# 🔐 1. AuthService (Authentication Service)

👉 Ye ek Angular service hai jo **login, register, token store, logout** sab handle karti hai.

**Iske main kaam:**

1. **Login & Register**
   * login(email, password) → backend ko request bhejta hai token lene ke liye.
   * register(fullName, email, password, role) → backend pe naya user create karta hai.
2. **Token & Role Store karna**
   * saveAuth(token, role) → login ke baad token & role ko localStorage me save karta hai.
3. **Logout**
   * logout() → localStorage se token & role delete kar deta hai.
4. **Getter Methods**
   * getToken() → localStorage se token nikalta hai.
   * getRole() → user ka role nikalta hai (admin/user).
   * isLoggedIn() → check karta hai user login hai ya nahi (token present hai to true).

👉 Matlab **AuthService ek helper hai jo pura authentication data handle karta hai.**

# 🛡️ 2. AuthGuard & AdminGuard (Route Guards)

👉 Ye Angular ke guards hain jo **route ko protect** karte hain.

**AuthGuard**

* Kaam: **Sirf logged-in users ko hi route access karne deta hai.**
* Agar user login nahi hai → router.navigate(['/login']) se usko login page bhej deta hai.

Use case:

{ path: 'dashboard', component: DashboardComponent, canActivate: [AuthGuard] }

👉 Is route pe sirf login hone ke baad hi jaa sakte ho.

**AdminGuard**

* Kaam: **Sirf admin users ko hi route access karne deta hai.**
* Agar role admin nahi hai → usko /home pe redirect kar deta hai.

Use case:

{ path: 'admin', component: AdminPanelComponent, canActivate: [AdminGuard] }

👉 Is route pe sirf **admin role** wale user hi jaa sakte hain.

# 🌐 3. AuthInterceptor (HTTP Interceptor)

👉 Ye ek interceptor hai jo **har API request me automatically JWT token add karta hai**.

**Kaam:**

1. **Request Modify karna**
   * Agar token localStorage me hai → har request ke header me Authorization: Bearer <token> add karega.
2. **Error Handle karna**
   * Agar backend 401 (Unauthorized) ya 403 (Forbidden) bhejta hai →
     + localStorage se token & role delete kar deta hai.
     + User ko login page pe redirect kar deta hai.

👉 Matlab tumhe manually har request me token nahi bhejna padta, interceptor automatically karta hai.

## 🎯 Overall Flow

1. User **register** karta hai → backend ek user create karta hai.
2. User **login** karta hai → backend se ek **JWT token + role** aata hai.
3. AuthService.saveAuth() token & role ko localStorage me store karta hai.
4. Jab bhi koi **API request** hoti hai → AuthInterceptor us request ke header me automatically token daal deta hai.
5. **Route Guards** ensure karte hain ki:
   * Normal user login ke bina restricted page pe na jaa sake.
   * Normal user admin panel me na ghus sake.

## 👉 Short me:

* **AuthService** = Token & User info manage karta hai.
* **AuthGuard** = Sirf logged-in users ko access deta hai.
* **AdminGuard** = Sirf admin users ko access deta hai.
* **AuthInterceptor** = Har API request me token inject karta hai aur unauthorized errors handle karta hai.

**🔥 JWT Questions & Answers with Code**

**1. JWT token generate kahan hota hai?**

👉 **Answer:**  
Token generate **login ke time** hota hai. Jab user ka email-password verify ho jaata hai, toh backend jwt.sign() function se token banata hai aur client ko bhejta hai.

✅ Tumhare code me token generate /auth/login route ke andar ho raha hai.

const token = jwt.sign(

{ id: user.id, role: user.role }, // payload

JWT\_SECRET, // secret key

{ expiresIn: '1h' } // options (expiry time)

);

🔎 **Breakdown:**

* **Payload:** { id: user.id, role: user.role } → user ki info store hoti hai
* **Secret key:** JWT\_SECRET = "mysecretkey" → sirf server ke paas hoti hai
* **Expiry:** { expiresIn: '1h' } → token 1 ghante ke liye valid hai

**2. JWT verify kaise hota hai?**

👉 **Answer:**  
Jab client koi protected API call karta hai (jaise /profile), toh wo request ke header me Authorization: Bearer <token> bhejta hai.

Server side pe jwt.verify() use hota hai token ko check karne ke liye.

✅ Tumhare code me ye middleware me likha hai:

jwt.verify(token, JWT\_SECRET, (err, decoded) => {

if (err) return res.status(403).json({ message: "Invalid Token" });

req.user = decoded; // decoded payload (id, role)

next();

});

🔎 **Breakdown:**

* token → client se aaya
* JWT\_SECRET → server compare karega
* Agar **match ho gaya** → access milta hai
* Agar **expire/invalid** → 401/403 error

**3. JWT ka structure kaisa hota hai?**

👉 **Answer:**  
JWT 3 parts me hota hai → Header.Payload.Signature

Example:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwicm9sZSI6ImFkbWluIiwiaWF0IjoxNzM2MDYzNzcsImV4cCI6MTczNjA2NzM3fQ.XYZabc123

🔎 **Breakdown:**

1. **Header** → Algorithm info (HS256, RS256 etc.)
2. **Payload** → User data (id, role, exp)
3. **Signature** → Token ki authenticity verify karne ke liye

**4. JWT token ko decode kaise karte ho?**

👉 **Answer:**  
Decode karne ke liye jwt.decode() use karte hain. Isse hume **header + payload** milta hai, lekin signature verify nahi hota.

✅ Tumhare code me /decode route me likha hai:

const decoded = jwt.decode(token, { complete: true });

👉 Example Output:

{

"header": { "alg": "HS256", "typ": "JWT" },

"payload": { "id": 1, "role": "admin", "exp": 1736067370 },

"signature": "kXbc89hJGY..."

}

**5. JWT expire hone ke baad kya hota hai?**

👉 **Answer:**  
Agar token expire ho gaya toh jwt.verify() **error throw** karega.

✅ Tumhare code me error handle likha hai:

if (err) return res.status(403).json({ message: "Invalid Token" });

👉 Matlab client ko dubara login karna padega naya token lene ke liye.

**6. JWT kyu use karte hain session ki jagah?**

👉 **Answer:**

* **Session based auth:** Server pe session store karna padta hai (memory/database use hota hai).
* **JWT based auth:** Server pe kuch store nahi hota, token hi client ke paas hai. Server sirf verify karta hai → **stateless authentication**.

**7. Token kaha store karna chahiye (Frontend)?**

👉 **Answer:**  
2 options hote hain:

1. localStorage → easy but XSS attack possible
2. httpOnly cookie → secure option

✅ Tumhare Angular code me token ko localStorage.setItem("auth\_token", token) me save kiya gaya hai.

**8. JWT refresh token kya hota hai?**

👉 **Answer:**

* **Access Token:** Short expiry (1 hr)
* **Refresh Token:** Long expiry (7 days, 30 days)
* Jab access token expire hota hai → refresh token se naya access token generate hota hai → user ko baar baar login nahi karna padta.

Code Example:

const refreshToken = jwt.sign({ id: user.id }, JWT\_SECRET, { expiresIn: "7d" });