作って学ぶ Next.js/React Webサイト構築

エビスコム 著

副読本

Next.js 13 対応ガイド（速報版）

マイナビ

Next.js 13 が 10 月 25 日にリリースされました。そのため、create-next-app でプロジェクトを作成すると、標準で Next.js 13 がインストールされます。

```
$ npx create-next-app blog
Need to install the following packages:
  create-next-app@13.0.0
Ok to proceed? (y)
Creating a new Next.js app in /home/xxxx/blog.
```

```
{
  "name": "blog",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "next": "13.0.0",
    "react": "18.2.0",
    "react-dom": "18.2.0"
  },
  "devDependencies": {
    "eslint": "8.26.0",
    "eslint-config-next": "13.0.0"
  }
}
```

package.json

「作って学ぶ Next.js React Web サイト構築」は、Next.js 12 を前提として解説しているため、今回のバージョンアップによる変更の中で、以下の 2 つの点で注意する必要があります。

- next/image
- next/link

❖ next/image

Next.js 13 では、

- `next/image` → `next/legacy/image`
- `next/future/image` → `next/image`

へと、置き換えられました。本書では、従来の `next/image`、つまり、`next/legacy/image` を前提としたスタイリングをしていますので、`next/legacy/image` をimportして利用する必要があります。

```
import Image from "next/legacy/image"
```

`next/image`（従来の `next/future/image`）を使う場合には、スタイリングの変更が必要です。

❖ next/link

next/link も変更されました。これまで、以下のように <a> を独立させて書かなければならなかった <Link> コンポーネントですが、

```
<Link href="/">
  <a className={boxOn ? styles.box : styles.basic}>CUBE</a>
</Link>
```

Next.js 13 では、以下のように書く形になりました。

```
<Link href="/" className={boxOn ? styles.box : styles.basic}>
  CUBE
</Link>
```

❖ すでに進めている本書のプロジェクトをNext.js 13へアップデート

すでに進めているプロジェクトを Next.js 13 へアップデートする場合は、公式のドキュメントに従って、以下のコマンドで Next.js 13 へアップデートします。

```
$ npm i next@latest react@latest react-dom@latest eslint-config-next@latest
```

この状態で、npm run dev や npm run build を実行すると、エラーや画像の表示が崩れますので、以下のファイルの `next/image` と `next/link` を修正します。

next/imageの修正が必要なファイル

- components/convert-body.js
- components/hero.js
- components/posts.js
- pages/about.js
- pages/blog/[slug].js

next/linkの修正が必要なファイル

- components/logo.js
- components/nav.js
- components/pagination.js
- components/post-categories.js
- components/posts.js

ただし、この修正を行うための codemod が用意されていますので、まとめて修正することができます。

codemod が安全設計なため、このコマンドを実行する前に、git で `commit` か `stash` して、巻き戻せるようにしておく必要があります。

codemods
https://nextjs.org/docs/advanced-features/codemods

```
# プロジェクトのディレクトリに移動した状態で、以下のコマンドを実行します

$ git で commit or stash する
$ npx @next/codemod new-link ./components/

$ git で commit or stash する
$ npx @next/codemod next-image-to-legacy-image ./components/

$ git で commit or stash する
$ npx @next/codemod next-image-to-legacy-image ./pages/
```

以上で変換が完了します。この段階で、Next.js 13 での dev & build が問題なく実行できるようになります。

ただし、components/nav.js を確認してみると、`next/link` が `legacyBehavior` 属性を使う形で、従来のままになっているのが確認できます。`onClick` 属性による処理があるためのようです。

```
      <li>
        <Link href="/" legacyBehavior>
          <a onClick={closeNav}>Home</a>
        </Link>
      </li>
      <li>
        <Link href="/about" legacyBehavior>
          <a onClick={closeNav}>About</a>
        </Link>
      </li>
      <li>
        <Link href="/blog" legacyBehavior>
          <a onClick={closeNav}>Blog</a>
        </Link>
      </li>
    </ul>
  </nav>
```

package.json

これはこのままでも問題はありませんし、以下のように修正することもできます。

```
      <li>
        <Link href="/" onClick={closeNav}>
          Home
        </Link>
      </li>
      <li>
        <Link href="/about" onClick={closeNav}>
          About
        </Link>
      </li>
      <li>
        <Link href="/blog" onClick={closeNav}>
          Blog
        </Link>
      </li>
    </ul>
  </nav>
```

package.json

以上で、Next.js 13 への対応は完了です。

• • •

Next.js 13 で新しく導入された新しい file-system based router である、app ディレクトリに関して、本書のプロジェクトの移行方法なども、この PDF での公開を予定しています（年内に出せればと考えています）。

参照：https://nextjs.org/blog/next-13