

Quiz后端-7、实现题目页的删除与查询

日期：2025.07.24 作者：tfzhang

后端主要提供用户与题目的注册、查询、删除等操作。使用到的开发工具：

- IDEA2024; 关注公众号”青椒工具”，发送”IDEA”，获取windows下的IDEA安装包
- mysql 5.7; 关注公众号”青椒工具”，发送”mysql”，获取windows下的mysql5.7安装包；

1、题目的删除；

- 根据id删除题目；
- 主要采用逻辑删除的方式；

参考删除用户的代码来实现。

步骤1：questionmapper中添加删除方法

在QuestionMapper中添加如下的代码：

```
1 //根据id删除题目;
2 @Update("UPDATE questions SET isDelete = 1 WHERE id = #{id} AND isDelete
= 0")
3 public int delQuestionById(Integer id);
```

步骤2：测试QuestionMapper方法

测试上述的删除方法。

步骤3：在QuestionService中添加接口及方法

```
1 //接口;
2 public boolean delQuestion(Integer id);
3
4 //实现;
5 public boolean delQuestion(Integer id){
6     int result = questionMapper.delQuestionById(id);
7     return result>0;
8 }
```

步骤4：controller中添加代码

```
1 @GetMapping ("/delQuestion")
2 public Result deleteQuestion(Integer id) {
3     boolean success = questionService.delQuestion(id);
4     if (success) {
5         return Result.success("用户成功删除");
6     }
7     return Result.error("用户不存在或已被删除");
8 }
```

步骤5：在apifox中测试

2、题目的查询；

题目的查询来自用户端和管理端，为了作为区分：

用户端的查询称为：获取题目；

管理端的查询称为：查询题目；其中管理端的查询与用户查询类似，分为：

- 分页查询与按关键词查询；

用户端获取题目的实体类：

用户端需要的数据格式：

```
1 const quizQuestions = [
2   {
3     question: "What is the capital of France?",
4     answers: [
5       { text: "London", correct: false },
6       { text: "Berlin", correct: false },
7       { text: "Paris", correct: true },
8       { text: "Madrid", correct: false },
9     ],
10   },
11   {
12     question: "Which planet is known as the Red Planet?",
13     answers: [
14       { text: "Venus", correct: false },
15       { text: "Mars", correct: true },
16       { text: "Jupiter", correct: false },
17       { text: "Saturn", correct: false },
18     ],
19   }
20 ]
```

根据上述用户端的数据格式，创建实体类QSBeanOut:

```
1 public class QSBeanOut{
2   private String question;
3   private List<AnsBean> answers;
4   //setter.getter.
5 }
```

再创建AnsBean类：

```
1 public class AnsBean{  
2     private String text;  
3     private boolean correct;  
4     //类构造方法:  
5     public AnsBean(String text, boolean correct) {  
6         this.text = text;  
7         this.correct = correct;  
8     }  
9     //setter.getter.  
10 }
```

获取题目的**QuestionMapper**:

用户每次获取题目的数量是固定的，就是5个题目；

对应的sql语句如下：

```
1 | SELECT * FROM questions ORDER BY RAND() LIMIT 5;
```

对应的代码：

```
1 @Mapper  
2 public interface QuestionMapper {  
3     @Select("SELECT * FROM questions where isDelete=0 ORDER BY RAND() LIMIT  
4     5")  
5     public List<Question> getQuestions();  
6 }
```

测试**getQuestions**方法

```
1 @Test  
2 public void testQuestionList(){  
3     List<Question> qlist = questionMapper.getQuestions();  
4     for (Question q : qlist) {  
5         System.out.println(q);  
6     }  
7 }
```

Controller层创建方法：

```
1 @GetMapping("/getQuestion")  
2 public Result getQuestion() {  
3     List <QSBeanOut> qsBeanOut = questionService.getQuestions();  
4     return Result.success(qsBeanOut);  
5 }
```

此处返回的对象是QSBeanOut的List，但是在mapper层返回的是Question类型的list，所以此处我们需要作一个转换，我们在创建utils目录，并在其中撰写转换的工具方法；

```
1 public static List<QSBeanOut> convertToQSBeanList(List<Question>  
questionList) {  
2     List<QSBeanOut> result = new ArrayList<>();  
3  
4     for (Question question : questionList) {
```

```

5     QSBeanOut qsBean = new QSBeanOut();
6     qsBean.setQuestion(question.getQuestionText());
7
8     List<AnsBean> answers = new ArrayList<>();
9     answers.add(new AnsBean(question.getAnswer1Text(),
10                           question.getAnswer1Correct()));
11    answers.add(new AnsBean(question.getAnswer2Text(),
12                           question.getAnswer2Correct()));
13    answers.add(new AnsBean(question.getAnswer3Text(),
14                           question.getAnswer3Correct()));
15    answers.add(new AnsBean(question.getAnswer4Text(),
16                           question.getAnswer4Correct()));
17
18     qsBean.setAnswers(answers);
19     result.add(qsBean);
20 }
21
22 return result;
23 }
```

Service层的接口与代码

```

1 //interface:
2 //获取5个新问题;
3 public List<QSBeanOut> getQuestions();
4
5 //impl
6 public List<QSBeanOut> getQuestions(){
7     //获取question list;
8     List<Question> questionList = questionMapper.getQuestions();
9     //将question list转化为QSBeanOut;
10    List<QSBeanOut> qsBeanOutList =
11        Tools.convertToQSBeanList(questionList);
12    return qsBeanOutList;
13 }
```

apifox端测试

3、管理端的查询

管理端的题目查询与用户的查询比较一致，因此可以参考用户的查询：

- 分页查询
- 按题目的关键词查询；

首先要观察管理端的前端页面，来确定实体类：

Quiz后台管理

The screenshot shows a management interface for a quiz system. On the left, there's a sidebar with '管理选项' (Management Options) expanded, showing '用户管理' (User Management) and '题目管理' (Question Management). The main area has a search bar with placeholder '请输入题目关键词' (Enter question keyword), a '查询' (Search) button, and a green '添加题目' (Add Question) button. Below is a table listing questions:

