

## The App Router

7 min

For this course, our project configuration will use the **App Router**. The App Router is a new Router available out of the box in the /app directory of your Next.js project. This feature of Next.js tackles a significant problem in React: the lack of a native router implementation. It simplifies web development by providing a powerful and flexible routing system, efficient data fetching, improved performance, and easy adoption in existing applications.

The App Router is a file-system based router where the structure of your /app directory determines the routes and URL paths available for your entire application. We'll explore routing more deeply in the upcoming lesson. For now, we'll explore the /app directory and how it generates basic routes.

In the App Router, each folder name determines a route that exists. To make the route accessible, a page.tsx file must live in the directory.

For example, the root /app folder can be treated as a home page and adding a page.tsx makes it accessible; the page will

Preview: Docs Loading link description

[map](#)

to the / URL path. The UI of the home page, or any pages in a route, is contained in the page.tsx file.

The content of page.tsx would look like this:

```
export default function Page() {  
  return <h1>Hello, World!</h1>  
}
```

This is a default exported Page component that will hold the UI for / path, which will contain a heading that says "Hello, World!"

### Instructions

1. Checkpoint 1 Passed

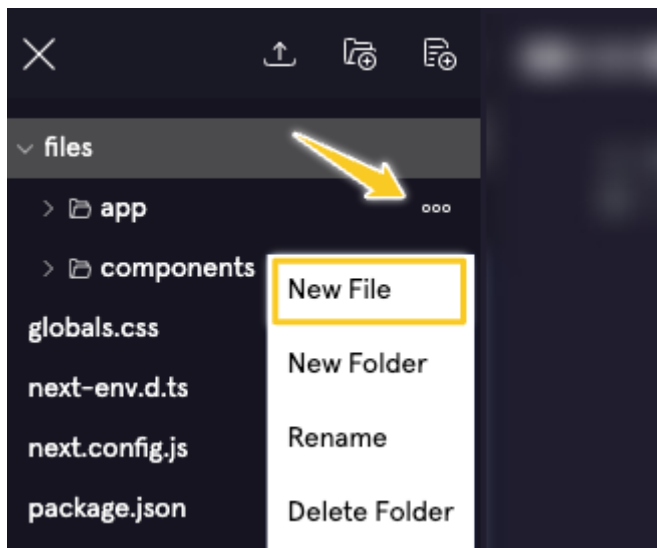
#### 1.

In the file navigation, create a page.tsx file in the /app directory with the add file button.

Hint

The file navigation pane is on the top left of the code editor.

Reference the following to add a new file to the /app directory:



Ensure your page.tsx file is *in* the /app directory.

2. Checkpoint 2 Passed

2.

Within page.tsx, define a component named Page. The component should return an `<h1>` with the text "Hello, Next.js!".

Then, default export the component.

Hint

The Page component can be declared with the following syntax:

```
function Page() {}
```

3. Checkpoint 3 Passed

3.

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.

Notice that the URL mapped to this page is the `/` path found in `localhost:4001/`.

Hint

Enter the command inside the terminal window.

The window can be refreshed when it displays "Ready."

```
> dev
```

```
> next dev -p 4001
```

▲ Next.js 14.0.1

- Local: http://localhost:4001

✓ Ready in 1156ms

### **page.tsx**

```
export default function Page() {  
  return (  
    <h1>Hello, Next.js!</h1>  
  )  
}
```