

データ削除後のページ破損原因調査レポート

調査日

2024年（現在）

問題の概要

データを削除した後、`app/analytics/page.tsx` と `app/organization/page.tsx` が壊れてしまった。

調査結果

1. `app/organization/page.tsx` の問題

問題箇所1: nullチェック不足

ファイル: `app/organization/page.tsx`

行番号: 1314, 1319

問題:

```
<OrgChart
  data={filteredOrgData || orgData} // ← 両方がnullの可能性がある
  onClick={handleNodeClick}
/>
```

`filteredOrgData` と `orgData` の両方が `null` になる可能性があるのに、nullチェックなしで `OrgChart` コンポーネントに渡している。

影響:

- `OrgChart` コンポーネントは `data: OrgNodeData` 型を期待しており、nullを許容していない
- `OrgChart.tsx` の96行目で `data.children` に直接アクセスしているため、`data` がnullの場合にエラーが発生する

問題箇所2: `selectedNode`のnullチェック不足

ファイル: `app/organization/page.tsx`

行番号: 1350, 1353, 1369, 1372

問題:

```
{selectedNode && (
  // ...
  router.push(`/organization/detail?id=${selectedNode.id}`); // ←
```

```
selectedNodeがnullでないことは確認済みだが、idがundefinedの可能性はある
})}
```

`selectedNode` はnullチェックされているが、`selectedNode.id` が `undefined` の可能性がある。

影響:

- URLが `/organization/detail?id=undefined` になる可能性がある

問題箇所3: `OrgBubbleChart`へのnull渡し

ファイル: `app/organization/page.tsx`

行番号: 1319

問題:

```
<OrgBubbleChart
  data={filteredOrgData || orgData} // ← 両方がnullの可能性はある
  onClick={handleNodeClick}
/>
```

`OrgBubbleChart` コンポーネントも `data: OrgNodeData` 型を期待しており、nullを許容していない。

影響:

- `OrgBubbleChart.tsx` の73行目で `convertToHierarchy(data, 0)` を呼び出しているため、`data` がnullの場合にエラーが発生する

2. `app/analytics/page.tsx` の問題

問題箇所1: 削除されたデータへの参照

ファイル: `app/analytics/page.tsx`

行番号: 396-400, 429-443, 468-602

問題:

```
// テーマに関連する注力施策を取得
const relatedInitiatives = initiatives.filter((init) =>
  theme.initiativeIds?.includes(init.id) ||
  init.themeId === theme.id ||
  (Array.isArray(init.themeIds) && init.themeIds.includes(theme.id))
);

// このテーマに関連する組織を収集（注力施策から組織IDを取得）
const organizationIds = new Set<string>();
relatedInitiatives.forEach((init) => {
  if (init.organizationId) {
```

```

        organizationIds.add(init.organizationId);
    }
    // ...
});

```

削除された組織や注力施策への参照が残っている可能性がある。

影響:

- `getOrgName` 関数で削除された組織IDを参照すると、組織名が見つからずにIDがそのまま表示される
- 削除された注力施策が `initiatives` 配列に残っていると、存在しない組織へのリンクが作成される

問題箇所2: トピックIDの参照エラー

ファイル: `app/analytics/page.tsx`

行番号: 529-600

問題:

```

parsedTopicIds.forEach((topicId) => {
    const matchingTopics = topics.filter(t => {
        const matches = t.id === topicId;
        // ...
        return matches;
    });

    const topic = matchingTopics.length > 0 ? matchingTopics[0] : null;

    if (topic) {
        // トピックノードを追加
    } else {
        // 見つからなかったトピックIDを記録
        missingTopicIds.add(topicId);
    }
});

```

削除されたトピックIDが `parsedTopicIds` に含まれていると、`topics` 配列から見つからずに警告が出力されるが、処理は続行される。

影響:

