

测试输入数据://测试结束后请调用删除函数删除脏数据

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

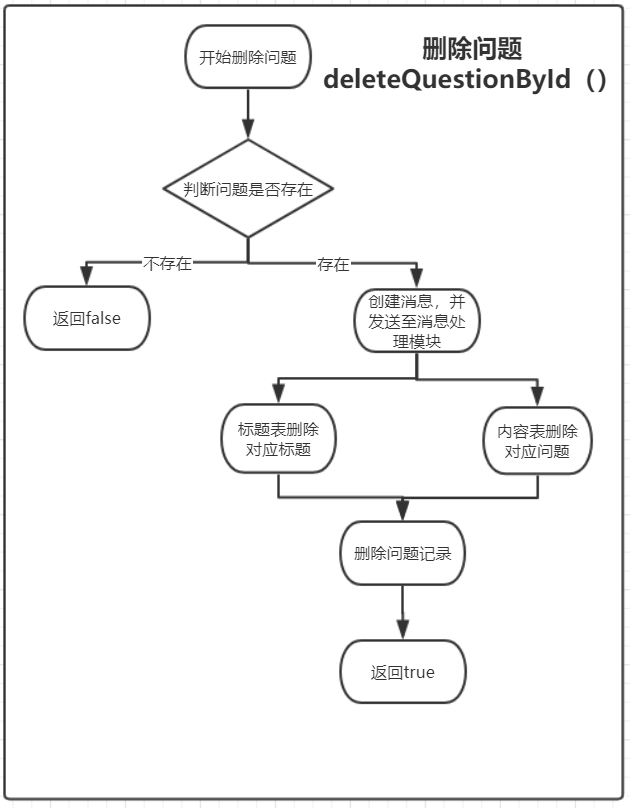
“authorId”://testMan1的id

“createTime”:date

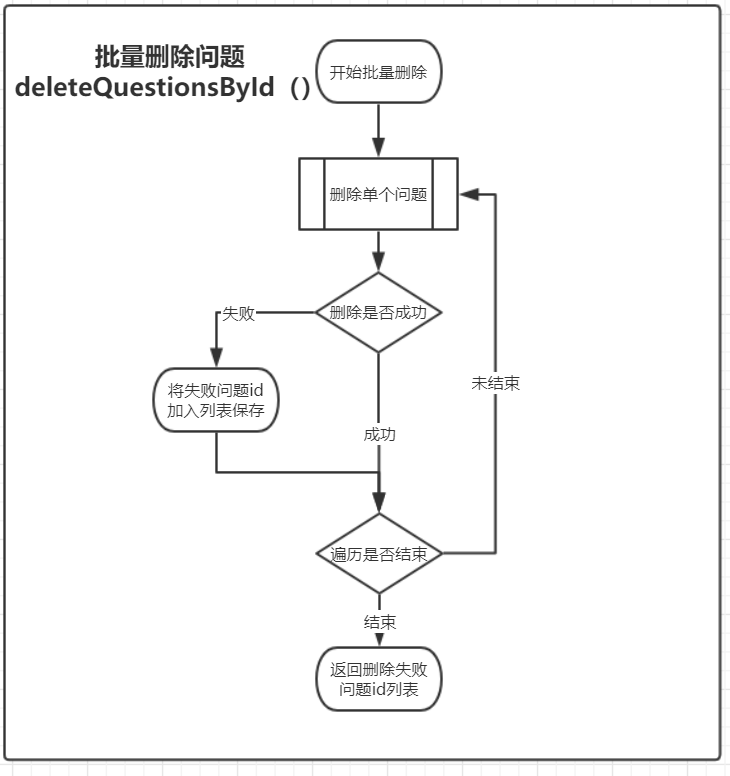
}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| addQuestion | 输入新增问题 | 数据库新增该问题的记录  创建作者提问数+1，积分获得增加  创作作者的消息数+1 | 与预期相符 | 正确 |



