

像 MAGIC 一樣厲害的 XState

Terry Lin @MediaTek

What is XState?

XState 是一個用於狀態管理的 JS 函式庫

利用有限狀態機 (FSM) 和狀態圖 (Statecharts)

來描述與管理應用中的各種狀態與轉換，使邏輯更加明確且易於維護

- 可視化工具
- 框架無關性
- 強大的擴充性

能讓開發者更輕鬆地追蹤狀態轉變的過程

還可降低程式因條件混亂或多層 if-else 導致的錯誤

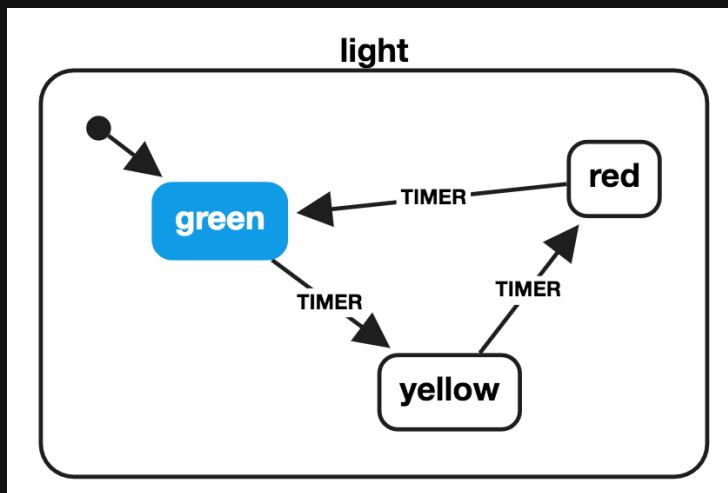
避免因小狀態之間的多種自由組合，導致結果過於繁雜



Finite State Machine

有限狀態機 (FSM)，是一種用來描述系統行為的數學模型

- 有限狀態：系統的所有可能狀態是有限且固定的，每次系統僅能處於其中一個狀態
- 狀態轉換：根據外部輸入或事件，系統會從當前狀態轉換到另一個狀態
- 輸入條件：轉換的觸發條件通常由外部輸入決定，不同的輸入對應不同的轉換路徑
- 動作或輸出：在狀態轉換過程中，系統可能會執行特定動作或產生相應輸出



Quick Start

```
1  const toggleMachine = createMachine({
2    id: 'toggle',
3    context: ({ input }) => ({
4      count: 0,
5      maxCount: input.maxCount,
6    }),
7    initial: 'Inactive',
8    states: {
9      Inactive: {
10        on: {
11          toggle: {
12            guard: ({ context }) => context.count < context.maxCount,
13            target: 'Active',
14          },
15        },
16      },
17      Active: {
18        entry: assign({
19          count: ({ context }) => context.count + 1,
20        }),
21        on: { toggle: 'Inactive' },
22      },
23    },
24  });
```

Quick Start

1. Parent states
2. Parallel states

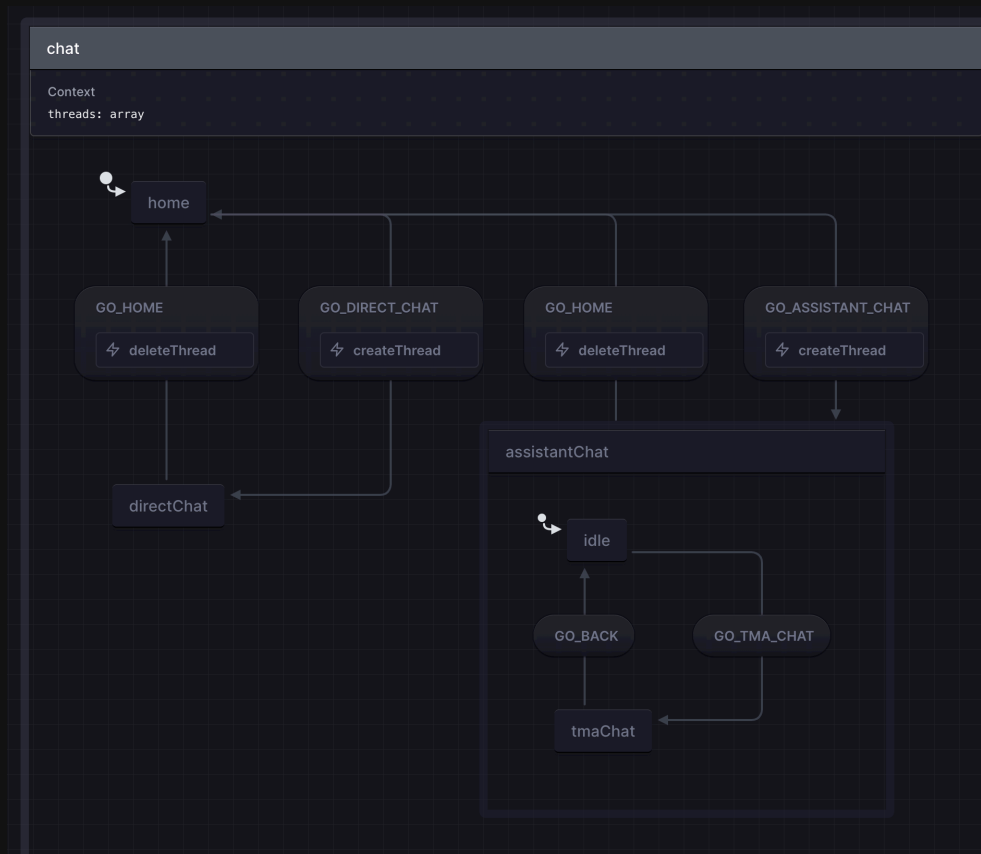
State Chart Demo

以 Codebase 中出現的主要狀態做範例：

1. home
2. directChat
3. assistantChat
4. tmaChat

要 Simulate 請到 Stately Editor


或者用 VSCode Extension



Todos

1. 深入探索 XState 更多功能，參考官方文件
2. 若要與 Jotai 的整合，可用 jotai-xstate
3. 可以在適合的場景中進行小規模嘗試
 - XState :
 - 適用於狀態邏輯較複雜、流程繁多且需要嚴謹控制的情況
 - 因其能夠明確定義各狀態、事件與轉換，並可視覺化追蹤整個狀態流程
 - Jotai :
 - 適用於狀態邏輯較簡單或需要細粒度、局部管理的場景
 - 其極簡的 API 設計只會更新實際依賴該 atom 的元件，從而提升效能

Slide Repo Link

Powered by  Slidev