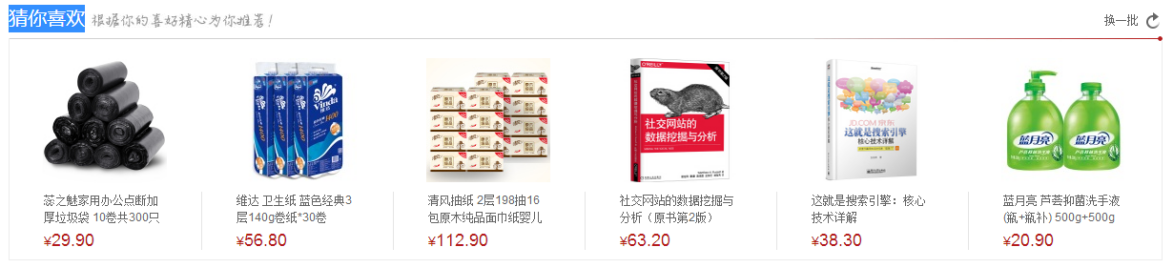
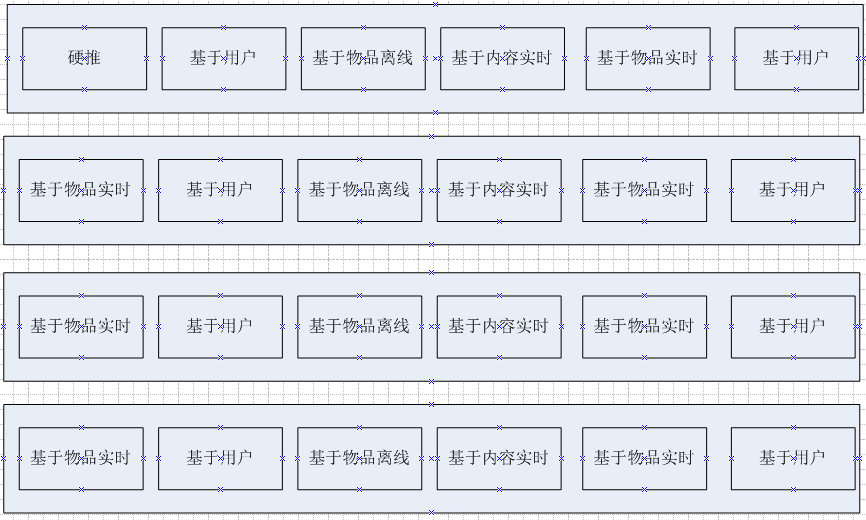
# 1、业务需求

* 某电商网站首页有猜你喜欢推荐位，该推荐位一次能展示6个商品，推荐内容可以更换四次，共需推荐24个商品。



* 需要使用协同过滤算法（user CF & Item CF）及基于物品内容的算法进行混合推荐。
* 一次性展示的6个商品中，从左到右的顺序分别是：
  + 第一位：基于物品的实时推荐结果
  + 第二位：基于用户的离线推荐结果
  + 第三位：基于物品的离线推荐结果
  + 第四位：基于内容的实时推荐结果
  + 第五位：基于物品的实时推荐结果
  + 第六位：基于用户的离线推荐结果
* 如有业务需要推广产品，可以指定推广产品出现在某一个位置上。如下图，在第一位上硬推某产品。



# 2、需求分析

* 大型网站的推荐位不仅仅只有一个，需要对每个广告位进行编号，比如猜你喜欢的广告位编号是121
* 每个推荐位是一个独立的推荐产品，需要对每个广告位开发独立的推荐模型
* 每个推荐位需要配置特有的推荐规则和排序规则
* 为了容错，每个推荐位都需要默认的推荐产品，当推荐系统无法计算正常的结果时，使用默认产品进行推荐。
* 各个推荐模型推送的商品可以能重复和下线的商品，需要对商品进行进行去重和过滤处理
* 推荐结果计算完毕之后，将硬推广告放进去。

