

多人协同知识库系统设计

刘卓瞳^{1,2}, 任会峰^{1,3*}, 郭雅瞳²

¹无锡学院物联网工程学院, 江苏 无锡

²南京信息工程大学滨江学院, 江苏 无锡

³湖南中自工业智能研究院有限公司, 湖南 长沙

收稿日期: 2024年4月24日; 录用日期: 2024年6月17日; 发布日期: 2024年6月24日

摘要

提出一种多人协同知识库系统设计方法, 提供用户身份认证、协同邀请、范围控制、树形目录和协同编辑等功能, 以用户、角色和权限及其关系为授权认证的核心自定义登录认证逻辑, 基于开源的Yjs协作框架, 将其共享数据类型与编辑器组件, 并将网络服务和离线编辑服务打包为Vue组件, 支持多人在线实时协同编辑与离线编辑同步并定义授权逻辑, 生成Token缓存在Redis中以降低系统I/O压力, 并给出一个食品协作加工知识库系统的运行实例, 表明用户可以更高效地共享信息和协作创建知识库, 提高多人协作过程中收集和整理数据的能力。

关键词

系统设计, 知识库系统, 协同编辑, 协作框架, 授权认证

Design of a Multi-Manned Collaborative Knowledge Repository System

Zhuotong Liu^{1,2}, Huifeng Ren^{1,3*}, Yatong Guo²

¹School of IoT Engineering, Wuxi University, Wuxi Jiangsu

²Binjiang College, Nanjing University of Information Science & Technology, Wuxi Jiangsu

³Hunan Zhongzi Industrial Intelligence Research Institute Co., Ltd., Changsha Hunan

Received: Apr. 24th, 2024; accepted: Jun. 17th, 2024; published: Jun. 24th, 2024

Abstract

A design method for a multi-manned collaborative knowledge repository system is proposed, which

*通讯作者。

文章引用: 刘卓瞳, 任会峰, 郭雅瞳. 多人协同知识库系统设计[J]. 设计, 2024, 9(3): 764-772.

DOI: 10.12677/design.2024.93381

provides functions such as user identity authentication, collaborative invitation, scope control, tree directory, and collaborative editing. The core of authorization authentication is based on user, role, permission, and their relationships, with custom login authentication logic. Based on the open-source Yjs collaboration framework, it shares data types and editor components, and packages network services and offline editing services into Vue components. It supports real-time online collaborative editing and offline editing synchronization for multiple people, defines authorization logic, generates tokens cached in Redis to reduce system I/O pressure, and provides a running example of a food collaborative processing knowledge base system, indicating that users can more efficiently share information and collaborate to create knowledge bases, improving the collection and collaboration ability to organize data.

Keywords

System Design, Knowledge Repository System, Collaborative Editing, Collaborative Framework, Authorization Authentication

Copyright © 2024 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

随着互联网技术的不断提高和技术更新迭代,以及办公和学习场景的变化,团队成员可以远程协作[1],从而降低地域限制,提高工作效率。协同知识库通过提供丰富的知识管理和协作功能,为信息收集、实时协作、知识共享与交流、创新能力的提升等带来了极大的方便,帮助人们更好地适应新的学习和办公场景时代[2],特别是对带有多种支持数据组织结构的编辑器的协同知识库,能够更有效地进行知识和信息管理,其效果优于实时通信软件。

目前被广为使用的知识库协同设计方法有数据冲突处理算法为 OT (Operational Transformation)和 CRDT (Conflict-free Replicated Data Type)即“无冲突复制数据类型”[3]。OT 被应用到 Google docs 和 Office 365,国内近些年来出现的一些文档类产品,包括石墨、钉钉、腾讯文档等等也都是基于 OT 的。不同的协同知识库产品根据用户需求和场景的不同,具有各自的特点和优势:

(1) 企业微信知识库:针对企业用户,提供了丰富的知识管理功能,包括文档管理、多人协作编辑、权限控制等,帮助企业实现高效协同办公。

(2) 钉钉知识库:作为阿里巴巴旗下的企业级协同办公平台,钉钉知识库同样具备强大的知识管理功能,并通过云端同步,实现了跨平台的知识共享。

(3) Google Docs: Google 推出的在线文档协作工具,为用户提供了实时协作编辑、评论、版本管理等功能,支持多种文件格式,方便用户随时随地访问和编辑文档。

(4) 语雀:作为一款国产在线文档协作平台,语雀以 Markdown 为核心,为用户提供了丰富的文档编辑、管理、协作功能,支持知识库、团队协作空间等多种组织形式,适用于多种场景。

在使用 OT 时,其压力主要在服务端,每个用户都会将其编辑操作发送给中央服务器。服务器会根据当前的文档状态和接收到的操作,执行操作转换算法来调整操作的顺序和内容,以确保最终的结果是符合用户意图的,客户端无需像 CRDT 那样存储大量额外元数据,因此压力较小。但是,OT 需要中央服务器负责解决操作转换逻辑,以确保转换后的指令能够正确反映用户的编辑意图,这意味着 OT 对网络的要求较高,并且可以实现强一致性。如果某个用户出现网络异常,导致缺少了一些编辑指令,那么

基于已有状态进行的转换可能会出现问题。此外，随着使用场景的复杂性增加，需要考虑多个操作之间的顺序、位置和内容，以及可能的冲突解决策略，OT 转换算法的复杂性也会呈指数级增长。随着去中心化的流行，越来越多的产品更多地开始基于 CRDT 开发协同编辑功能，Yjs 正是基于 CRDT 的协同编辑框架专门为在 Web 上构建协同应用而推广的新方法，允许多个用户同时进行修改，确保更改的一致和连贯，保持无缝协作和实时更新。

互联网时代，过多的分散信息给员工之间的配合带来麻烦，信息难以共享，团队成员重复劳作，业务经验无法沉淀，知识资产流失严重。为此，本文提出一种基于 Yjs 的协同知识库设计方法并将其用于现实世界，实现了高效协同工作。

2. 设计方法与架构

协同知识库前端基于 Vue.js 框架，使用 Element-UI 组件库完成页面设计，并结合 Yjs 框架为 Tiptap 编辑器添加协同编辑功能[4]。后端采用 SpringBoot 框架作为基础，SpringSecurity 实现用户登录认证和权限验证，SpringMVC 进行请求分发控制，Mybatis 用于与关系型数据库进行数据交互。此外，系统还使用了 Redis 作为缓存和全局唯一 ID 生成器[5]，同时使用 MySQL 作为关系型数据库。

图 1 是多人协同知识库的架构图。知识库编辑功能基于 SpringSecurity 和 MySQL 实现了基于角色的访问控制，包含三种角色：创作者、协作者和普通用户。此外，还添加了管理员角色，管理员拥有全部权限，可以对用户、角色、权限、知识库和代码分享片段进行统一管理。

