Echo 的评论是如何显示的

原创 飞天小牛肉 [飞天小牛肉](javascript:void(0);)

**飞天小牛肉**

微信号 CS-Wiki

功能介绍 InfoQ & 阿里云签约作者，分享原创技术干货和成长经验

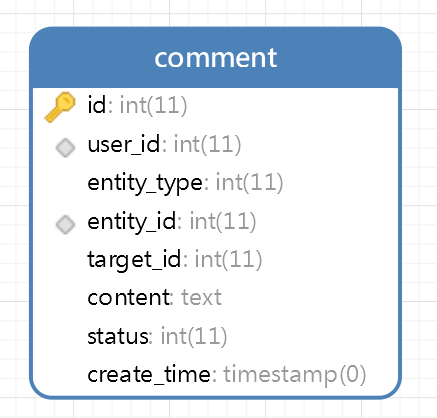
2021-04-27[原文](https://mp.weixin.qq.com/s?__biz=MzI0NDc3ODE5OQ==&mid=2247486703&idx=2&sn=8075851ea6f75e34019e2707adda4f53&chksm=e959d957de2e5041818f499010545fb2a0f7e24124f5de81e3ca67cbefdf06e44d44065d023c&scene=178&cur_album_id=1744497649518493697#rd) 发表于

收录于合集 #Echo 学习教程 18个

关于评论模块需要注意的就是评论表 comment 的设计，这个表应该是相对来说最复杂的一张了。因为不仅有评论（对帖子的评论），还有对评论的回复，都放在这一张表里面了。把握其中字段的含义，尤其是 entity\_id 这个字段，才能透彻了解这个功能的逻辑。



先来回顾一下 comment 表：



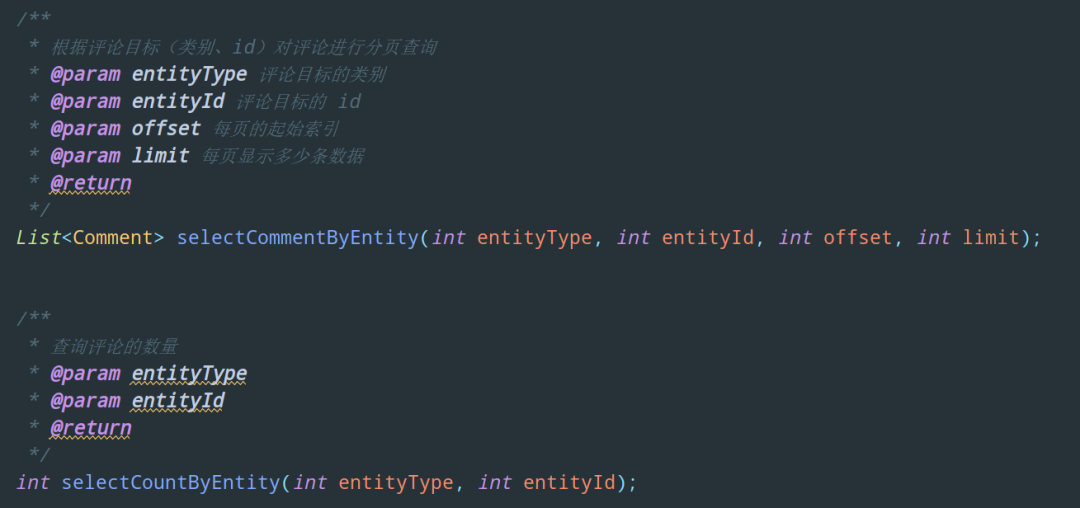
* id：评论/回复的唯一标识
* user\_id：用户 id（哪个用户发布了这个评论/回复）
* entity\_type：实体类型（表示这条 comment 是针对哪个类型的，如果是针对帖子的，那么这个 comment 就是评论；如果是针对评论的，那么这条 comment 就是回复）
* entity\_id：实体的 id（如果是对帖子的评论，就存储帖子的 id；如果是对评论的回复，就存储评论的 id；还有对回复的回复，存储的仍然是所属评论的 id。也就是说，**某个帖子下的所有评论，它们的 entity\_id 都是这个帖子的 id。某条评论下的所有回复，它们的 entity\_id 都是这条评论的 id**。）
* target\_id：目标用户 id（表示这条评论/回复是针对哪个用户的。比如用户 admin 发了一个帖子，用户 master 评论了这个帖子，那么这里的 target\_id 存储的就是用户 admin 的 id。）
* content：评论/回复的内容
* status：评论/回复状态
  + 0 - 正常（默认）
  + 1 - 禁用（这个状态截止目前还没有使用，各位可以自行扩展哈）
* create\_time：评论/回复发布时间

