Next.js with NextAuth.js Project Structure

NextAuth.js integration with protected routes, middleware, and session management.

#nextjs #react #typescript #auth #nextauth

PNG PDF Copy Prompt

Project Directory

```
my-app/
  app/
    layout.tsx
    page.tsx  Public home page
    globals.css
    (public)/  Unauthenticated…
      login/
        page.tsx
      register/
        page.tsx
    (protected)/  Requires authen…
      layout.tsx  Auth guard layo…
      dashboard/
        page.tsx
      profile/
        page.tsx
      settings/
        page.tsx
    api/
      auth/
        [...nextauth]/  Auth.js handler
          route.ts
  components/
    auth/  Auth-specific c…
      login-form.tsx
      register-form.tsx
      sign-out-button.tsx
      user-avatar.tsx
    providers/
      session-provider.tsx  Client session …
    ui/
      button.tsx
      input.tsx
  lib/
    auth.ts  NextAuth config…
    auth.config.ts  Auth options (f…
  types/
    next-auth.d.ts  Session type au…
  middleware.ts  Route protection
  next.config.js
  package.json
  tsconfig.json
  .env.local  AUTH_SECRET, pr…
  .env.example
  .gitignore
```

Why This Structure?

This structure focuses on authentication with NextAuth.js (Auth.js v5). Route groups separate public and protected pages. Middleware handles route protection at the edge, before pages render. The split config pattern (`auth.ts` + `auth.config.ts`) enables middleware usage without importing heavy dependencies.

Key Directories

app/(public)/ - Login, register—accessible without auth

app/(protected)/ - Dashboard, profile—require authentication

components/auth/ - Login forms, sign-out button, user avatar

lib/auth.ts - NextAuth handlers and `auth()` helper

lib/auth.config.ts - Auth options for edge middleware

Authentication Flow

middleware.ts - Runs on edge, checks session, redirects

lib/auth.config.ts - Providers and callbacks (edge-compatible)

lib/auth.ts - Full NextAuth export with adapters

components/providers/session-provider.tsx - Client-side session context

Middleware Pattern

```ts
// middleware.ts
export { auth as middleware } from "@/lib/auth"

export const config = {
  matcher: ["/((?!api|_next|.*\..*).*)"]
}
```

Getting Started

```bash
npm install next-auth@beta
npx auth secret  # generates AUTH_SECRET
# Add provider keys to .env.local
npm run dev
```

When To Use This

- Apps requiring user login/logout
- Multi-provider auth (Google, GitHub, email)
- Session-based access control
- Projects not needing a full database yet

Trade-offs

No database adapter - Add Prisma adapter for persistent sessions

Split config pattern - Two files needed for edge middleware

Beta version - Auth.js v5 API may have breaking changes