Engineer

Next.js

N Webアプリケーションを作るためのReactフレームワーク

どのようなものか

Next.jsは、部品から利用者の画面を作るライブラリーReactを基盤に、Reactが各アプリケーションに任せる部分を提供するフレームワークです。ルーティング、サーバー描画、データ取得、設定済みのビルドなどです。パッケージもコマンドも `next` という名前で、 `next dev` や `next build` として使います。

最初の公開は2016年で、作者の言葉では、「Reactアプリケーションを、サーバーで簡単に描画できるようにするため」でした。初期の設計には、いくつかの原則がありました。設定不要、ファイルシステムをAPIとし、 `pages/` 内のファイルが経路になること、サーバー描画とコード分割の自動化、データ取得は開発者に任せることです。2023年には、リリースノートが、Vercelの中心チームに加え、2,600人を超える開発者の貢献を挙げていました。

Next.js 13は、2022年10月25日に、入れ子のレイアウトとストリーミングを備えた、React Server Componentsに基づく `app` ディレクトリーをベータとして追加しました。2023年5月4日のNext.js 13.4で、App Routerを安定版とし、以来、文書は二つのルーターを説明します。Pages Routerは、「元からあるルーターで、現在もサポートと改善が続いているもの」です。使うReactさえ違います。App Routerは、Reactのcanaryリリースを内蔵し、Pages Routerは、プロジェクトの `package.json` が指定する版を使います。App Routerでは、ファイルが `"use client"` でブラウザー側を選ばない限り、ページとレイアウトはサーバーで描画します。

Next.js 16は、2025年10月21日に登場しました。既定のバンドラーをTurbopackにし、 `middleware.ts` を `proxy.ts` へ改名し、Cache Componentsを導入しました。 `"use cache"` でキャッシュ可能な部分を示し、それ以外はリクエスト時に描画するモデルです。2026年10月6日のNext.js 16.4は、すべてのアプリにCache Componentsを勧め、 `create-next-app` で作る新規プロジェクトでは有効にし、「Next.js 17では既定になる」とします。

概念を判断に生かす

「Next.jsとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

なぜ重要なのか

Next.jsで判断するのは、Reactアプリケーションの経路、描画、サーバーをフレームワークに任せるか、自分で組み合わせて選択権を保つかです。

提供するのは、設定の代わりの規約です。経路はファイルから決まり、バンドラーとコンパイラーは自動で設定し、経路ごとに事前生成、リクエストごとのサーバー描画、ブラウザー描画を選べます。Server Componentsは、データへのアクセスと秘密をサーバーに保ち、ブラウザーへ送るJavaScriptを減らします。公開ページ、ログイン後の領域、独自のバックエンドを混ぜたアプリケーションには、この経路ごとの選択が主な魅力です。

最初の代償は、サーバーです。Next.js 16には、Node.js 20.9以降が必要です。`output: 'export'` なら、どの静的Webサーバーでも使える、経路ごとのHTMLを一ファイルずつ書きます。ガイドは、静的な形から始め、後でサーバー機能を加える方法と紹介します。ただし、Node.jsサーバーが必要な一覧は、Next.jsの特徴の多くを含みます。Proxy、Server Actions、Cookie、書き換え、リダイレクト、ヘッダー、Incremental Static Regeneration、既定のローダーによる画像最適化です。

二つ目の代償は、変化の速さです。16.0の後、一年に四回の機能リリースがあり、キャッシュの既定も、また変わろうとしています。16.4は、Cache Componentsが「以前のApp Routerの版での、暗黙のキャッシュ動作」を置き換えると記します。Next.js 13から15向けのコードとは、キャッシュの規則が違います。ネット上の解説も、二つのルーターに分かれています。

三つ目は、更新がセキュリティーの仕事でもあることです。2025年12月3日には、React Server Componentsプロトコルの重大な脆弱性CVE-2025-66478について、直ちに修正版へ更新するよう通知しました。ブログには、2026年9月30日か

ら10月14日までにも、セキュリティーの記事がさらに三つ載りました。採用するチームは、リリースを密に追う必要があります。誰が読んでも同じ文書を公開するチームは、ほぼ使わないサーバーモデルの代価を払います。

