

这篇不一样，是真的手写的。理一下组件吧。

2026.01.19

首先最简单的是 App.svelte 和 Navbar.svelte，除了对 activePage 的双向绑定和简单的条件渲染之外，就只有对 IndexedDB 和悬浮球组件的引用。

顺便清理一下 vite init 时用于教学的组件。

别看现在只有 ApiTest.svelte、FloatingBall.svelte 和 ApiTest 这三个核心组件，但实际上他们仨都是重头戏。

嘛，今天的计划还是理一下 lib/core，这个定下来以后预计不会动了。当然，肯定要考量 Dispatcher 的部分。由于现在还没有 core-runner，所以得提前考虑，不然 Dispatcher 和 core/ 的具体组件也不好设计。

14:39: 好吧，思考了一下觉得 Dispatcher 的模式不怎么样。干脆实现一个消息总线吧，或者调研一下有没有现成的。实际上之后悬浮球组件要扩展成与本地存储沟通的桥梁，也会是个比较大型的东西；另外，core-runner 也会与 UI 层有频繁的沟通。我在想，说是 core 层和 UI 层有双向的消息交互，但实际上核心的点还是 core 层为主导，从 UI 层到 core 层的消息一般不会引起 UI->core->UI->core 这样的链式死循环，最多只会有 UI->core->UI 这样三层的消息同步，而且应当适当为 UI 层加入锁定机制以免状态机混乱。