山 东 大 学

非关系数据库系统报告

**姓 名： 汪逢生**

**专 业： 软件工程**

**年 级： 研一**

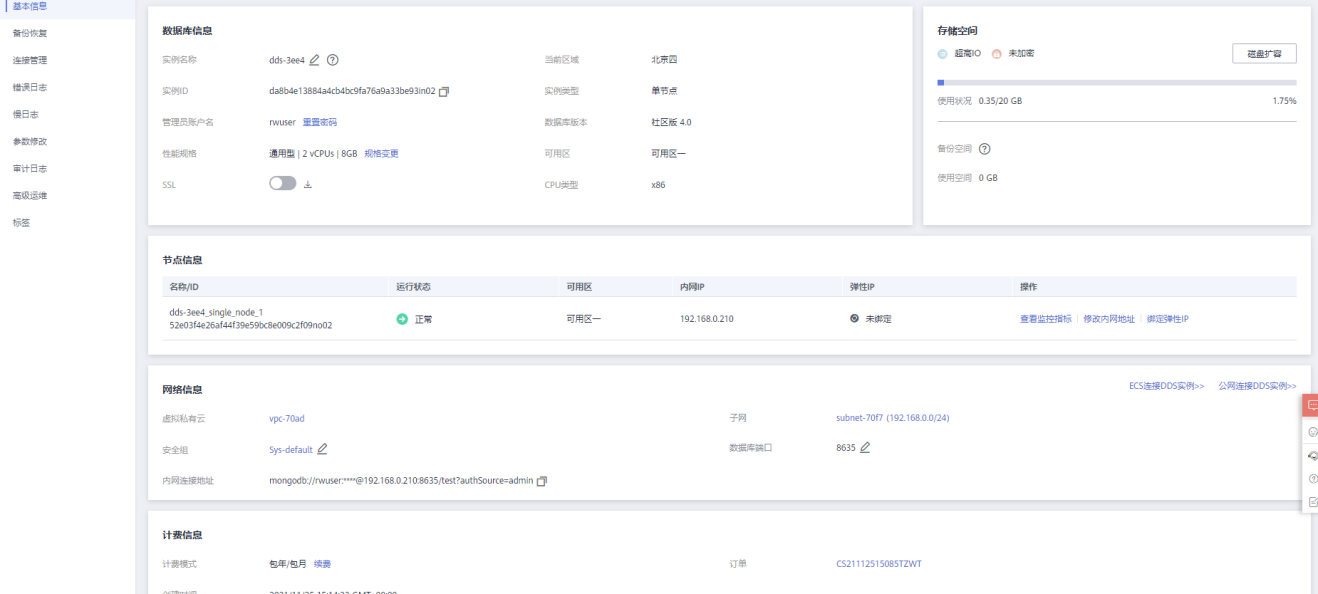
**学 号： 20215215**

**学 院： 软件学院**

**指导老师： 李晖**

1. 熟悉实验环境和实验数据
2. 熟悉华为RDS for MongoDB数据库的操作。

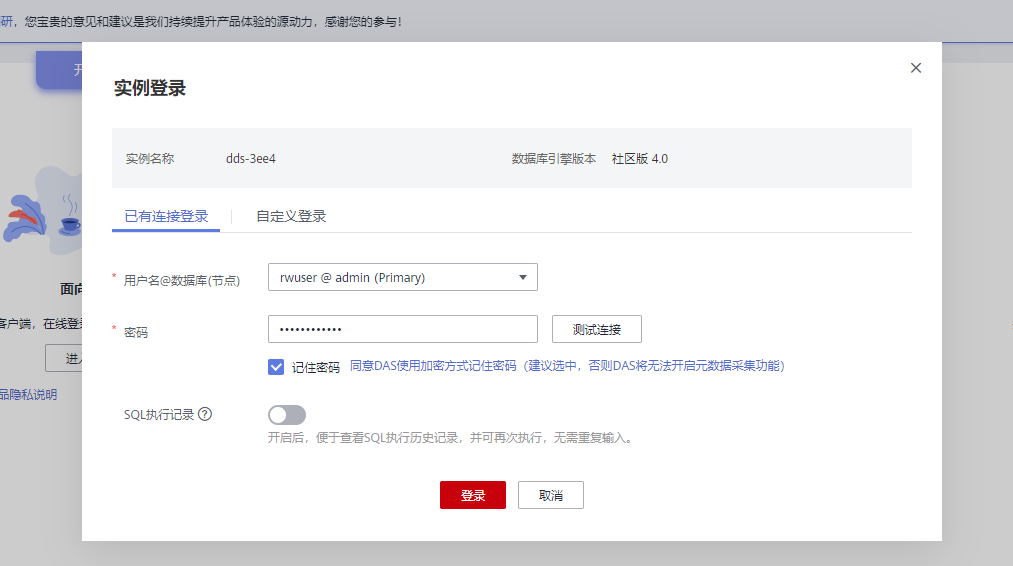
打开主页面负责数据库连接



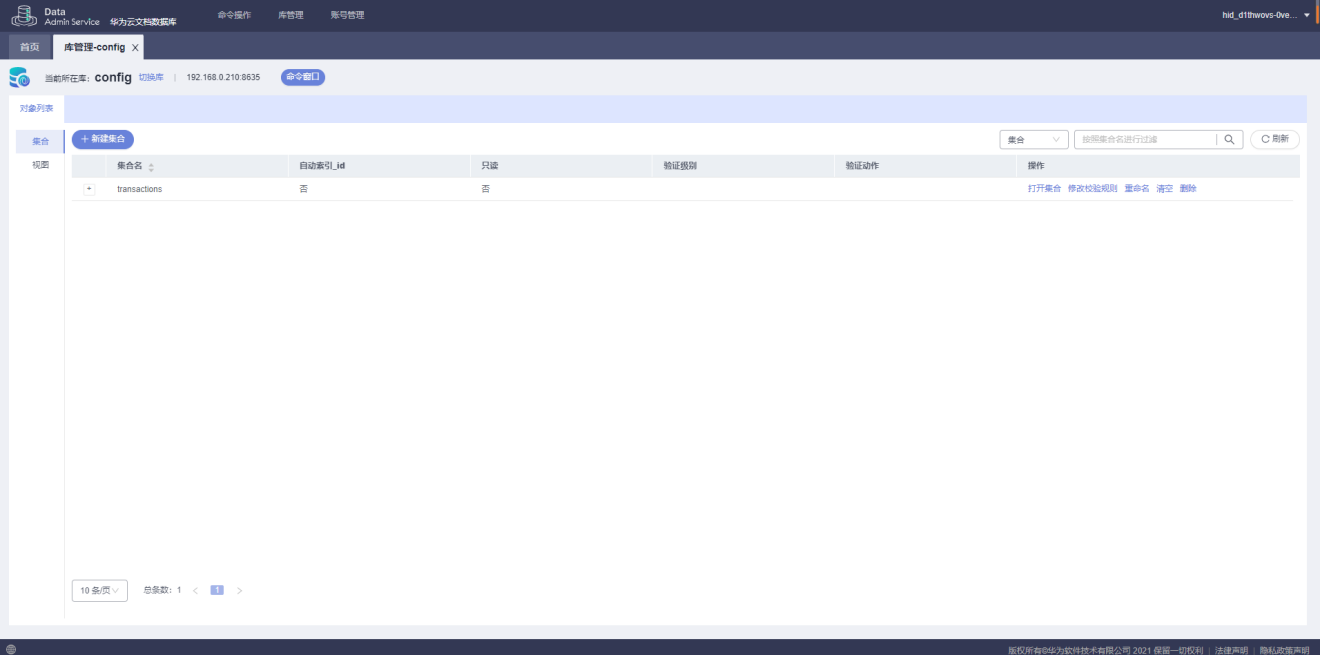
在“实例管理”页面，选择目标实例，单击操作列的“登录”，进入数据管理服务数据库登录界面



使用DAS连接单节点



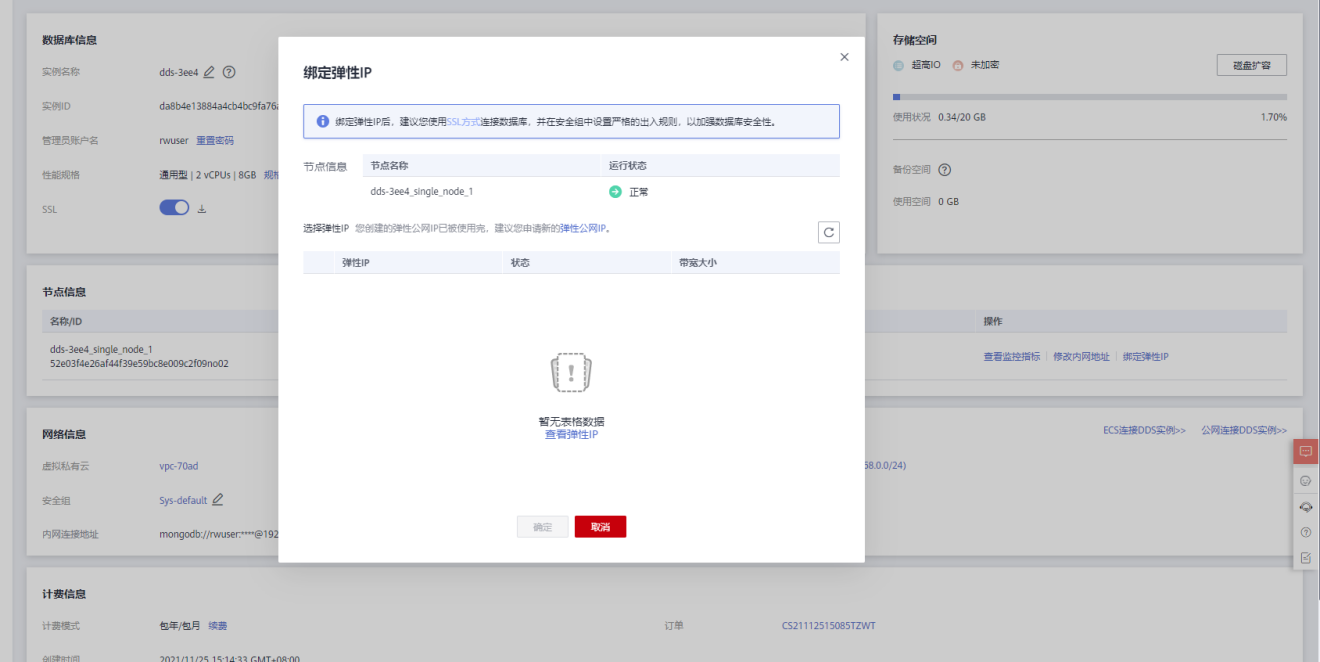
正确输入管理员帐户名和密码，单击“登录”，即可进入数据库并进行管理。



--------------------------------------------------------------------------------------------------------------------------------------