**Dao 层**

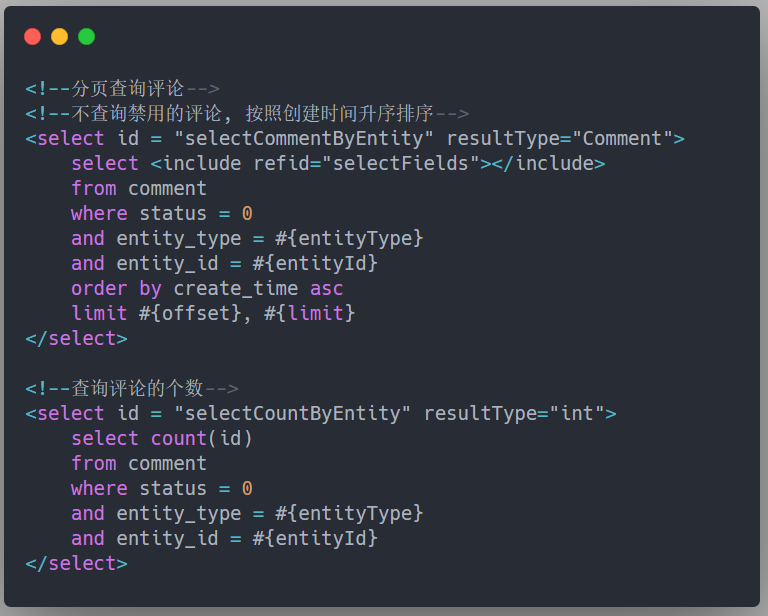
关于评论的 Dao 层接口定义在 CommentMapper 中。

解释下下图中 selectCommentByEntity 这个方法，和帖子分页差不多，它的功能就是根据 entityType 和 entityId 来分页查询评论。更通俗来说，这个方法可以通过 entityType 来指定是查询评论还是查询回复，如果是查询评论，可以继续通过 entityId 来指定查询针对哪个帖子的评论；如果是查询回复，同样也是通过 entityId 来继续指定查询针对哪条评论的回复。

