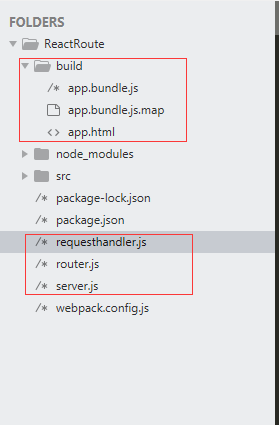
**服务器路径：**



**server.js代码**

var http = require('http');

var router = require('./router');

http.createServer(function (request, response) {

    // 路由

var res = router.route(request.url);

// 状态码

response.writeHead(res.code,{});

// 响应内容

response.write(res.content);

response.end();

}).listen(8888);

// 终端打印如下信息

console.log('Server running at <http://127.0.0.1:8888/');>

**router.js**

var url = require('url');

var querystring = require('querystring');

var requesthandler = require('./requesthandler');

var rootDir = "/build";

// todo 后期可以扩展路由部分

function route(request\_url) {

// 解析地址

var url\_parse = url.parse(request\_url);

var url\_path = url\_parse['path'];

var res = {};

var code = 200;

// 判断路径是否存在

var content = requesthandler.handle(rootDir + url\_path);

return {

'code':code,

'content':content

}

}

exports.route = route;

**requesthandler.js**

var fs = require('fs');

// 默认检索文件，可以修改为配置项

var default\_files = [

'index.htm',

'index.html'

];

function handle(url\_path) {

// 文件路径构造

var file\_path = '.' + url\_path;

// 判断路径是否存在

var f\_path = \_findFile(file\_path);

if(!f\_path) {

return 'the path "'+file\_path+'" is not exist';

}else {

var content = fs.readFileSync(f\_path);

return content;

}

}

/\*\*

\* 检索目录下的不同文件名是否存在，建立优先级

\* @return {[type]} [description]

\*/

function \_findFile(file\_path) {

var is\_file = \_isFile(file\_path);

// 如果文件存在，直接返回路径

if(is\_file) {

return file\_path;

}

// 文件不存在，构造路径寻找文件

var regex = /^\.\/[\w]\*([\/][\w]+)?/;

var regex\_res = file\_path.match(regex);

// 匹配出目录路径 ./vue/test/s => ./vue/test

if(!regex\_res) {

return '';

}else {

// 这里没有使用forEach，因为会遍历所有的值

for(var i=0; i<default\_files.length; i++) {

var new\_path = regex\_res[0]+'/'+default\_files[i];

if(\_isFile(new\_path)) {

return new\_path;

}

}

}

}

/\*\*

\* 文件是否存在

\* @param {[type]} file\_path [description]

\* @return {Boolean} [description]

\*/

function \_isFile(file\_path) {

try {

// 同步会抛出异常，所以用try来保证正常执行

var stat = fs.statSync(file\_path);

// 返回是否是文件

return stat.isFile();

}catch(e) {

console.info(e.message);

}

}

exports.handle = handle;

**运行服务器**

cmd下切换到ReactRoute目录，执行以下代码：

node server.js