

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)添加要回复的问题

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”://testMan1的id

“createTime”:date

}

}

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

//调用ResponseService.insertResponse(ResponseVO responseVO)创建回复消息

{

“response”:{

“questionId”://上述问题的id,

“authorId”://testMan的id,

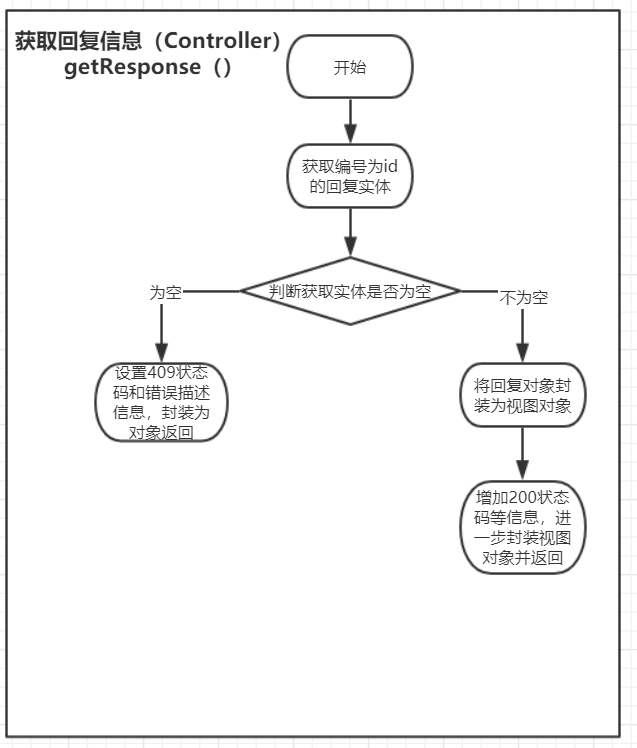
“createTime”:”2020-03-21”,

},

“content”:”responseTestContent1”

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| insertResponse | 添加对某个问题的回复 | 数据库增加该问的回复记录  对应问题的回复数+1  回复者的回复数+1，积分、经验值增加对应分数  返回true | 数据库成功增加该问的回复记录  对应问题的回复数没变  回复者的回复数+1，积分、经验值增加对应分数  返回true | 错误 |
| 回复的问题不存在 | 返回false | 返回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)添加要回复的问题

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”://testMan1的id

“createTime”:date

}

}

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

//调用ResponseService.insertResponse(ResponseVO responseVO)创建回复消息

{

“response”:{

“questionId”://上述问题的id,

“authorId”://testMan的id,

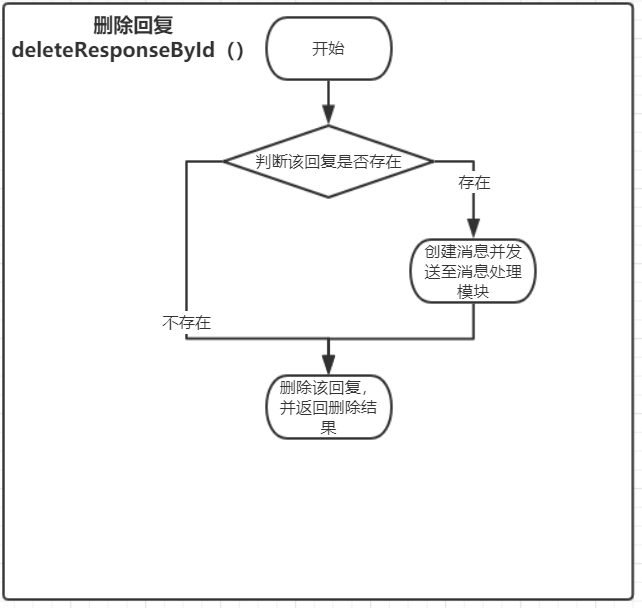
“createTime”:”2020-03-21”,

},

“content”:”responseTestContent1”

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| getResponseById | 获取存在的回复 | 成功获取对应回复记录 | 成功获取对应回复记录 | 正确 |
| 获取不存在回复 | 返回null | 返回null | 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)添加要回复的问题

