

What is NextJS? All the basic knowledge you should know

NextJS is an open-source React front-end framework that offers optimization features like server-side rendering and static website generation.

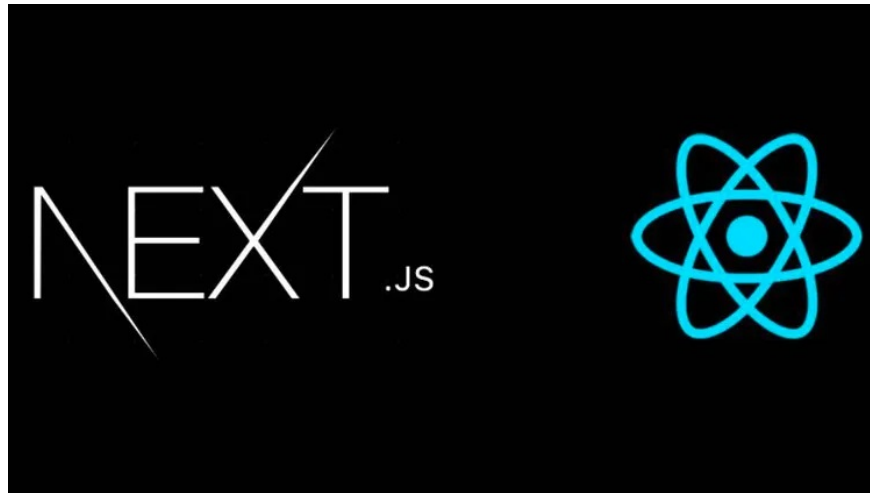


In the web development community, NextJS is known as a framework specialized in creating extremely useful front-end web applications. So what is NextJS, what are the important functions of this framework? Let TipsMake help you find out in the article below!

What is NextJS?

Next.js is an open-source React front-end framework that provides optimized features like server-side rendering and static site generation. Built on top of the React library, Next.js leverages the core React engine **and** adds additional capabilities. SSR helps the server process and render the page with all the necessary data, then sends the entire finished page back to the browser for immediate display. This helps websites load faster, improving the user experience with smoother responsiveness.

Furthermore, using SSR also provides SEO benefits, helping your website rank higher in search results. Websites using SSR load faster and have more website content that can be crawled by SEO tools, contributing to higher rankings. Tags in Next.js also allow editing of page metadata, which is something that React cannot do, helping to improve the SEO of the website.



What is NextJS?

Important Key Features of NextJS

NextJS has the following outstanding features:

Image Optimization

With Image Optimization feature, NextJS helps you automatically optimize images to reduce file size without losing quality. It not only improves page load time but also boosts user experience and SEO.

NextJS uses techniques like lazy loading, image compression, and modern format support. This integration saves developers time and effort compared to manually optimizing images.

Routing support

NextJS uses a file-based routing system, making creating routes simple and intuitive. Just add a JavaScript file to the pages folder, and NextJS will automatically generate the corresponding route.

In addition, NextJS also supports Dynamic Routing and Catch-All Routes, allowing for the creation of complex routes with ease. This is especially useful for applications that require complex and dynamic routing structures.

Analysis mode

With built-in analytics that help you monitor and analyze your application performance. With this analytics mode, you can easily identify performance issues and optimize your application.

NextJS's analysis mode includes tools like Bundle Analyzer, which helps you better understand the structure of your application's bundle and identify which parts need to be optimized.

Automatic compilation and packaging

NextJS uses Webpack under the hood to automatically compile source code and optimize application bundles. This eliminates the need for developers to manually configure complex compilation tools.

This automatic compilation and bundling process helps reduce errors and increase productivity. In addition, NextJS also provides automatic code splitting, which helps pages load faster by loading only the necessary code.

TypeScript configuration support

NextJS automatically detects and configures TypeScript files in your project. This improves the development experience and ensures that your code always complies with TypeScript rules and type checking.

Provides Routing capabilities

NextJS is providing a powerful and flexible routing system. With NextJS routing system, you can easily create static and dynamic routes. This system also supports Nested Routes, which helps you organize the route structure of the application in a reasonable and easy-to-manage way. Besides, the feature also supports Internationalized Routing, which helps to build multilingual applications easily.

Built-in CSS Support

NextJS provides built-in CSS support, making it easier to manage and use CSS in your application. You can use CSS Modules to apply styles locally to components, ensuring that there are no style conflicts between different components.

In addition, NextJS also supports CSS-in-JS tools such as styled-components and Emotion, which help create dynamically styled components easily and efficiently. This built-in support helps optimize the development process and improve application performance.

Optimize SSR and SSG

SSR helps to load pages faster by rendering the page on the server before sending it to the browser. This will greatly improve user experience and optimize SEO.

SSG will allow the generation of static pages before deployment, which will increase the performance and security of the application. You can combine both SSR and SSG in the same NextJS application, creating a flexible and powerful system that meets different project requirements.

Benefits of using NextJS

NextJS is a modern web development framework built on React, which brings significant benefits to developers and businesses.

? **Server-side rendering (SSR):** With SSR, web pages are rendered on the server and sent to the browser as complete HTML, improving page load speed and user experience, especially for web applications with dynamic content.

? **SEO optimization:** Since website content is rendered server-side, search engines can easily crawl and index your website. This is important for e-commerce websites and websites that need good visibility in search results.

