# day04

复习

fs

mkdir/rmdir/readdir/create/writeFile/appendFile/readFile/unlink/existsSync/createReadStream/createWriteStream

http协议

通用头信息 请求方法，请求URL，响应状态码

请求头信息

响应头信息 响应文件类型，压缩类型 gzip

请求主体

http模块

http.get(url,function(res){

res.statusCode;

res.on(‘data’,function(chunk){ });

});

http.createServer();

listen(8080);

app.on(‘request’,function(req,res){

req.method/url/headers

res.writeHead 设置响应的头信息

res.write 设置相应的内容

res.end 结束信息

})

express

## 1.使用zlib模块对服务器端响应压缩

|  |
| --- |
| //1.引入zlib模块  const zlib=require(‘zlib’);  //2.设置内容压缩类型  ‘Content-Encoding’:’gzip’  //3.创建压缩  var gzip=zlib.createGzip()  //4.将读取的文件流放入到压缩，发送到浏览器  fs.createReadStream(‘demo.html’).pipe(gzip).pipe(res) |

## 2.express

（1）路由

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来作出响应。

三要素：请求的方法、请求的URL、响应

res 响应对象

send() 响应一段内容

redirect() 设置响应的重定向，也就是跳转

sendFile() 响应文件，要求使用绝对路径 \_\_dirname

注意：在路由中响应只能使用一次

req 请求对象

method 获取请求的方法

url 获取请求的URL

headers 获取请求的头信息

query 获取get请求的数据（查询字符串），返回对象

练习：创建web服务器，添加多个路由(/login 响应’这是登录页面’；/study 响应跳转到[http://www.tmooc.cn;/reg 响应一个文件reg.html](http://www.tmooc.cn;/reg%20响应一个文件reg.html))

（2）路由传递数据

设置路由中接收的名称

|  |
| --- |
| app.get(‘/detail/:lid’,function(req,res){  req.params //获取传递的数据，格式为对象  }); |

浏览器传递数据

http://127.0.0.1:8080/detail/27

练习：创建购物车路由，请求方法:get,请求<url:/shopping>,传递商品的价格price和名称pname,接收传递的数据，并将数据响应到浏览器中

商品模块

列表 详情 修改 删除

/list /detail /update /delete

用户模块

列表 详情 修改 删除

/list /detail /update /delete

## 3.路由器

路由在使用的过程，不同的模块下的路由可能出现相同的URL，将不同模块下的路由区分开来，把同一模块下的路由放到同一个路由器。

创建路由器

|  |
| --- |
| const express=require(‘express’);  var router=express.Router();//创建空的路由器对象  router.get(‘/list’,function(req,res){ });//添加路由  module.exports=router;//导出路由器对象 |

Web服务器使用路由器

|  |
| --- |
| const productRouter=require(‘./product.js’);  app.use(‘/product’,productRouter);//把使用路由器，并在路由的url中添加前缀，访问形式/product/list |

练习：创建用户路由器user.js,添加用户列表，导出路由器；在web服务器下引入并使用，添加前缀/user

## 4.中间件

中间件为主要的业务逻辑服务

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

也称为自定义中间件，是一个函数，需要配合路由或者其他中间件一起使用。

app.user(‘/reg’,function(req,res,next){ })

拦截url为/reg的路由

app.user(function(req,res,next){ })

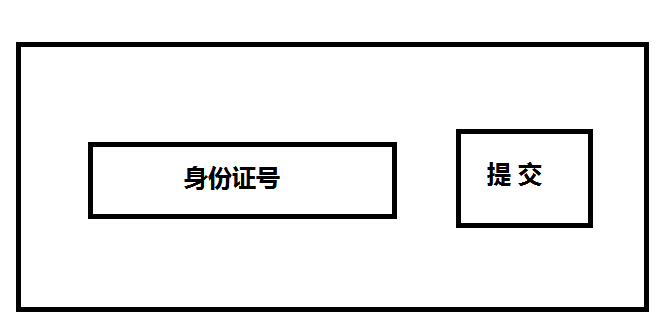
拦截所有的路由

next();执行下一个中间件或者路由。

课后任务：

（1）复习，整理思维导图

（2）练习：创建web服务器，向服务器请求查询生日的html文件,输入身份证号，点击提交，响应生日和性别到浏览器端



（3）复习mysql