1. 配置

阿里云

Use admin //使用admin权限用户

db.createUser(

  {

    user: "oneadmin",

    pwd: "tyu879!KT1213",

    roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]

  }

)

db.auth('oneadmin','tyu879!KT1213')

1. 笔记  
   1.查看位置

2.开启mongod

sudo service mongodb start  
 或者配置path不用sudo   
 <http://www.cnblogs.com/pingfan1990/p/4988525.html>3.项目和数据库的通信  
 通过mongoose

var mongoose = require('mongoose')  
4.基本命令  
 show dbs 显示所有数据库

Switched to db xxxx 切换到xxx数据库  
 db.users.find({}) 通过存空信息查询所有信息

5.populate填充

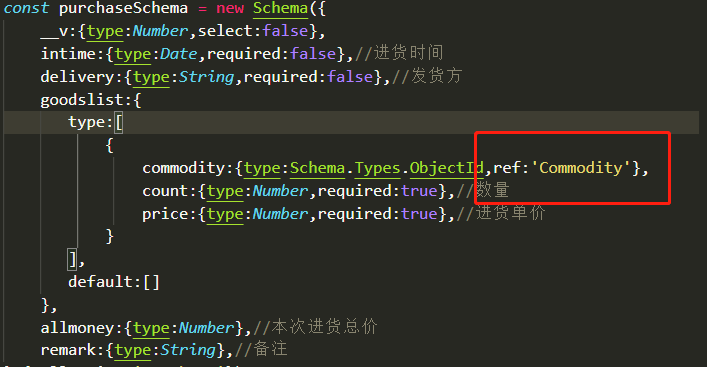
使用姿势

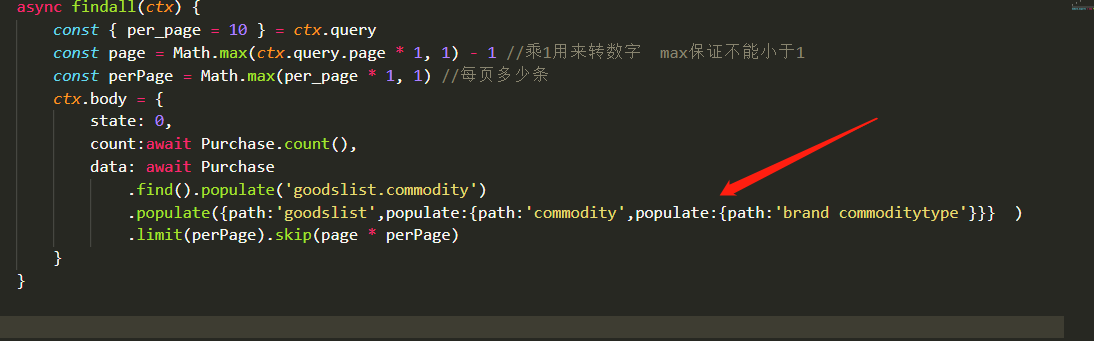


多级填充

<https://blog.csdn.net/letterTiger/article/details/84710937>

通过嵌套对象实现多级填充





6.日期

Mongodb默认用UTC 0区时间,本地时间要转成时间戳加8小时再传入

if(*ctx*.request.body.ordertime){

*ctx*.request.body.ordertime = new *Date*(*Date*.parse(*ctx*.request.body.ordertime)).getTime() + 8 \* 60 \* 60 \* 1000

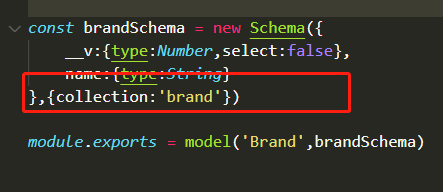
        }else{

*ctx*.request.body.ordertime = new *Date*().getTime() + 8 \* 60 \* 60 \* 1000

        }

7.mongoose

Mongoose默认会给模型名变复数,要在第二参数声明下



8.卸载

$ sudo service mongod stop

$ sudo apt-get purge mongodb-org\*

$ sudo apt-get autoremove

<https://blog.csdn.net/Cryhelyxx/article/details/79234430>

9.切换数据库

比如切换到admin数据库: use admin

10.权限

Mongodb管理员和数据库使用员是两码事,

数据库的内部角色才能使用数据库,比如增删改查建表,数据库的管理员角色管理的对象是内部角色成员.