通过弹性IP连接DDS实例

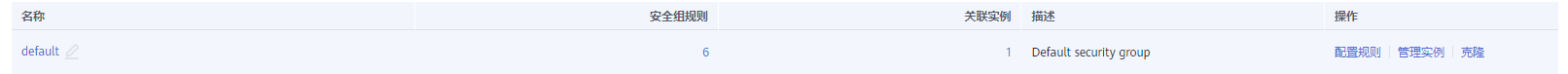
绑定弹性IP





点击安全组





在“网络信息”模块，单击安全组名称，进入安全组页面。



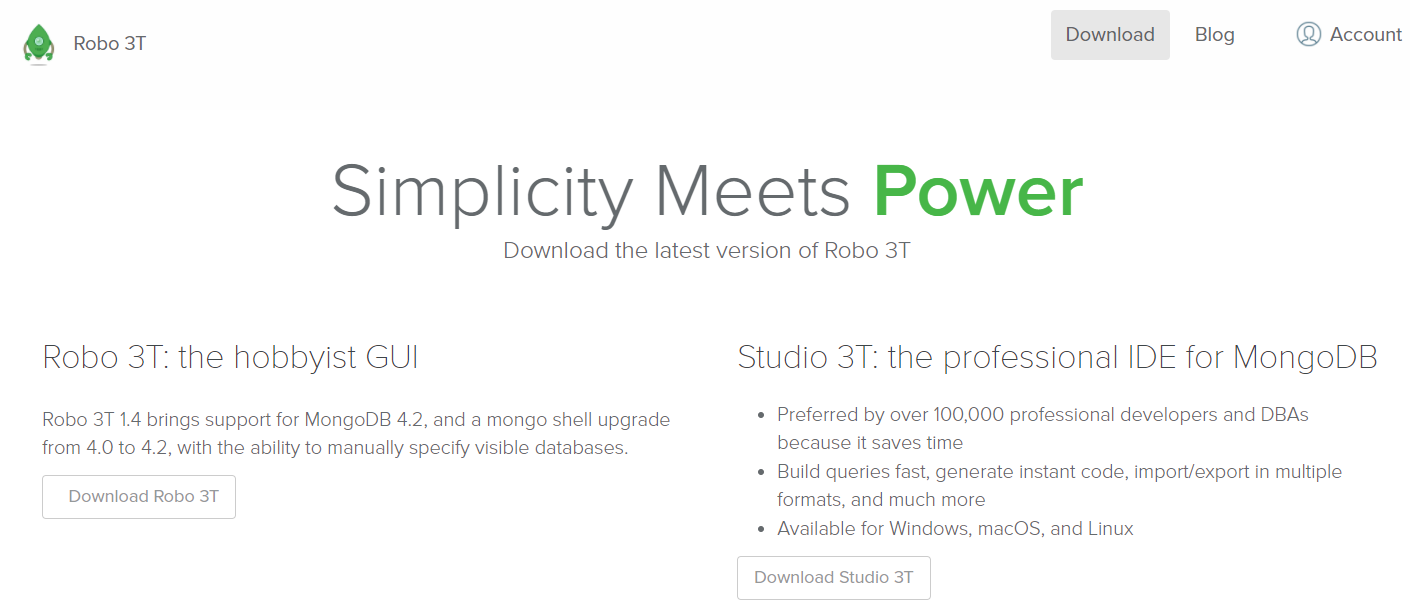
选择“入方向规则”页签，单击“添加规则”，在弹出窗口中，添加端口为数据库端口的入方向规则。

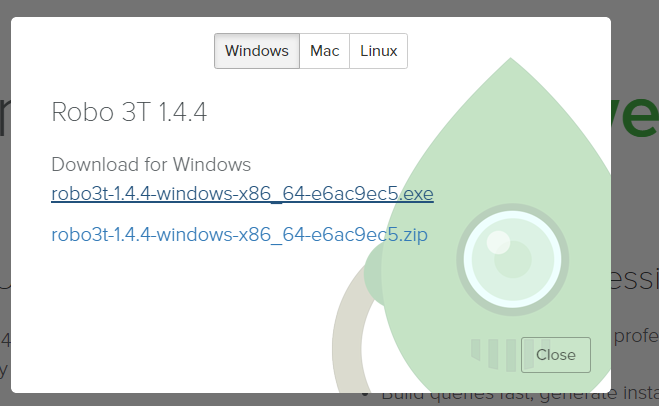
--------------------------------------------------------------------------------------------------------------------------------------