序号	题目	选项	答案	操作
1	法国的首都是哪个城市?	巴黎, 巴黎, 巴黎, 巴黎	巴黎	<button>编辑</button> <button>删除</button>
2	法国的首都是哪个城市?	巴黎, 巴黎, 巴黎, 巴黎	巴黎	<button>编辑</button> <button>删除</button>
3	法国的首都是哪个城市?	巴黎, 巴黎, 巴黎, 巴黎	巴黎	<button>编辑</button> <button>删除</button>
4	法国的首都是哪个城市?	巴黎, 巴黎, 巴黎, 巴黎	巴黎	<button>编辑</button> <button>删除</button>

At the bottom, there are navigation buttons: < (disabled), 1 (highlighted in blue), 2, 3, 4, 5, 6, ..., 100, >.

分页和按题目关键词查询的题目都包含三个部分：

- 题目；
- 选项；
- 答案；

对应的数据接口：

```
1 | {
2 |     questionText: "题目内容",
3 |     options: ["a", "b", "c", "d"],
4 |     answer: "b"
5 | }
```

在model目录中，创建相应的类：

```
1 | public class QSBeanPage{
2 |     private int total;
3 |     private List<QSBeanOutManage> qsBeanList;
4 | }
5 |
6 | public class QSBeanOutManage {
7 |     private String questionText;
8 |     private List<String> options;
9 |     private String answer;
10| }
```

步骤2：questionmapper中添加代码

```
1 | @Select("SELECT COUNT(*) FROM questions WHERE isDelete=0")
2 | public int count();
3 |
4 | @Select("SELECT * FROM questions WHERE isDelete=0 limit #{start},#{pagesize}")
5 | public List<Question> page(Integer start, Integer pageSize);
```

步骤3：QuestionController中添加代码

```

1  @GetMapping("/questions")
2  public Result getPage(@RequestParam(defaultValue="1")Integer page,
3      @RequestParam(defaultValue="5")Integer pageSize){
4      QSBeanPage qsBeanPage=questionService.page(page, pageSize);
5      return Result.success(qsBeanPage);
6  }

```

步骤4: Question Service层中添加代码

```

1 //接口:
2 public QSBeanPage page(Integer page, Integer pageSize);
3
4 //实现:
5 public QSBeanPage page(Integer page, Integer pageSize){
6     //获取总的记录数;
7     Integer total=questionMapper.count();
8
9     //获取分页查询结果列表;
10    Integer start = (page-1)*pageSize;
11    List<Question> questionList=questionMapper.page(start, pageSize);
12
13    //将questionList转化为QSBeanOutManage
14    //to be done;
15    QSBeanOutManage qsBeanOutManage =
16    Tools.convertToQSBeanManageList(questionList);
17
18    //封装PageBean对象;
19    QSBeanPage qsBeanPage = new QSBeanPage();
20    qsBeanPage.setTotal(total);
21    qsBeanPage.setQsBeanList(qsBeanOutManage);
22
23    return qsBeanPage;
}