? **Optimized application performance:** Only the parts of the application that the user needs are loaded, minimizing file size and increasing page load speed. Furthermore, NextJS supports static website export,

allowing you to easily create static pages from React applications.

? Easy integration with backend systems and APIs: Thanks to NextJS's API routes, you can build API endpoints directly in your application without having to use a separate backend server. This reduces configuration work and speeds up application development.

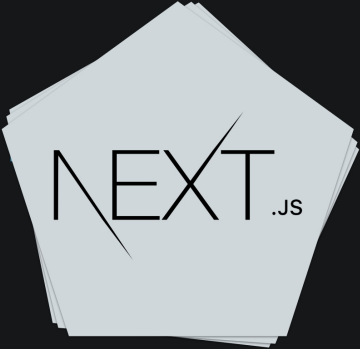
? Strong development community and rich documentation: Developers will find it easy to find support and resolve technical issues. Updates and new features are also released regularly, ensuring that NextJS stays up to date with the latest web technologies.

```
import { useRouter } from 'next/router'

const Header = () => {
  const router = useRouter()

  return (
    <header>
      <button onClick={() =>
        <nav>
          <a href="/">Home</a>
        </nav>
      </header>
    )
  }

  export default Header
```

The image shows a code editor with NextJS code on the left and the NextJS logo on the right. The logo is a light blue pentagon with the text 'NEXT .JS' inside it.

Benefits of using NextJS

Limitations of NextJS

Although NextJS has many advantages, it also has the following limitations:

? Complexity in configuration and deployment: For beginners, understanding and configuring NextJS can be difficult, especially when working with advanced features like SSR or dynamic pagination.

? Learning and migration costs: For developers who are familiar with React, switching to NextJS can require time and effort to get familiar with new concepts and ways of working. It can slow down the initial development process and require additional training for team members.

? Difficulty in working with libraries and tools that are not compatible with SSR: Some JavaScript libraries work well only in the browser and may have problems when used in a server-side environment. This requires developers to find workarounds or adjust the source code to ensure compatibility.

? Performance Optimization: In large applications, performance optimization can be a laborious and time-consuming process. Although NextJS provides many tools to aid in optimization, using them effectively requires a deep understanding of how the framework works and performance optimization techniques.

Difference between NextJS and React

NextJS and React are both powerful web development technologies, but they have some important differences.

React is a JavaScript library for building user interfaces, focused on creating reusable components. It does not dictate how to organize code or manage data, giving developers the freedom to choose and configure additional tools.

In contrast, NextJS is a comprehensive framework built on React, providing a ready-made structure and tools for web application development. It includes features like SSR, static page rendering, routing, and API routes, which simplify the application development and deployment process.

Another big difference is how routing is handled. In React, you need to use an additional library like React Router to manage routing. Meanwhile, NextJS has a built-in routing system that allows you to create new pages by adding files to the pages folder. This makes managing routing in NextJS simpler and more intuitive.

NextJS also supports performance and SEO optimization features that React doesn't provide by default. With NextJS, you can easily implement SSR and static page rendering, which helps improve page load speed and search engine visibility.

Basic NextJS User Guide

To get started with NextJS, you need to install Node.js and npm on your computer. Once installed, you can create a new NextJS project using the following command:

```
npx create-next-app my-nextjs-app
```

```
cd my-nextjs-app
```

```
npm run dev
```

The above command will create a new NextJS project and start the development server. You can access your application at <http://localhost:3000>.

NextJS uses the pages directory to manage the application's pages. Each file in this directory corresponds to a route in the application. For example, to create a new page with the /about path, you just need to create an about.js file in the pages directory:

```
// pages/about.js
```

```
export default function About() {
```

```
  return
```

```
    About Page ;
```

```
}
```

To use SSR in NextJS, you can use the `getServerSideProps` function. This function will run on the server every time the page is requested and return data to render the page:

```
// pages/ssr.js
```

```
export async function getServerSideProps() {  
  // Fetch data from external API  
  const res = await fetch('https://api.example.com/data');  
  const data = await res.json();  
  // Pass data to the page via props  
  return { props: { data } };  
}  
  
export default function SSRPage({ data }) {  
  return  
  {JSON.stringify(data)}  
;  
}
```

To create API routes in NextJS, you just need to create files in the pages/api directory. Each file will correspond to an API endpoint. For example, to create an /api/hello endpoint, you can create a hello.js file in the pages/api directory:

```
// pages/api/hello.js  
  
export default function handler(req, res) {  
  res.status(200).json({ message: 'Hello, world!' });  
}
```

NextJS also supports static page generation using the getStaticProps function. This function runs during the build process and generates static pages:

```
// pages/static.js  
  
export async function getStaticProps() {  
  // Fetch data from external API  
  const res = await fetch('https://api.example.com/data');  
  const data = await res.json();  
  // Pass data to the page via props  
  return { props: { data } };  
}
```

```
}  
  
export default function StaticPage({ data }) {  
  
  return  
  
  {JSON.stringify(data)}  
  
  ;  
  
}
```

Above is all the information related to Nextjs. This is an important framework that helps develop user interfaces on the web more effectively. However, depending on each project, you can consider whether to use it or not to suit the purpose.

You finished reading the article "**What is NextJS? All the basic knowledge you should know**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.