

75 Next.js Interview Questions

Fundamentals

- 1. What is Next.js and how does it differ from React?** (Common: Meta, Google, Amazon)
Next.js is a React framework that provides server-side rendering, static site generation, API routes, and other features out of the box. While React is a library focused on UI components, Next.js provides a complete framework with routing, rendering strategies, and optimizations.
- 2. Explain Server-Side Rendering (SSR) in Next.js.** (Common: Vercel, Airbnb, Shopify)
SSR renders pages on the server for each request, sending fully rendered HTML to the client. This improves performance, SEO, and user experience by reducing time-to-first-meaningful-paint. In Next.js, SSR is implemented using the `getServerSideProps` function.
- 3. What is Static Site Generation (SSG) in Next.js?** (Common: Vercel, Stripe, Netflix)
SSG pre-renders pages at build time rather than on each request. It's ideal for pages where content doesn't change frequently. Implemented using `getStaticProps` and optionally `getStaticPaths` for dynamic routes.
- 4. Explain the App Router vs Pages Router in Next.js.** (Common: Vercel, Microsoft, Spotify)
App Router (introduced in Next.js 13) uses a file-system based router built on React Server Components, enabling nested layouts, shared components, and improved server/client rendering separation. Pages Router is the traditional approach with a simpler file-system structure but less advanced features.
- 5. What are the advantages of using Next.js?** (Common: Amazon, Uber, Twitch)
 - Server-side rendering for improved SEO and performance
 - Automatic code splitting
 - File-system based routing
 - API routes
 - Image optimization
 - Zero configuration
 - TypeScript support
- 6. How does routing work in Next.js App Router?** (Common: Vercel, Meta, Airbnb)
App Router uses a file-system based routing where folders define routes and special files define UI. Files like `page.js`, `layout.js`, `loading.js`, and `error.js` serve specific purposes within a route segment.
- 7. What is Incremental Static Regeneration (ISR) in Next.js?** (Common: Vercel, Shopify, Netlify)

ISR allows static pages to be updated after build time without rebuilding the entire site. It's implemented by adding a `revalidate` property to `getStaticProps`, specifying how often the page should be regenerated.

8. **Explain Next.js API routes.** (Common: Stripe, PayPal, Twitter)

API routes allow creating serverless API endpoints as part of a Next.js application. In the Pages Router, they're placed in the `/pages/api` directory. In App Router, they use the Route Handlers with `route.js` files.

9. **What are React Server Components in Next.js?** (Common: Vercel, Meta, Google)

React Server Components render on the server and can access backend resources directly. They don't include client-side JavaScript, reducing bundle size. They're a default feature in Next.js App Router.

10. **How do environment variables work in Next.js?** (Common: Adobe, Spotify, Twilio)

Next.js provides built-in support for environment variables. Files like `.env.local` can store variables, prefixed with `NEXT_PUBLIC_` for client-side exposure. Server-side code can access any environment variable.

Intermediate

11. **How can you optimize images in Next.js?** (Common: Vercel, Cloudflare, Shopify)

Next.js provides the `Image` component that automatically optimizes images with lazy loading, preventing layout shifts, and serving properly sized images based on the device.

12. **What is the `<Link>` component in Next.js?** (Common: LinkedIn, Uber, Airbnb)

The `Link` component enables client-side navigation between routes without full page refreshes. It preloads pages in the background for faster transitions.

13. **Explain Next.js middleware.** (Occasional: Vercel, Auth0, Okta)

Middleware runs before a request is completed, allowing code execution before a page is rendered or an API route is called. It's defined in the `/middleware.js` file and can modify responses, redirect users, or rewrite URLs.

14. **What is the Next.js `<Script>` component?** (Occasional: Google, Vercel, Adobe)

The `Script` component is used to load third-party scripts with various loading strategies (`beforeInteractive`, `afterInteractive`, `lazyOnload`, or `worker`) to optimize performance.

15. **How do you implement internationalization in Next.js?** (Occasional: Airbnb, [Booking.com](https://www.booking.com), Duolingo)

Next.js supports i18n through the built-in internationalized routing in the configuration file. It can detect the user's preferred language and route accordingly. For content translation, libraries like `next-intl` or `react-i18next` are commonly used.

