# 实验12：socket Io数据传递与管理

实验背景

利用已掌握的知识，完成实例。如图所示：

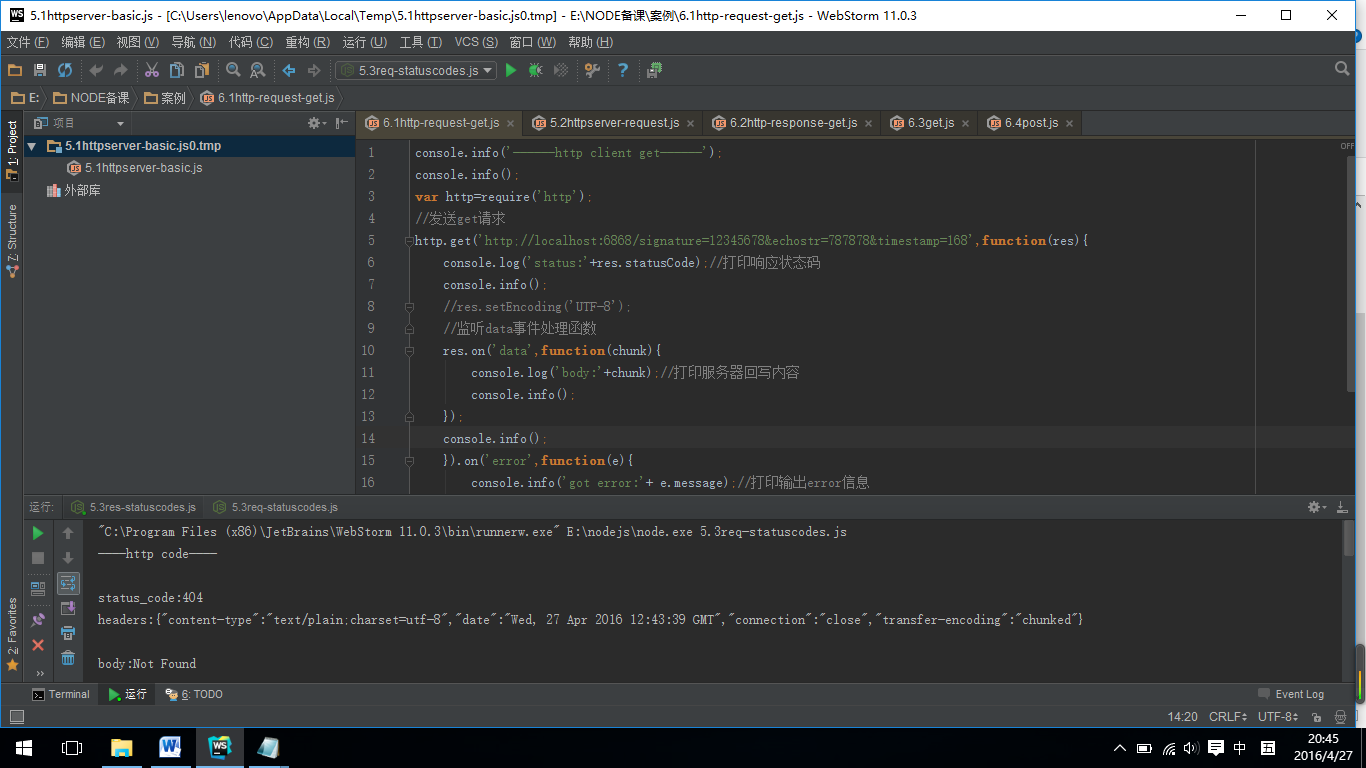


图1-1

实验目的

1、掌握安装socket Io

2、掌握创建服务器

3、掌握创建客户端

4、掌握socket.Io入门应用

实验分析

本次实验要完成socket.Io入门小应用。

实验步骤

1、添加index.html到myapp文件夹

<!doctype html>

<html>

<head>

<title>Socket.IO chat</title>

<style>

\* { margin: 0; padding: 0; box-sizing: border-box; }

body { font: 13px Helvetica, Arial; }

form { background: #000; padding: 3px; position: fixed; bottom: 0; width: 100%; }

form input { border: 0; padding: 10px; width: 90%; margin-right: .5%; }

form button { width: 9%; background: rgb(130, 224, 255); border: none; padding: 10px; }

#messages { list-style-type: none; margin: 0; padding: 0; }

#messages li { padding: 5px 10px; }

#messages li:nth-child(odd) { background: #eee; }

</style>

</head>

<body>

<ul id="messages"></ul>

<form action="">

<input id="m" autocomplete="off" /><button>Send</button>

</form>

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

<script src="http://code.jquery.com/jquery-1.11.1.js"></script>

<script>

var socket = io();

$('form').submit(function(){

socket.emit('chat message', $('#m').val());

$('#m').val('');

return false;

});

socket.on('chat message', function(msg){

$('#messages').append($('<li>').text(msg));

});

</script>

</body>

</html>

2、修改app.js

var app = require('express')();

var http = require('http').Server(app);

var io = require('socket.io')(http);

app.get('/', function(req, res){

res.sendfile('index.html');

});

io.on('connection', function(socket){

console.log('a user connected');

socket.on('chat message', function(msg){

console.log('message: ' + msg);

io.emit('chat message', msg);

});

});

app.set('port', process.env.PORT || 3000);

var server = http.listen(app.get('port'), function() {

console.log('start at port:' + server.address().port);

});扩展练习

1.思考并实现，完成案例《socket.Io入门小应用》并实现测试

实验思考

1.思考并实现，独立设计一个小应用。