

 字节跳动云原生 |  稀土掘金

JS和Rst之间的通信

有趣的几何

```
#[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())
        }
    }
}
```

下载功能 [JS 调 Rust]

```
invoke('download_file', { params: { url, filename } });
```

JS

Rust

```
pub fn show_toast(window: &Window, message: &str) {  
    let script = format!(r#"pakeToast("{}");"#, message);  
    window.eval(&script).unwrap();  
}
```

Toast 功能 [Rust 调 JS]

事件通信 [listen]

指令调用[invoke]

前端

Tauri 核心

事件

前端

Tauri 核心



前端

Tauri 核心

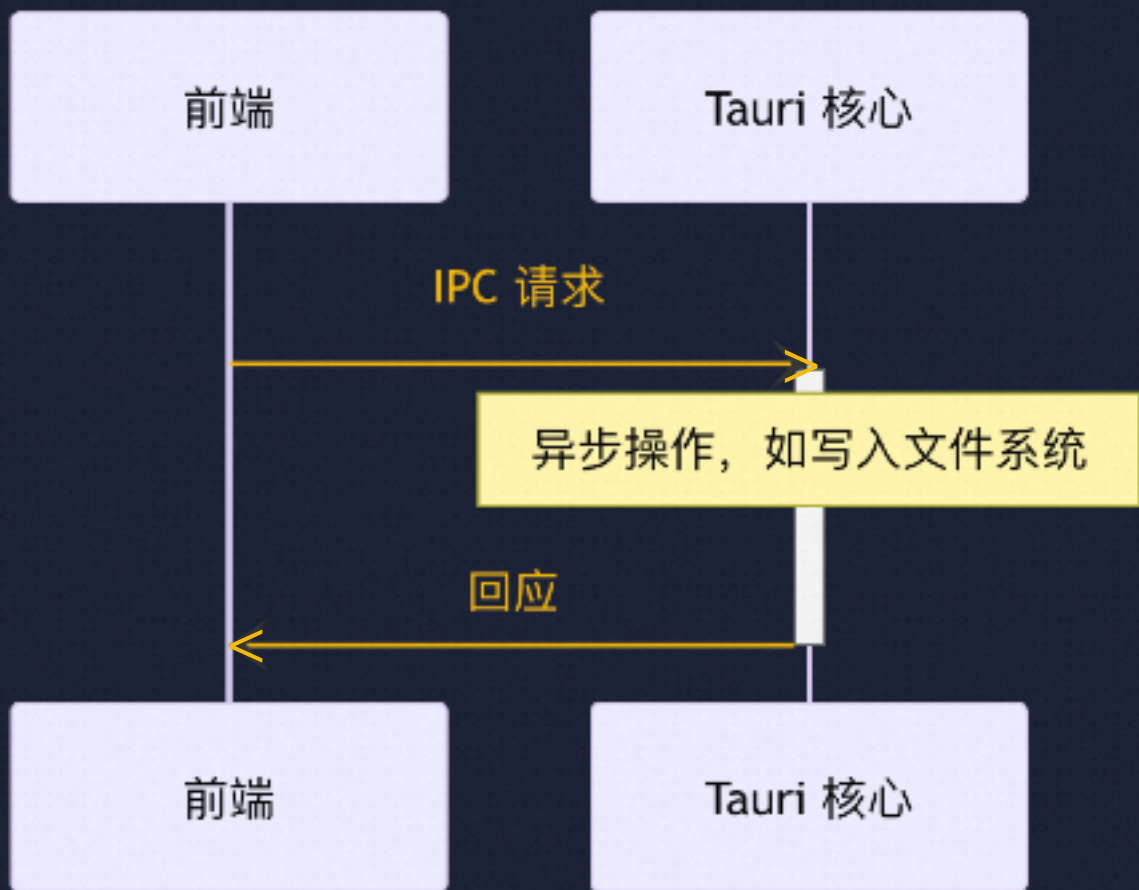
IPC 请求

异步操作，如写入文件系统

回应

前端

Tauri 核心



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

