

目录

目录

1. 讨论grpc使用注意事项
2. 本次讨论修改影响涉及直接UI模块（其他模块引入该体系需要注意的范围）
3. 数据结构约定及一些注意事项
 1. 枚举DiscussionSourceType
 2. 发送评论中的message特殊内容拼接
 3. DiscussionProvider.cs中GetInstance需添加新讨论库的配置
 4. 如果是已有讨论整合进来，需修自行改数据库实体
 5. 返回前端的实体约定
4. 接口逻辑介绍
 1. 添加新讨论
 2. 获取讨论列表(组合获取关于自己的讨论计数)
 3. 获取指定单条讨论
 4. 获取讨论计数（暂时没有独用，考虑到以后计数单独显示，所以先单独暴露出该接口）
 5. 删除讨论

1. 讨论grpc使用注意事项

划重点：目前任务标记完成与否的切换中会修改node计数，项目在标记归档与否时亦是如此，不过这个node计数修改目前 没有抽出来，如果新的讨论模块需要类似的逻辑，另行整合

2. 本次讨论修改影响涉及直接UI模块（其他模块引入该体系需要注意的范围）

- 1：web端讨论增删改查（任务和项目重写，动态中需要修改message为新格式，以及对老格式的兼容）
- 2：移动端api讨论增删改查（任务和项目兼容旧格式）
- 3：web端inbox消息提醒（任务和项目重写）
- 4：移动端api消息MessageController（任务和项目兼容旧格式）

3. 数据结构约定及一些注意事项

1. 枚举DiscussionSourceType

（位置：主站MD.Core=>MD.Enum下），目前有如下所示类型，已有如下枚举类型，如不够需自行添加

```

/// <summary>
/// 讨论源类型
/// </summary>
public enum DiscussionSourceType
{
    //
    // 摘要:
    //      * 任务
    [Description("任务")]
    Task = 1,
    //
    // 摘要:
    //      * 项目
    [Description("项目")]
    Folder = 2,
    //
    // 摘要:
    //      * 日程
    [Description("日程")]
    Calendar = 3,
    //
    // 摘要:
    //      * 知识
    [Description("知识")]
    Knowledge = 4,
    //
    // 摘要:
    //      * 审批
    [Description("审批")]
    Approval = 5,
    //
    // 摘要:
    //      * 考勤
    [Description("考勤")]
    Attendance = 6
}

```

2. 发送评论中的message特殊内容拼接

解释文本: @穆晓 @任务全体成员
code文本: [aid]bdc9ef24-94c5-49f7-81d8-067243d20a4f[/aid] [all]Task[/all]

其中（@全体成员）中的英文字符为上述枚举的英文字符串

3. DiscussionProvider.cs中GetInstance需添加新讨论库的配置

```

public static DiscussionProvider GetInstance(DiscussionSourceType type)
{
    if (!DiscussionProviderPool.ContainsKey(type))
    {
        lock (syncRoot)
        {
            if (!DiscussionProviderPool.ContainsKey(type))
            {
                string conUrl = string.Empty;
                string collection = string.Empty;
                string database = string.Empty;
                switch (type)
                {
                    case DiscussionSourceType.Task:
                        conUrl = MD.Framework.ConfigHelper.MongoMdTaskUrl;
                        collection = "taskTopic";
                        database = "taskcenter3";
                        break;
                    case DiscussionSourceType.Folder:
                        conUrl = MD.Framework.ConfigHelper.MongoMdTaskUrl;
                        collection = "folderTopic";
                        database = "taskcenter3";
                        break;
                    //这里新增其他模块讨论的配置文件的赋值
                }
                DiscussionProviderPool.TryAdd(type, new DiscussionProvider(conUrl, collection, database));
            }
        }
    }
    return DiscussionProviderPool[type];
}