16. **What are Dynamic Imports in Next.js?** (Occasional: Netflix, Twitch, Shopify)

Dynamic imports allow loading JavaScript modules on demand rather than at initial load time. In Next.js, you can use `next/dynamic` to implement code splitting and lazy load components.

```
import dynamic from 'next/dynamic';

const DynamicComponent = dynamic(() => import('../components/hello'));
```

17. **How do you implement authentication in Next.js?** (Common: Auth0, Okta, Microsoft)

Next.js applications can implement authentication using solutions like NextAuth.js, Auth0, or custom implementations using middleware and API routes to protect routes and resources.

18. **What is hydration in Next.js?** (Occasional: Meta, Vercel, Google)

Hydration is the process where React attaches event listeners to the server-rendered HTML, making the page interactive. In Next.js, this happens automatically after the initial HTML is loaded.

19. **How do you handle forms in Next.js?** (Common: Stripe, PayPal, Salesforce)

Forms can be handled using React state and form submissions, or through libraries like React Hook Form or Formik. With Server Actions (Next.js 13+), forms can also submit directly to server functions.

20. **Explain `getInitialProps`, `getStaticProps`, and `getServerSideProps`.** (Common: Vercel, Amazon, Shopify)

- `getInitialProps` : Legacy method for data fetching that runs on both server and client
- `getStaticProps` : Runs at build time for static generation
- `getServerSideProps` : Runs on every request for server-side rendering

21. **What are custom app and document files in Next.js?** (Occasional: Vercel, Twitter, Airbnb)

- `_app.js` : Initializes pages, allows adding global CSS, layouts, and maintaining state
- `_document.js` : Customizes the HTML document structure, useful for adding HTML attributes or custom fonts

22. **How do you implement CSS in Next.js?** (Common: Vercel, Netflix, Uber)

Next.js supports various CSS implementations:

- CSS Modules
- Global CSS
- Styled JSX
- CSS-in-JS libraries (styled-components, emotion)
- Tailwind CSS

23. **What is the purpose of `next.config.js`?** (Common: Vercel, Cloudflare, Microsoft)

The `next.config.js` file is used to customize Next.js behavior, including redirects, rewrites, environment variables, webpack configuration, image optimization, and internationalization settings.

24. **Explain data fetching patterns in Next.js App Router.** (Common: Vercel, Meta, Netflix)

App Router provides several data fetching approaches:

- React Server Components fetch data on the server
- `fetch` API with automatic deduplication
- `use` hook for using Promises in components
- Server Actions for form submissions and mutations

25. **What is the `useRouter` hook in Next.js?** (Common: Uber, Twitter, LinkedIn)

The `useRouter` hook provides access to the router object, allowing programmatic navigation, access to route parameters, query strings, and the current path.

Advanced

26. **How do you implement error handling in Next.js?** (Occasional: Microsoft, Stripe, Twilio)

Next.js provides error handling through:

- `error.js` files in the App Router for route segments
- Custom error pages like `404.js` and `500.js` in the Pages Router
- Try/catch blocks in data fetching functions

27. **What are Next.js rewrites and redirects?** (Occasional: Vercel, Cloudflare, Netlify)

Rewrites and redirects are configured in `next.config.js` :

- Rewrites internally change the URL while keeping the original URL visible to users
- Redirects send users to a different URL and change the URL in the browser

28. **How do you implement streaming in Next.js?** (Occasional: Vercel, Netflix, Twitch)

Streaming allows sending UI pieces progressively to the browser before all data is loaded. In

Next.js, this is implemented using:

- `Loading.js` files in App Router
- The `Suspense` component for wrapping data-fetching components

29. **Explain Edge Runtime in Next.js.** (Rare: Vercel, Cloudflare, Fastly)

Edge Runtime is a lightweight JavaScript environment that allows running code closer to users for improved performance. In Next.js, you can specify Edge Runtime for API routes, middleware, or page rendering.

30. **How do you optimize performance in Next.js?** (Common: Google, Meta, Amazon)

Performance optimization strategies include:

- Using appropriate rendering methods (SSG, ISR, SSR)
- Image and font optimization
- Code splitting
- Bundle analysis
- Lazy loading
- Proper caching strategies

31. **What is Turbopack in Next.js?** (Occasional: Vercel, Microsoft, Shopify)

Turbopack is a Rust-based successor to Webpack that provides faster build times. In Next.js 15, it powers the development server for improved developer experience with faster refreshes and builds.