```

其中的Tools.convertToQSBeanManageList是定义在utils中的工具类:

```

1  public static List<QSBeanOutManage>
2  convertToQSBeanManageList(List<Question> questionList) {
3      List<QSBeanOutManage> result = new ArrayList<>();
4      if (questionList == null) {
5          return result;
6      }
7
8      for (Question question : questionList) {
9          // 跳过已删除的问题
10         if (question.getIsDelete() != null && question.getIsDelete() ==
11             1) {
12             continue;
13         }
14
15         QSBeanOutManage bean = new QSBeanOutManage();
16         bean.setQuestionText(question.getQuestionText());
17
18         // 构建选项列表（固定4个选项）
19         List<String> options = new ArrayList<>(4);
20
21         // 构建答案列表
22         List<QSBeanOutAnswer> answers = new ArrayList<>(4);
23
24         // 构建评论列表
25         List<QSBeanOutComment> comments = new ArrayList<>(4);
26
27         // 构建点赞列表
28         List<QSBeanOutLike> likes = new ArrayList<>(4);
29
30         // 构建收藏列表
31         List<QSBeanOutCollect> collects = new ArrayList<>(4);
32
33         // 构建举报列表
34         List<QSBeanOutReport> reports = new ArrayList<>(4);
35
36         // 构建分享列表
37         List<QSBeanOutShare> shares = new ArrayList<>(4);
38
39         // 构建回复列表
40         List<QSBeanOutReply> replies = new ArrayList<>(4);
41
42         // 构建点赞统计
43         QSBeanOutLikeCount likeCount = new QSBeanOutLikeCount();
44         likeCount.setCount(0);
45
46         // 构建收藏统计
47         QSBeanOutCollectCount collectCount = new QSBeanOutCollectCount();
48         collectCount.setCount(0);
49
50         // 构建举报统计
51         QSBeanOutReportCount reportCount = new QSBeanOutReportCount();
52         reportCount.setCount(0);
53
54         // 构建分享统计
55         QSBeanOutShareCount shareCount = new QSBeanOutShareCount();
56         shareCount.setCount(0);
57
58         // 构建回复统计
59         QSBeanOutReplyCount replyCount = new QSBeanOutReplyCount();
60         replyCount.setCount(0);
61
62         // 构建评论统计
63         QSBeanOutCommentCount commentCount = new QSBeanOutCommentCount();
64         commentCount.setCount(0);
65
66         // 构建点赞数
67         QSBeanOutLikeCount likeCount2 = new QSBeanOutLikeCount();
68         likeCount2.setCount(0);
69
70         // 构建收藏数
71         QSBeanOutCollectCount collectCount2 = new QSBeanOutCollectCount();
72         collectCount2.setCount(0);
73
74         // 构建举报数
75         QSBeanOutReportCount reportCount2 = new QSBeanOutReportCount();
76         reportCount2.setCount(0);
77
78         // 构建分享数
79         QSBeanOutShareCount shareCount2 = new QSBeanOutShareCount();
80         shareCount2.setCount(0);
81
82         // 构建回复数
83         QSBeanOutReplyCount replyCount2 = new QSBeanOutReplyCount();
84         replyCount2.setCount(0);
85
86         // 构建评论数
87         QSBeanOutCommentCount commentCount2 = new QSBeanOutCommentCount();
88         commentCount2.setCount(0);
89
90         // 构建点赞数2
91         QSBeanOutLikeCount likeCount3 = new QSBeanOutLikeCount();
92         likeCount3.setCount(0);
93
94         // 构建收藏数2
95         QSBeanOutCollectCount collectCount3 = new QSBeanOutCollectCount();
96         collectCount3.setCount(0);
97
98         // 构建举报数2
99         QSBeanOutReportCount reportCount3 = new QSBeanOutReportCount();
100        reportCount3.setCount(0);
101
102        // 构建分享数2
103        QSBeanOutShareCount shareCount3 = new QSBeanOutShareCount();
104        shareCount3.setCount(0);
105
106        // 构建回复数2
107        QSBeanOutReplyCount replyCount3 = new QSBeanOutReplyCount();
108        replyCount3.setCount(0);
109
110        // 构建评论数2
111        QSBeanOutCommentCount commentCount3 = new QSBeanOutCommentCount();
112        commentCount3.setCount(0);
113
114        // 构建点赞数3
115        QSBeanOutLikeCount likeCount4 = new QSBeanOutLikeCount();
116        likeCount4.setCount(0);
117
118        // 构建收藏数3
119        QSBeanOutCollectCount collectCount4 = new QSBeanOutCollectCount();
120        collectCount4.setCount(0);
121
122        // 构建举报数3
123        QSBeanOutReportCount reportCount4 = new QSBeanOutReportCount();
124        reportCount4.setCount(0);
125
126        // 构建分享数3
127        QSBeanOutShareCount shareCount4 = new QSBeanOutShareCount();
128        shareCount4.setCount(0);
129
130        // 构建回复数3
131        QSBeanOutReplyCount replyCount4 = new QSBeanOutReplyCount();
132        replyCount4.setCount(0);
133
134        // 构建评论数3
135        QSBeanOutCommentCount commentCount4 = new QSBeanOutCommentCount();
136        commentCount4.setCount(0);
137
138        // 构建点赞数4
139        QSBeanOutLikeCount likeCount5 = new QSBeanOutLikeCount();
140        likeCount5.setCount(0);
141
142        // 构建收藏数4
143        QSBeanOutCollectCount collectCount5 = new QSBeanOutCollectCount();
144        collectCount5.setCount(0);
145
146        // 构建举报数4
147        QSBeanOutReportCount reportCount5 = new QSBeanOutReportCount();
148        reportCount5.setCount(0);
149
150        // 构建分享数4
151        QSBeanOutShareCount shareCount5 = new QSBeanOutShareCount();
152        shareCount5.setCount(0);
153
154        // 构建回复数4
155        QSBeanOutReplyCount replyCount5 = new QSBeanOutReplyCount();
156        replyCount5.setCount(0);
157
158        // 构建评论数4
159        QSBeanOutCommentCount commentCount5 = new QSBeanOutCommentCount();
160        commentCount5.setCount(0);
161
162        // 构建点赞数5
163        QSBeanOutLikeCount likeCount6 = new QSBeanOutLikeCount();
164        likeCount6.setCount(0);
165
166        // 构建收藏数5
167        QSBeanOutCollectCount collectCount6 = new QSBeanOutCollectCount();
168        collectCount6.setCount(0);
169
170        // 构建举报数5
171        QSBeanOutReportCount reportCount6 = new QSBeanOutReportCount();
172        reportCount6.setCount(0);
173
174        // 构建分享数5
175        QSBeanOutShareCount shareCount6 = new QSBeanOutShareCount();
176        shareCount6.setCount(0);
177
178        // 构建回复数5
179        QSBeanOutReplyCount replyCount6 = new QSBeanOutReplyCount();
180        replyCount6.setCount(0);
181
182        // 构建评论数5
183        QSBeanOutCommentCount commentCount6 = new QSBeanOutCommentCount();
184        commentCount6.setCount(0);
185
186        // 构建点赞数6