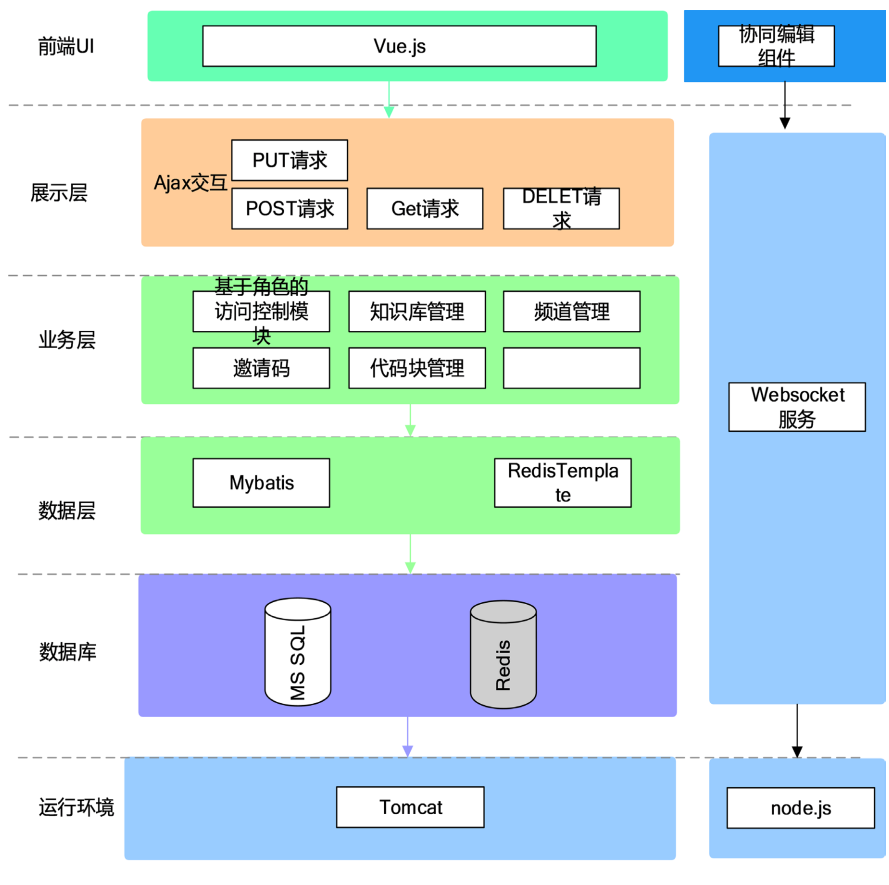


Figure 1. System schematic diagram
图 1. 系统架构图

图 2 为实体关系图，其中，“用户”实体包含基本的编号、用户名和密码，还包含一些个人信息如电话和邮箱；“角色”实体将编号和知识库编号作为唯一主属性，这样做是为了让每个知识库都有自己的管理员、协作者和普通用户角色。角色实体还包含了角色名和角色昵称，角色名是为了在更好的在程序中标示；“角色”和“用户”两个实体之间的“拥有”关系是为了记录角色和用户的多对多关系而设置的，角色编号属性和用户编号属性共同构成唯一主属性；“权限”实体包含基本的编号属性，知识库编号属性是为了区分权限所属不同的知识库，这是由于每个知识库都有相同的权限名，添加了知识库编号就可以唯一对应一个主属性；“角色”和“权限”两个实体之间的“拥有”关系是为了记录角色和用户的多对多关系而设置的，角色编号属性和权限编号属性共同构成唯一主属性；“知识库”库实体包含基本的编号属性，以及知识库信息知识库名、简介、创建者名称；“邀请码”实体包含基本的编号属性，所属的知识库的编号属性，邀请码属性遵循唯一原则，将 Redis 的自增值进行 62 进制转换得到，失效时间属性是为了在查询时免去查询 Redis 而添加；“频道”实体包含基本的编号属性作为唯一主属性，父级编号属性是为了将频道构成树形结构，顶级频道的父级编号属性为 0，其它非顶级频道都有父频道，此外还有频道的基本信息属性频道名、频道简介、创建者名称；“代码块”实体包含基本的编号属性，代码块实体的设置是为了让知识库更加适应程序员的应用场景，还包含知识库编号、代码块信息、代码块名称、代码块简介、创建者名称等属性。

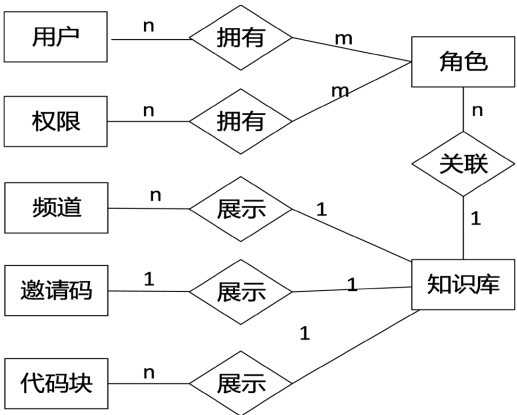


Figure 2. Entity relationship diagram
图 2. 实体关系图

3. 案例研究与讨论

3.1. 认证功能

图 3 是实现的一款食品加工工业中的多人协同知识库首界面，包含当前知识库名称显示，根据不同的知识库动态的显示不同的名称，还包含更多操作下拉框，根据访问者角色的不同来提供不同的操作。还有探索页面跳转按钮、频道折叠面板以及协同编辑组件。频道折叠面板将频道分为了两种，功能频道和文字频道，功能频道用于适应不同的应用场景，本系统提供了适应程序员场景的代码集锦频道，用于代码块的创建和分享。