连接DDS实例

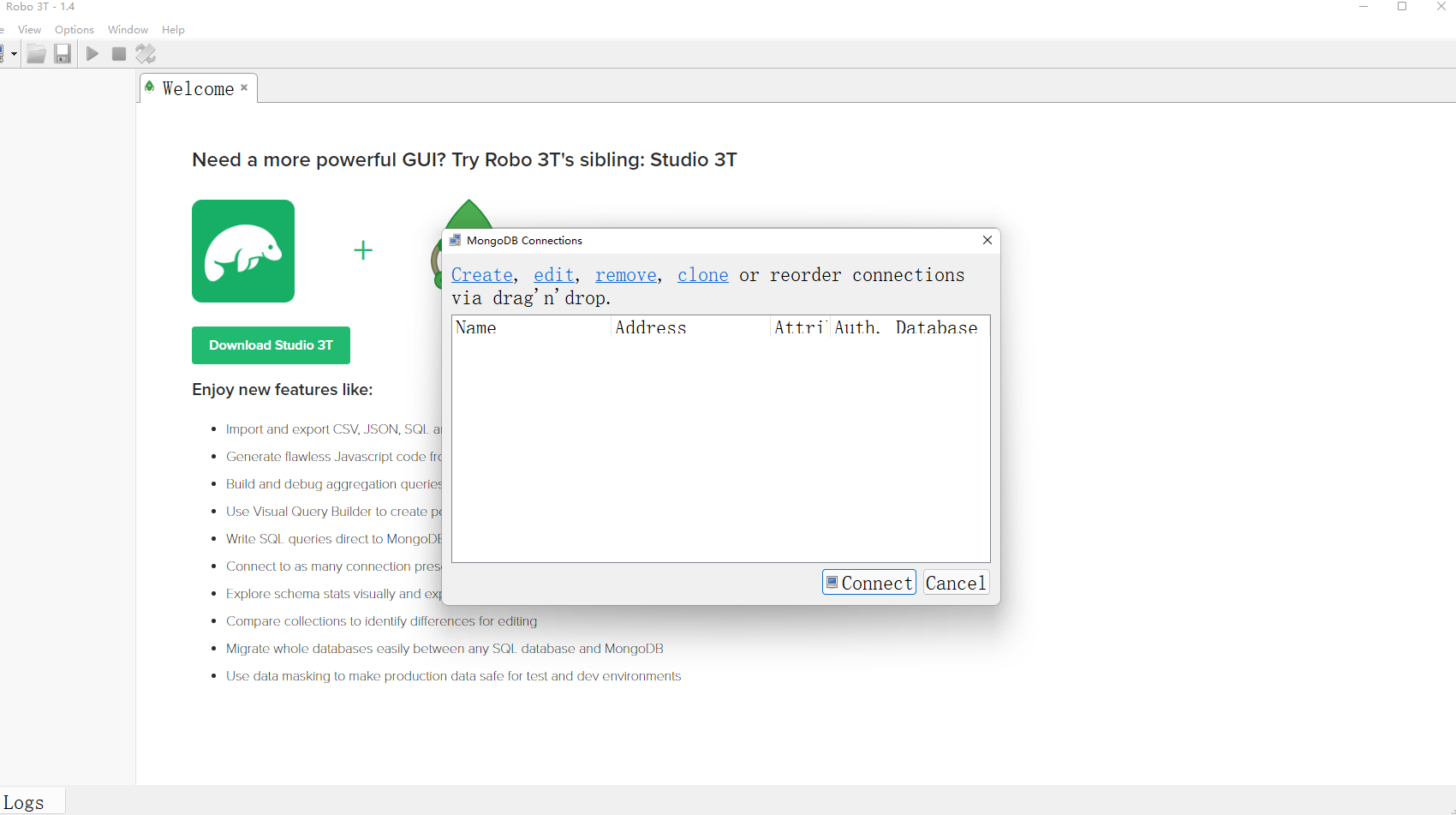
Robo 3T下载



在弹出框中，下载“robo3t-1.3.1-windows-x86\_64-7419c406.exe”。

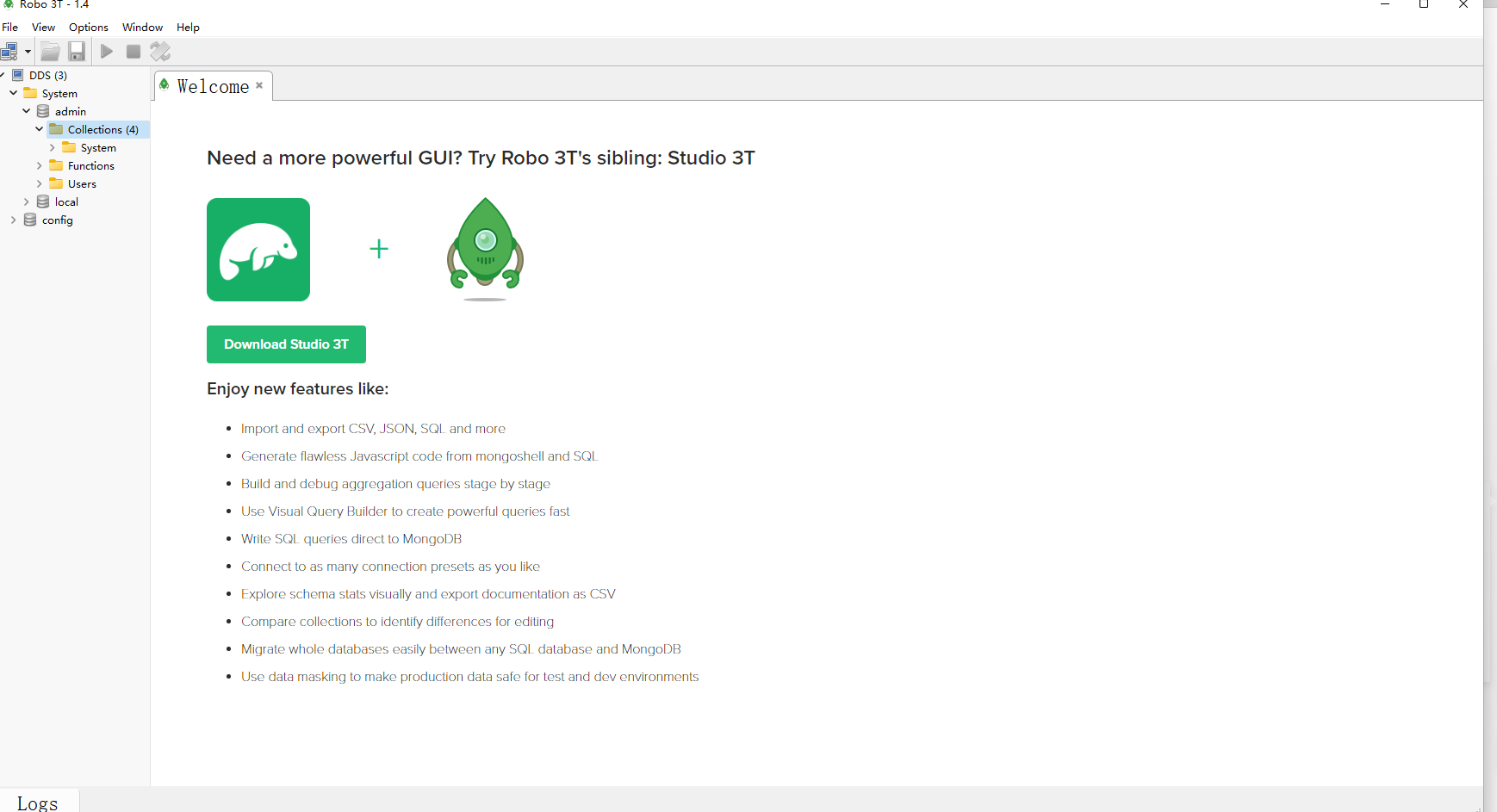


安装完成后，在弹出的“Connection Settings”窗口，设置新建连接的参数

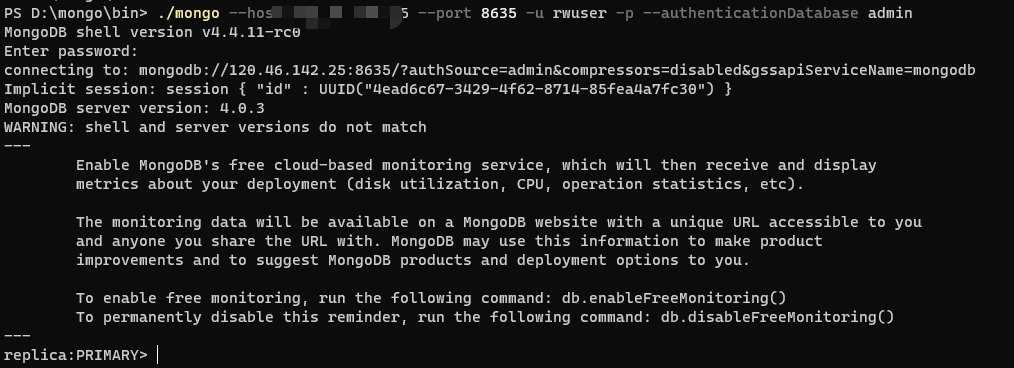




成功连接DDS实例，工具界面显示如下：

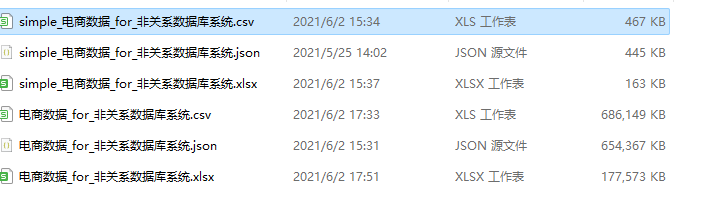


下载mongoshell连接数据库

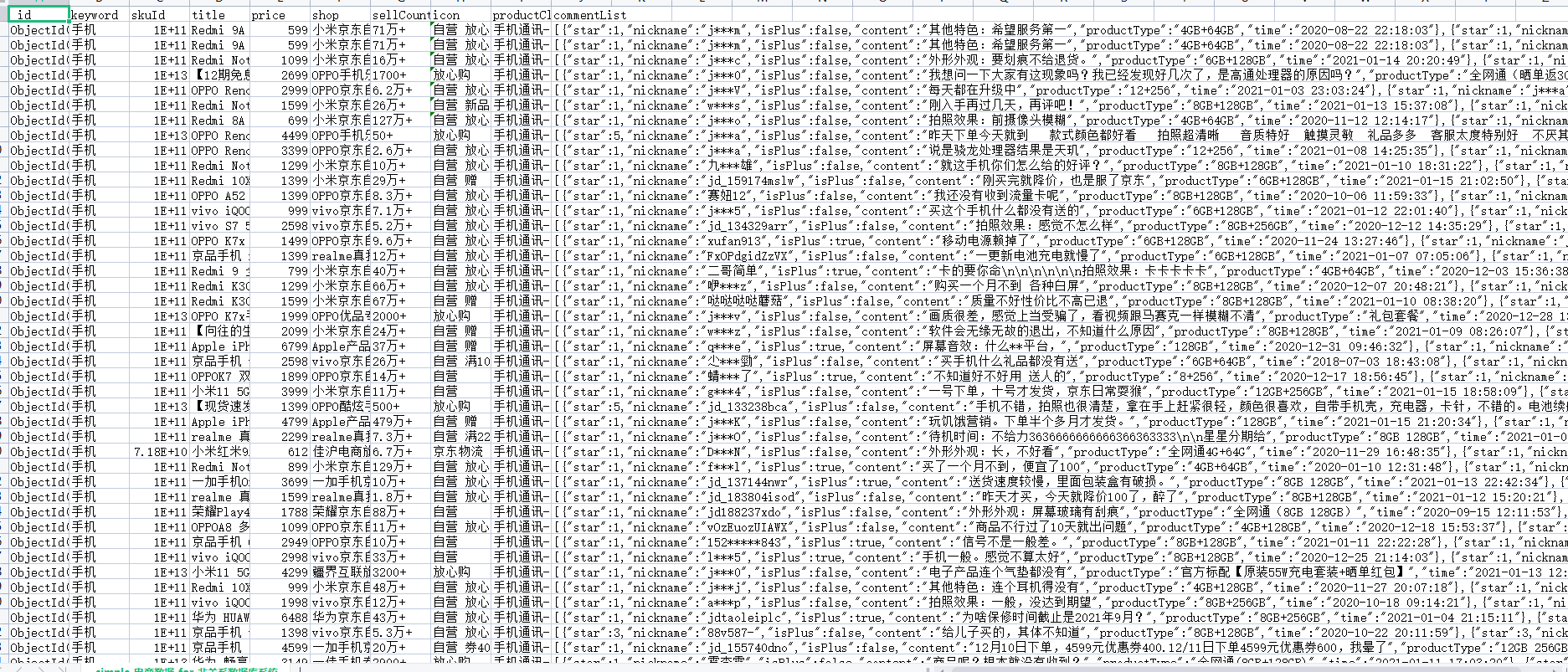


2、下载并了解实验数据。

下载数据集



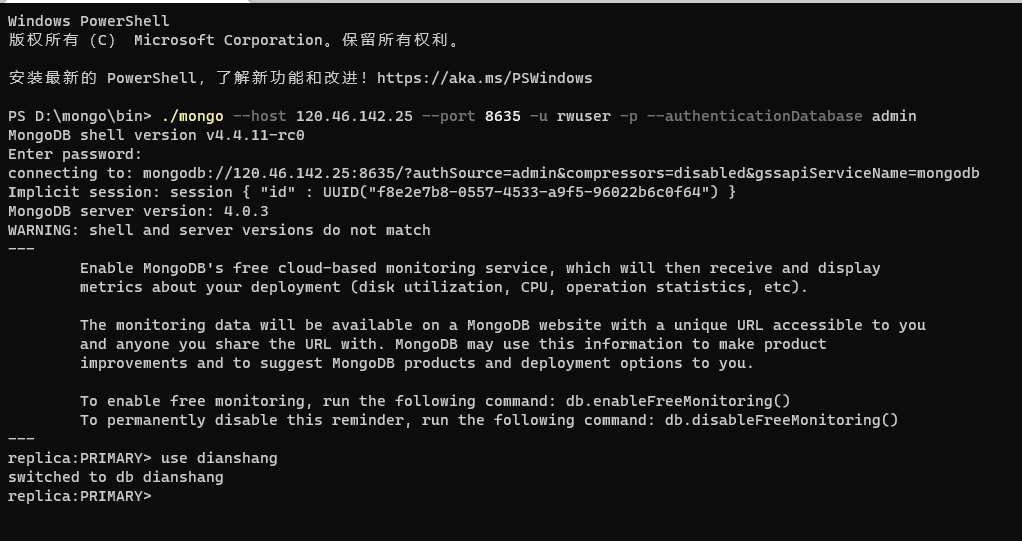
打开NoSQL实验电商数据可以看到数据集总共有10个属性



其中某一商品productclass 中手机通讯-手机-手机-小米中手机通讯为一级分类，手机为二级分类其中可以 一级分类应该要提出来，二级分类也要提出来，因此这两个要有，商品购买量要有，店铺名称要有，first class，scond class, price, shop,sellcount,commenList，

### 实验二 设计MongoDB数据库、集合和文档

1. 在华为DDS云文档数据库中创建数据库。



远程连接华为云文档数据库创建数据库名字为电商

2. 根据实验数据库的特点，设计集合模式。集合和文档要求能够科学地组织和存储数据、高效方便地获取和维护数据。

设计集合模式

db.dianshang.commodity.insert({

'keyword': '手机',

'skuId': '100014348492',

