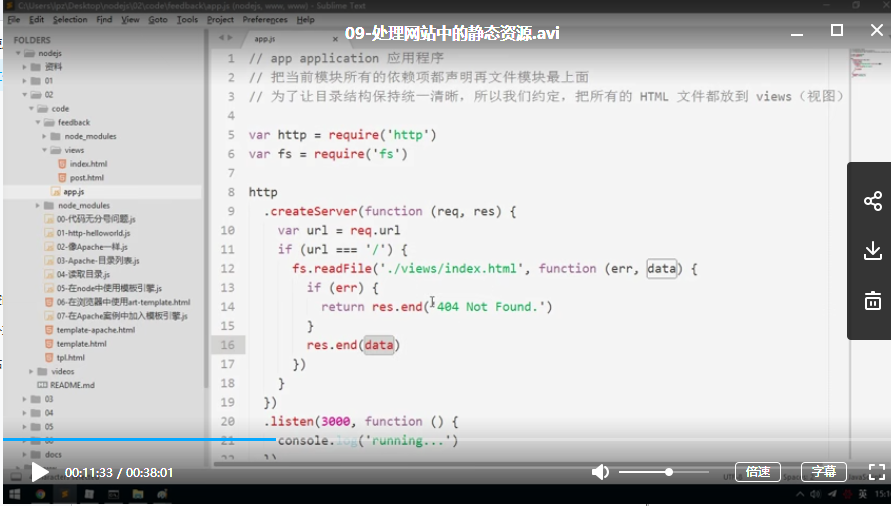
09先进行静态资源的处理

1. app.js

写代码：如果url=/



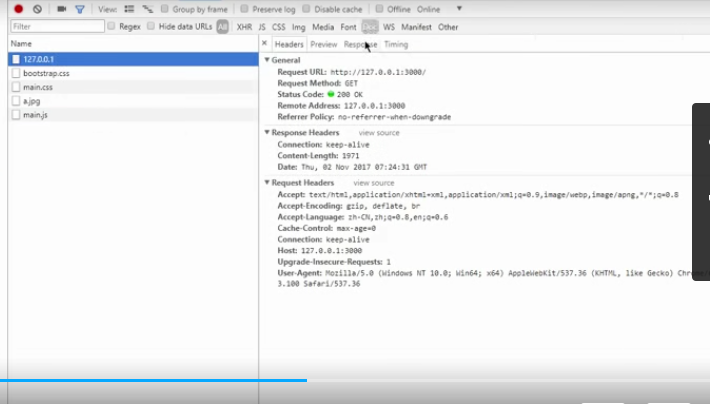
1. 然后启动服务器

发现服务端给我们响应的没有样式

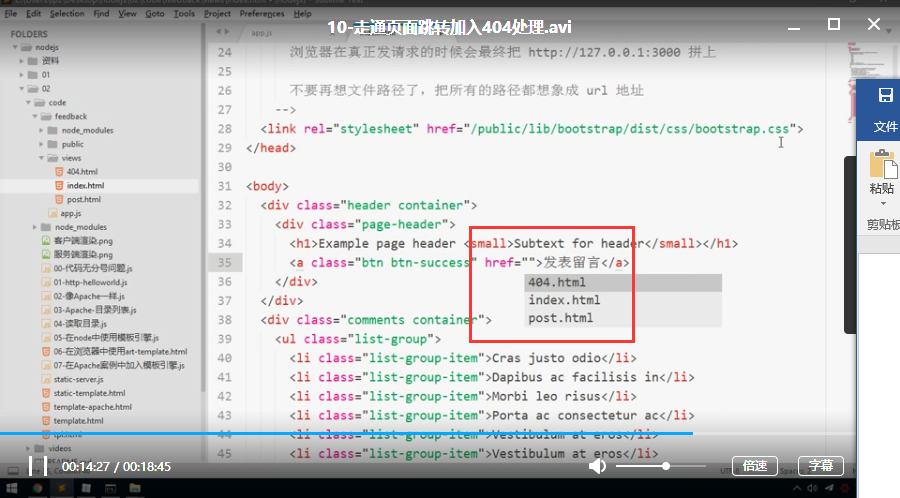
打开network

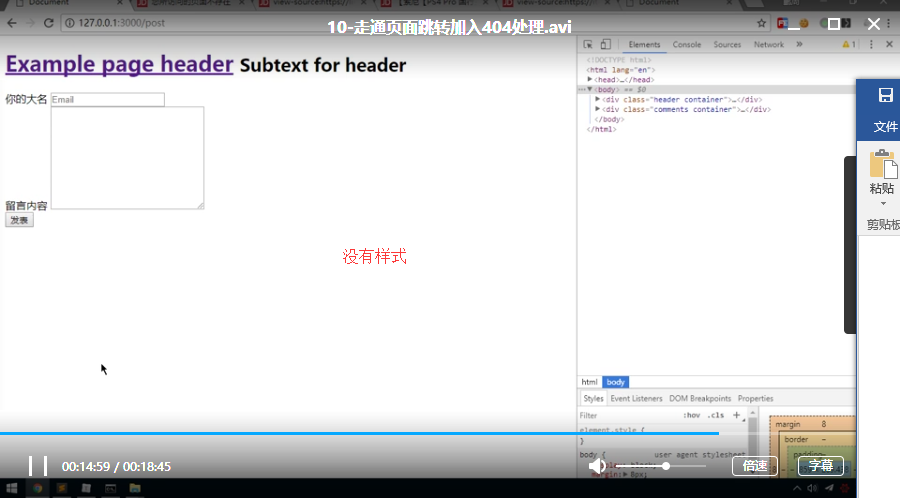


多写几个不存在的，会发现network中有



/404.html