看到这里想必大家也就明朗了，**通过 entityType 和 entityId 这个组合我们就可以唯一的指定一条评论或者回复**。



来看这俩接口的具体实现：



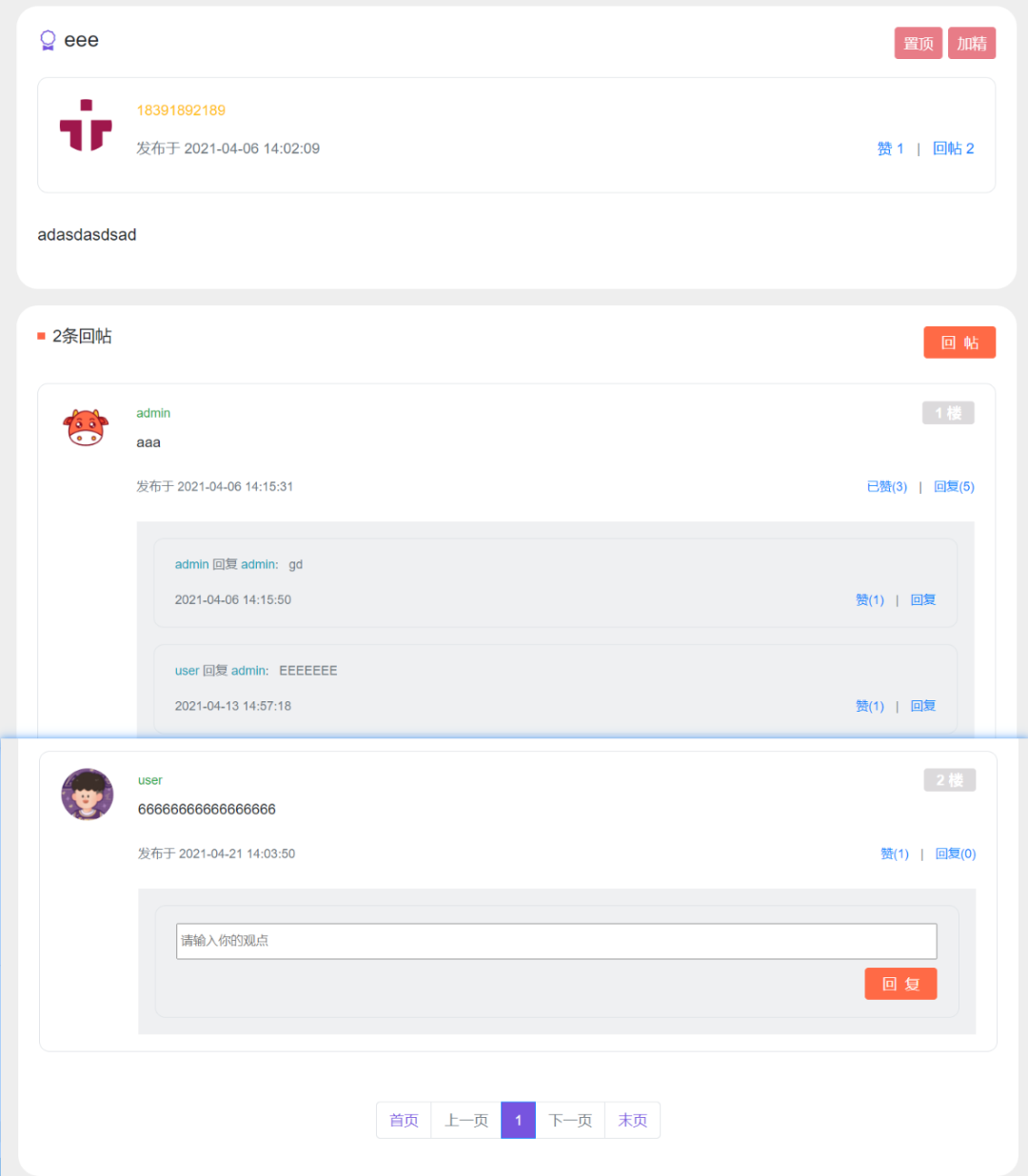
这个没啥好说的吧，各位直接看代码就行。

另外，这里的 Service 层也比较简单，就是调用了一下 Dao，我就不再多费口舌了。

**表现层**

评论与回复是在哪里显示的呢？帖子的详情页里面，对吧。

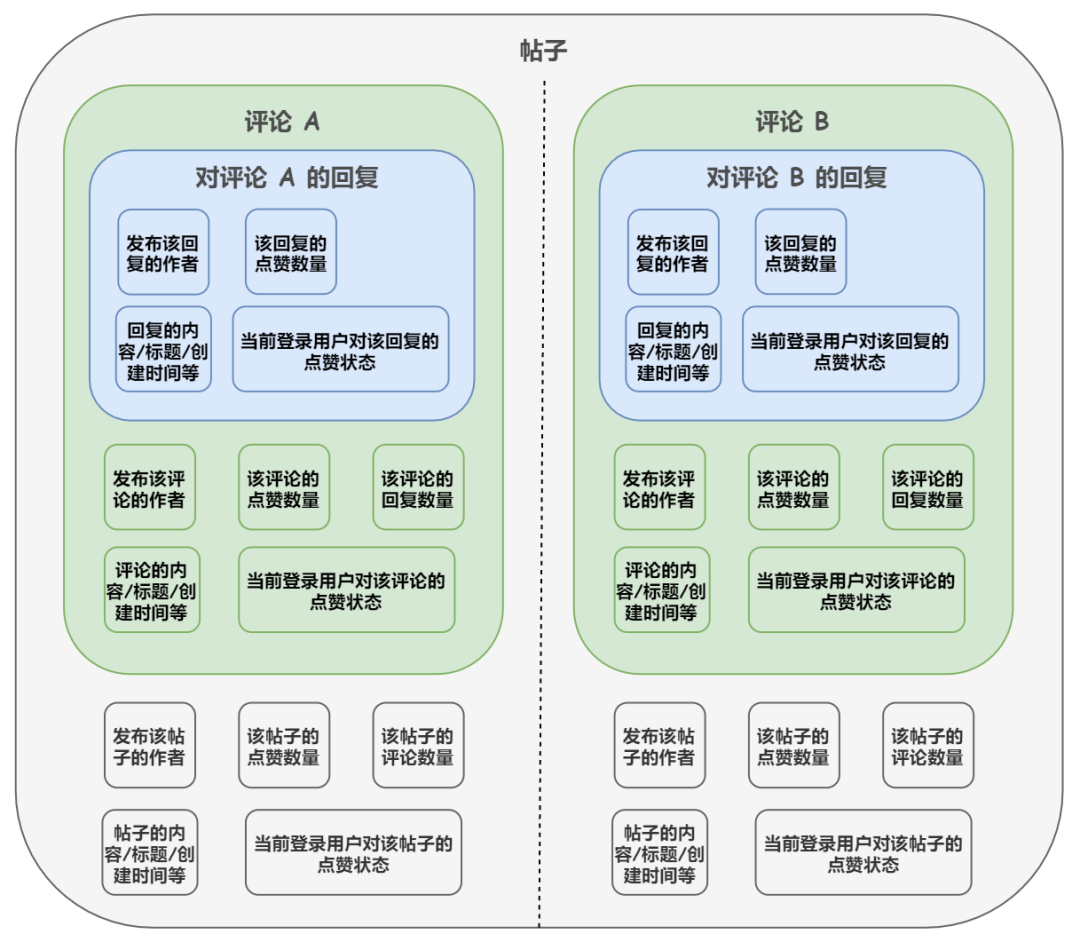
一篇帖子的详情页需要哪些东西呢？我们来看看



还是很清楚的，我们需要后端传过来的数据有如下这些：

* 帖子的相关信息：包括内容、标题、发帖时间、以及作者信息、点赞数量、评论数量、当前登录用户对该帖子的点赞状态等