'title': 'Redmi 9A 5000mAh大电量 大屏幕大字体大音量 1300万AI相机 八核处理器 人脸解锁 4GB+64GB 砂石黑 游戏智能手机 小米 红米',

'price': 599,

'shop': '小米京东自营旗舰店',

'icon': '自营 放心购 赠 ',

'first\_class': '手机通讯',

'second\_class': '手机',

'commentList':

[

{"star":1,"nickname":"j\*\*\*m","isPlus": False,"content":"其他特色：希望服务第一","productType":"4GB+64GB","time":"2020-08-22 22:18:03","month": 8, "quarter":2},

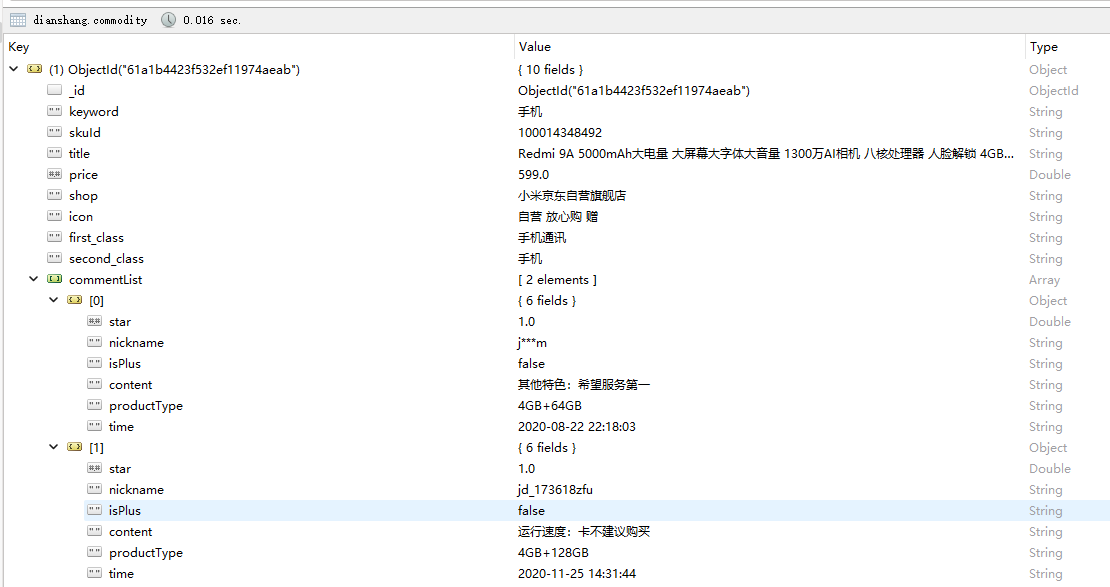
{"star":1,"nickname":"jd\_173618zfu","isPlus":False,"content":"运行速度：卡不建议购买","productType":"4GB+128GB","time":"2020-11-25 14:31:44","month": 8, "quarter":2}

]

})



创建集合模式，直接插入一条数据，为了查询方便这里在原有数据productclass上另取了一级分类和二级分类分别作为一个属性。



可以看到mongodb已经自动为我们生成好了数据对应的类型。

实验三 将实验数据插入数据库中

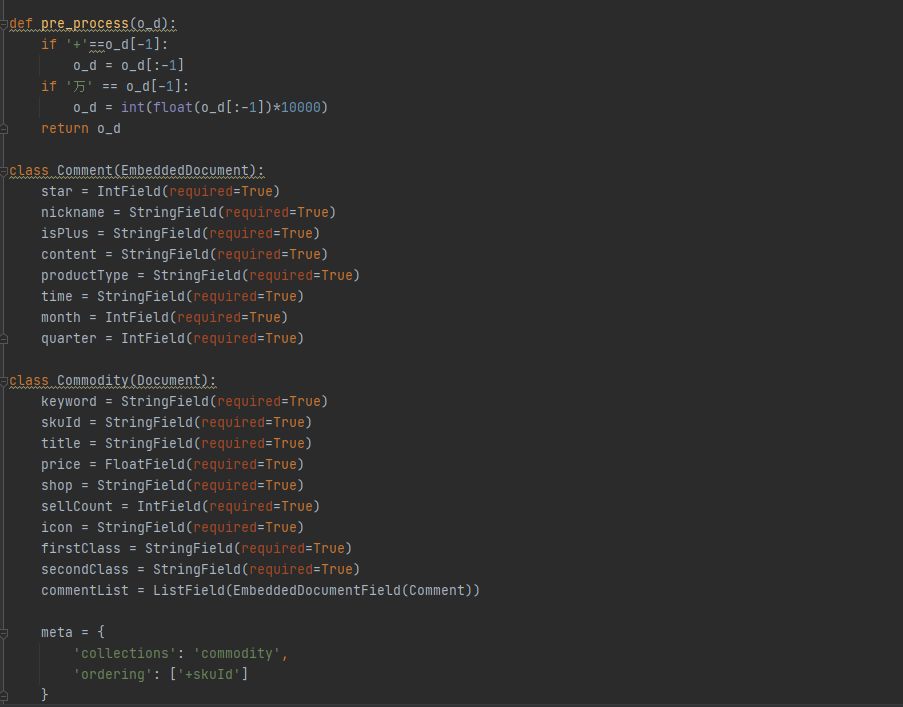
使用高级程序设计语言（语言不限）将实验数据插入实验二设计好的集合中，并按照设计好的文档结构组织数据。

使用对象文档映射ODM 实现对MongoDB数据库增删改查操作, 把数据库里面的记录对象简化操作，每一个文档转化为对象，更便于对数据进行处理，为每个字段设置其对应的类型。