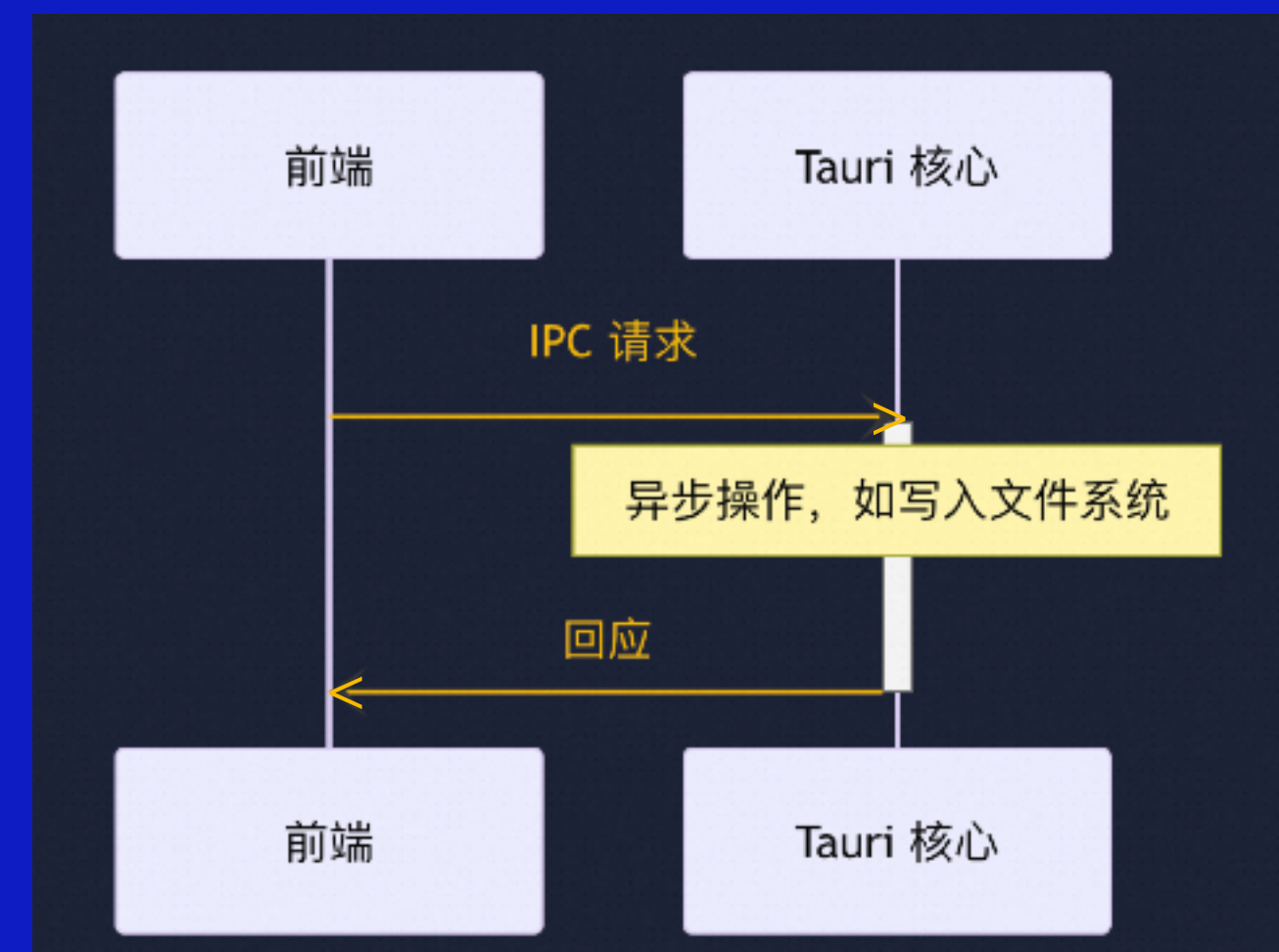
```
#[command]
pub async fn download_file(
    app: AppHandle,
    params: DownloadFileParams
) -> Result<(), String> {
    let window: Window = app.get_window("pake").unwrap();
    let output_path = api::path::download_dir().unwrap().join(params.filename);
    let file_path = check_file_or_append(output_path.to_str().unwrap());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

pub fn show_toast(window: &Window, message: &str) {
    let script = format!(r#"pakeToast("{}");"#, message);
    window.eval(&script).unwrap();
}
```