* 评论的相关信息：包括内容、发布该评论的时间、以及发布该评论的作者信息、点赞数量、回复数量、回复的相关信息、当前登录用户对该评论的点赞状态等
* 回复的相关信息被封装在评论的相关信息里面，既然回复和评论我们都放在一张表里面了，那自然不用多说，它和显示评论所需要的信息是一样的。

文字叙述还是略显生涩，各位直接看图吧，一个帖子的详情页需要封装的信息大概如下：



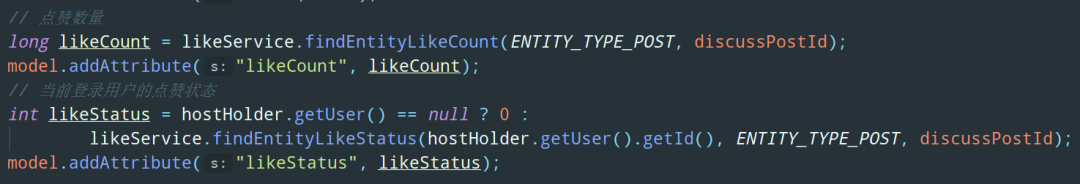
img

这段代码放在 DiscussPostController 中的 getDiscussPost 方法里面，代码非常长，很容易懵逼，我就不一整个方法全部截下来了，按照我们上面所说的需要的信息，一段一段的给大家展示出来。