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”://testMan1的id

“createTime”:date

}

}

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

//调用ResponseService.insertResponse(ResponseVO responseVO)创建回复消息

{

“response”:{

“questionId”://上述问题的id,

“authorId”://testMan的id,

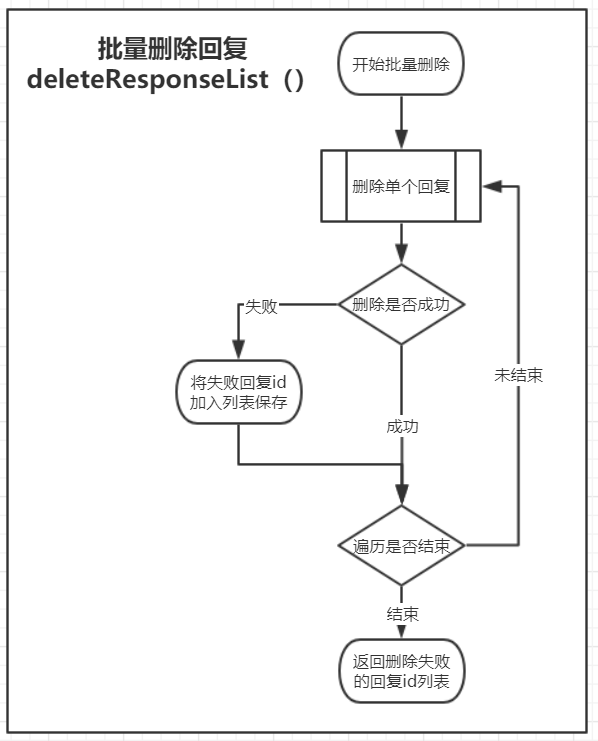
“createTime”:”2020-03-21”,

},

“content”:”responseTestContent1”

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| deleteResponseById | 删除一条存在的回复 | 成功删除记录  问题回复数-1  作者回复数-1  函数返回1 | 成功删除记录  问题回复数没变  作者回复数-1  函数返回1 | 错误 |
| 删除一条不存在的回复 | 函数返回0 | 函数返回0 | 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)添加要回复的问题

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”://testMan1的id

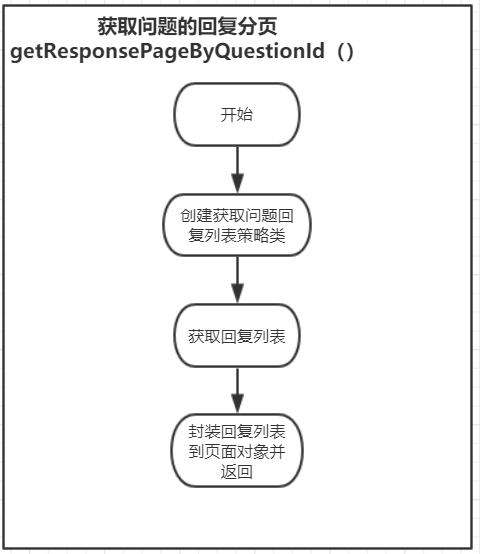
“createTime”:date

}

}

通过insertResponse循环添加10条对上述问题的回复

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试 |
| deleteResponseList | 删除列表中的回复都存在 | 列表中的回复均被删除  测试用户回复数减去对应的删除数  函数返回列表长度为0 | 列表中的回复均被删除  测试用户回复数减去对应的删除数  函数返回列表长度为0 | 正确 |
| 删除列表中含有不存在的回复 | 存在的回复都被删除  测试用户回复-减去对应的删除数  函数返回列表包含不存在的回复的ID | 存在的回复都被删除  测试用户回复-减去对应的删除数  函数返回列表包含不存在的回复的ID | 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)添加要回复的问题

{

“title”:”testTitle1”,

“content”:”testContent1”

“question”:{

“authorId”://testMan1的id

“createTime”:date

}

}

通过insertResponse循环添加10条对上述问题的回复

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| getResponsePageByQuestionId() | 获取上述问题的回复列表 | 获取回复列表并按点赞数排序 | 成功获取回复列表并按点赞数排序 | 正确 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| getResponsePageBeReported() | 获取被举报的所有回复列表 | 获取所有被举报的回复列表并按举报数排序 | 获取所有被举报的回复列表并按举报数排序 | 正确 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| convertToVO() | 匿名回复 | 回复是否匿名成功 | 回复是否匿名成功 | 正确 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试函数 | 测试描述 | 预期结果 | 实际结果 | 测试评价 |
| addListRelationToUid() | 给回复列表添加用户的关联信息 | 回复被点赞一次后点赞数+1  回复被举报后变为被举报状态 | 回复被点赞一次后点赞数+1  回复被举报后变为被举报状态 | 正确 |