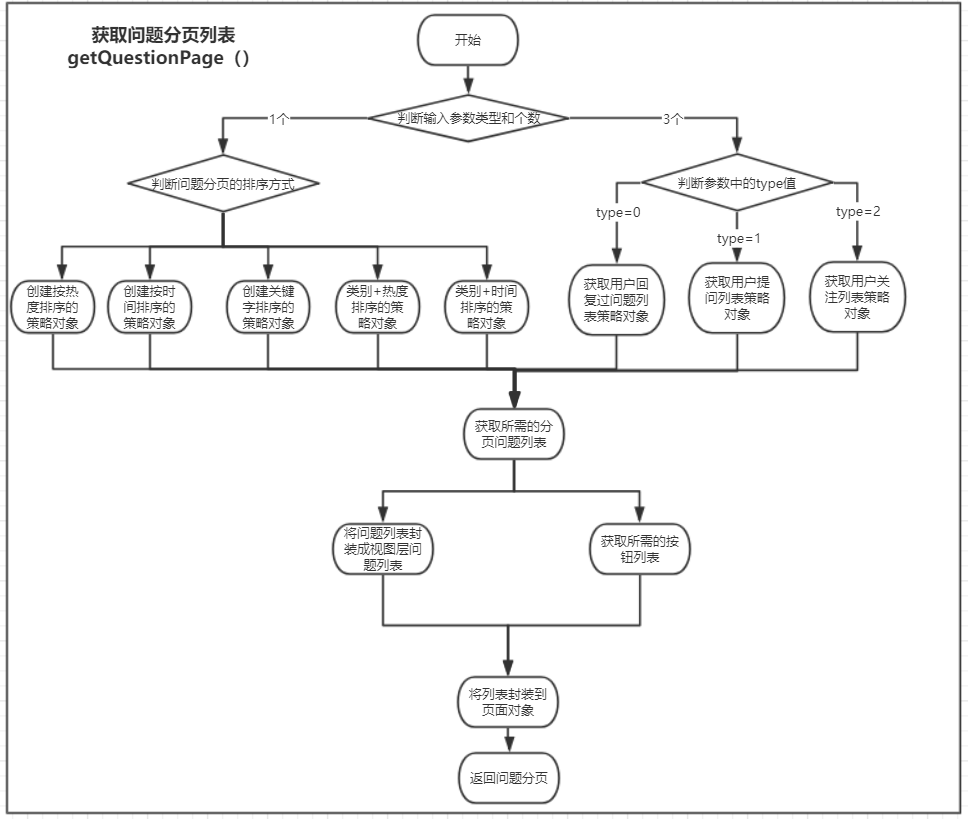
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| DeleteQuestionById | 删除一条已存在的问题记录 | 数据库中该条记录被删除  作者提问数-1  作者的消息数+1  函数返回true | 数据库对应记录被删除  作者提问数-1  作者消息数+1  函数返回true | 正确 |
| 删除一条不存在的问题记录 | 函数返回false | 函数返回false | 正确 |



测试前数据库填充数据：

调用addQuestion方法添加10条作者为test1用户的问题记录，并记录id序列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| deleteQuestionById | 删除新增问题id序列的问题  （全对） | 数据库中id序列的问被删除;作者问题数-10;返回失败列表长度为0  消息数+10 | 符合预期 | 正确 |
| 正确删除问题后重新再删除一遍上述的id列表（全错） | 返回失败列表长度为10，列表元素为传入的10个元素  用户问题数不变  消息数不变 | 符合预期 | 正确 |
|  | 新增10个问题，将id加入删除序列，再填加5个不存在问题的id到序列（部分对） | 正确的id问题被删除  失败列表长度为5，且为刚添加的不存在的id列表  作者问题数-10  作者消息数+10 | 符合预期 | 正确 |



测试输入数据://测试结束后请调用删除函数删除脏数据

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”:10,

“createTime”:”当前日期”

“kindId”:0,

“anoymous”:0,

“responseNum”:700

}

}，

{

“title”:”testTitle2”,

“content”:”testContent1”

“question”:{

“authorId”:10,

“createTime”:”明天”,

“kindId”:1,

“anoymous”:0,

“responseNum”:100

}

}，

{

“title”:”test3”,

“content”:”testContent1”

“question”:{

“authorId”:10,

“createTime”:”后天”,

“kindId”:0,

“anoymous”:1,

“responseNum”:500

}

}，

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

//调用ResponseService.insertResponse(ResponseVO responseVO)添加回复

{

“response”:{

“questionId”://问题testTitle1的id

“authorId”://testMan1的id

“createTime”:”2020-10-02”

},

“content”:”responseContent1”

},

{

“response”:{

“questionId”://问题testTitle2的id

“authorId”://testMan1的id

“createTime”:”2020-10-03”,

},

“content”:”responseContent2”