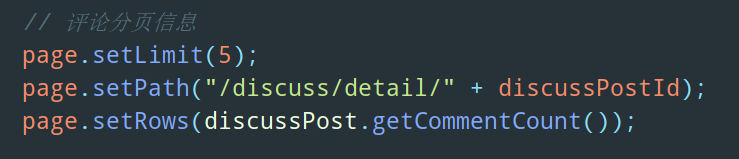
首先，封装帖子及其作者的相关信息：



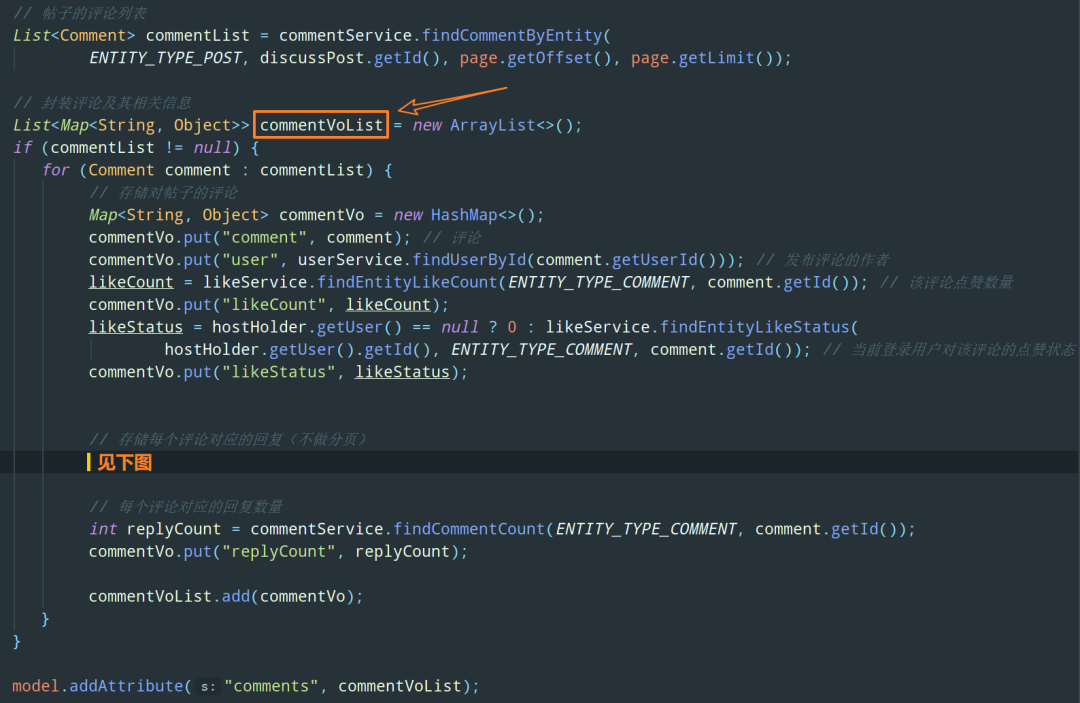
封装点赞相关信息：



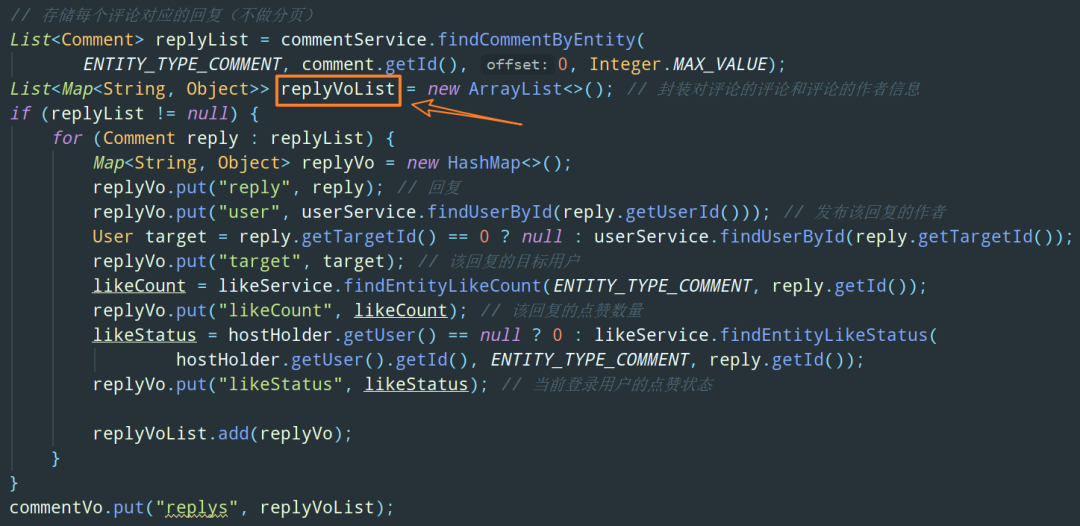
封装评论的分页信息，这里就显示出我们分页模型的强大了，一套代码随处用，不了解的各位强烈推荐回看上篇文章 [**Echo 的帖子列表与分页是怎么做的**](https://mp.weixin.qq.com/s?__biz=MzI0NDc3ODE5OQ==&mid=2247486553&idx=2&sn=21c7b77406e261e21814255109bacdc2&scene=21#wechat_redirect)



