1. What is Spring Security?

Answer: Spring Security is a powerful authentication and access control framework for Java applications.

It provides comprehensive security services for Java EE-based enterprise software applications. The

primary goal of Spring Security is to secure Java applications at both the method and the request level.

Spring Security is a Java framework that helps secure applications by providing authentication,

authorization, and protection against common security issues. It simplifies the implementation

of secure features in Spring-based applications.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2. Explain Spring Security Oauth2

Answer: Spring Security OAuth2 is an extension of the Spring Security framework that facilitates

secure authorization and authentication in web applications through OAuth 2.0. It allows applications

to delegate user authentication to an external identity provider (like Google or Facebook) and manage

access tokens for secure resource sharing. Spring Security OAuth2 simplifies the integration of

OAuth 2.0 authentication and authorization into Java-based applications, providing a robust and

standardized security solution.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3. What are some essential features of Spring Security?

Answer: Some essential features of Spring Security include:

\* Supports authentication and authorization in a flexible and comprehensive manner.

\* Detection and prevention of attacks including session fixation, clickjacking, cross-site request forgery, etc.

\* Integrate with Servlet API.

\* Offers optional integration with Spring Web MVC (Model-View-Controller).

\* Java Authentication and Authorization Service (JAAS) is used for authentication purposes.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

4. What is Spring security authentication and authorization?

Answer:

\* Authentication: This refers to the process of verifying the identity of the user, using the

credentials provided when accessing certain restricted resources. Two steps are involved

in authenticating a user, namely identification and verification. An example is logging into

a website with a username and a password. This is like answering the question Who are you?

\* Authorization: It is the ability to determine a user's authority to perform an action or to

view data, assuming they have successfully logged in. This ensures that users can only

access the parts of a resource that they are authorized to access. It could be thought

of as an answer to the question Can a user do/read this?

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5. What do you mean by session management in Spring Security?

Answer: Session management in Spring Security refers to the handling and control of

user sessions in a web application. It involves mechanisms for user authentication,

authorization, and tracking user sessions to ensure secure and seamless interactions.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6. Explain SecurityContext and SecurityContext Holder in Spring security ?

Answer: There are two fundamental classes of Spring Security: SecurityContext and SecurityContextHolder.

1. SecurityContext:

In this, information/data about the currently authenticated user (also known as the principal)

is stored. So, in order to obtain a username or any other information about the user, you must

first obtain the SecurityContext.

2. SecurityContextHolder:

Holds the SecurityContext for the duration of a request, providing a way to access the security

information associated with the current thread.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7. Explain spring security OAuth2 ?

Answer: Spring Security OAuth2 is an extension of the Spring Security framework that provides

support for OAuth 2.0 authentication and authorization for secure user delegation and integration

with third-party providers (like Google or Facebook) and enables the creation of OAuth 2.0

compliant authorization servers and resource servers.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8. What is method security and why do we need it?

Answer: Method security in Spring Security refers to the capability of securing individual

methods or functions within a Java application. It allows you to control access to specific

methods based on user roles, permissions, or other criteria.

It's needed for fine-grained access control, ensuring only authorized users can execute

specific operations in an application.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9. What is PasswordEncoder?

Answer: Password encoding is provided by Spring Security using the PasswordEncoder interface.

This interface defines two methods:

encode(): It converts a plain password into an encoded form.

matches(): It compares an encoded password from the database with a plain password (input by the

user) that's been encoded using the same salting and hashing algorithm as the encoded password.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10. What is SpEL (Spring Expression Language)?

Answer: Spring Expression Language (SpEL) is a powerful expression language that allows

dynamic evaluation of expressions at runtime within the Spring Framework. It is used for

configuring and managing beans, defining conditions in annotations, and processing data

during runtime.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11. Name security annotations that are allowed to use SpEL?

Answer: Some security annotations that are allowed to use SpEL include:

@PreAuthorize

@PreFilter

@PostAuthorize

@PostFilter

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12. Explain what is AuthenticationManager in Spring security?

Answer: A Spring Security component called AuthenticationManager tells "How authentication

will happen". Because the how part of this question depends on which authentication provider

we are using for our application, an AuthenticationManager contains references to all the

AuthenticationProviders. AuthenticationManager is the strategy interface for authentication,

which has only one method:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13. Explain what is ProviderManager in Spring security.

Answer: The default implementation of AuthenticationManager is ProviderManager. It does

not handle the authentication request itself, rather delegates the authentication process to a

list of configured AuthenticationProviders.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

14. What is JWT?

Answer: JWT stands for JSON Web Token. It is a compact, URL-safe means of representing

claims between two parties. JWT is commonly used for authentication and authorization in

web development. It consists of a JSON object encoded as a string, and it can be signed to

verify its integrity. JWTs are often used to securely transmit information between parties,

such as user authentication data.