连接华为云mongbdb数据库

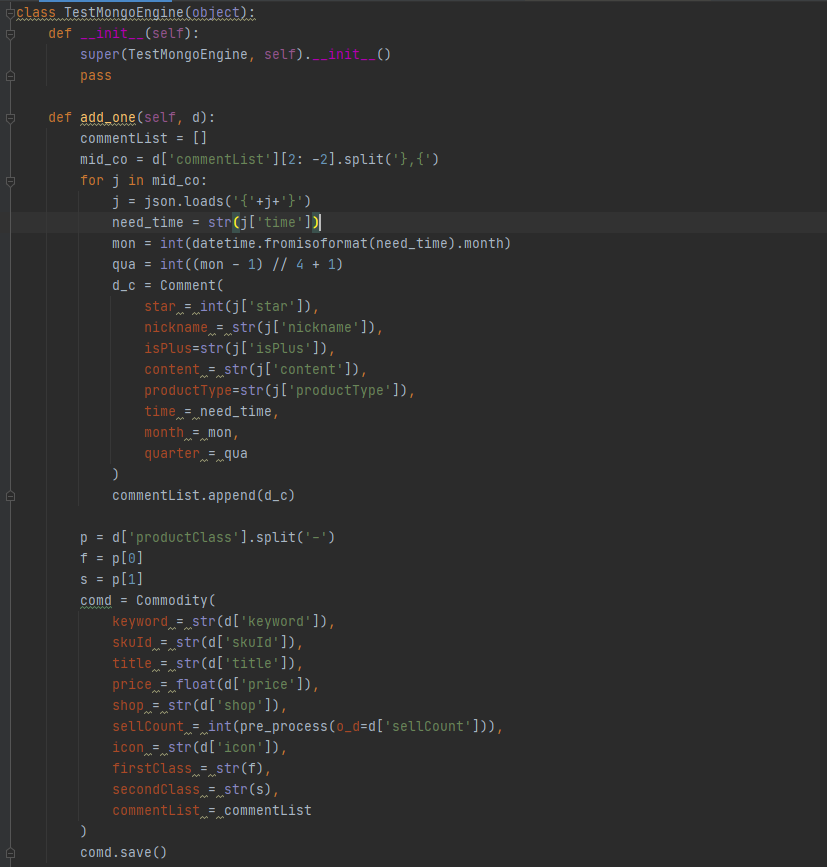


创建Comment和Commodity两个对象 分别表示评论和商品，实现文档和文档嵌套。



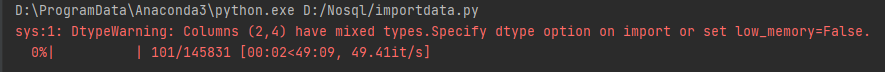
接下来连接mongodb并插入数据

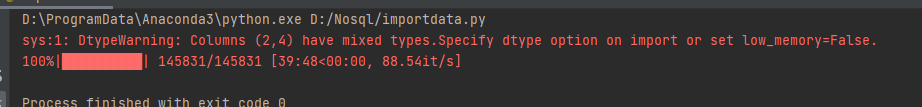
首先对数据进行预处理将其处理为数据直接插入到数据库中去



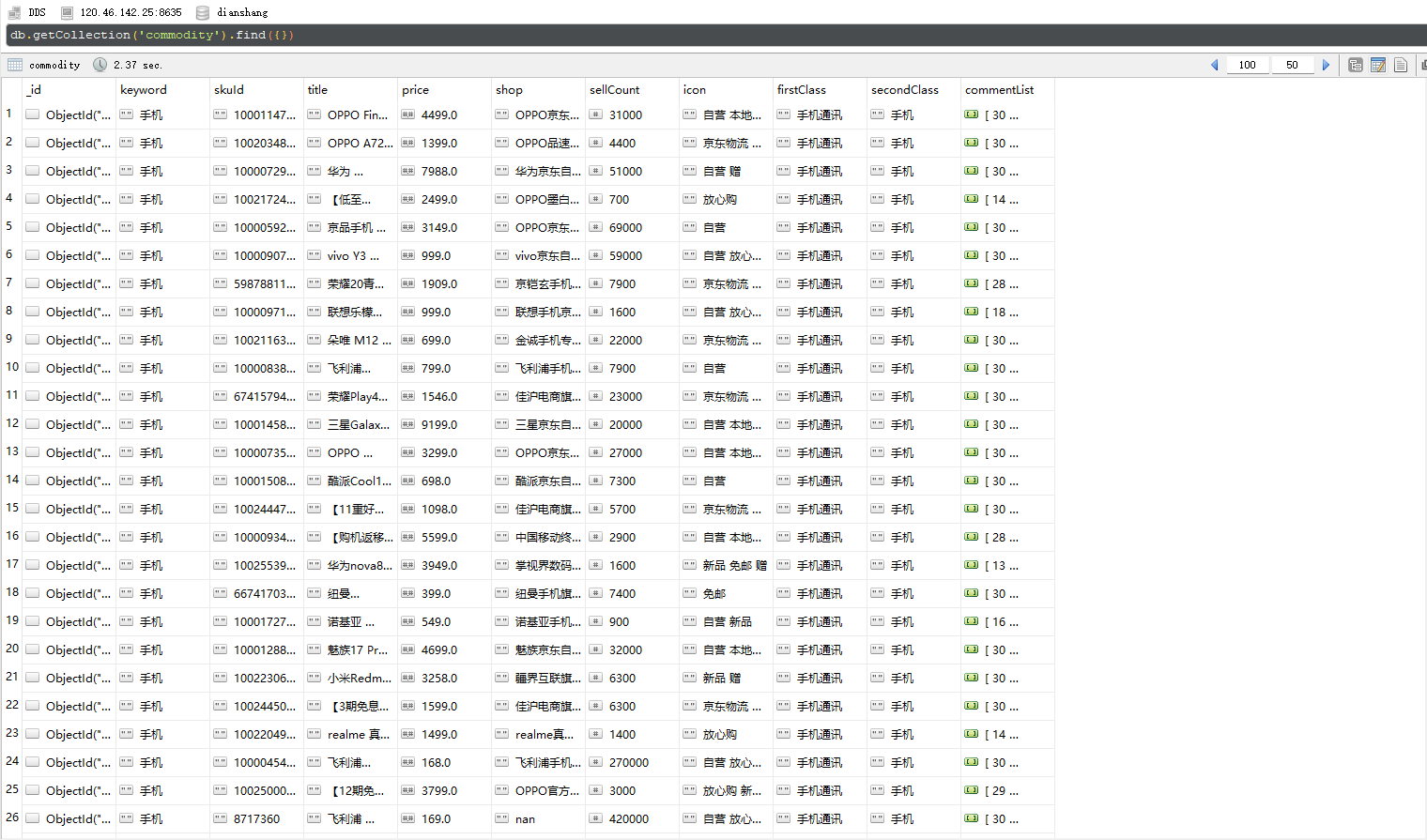


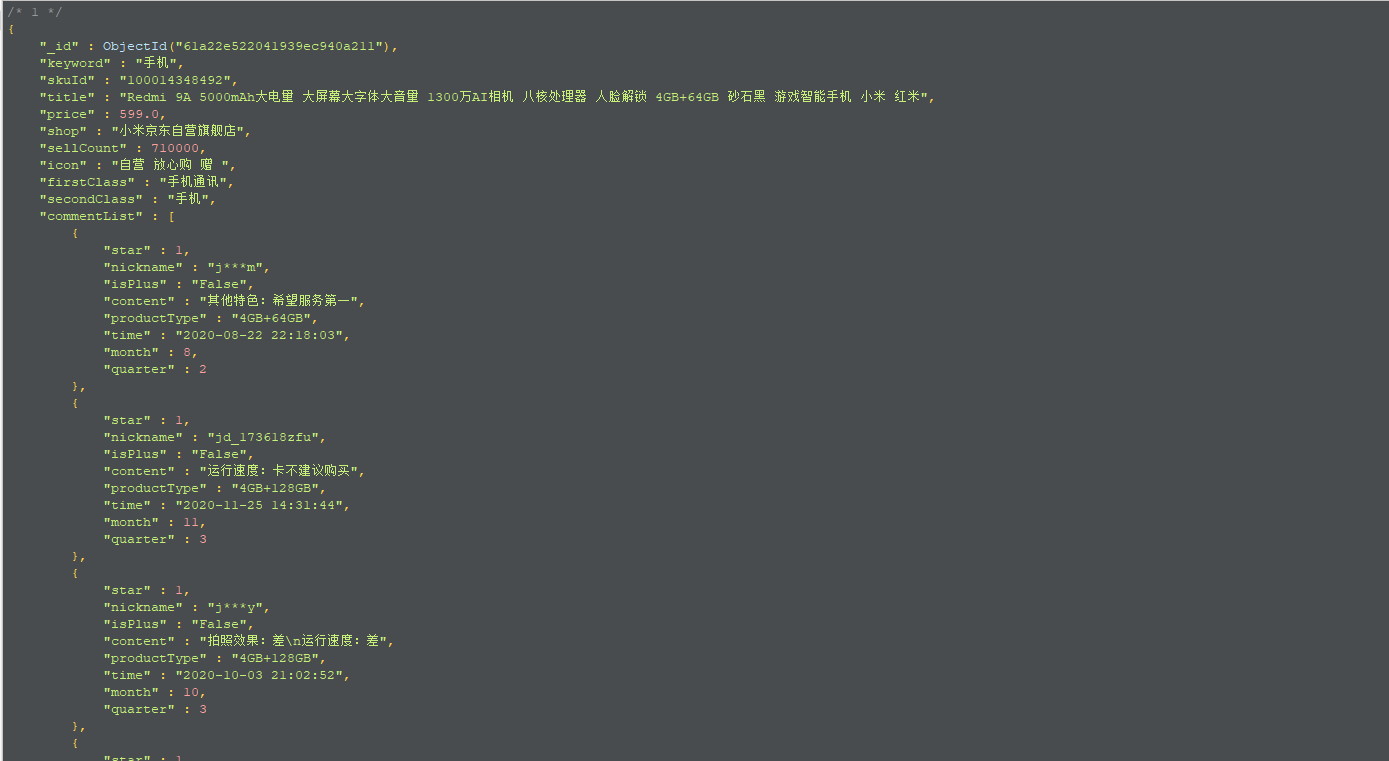
运行完毕插入数据 如果数据包含空值直接抛弃





查看插入的数据





实验四 使用高级程序设计语言进行数据统计

自选高级程序设计语言，自行设计界面，完成以下数据统计并将统计信息进行展示。

1、展示某一商品一级分类下各个二级分类商品的均价、中位数、最高价、最低价；首先将所有的一级分类展示，用户可以选择一级分类，根据用户选择的一级分类，展示该一级分类下面二级分类的价格信息；