封装评论的相关信息，这里为了各位看的清楚，我把回复相关的信息暂时去掉了：



在评论的相关信息里面封装回复的相关信息：



这样，帖子详情页前端所需要的所有信息，我们已准备就绪。

总结下我们放进 Model 里的信息：

1）post：帖子相关信息

2）user：发帖作者相关信息（包含昵称 username、头像地址 headerUrl 等，详见 User 类）

3）likeCount：该帖的点赞数量

4）likeStatus：当前登录用户对该帖的点赞状态

5）replyCount：每个评论对应的回复数量

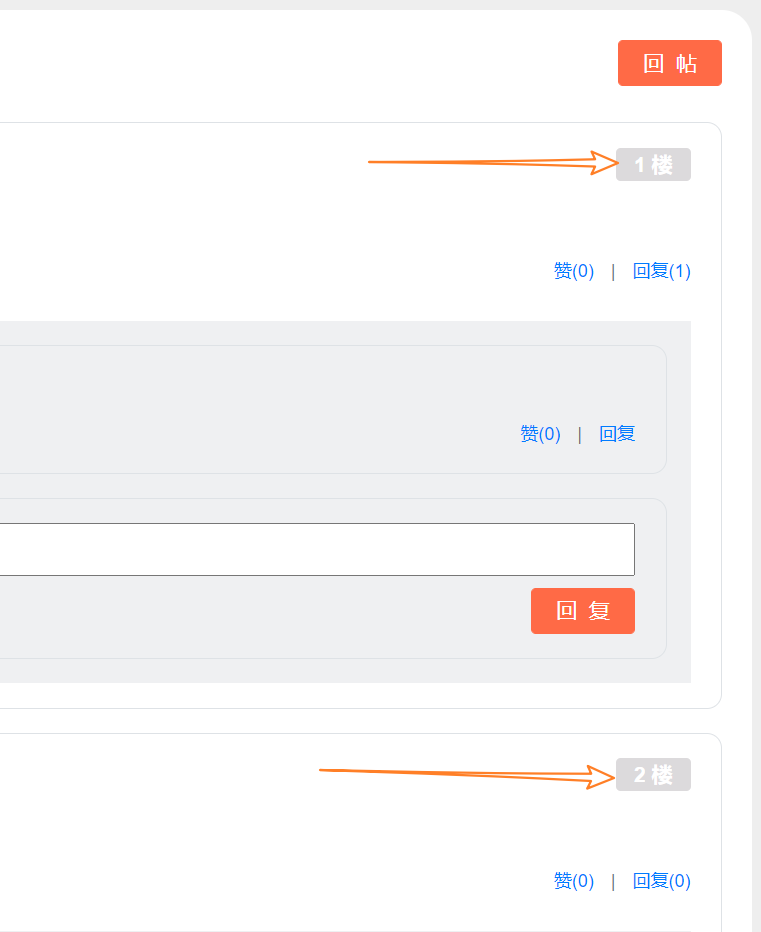
6）comments：该贴的所有评论及其相关信息

* comment：评论（包含内容 content、发布时间 createTime 等，详见 Comment 类）
* user：发布评论的作者相关信息（包含昵称 username、头像地址 headerUrl 等，详见 User 类）
* likeCount：每个评论的点赞数量
* likeStatus：当前登录用户对每个评论的点赞状态
* replys：每个评论对应的所有回复信息
  + reply：（包含内容 content、发布时间 createTime 等，详见 Comment 类）
  + user：发布回复的作者相关信息（包含昵称 username、头像地址 headerUrl 等，详见 User 类）
  + target：该回复的目标用户（这个回复必然是针对某条评论的，那么这条评论是哪个用户发布的呢，这里的 target 就是这个用户）
  + likeCount：每个回复的点赞数量
  + likeStatus：当前登录用户对每个回复的点赞状态