187        QSBeanOutLikeCount likeCount7 = new QSBeanOutLikeCount();
188        likeCount7.setCount(0);
189
190        // 构建收藏数6
191        QSBeanOutCollectCount collectCount7 = new QSBeanOutCollectCount();
192        collectCount7.setCount(0);
193
194        // 构建举报数6
195        QSBeanOutReportCount reportCount7 = new QSBeanOutReportCount();
196        reportCount7.setCount(0);
197
198        // 构建分享数6
199        QSBeanOutShareCount shareCount7 = new QSBeanOutShareCount();
200        shareCount7.setCount(0);
201
202        // 构建回复数6
203        QSBeanOutReplyCount replyCount7 = new QSBeanOutReplyCount();
204        replyCount7.setCount(0);
205
206        // 构建评论数6
207        QSBeanOutCommentCount commentCount7 = new QSBeanOutCommentCount();
208        commentCount7.setCount(0);
209
210        // 构建点赞数7
211        QSBeanOutLikeCount likeCount8 = new QSBeanOutLikeCount();
212        likeCount8.setCount(0);
213
214        // 构建收藏数7
215        QSBeanOutCollectCount collectCount8 = new QSBeanOutCollectCount();
216        collectCount8.setCount(0);
217
218        // 构建举报数7
219        QSBeanOutReportCount reportCount8 = new QSBeanOutReportCount();
220        reportCount8.setCount(0);
221
222        // 构建分享数7
223        QSBeanOutShareCount shareCount8 = new QSBeanOutShareCount();
224        shareCount8.setCount(0);
225
226        // 构建回复数7
227        QSBeanOutReplyCount replyCount8 = new QSBeanOutReplyCount();
228        replyCount8.setCount(0);
229
230        // 构建评论数7
231        QSBeanOutCommentCount commentCount8 = new QSBeanOutCommentCount();
232        commentCount8.setCount(0);
233
234        // 构建点赞数8
235        QSBeanOutLikeCount likeCount9 = new QSBeanOutLikeCount();
236        likeCount9.setCount(0);
237
238        // 构建收藏数8
239        QSBeanOutCollectCount collectCount9 = new QSBeanOutCollectCount();
240        collectCount9.setCount(0);
241
242        // 构建举报数8
243        QSBeanOutReportCount reportCount9 = new QSBeanOutReportCount();
244        reportCount9.setCount(0);
245
246        // 构建分享数8
247        QSBeanOutShareCount shareCount9 = new QSBeanOutShareCount();
248        shareCount9.setCount(0);
249
250        // 构建回复数8
251        QSBeanOutReplyCount replyCount9 = new QSBeanOutReplyCount();
252        replyCount9.setCount(0);
253
254        // 构建评论数8
255        QSBeanOutCommentCount commentCount9 = new QSBeanOutCommentCount();
256        commentCount9.setCount(0);
257
258        // 构建点赞数9
259        QSBeanOutLikeCount likeCount10 = new QSBeanOutLikeCount();
260        likeCount10.setCount(0);
261
262        // 构建收藏数9
263        QSBeanOutCollectCount collectCount10 = new QSBeanOutCollectCount();
264        collectCount10.setCount(0);
265
266        // 构建举报数9
267        QSBeanOutReportCount reportCount10 = new QSBeanOutReportCount();
268        reportCount10.setCount(0);
269
270        // 构建分享数9
271        QSBeanOutShareCount shareCount10 = new QSBeanOutShareCount();
272        shareCount10.setCount(0);
273
274        // 构建回复数9
275        QSBeanOutReplyCount replyCount10 = new QSBeanOutReplyCount();
276        replyCount10.setCount(0);
277
278        // 构建评论数9
279        QSBeanOutCommentCount commentCount10 = new QSBeanOutCommentCount();
280        commentCount10.setCount(0);
281
282        // 构建点赞数10
283        QSBeanOutLikeCount likeCount11 = new QSBeanOutLikeCount();
284        likeCount11.setCount(0);
285
286        // 构建收藏数10
287        QSBeanOutCollectCount collectCount11 = new QSBeanOutCollectCount();
288        collectCount11.setCount(0);
289
290        // 构建举报数10
291        QSBeanOutReportCount reportCount11 = new QSBeanOutReportCount();
292        reportCount11.setCount(0);
293
294        // 构建分享数10
295        QSBeanOutShareCount shareCount11 = new QSBeanOutShareCount();
296        shareCount11.setCount(0);
297
298        // 构建回复数10
299        QSBeanOutReplyCount replyCount11 = new QSBeanOutReplyCount();
300        replyCount11.setCount(0);
301
302        // 构建评论数10
303        QSBeanOutCommentCount commentCount11 = new QSBeanOutCommentCount();
304        commentCount11.setCount(0);
305
306        // 构建点赞数11
307        QSBeanOutLikeCount likeCount12 = new QSBeanOutLikeCount();
308        likeCount12.setCount(0);
309
310        // 构建收藏数11
311        QSBeanOutCollectCount collectCount12 = new QSBeanOutCollectCount();
312        collectCount12.setCount(0);
313
314        // 构建举报数11
315        QSBeanOutReportCount reportCount12 = new QSBeanOutReportCount();
316        reportCount12.setCount(0);
317
318        // 构建分享数11
319        QSBeanOutShareCount shareCount12 = new QSBeanOutShareCount();
320        shareCount12.setCount(0);
321
322        // 构建回复数11
323        QSBeanOutReplyCount replyCount12 = new QSBeanOutReplyCount();
324        replyCount12.setCount(0);
325
326        // 构建评论数11
327        QSBeanOutCommentCount commentCount12 = new QSBeanOutCommentCount();
328        commentCount12.setCount(0);
329
330        // 构建点赞数12
331        QSBeanOutLikeCount likeCount13 = new QSBeanOutLikeCount();
332        likeCount13.setCount(0);
333
334        // 构建收藏数12
335        QSBeanOutCollectCount collectCount13 = new QSBeanOutCollectCount();
336        collectCount13.setCount(0);
337
338        // 构建举报数12
339        QSBeanOutReportCount reportCount13 = new QSBeanOutReportCount();
340        reportCount13.setCount(0);
341
342        // 构建分享数12
343        QSBeanOutShareCount shareCount13 = new QSBeanOutShareCount();
344        shareCount13.setCount(0);
345
346        // 构建回复数12
347        QSBeanOutReplyCount replyCount13 = new QSBeanOutReplyCount();
348        replyCount13.setCount(0);
349
350        // 构建评论数12
351        QSBeanOutCommentCount commentCount13 = new QSBeanOutCommentCount();
352        commentCount13.setCount(0);
353
354        // 构建点赞数13
355        QSBeanOutLikeCount likeCount14 = new QSBeanOutLikeCount();
356        likeCount14.setCount(0);
357
358        // 构建收藏数13
359        QSBeanOutCollectCount collectCount14 = new QSBeanOutCollectCount();
360        collectCount14.setCount(0);
361
362        // 构建举报数13
363        QSBeanOutReportCount reportCount14 = new QSBeanOutReportCount();
364        reportCount14.setCount(0);
365
366        // 构建分享数13
367        QSBeanOutShareCount shareCount14 = new QSBeanOutShareCount();
