
🔐 Full Workflow: Angular + Node.js + MongoDB Atlas + JWT Authentication

---

🌐 Step 1: Set Up MongoDB Atlas (Cloud Database

- Create a **MongoDB Atlas** account and set up a free cluster.
- Create a **database** (e.g., myApp).
- Add two collections:
    - users
    - posts
- Optionally insert sample documents (or handle this from backend).

⚙️ Step 2: Build Node.js + Express Backend

- Initialize a **Node.js** project with **Express.js**.
- Connect to **MongoDB Atlas** using a connection string.
- Create RESTful API routes for:
    - **User registration** (POST /register)
    - **User login** (POST /login)
    - **Get posts** (GET /posts)
    - **Create post** (POST /posts)
- Add **JWT Authentication** middleware to secure protected routes.

🔑 Step 3: Implement JWT Authentication

- Upon successful login:
    - Backend verifies user credentials from users collection.
    - If valid, generates a **JWT token** and sends it to the frontend.
- Use **middleware** to protect routes:

- ◦ Only authenticated users (with a valid token) can access/post data.

💻 Step 4: Angular Frontend Setup

- Create Angular components:
  - LoginComponent & RegisterComponent
  - PostsComponent (displays posts in cards)
- On login:
  - Angular receives and **stores the JWT token** (e.g., localStorage).
- For secure API calls:
  - Angular includes the token in headers:
    Authorization: Bearer <token>

🔄 Step 5: User Interaction Flow

1. User logs in or registers via Angular frontend.
2. Angular sends data to backend.
3. Node.js backend validates and responds with a JWT token.
4. Angular stores the token securely.
5. Angular sends the token with protected API requests.
6. Node.js backend verifies the token before responding.
7. If token is valid:
   - Data (e.g., posts) is fetched and displayed in Angular UI.

🧱 Step 6: Final System Behavior

🔧 **Component**	📌 **Responsibility**
MongoDB Atlas	Cloud-based database storing users and posts

🔧 **Component**	📌 **Responsibility**
Node.js / Express	Backend server: handles auth, routes, JWT validation
JWT Token	Secure token passed between frontend and backend
Angular Frontend	Handles UI, stores token, and communicates with backend securely

---