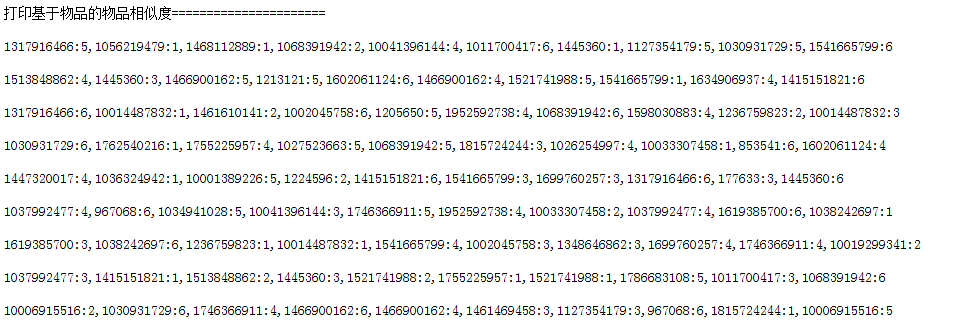
# 3、数据准备

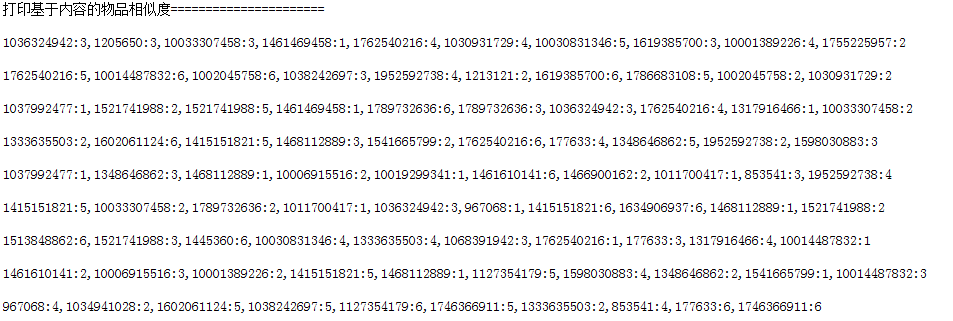
* 为用户guyong准备基于用户的推荐结果

C:\Users\maoxiangyi\AppData\Local\YNote\data\wjymaoxiangyi@163.com\d31cb22cb6f441f5844ff38b1f85cefb\clipboard.png

* 为用户guyong准备基于物品的推荐结果，该结果基于用户上一次的浏览记录生成

C:\Users\maoxiangyi\AppData\Local\YNote\data\wjymaoxiangyi@163.com\9af09f2a721342948e7b9bb25205e5f8\clipboard.png

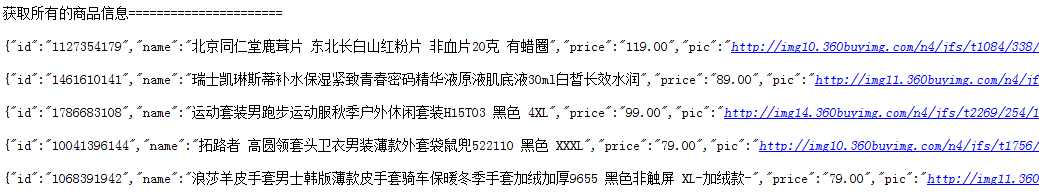
* 准备物品与物品的相似度数据--基于物品的相似度推荐  
  
* 准备物品与物品的相似度数据----基于内容的相似度推荐



* 为广告位（猜你喜欢）121准备默认的推荐结果

C:\Users\maoxiangyi\AppData\Local\YNote\data\wjymaoxiangyi@163.com\68ad0c30350a46619297b14d9e8c70cf\clipboard.png