比如dbAdmin角色可以管理readWrite角色,但是dbadmin没有数据库的读写增删改查建表权限,readwrite可以

11.关联表查询

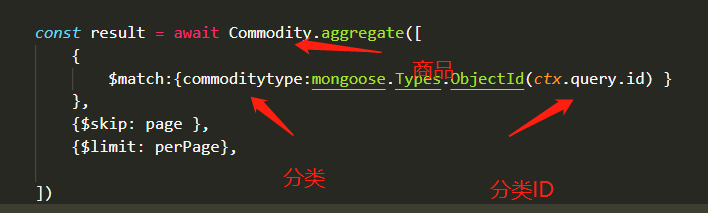
子表查主表用polulate填充字段，主表查子表先用match过滤，然后用lookup关联表查询。Orm用的是mongoose的时候，如果要过滤\_id要加mongoose.Types.ObjectId



https://www.cnblogs.com/huangxincheng/p/5728791.html

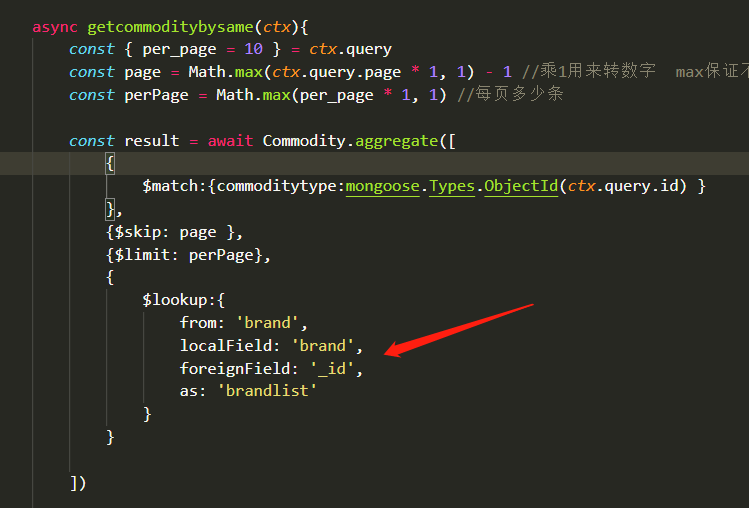
12.主表筛选子表

分类是商品的外键，需要：查一个分类下所有的商品



如果想对查询结果做populate，需要用lookup，

比如查询结果默认显示的品牌ID，需要显示品牌名称



完整带总数的例子

 async getcommoditybysame(*ctx*){

*const* { per\_page = 10 } = *ctx*.query

*const* page = Math.max(*ctx*.query.page \* 1, 1) - 1 //乘1用来转数字  max保证不能小于1

*const* perPage = Math.max(per\_page \* 1, 1) //每页多少条

*const* result = await Commodity.aggregate([

            {

                $match:{commoditytype:mongoose.Types.ObjectId(*ctx*.query.id) }

            },

            {$skip: page },

            {$limit: perPage},

            {

                $lookup:{

                    from: 'brand',

                    localField: 'brand',

                    foreignField: '\_id',

                    as: 'brandlist'

                }

            }

        ])

*const* count\_result = await Commodity.aggregate([

            {

                $match:{commoditytype:mongoose.Types.ObjectId(*ctx*.query.id) }

            },

            {

                $group:{

                    \_id:"$commoditytype",

                    count:{$sum:1}

                }

            }

        ])

        if(!result){

*ctx*.throw(404,'分类不存在')

        }

*ctx*.body = {

            state:0,

            count:count\_result,

            data:result

        }

    }