32. **How do you implement caching in Next.js?** (Occasional: Vercel, Cloudflare, Amazon)

Next.js provides several caching mechanisms:

- Full Route Cache for statically rendered routes
- Router Cache for client-side navigation
- Data Cache for `fetch` requests
- Request Memoization for duplicate data requests in a single render pass

33. **What are Server Actions in Next.js?** (Common: Vercel, Meta, Google)

Server Actions are async functions that execute on the server, allowing form handling and data mutations directly from client components. They're defined using the "use server" directive.

```
// server-action.js
'use server';

export async function submitForm(formData) {
  const name = formData.get('name');
  // Process the data on the server
  return { success: true };
}
```

34. **How do you implement route handlers in Next.js App Router?** (Occasional: Vercel, Stripe, PayPal)

Route handlers in App Router are created with `route.js` files, which export HTTP methods (GET, POST, etc.) as functions. They replace API routes from the Pages Router.

```
// app/api/hello/route.js
export async function GET() {
  return Response.json({ message: 'Hello World' });
}
```

35. **What is Parallel Routes in Next.js?** (Rare: Vercel, Spotify, Airbnb)

Parallel Routes allow rendering multiple pages in the same view simultaneously. They're implemented using the `@folder` naming convention and the slot system in the App Router.

36. **Explain Intercepting Routes in Next.js.** (Rare: Vercel, Meta, Netflix)

Intercepting Routes allow overlaying content on top of the current page. They use the `(.)` syntax to intercept same-level routes, `(..)` for parent routes, or `(..)(..)` for higher-level routes.

37. **How do you implement SEO optimization in Next.js?** (Common: Airbnb, Shopify, [Booking.com](https://www.booking.com))

SEO optimization in Next.js involves:

- Server-side rendering for search engine crawlers
- The Metadata API or `next/head` for titles, descriptions, and meta tags
- Sitemaps and robots.txt generation
- Structured data markup

38. **What is the difference between client and server components in Next.js?** (Common: Vercel, Meta, Google)

- Server Components render on the server with no JavaScript sent to the client, reducing bundle size. They can access server resources but can't use hooks or browser APIs.
- Client Components render on the client, can use interactivity features like hooks, but increase bundle size.

39. **How do you share state between routes in Next.js?** (Occasional: Uber, Twitter, Spotify)

State can be shared between routes using:

- Context API at a high level in the component tree
- State management libraries (Redux, Zustand, Jotai)
- URL parameters and query strings
- `localStorage/sessionStorage` for persistent state

40. **Explain on-demand Incremental Static Regeneration.** (Rare: Vercel, Shopify, Netlify)

On-demand ISR allows invalidating the cache for specific pages programmatically rather than on a time-based schedule, using the `revalidatePath` or `revalidateTag` functions.

Next.js 15 Features

41. **What are the key features of Next.js 15?** (Common: Vercel, Meta, Google)

Key features include:

- Improved Turbopack performance
- Multiple Regions Deployments
- Browser Extension API
- Enhanced partial prerendering
- Improved view transitions
- Server Actions optimizations

42. **What is Partial Prerendering in Next.js?** (Occasional: Vercel, Netflix, Shopify)

Partial Prerendering allows parts of a page to be statically generated while other parts are server-rendered or client-rendered. It combines the benefits of static generation with dynamic content.

43. **Explain View Transitions API in Next.js 15.** (Occasional: Vercel, Meta, Adobe)

View Transitions API enables smooth animations between page navigations. Next.js 15 provides improved support with the `useViewTransition` hook and automatic setup with the `Link` component.

44. **What is multiple regions deployment in Next.js 15?** (Rare: Vercel, Amazon, Cloudflare)

Multiple regions deployment allows hosting apps in different geographic regions simultaneously to reduce latency for users worldwide. Next.js 15 provides built-in support for coordinating deployments across regions.