* 初始化所有的商品信息



# 4、功能实现分析

* 1）用户在商城浏览商品，将用户的浏览记录保存到Cookie，随着用户的请求传送给推荐服务接口。推荐服务接收到用户的基本信息和浏览信息。

另一种思路，可以通过**消费**点击流日志，将用户的行为保存到Redis中，推荐服务通过访问Redis获取用户的行为记录。

* 2）推荐接口从用户的基本信息中获取到三种推荐结果**（离线推荐结果）**
  + **基于历史数据**，计算的基于用户的协同过滤的推荐结果，推荐数量24。
  + **基于用户上一次行为记录**，计算的基于**物品的协同过滤**推荐结果，推荐数量24。这里根据用户对某一个商品的浏览次数进行加权。
  + **基于用户上一次行为记录**，计算的**基于内容的推荐结果**，推荐数量24。这里根据用户对某一个商品的浏览次数进行加权。
* 3）推荐接口从用户的浏览信息中获取用户当前会话的的行为记录，并以此计算基于物品和基于内容的**实时推荐结果**
  + 基于用户本次会话的记录，计算**基于物品的推荐结果**，推荐数量为24
  + 基于用户本次会话的记录，计算**基于内容的推荐结果**，推荐数量为24。
* 4）对以上的反馈的推荐结果进行排序，排序的过程中对商品去重
  + 按照业务需求对结果排序，第一位是基于物品的实时推荐结果，依次类推。在排序的过程汇中，需要对推荐的商品进行排序。
  + 推荐结果生成完毕之后，对整体的推荐结果的产品数量进行补全和删除操作。补全使用该推荐位的默认推荐产品进行补全。
* 5）设置业务人员**强推的商**品，根据业务人员指定的商品序号，替换掉推荐结果中对应序号的推荐商品。

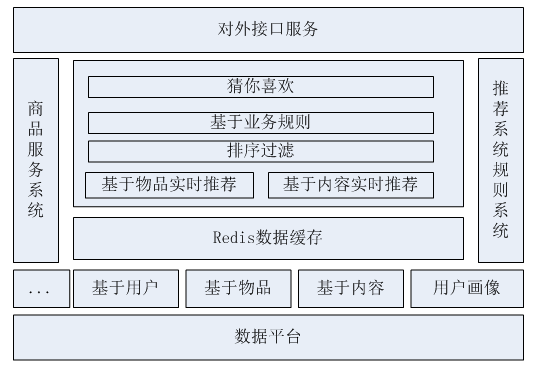
注：整体思路如此，后续推荐模型可以持续进行优化，包括对推荐结果进行加权。

# 5、产品原型设计

产品经理进行原型设计

# 6、项目开发

## 6.1、整体架构



1）**数据平台**：在数据平台上，针对每个用户计算好三个推荐结果，基于**用户的推荐结果、基于物品的推荐结果、基于内容的推荐结果**。基于物品的相似度、基于内容的相似度。**(召回推荐结果)**

2）**Redis数据缓存**：通过独立的Java应用将每个用户的推荐结果和**基于物品的相似度与基于内容的相似度信息**导入到Redis缓存集群中。

3）**获取推荐结果**；推荐结果有两种：一种是已经计算好的离线推荐结果，直接获取即可，另一种是根据**用户实时的浏览记录**计算新的推荐结果。第二种推荐结果主要依赖三种数据，用户的浏览记录、基于物品的相似度、基于内容的相似度。