実際の活用例

NextMarinerにおける創業者の仕事では、Next.js 16のフロントエンドが、各領域に合う方法で描画し、経路ごとの選択を実践します。広報用と公開のページは静的に生成し、本当に対話が必要な部分は、クライアント描画を保ちます。認証後のダッシュボードと大きなディレクトリー一覧では、ファクトリー `createPrefetchedServerPage` が、サーバーでページのデータを取得し、あらかじめ揃えた形でクライアントへ渡します。一ページずつファクトリーへ移し、残りの認証後アプリは、まだクライアント描画のまま、移行を待っているとも明記します。「機能する仕組みがあり、次に移すページも明らかな移行であり、全体を終えたものではない」。App Routerも、そのような移動に向けた設計で、13.4のリリースノートは、段階的な採用を説明します。

層別のテスト群は、Next.jsフロントエンドにも、一つの層を与えます。VitestとPlaywrightの543件のテストがあり、48のエンドツーエンド用ファイル、30件のブラウザーテストを含み、Go APIとデータベースには、別の層があります。すべてをブラウザーから試すのではなく、各層の動作がある場所で調べます。コンポーネントツリーが、一部はサーバー、一部はブラウザーで動くフレームワークでは、重要な分け方です。

NextMarinerで仕事をする創業者が、アプリケーションを堅牢化した際、ブラウザー向けの防御は、名前を変えたミドルウェア `proxy.ts` に置きました。リクエストごとのnonceと `strict-dynamic` を使うContent-Security-Policy、HSTS、SameSite Cookieを設定します。認可は別にあります。ティア、ロール、組織、船ごとの範囲を含む利用権限を、リクエストごとにサーバーで確認し直し、フロントエンドの制限は、UXだけとします。この分担は、Proxyを最後の手段と位置付け、「Proxyだけに頼らず、各Server Function内で」認可を確かめるよう求める、Proxyの参照資料にも合います。

問題が起きるところ

ログインのリダイレクトを足すと、ログイン画面のスタイルが消えます。

`matcher` がないと、Proxyは、 `_next/static` のビルド済みファイル、 `_next/image` の最適化画像、 `public/` の全内容も含む、全リクエストで動きます。ページ用のリダイレクトが、ログイン画面に必要なCSS、JavaScript、画像まで捕まえます。その経路を除くmatcherを設定します。

```
// proxy.ts
export const config = {
    matcher: ["/((?!_next/static|_next/image|favicon.ico).*)"],
};
```

`public/` のファイルには共通の先頭部分がないため、それらの経路も加えます。

保護されたように見える経路で、ログアウト中でもフォームの操作が動きます。 Server Functionsには独自の経路がなく、使うページへのPOSTリクエストとして届きます。matcherが経路を対象外にすると、その関数もProxyの対象外になります。参照資料は、matcherの変更や、リファクタリング中の別経路への移動が、「気づかないうちにProxyの保護範囲から外す可能性がある」と警告します。各Server Function内で、セッションと利用者の権限を確認し、Proxyは外側の一層とし、唯一の防御にはしません。

保存しても、ページは古い値のままです。 Next.js 16の `revalidateTag(tag, 'max')` は、タグを付けた項目を、stale-while-revalidateで扱います。新しいコピーを取る間も、キャッシュを返し続けます。結果整合性を許容できる内容には合いますが、保存を押した直後の人を驚かせます。変更したServer Actionで `updateTag(tag)` を呼ぶと、その項目を失効させ、同じリクエスト内で最新データを読みます。一引数の古い `revalidateTag(tag)` は非推奨で、今はTypeScriptエラーになります。

更新用のcodemodが終わっても、 `params` や `cookies()` でビルドが失敗します。 Next.js 15は、リクエストAPIを非同期にし、一時的に同期で使える互換性を残しました。16では、それを除きます。 `upgrade` のcodemodは、すべての移行を実行しないため、 `npx @next/codemod@canary next-async-request-api .` も実行し、次に `npx next typegen` で、 `PageProps` 、 `LayoutProps` 、 `RouteContext` の補助型を生成します。同じ更新で `next lint` も消え、 `next build` もリンターを動かさなくなるため、ESLintかBiomeを独立した手順にします。

参考資料

- Next.jsの文書 — 各ページでApp Router版とPages Router版を切り替えられる、公式文書です。
- Next.js 16 — Cache Components、 `proxy.ts` 、既定のバンドラーTurbopackを導入した版と、互換性を壊す変更です。
- バージョン16へ更新する方法 — codemod、必要な版、削除、改名したすべてのAPIです。

- proxy.js — matcher、リクエスト順でのProxyの位置、ミドルウェアを改名した理由です。
- Next.js 16.4 — Cache Componentsを勧める理由と、Next.js 17で既定にする計画です。
- 静的な書き出しを作る方法 — サーバーが必要な機能と、`output: 'export'` が書くものです。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「Next.jsとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く 連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/nextjs/