45. **How does Server Actions optimization work in Next.js 15?** (Occasional: Vercel, Meta, Stripe)

Next.js 15 improves Server Actions with:

- Reduced client-side JavaScript
- Better progressive enhancement
- Optimized form submissions
- Improved error handling

Ecosystem and Integration

46. **How do you implement testing in Next.js?** (Common: Microsoft, Google, Airbnb)

Testing in Next.js typically uses:

- Jest for unit testing
- React Testing Library for component testing
- Cypress or Playwright for end-to-end testing
- Testing utilities from `next/test-utils`

47. **How do you deploy a Next.js application?** (Common: Vercel, AWS, Netlify)

Next.js applications can be deployed on:

- Vercel (optimized for Next.js)
- AWS Amplify or AWS with container services
- Netlify
- Self-hosted servers using Node.js

48. **How do you integrate TypeScript with Next.js?** (Common: Microsoft, Stripe, Airbnb)

Next.js has built-in TypeScript support. Add a `tsconfig.json` file, rename files to `.tsx` or `.ts`, and Next.js automatically sets up the TypeScript configuration.

49. **How do you integrate a CMS with Next.js?** (Occasional: Contentful, Sanity, Shopify)

CMSes can be integrated by fetching data in `getStaticProps`, `getServerSideProps`, or directly in Server Components. Popular headless CMSes used with Next.js include Contentful, Sanity, Strapi, and WordPress.

50. **How do you implement state management in Next.js?** (Common: Meta, Redux, Zustand)

Options for state management include:

- React Context API
- Redux or Redux Toolkit
- Zustand
- Jotai or Recoil
- React Query or SWR for server state

51. **How do you use Next.js with GraphQL?** (Occasional: Apollo, GitHub, Shopify)

Next.js can be integrated with GraphQL using:

- Apollo Client for client-side queries
- Server-side data fetching in `getServerSideProps` or Server Components
- API routes to create GraphQL endpoints

52. **How do you implement file uploads in Next.js?** (Occasional: Cloudinary, AWS, Dropbox)

File uploads can be implemented with:

- Form data and multipart/form-data requests
- API routes with formidable or multer libraries
- Direct uploads to storage services like S3
- Third-party services like Cloudinary or Uploadcare

53. **How do you implement real-time features in Next.js?** (Occasional: Vercel, [Socket.io](https://socket.io/), Pusher)

Real-time features can be added using:

- WebSockets via libraries like [Socket.io](https://socket.io/)
- Server-Sent Events
- Real-time services like Pusher or Ably
- Next.js API routes as endpoints for real-time services

54. **How do you integrate databases with Next.js?** (Common: MongoDB, Prisma, Supabase)

Databases can be integrated using:

- ORM libraries like Prisma or TypeORM
- Native database drivers
- Serverless database services like Supabase or Firebase
- Server Components or API routes for database operations

55. **How do you implement pagination in Next.js?** (Occasional: Meta, Amazon, Shopify)

Pagination can be implemented using:

- Query parameters for page numbers
- Cursor-based pagination in data fetching functions
- Infinite scrolling with SWR or React Query
- Pagination components from UI libraries

Performance and Optimization

56. **How do you analyze bundle size in Next.js?** (Occasional: Google, Meta, Vercel)

Bundle analysis can be done using:

- next/bundle-analyzer plugin
- The built-in build output information
- Chrome DevTools network tab
- Lighthouse reports

57. **What is the purpose of next/dynamic for code splitting?** (Common: Netflix, Spotify, Airbnb)

next/dynamic allows code splitting and lazy loading components to reduce the initial bundle size.

It's especially useful for components that are large or only needed conditionally.

58. **How do you implement infinite scrolling in Next.js?** (Occasional: Twitter, Instagram, Pinterest)

Infinite scrolling can be implemented using:

- Intersection Observer API
- Libraries like react-infinite-scroll-component
- SWR or React Query for data fetching
- Custom hooks with pagination logic

59. **How do you implement PWA features in Next.js?** (Occasional: Google, Uber, Twitter)

PWA features can be added using:

- next-pwa plugin
- Service workers
- Web manifest files
- Offline caching strategies

60. **How do you implement font optimization in Next.js?** (Occasional: Vercel, Google, Adobe)

Next.js provides the `next/font` module for font optimization, including:

- Zero layout shift
- Self-hosting
- Preloading
- Variable fonts support

Debugging and Troubleshooting

61. **How do you debug a Next.js application?** (Common: Microsoft, Meta, Google)

Debugging can be done using:

- Chrome DevTools
- The VS Code debugger
- next/logger or custom logging

- React Developer Tools
- Network tab for API requests

62. **What are common issues with hydration in Next.js?** (Occasional: Vercel, Meta, Netflix)