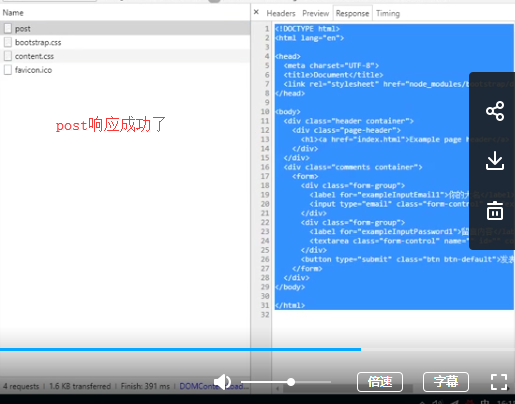


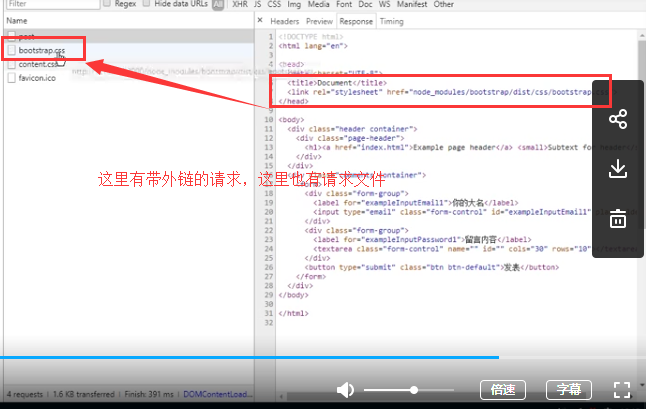


分析：为什么没有样式：

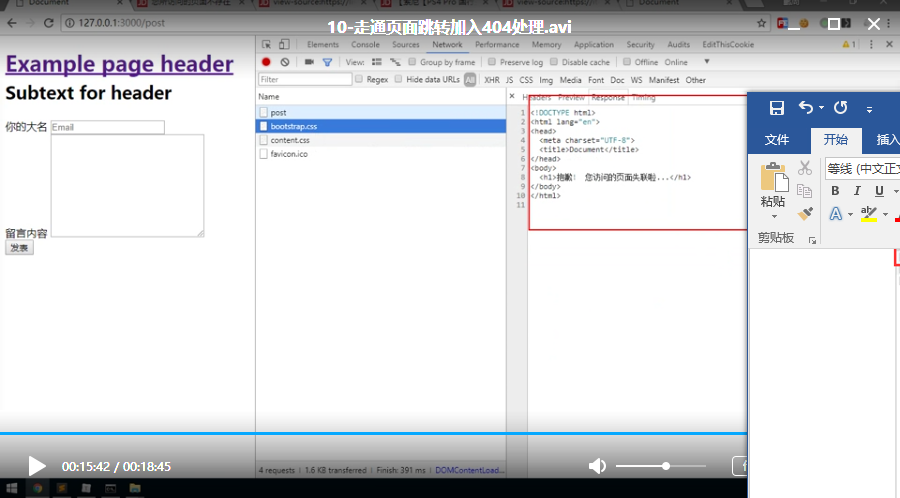
1. 打开network---》刷新

看post响应





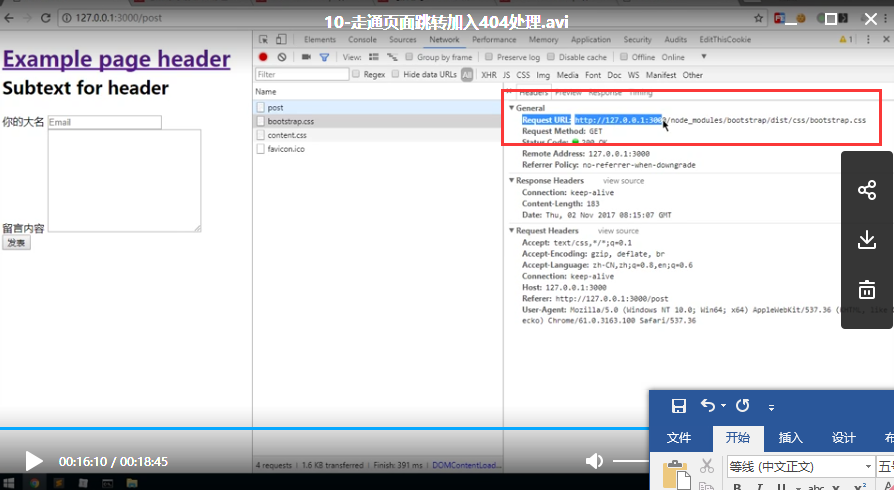
打开这个bootstrap看看响应的是什么



为什么？看请求路径，出现页面失联，说明服务端走到



打开header就可以看到



