



5

Navigating Between Pages

In the previous chapter, you created the dashboard layout and pages. Now, let's add some links to allow users to navigate between the dashboard routes.

In this chapter...

Here are the topics we'll cover



How to use the `next/link` component.



How to show an active link with the `usePathname()` hook.



How navigation works in Next.js.

Why optimize navigation?

To link between pages, you'd traditionally use the `<a>` HTML element. At the moment, the sidebar links use `<a>` elements, but notice what happens when you navigate between the home, invoices, and customers pages on your browser.

Did you see it?

There's a full page refresh on each page navigation!

The `<Link>` component

In Next.js, you can use the `<Link />` Component to link between pages in your application.

Chapter 5

Navigating Between Pages

To use the `<Link />` component, open `/app/ui/dashboard/nav-links.tsx`, and import the `Link` component from `next/link`. Then, replace the `<a>` tag with `<Link>`:

```
TS /app/ui/dashboard/nav-links.tsx

1  import {
2    UserGroupIcon,
3    HomeIcon,
4    DocumentDuplicateIcon,
5  } from '@heroicons/react/24/outline';
6  import Link from 'next/link';
7
8  // ...
9
10 export default function NavLinks() {
11   return (
12     <>
13       {links.map((link) => {
14         const LinkIcon = link.icon;
15         return (
16           <Link
17             key={link.name}
18             href={link.href}
19             className="flex h-[48px] grow items-center justify-center gap-2 rounded
20           >
21             <LinkIcon className="w-6" />
22             <p className="hidden md:block">{link.name}</p>
23           </Link>
24         );
25       })}
26     </>
27   );
28 }
```

As you can see, the `Link` component is similar to using `<a>` tags, but instead of ``, you use `<Link href="...">`.

Save your changes and check to see if it works in your localhost. You should now be able to navigate between the pages without seeing a full refresh. Although parts of your application are rendered on the server, there's no full page refresh, making it feel like a web app. Why is that?

Automatic code splitting and prefetching

Chapter 5

Navigating Between Pages

route segments. This is different from a traditional React [SPA ↗](#), where the browser loads all your application code on initial load.

Splitting code by routes means that pages become isolated. If a certain page throws an error, the rest of the application will still work.

Futhermore, in production, whenever `<Link>` components appear in the browser's viewport, Next.js automatically **prefetches** the code for the linked route in the background. By the time the user clicks the link, the code for the destination page will already be loaded in the background, and this is what makes the page transition near-instant!

Learn more about [how navigation works](#).



It's time to take a quiz!

Test your knowledge and see
what you've just learned.

What does Next.js do when a `<Link>` component appears in the browser's viewport in a production environment?

- A** Downloads additional CSS
- B** Preloads images
- C** Prefetches the code for the linked route
- D** Enables lazy loading for the linked route

Check Answer

Chapter 5

Navigating Between Pages

A common UI pattern is to show an active link to indicate to the user what page they are currently on. To do this, you need to get the user's current path from the URL. Next.js provides a hook called `usePathname()` that you can use to check the path and implement this pattern.

Since `usePathname()` [↗] is a hook, you'll need to turn `nav-links.tsx` into a Client Component. Add React's `"use client"` directive to the top of the file, then import `usePathname()` from `next/navigation`:

`TS` /app/ui/dashboard/nav-links.tsx

```
1  'use client';
2
3  import {
4    UserGroupIcon,
5    HomeIcon,
6    InboxIcon,
7  } from '@heroicons/react/24/outline';
8  import Link from 'next/link';
9  import { usePathname } from 'next/navigation';
10
11  // ...
```

Next, assign the path to a variable called `pathname` inside your `<NavLinks />` component:

`TS` /app/ui/dashboard/nav-links.tsx

```
1  export default function NavLinks() {
2    const pathname = usePathname();
3    // ...
4  }
```

You can use the `clsx` library introduced in the chapter on [CSS styling](#) to conditionally apply class names when the link is active. When `link.href` matches the `pathname`, the link should be displayed with blue text and a light blue background.

Here's the final code for `nav-links.tsx`:

Chapter 5

Navigating Between Pages

```
1  'use client';
2
3  import {
4    UserGroupIcon,
5    HomeIcon,
6    DocumentDuplicateIcon,
7  } from '@heroicons/react/24/outline';
8  import Link from 'next/link';
9  import { usePathname } from 'next/navigation';
10 import clsx from 'clsx';
11
12 // ...
13
14 export default function NavLinks() {
15   const pathname = usePathname();
16
17   return (
18     <>
19       {links.map((link) => {
20         const LinkIcon = link.icon;
21         return (
22           <Link
23             key={link.name}
24             href={link.href}
25             className={clsx(
26               'flex h-[48px] grow items-center justify-center gap-2 rounded-md k
27             {
28               'bg-sky-100 text-blue-600': pathname === link.href,
29             },
30           )}
31         >
32           <LinkIcon className="w-6" />
33           <p className="hidden md:block">{link.name}</p>
34         </Link>
35       )};
36     )}
37   </>
38 );
39 }
```

Save and check your localhost. You should now see the active link highlighted in blue.

Chapter 5

Navigating Between Pages



You've Completed Chapter 5

You've learned how to link between pages and leverage client-side navigation in Next.js.

Next Up

6: Setting Up Your Database

Let's create a database to start fetching real data!

Start Chapter 6

Was this helpful?



Resources

- Docs
- Learn
- Showcase
- Blog
- Analytics
- Next.js Conf
- Previews

More

- Next.js Commerce
- Contact Sales
- GitHub
- Releases
- Telemetry
- Governance

About Vercel

- Next.js + Vercel
- Open Source Software
- GitHub
- Twitter

Legal

- Privacy Policy

Subscribe to our newsletter

Stay updated on new releases and features, guides, and case studies.



Chapter 5
Navigating Between Pages