Common hydration issues include:

- Mismatches between server and client rendering
- Using browser-only APIs in components rendered on the server
- Different content based on user authentication state
- Date or random number generation differences

63. **How do you troubleshoot slow build times in Next.js?** (Occasional: Vercel, Netflix, Airbnb)

Slow build times can be addressed by:

- Analyzing what's being included in the build
- Limiting the number of pages with `getStaticProps`
- Using ISR instead of rebuilding all static pages
- Optimizing dependencies and images
- Using Turbopack (in Next.js 15)

64. **How do you handle CORS issues in Next.js API routes?** (Occasional: Stripe, PayPal, Twilio)

CORS issues can be handled by:

- Setting appropriate headers in API routes
- Using the cors middleware package
- Implementing CORS handling in middleware
- Proxying requests through Next.js API routes

65. **How do you fix "Cannot find module" errors in Next.js?** (Occasional: Vercel, Microsoft, Amazon)

These errors can be fixed by:

- Checking import paths
- Ensuring the module is installed
- Clearing the `.next` cache directory
- Checking for case sensitivity issues
- Verifying `tsconfig` or `jsconfig` paths

Best Practices and Architecture

66. **What folder structure do you recommend for a large Next.js application?** (Common: Vercel, Netflix, Airbnb)

Recommended folder structure:

- `app/` or `pages/` (depending on router)
- `components/` (shared UI components)

- lib/ (utilities and helpers)
- services/ (API clients and data fetching)
- hooks/ (custom React hooks)
- styles/ (global styles and themes)
- public/ (static assets)

67. **How do you implement role-based access control in Next.js?** (Occasional: Auth0, Okta, Microsoft)

RBAC can be implemented using:

- Middleware for route protection
- HOCs to wrap protected components
- Server Components to check permissions server-side
- Context providers for user roles

68. **What are best practices for handling forms in Next.js?** (Common: Stripe, PayPal, Salesforce)

Form handling best practices include:

- Using libraries like React Hook Form for validation
- Implementing Server Actions for form submission
- Proper error handling and user feedback
- Progressive enhancement for non-JS environments
- CSRF protection

69. **How do you implement feature flags in Next.js?** (Occasional: Microsoft, Netflix, Meta)

Feature flags can be implemented using:

- Environment variables
- Remote configuration services
- Database-stored flags fetched during rendering
- Context providers for flag state

70. **How do you handle API rate limiting in Next.js?** (Occasional: Stripe, Twitter, GitHub)

API rate limiting can be implemented using:

- Middleware for incoming requests
- Redis or similar for tracking request counts
- Token bucket algorithms
- Third-party services like Upstash or rate-limiter-flexible

Miscellaneous

71. **What is Next.js Middleware and how is it different from API routes?** (Occasional: Vercel, Auth0, Cloudflare)

Middleware runs before rendering a page or API route, allowing inspection and modification of the request/response. API routes are endpoints that handle specific functionality. Middleware is used for cross-cutting concerns like authentication, while API routes handle specific business logic.

72. How do you implement analytics in Next.js? (Occasional: Google, Meta, Adobe)

Analytics can be implemented using:

- Script component for third-party analytics
- Custom events with the router's events
- Server-side tracking in API routes
- Middleware for page view tracking

73. What is the difference between shallow routing and regular routing in Next.js? (Rare: Vercel, Uber, Twitter)

Shallow routing updates the URL without running data fetching methods, useful for updating query parameters without full page reloads. Regular routing performs a complete navigation with all data fetching.

74. How do you implement dark mode in Next.js? (Occasional: Spotify, Twitter, GitHub)

Dark mode can be implemented using:

- CSS variables with media queries
- Context API for theme state
- localStorage for theme persistence
- next-themes library for SSR compatibility

75. What are the differences between Next.js and similar frameworks like Remix or Gatsby?

(Common: Vercel, Meta, Netlify)

- Next.js vs Remix: Remix focuses on nested routes and progressive enhancement. Next.js offers more rendering strategies and better serverless support.
- Next.js vs Gatsby: Gatsby is more focused on static site generation from external data sources, while Next.js provides more flexible rendering options.
- Next.js vs Astro: Astro is island-based architecture focusing on zero JS by default, while Next.js provides a more complete React framework.