}

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

//调用AttentionService.insertAttention(Attention attention)插入关注记录

{

“userId”://testMan1的id,

“question”://testTiltle2的id,

“createTime”:”2020-03-12”,

“flag”:1

}

{

“userId”://testMan1的id,

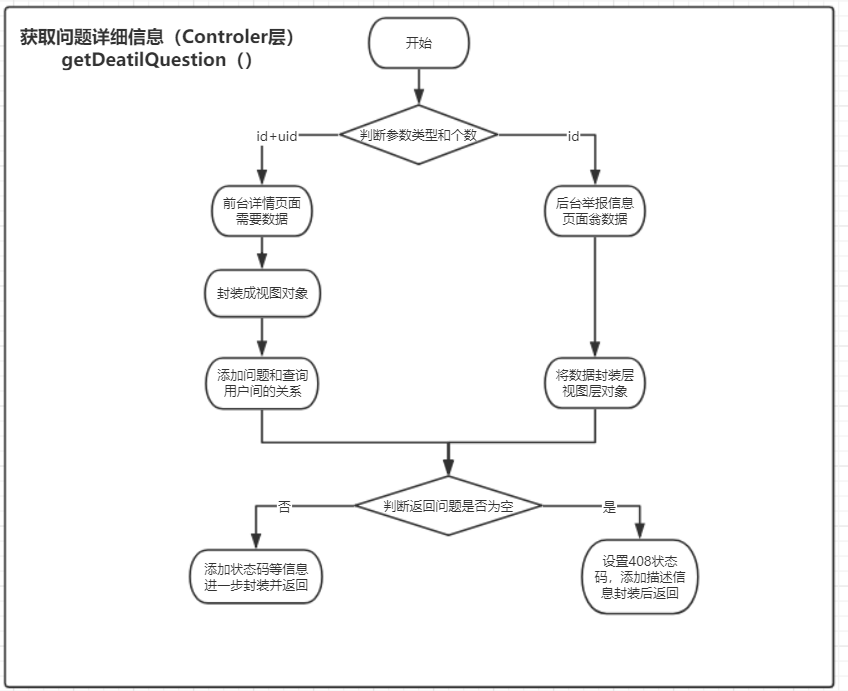
“question”://testTiltle3的id,

“createTime”:”2020-03-12”,

“flag”:1

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| getQuestionPage  (QuestionPage) | 按热度排序 | 获取列表的回复数按递减排列 | 符合预期 | 正确 |
| 按时间排序 | 返回列表顺序按时间由近到远排序 | 符合预期 | 正确 |
| 按关键字查询 | 返回问题序列的标题包含查询关键字 | 符合预期 | 正确 |
| 按类别0+热度排序 | 返回类别为0的问题序列并按热度排序 | 符合预期 | 正确 |
| 按类别1+时间排序 | 返回类别为1的问题序列并按时间由近到远排序 | 符合预期 | 正确 |
| getQuestionPage  (Integer userId, QuestionPage page, int type) | 查看用户回复过的问题列表(type=0) | 返回问题序列，数据库存在用户对列表中包含问题的回复记录 | 符合预期 | 正确 |
| 查看用户提问列表(type=1) | 返回问题序列，序列中包含问题作者为测试用户 | 符合预期 | 正确 |
| 获取用户关注问题列表(type=2) | 返回问题序列存在对应的关注记录，同时记录标志位为1 | 符合预期 | 正确 |



测试输入数据://测试结束后请调用删除函数删除脏数据

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”:10,

“createTime”:”2020-01-01”

“kindId”:0,

“anoymous”:0,

“responseNum”:10

}

}，

---------------------------------------------------------------------------------------------------------------------------------//调用AttentionService.insertAttention(Attention attention)插入关注记录

{

“userId”://testMan1的id,

“question”://testTiltle2的id,

“createTime”:”2020-03-12”,

“flag”:1

}

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

//调用AttentionService.insertAttention(Attention attention)插入关注记录

{

“reportorId”://举报人testMan1的id,

“questionId”://testTiltle1的id,

“createTime”:”2020-03-12”,

“flag”:1

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| getQuestionById | 查询存在的问题 | 返回问题的数据库实体数据 | 符合预期 | 正确 |
| 查询不存在的问题 | 返回null | 符合预期 | 正确 |
| addRelationToUId | 添加查询问题与查者之间的关系（没关注，有投诉） | questionVO.doesAttention=false  questionVO.doesReported=true | 符合预期 | 之前 |
| 添加查询问题与查者之间的关系（有关注，没投诉） | questionVO.doesAttention=true  questionVO.doesReported=false | 符合预期 | 正确 |