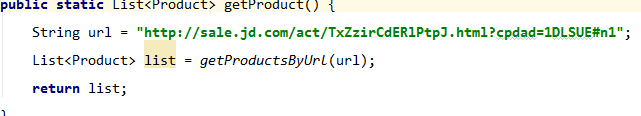
4）**排序过滤**：将推荐的结果按照业务规则进行混合排序及去重等操作。

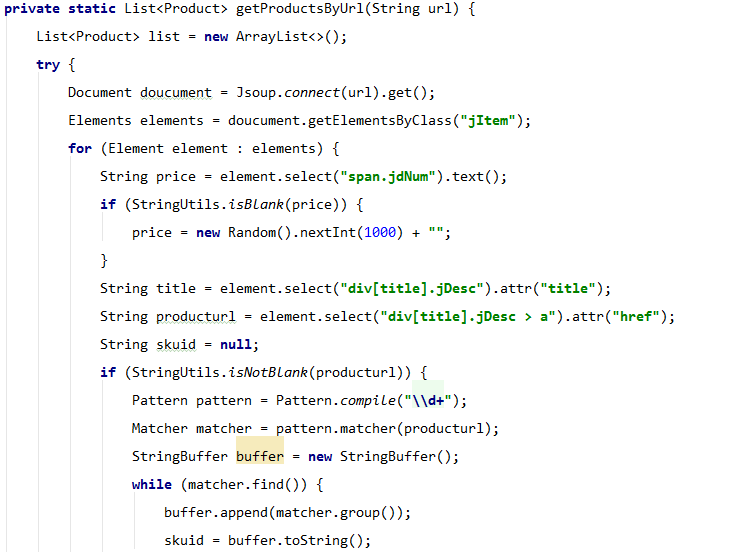
5）**最终推荐结果：**基于业务业务规则对业务推荐的产品进行设置。

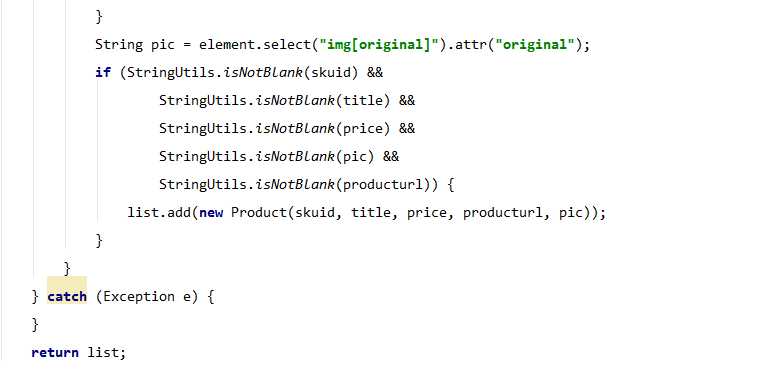
## 代码开发

### 获取某电商的产品信息

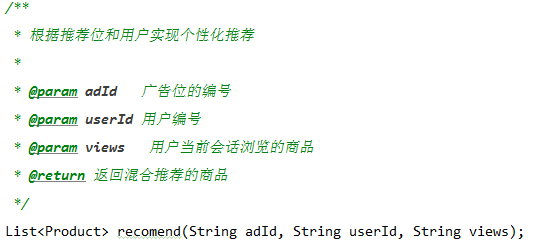
通过爬取某电商专题页的产品信息，作为本案的基础数据。



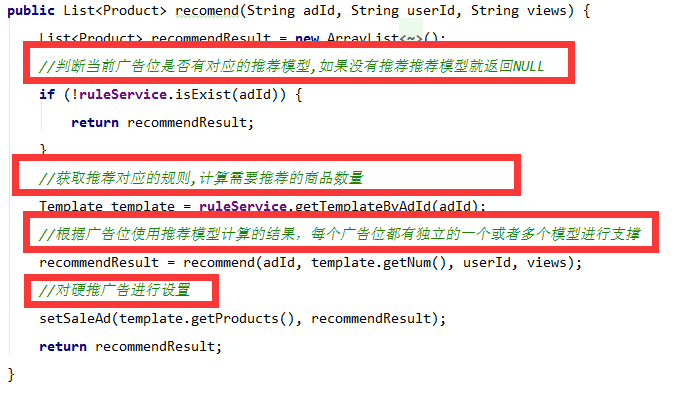




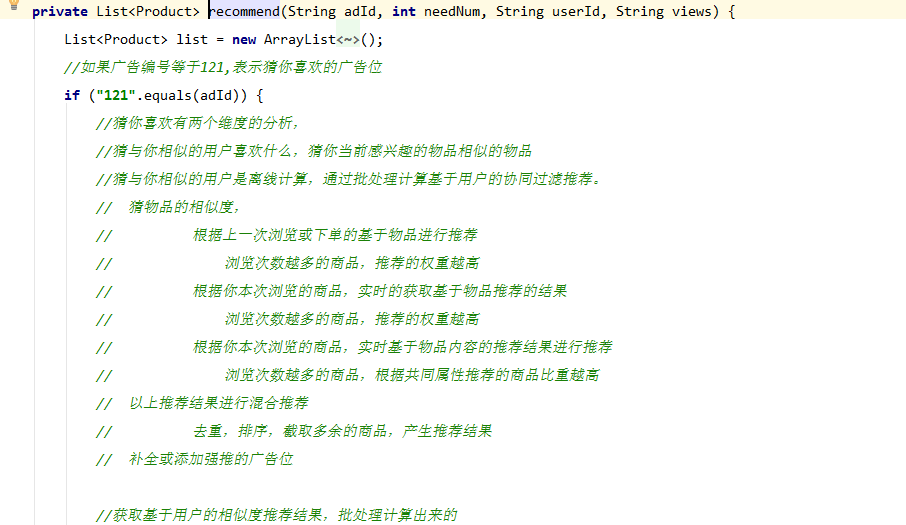