当用户首次访问知识库系统时，浏览器向服务器端发送请求。如果用户此时还没有登录，Spring Security 将予以检测，并通过 Username Password Authentication Filter 过滤器抛出一个异常，表示用户未认证。Spring Security 接着清除 Security Context Holder 中的身份认证信息。然后使用自定义的 Authentication Entry Point 类处理这个未认证请求。自定义的 Authentication Entry Point 返回 401 状态码和

认证失败的消息。然后，前端重定向用户到登录页面，要求用户输入正确的用户名和密码来登录。一旦用户输入了正确的登录信息，后端会生成一个 token，并把这个 token 存储在 Redis 数据库中，用于跟踪用户的会话。每次用户的请求都会通过该 Redis token 验证用户的登录状态。如果用户已登录认证，身份信息会被放入 Security Context Holder 完成认证过程。



Figure 3. Multi-manned collaborative knowledge repository system homepage
图 3. 多人协同知识库首页面

根据角色的不同，每个用户可以进行的操作不同。图 4 和图 5 展示了管理员创建者、非管理员创建者的主要操作。管理员功能包括邀请其他人参与协同、设置信息的可见范围、添加频道和修改部落信息、删除部落等；非管理员的功能包括邀请其他人、添加频道和退出部落等。



Figure 4. Display of administrator creator's functions
图 4. 管理员创建者的功能展示



Figure 5. Function display for non administrator creators
图 5. 非管理员创建者的功能展示

3.2. 协同邀请

协同知识库的邀请功能通过生成邀请链接来实现，其基本操作步骤为：

(1) 邀请者复制邀请链接给被邀请者。

(2) 为用户分配该知识库的协作者角色，被邀请者将具备该知识库协作者的操作权限，并被重定向到对应的知识库中。

(3) 生成邀请链接：邀请链接由请求路径和邀请码构成，邀请码是长度为 6 的字符序列，通过对 Redis 生成的递增序列进行 62 进制编码生成。编码后的邀请码值包含数字和大小写字母，不包含特殊字符，通过使用 Redis 生成递增序列可以有效避免并发时的问题。长度为 6 的 62 进制序列可生成多达五百六十亿次邀请码。

(4) 生成的邀请码将会被 Redis 缓存，并设置过期时间，同时也会存储到数据库中。前端发送查询邀请码请求时，后端会先查询数据库中是否存在未过期的邀请链接。如果存在，则刷新缓存和数据库的过期时间，并将邀请码发送给前端。前端收到邀请码后将邀请码与请求路径拼接生成邀请链接。

知识库的创建者和协作者可以使用邀请功能，邀请链接为后端生成的邀请码和前端当前域名和接收邀请的请求路径拼接而成，重置链接按钮将会向后端发送请求，得到后端生成的邀请码后，再将邀请码重新进行拼接，如图 6 所示。

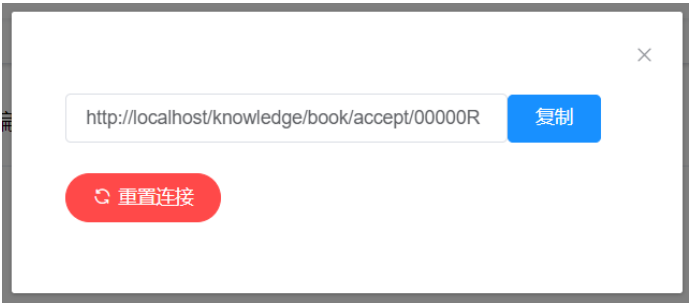


Figure 6. Invitation function display
图 6. 邀请功能展示

3.3. 范围控制

图 7 是修改控制知识库可见范围的功能展示，修改知识库可见范围表单包含一个单选框和一个更新按钮，将知识库的可见范围设置为公开访问并提交，页面将会向后端发送请求，后端修改数据库中该知识库的可见范围字段后，该知识库将在探索页面被其他非协作者和创建者看到。