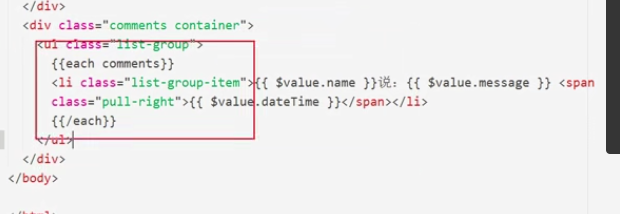
这里没有改，找到pst 改成/public

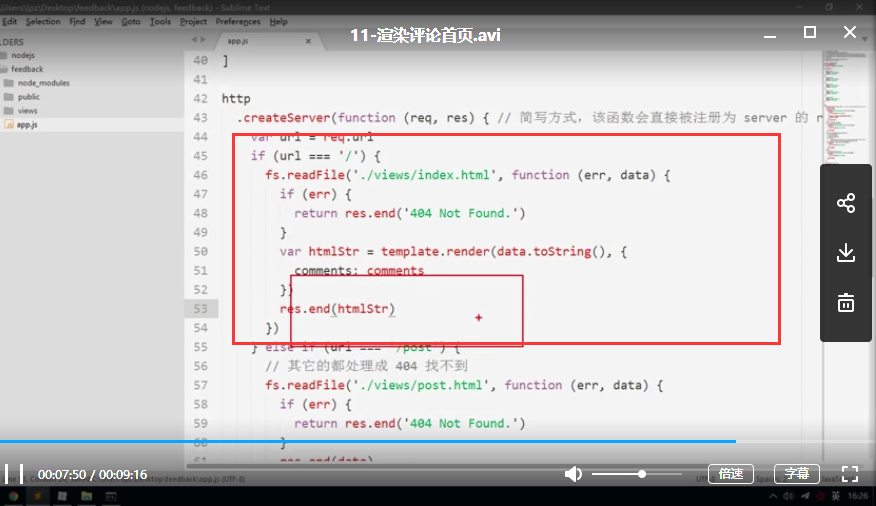
1. 渲染数据：

要有数据，渲染到li 中

写模板：

Pull-right





3)点发表，然后将数据添加到数组里面

做简单的验证



以前：

表单中需要提交的空件元素必须有name属性

日期由服务端生成，

表单提交：默认的提交行为

Action就是表单请求的地址 method:请求方法

异步提交（需要我们单独写代码处理）

