

测试输入数据://测试结束后请调用删除函数删除脏数据

//调用questionService.addQuestion(QuestionVO questionVO)添加回复所属问题

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”://testMan1的id

“createTime”:date

}

}

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

//调用responseService.insertResposne(ResponseVO responseVO)创建点赞回复

//数据库中没有记录的情况

{

“response”:{

“questionId”://上述testTitle1的id,

“authorId”://testMan1的id

“createTime”:”2020-02-01”

}

}

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

//调用LikeService.insertLikeInfo(Likes like)创建点赞记录

{

“userId”://testMan1的id,

“responseId”://上述回复id,

“flag”:1

}

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

//调用LikeService.insertLikeInfo(Likes like)取消点赞记录

{

“userId”://testMan1的id,

“responseId”://上述回复id,

“flag”:1

}

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

//调用responseService.insertResposne(ResponseVO responseVO)创建点赞回复

//数据库中有记录的情况

{

“response”:{

“questionId”://上述testTitle1的id,

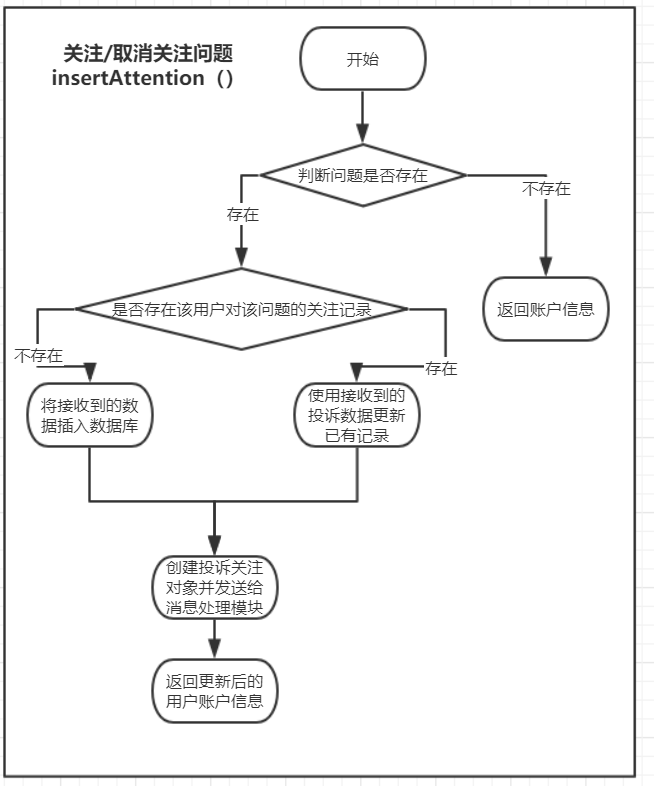
“authorId”://testMan1的id

“createTime”:”2020-02-01”

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| insertLikeInfo | 数据库没有测试用户关于该问题的点赞记录时点赞 | 成功添加点赞记录  该回复的点赞数+1  被点赞用户积分增加对应分数  返回结果true | 成功添加点赞记录  该回复的点赞数+1  被点赞用户积分增加对应分数  返回结果true | 正确 |
| 数据库存在测试用户关于该问题的点赞记录时取消点赞 | 点赞记录标志位被设为0  该条回复点赞数-1  回复作者扣除相应积分  返回结果true | 点赞记录标志位被设为0  该条回复点赞数-1  回复作者扣除相应积分  返回结果true | 正确 |
| 点赞的回复不存在 | 返回结果false | 返回结果false | 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

//调用questionService.addQuestion(QuestionVO questionVO)添加要关注的问题

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”://testMan1的id

“createTime”:date

}

}

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

//创建关注消息（当前数据库没有本测试用户对该问题的关注记录）

{

“userId”://testMan1的id,

“questionId”://上述创建问题的id,

“createTime”:”2020-02-21”,

“flag”:1

}

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

//取消关注消息（当前数据库有本测试用户对该问题的关注记录）

{

“userId”://testMan1的id,

“questionId”://上述创建问题的id,

“createTime”:”2020-02-21”,

“flag”:0

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| InsertAttention | 创建对某个问题的关注（数据库不存在测试用户对该问题的关注记录） | 数据库创建该关注记录  用户关注数+1 | 数据库成功创建该关注记录  用户关注数+1 | 正确 |
| 取消对某个问题的关注（数据库存在测试用户对该问题的关注记录） | 数据库记录的标志位设为0  用户关注数-1 | 数据库记录的标志位设为0  用户关注数-1 | 正确 |
| 创建对某个问题的关注，数据库不存在该问题 | 用户关注数不变 | 用户关注数不变 | 正确 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| getAttentionQuestionList | 获取某个用户所有关注的问题的列表（该用户存在） | 获取的问题ID与上述插入的问题ID一致 | 获取的问题ID与上述插入的问题ID一致 | 正确 |
| 获取某个用户所有关注的问题的列表（该用户不存在） | 获取的问题ID列表长度为0 | 获取的问题ID列表长度为0 | 正确 |