Expree框架

基于NodeJS平台，快速、开放、极简的web开发框架

中文网址：www.expressjs.com.cn

下载：npm install express

1、创建web服务器

const express = require("express")//引入

const server = express();//创建web服务器

server.listen(8081) //设置端口

2、路由

浏览器端发来请求，通过路由来处理特定的请求并作出响应

路由组成：请求的方法、请求的URL、回调函数(作出响应)

response响应对象：

response.send() 设置响应的内容并发送

response.sendFile() 响应文件并发送，需要使用绝对路径\_\_dirname

response.redirect() 响应的重定向，跳转到另一个URL

request请求对象：

request.url 获取请求的URL

request.method 获取请求的方法

request.query 获取以查询字符串传递的数据，格式为对象

|  |  |  |
| --- | --- | --- |
| 传递数据方式 | 格式 | 获取 |
| get 查询字符串 | keyword=手机 | request.query |
| post传递 | URL不可见，通过请求体 | 以流的方式传递，通过事件获取， request.on('data',(chunk)=>{ chunk;//格式为buffer，转字符串后为查询字符串，使用querystring.parse()解析为对象 }) |
| 路由传参 | http://127.0.0.1:8081/package/add/pc | 通过形参来接收 server.get("/package/:add/:pc",(request,response)=>{ request.params;//获取数据，格式为对象 }) |

路由器的使用

路由器：

const r = express.Router();

r.get("/list",(req,res)=>{ //往路由器添加路由

})

module.exports = r; //导出路由器

web服务器引入并挂载。

app.use(添加的前缀,引入的路由器)

创建商品路由器product.js，添加若干个路由，在服务器下引入并挂载

中间件

处于浏览器和路由之间，作用是拦截对路由的请求，最终为路由所服务。

express中间件分为应用级中间件、路由级中间件、内置中间件、第三方中间件、错误处理中间件

①应用级中间件

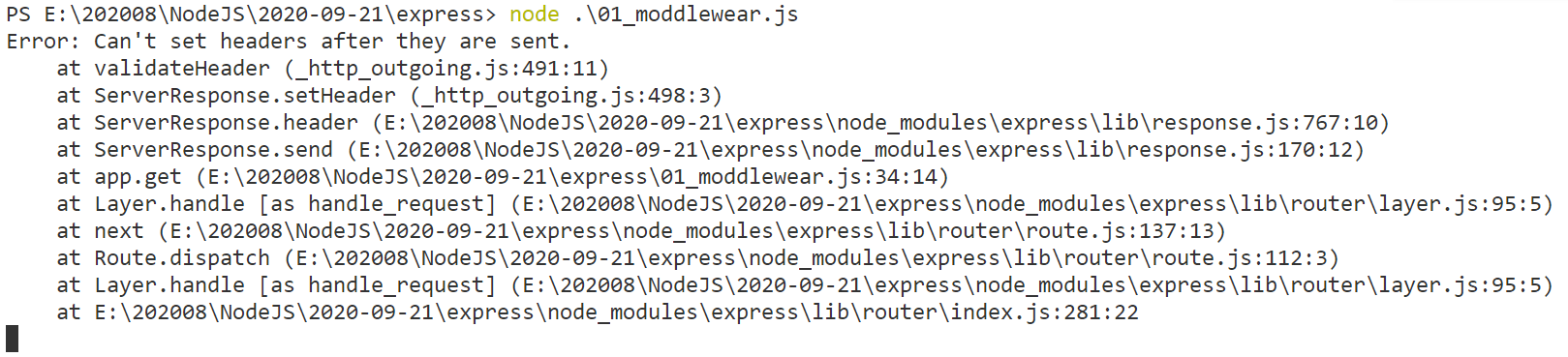
app.use("要拦截的URL",(req,res,next)=>{

})

一旦拦截到对url的请求，会自动调用回调函数，中间件的回调函数也有请求对象和响应对象，回调函数中还有第三个参数next，是一个函数，用于往后执行下一个中间件或者路由。

Error can`t set headers after they are sent

路由中有多次send执行



②路由器中间件

③内置中间件

托管静态资源，当浏览器端请求服务器端静态资源html、css、js、图像，不需要创建路由响应文件，而是让浏览器自动去找所需要的文件。

app.use(express.static('目录的路径'))

④第三方中间件

body-parser

属于第三方模块，需要先下载安装

下载安装 npm install body-parser

通常会和express附带下载

1、引入body-parser模块

const bodyParser = require("body-parser");

2、使用body-parser中间件，可以将所有post请求数据解析为对象。

app.use(bodyParser.urlencoded({extended:false}))//false->不使用第三方查询字符串模块

3、在路由中获取post请求数据，格式为对象。

request.body

mysql模块

Nodejs专门用于操作mysql数据库的模块

mysql连接：

普通连接：

createConnection()

query("SQL语句",回调函数) //通过回调函数获取SQL命令的结果

连接池：

createPool()

query("SQL语句",回调函数)

query("select \* from emp where sex=0",(err,data)=>{

err;//可能产生的错误

data;//SQL命令的结果

})

SQL注入：在SQL命令中，让用户提供的值里边添加了额外的条件。

解决SQL注入：对用户提供的值进行过滤。

mysql提供了占位符(?)，先对用户提供的值过滤，过滤后在替换占位符

占位符：

for(var i=0;i<3;i++){ //函数自调用立即执行

setTimeout((function(){

console.log(i);

})(),0);

console.log(i)

}