express框架

基于nodejs的，快速、开发、极简的web开发框架

www.expressjs.com.cn

安装 npm install express

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

var server=express() //创建web服务器

server.listen(8080) //设置端口

（1）路由

当浏览器向web服务器发请求，web服务器根据请求的方法和请求的url来做出响应

三要素：请求的方法、请求的URL、响应（回调函数）

响应的对象（res）

send() 响应的内容，只能响应一次

sendFile() 响应文件，必须使用绝对路径（\_\_dirname+‘/xxx.html’）;

redirect() 响应重定向（跳转）

/\*//引入express

const express=require('express');

//创建web服务器

var server=express()

//设置端口

server.listen(8080);

//路由

//get请求的方法

//参数1：请求的URL

//参数2：响应

server.get('/index',function(req,res){

//req:请求的对象；

//res:响应的对象

res.send('你还好吗');

})\*/

//练习：创建路由，请求的方法为get，请求的URL:/login ,响应'请登录'

/\*const express=require('express')

var server=express();

server.listen(8080)

server.get('/login',function(req,res){

//send调用一次，就代表响应结束；就不允许再次调用

res.send('请登录<br>欢迎');

})\*/

//创建路由，请求的方法get,请求URL:/list,响应的内容’这是商品列表‘

const express=require('express')

var server=express();

server.listen(8080)

server.get('/list',function(req,res){

//res.send('这是商品列表')

//响应listen.html文件到浏览器

res.sendFile(\_\_dirname+'/listen.html')

})

//练习：创建文件03-express，使用express创建web服务器，添加以下路由

get /lidex '这是首页'

get /login 响应文件login.html

post /reg '欢迎注册'

get / 跳转到/index

const express=require('express')

var server=express()

server.listen(8080)

server.get('/index',function(req,res){

res.send('这是首页')

})

server.get('/login',function(req,res){

res.sendFile(\_\_dirname+'/login.html')

})

server.post('/reg',function(req,res){

res.send('欢迎注册')

})

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

res.redirect('/index')

})

请求的对象（req）

req.method 请求的方法

req.url 请求的URL

req.headers 请求的头像

req.query获取请求时查询字符串传递的数据，并格式化为对象

练习 创建文件04-post.js，使用express创建web服务器，添加路由

get /reg 响应文件 reg.html

2、post和get传递数据

post 请求时通过表单提交（暂时）传递数据，服务器端是通过事件的形式获取数据（后期会有简单的方法）

req.on('data',function(buf){

//buf就是获取的数据，格式为buffer,转为字符串后格式为查询字符串，需要借助查询字符串模块格式转化为对象

})

get 请求已查询字符串形式传递数据，服务器端使用req.query获取数据，结果是对象

pos查询字符串传递数据容易被浏览器所缓存，而post传递数据不会出现在地址栏

//练习 创建文件04-post.js，使用express创建web服务器，添加路由

//get /reg 响应文件 reg.html

const express=require('express')

//引入查询字符串模块

//const querystring=require('querystring')

var server=express()

server.listen(8080)

/\*server.get('/reg',function(req,res){

res.sendFile(\_\_dirname+"/reg.html")

})

//根据表单的请求添加对应的路由

//获取post请求传递的数据

//以事件的形式

//当有数据传递自动触发事件

//使用回调函数接收数据

server.post('/myreg',function(req,res){

req.on('data',function(buf){

//buf传递的数据,是buffer形式

//console.log(buf);

var str=buf.toString()

//查询字符串（que）格式化为对象

var obj=querystring.parse(str)

console.log(obj)

})

res.send('恭喜你注册成功')

})\*/

练习添加路由，请求的方法get ,请求的url为/login,响应login.html,在html文件中创建一个登陆的表单

const express=require('express')

var server=express()

server.listen(8080)

server.get('/login',function(req,res){

res.sendFile(\_\_dirname+'/login.html')

//server.post('')

})

//根据表单的请求创建对应的路由

server.get('/mylogin',function(req,res){

console.log(req.query)

res.send('这是列表');

})

3、路由传递数据——-路由传参

这只路由中接收的名称

server，get('/detail/:lid',function(req,res){

req.params//获取路由传递的数据，格式为对象

})

浏览器传递

http://127.0.0.1:8080/detail/

//路由传参

const express=require('express')

var server=express()

server.listen(8080)

server.get('/datail/:lid',function(req,res){

//接收路由传递的数据

console.log(req.params)

res.send('商品详情')

})

练习创建购物车的路由，请求的方法get,请求的url:/shopping,传递商品的名称（pname）和价格(price)

const express=require('express')

var server=express()

server.listen(8080)

server.get('/shopping/:pname/:price',function(req,res){

console.log(req.params)

res.send('购物')

})

商品模块

商品列表/list 删除/delete 修改/update

用户模块

用户列表/list 删除/delete 修改/update

4、路由器

路由的使用过程中，不同的模块可能出现相同的url,把同一个模块下的所有路由放到一个容器，这个容器就是路由器

路由器最终要引入到web服务器下才能使用

创建路由器模块

const express=require('express')

var router=express.Router();//创建路由器对象

router.get('/list',function(req,res){})//添加路由

module.exports=router;导出路由

const productRouter=require('./product.js')

server.use(./product',productRouter)

//把路由器挂载到/product下

访问形式/product/list

练习创建用户模块路由器（user.js）,添加路由用户列表（/list,get）,在web服务器下引入并使用，挂载到/user下

const express=require('express')

var router=express.Router()

router.get('/list',function(req,res){

res.send('这是用户')

})

module.exports=router

//引入模块

const express=require('express')

//引入路由器(自定义模块)

const userRouter=require('./user.js')

//创建web服务器

var server=express()

//设置端口

server.listen(8080)

//使用商品路由器

//挂载到/product下

///product/list

//参数1：挂载的位置

//参数2：要使用的路由器

server.use('/user',userRouter)

课后练习：

创建一个web服务器，添加路由（get /page ），响应一个html文件；输入身份证号，点击提交（post，/mypage），响应出生的年月日以及性别

server.listen(8080)

/\*server.get('/paeg',function(req,res){

res.sendFile(\_\_dirname+'/paeg.html')

})\*/

server.post('/paeg',function(req,res){

req.on('data',function(buf){

var obj=buf.toString()

// 响应的对象格式化为对象

var str=querystring.parse(obj)

res.send(`

${str1.substr(6,4)}年

${str1.substr(10,2)}月

${str1.substr(10,2)}日

性别${str1.substr(-2,1)%2==0?'女':'男'}`)

/\*var str1=str.uname

var year=str1.substr(6,4)

var moth=str1.substr(10,2)

var date=str1.substr(12,2)

var sex=str1.substr(-2,1)%2==0?'女':'男'

//响应

res.send(year+'年'+moth+'月'+date+'日'+sex)

console.log(str2)

})

//res.send()\*/

})

})

<form method='post'action="paeg" >

用户:<input type="text" name='uname'><br>

<input type="submit">

</form>