

Data Fetching in NextJS

In Next.js, **getStaticProps**, **getServerSideProps**, and **getStaticPaths** are functions used for data fetching and generating pages. Each serves a different purpose depending on the nature of the data and the page being built. Here's a breakdown of each function:

1. getStaticProps

- **Purpose:** Fetches data at build time for Static Site Generation (SSG).
- **When to Use:** Use `getStaticProps` when you want to generate static pages at build time. The data fetched is available at build time and does not change on subsequent requests unless you use Incremental Static Regeneration (ISR).

Example:

```
javascript Copy code

export async function getStaticProps() {
  const res = await fetch('https://api.example.com/data');
  const data = await res.json();

  return {
    props: {
      data,
    },
    revalidate: 10, // Optional: Revalidate data every 10 seconds (ISR)
  };
}

export default function Page({ data }) {
  return (
    <div>
      <h1>Data from getStaticProps</h1>
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </div>
  );
}
```

2. getServerSideProps

- **Purpose:** Fetches data on each request for Server-Side Rendering (SSR).
- **When to Use:** Use `getServerSideProps` when the data required to render the page is dynamic and needs to be fetched on every request. This is useful for pages that need to be up-to-date with the latest data.

Example:

```
javascript Copy code  
  
export async function getServerSideProps() {  
  const res = await fetch('https://api.example.com/data');  
  const data = await res.json();  
  
  return {  
    props: {  
      data,  
    },  
  };  
}  
  
export default function Page({ data }) {  
  return (  
    <div>  
      <h1>Data from getServerSideProps</h1>  
      <pre>{JSON.stringify(data, null, 2)}</pre>  
    </div>  
  );  
}
```

3. getStaticPaths

- **Purpose:** Defines dynamic routes to be pre-rendered at build time when using `getStaticProps`.
- **When to Use:** Use `getStaticPaths` when you have dynamic routes and you want to pre-render pages for specific paths at build time. This function is typically used in combination with `getStaticProps`.

```
export async function getStaticPaths() {
  const res = await fetch('https://api.example.com/items');
  const items = await res.json();

  const paths = items.map((item) => ({
    params: { id: item.id.toString() },
  }));

  return {
    paths,
    fallback: false, // or 'blocking' or 'true'
  };
}

export async function getStaticProps({ params }) {
  const res = await fetch(`https://api.example.com/items/${params.id}`);
  const data = await res.json();

  return {
    props: {
      data,
    },
  };
}

export default function ItemPage({ data }) {
  return (
    <div>
      <h1>Item {data.id}</h1>
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </div>
  );
}
```

Key Differences:

- **getStaticProps:** Runs at build time. Use for static content.
- **getServerSideProps:** Runs on every request. Use for dynamic content that needs to be up-to-date.

- **getStaticPaths:** Works with `getStaticProps` to generate dynamic routes at build time.

These functions are integral to how Next.js handles data fetching and rendering, giving you flexibility in optimizing performance and user experience.



Srb IT Solution

Convert your ideas into Application