```

4. 如果是已有讨论整合进来，需修自行改数据库实体

5. 返回前端的实体约定

(共4个，代码实体为大驼峰，前端接受实体为小驼峰)

```
/// <summary>
/// 讨论实体（获取讨论列表使用）
/// </summary>
public class DiscussionListModel
{
    /// <summary>
    /// 讨论
    /// </summary>
    public List<DiscussionModel> Discussions { set; get; }
    /// <summary>
    /// 与我有关计数
    /// </summary>
    public long CountWithMe { set; get; }
}
/// <summary>
/// 讨论公共模块前端实体
/// </summary>
public class DiscussionModel
{
    /// <summary>
    /// 讨论ID
    /// </summary>
    public string DiscussionId { get; set; }
    /// <summary>
    /// 依附主体id
    /// </summary>
    public string SourceId { set; get; }
    /// <summary>
    /// 消息
    /// </summary>
    public string Message { get; set; }
    /// <summary>
    /// 创建时间
    /// </summary>
    public string CreateTime { get; set; }
    /// <summary>
    /// 创建者
    /// </summary>
    public DiscAccountModel CreateAccount { get; set; }
    /// <summary>
    /// 回复 人信息
    /// </summary>
    public DiscAccountModel ReplyAccount { get; set; }
    /// <summary>
    /// 回复那条讨论ID
    /// </summary>
    public string ReplyId { get; set; }
    /// <summary>
    /// 附件列表
    /// </summary>
    public List<string> Attachment { set; get; }
    /// <summary>
    /// 讨论所在的网络ID
    /// </summary>
    public string ProjectId { get; set; }
    /// <summary>
    /// 讨论message中的用户信息
    /// </summary>
    public List<DiscAccountInMessageModel> AccountsInMessage { set; get; }
    /// <summary>
    /// 新增成员（目前只有任务用到）
    /// </summary>
    public List<DiscAccountModel> NewAccounts { set; get; }
}
/// <summary>
/// 前端组件遗留问题，这个改了前端改动比较大
/// </summary>
public class DiscAccountInMessageModel
{
    /// <summary>
    /// accountid
    /// </summary>
    public string Aid { set; get; }
    /// <summary>
    /// 名称
    /// </summary>
    public string Name { set; get; }
```

```

}
/// <summary>
/// 讨论中用户实体
/// </summary>
public class DiscAccountModel
{
    /// <summary>
    /// 用户id
    /// </summary>
    public string AccountId { set; get; }
    /// <summary>
    /// 用户名称
    /// </summary>
    public string Fullname { set; get; }
    /// <summary>
    /// 用户头像
    /// </summary>
    public string Avatar { set; get; }
    /// <summary>
    /// 状态
    /// </summary>
    public MD.Enum.AccountStatus Status { set; get; }
}

```

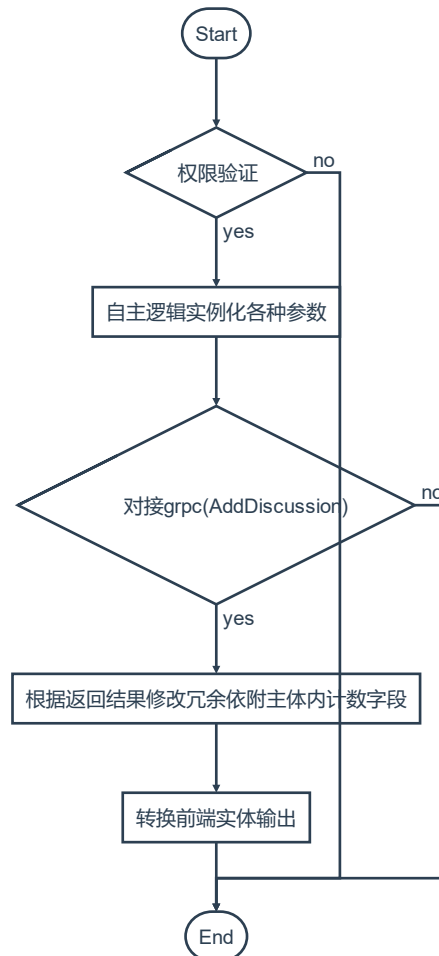
4. 接口逻辑介绍

[GRPC 接口及返回参数MD文档](#)

