

依存関係があるコンポーネントは
Barrel ファイルでまとめよう

azukiazusa

自己紹介

- azukiazusa
- <https://azukiazusa.dev>
- FE（フロントエンド|ファイアーエムブレム）が好き

依存関係があるコンポーネントは
Barrel ファイルでまとめよう

依存関係があるコンポーネント

- `<select>` と `<option>` のような関係
 - `<option>` は必ず `<select>` の子要素として使う必要がある
- 共有状態を相互に依存していて `Context` でデータを受け渡されることが多い

コンポーネントに依存関係があること
がわかるようにしたい

Compound Components パターン

```
import { Tab } from "./Tab";

function App() {
  return (
    <Tab.Root>
      <Tab.List>
        <Tab.Trigger>Tab 1</Tab.Trigger>
        <Tab.Trigger>Tab 2</Tab.Trigger>
      </Tab.List>
      <Tab.Panels>
        <Tab.Panel>Panel 1</Tab.Panel>
        <Tab.Panel>Panel 2</Tab.Panel>
      </Tab.Panels>
    </Tab.Root>
  );
}
```

しかし Svelte では...

- 1つのファイルにつき 1つのコンポーネント
- `default` エクスポートしかできない

```
<script lang="ts">
  import TabRoot from "./Tab/TabRoot.svelte";
  import Tab from "./Tab/Tab.svelte";
  import TabList from "./Tab/TabList.svelte";
  import TabPanel from "./Tab/TabPanel.svelte";
</script>
```

- import が多くなりがち
- import する側で任意の名前がつけられる

依存関係があるコンポーネントは
Barrel ファイルでまとめよう

Barrel ファイルとは

- 複数のモジュールのエクスポートを 1 つのファイルにまとめる
- Barrel ファイルでは単に他のモジュールをエクスポートするだけ
- 関連がある関数やコンポーネントを 1 つのモジュールから一括で import できる

ディレクトリ構造

```
components/  
  Tab/  
    index.ts  
    TabRoot.svelte  
    Tab.svelte  
    TabList.svelte  
    TabPanel.svelte
```

Tab/index.ts

```
import Tab from "./Tab.svelte";
import TabRoot from "./TabRoot.svelte";
import TabList from "./TabList.svelte";
import TabPanels from "./TabPanels.svelte";
import TabPanel from "./TabPanel.svelte";

Tab.Root = TabRoot;
Tab.List = TabList;
Tab.Panels = TabPanels;
Tab.Panel = TabPanel;

export { Tab };
```

■ でアクセスしたくない場合

```
export { default as Tab } from "./Tab.svelte";
export { default as TabRoot } from "./TabRoot.svelte";
export { default as TabList } from "./TabList.svelte";
export { default as TabPanels } from "./TabPanels.svelte";
export { default as TabPanel } from "./TabPanel.svelte";
```

まとめて import できる

```
<script lang="ts">
  import { Tab } from "./components/Tab";
</script>

<Tab.Root>
  <Tab.List>
    <Tab>Tab 1</Tab>
    <Tab>Tab 2</Tab>
  </Tab.List>
  <Tab.Panels>
    <Tab.Panel>Panel 1</Tab.Panel>
    <Tab.Panel>Panel 2</Tab.Panel>
  </Tab.Panels>
</Tab.Root>
```

Barrel ファイルのデメリット

- 循環 import が発生しやすい
- パフォーマンスの低下
- Barrel ファイルではなく個別のモジュールから import されてしまう
 - 個々の module がプライベートになるわけではない
- Barrel ファイルを禁止する ESLint ルールもある
 - `eslint-plugin-no-barrel-files`

<https://tkdodo.eu/blog/please-stop-using-barrel-files>

まとめ

- UI コンポーネントには依存関係が生まれることがある
- React では Compound Components パターンが使われるが、Svelte では 1 つのファイルに 1 つのコンポーネントしか書けない
- Barrel ファイルを使えば依存関係のあるコンポーネントをまとめることができる
- Barrel ファイルのデメリットもあるので注意が必要