React SuspenseとNext.js Loading UI

効果的なローディング表示の実装

Suspenseとは？

- React 18から正式サポート
- 子コンポーネントの読み込み完了まで
  フォールバックUIを表示
- 使い方は簡単：Suspenseでラップするだけ

Suspenseの使用例

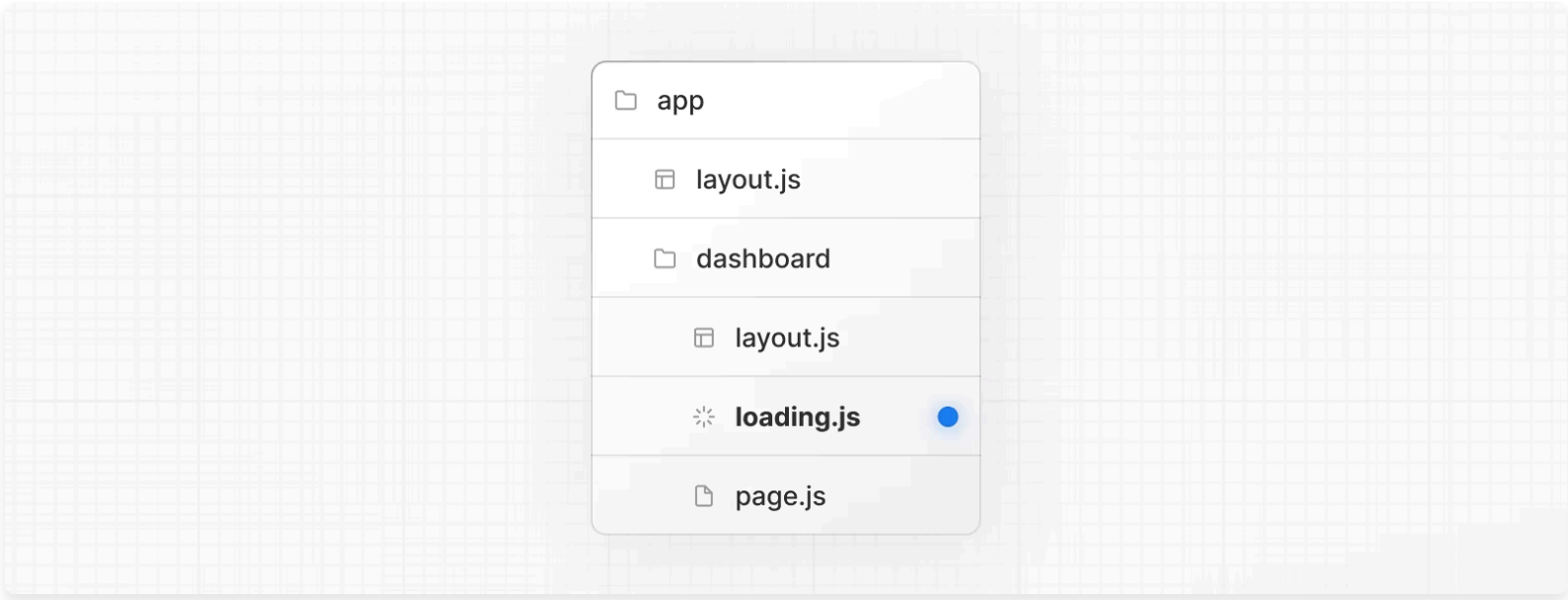
```
import { Suspense } from 'react';

function MyComponent() {
  return (
    Loading ...}>


  );
}
```

詳細: https://ja.react.dev/reference/react/Suspense

Next.js Loading UIとは？



- Next.js 13から導入
- App Routerと併用
- コンポーネントごとにカスタマイズ可能なローディング表示
- Suspenseと連携して動作

Loading UIの使用方法

1. `app`ディレクトリ直下に`loading.tsx`を作成
2. ローディング中に表示したいUIをコンポーネントとして実装
3. 必要に応じて、サブディレクトリにも`loading.tsx`を配置してカスタマイズ

```
// app/loading.tsx
export default function Loading() {
  return Loading...;
}
```

ディレクトリ構造例 {.split-flex}

```
app
├── App1
│   └── page.tsx
├── App2
│   ├── loading.tsx
│   └── page.tsx
├── layout.tsx
├── loading.tsx
└── page.tsx
```

- ルートの `loading.tsx` はデフォルトのローディングUI
- `App2` ディレクトリ内の `loading.tsx` は
  そのルート専用のカスタムローディングUI
- Next.jsが適切なローディングUIを自動的に選択

実装例：カスタムローディングUI

```
// app/App2/loading.tsx
export default function Loading() {
  return (

      Loading App2...

  );
}
```

Next.jsは、`App2`ルートでこのカスタムローディングUIを使用します。

まとめ

- **Suspense**:
    - React 18の機能
    - コンポーネントのローディング状態を簡単に管理
- **Next.js Loading UI**:
    - App Router専用機能
    - ルートごとにカスタマイズ可能
    - Suspenseと連携して動作

両機能を組み合わせることで、
ユーザーエクスペリエンスを大幅に向上させることができます。

参考リンク・サンプルコード {.split-flex}

公式ドキュメント

- React Suspense
- Next.js Loading UI

サンプルコード

GitHubリポジトリ:

Learn-NextJs-LoadingUI

実際の実装例を確認できます！

ご清聴ありがとうございました

効果的なローディング表示で、
アプリケーションのUXを向上させましょう！