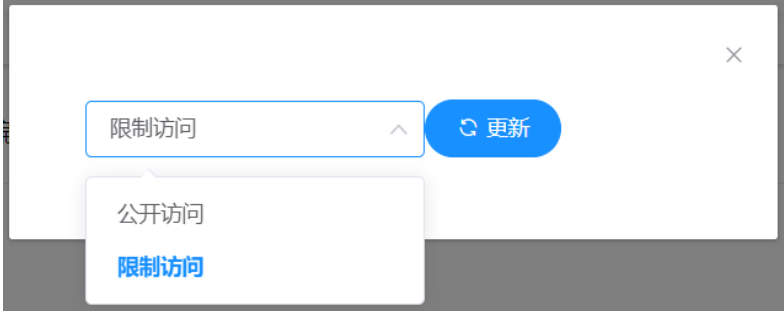


Figure 7. Display of visible range control for knowledge base
图 7. 知识库可见范围控制展示

范围控制的实现包含了前端和后端两个部分。前端部分使用 Element-UI 的 el-pagination 组件，主要用到了它的 current-page 属性以及 current-change 事件。current-page 属性表示当前页面的页数，它支持 sync 修饰符，可以与自定义的 current-page 属性进行双向绑定。当切换页面时，current-page 的改变会引起 current-page 属性改变。同时，监听的 current-change 事件，定义了一个处理器来向后端发送分页查询请求，在 current-change 事件触发时就会调用处理器方法。后端部分使用 MyBatis-PageHelper 插件，先从请求中获取请求参数页码和每页显示记录数，调用 startPage 方法即可实现分页查询。PageHelper 为 SQL 查询语句添加 LIMIT 子句和参数。

3.4. 树形目录

图 8 是频道添加的功能展示，顶级频道可以在当前频道的下拉框中进行添加，每个频道的子频道可以点击树形结构导航栏节点的右边+号进行添加。添加表单包含一个输入框和一个创建按钮。

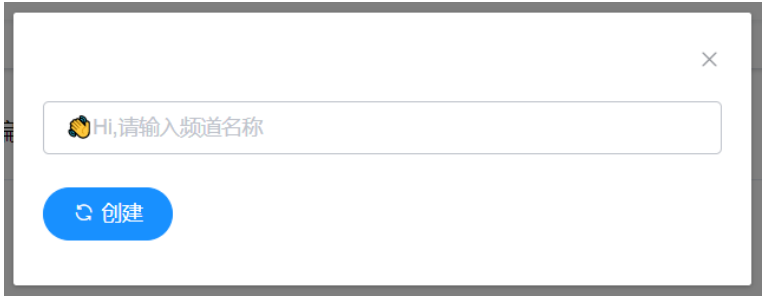


Figure 8. Channel addition function display
图 8. 频道添加功能展示

知识库的频道拥有子频道形成树形结构，为了展示该结构使用 Element-UI 的 el-tree 组件，后端涉及树形结构数据库表的设计以及树形结构查询。el-tree 需要特定的数据结构每个节点需要包含 label 和 children 字段，其中 children 字段为字节节点数组，在后端对数据库查询得到的链表进行了结构转换。el-tree 使用 data 属性、node-click 事件和 slot-scope 插槽。它们的作用如下所示。

(1) data 用于绑定转换后的数据结构，监听 node-click 事件可以在 el-tree 节点被点击时触发自定义处理器的执行。

(2) 处理器可以接收 data 参数，data 参数可以提供当前节点信息，可以使用 data 参数动态更新当前频道的状态。

(3) 使用 slot-scope 为 el-tree 自定义节点元素，传递 data、node 参数给自定义节点元素，这样节点元素就可以访问当前节点的数据。通过 slot-scope 为 el-tree 节点添加了 show-more 图标，点击图标可以对该节点进行更多操作，如添加子频道、修改频道信息。

