

JS

What is NEXT.js



Saad Irfan
@DevWithSaad



Next.js

Next.js is a popular React framework that enables server-side rendering (SSR) and static site generation (SSG) for React applications.

It provides an easy-to-use and powerful setup for building server-rendered React applications with great performance and SEO benefits.



Saad Irfan
@DevWithSaad



Installation



You can create a new Next.js project by using the following command:

```
● ● ●  
npx create-next-app my-next-app  
cd my-next-app
```



Saad Irfan
@DevWithSaad



Basic Structure

Next.js v12 follows a conventional folder structure:

```
my-next-app
├── pages
│   ├── index.js
│   └── about.js
├── public
│   └── images
├── components
│   ├── Header.js
│   └── Footer.js
├── styles
└── global.css
```



Saad Irfan

@DevWithSaad



Basic Structure II



- **pages:** Each file inside this folder becomes a route accessible in the application.
- **public:** Contains static assets like images, which can be used in pages.
- **components:** Reusable components used across multiple pages.
- **styles:** Global styles and CSS files.





Basic Page

To create a basic page, simply create a new file inside the pages directory:

```
index.js

// pages/index.js

import React from 'react';

const HomePage = () => {
  return (
    <div>
      <h1>Welcome to Next.js</h1>
      <p>This is a basic Next.js page.</p>
    </div>
  );
};

export default HomePage;
```



Saad Irfan

@DevWithSaad





Dynamic Routing

Next.js allows you to create dynamic routes using brackets []. For example, to create a dynamic post page:

```
[postId].js

// pages/posts/[postId].js

import React from 'react';
import { useRouter } from 'next/router';

const PostPage = () => {
  const router = useRouter();
  const { postId } = router.query;

  return (
    <div>
      <h1>Post ID: {postId}</h1>
      { /* Fetch and display post details using postId */ }
    </div>
  );
};

export default PostPage;
```



Saad Irfan
@DevWithSaad



Server-side Rendering (SSR)

Next.js supports server-side rendering, allowing you to fetch data and render pages on the server before sending them to the client:

```
ssr.js

// pages/ssr.js
import React from 'react';

const SSRPage = ({ serverData }) => {
  return (
    <div>
      <h1>Server-Side Rendering</h1>
      <p>Data fetched on the server: {serverData}</p>
    </div>
  );
};

export async function getServerSideProps() {
  const serverData = 'This data was fetched on the server-side.';
  return {
    props: {
      serverData,
    },
  };
}

export default SSRPage;
```



Saad Irfan

@DevWithSaad



API Routes



Next.js allows you to create API routes, providing a simple way to create custom serverless API endpoints:

```
hello.js

// pages/api/hello.js

export default function handler(req, res) {
  res.status(200).json(
    { message: 'Hello, this is an API endpoint!' }
  );
}
```



Saad Irfan
@DevWithSaad

