目录

[头条评论设计文档 2](#_Toc480449623)

[1. 项目背景 2](#_Toc480449624)

[2. 项目责任人 2](#_Toc480449625)

[3. 术语定义 2](#_Toc480449626)

[4. 业务模块划分及各模块介绍 2](#_Toc480449627)

[5. 数据结构 2](#_Toc480449628)

[5.1. 表结构 2](#_Toc480449629)

[5.2. Cache使用方案 3](#_Toc480449630)

[5.3. MQ使用方案 3](#_Toc480449631)

[6. 对上下游业务的影响及潜在风险 4](#_Toc480449632)

[7. 性能效率,访问量预估, 资源申请 4](#_Toc480449633)

[8. 出错处理/容灾 4](#_Toc480449634)

[9. 安全问题 5](#_Toc480449635)

# 头条评论设计文档

## 项目背景

## 项目责任人

## 术语定义

## 业务模块划分及各模块介绍

## 数据结构

### 表结构

**Discuz结构：**

CREATE TABLE pre\_forum\_postcomment (

id int(10) unsigned NOT NULL AUTO\_INCREMENT,

tid mediumint(8) unsigned NOT NULL DEFAULT '0',

pid int(10) unsigned NOT NULL DEFAULT '0',

author varchar(15) NOT NULL DEFAULT '',

authorid mediumint(8) NOT NULL DEFAULT '0',

dateline int(10) unsigned NOT NULL DEFAULT '0',

`comment` varchar(255) NOT NULL DEFAULT '',

score tinyint(1) NOT NULL DEFAULT '0',

useip varchar(15) NOT NULL DEFAULT '',

`port` smallint(6) unsigned NOT NULL DEFAULT '0',

rpid int(10) unsigned NOT NULL DEFAULT '0',

PRIMARY KEY (id),

KEY tid (tid),

KEY authorid (authorid),

KEY score (score),

KEY rpid (rpid),

KEY pid (pid,dateline)

) TYPE=MyISAM;

**评论展示侧**

头条展示只分为两层： 首层评论以及首层评论的评论。

comments表：

1. {
2. "key": "文章ID",
3. "itemKey": "评论ID",
4. "item": {
5. "content": "评论评论评论",
6. "hotSubCommentIds": [0, 1],
7. "user": {
8. "userId": 0,
9. "userName": "",
10. "headImg": ""
11. },
12. "likeNum": 0,
13. "unlikeNum": 0,
14. "createTime": 0,
15. "modifiedTime": 0,
16. "parentId": 0,
17. "ancestorId": 0,
18. "level": 0,
19. "isdel": 0
20. }
21. }

**谁点赞：**

likes与unlikes表, 不推荐做这个功能, 数据无法支撑, 只能用mysql存储

1. {
2. "key": "文章ID",
3. "itemKey": "评论ID",
4. "item": {
5. "likes": [0, 1],
6. "unlikes": [3, 4]
7. }
8. }

**用户侧**

展示曾经发过的评论, 应该是按照时间倒叙来展示

1. {
2. "key": "用户",
3. "itemKey": "资讯ID\_评论ID",
4. "item": {
5. "newsID": 0,
6. "commentID": 0,
7. "commentAbstract": "评论摘要",
8. "createTime": 0
9. }
10. }

点过赞的列表,在进入评论页时使用,显示点赞过的评论

1. {
2. "key": "用户",
3. "itemKey": "资讯ID",
4. "commentIDs": [0, 1]
5. }

**消息通知:**

应该是调用58APP现有接口推送.

**我的收藏**

1. {
2. "key": "用户ID",
3. "itemKey": "资讯ID",
4. "list": {
5. "articleID": 0,
6. "articleTitle": "title",
7. "imgs": ["URL1", "URL2"]
8. }
9. }

### MySQL使用

在全文检索的使用场景下, 由于可用性原因,不应使用MySQL作为存储介质, 而应选择ES等.

### Cache使用方案

前期流量小时不需使用缓存(WList有系统缓存).

后期流量大时, 可以使用redis构建分页缓存.

### MQ使用方案

为了减轻系统压力, 增加/删除评论操作时, 不直接调用后端接口, 而是采用发送消息, 系统接收消息再进行增加/删除评论操作.

增加/删除评论失败时, 可以利用消息重发来达到失败重试的目的, 保证数据最终一致性.

#### 使用组件

ESB-FurtherMQ

#### 消息详情

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 消息名称 | Producer | Consumer |
| 1 | 增加评论 |  |  |
| 2 | 删除评论 |  |  |

#### 使用场所

增加删除评论

评论队列

数据库

## 对上下游业务的影响及潜在风险

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 业务方及模块 | 影响/风险 | 相关模块负责人 |
| 1 | 资讯主体 | 评论数/点赞数的数据一致性 |  |

## 性能效率,访问量预估, 资源申请

## 出错处理/容灾

## 安全问题

1. xss攻击
2. csrf攻击
3. 不符合法律法规内容过滤