2、商品购买量与季节/月份的关系展示：通过商品评论中的time进行统计；

3、热点店铺展示：展示某一商品一级分类下商品销量前几名的热点店铺。

4、自行设计题目，进行数据统计分析

特别提醒：商品分类的一级分类以及二级分类。实验数据的单条商品数据中，商品分类是类似“手机通讯-手机-手机-小米”格式的字符串，而商品的每一级分类是被短横线分割开的字符串组。以上述商品分类字符串为例，其中“手机通讯”就是商品的一级分类，而“手机”为二级分类。

可以参考的工具：Echarts前端模块用于数据展示；Redis用于缓存(可以不用考虑)。

本次实验使用python语言来处理数据

前端使用vue2，element-ui+vue-echarts组件，后端使用flask

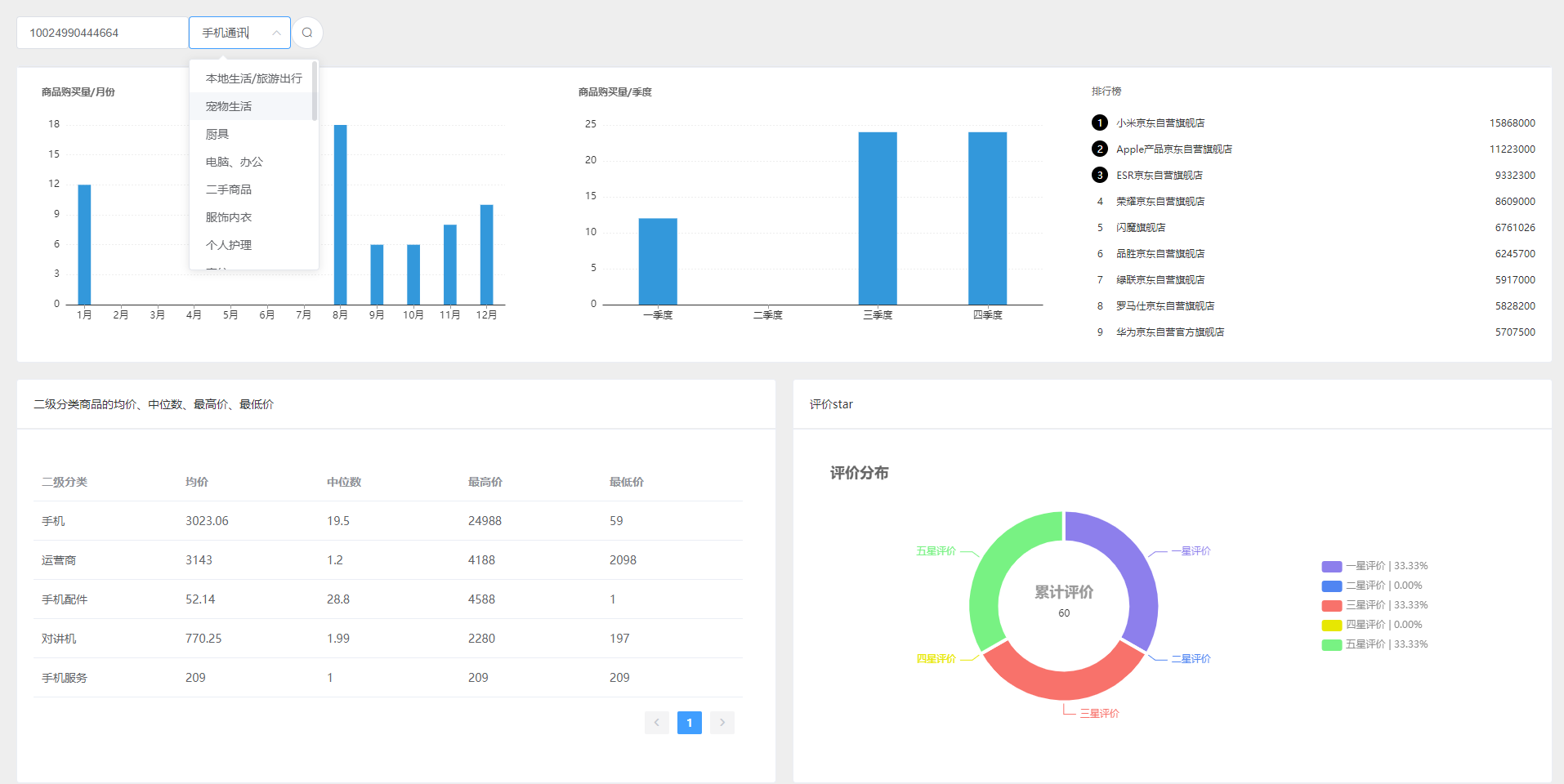
总共实现了对某一物品(skuId)实现了四种信息展示，

1. 某一商品一级分类下各个二级分类商品的均价、中位数、最高价、最低价。

首先将所有的一级分类展示，用户可以选择一级分类，根据用户选择的一级分类，展示该一级分类下面二级分类的价格信息；

1. 商品购买量与月份的关系展示。
2. 通过商品评论中的time进行统计，热点店铺展示。
3. 展示某一商品所有评论的饼图展示。