368        shareCount14.setCount(0);
369
370        // 构建回复数13
371        QSBeanOutReplyCount replyCount14 = new QSBeanOutReplyCount();
372        replyCount14.setCount(0);
373
374        // 构建评论数13
375        QSBeanOutCommentCount commentCount14 = new QSBeanOutCommentCount();
376        commentCount14.setCount(0);
377
378        // 构建点赞数14
379        QSBeanOutLikeCount likeCount15 = new QSBeanOutLikeCount();
380        likeCount15.setCount(0);
381
382        // 构建收藏数14
383        QSBeanOutCollectCount collectCount15 = new QSBeanOutCollectCount();
384        collectCount15.setCount(0);
385
386        // 构建举报数14
387        QSBeanOutReportCount reportCount15 = new QSBeanOutReportCount();
388        reportCount15.setCount(0);
389
390        // 构建分享数14
391        QSBeanOutShareCount shareCount15 = new QSBeanOutShareCount();
392        shareCount15.setCount(0);
393
394        // 构建回复数14
395        QSBeanOutReplyCount replyCount15 = new QSBeanOutReplyCount();
396        replyCount15.setCount(0);
397
398        // 构建评论数14
399        QSBeanOutCommentCount commentCount15 = new QSBeanOutCommentCount();
400        commentCount15.setCount(0);
401
402        // 构建点赞数15
403        QSBeanOutLikeCount likeCount16 = new QSBeanOutLikeCount();
404        likeCount16.setCount(0);
405
406        // 构建收藏数15
407        QSBeanOutCollectCount collectCount16 = new QSBeanOutCollectCount();
408        collectCount16.setCount(0);
409
410        // 构建举报数15
411        QSBeanOutReportCount reportCount16 = new QSBeanOutReportCount();
412        reportCount16.setCount(0);
413
414        // 构建分享数15
415        QSBeanOutShareCount shareCount16 = new QSBeanOutShareCount();
416        shareCount16.setCount(0);
417
418        // 构建回复数15
419        QSBeanOutReplyCount replyCount16 = new QSBeanOutReplyCount();
420        replyCount16.setCount(0);
421
422        // 构建评论数15
423        QSBeanOutCommentCount commentCount16 = new QSBeanOutCommentCount();
424        commentCount16.setCount(0);
425
426        // 构建点赞数16
427        QSBeanOutLikeCount likeCount17 = new QSBeanOutLikeCount();
428        likeCount17.setCount(0);
429
430        // 构建收藏数16
431        QSBeanOutCollectCount collectCount17 = new QSBeanOutCollectCount();
432        collectCount17.setCount(0);
433
434        // 构建举报数16
435        QSBeanOutReportCount reportCount17 = new QSBeanOutReportCount();
436        reportCount17.setCount(0);
437
438        // 构建分享数16
439        QSBeanOutShareCount shareCount17 = new QSBeanOutShareCount();
440        shareCount17.setCount(0);
441
442        // 构建回复数16
443        QSBeanOutReplyCount replyCount17 = new QSBeanOutReplyCount();
444        replyCount17.setCount(0);
445
446        // 构建评论数16
447        QSBeanOutCommentCount commentCount17 = new QSBeanOutCommentCount();
448        commentCount17.setCount(0);
449
450        // 构建点赞数17
451        QSBeanOutLikeCount likeCount18 = new QSBeanOutLikeCount();
452        likeCount18.setCount(0);
453
454        // 构建收藏数17
455        QSBeanOutCollectCount collectCount18 = new QSBeanOutCollectCount();
456        collectCount18.setCount(0);
457
458        // 构建举报数17
459        QSBeanOutReportCount reportCount18 = new QSBeanOutReportCount();
460        reportCount18.setCount(0);
461
462        // 构建分享数17
463        QSBeanOutShareCount shareCount18 = new QSBeanOutShareCount();
464        shareCount18.setCount(0);
465
466        // 构建回复数17
467        QSBeanOutReplyCount replyCount18 = new QSBeanOutReplyCount();
468        replyCount18.setCount(0);
469
470        // 构建评论数17
471        QSBeanOutCommentCount commentCount18 = new QSBeanOutCommentCount();
472        commentCount18.setCount(0);
473
474        // 构建点赞数18
475        QSBeanOutLikeCount likeCount19 = new QSBeanOutLikeCount();
476        likeCount19.setCount(0);
477
478        // 构建收藏数18
479        QSBeanOutCollectCount collectCount19 = new QSBeanOutCollectCount();
480        collectCount19.setCount(0);
481
482        // 构建举报数18
483        QSBeanOutReportCount reportCount19 = new QSBeanOutReportCount();
484        reportCount19.setCount(0);
485
486        // 构建分享数18
487        QSBeanOutShareCount shareCount19 = new QSBeanOutShareCount();
488        shareCount19.setCount(0);
489
490        // 构建回复数18
491        QSBeanOutReplyCount replyCount19 = new QSBeanOutReplyCount();
492        replyCount19.setCount(0);
493
494        // 构建评论数18
495        QSBeanOutCommentCount commentCount19 = new QSBeanOutCommentCount();
496        commentCount19.setCount(0);
497
498        // 构建点赞数19
499        QSBeanOutLikeCount likeCount20 = new QSBeanOutLikeCount();
500        likeCount20.setCount(0);
501
502        // 构建收藏数19
503        QSBeanOutCollectCount collectCount20 = new QSBeanOutCollectCount();
504        collectCount20.setCount(0);
505
506        // 构建举报数19
507        QSBeanOutReportCount reportCount20 = new QSBeanOutReportCount();
508        reportCount20.setCount(0);
509
510        // 构建分享数19
511        QSBeanOutShareCount shareCount20 = new QSBeanOutShareCount();
512        shareCount20.setCount(0);
513
514        // 构建回复数19
515        QSBeanOutReplyCount replyCount20 = new QSBeanOutReplyCount();
516        replyCount20.setCount(0);
517
518        // 构建评论数19
519        QSBeanOutCommentCount commentCount20 = new QSBeanOutCommentCount();
520        commentCount20.setCount(0);
521
522        // 构建点赞数20
523        QSBeanOutLikeCount likeCount21 = new QSBeanOutLikeCount();
524        likeCount21.setCount(0);
525
526        // 构建收藏数20
527        QSBeanOutCollectCount collectCount21 = new QSBeanOutCollectCount();
528        collectCount21.setCount(0);
529
530        // 构建举报数20
531        QSBeanOutReportCount reportCount21 = new QSBeanOutReportCount();
532        reportCount21.setCount(0);
533
534        // 构建分享数20
535        QSBeanOutShareCount shareCount21 = new QSBeanOutShareCount();
536        shareCount21.setCount(0);
537
538        // 构建回复数20
539        QSBeanOutReplyCount replyCount21 = new QSBeanOutReplyCount();
540        replyCount21.setCount(0);
541
542        // 构建评论数20
543        QSBeanOutCommentCount commentCount21 = new QSBeanOutCommentCount();
544        commentCount21.setCount(0);
545
546        // 构建点赞数21
547        QSBeanOutLikeCount likeCount22 = new QSBeanOutLikeCount();