注：为了友好的返回结果提示，所有Grpc接口返回值都有isSuccess和error，当isSuccess为false时：error中会打印具体的错误信息，具体信息不一一罗列，具体的方法名和参数名见以上文档

以下为接入GRPC的主要逻辑流程和部分参数的详细介绍(grpc5个接口，最后组装4个页前端统一调用，附大致流程图和基本讨论请求参数，详细的最好看代码)

1. 添加新讨论



前端接口添加讨论：

请求参数	参数名	类型
依附主体id	sourceId	string
讨论信息	message	string
被回复讨论id	replyId	string
自主上传附件	attachments	string
知识附件	knowledgeAtt	string
APPID	APPID	string

接入grpc中非常规参数详解：

1)accountIdsMentioned 被@到的aid

通过以下正则解析出所有需要被推送push消息的人

```
step1:
    验证@全部成员正则:  @"\[all](.*)\[all]"
    true: accountIdsMentioned 加入所有依附主体的有效成员（任务项目为所有正常状态成员）
step2:
    验证是否包含成员正则:  @"\[aid\\](\\w{8}-\\w{4}-\\w{4}-\\w{4}-\\w{12})\[aid\\]"
    accountIdsMentioned 加入符合条件抠出所有直接提到的成员
```

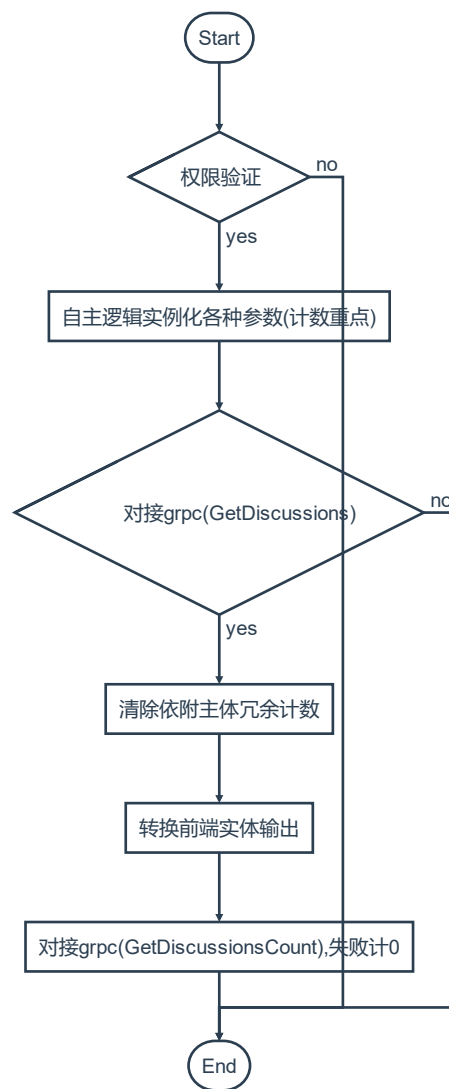
2) accountIdsNode node需要操作计数的人

这里主要作用为，发送讨论后给所有该讨论依附主体的有效成员添加node计数，任务和项目中的作用为导航条的小红点，一般情况下为依附主体的有效成员（也有特殊设定，比如在任务标记完成后，或者项目归档后，accountIdsNode为空）

3) memberAccountIds 依附主题的有效成员

全部有效成员

2. 获取讨论列表(组合获取关于自己的讨论计数)



