





#### Shared Module: ``auth.ts``

- `signToken(payload)`
- `verifyToken(token)`

#### Used in:

- `middleware.ts`
- `AuthGuard` (`page.tsx`)
- `withAuth()` in API route security

### For using middleware - First level security

1. Common [lib/auth.ts](#) file
2. Common [middleware.ts](#) file - uses verifytoken()
3. Login file [api/login.ts](#) - uses signtoken()

### For using guards - for page level security

1. Common [lib/auth.ts](#) file
2. Authguard file - // [lib/authGuard.tsx](#)
3. Roleguard file - // [lib/roleGuard.tsx](#)
4. Using them file - [app/dashboard/page.tsx](#)

### For using withAuth - For API level security

1. Common [lib/auth.ts](#) file
2. [/lib/withAuth.ts](#) file - for common file to use verifytoken()
3. [/app/api/tasks/route.ts](#) - for using withauth file in every api call

**If no role needed...you will skip the role part**

**In nestjs like backend ..its managed if just you add authguard,and roleguard.....they will automatically manage API level security...as you did in HireSmart project in gitHub**