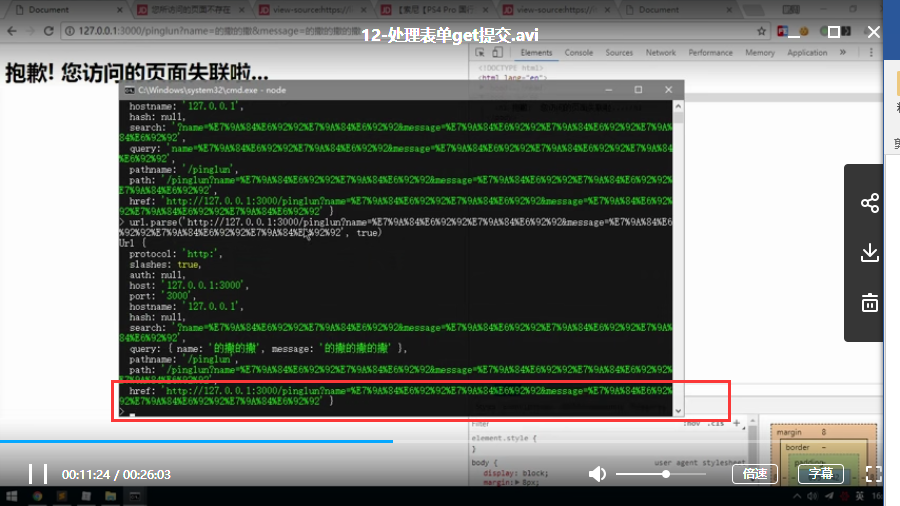




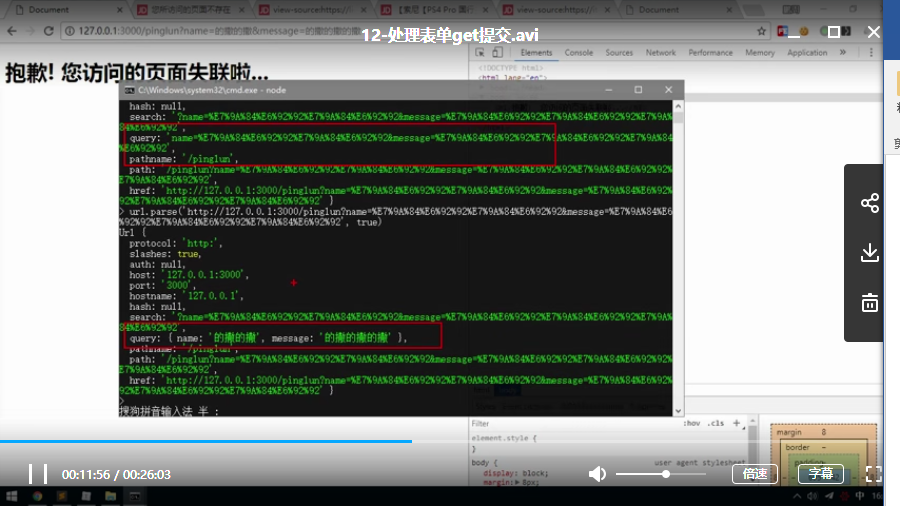


怎么判断：

有一个模块叫url,



看看有没有变化



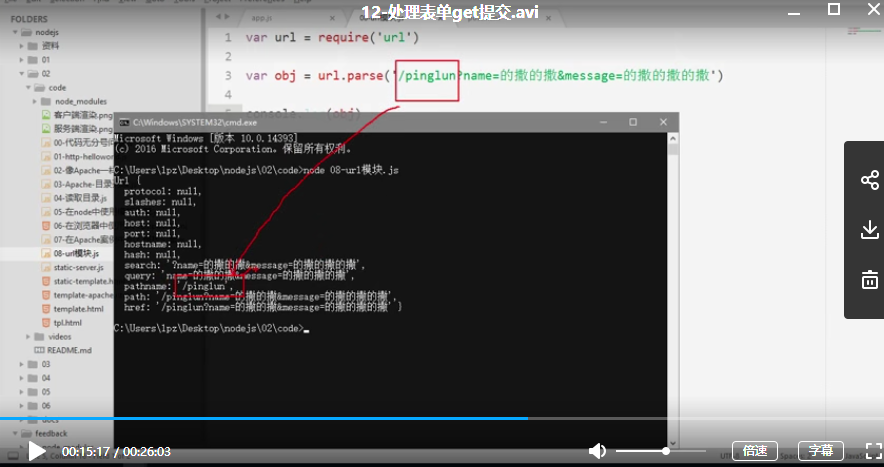
要拿到name .query.name就是里面的name

由于表单提交的请求路径，由于其中具有用户动态填写的内容，

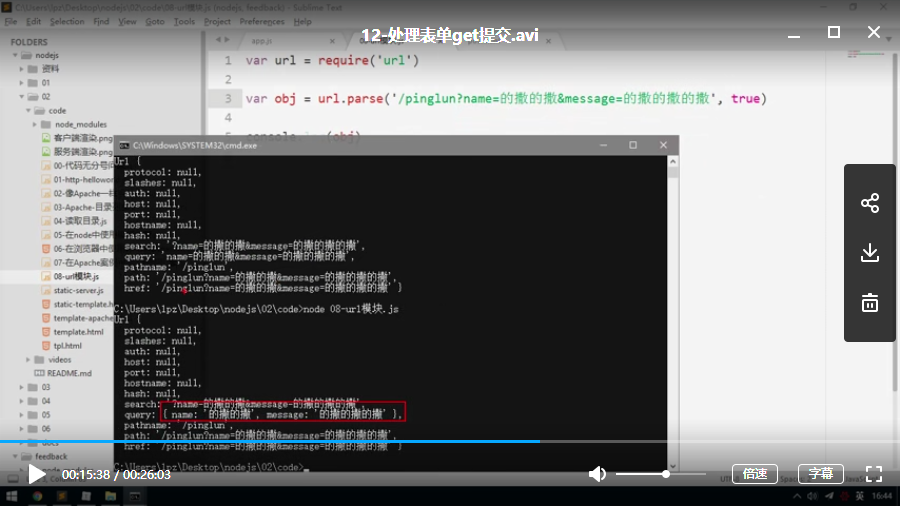
所以你不可能通过判断完整的url路径来处理请求，因为这个url路径可能是动态变化的

