common.js 为nodejs使用规范

通过 **module.exports={}===exports.username=’david’—输出一个模块，require加载一个模块；**

服务器运行 node www

Node 新建一个服务，能够访问静态的html文件

需要引入 http,fs,url,util模块

1. 使用http模块创建服务

let server = http.createServer(function(req,res){

let pathname = url.parse(req.url).pathname.substr(1);//解析url

fs.readFile(pathname,(err,data)=>{

if(err) {

throw err

}else{

res.writeHead('200','Content-Type:text/plain')

}

res.end(data.toString())

console.log(data.toString())

})

}).listen('3000','127.0.0.1',()=>{

console.log('服务器已经启动')

})

第三方

http.get('http://nodejs.cn/index.json',function(res){

var dataN = '';

res.on('data',function(r){

console.log(r)

dataN+=r

})

res.on('end',function(){

console.log(res)

// console.log(util.inspect(data))

console.log(dataN)

})

})

安装express生成器

**cnpm i -g express-generator**

生成一个server-demo

express server-demo

设置引擎为html

1安装ejs： cnpm i ejs –save2.app.js ：app.engine(‘.html’,ejs.\_\_express);app.set(‘views engine’,’html’)

Mododb;非关系型

1. 安装需要配置存储位置data，以及logs 以及启动配置etc(mogodb.conf)
2. 需要进入安装目录的bin中，在shell中执行mongod+配置文件，运行
3. 配置好后，直接mongod –config mogodb.conf就能运行
4. 或者直接添加到本地服务中，需要shell获得最高权限（**管理员**运行）mogod –config 配置文件位置 –install –serviceName mongoDB
5. 删除服务 regedit(命令行) HKEY\_LOCAL\_MACHINE/[SYSTEM](https://www.baidu.com/s?wd=SYSTEM&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)/CurrentCountroSet/services/DTLService

不能启动：找到data文件夹db下面的mongod.lock文件，并删除掉。

找到data文件夹db下面的storage.bson文件，并删除掉

1. 此时在bin目录下的mongo运行，然后在show dbs,即可看到数据库
2. 可以加加入环境变量我的电脑-属性-高级-bin目录位置

配置管理员

1. windows 下启动-执行mongod –config 配置文件文职（etc/.conf）

创建表1.use 创建数据库2.db.**createCollection**(‘user’)或者db.users.**insert**({id:123})3.删除数据库db.**dorpDatabase**();4.删除集合db.user.drop();5.查看，db.user.**find**({age:{$gt:30}})/findOne();6.更新db.user.**updata**({name:’jack’},{$set:{name:’david’}/更新子文档{‘class.age’:’30’}}})7.db.user.**remove**({})

导入数据，shell单条导入，工具 mongohub.app;或shell 导入文件mongoimport -d shopdemo -c users --file –文件位置拉入 **必须要退出mongo环境**

Etc--- #dbpath：数据存放目录。

dbpath=c:\mongodb\data

#logpath：指定日志文件，该文件将保存所有的日志记录、诊断信息。除非另有指定，mongod将所有的日志信息输出到标准输出。如果没有指定logappend，重启则日志会进行覆盖操作。

**logpath**=c:\mongodb\logs\mongodb.log

#logappend：写日志的模式：设置为true为追加。默认是覆盖。如果未指定此设置，启动时MongoDB的将覆盖现有的日志文件。

logappend=true

#过滤无用的log信息

quiet = false

port=27017

#httpinterface = false

storageEngine=mmapv1

启动node调试

node www /terminal/webstrom 配置 run->debug configurations-name(api)-working directory(项目空间)-JavaScript file(js执行文件 www)

app.js 里边为一级路由 app.use(‘/’,index)/app.use(‘/user’,user);

pm2

安装 cnpm I pm2 -g 版本 pm2 -v;日志 pm2 logs;启动pm2 start www;重启所有，pm2 reload all;pm2 list

Mongoose(mongodb封装) 安装 cnpm I mongoose –save

查询接口实现

1. 需要models;创建model导出model;基mongoose框架；let p=new mongoose.**Schema**({})

Module.exports=mongoose.**Model**(‘Good’,p)

1. 操作数据库；基于express 与express.**Router**() 和mongoose与**model**（good.find()）

Mongoose.connect(‘mongodb:账号：密码@地址’， {useNewUrlParser:true})

Mongoose.connection.on(‘connected/error/disconnected’，function(){})

1. Router.get(‘/-一级路由’,function(req,res,err){})

解除端口占用 1.netstat -ano|findstr 8080

1. taskkill /f /t /im 19644(为pid)
2. 无法杀死进程https://www.jianshu.com/p/6410b4c0499b

分页后端：1.操作Model;find({}),sort({})

2.Model.find({}).skip(number).limit(number).sort({‘price’:1})

1. model.exec(function(err,doc){})

分页前端：1.（）=>{this}箭头函数没有自己的this，它的this是继承而来，默认指向在定义它时所处的对象(宿主对象)。

2. this.sort ? 1 : -1; vue-infinite-scroll--save:npmjs 加载插件

Express运行：入口 www (createServer)+listen,app:use(‘/goods’,router(如goods))-goods处理，返回

价格排序，Model.find({});参数必须为obj;{}||{salePrice:{$gt:1000,$lte:2000}

Loading--<img src=’.svg’v-show=’’/>可以去github

加入购物车：  
router.post(‘/addcart’,function(req,res,next){  
 Users.findOne({},function(err,docUser){  
 Goods.findOne({},function(err,docGood){  
 **docUser**.listCart.push(docGood)  
 docUser.save(function(err,doc){})

})

})  
})

总结；1，查用户2.查模型，3.去重插入用户collection

登录依旧需要连接数据库

1. findOne查数据库，设置cookie;可以分别设置两个res.cookie('userId',docUser.userId,{  
    path:'/',  
    maxAge:1000\*60\*60  
   })
2. 后端可以截断请求，返回状态码；app.use(function(){})从cookie中获取看是否登录，设置白名单；req.originalUrl或者req.path
3. 检查登录，判断cookie是否存在并返回

删除购物车中某一条

Users.**update**(**{userId:userId},{$pull:{'cartList':{'productId':productId}}}**,function (err,doc) {})

添加购物车数量

Users.update(**{userId:userId,'cartList.productId':productId }**,{  
 '**cartList.$.productNum':productNum  
}，function(){}**

Modal 模态框组件一旦确定就必须把数据弄好props:[];this.$emit(‘praent事件’)

新建模型时，更新插入数据时，必须严格按照模型；比如Array

获取参数

#==>location.hash获取的为#后边

？==>location.search 获取的为？后边的

NVM去管理node版本