

Review

1 min

Great job learning about routing in Next.js. In this lesson, you've learned about:

- the Next.js file-system App Router and how it creates a tree data structure
- the roles of folders and reserved files in building URL segments
- the shared UI reserved file layout.tsx and how it preserves state on nested segment navigation
- the shared UI reserved file template.tsx and how it gets re-instantiated on nested segment navigation
- creating nested URL segments by nesting folders
- creating dynamic URL segments and how to use the params prop to retrieve the data
- using <Link> components for declarative navigation
- using the useRouter() hook for programmatic navigation
- creating catch-all and optional catch-all dynamic routes
- error.tsx, loading.tsx, and not-found.tsx reserved files and their specific use cases
- the Next.js component hierarchy and how it determines how things are rendered in the UI

In this lesson, we covered the core routing concepts in Next.js. There are many more features you can learn about that build off of the concepts you've learned in this lesson.

Instructions

Feel free to continue experimenting with our lesson application. Some possible ideas are:

- auto redirecting to another URL path when users navigate to the root / segment
- adding an error.tsx to catch errors
- moving the not-found.tsx to other segments and seeing how that changes what gets rendered
- applying active styling to <Link> components when the path contains child segments

Run the code using `npm run dev` in the terminal. Once you see a console message indicating "Ready," click the Refresh Browser button in the workspace browser bar.