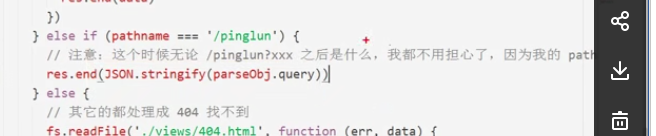
结论：对于我们来讲，其实只要判定，如果你的请求路径是/pinglun的时候，那我就认为你提交表单的请求过了

演示：url模块

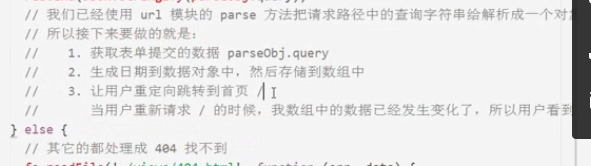


将后面的字符串顺带转化为对象

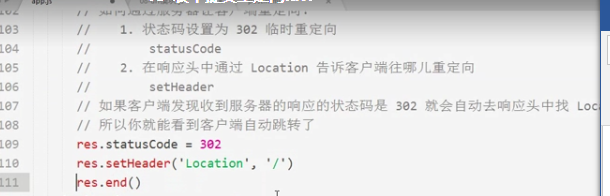




将你提交的数据响应回来

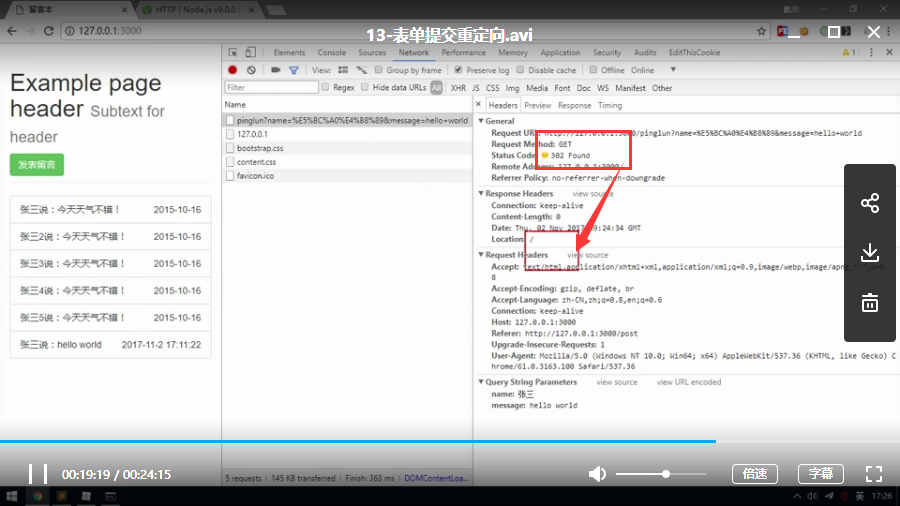






一次请求对应一次响应

往前插unshift



看到状态为302就会从新请求location

总结：

Node不适合从来没有接触过服务端的学习，

// /index.html

// 开放public 时候响应，public目录具体资源

// /post 响应post.html

// /pinglun

4.1处理表单提交数据，

4.2存储表单数据

4.3让表单重定向

StatusCode

setHeader

(1)请求index.html

var http = require('http');

var fs= require('fs');

var server = http.createServer();

server.on('request',function (req, res){

var url = req.url;

//如果 请求/返回的是/index.html

if (url === '/') {

fs.readFile('./views/index.html', function (err, data) {

res.end(data);

})

}

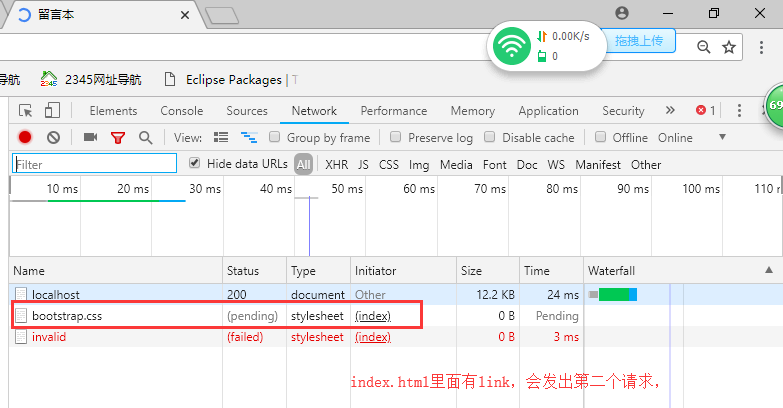
});

