Spring Security - Master Level Roadmap

1. Request Flow in Spring Security

Every HTTP request first passes through Spring Security's Filter Chain.

Example Flow:

- Incoming Request -> Spring Security Filters
- FilterChain checks Authentication -> If not authenticated, redirect to /login or return 401
- If authenticated -> Checks Authorization (Role-based access)
- If allowed -> request proceeds to Controller
- If denied -> return 403 Forbidden

2. Core Features Required for Real-World Apps

1. Form Login & OAuth2 (Google, GitHub) - DONE
2. Role-Based Access (RBAC) - ROLE_USER, ROLE_ADMIN
3. CSRF Protection
4. BCrypt Password Encoding
5. Exception Handling (Custom error pages)
6. Remember Me Functionality
7. Session Management (Concurrent session limits)
8. Custom Success / Failure Handlers
9. JWT Security for REST APIs
10. Method Level Security (@PreAuthorize)
11. Audit Logging (track login/logout actions)
12. Two-Factor Authentication (MFA)
13. Security Headers (XSS, Clickjacking prevention)
14. CORS Configuration (for frontend-backend interaction)

3. Learning Path

Spring Security - Master Level Roadmap

Beginner: Form Login, Role-based pages

Intermediate: Session control, custom login flow, error pages

Advanced: JWT tokens, stateless API security

Pro: MFA, Audit logs, Security Events, Cloud Identity integration

4. Tips for Implementation

- Always encode passwords using BCrypt
- Always secure endpoints using role-based rules
- Avoid exposing endpoints without access control
- Use HTTPS in production
- Keep Spring Boot, Security, and OAuth dependencies up-to-date
- Validate input and sanitize user data
- Implement proper logout functionality
- Store secrets securely (not in plain application.properties)