



LEARNINGS

Next.js Setup & Project Structure: The Ultimate Guide

Ready to start your Next.js journey? This guide walks you through the installation process using create-next-app and demystifies the essential project structure. Learn about the App Router, top-level configuration files, and master advanced routing techniques like dynamic, parallel, and intercepted routes.

[NEXT.JS](#)[REACT](#)[FRONTEND](#)[WEB DEVELOPMENT](#)[JAVASCRIPT](#)[TYPESCRIPT](#)[ROUTING](#)

Ibtissam Oulahchoum

13 September 2026

Getting Started with Next.js Setting up a scalable web application begins with a solid foundation.

This guide covers the essential steps to install Next.js and understand its file architecture.

1. Installation The recommended way to start a new Next.js app is using the automatic setup tool.

Command: Run `npx create-next-app@latest` in your terminal.

Configuration: You will be prompted to configure your project, including choices for TypeScript, ESLint, Tailwind CSS, the `src/` directory, and the App

2. Project Structure: Top-Level Folders Understanding where your code lives is crucial. `app`:

specific to the App Router. `pages`: specific to the classic Pages Router.

public: Stores static assets (images, fonts) to be served.

src: An optional folder to keep your application source code (like components and lib) separate from config files.

3. Essential Configuration Files Your project root will contain several key files:

next.config.js: The main configuration file for Next.js.

middleware.ts: Handles request middleware.

.env files: Manages environment variables for local, production, and general use.

tsconfig.json / jsconfig.json: Configuration for TypeScript or JavaScript.

4. Mastering File-Based Routing Next.js uses the file system to define routes, making navigation intuitive.

Nested Routes: Create subdirectories to represent nested URL paths (e.g., `folder/folder`).

Dynamic Routes: **[folder]**: Captures a variable (e.g., `/user/[id]`).

[...folder]: Catch-all segment for multiple path segments.

[...folder]: Optional catch-all.

5. Advanced Routing Patterns For complex applications, Next.js offers advanced structural tools:

Route Groups (folder): Organize routes without affecting the URL structure.

Private Folders _folder: Exclude folders and their children from routing.

Parallel Routes @folder: Create named slots for complex layouts.

Intercepted Routes: Load a route from another part of your app within the current layout (e.g., `(.)folder` for same level, or `(..)folder` for two levels up).