server.listen(3000,function (){

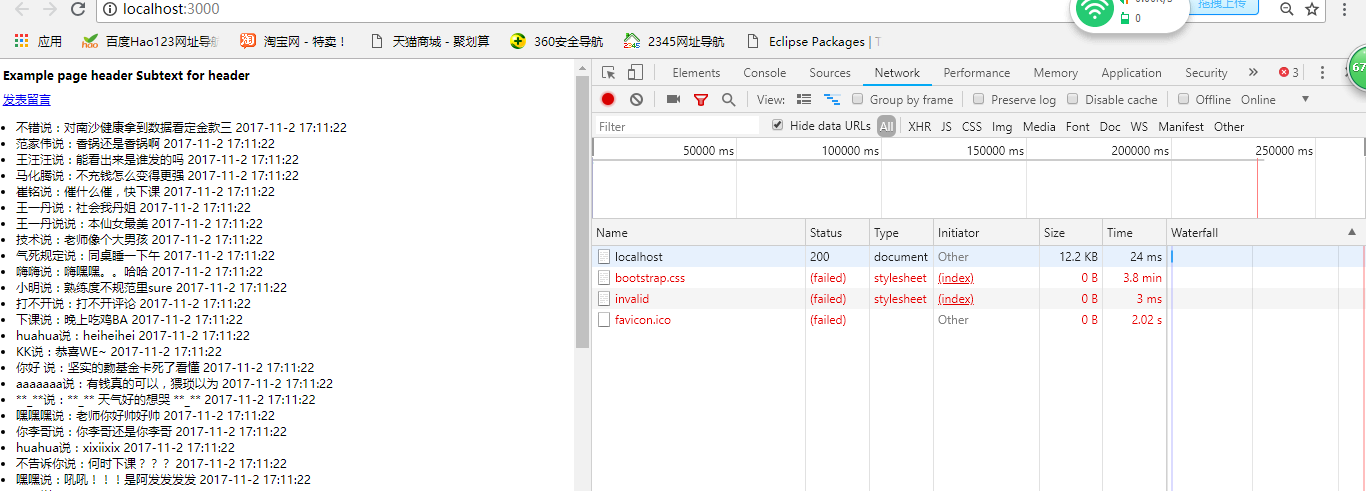
console.log('server is running');

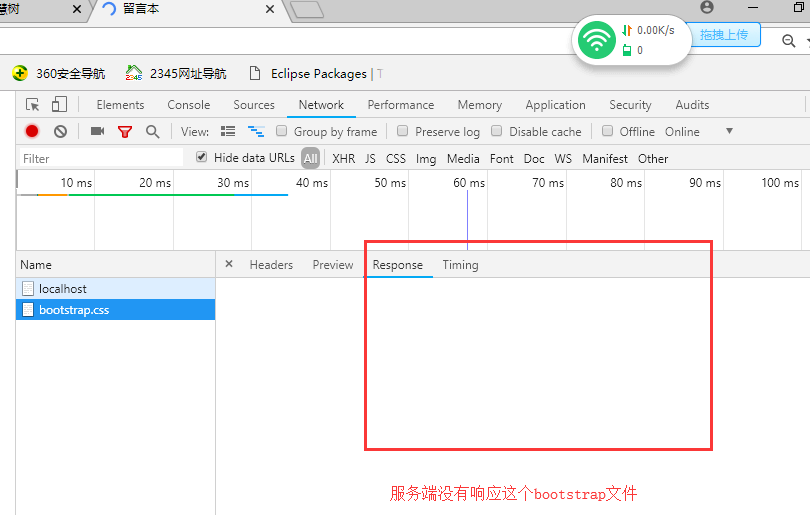
})