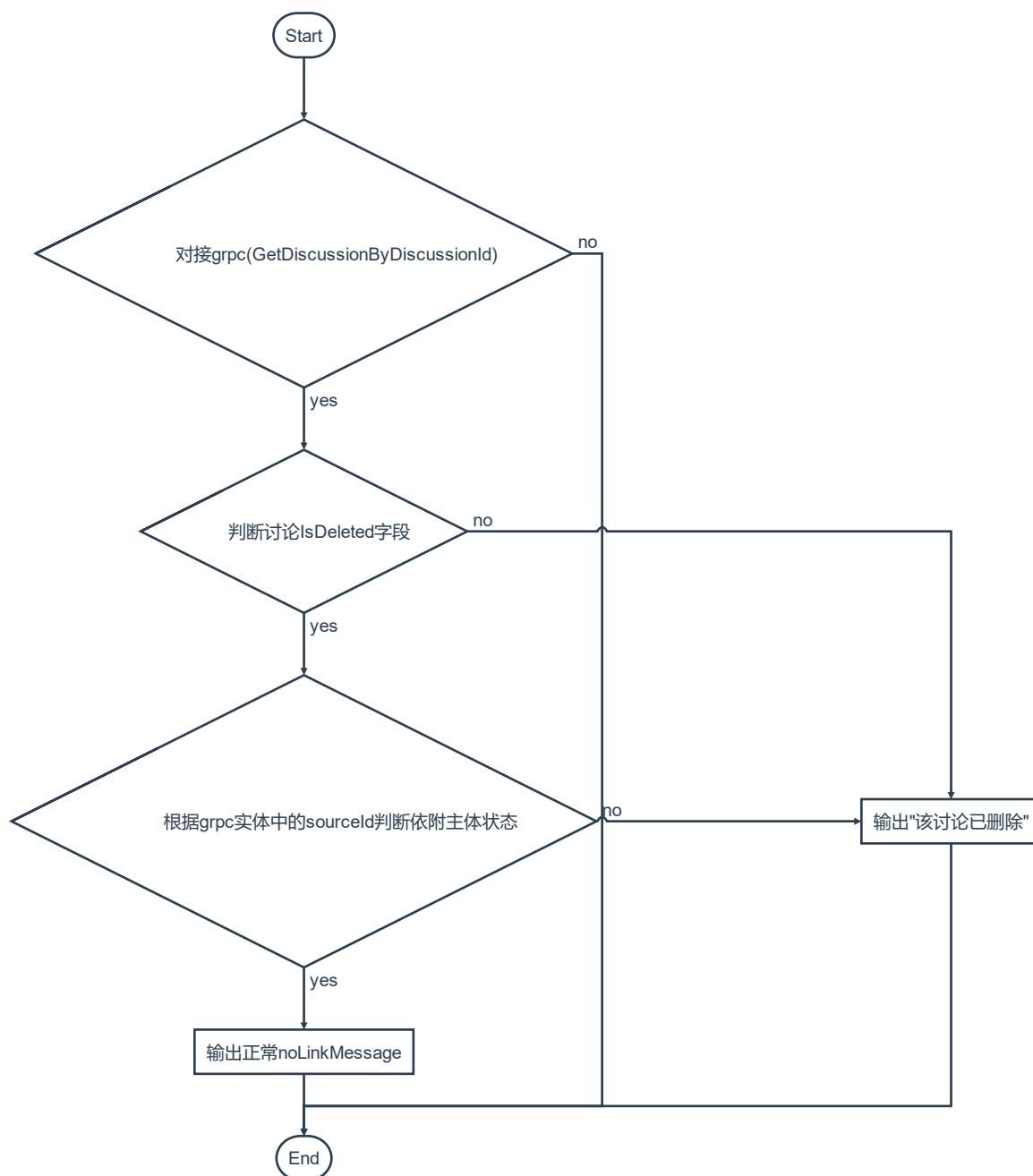
前端接口获取讨论：

请求参数	参数名	类型
依附主体id	sourceId	string
是否只看自己	isFocus	bool
请求页码	pageIndex	int
页面尺寸	pageSize	int

接入grpc中非常规参数详解：

1)