### 猜你喜欢推荐接口，输入广告位编号、用户编号、用户当前会话浏览的商品

口

### 猜你喜欢推荐模块主要流程



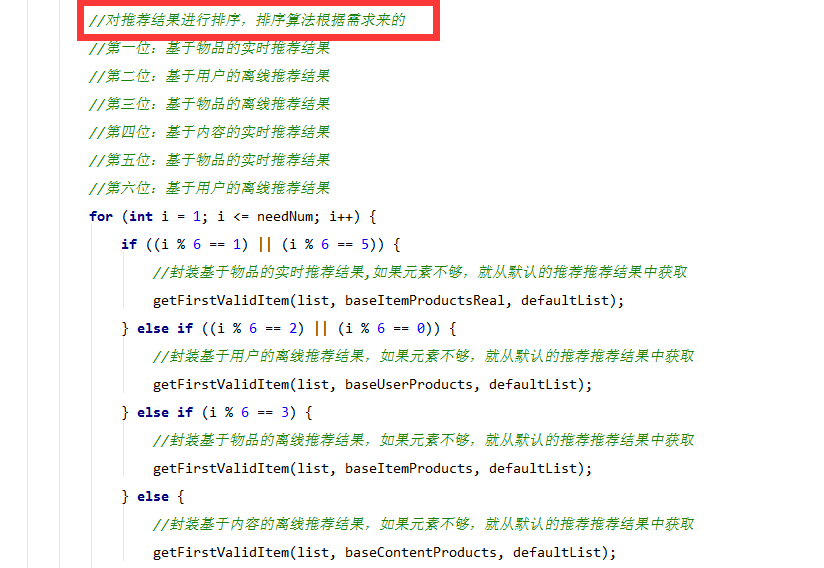
### 为猜你喜欢广告位进行推荐，分析猜你喜欢的推荐思路



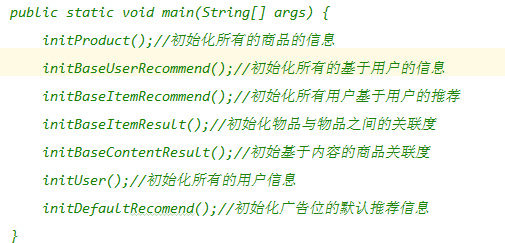
### 猜你喜欢获取推荐结果的步骤



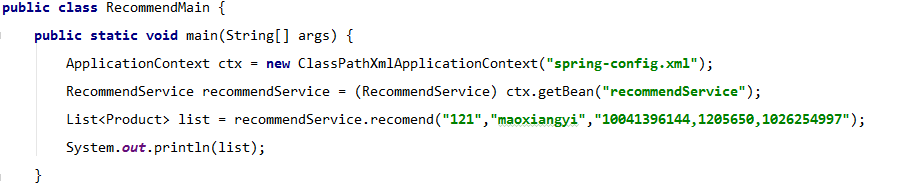
### 基于业务需求的推荐规则的算法



### 数据初始化模块，初始化各种基础信息



### 为用户实时推荐商品





# 7.推荐系统要求实时性

**推荐结果的时间尽量控制在：30-50ms，300ms以下**