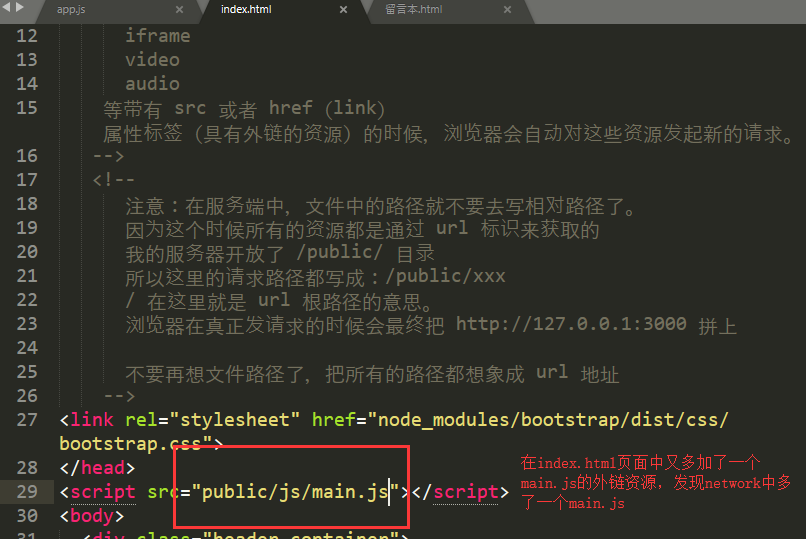
(2

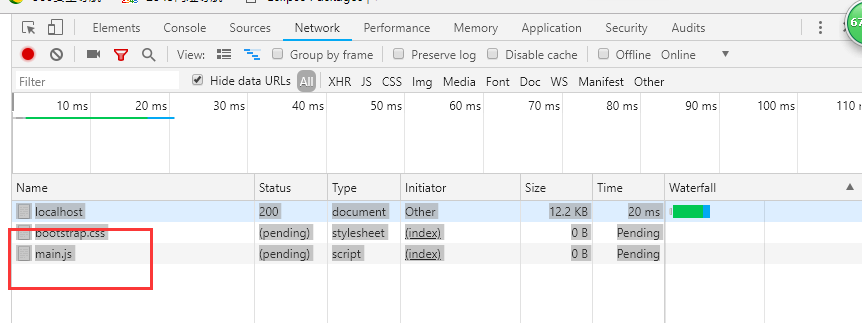
)