OK，接下来做的事情就是去前端取出这些数据就行了。这里我就不再详细说了，简单截几个代码段大家看看：



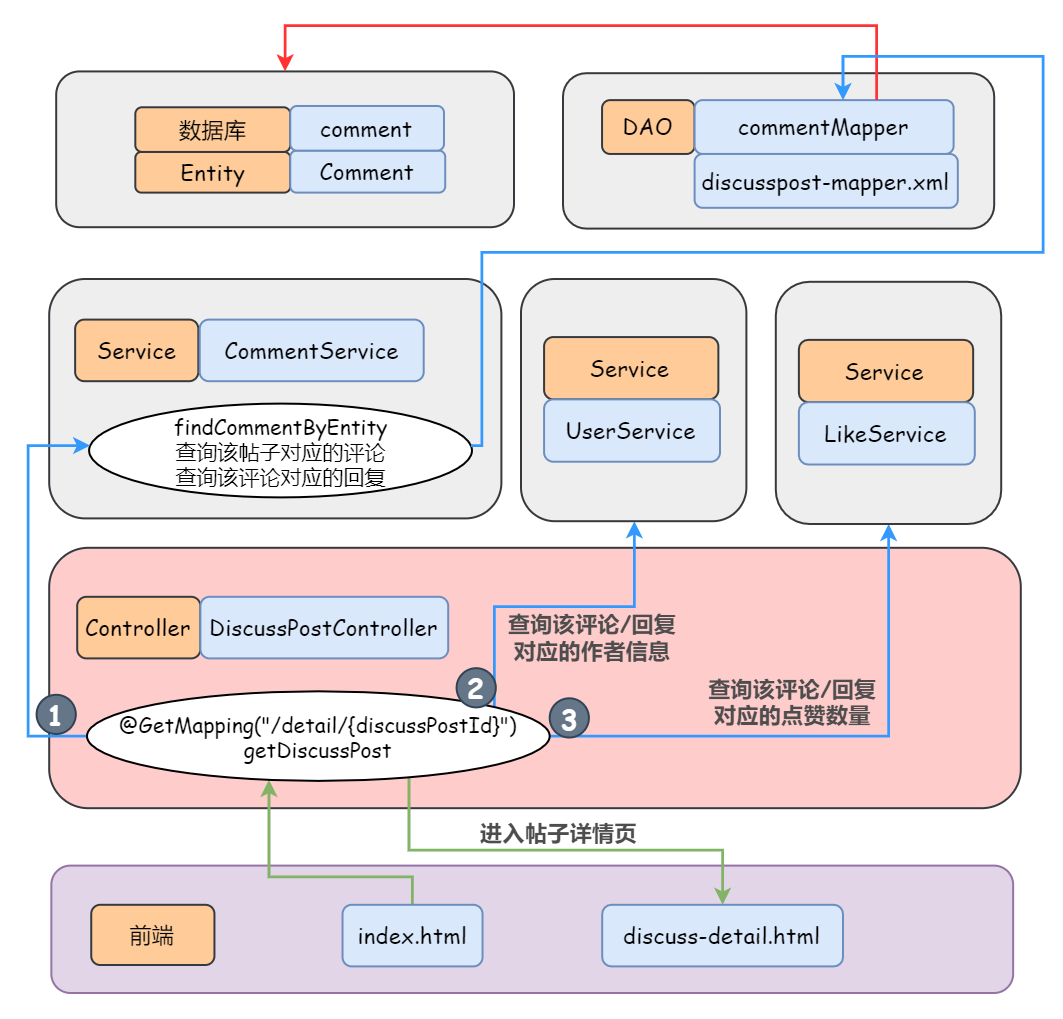
对了，这里多提一嘴，虽然不是什么值得注意的地方，各位应该发现了我们的评论记录了楼层数：



关于这个的实现其实就是在我们循环变量名的后面加上 Stat，比如 cvoStat，这是一种固定表达，可用于表示每次的循环对象，而 xxxStat.count 就表示当前是第几次循环，也就是我们的楼层数。

**小结**

这部分的业务逻辑其实没啥难的，就是字段太多容易让人懵逼，整体的逻辑各位可以看下图再回顾下：



### 精选留言

暂无...