

Mastering Server-Side Rendering (SSR) in Next.js

: A Comprehensive Guide  



Jimmy Ramani
@jimmyramani



What is Server-Side Rendering (SSR)?

Server-Side Rendering (SSR) is a method of rendering web pages on the server and sending fully-rendered HTML to the client's browser. Unlike client-side rendering (CSR), where HTML is generated by JavaScript in the browser, SSR generates HTML on the server, resulting in faster initial page loads and better search engine optimization (SEO).

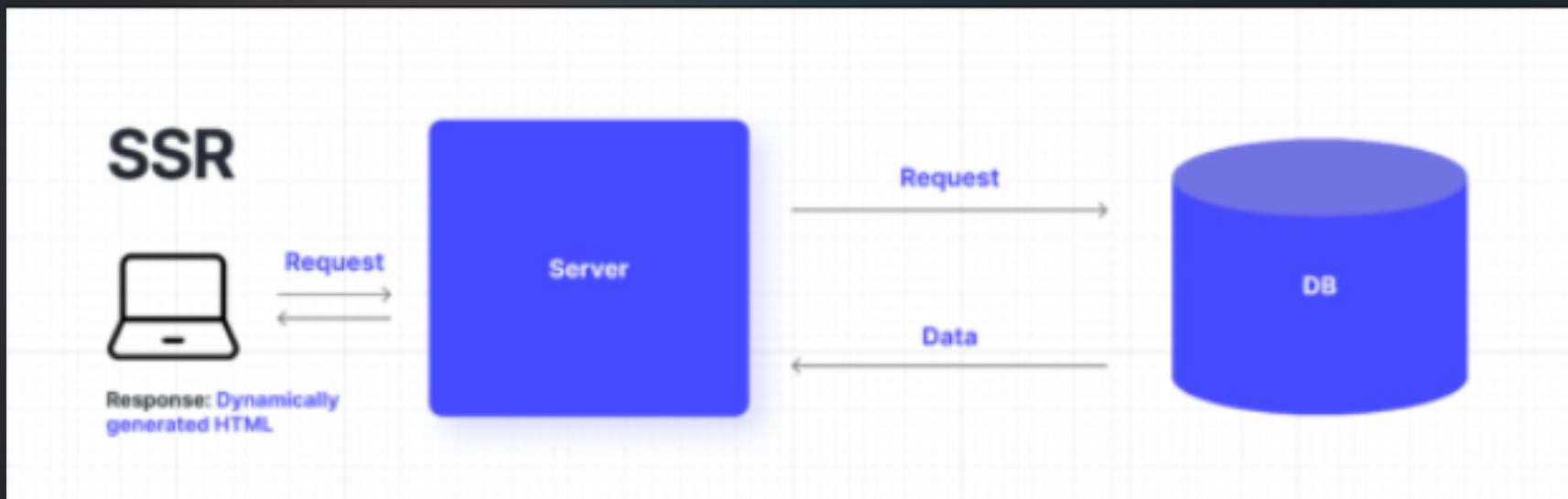


Jimmy Ramani
@jimmyramani



How does SSR work?

SSR involves the server processing a request by fetching data, rendering the HTML template with the data, and sending the fully-rendered HTML page to the client. This approach ensures that the client receives a complete page ready for display, eliminating the need for additional client-side rendering. SSR is commonly used with frameworks like Next.js to simplify the implementation of server-rendered React applications.



Jimmy Ramani
@jimmyramani



Example

- ● ● Let's create a simple example to illustrate CSR in Next.js.

```
1 // pages/index.js
2
3 import { getLatestPosts } from "../lib/posts";
4
5 export default function Home({ posts }) {
6   return (
7     <div>
8       <h1>Latest Posts</h1>
9       <ul>
10         {posts.map((post) => (
11           <li key={post.id}>{post.title}</li>
12         )))
13       </ul>
14     </div>
15   );
16 }
17
18 export async function getServerSideProps() {
19   const posts = await getLatestPosts();
20   return {
21     props: {
22       posts,
23     },
24   };
25 }
```

In this example, the `getServerSideProps` function fetches the latest posts from the server and passes them as props to the `Home` component. This function runs on the server each time a request is made to the page, ensuring that the HTML is generated dynamically with fresh data.



Jimmy Ramani
@jimmyramani



When to use SSR?

► SEO Optimization:

SSR ensures that search engines can crawl and index your content effectively since pages are fully rendered on the server.

► Improved Performance:

SSR can lead to faster initial page loads, especially for content-heavy applications, by delivering pre-rendered HTML directly to the client.

► Authentication and Authorization:

SSR is beneficial for applications that require user authentication and authorization since sensitive data can be securely processed on the server.



Jimmy Ramani
@jimmyramani



Real-life Use Cases :

► E-commerce Platforms:

E-commerce websites often use SSR to render product listings, cart pages, and checkout flows for improved performance and SEO.

► Content Publishing Platforms:

Content publishing platforms like blogs and news websites utilize SSR to render articles and news stories with dynamic content.

► Dashboards and Admin Panels:

Dashboards and admin panels rely on SSR to render user-specific data and provide secure access to sensitive information.



Jimmy Ramani
@jimmyramani





Jimmy Ramani



Join **ME** on a journey of Full Stack Development !