- 削除されたトピックへの参照が残っていると、リンクが不完全になる可能性がある
- コンソールに大量の警告が出力される

問題箇所3: 無効なリンクの検出

ファイル: `app/analytics/page.tsx`

行番号: 609-632

問題:

```
// リンクの検証: すべてのリンクのsourceとtargetがノードとして存在するか確認
const nodeIds = new Set(diagramNodes.map(n => n.id));
const invalidLinks: Array<{ source: string; target: string; type?: string }> = [];
diagramLinks.forEach(link => {
  const sourceId = typeof link.source === 'string' ? link.source : link.source.id;
  const targetId = typeof link.target === 'string' ? link.target : link.target.id;
  if (!nodeIds.has(sourceId) || !nodeIds.has(targetId)) {
    invalidLinks.push({
      source: sourceId,
      target: targetId,
      type: link.type,
    });
  }
});
```

無効なリンクは検出されているが、エラーとして記録されるだけで、ページの動作は続行される。

影響:

- 削除されたデータへの参照が残っていると、無効なリンクが作成される
- グラフの表示が不完全になる可能性がある

3. 根本原因

1. データ削除時の参照整合性の問題

- 組織を削除すると、関連する注力施策、トピック、エンティティなども削除される ([src-tauri/src/database/organization.rs](#) の483-518行目を参照)
- しかし、フロントエンド側の状態 ([initiatives](#), [topics](#) など) が即座に更新されない
- 削除されたデータへの参照が残っている状態で、ページが再レンダリングされる

2. nullチェックの不足

- [OrgChart](#) と [OrgBubbleChart](#) コンポーネントがnullを許容していない
- [filteredOrgData](#) || [orgData](#) がnullになる可能性があるのに、nullチェックなしでコンポーネントに渡している

3. データ再取得のタイミング

- データ削除後、組織ツリーは再取得されているが、[initiatives](#) や [topics](#) の再取得が適切に行われていない可能性がある

推奨される修正方法

1. [app/organization/page.tsx](#) の修正

修正1: nullチェックの追加

```
{!orgData ? (  
  <div>組織データがありません</div>  
) : viewMode === 'hierarchy' ? (  
  <OrgChart  
    data={filteredOrgData || orgData}  
    onNodeClick={handleNodeClick}  
  />  
) : (  
  <OrgBubbleChart  
    data={filteredOrgData || orgData}  
    onNodeClick={handleNodeClick}  
    width={1200}  
    height={800}  
  />  
)}
```

修正2: selectedNode.idのnullチェック

```
onNavigateToDetail={() => {  
  if (selectedNode?.id) {  
    router.push(`/organization/detail?id=${selectedNode.id}`);  
  }  
}}
```

2. app/analytics/page.tsx の修正

修正1: 削除されたデータのフィルタリング

```
// 組織データが存在する場合のみ、存在する組織IDの注力施策をフィルタリング  
const validInitiatives = orgData  
  ? initiatives.filter(init => {  
    if (!init.organizationId) return false;  
    // 組織ツリーから組織が存在するか確認  
    const findOrg = (node: OrgNodeData): OrgNodeData | null => {  
      if (node.id === init.organizationId) return node;  
      if (node.children) {  
        for (const child of node.children) {  
          const found = findOrg(child);  
          if (found) return found;  
        }  
      }  
      return null;  
    };  
    return findOrg(orgData) !== null;  
  })
```

```
    })  
    : initiatives;
```

修正2: トピックIDの存在確認

```
// 存在するトピックIDのみを使用  
const validTopicIds = parsedTopicIds.filter(topicId =>  
  topics.some(t => t.id === topicId)  
);
```

3. コンポーネントの修正

OrgChart.tsx と OrgBubbleChart.tsx の型定義を修正

```
export interface OrgChartProps {  
  data: OrgNodeData | null; // nullを許容  
  // ...  
}
```

ただし、nullの場合の処理も追加する必要がある。

まとめ

データ削除後にページが壊れる主な原因は：

1. **nullチェックの不足**: `orgData` や `filteredOrgData` がnullになる可能性があるのに、nullチェックなしでコンポーネントに渡している
2. **削除されたデータへの参照**: 削除された組織、注力施策、トピックへの参照がフロントエンドの状態に残っている
3. **データ再取得のタイミング**: データ削除後、関連データ（`initiatives`, `topics` など）の再取得が適切に行われていない

これらの問題を修正することで、データ削除後もページが正常に動作するようになります。