

 字节跳动云原生 |  稀土掘金

怎么实现会比较好

😏 完善的标题栏:

😂 基本的菜单栏:

😂 记住大小位置:

💡 发现插件模式可以在生命周期里埋入钩子做些事情，能够很好用于抽象通用能力。



+ New chat



Today



手机版本



度假酒店体验评价



OpenAI 后续 Sam Alt

主要观点

- OpenAI 目前受 GPU 限制，短期计划受到延迟
- 目前仍存在上下文长度和令牌数量方面的挑战
- 微调 API 和专用模型服务受到 GPU 可用性的限制

應用窗簾能力

有趣的几何

首添使用の困感



我想要的

无感
支持

开关
配置

还要
好用



+ New chat



Today

🗨 手机版本

🗨 度假酒店体验评价



OpenAI 后续 Sam Alt

主要观点

- OpenAI 目前受 GPU 限制，短期计划受到延迟
- 目前仍存在上下文长度和令牌数量方面的挑战
- 微调 API 和专用模型服务受到 GPU 可用性的影响

应用窗口能力

有趣的功能

首次使用的困惑

- 😲 啥？Mac上很常见的沉浸式标题栏居然不好使？还拖不动！
- 😱 啥？顶部菜单栏还需要自己写才可以用？一点都不像应用！
- 😓 啥？居然不能默认记住上次窗口启动的大小和位置？麻烦！

我想要的

无感
支持

开关
配置

还要
好用

怎么实现会比较好

- 😏 完善的标题栏：写入 HTML / CSS 制定虚拟栏，拖动/双击时候 JS 调用 Tauri 的 Api 实现应用效果
- 😂 基本的菜单栏：在 Rust 层将一个一个个菜单项以及动作写好，默认内置，快捷键效果进行传递
- 😄 记住大小位置：通过 Tauri 不那么丰富的插件能力 plugin-window-state 注入到启动层实现

💡 发现插件模式可以在生命周期里埋入钩子做些事情，能够很好用于抽象通用能力。

JS 和 Rust 之间的通信

有趣的功能

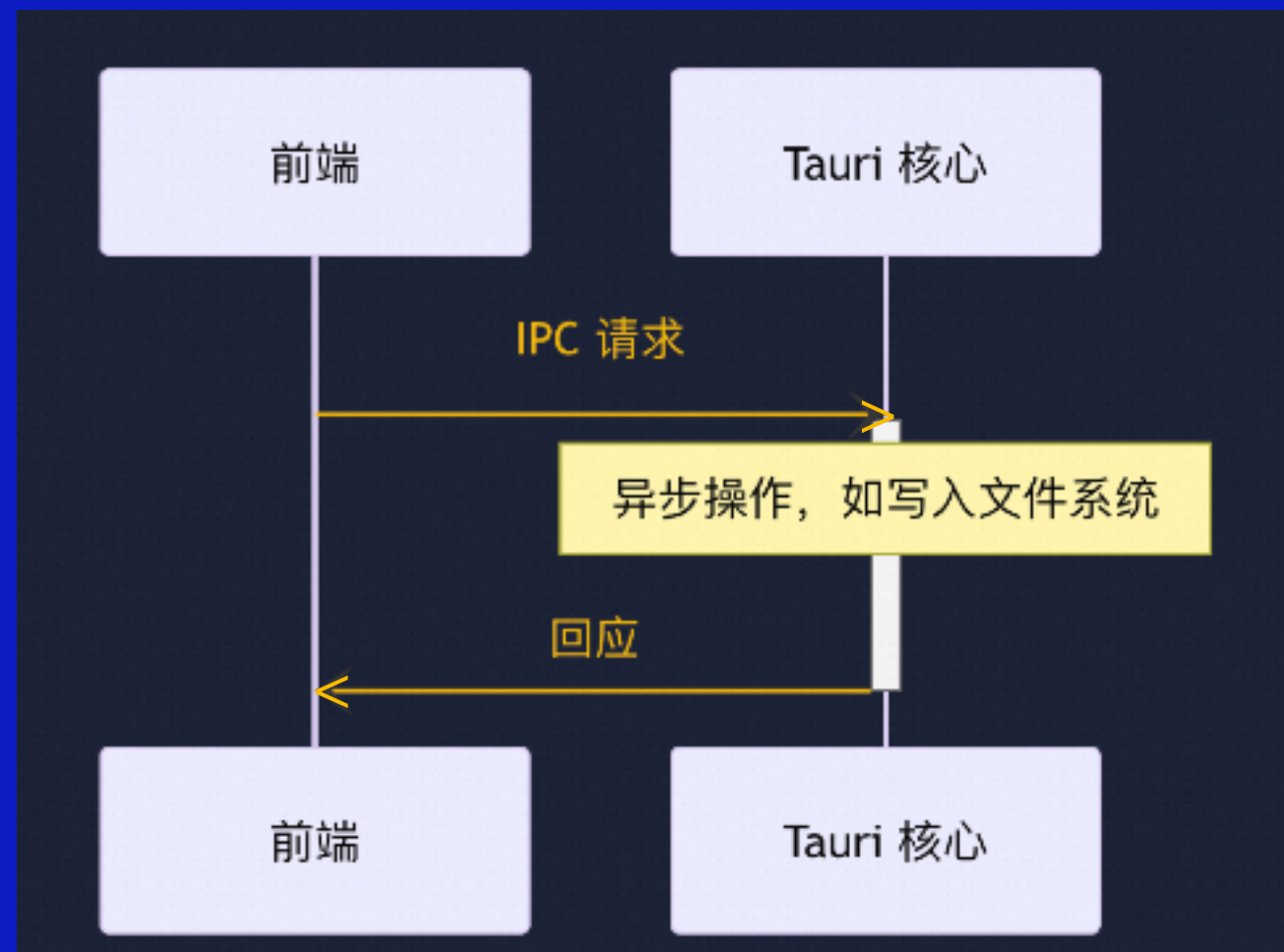
下载功能 [JS 调 Rust] `invoke('download_file', { params: { url, filename } });`

JS

```
#[command]
pub async fn download_file(
    app: AppHandle,
    params: DownloadFileParams
) -> Result<(), String> {
    let window: Window = app.get_window("pake").unwrap();
    let output_path = api::path::download_dir().unwrap().join(params.filename);
    let file_path = check_file_or_append(output_path.to_str().unwrap());
    let download = Download::new(&params.url, Some(&file_path), None);
    match download.download() {
        Ok(_) => {
            show_toast(&window, &get_download_message());
            Ok(())
        }
        Err(e) => {
            show_toast(&window, &e.to_string());
            Err(e.to_string())
        }
    }
}
```

Rust

指令调用 [invoke]



事件通信 [listen]

