

Authentication

Terms

Authentication session

Database adapter

JSON Web Token (JWT)

Middleware

Next Auth

Summary

- NextAuth.js is a popular authentication library for Next.js applications. It simplifies the implementation of secure user authentication and authorization. It supports various authentication providers (eg Google, Twitter, GitHub, Credentials, etc).
- When a user signs in, Next Auth creates an *authentication session* for that user. By default, authentication sessions are represented using *JSON Web Tokens* (JWTs). But sessions can also be stored in a database.
- To access the authentication session on the client, we have to wrap our application with SessionProvider. This component uses React Context to pass the authentication session down the component tree. Since React Context is only available in client components, we have to wrap SessionProvider with a client component.
- Using *middleware* we can execute code before a request is completed. That's an opportunity for us to redirect the user to the login page if they try to access a private part of our application without having a session. Next Auth includes built-in middleware for this purpose.
- Next Auth comes with many *database adapters* for storing user and session data.

Setting Up Next Auth

```
export const authOptions: NextAuthOptions = {
  providers: [
    GoogleProvider({
      clientId: process.env.GOOGLE_CLIENT_ID!,
      clientSecret: process.env.GOOGLE_CLIENT_SECRET!,
    })
  ],
  adapter: PrismaAdapter(prisma),
  session: {
    strategy: 'jwt'
  }
};

const handler = NextAuth(authOptions);

export { handler as GET, handler as POST };
```

Auto-generated Endpoints

```
# Sign in
/api/auth/signin

# Sign out
/api/auth/signout
```

Protecting Routes

```
// middleware.ts
export { default } from 'next-auth/middleware';
export const config = { matcher: ["/admin/:path*"] }
```

Accessing the Session

```
// On the client
const { data: session, status } = useSession();

// On the server
const session = await getServerSession(authOptions);
```