Spring Boot Interview Questions & Answers

Chapter 19: Spring Boot – Security

Q: What is Spring Security in Spring Boot?

Spring Security is a framework that provides authentication, authorization, and protection against common security threats. In Spring Boot, it integrates easily with minimal configuration.

Use-Case / Example: A banking app uses Spring Security to ensure only logged-in users can check their account balance.

Q: How does Spring Boot enable security by default?

When you add the spring-boot-starter-security dependency, Spring Boot automatically secures all endpoints with basic authentication and generates a default username and password.

Use-Case / Example: Running the app and accessing http://localhost:8080/ will prompt for a username and password.

Q: How do you create a custom user login in Spring Boot Security?

You define a custom UserDetailsService that loads user details from a database or in-memory. Then configure it with Spring Security.

Use-Case / Example: A school management system can authenticate students using credentials stored in a database.

Q: What is the difference between authentication and authorization?

- Authentication → verifying the user's identity.
- Authorization → checking what actions the user is allowed to perform.

Use-Case / Example: A user logs in with credentials (authentication), then only admins are allowed to access /admin endpoints (authorization).

Q: How do you secure REST APIs with Spring Boot?

You can secure APIs using HTTP Basic, form login, or JWT (JSON Web Token). JWT is widely used for stateless authentication in REST APIs.

Use-Case / Example: An online store issues a JWT token when a user logs in, and the token must be included in the header for every request.

Q: How do you define role-based access in Spring Boot Security?

You can use annotations like @PreAuthorize("hasRole('ADMIN')") or configure access rules in the security configuration.

Use-Case / Example: In a hospital system, only doctors (ROLE_DOCTOR) can access patient records, while receptionists (ROLE_RECEPTIONIST) can only book appointments.

Q: How do you configure password encoding in Spring Security?

Spring Security requires passwords to be encoded using a PasswordEncoder like BCryptPasswordEncoder.

Use-Case / Example: When a user registers, the password is stored in the database in an encrypted form instead of plain text.

Q: How do you disable default security in Spring Boot?

You can exclude spring-boot-starter-security dependency or override the security configuration to allow public access.

Use-Case / Example: A public blog’s homepage may be accessible without authentication, while other pages remain secured.

Q: What is CSRF protection in Spring Boot Security?

CSRF (Cross-Site Request Forgery) protection prevents malicious websites from tricking users into performing actions unknowingly. Spring Security enables it by default for web apps.

Use-Case / Example: A banking app ensures that only requests with a valid CSRF token can transfer money, blocking fraudulent requests.

Q: Why is Spring Security important in real-world applications?

It provides built-in mechanisms to handle common security requirements like login, roles, token-based authentication, and attack prevention, saving developers time and effort.

Use-Case / Example: An enterprise HR system uses Spring Security for employee logins, restricting access based on departments and roles.