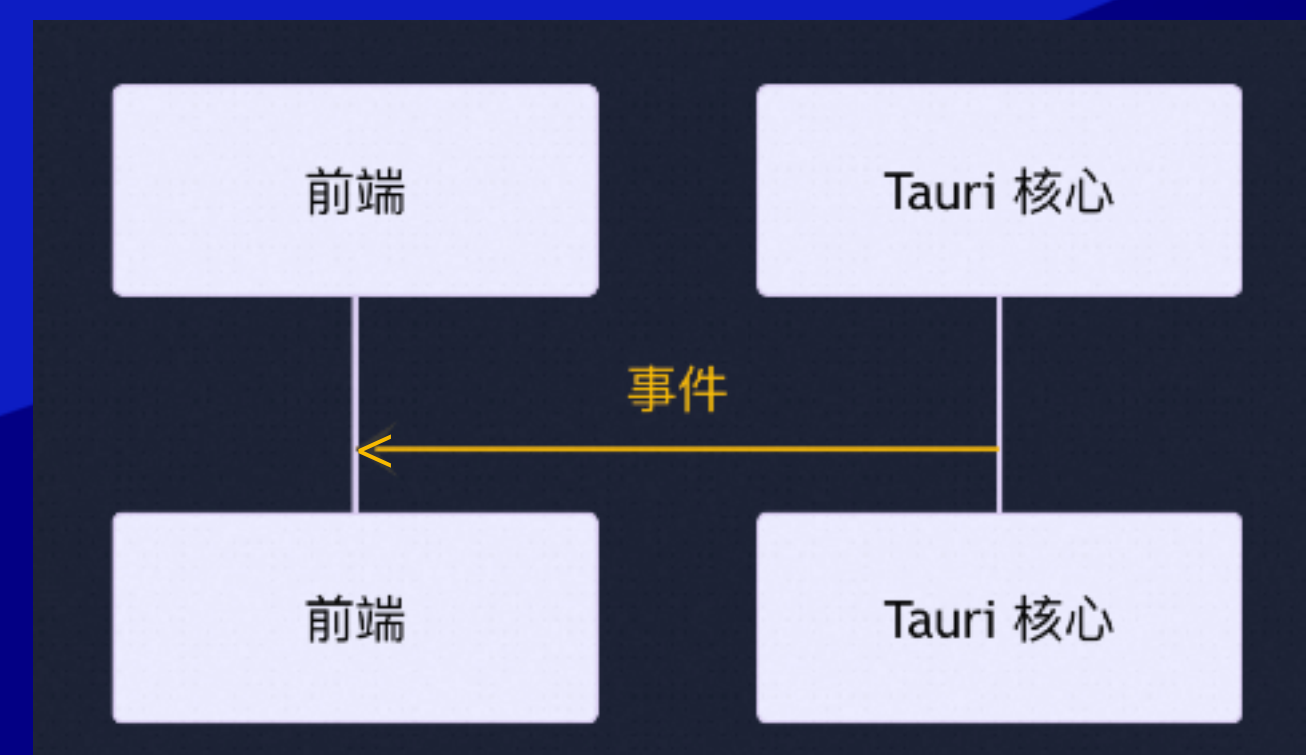
Rust

Toast 功能 [Rust 调 JS]

指令调用 [invoke]



事件通信 [listen]



各类事件能力承接

有趣的功能

