

协同文档实现要求和建议

洪志远 2018 年 6 月末



这次的任务是一期任务量较大、涉及知识较多的任务，这次任务基于之前我在团队的分享：协同文档的系统设计。所以如果没有听过这个分享，可以看一下我导出了 pdf。

废话不多说 😊，下面是任务简介和详细的任务要求：

这次任务的主要内容是，实现一个带有用户权限的可以协同编辑的 markdown 编辑器。😎

功能

1. 可以协同编辑
 1. 协同编辑采用 ot 算法来实现，也就是我分享的
 2. 暂时只考虑小规模用户
2. 支持完整的 github markdown 特性 😊
 1. 对于图片，添加一个上传本地图片到服务端的功能
 2. 注意通过某种手段保护这些图片，免得被外链
3. 可以一边写作，同时可以预览。下面是简易的几种实现方式
 1. 左右分栏：代码编辑器、预览
 1. 如果做成这样，保证代码编辑器超出屏幕滚动的时候，预览可以跟着一起滚动
 2. 编辑过程中，因为在中间改动而新添加的文字、元素，要能在右边预览中体现出来
 3. 代码编辑器要完全高亮
 2. 实时预览，所见非所得，例如 typora。
4. 带有用户系统（最后一期后端任务的完成度差强人意，这里把那些问题改好）
5. 带有虚拟的文件系统
6. 文档拥有者可以设置文档的权限：
 1. 所有人可编辑
 2. 部分人可编辑
 3. 不许谁编辑
 4. 所有人可查看（不可编辑）
 5. 部分人可查看
 6. 不给谁看
 7. 等等。。。也可以是上面这些规则的组合
7. 可以通过一个 URL 访问这个文档。

暂时不要太多功能，目前这个任务的功能是应用开发中比较典型的一些，你们以后如果从事 web 方面的工作，一定会遇到类似情景。遇到希望诸位可以认真对待。

技术方面的要求

技术方面的要求是为了进一步保障任务完成质量。

1. 不允许使用 iframe 😡
2. 可以使用任何第三方库但是不能用下面这些
 1. etherpad
 2. automerge
 3. 等等帮助实现了大量本次任务要求的内容的库
3. 模块化开发
4. 前端不能有任何体验上的卡顿
5. 使用自己能拿下的最新的 es（在原型链、闭包什么的彻底搞懂的情况下）
6. 编码习惯不说了，自己注意 😊

建议

可能可以帮助你更快速、高质量的完成任务，或者收获更多干货。

1. 对于编辑器模块和协同编辑模块的组合，可以一步一步来，例如先使用最简单的 textarea，然后协同编辑发现没问题了，上代码编辑器
2. 由于 UI 还是不算复杂，没有强烈的需要使用 react、vue 等框架的需求，请诸位自行评估
3. 建议先慎重思考 🤔、仔细选择技术选型，然后再去写代码。整个软件的结构、模块化等等，不要随意瞎搞胡搞
4. 可以尝试解包 typora 的安装包 🤔，将它的编辑器抽出来为己用，我试过，很简单的，但是能让你基本上不在 markdown 编辑器这一块费心思了

选做和进阶

下面这些要求是你们能力之内的、但具有挑战性的、需要额外学习的一些要求。

1. 学习 webrtc，重构协同编辑系统，使用 webrtc 将客户端与服务端长链接变为客户端之间 p2p 的长链接，从根本减轻服务端负载 🚀
2. 使用 C++ 或 Rust 或其他高性能的语言，实现 follow 的逻辑，在前端使用 wasm 调用，在后端使用 nodejs C++ addon 或 nodejs ffi 调用
3. 将整个软件通过 electron 封装 📦，赋予它操作文件系统等原生能力
4. 将协同编辑系统抽象出来成为一个库或者框架，使它可以方便的和任意已有应用组合，赋予应用协同编辑的能力 🙌

激励措施

考虑一下几点可能可以帮你让自己在暑假除了必要的休息 😴 娱乐 🎮 之外，也保持高效率：

1. 完整实现且体验很好的话，这个任务其实你们已经写成了一款产品 🎉，可以在团队内部推广自己的协同文档作为团队会议记录、资料积淀、分享演示等用的工具
2. 可以作为自己的笔记工具 📝
3. 可以放在简历上，实际上比什么一眼就能看出来是课设的东西有分量多了
4. 可以完整实现的话 🤖，你。。反正就是很牛逼了！👍

时间安排

在任务开始之前，你有一周的时间复习、学习完成这个任务需要的知识，然后在两周内完成这个任务。

绝对时间安排待定。🤔🤔🤔