548        likeCount22.setCount(0);
549
550        // 构建收藏数21
551        QSBeanOutCollectCount collectCount22 = new QSBeanOutCollectCount();
552        collectCount22.setCount(0);
553
554        // 构建举报数21
555        QSBeanOutReportCount reportCount22 = new QSBeanOutReportCount();
556        reportCount22.setCount(0);
557
558        // 构建分享数21
559        QSBeanOutShareCount shareCount22 = new QSBeanOutShareCount();
560        shareCount22.setCount(0);
561
562        // 构建回复数21
563        QSBeanOutReplyCount replyCount22 = new QSBeanOutReplyCount();
564        replyCount22.setCount(0);
565
566        // 构建评论数21
567        QSBeanOutCommentCount commentCount22 = new QSBeanOutCommentCount();
568        commentCount22.setCount(0);
569
570        // 构建点赞数22
571        QSBeanOutLikeCount likeCount23 = new QSBeanOutLikeCount();
572        likeCount23.setCount(0);
573
574        // 构建收藏数22
575        QSBeanOutCollectCount collectCount23 = new QSBeanOutCollectCount();
576        collectCount23.setCount(0);
577
578        // 构建举报数22
579        QSBeanOutReportCount reportCount23 = new QSBeanOutReportCount();
580        reportCount23.setCount(0);
581
582        // 构建分享数22
583        QSBeanOutShareCount shareCount23 = new QSBeanOutShareCount();
584        shareCount23.setCount(0);
585
586        // 构建回复数22
587        QSBeanOutReplyCount replyCount23 = new QSBeanOutReplyCount();
588        replyCount23.setCount(0);
589
590        // 构建评论数22
591        QSBeanOutCommentCount commentCount23 = new QSBeanOutCommentCount();
592        commentCount23.setCount(0);
593
594        // 构建点赞数23
595        QSBeanOutLikeCount likeCount24 = new QSBeanOutLikeCount();
596        likeCount24.setCount(0);
597
598        // 构建收藏数23
599        QSBeanOutCollectCount collectCount24 = new QSBeanOutCollectCount();
600        collectCount24.setCount(0);
601
602        // 构建举报数23
603        QSBeanOutReportCount reportCount24 = new QSBeanOutReportCount();
604        reportCount24.setCount(0);
605
606        // 构建分享数23
607        QSBeanOutShareCount shareCount24 = new QSBeanOutShareCount();
608        shareCount24.setCount(0);
609
610        // 构建回复数23
611        QSBeanOutReplyCount replyCount24 = new QSBeanOutReplyCount();
612        replyCount24.setCount(0);
613
614        // 构建评论数23
615        QSBeanOutCommentCount commentCount24 = new QSBeanOutCommentCount();
616        commentCount24.setCount(0);
617
618        // 构建点赞数24
619        QSBeanOutLikeCount likeCount25 = new QSBeanOutLikeCount();
620        likeCount25.setCount(0);
621
622        // 构建收藏数24
623        QSBeanOutCollectCount collectCount25 = new QSBeanOutCollectCount();
624        collectCount25.setCount(0);
625
626        // 构建举报数24
627        QSBeanOutReportCount reportCount25 = new QSBeanOutReportCount();
628        reportCount25.setCount(0);
629
630        // 构建分享数24
631        QSBeanOutShareCount shareCount25 = new QSBeanOutShareCount();
632        shareCount25.setCount(0);
633
634        // 构建回复数24
635        QSBeanOutReplyCount replyCount25 = new QSBeanOutReplyCount();
636        replyCount25.setCount(0);
637
638        // 构建评论数24
639        QSBeanOutCommentCount commentCount25 = new QSBeanOutCommentCount();
640        commentCount25.setCount(0);
641
642        // 构建点赞数25
643        QSBeanOutLikeCount likeCount26 = new QSBeanOutLikeCount();
644        likeCount26.setCount(0);
645
646        // 构建收藏数25
647        QSBeanOutCollectCount collectCount26 = new QSBeanOutCollectCount();
648        collectCount26.setCount(0);
649
650        // 构建举报数25
651        QSBeanOutReportCount reportCount26 = new QSBeanOutReportCount();
652        reportCount26.setCount(0);
653
654        // 构建分享数25
655        QSBeanOutShareCount shareCount26 = new QSBeanOutShareCount();
656        shareCount26.setCount(0);
657
658        // 构建回复数25
659        QSBeanOutReplyCount replyCount26 = new QSBeanOutReplyCount();
660        replyCount26.setCount(0);
661
662        // 构建评论数25
663        QSBeanOutCommentCount commentCount26 = new QSBeanOutCommentCount();
664        commentCount26.setCount(0);
665
666        // 构建点赞数26
667        QSBeanOutLikeCount likeCount27 = new QSBeanOutLikeCount();
668        likeCount27.setCount(0);
669
670        // 构建收藏数26
671        QSBeanOutCollectCount collectCount27 = new QSBeanOutCollectCount();
672        collectCount27.setCount(0);
673
674        // 构建举报数26
675        QSBeanOutReportCount reportCount27 = new QSBeanOutReportCount();
676        reportCount27.setCount(0);
677
678        // 构建分享数26
679        QSBeanOutShareCount shareCount27 = new QSBeanOutShareCount();
680        shareCount27.setCount(0);
681
682        // 构建回复数26
683        QSBeanOutReplyCount replyCount27 = new QSBeanOutReplyCount();
684        replyCount27.setCount(0);
685
686        // 构建评论数26
687        QSBeanOutCommentCount commentCount27 = new QSBeanOutCommentCount();
688        commentCount27.setCount(0);
689
690        // 构建点赞数27
691        QSBeanOutLikeCount likeCount28 = new QSBeanOutLikeCount();
692        likeCount28.setCount(0);
693
694        // 构建收藏数27
695        QSBeanOutCollectCount collectCount28 = new QSBeanOutCollectCount();
696        collectCount28.setCount(0);
697
698        // 构建举报数27
699        QSBeanOutReportCount reportCount28 = new QSBeanOutReportCount();
700        reportCount28.setCount(0);
701
702        // 构建分享数27
703        QSBeanOutShareCount shareCount28 = new QSBeanOutShareCount();
704        shareCount28.setCount(0);
705
706        // 构建回复数27
707        QSBeanOutReplyCount replyCount28 = new QSBeanOutReplyCount();
708        replyCount28.setCount(0);
709
710        // 构建评论数27
711        QSBeanOutCommentCount commentCount28 = new QSBeanOutCommentCount();
712        commentCount28.setCount(0);
713
714        // 构建点赞数28
715        QSBeanOutLikeCount likeCount29 = new QSBeanOutLikeCount();
716        likeCount29.setCount(0);
717
718        // 构建收藏数28
719        QSBeanOutCollectCount collectCount29 = new QSBeanOutCollectCount();
720        collectCount29.setCount(0);
721
722        // 构建举报数28
723        QSBeanOutReportCount reportCount29 = new QSBeanOutReportCount();
724        reportCount29.setCount(0);
725
726        // 构建分享数28