其中最终完成的页面如下图所示，用户可以在右上角搜索skuid，实现了所有结果的一个展示。

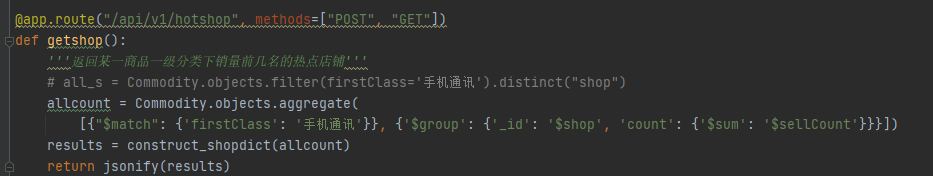


实现方法如下

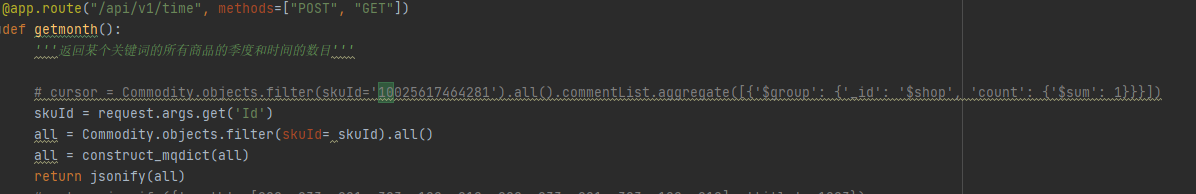
首先是用

一开始实现是以每个需求部分分别提供一个接口实现方法如下

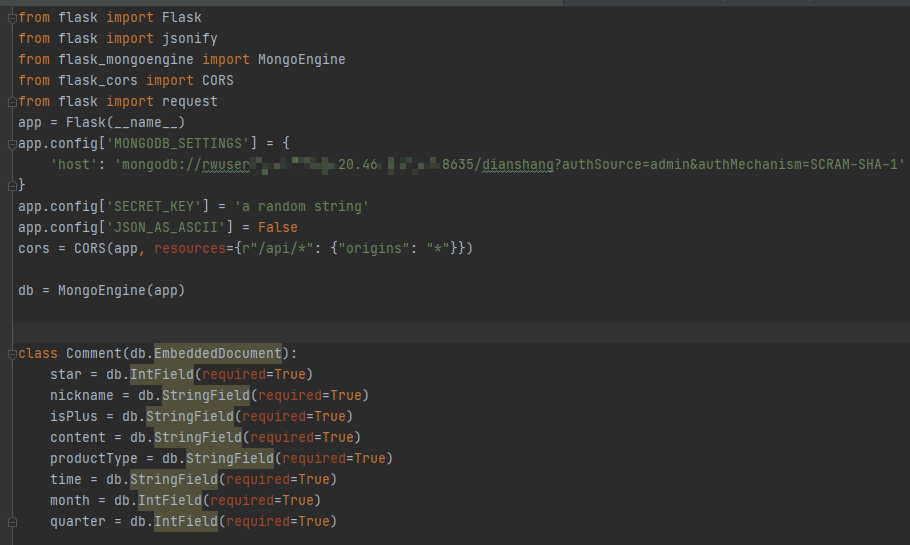
下面是热点商铺的介绍



下面是返回12个月份的购买量



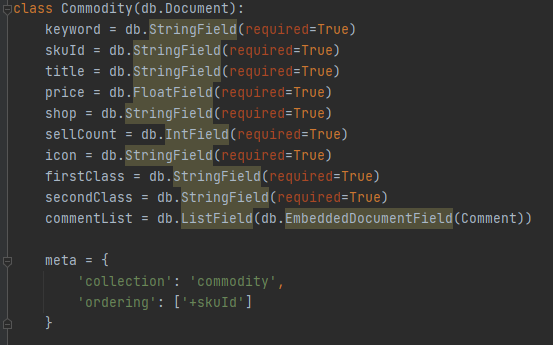
接下来使用flask-mongoengine的封装方便对数据库进行操作。

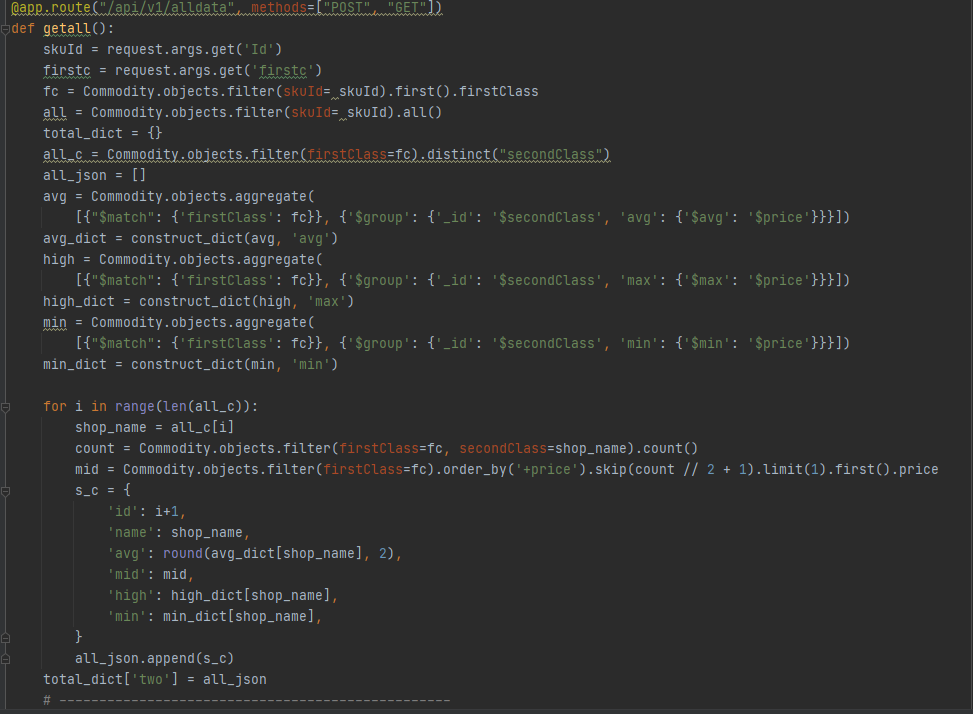


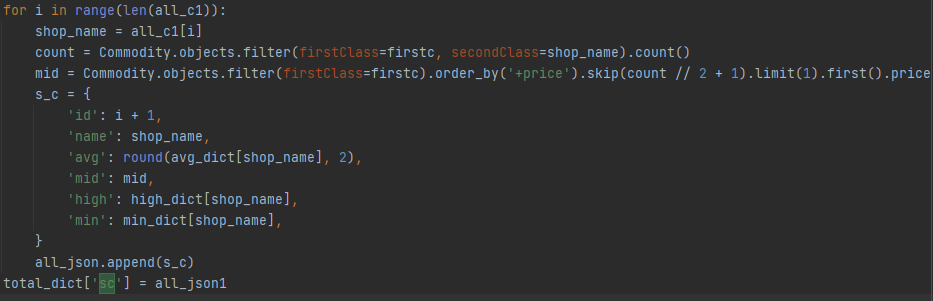


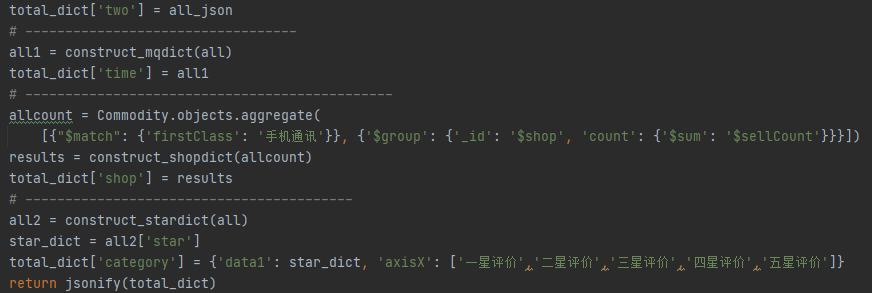
后面发现还不如用一个接口一次性返回所有数据

最终将其整合作为一个接口

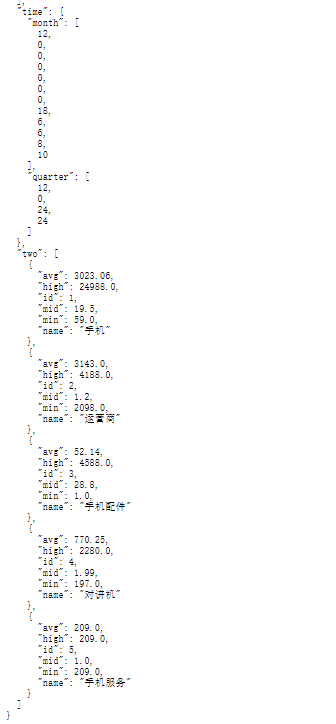
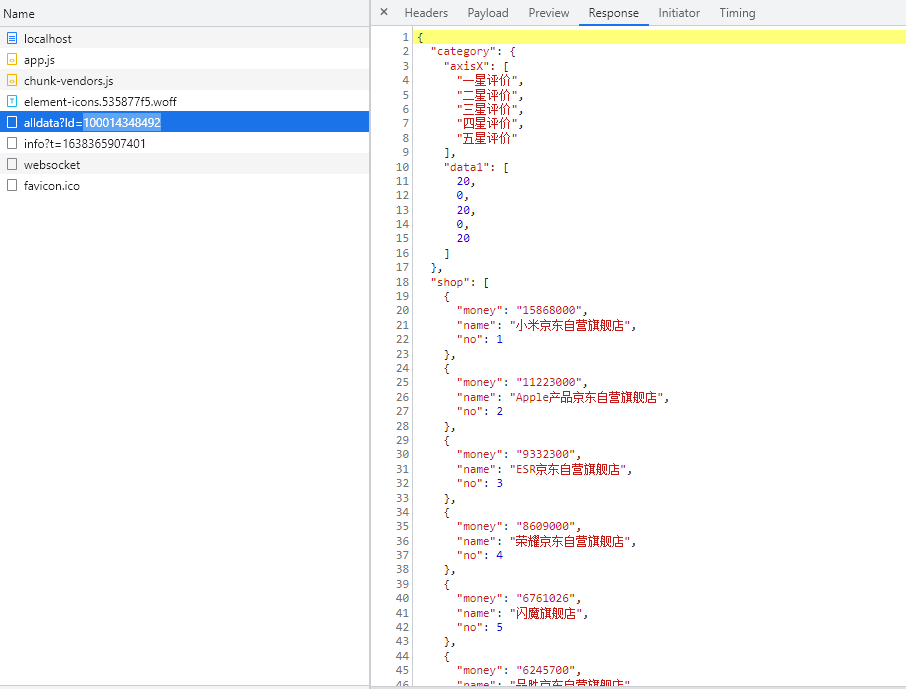