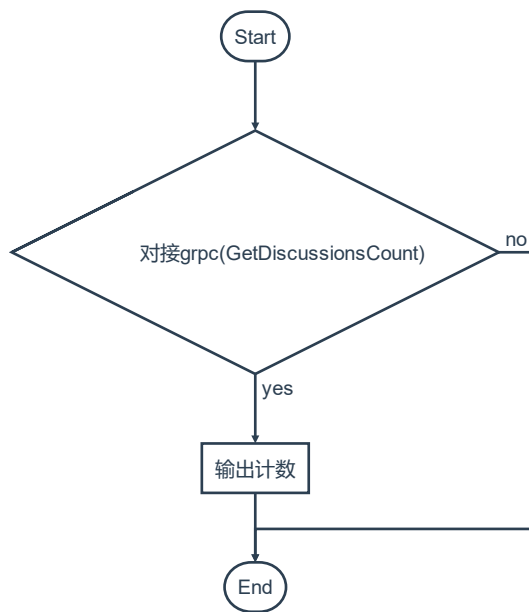
3. 获取指定单条讨论



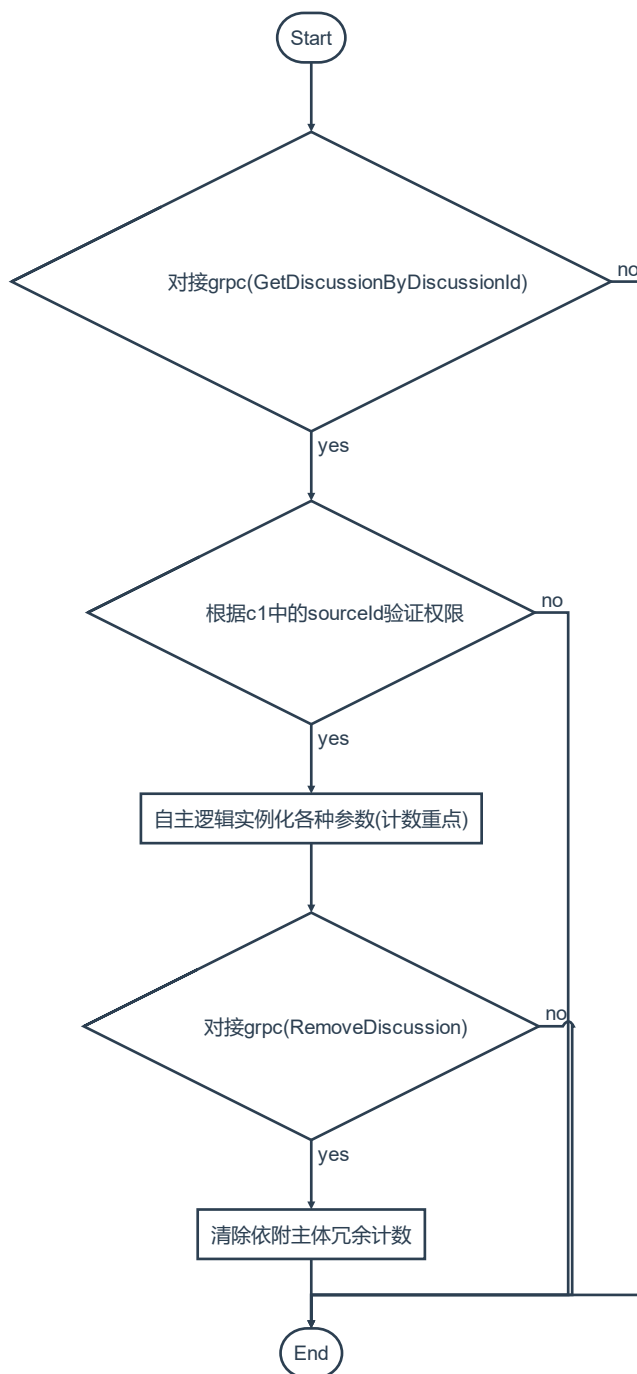
前端接口获取单条讨论：

请求参数	参数名	类型
讨论id	discussionId	string

4. 获取讨论计数（暂时没有独用，考虑到以后计数单独显示，所以先单独暴露出该接口）



5. 删除讨论



前端接口删除指定讨论：

请求参数	参数名	类型
讨论id	discussionId	string
是否删除附件	removeAtts	bool