(3)关掉服务器之后，发现可以请求到页面，但是没有样式

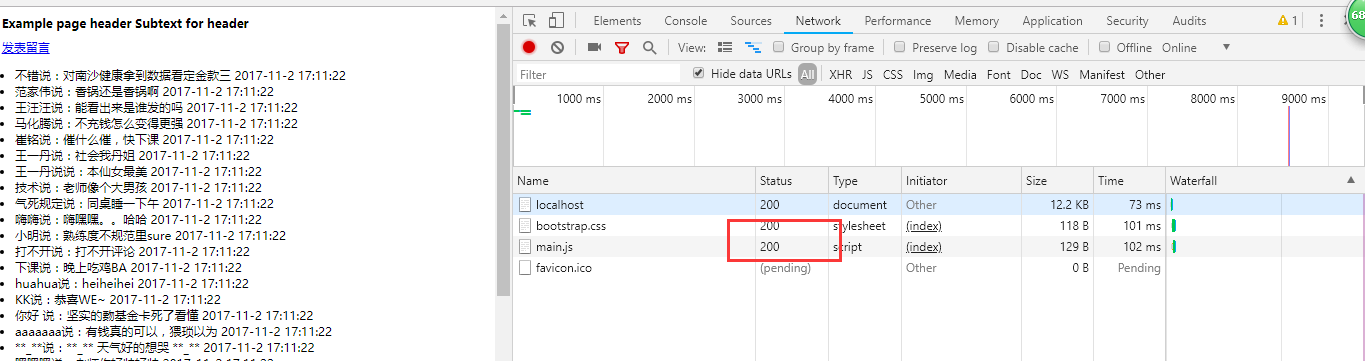




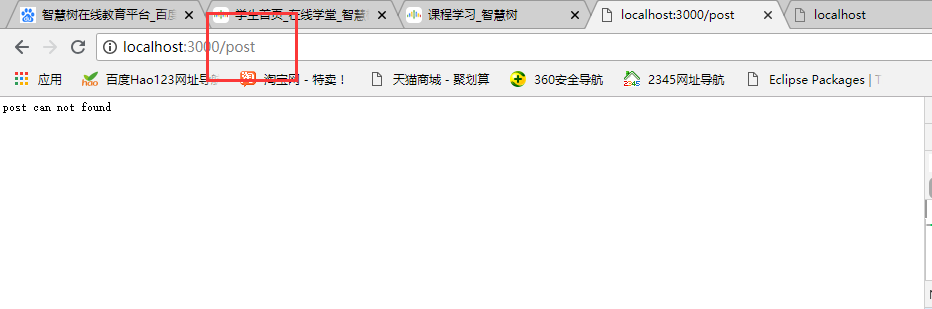




将script css都放在一个公共的public中，只要一遇到这些外链，就直接到public中去拿



但是还是没有渲染上样式



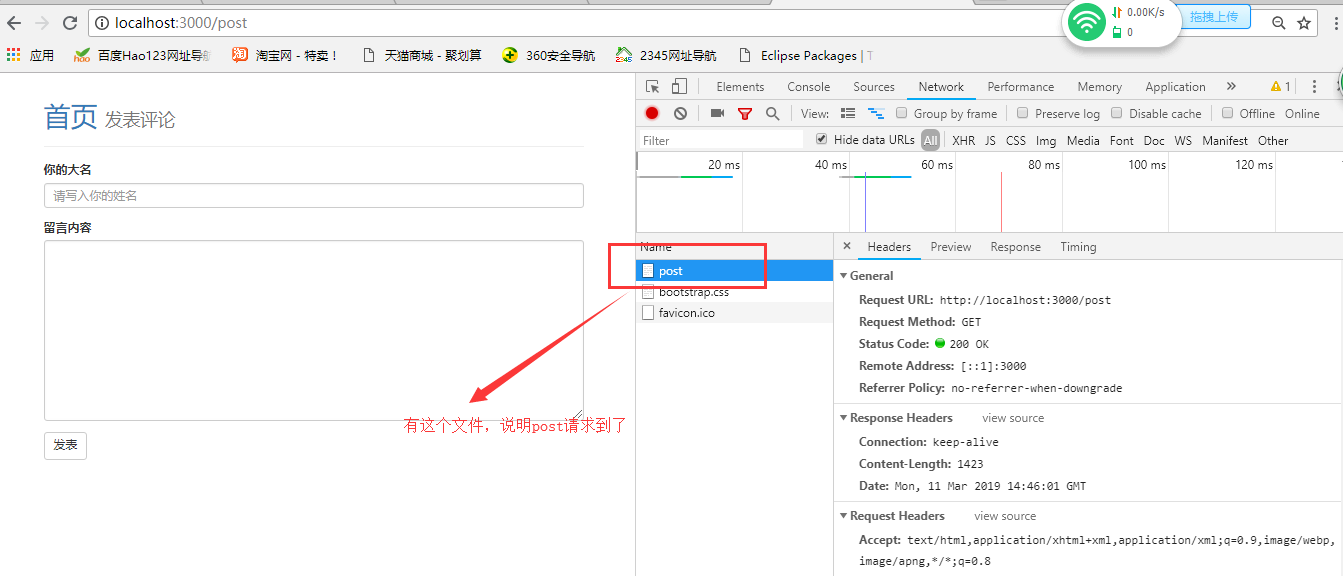
输出了post can not found说明代码已经执行到这一步

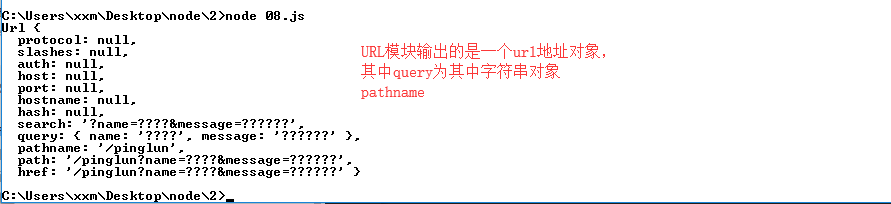


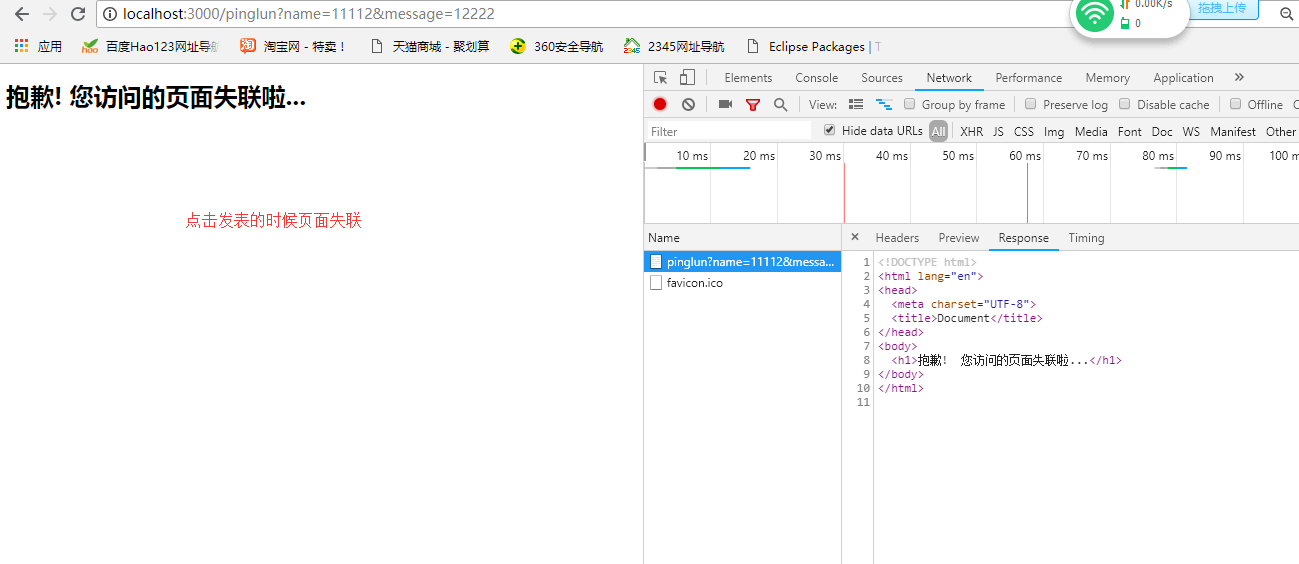
发现是我们的读取路径错了



打开控制台







可能是从定向出错，这里是写的路径不对