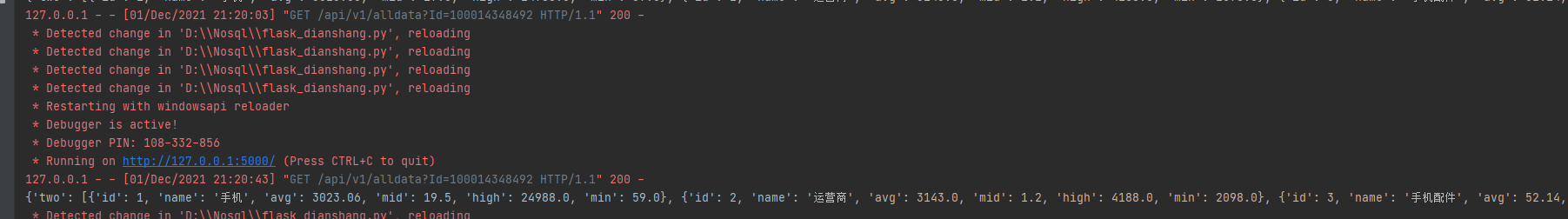


这是主要使用一些基本flask\_mongoengine的MongoEngine的操作方法filter, aggregate等完成了上述操作。

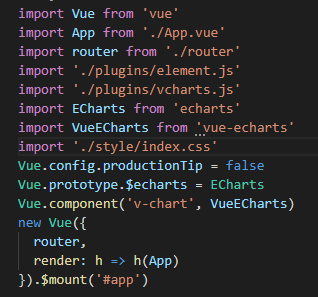
最终返回到前端的json数据



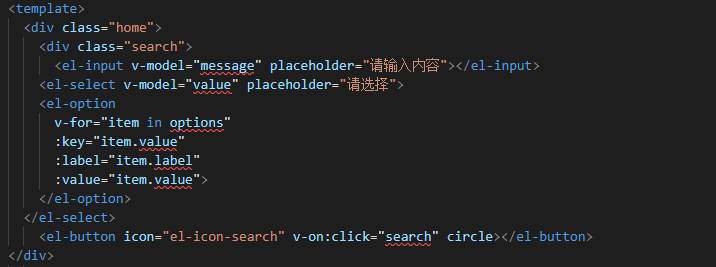
后端返回给前端的数据展示

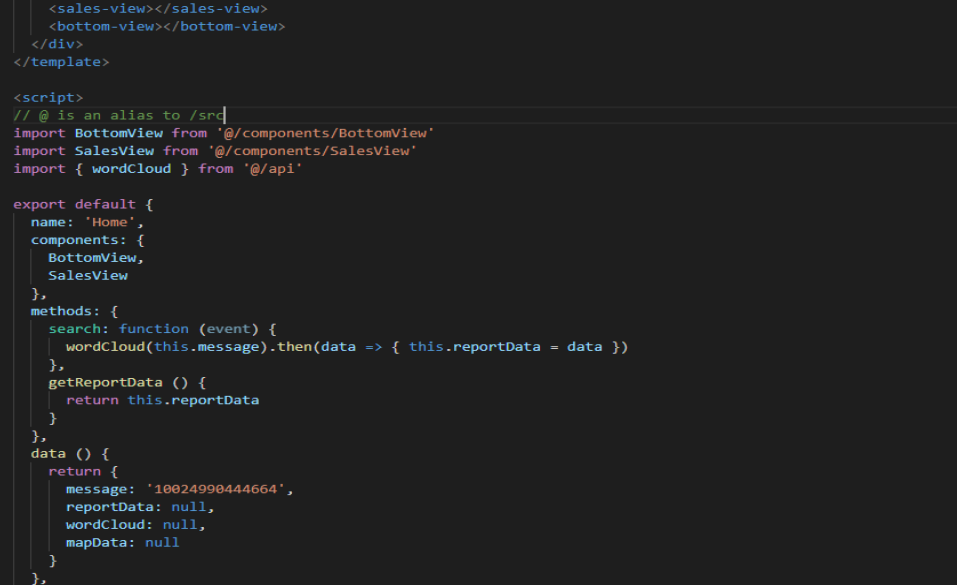


接下来是前端界面的实现



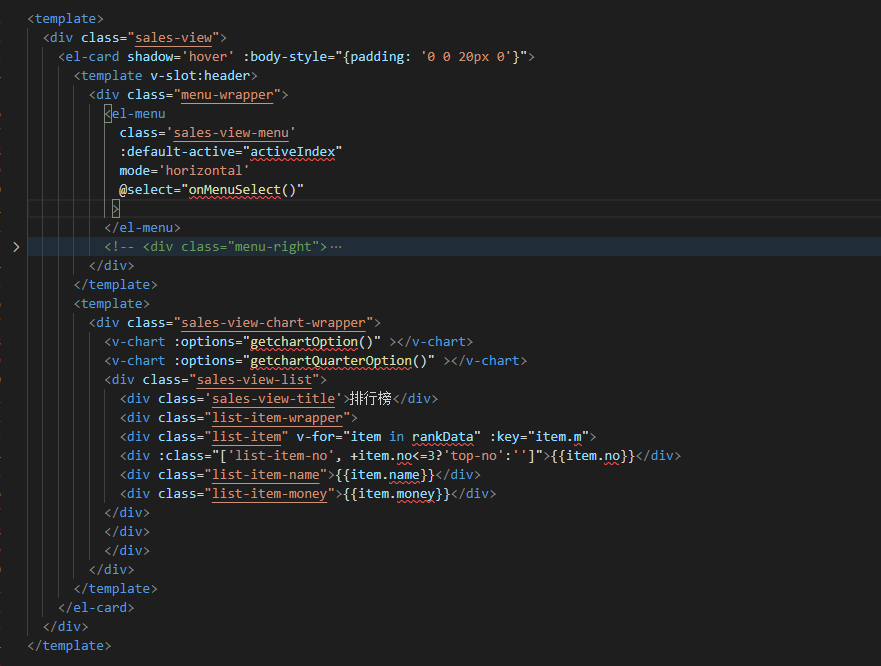
实现最上面的搜索界面



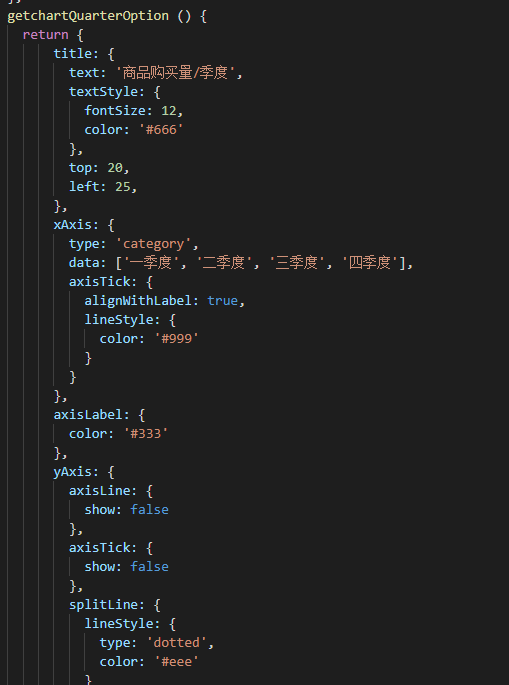
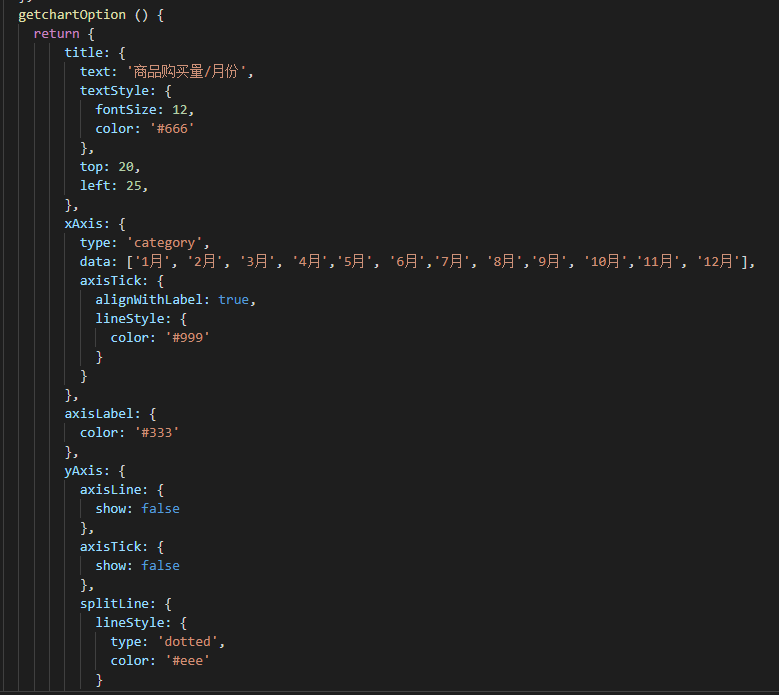


构建三个组件分别包括上面的查询框，默认查询的skuId是10024990444664和一级分类是手机通讯中间的两个柱状图和排行以及下面的表和饼图

中间的柱状图实现方案使用两个函数实现数据的绑定

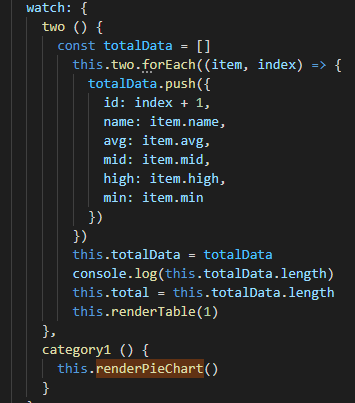


下面是具体的实现包括12个月和4个季度的实现

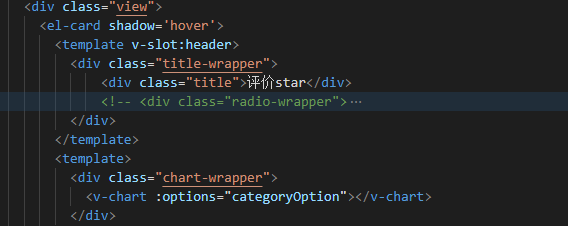


排行榜的实现比较简单后端处理后直接返回到前端展示。接下来是对二级分类商品属性的实现。

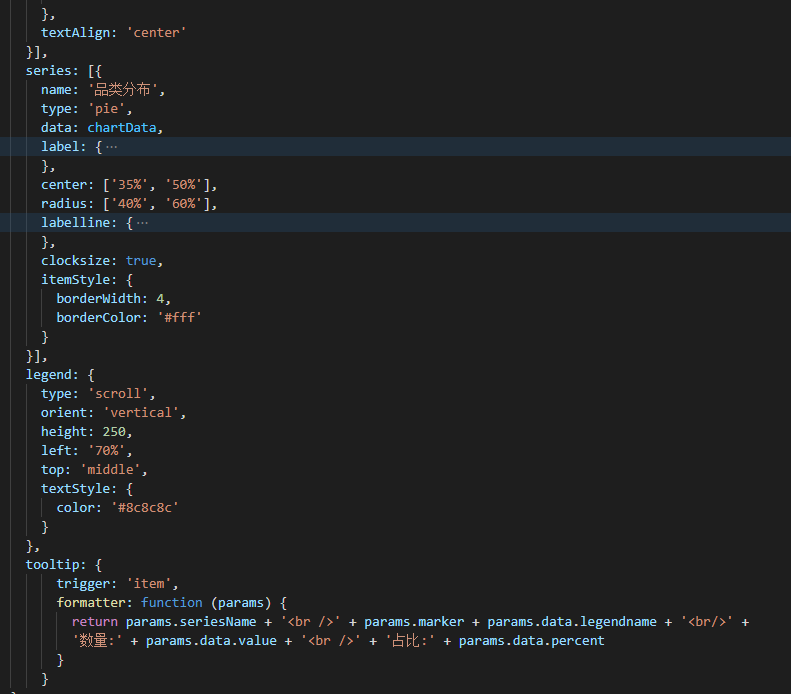




Watch实现标表盘数据的更新 接下来是评价表盘的实现。

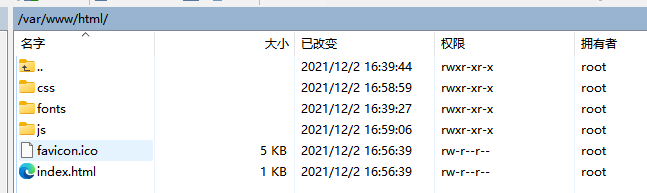






在实现了所有需求后，将其部署到服务器上。

打包放到服务器上



访问inthemetaverse.cool 即可实现对给定数据的查询

