件名','消息内容');

        //sockets.on('事件名',回调函数);

        //发送消息

        sockets.emit('tv1', 'Hello  Server');

        //接收消息

        sockets.on('cctv13', (m) => {

            console.log(m);

        });

    </script>

</head>

<body>

</body>

</html>

1. 使用Koa生成器快速部署项目：

1、安装koa生成器：

npm i koa-generator -g

2、使用kao生成器部署项目：

koa2 -e[使用ejs模板引擎] baofeng(项目名称)

3、切换到baofeng项目目录中：

cd baofeng

4、使用npm install安装项目依赖包：

npm install

1. Koa与express的区别？

1、express有中间件、koa没有中间件

2、koa使用级联操作替代express中的回调方法

3、koa中的context封装了express中的request对象、response对象

1. Koa与express怎样取舍？

Express有丰富的中间件及包，如果想使用这些封装好的功能那就选择express，而koa强调性能，如果对项目定制要求较高的话则选择koa，另外如果之前用express开始旧项目那还继续使用express。

1. webpack工作过程：

webpack是自动化打包工具，它可以根据配置文件去查询项目目录，将javascript模块文件及其它语言的代码根据配置文件中的规则去转换打包生成浏览器能识别的文件。