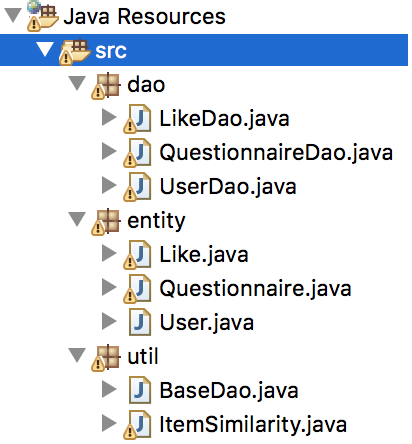
推荐算法部分使用文档

1. 算法相关系统板块
2. 板块名称：社区板块
3. 社区板块负责人：朱雪月
4. 相关功能：推荐问卷靠后部分推荐问卷排序
5. 算法部分代码使用

打开RecommandedSystem文件夹，把Java Resource-src中的dao、entity和util全部复制出来即可



1. 接口
2. 接口路径：

RecommandedSystem-Java Resource-src-util-ItemSimilarity.java-line 12~22

对应代码：

**public** **static** **void** main(String[] args) {

//测试小Demo，在正式整合的时候需要和前端界面结合，input是当前登陆的用户id

System.***out***.println("请输入用户希望推荐的用户id : ");

Scanner input = **new** Scanner(System.***in***);

**int** userID = input.nextInt();

ArrayList<Questionnaire> v = *recommend*(userID);

System.***out***.println("正在生成针对用户id为"+userID+"的推荐...");

**for**(**int** i = 0; i < v.size(); i++)

System.***out***.println("第"+(i+1)+"个推荐: 问卷题目:"+ v.get(i).getTitle()+"");

}

1. 代码修改：

~~这部分代码要删掉~~ 这部分代码要删掉

【这部分代码要做替换】 这部分代码要做替换

对应代码：

**public** **static** **void** main(String[] args) {

~~//测试小Demo，在正式整合的时候需要和前端界面结合，input是当前登陆的用户id~~

~~System.out.println("请输入用户希望推荐的用户id : ");~~

~~Scanner input = new Scanner(System.in);~~

**int** userID = ~~input.nextInt();~~ 【闫羽处读入当前登陆用户的userID，如果未登陆，当前登陆用户的userID为0】

ArrayList<Questionnaire> v = *recommend*(userID);

~~System.out.println("正在生成针对用户id为"+userID+"的推荐...");~~

**for**(**int** i = 0; i < v.size(); i++)

~~System.out.println("第"+(i+1)+"个推荐: 问卷题目:"+~~ v.get(i).getTitle()~~+"");~~【把v.get(i).getTitle()传出去】

}

1. 环境配置
   1. 添加jar包：

hibernate-jpa-2.1-api-1.0.0.final.jar

fastjson-1.2.62.jar

mysql-connector-java-5.1.38-bin.jar

* 1. Mysql配置：

代码链接数据库：

RecommandedSystem-Java Resource-src-util-BaseDao.java-line 12~15

对应代码：

private static final String driver="com.mysql.jdbc.Driver";

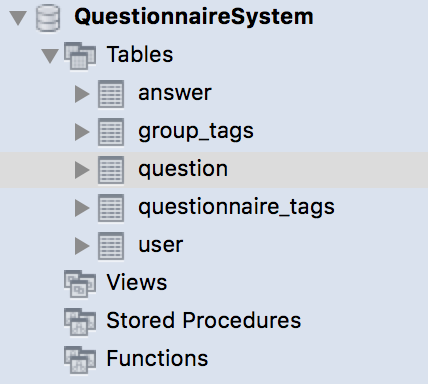
private static final String url="jdbc:mysql://localhost:3306/QuestionnaireSystem?serverTimezone=Asia/Shanghai&useUnicode=true&characterEncoding=utf8&useSSL=false";

private static final String uid="root";

private static final String pwd="webshufe";

1. 数据库说明

Scheme名称：QuestionnaireSystem，包含表：answer、group\_tags、question、questionnaire\_tags、user



1. 代码逻辑
   1. 代码核心逻辑：

基于物品的协同过滤算法，即：**问卷A和问卷B具有很大的相似度是因为喜欢问卷A的用户大都也喜欢问卷B。**

* 1. 问卷相似度计算公式：

 ，N(i)是喜欢物品i的用户数

* 1. 代码流程图

