Homework 5 Berk Balcı

1- What are Authentication and Authorization ?

In simple terms, authentication is the process of verifying who a user is, while authorization is the process of verifying what they have access to.

2- What is Hashing in Spring Security ?

Hashing is the process of generating a string, or hash, from a given message using a mathematical function known as a cryptographic hash function.

While there are several hash functions out there, those tailored to hashing passwords need to have four main properties to be secure:

1- It should be deterministic: the same message processed by the same hash function should always produce the same hash

2-It's not reversible: it's impractical to generate a message from its hash

3-It has high entropy: a small change to a message should produce a vastly different hash

4-And it resists collisions: two different messages should not produce the same hash

A hash function that has all four properties is a strong candidate for password hashing since together they dramatically increase the difficulty in reverse-engineering the password from the hash.

Although Java natively supports both the PBKDF2 and SHA hashing algorithms, it doesn't support BCrypt and SCrypt algorithms.

But Spring Security ships with support for all these recommended algorithms via the PasswordEncoder interface:

Pbkdf2PasswordEncoder gives us PBKDF2

BCryptPasswordEncoder gives us BCrypt, and

SCryptPasswordEncoder gives us SCrypt

3- What is Salting and why do we use the process of Salting ?

Salting is simply the addition of a unique, random string of characters known only to the site to each password before it is hashed, typically this “salt” is placed in front of each password. The salt value needs to be stored by the site, which means sometimes sites use the same salt for every password.

4- What is “intercept-url” pattern ?

The <intercept-url> element defines a pattern which is matched against the URLs of incoming requests using an ant path style syntax. The access attribute defines the access requirements for requests matching the given pattern.

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

Regarding security, session management relates to securing and managing multiple users' sessions against their request.In most cases, a session is initiated when a user supplies an authentication such as a password.

We can control exactly when our session gets created and how Spring Security will interact with it:

always – A session will always be created if one doesn't already exist.

ifRequired – A session will be created only if required (default).

never – The framework will never create a session itself, but it will use one if it already exists.

stateless – No session will be created or used by Spring Security

6- Why we need Exception Handling ?

Java exception handling is important because it helps maintain the normal, desired flow of the program even when unexpected events occur. If Java exceptions are not handled, programs may crash or requests may fail.

7- Explain what is AuthenticationManager in Spring security ?

Simply the AuthenticationManager is the main strategy interface for authentication. If the principal of the input authentication is valid and verified, AuthenticationManager#authenticate returns an Authentication instance with the authenticated flag set to true.

8- What is Spring Security Filter Chain ?

Spring Security's web infrastructure is based entirely on standard servlet filters. Spring Security maintains a filter chain internally where each of the filters has a particular responsibility and filters are added or removed from the configuration depending on which services are required.

9- What are the differences between OAuth2 and JWT ?

Basically, JWT is a token format. OAuth is an authorization protocol that can use JWT as a token. OAuth uses server-side and client-side storage. If you want to do real logout you must go with OAuth2. Authentication with JWT token can not logout actually. Because you don't have an Authentication Server that keeps track of tokens. If you want to provide an API to 3rd party clients, you must use OAuth2 also. OAuth2 is very flexible. JWT implementation is very easy and does not take long to implement. If your application needs this sort of flexibility, you should go with OAuth2. But if you don't need this use-case scenario, implementing OAuth2 is a bad idea

10- What is method security and why do we need it ?

In simple terms, Spring method security allows us to support / add authorization supports at the method level. On a high level, we can configure which roles are allowed to access what method within the same service class.

11- What Proxy means and how and where can be used ?

A proxy server acts as a gateway between you and the internet. It's an intermediary server separating end users from the websites they browse. Proxy servers provide varying levels of functionality, security, and privacy depending on your use case, needs, or company policy.

12- What is Wrapper Class and where can be used ?

Wrapper classes are used to convert any data type into an object. The primitive data types are not objects; they do not belong to any class; they are defined in the language itself. Sometimes, it is required to convert data types

into objects in Java language. wrapper class contains or wraps primitive data type.

13- What is SSL ? What is TLS ? What is the difference ? How can we use them ?

SSL is a cryptographic protocol that uses explicit connections to establish secure communication between web server and client. TLS is also a cryptographic protocol that provides secure communication between web server and client via implicit connections.

SSL (Secure Sockets Layer) encryption, and its more modern and secure replacement, TLS (Transport Layer Security) encryption, protect data sent over the internet or a computer network.

14- Why do you need the intercept-url ?

Most web applications using Spring Security only have a couple of intercept-url s because they only have very basic security requirements. You need to have unauthenticated access to the login and login-error screens and usually some aspect of the public site, so that can be a few URL patterns