

File-based Routing

3 min

When exploring a web application, content is served to us based on the URL (Uniform Resource Locator). This is known as **routing**.

Remember that routing uses the URL's *path*, which comes after the URL *domain* and is composed of URL *segments* delimited by forward slashes (/) to display content. For example, in the following URL:

<https://www.codecademy.com/catalog/subject/artificial-intelligence>

- The domain is codecademy.com.
- The path is /catalog/subject/artificial-intelligence.
- The segments are /catalog, /subject, and /artificial-intelligence.

Next.js uses these same concepts to help us build and organize our web application in its *file-based* routing system. In Next.js file-based routing, a *folder* determines a URL path segment, and *reserved files* determine what content is displayed.

We can visualize this routing structure as a *tree* where:

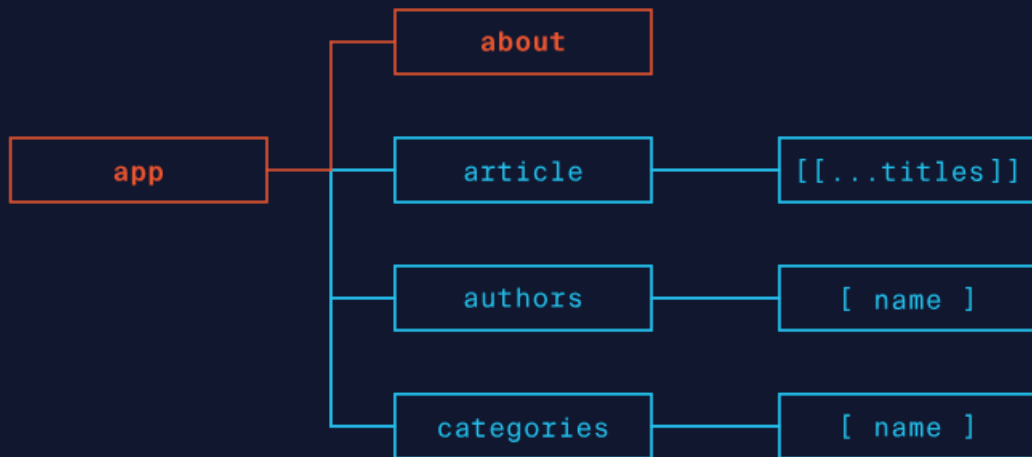
- Each folder is a *node* in the tree.
- The topmost folder is the *root* node.
- Each subfolder is the beginning of a *subtree*.
- A folder is a *leaf* node if it contains no subfolders.

Recall that we created a structure like this when we added page.tsx to the /app folder, as Next.js creates an **App Router** for us when we add an /app folder to our application, serving as our tree's root.

As we continue this lesson, we will learn more about Next.js reserved files and their hierarchy, as well as creating nested routes, creating dynamic routes, and navigation.

Instructions

Before proceeding, take a look at the provided diagram to visualize a Next.js app's relationship between folders, path segments, and its tree-like structure.



/about