在设计数据表时为属性结构表添加 parent_id 字段来构成树形的表结构，后端查询得到链表后根据 id 和 parent_id 进行数据结构的转化将链表结构转化为树形结构。算法逻辑为：先创建一个 ResultList，将链表元素都放入 HasnMap 中，遍历 RowList，如果父节点为 0 则放入 ResultLis 中，如果父节点不为 0 则一定有父节点，通过 perentId 在 HashMap 中查询父节点，并将自己放入父节点，循环这个过程直到结束，该算法时间复杂度为 $O(n)$ 。

3.5. 协同编辑

图 9 是协同编辑功能展示。协同编辑的主体是 Tiptap 编辑器，整合了多种插件以提供更多的数据组

织显示和功能，例如有序列表、无序列表、引言、代码块、行内代码、代码语法高亮、文本高亮、历史功能、文本内链接和表格等。此外，使用 TailWindcss 框架为其提供用户体验更好的样式。



Figure 9. Collaborative editing function display
图 9. 协同编辑功能展示

协同编辑的实现采用 Yjs 框架，该框架提供多种无冲突共享数据类型。在此设计中，使用 YDoc 数据类型，并借助 Yjs 的 Binding 功能将 YDoc 与 Tiptap 绑定。Tiptap 使用 Delt 数据模型记录编辑器内容的变化，以便第三方组件可以通过 Delt 操作 Tiptap 组件。因此，绑定后，Yjs 可以将实时协同数据更新到 Tiptap 中。为了实现该协同功能，还需要网络服务。因此，在本地运行一个 WebSocket 服务器，并配置相应端口。即可进行冲突处理并将处理后的数据更新到用户端。此外，还添加了离线编辑功能，即使没有网络，也可以继续编辑。新的更改将被存储在本地，并在联网时更新到 WebSocket 服务器。

完成以上配置后，利用 Vue 的监听功能以及 Tiptap 更新时的回调函数实现了协同编辑组件的双向绑定。这样就可以实时获取编辑器更新的数据，并进行持久化保存。此外，还暴露了用户名、频道号码和是否为只读等属性，提供了更好的交互性，使得用户可以看到其他用户的用户名。封装了 Tiptap 编辑器和 Yjs 框架的 Vue 协同编辑组件可以与其他人同时编辑同一份文档，即使在相同区域编辑也不会发生冲突，前端组件集成了 Yjs 框架的 Provider，后端运行了一个 WebSocket 服务器，可以进行全双工通信实时更新协作者的最新编辑。

4. 结论

基于 Yjs 框架的多人协同知识库通过访问控制和协同编辑，根据不同的角色访问不同的知识库，并邀请别人进行协同编辑，实现了高效的信息交流与共享，可用于食品合作加工、美工协同设计等需要多

人协作以提高设计效率的现实应用场景。

基金项目

江苏省高校哲学社会科学研究课题“基于多模态信息融合的网络舆情监测研究”(2022SJYB0982)。

注 释

文中所有图片均为作者自绘。

参考文献

- [1] 杨旻曦, 刘航, 高大化, 等. 面向知识库协同演进的知识通信[J]. 移动通信, 2024, 48(2): 27-33.
- [2] 杨倩, 朱平, 林鹤. 多算法协同的医疗器械产品竞争情报知识库构建方法研究[J]. 竞争情报, 2022, 18(5): 30-39.
- [3] 于庆源, 徐从, 崔勇骏. 基于 BIM 技术的设计项目分布式数据集成协同设计平台[J]. 内蒙古科技与经济, 2023(21): 111-114.
- [4] 彭颢. 基于 ANN 算法的高校信息管理系统设计[J]. 科技资讯, 2023, 21(12): 32-35.
- [5] 程远瑶, 王昕怡, 陈逸, 等. 基于 Redis 的 ATS 缓存系统设计与实现[J]. 计算机应用与软件, 2023, 40(7): 111-115.