727        QSBeanOutShareCount shareCount29 = new QSBeanOutShareCount();
728        shareCount29.setCount(0);
729
730        // 构建回复数28
731        QSBeanOutReplyCount replyCount29 = new QSBeanOutReplyCount();
732        replyCount29.setCount(0);
733
734        // 构建评论数28
735        QSBeanOutCommentCount commentCount29 = new QSBeanOutCommentCount();
736        commentCount29.setCount(0);
737
738        // 构建点赞数29
739        QSBeanOutLikeCount likeCount30 = new QSBeanOutLikeCount();
740        likeCount30.setCount(0);
741
742        // 构建收藏数29
743        QSBeanOutCollectCount collectCount30 = new QSBeanOutCollectCount();
744        collectCount30.setCount(0);
745
746        // 构建举报数29
747        QSBeanOutReportCount reportCount30 = new QSBeanOutReportCount();
748        reportCount30.setCount(0);
749
750        // 构建分享数29
751        QSBeanOutShareCount shareCount30 = new QSBeanOutShareCount();
752        shareCount30.setCount(0);
753
754        // 构建回复数29
755        QSBeanOutReplyCount replyCount30 = new QSBeanOutReplyCount();
756        replyCount30.setCount(0);
757
758        // 构建评论数29
759        QSBeanOutCommentCount commentCount30 = new QSBeanOutCommentCount();
760        commentCount30.setCount(0);
761
762        // 构建点赞数30
763        QSBeanOutLikeCount likeCount31 = new QSBeanOutLikeCount();
764        likeCount31.setCount(0);
765
766        // 构建收藏数30
767        QSBeanOutCollectCount collectCount31 = new QSBeanOutCollectCount();
768        collectCount31.setCount(0);
769
770        // 构建举报数30
771        QSBeanOutReportCount reportCount31 = new QSBeanOutReportCount();
772        reportCount31.setCount(0);
773
774        // 构建分享数30
775        QSBeanOutShareCount shareCount31 = new QSBeanOutShareCount();
776        shareCount31.setCount(0);
777
778        // 构建回复数30
779        QSBeanOutReplyCount replyCount31 = new QSBeanOutReplyCount();
780        replyCount31.setCount(0);
781
782        // 构建评论数30
783        QSBeanOutCommentCount commentCount31 = new QSBeanOutCommentCount();
784        commentCount31.setCount(0);
785
786        // 构建点赞数31
787        QSBeanOutLikeCount likeCount32 = new QSBeanOutLikeCount();
788        likeCount32.setCount(0);
789
790        // 构建收藏数31
791        QSBeanOutCollectCount collectCount32 = new QSBeanOutCollectCount();
792        collectCount32.setCount(0);
793
794        // 构建举报数31
795        QSBeanOutReportCount reportCount32 = new QSBeanOutReportCount();
796        reportCount32.setCount(0);
797
798        // 构建分享数31
799        QSBeanOutShareCount shareCount32 = new QSBeanOutShareCount();
800        shareCount32.setCount(0);
801
802        // 构建回复数31
803        QSBeanOutReplyCount replyCount32 = new QSBeanOutReplyCount();
804        replyCount32.setCount(0);
805
806        // 构建评论数31
807        QSBeanOutCommentCount commentCount32 = new QSBeanOutCommentCount();
808        commentCount32.setCount(0);
809
810        // 构建点赞数32
811        QSBeanOutLikeCount likeCount33 = new QSBeanOutLikeCount();
812        likeCount33.setCount(0);
813
814        // 构建收藏数32
815        QSBeanOutCollectCount collectCount33 = new QSBeanOutCollectCount();
816        collectCount33.setCount(0);
817
818        // 构建举报数32
819        QSBeanOutReportCount reportCount33 = new QSBeanOutReportCount();
820        reportCount33.setCount(0);
821
822        // 构建分享数32
823        QSBeanOutShareCount shareCount33 = new QSBeanOutShareCount();
824        shareCount33.setCount(0);
825
826        // 构建回复数32
827        QSBeanOutReplyCount replyCount33 = new QSBeanOutReplyCount();
828        replyCount33.setCount(0);
829
830        // 构建评论数32
831        QSBeanOutCommentCount commentCount33 = new QSBeanOutCommentCount();
832        commentCount33.setCount(0);
833
834        // 构建点赞数33
835        QSBeanOutLikeCount likeCount34 = new QSBeanOutLikeCount();
836        likeCount34.setCount(0);
837
838        // 构建收藏数33
839        QSBeanOutCollectCount collectCount34 = new QSBeanOutCollectCount();
840        collectCount34.setCount(0);
841
842        // 构建举报数33
843        QSBeanOutReportCount reportCount34 = new QSBeanOutReportCount();
844        reportCount34.setCount(0);
845
846        // 构建分享数33
847        QSBeanOutShareCount shareCount34 = new QSBeanOutShareCount();
848        shareCount34.setCount(0);
849
850        // 构建回复数33
851        QSBeanOutReplyCount replyCount34 = new QSBeanOutReplyCount();
852        replyCount34.setCount(0);
853
854        // 构建评论数33
855        QSBeanOutCommentCount commentCount34 = new QSBeanOutCommentCount();
856        commentCount34.setCount(0);
857
858        // 构建点赞数34
859        QSBeanOutLikeCount likeCount35 = new QSBeanOutLikeCount();
860        likeCount35.setCount(0);
861
862        // 构建收藏数34
863        QSBeanOutCollectCount collectCount35 = new QSBeanOutCollectCount();
864        collectCount35.setCount(0);
865
866        // 构建举报数34
8
```

```

18         options.add(question.getAnswer1Text() != null ?
question.getAnswer1Text() : "");
19         options.add(question.getAnswer2Text() != null ?
question.getAnswer2Text() : "");
20         options.add(question.getAnswer3Text() != null ?
question.getAnswer3Text() : "");
21         options.add(question.getAnswer4Text() != null ?
question.getAnswer4Text() : "");
22     bean.setOptions(options);
23
24     // 查找正确答案文本
25     bean.setAnswer(determineCorrectAnswer(question));
26     result.add(bean);
27 }
28
29     return result;
30 }
31
32 private static String determineCorrectAnswer(Question question) {
33     if (question.getAnswer1Correct() != null &&
question.getAnswer1Correct() {
34         return question.getAnswer1Text() != null ?
question.getAnswer1Text() : "";
35     }
36     if (question.getAnswer2Correct() != null &&
question.getAnswer2Correct() {
37         return question.getAnswer2Text() != null ?
question.getAnswer2Text() : "";
38     }
39     if (question.getAnswer3Correct() != null &&
question.getAnswer3Correct() {
40         return question.getAnswer3Text() != null ?
question.getAnswer3Text() : "";
41     }
42     if (question.getAnswer4Correct() != null &&
question.getAnswer4Correct() {
43         return question.getAnswer4Text() != null ?
question.getAnswer4Text() : "";
44     }
45     return ""; // 没有正确答案时返回空字符串
46 }

