

Mastering

Incremental Static Regeneration (ISR)

in Next.js

: A Comprehensive Guide 



Jimmy Ramani
@jimmyramani



What is Incremental Static Regeneration (ISR)?

Incremental Static Regeneration (ISR) is a technique introduced by Next.js that combines the benefits of static site generation (SSG) with dynamic content updates. It allows you to pre-render static pages at build time while also enabling them to be updated at runtime without requiring a full rebuild.



Jimmy Ramani
@jimmyramani



How does ISR work?

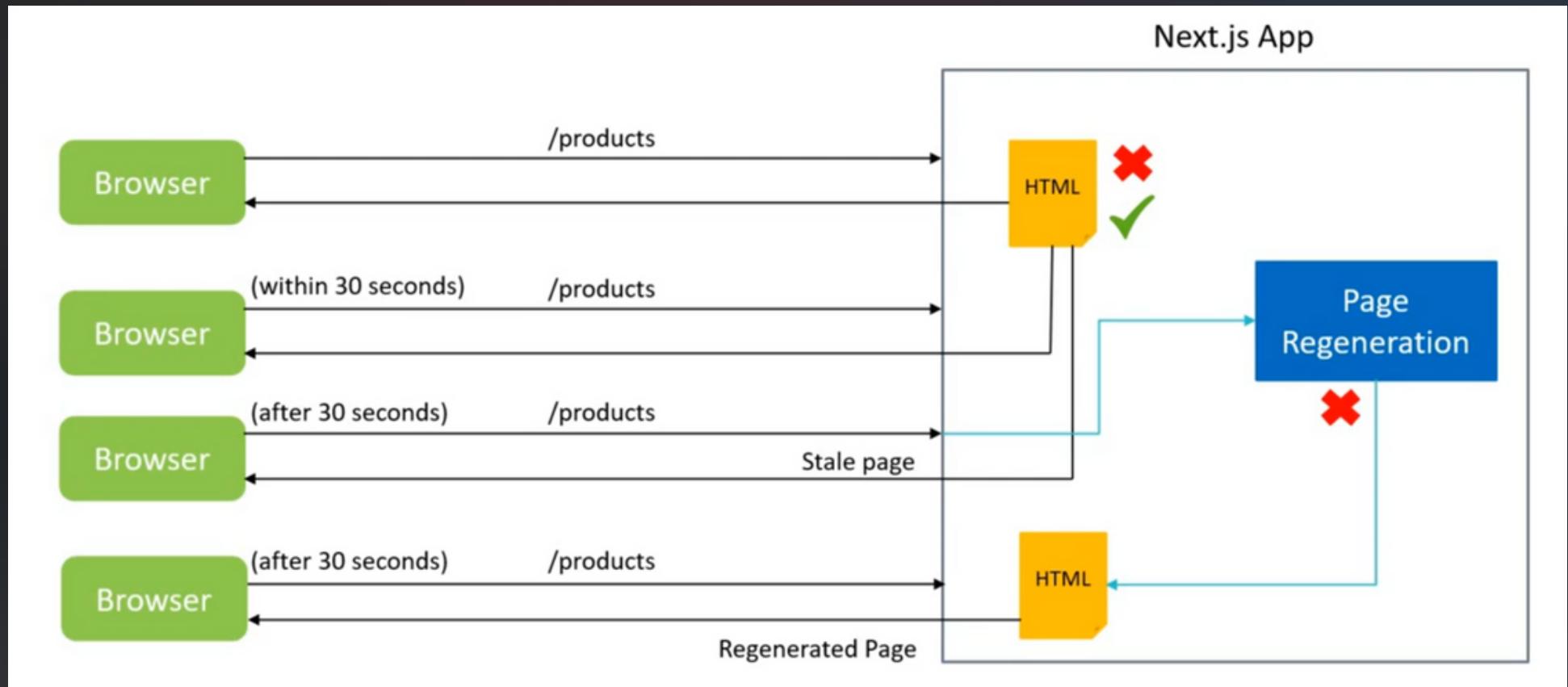
ISR works by allowing you to specify which pages should be statically generated at build time and which pages should be revalidated and regenerated in the background when requested by a user. When a user requests a page that is set up for ISR, Next.js serves the statically generated version of the page while simultaneously initiating a background process to regenerate the page with fresh data. Once the regeneration is complete, the new version of the page is swapped in, providing seamless updates to users.



Jimmy Ramani
@jimmyramani



Stale while revalidate



Jimmy Ramani
@jimmyramani



Example

● ● ● Let's create a simple example to demonstrate ISR in action

using Next.js.

```
1 // pages/posts/[slug].js
2
3 import { useRouter } from 'next/router';
4 import {getPostBySlug} from '../lib/posts';
5
6 export default function Post({ post }) {
7   const router = useRouter();
8
9   if (router.isFallback) {
10     return <div>Loading...</div>;
11   }
12
13   return (
14     <div>
15       <h1>{post.title}</h1>
16       <p>{post.content}</p>
17     </div>
18   );
19 }
20
21 export async function getStaticPaths() {
22   return {
23     paths: [],
24     fallback: true,
25   };
26 }
27
28 export async function getStaticProps({ params }) {
29   const post = await getPostBySlug(params.slug);
30   return {
31     props: {
32       post,
33     },
34     revalidate: 60, // Re-generate page every 60 seconds
35   };
36 }
```

In this example, we have a dynamic route for individual posts. The `getStaticProps` function fetches the post data and specifies that the page should be regenerated every 60 seconds using the `revalidate` option.



Jimmy Ramani
@jimmyramani



When to use ISR?

► Frequently Updating Content:

ISR is ideal for pages with content that changes frequently, such as news articles, blog posts, or product listings.

► Dynamic User Personalization:

If your application requires personalized content for each user, ISR can be used to generate personalized pages while still benefiting from static optimization.

► Improved Performance:

ISR can significantly improve the performance of your application by serving statically generated pages with near-instantaneous load times.



Jimmy Ramani
@jimmyramani



Real-life Use Cases :

- ▶ E-commerce Websites:

Product pages often require real-time updates for stock availability, pricing, and promotions.

- ▶ News Portals:

News articles need to be updated frequently to provide the latest information to users.

- ▶ Blogs and Content Websites:

Blogs and content-heavy websites can benefit from ISR to serve fresh content to users while maintaining fast load times.



Jimmy Ramani
@jimmyramani





Jimmy Ramani



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