```

步骤5：分页查询测试；

如果出现错误，可以打开**MyBatis**的日志，输出错误信息到控制台：

[Day09-03. Mybatis-基础操作-删除\(预编译SQL\)](#)
哔哩哔哩bilibili

3、按照关键词查询

数据库预备知识：

要在用户名中查询某个特定的字符串，可以使用如下的sql模糊查询语句：

```
1 | SELECT * FROM user WHERE username Like '%abc%';
```

对应的MyBatis的mapper层的代码：

```
1  @Mapper
2  public interface Questionapper {
3      @Select("SELECT * FROM questions WHERE questionText LIKE CONCAT('%', #"
4      {keyword}, '%') AND isDelete=0")
5      List<Question> findByName(String keyword);
6  }
```

Controller中的代码：

```
1  @GetMapping("/findQuestion")
2  public Result getUser(String keyword){
3      List<qsBeanOutManage> qsBeanOut=questionService.findByName(keyword);
4      return Result.success(qsBeanOut);
5  }
```

Service层中的代码：

```
1 //接口;
2 public List<qsBeanOutManage> findByName(String keyword);
3
4 //实现:
5 public List<qsBeanOutManage> findByName(String keyword){
6     List<Question> userList=userMapper.findByName(keyword);
7
8     List<QSBeanOutManage>qsBeanOutManage =
9     Tools.convertToQSBeanManageList(questionList);
10    return qsBeanOutManage;
11 }
```