PayCore Java Spring Bootcamp [2.3]

Homework 5#

11.02.2022

QUESTION 1: What are Authentication and Authorization?

Answer:

If we simply compare authentication and authorization, authentication is a way to confirm to users that who they are. Authorization is the way that gives these users permission to access the resources. There are more differences between them. Let’s see it in this table:

|  |  |
| --- | --- |
| **Authentication** | **Authorization** |
| Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user. | Authorization works through settings that are implemented and maintained by the organization. |
| Authentication is the first step of a good identity and access management process. | Authorization always takes place after authentication. |
| Authentication is visible to and partially changeable by the user. | Authorization isn’t visible to or changeable by the user. |
| Authentication verifies who the user is. | Authorization determines what resources a user can access. |

QUESTION 2: What is Hashing in Spring Security?

Answer:

Hashing is the process of generating a hash from a given message using a mathematical function known as the cryptographic hash function. Hashes adapted for hashing passwords must have four main properties to be secure:

Determenistic: A great hashing must be deterministic. That means if the same message processed by the same hash function, always produce the same hash function, that means this hashing is deterministic.

Not Reversable: A great hashing cannot be reversed. That means, it is not great to generate massage from hash function.

High Entropy: A massage should produce a vastly different hash. This is called having a high entropy.

Resisting Collisions: That means 2 different massages cannot have same hash function.

There is also one more important thing. Password hashing functions should be slow. Because if it is fast, that means it would aid the brute force attack. PBKDF2, BCrypt, and SCrypt are the hash functions that has all of these properties.

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

Answer:

Salting is the way that makes our passwords safer. We use salting when we want to prevent an attack with rainbow tables. A salt is a sequence of randomly generated bytes that is hashed along with the password. The salt is stored in the storage and doesn’t need to be protected. Whenever the user tries to authenticate, the user’s password is hashed with the saved salt. The result should match the stored password. If the salt is long and random enough, it is impossible to find the hash in a rainbow table.

QUESTION 4: What is “intercept-url” pattern?

Answer:

The intercept-url patterns are processed in the order in which they appear in the security configuration models. Here is an example about it:

<http use-expressions="true">

<intercept-url pattern="/index.jsp" access="permitAll" />

<intercept-url pattern="/secure/extreme/\*\*"

access="hasRole('supervisor')" />

<intercept-url pattern="/secure/\*\*" access="isAuthenticated()" />

<intercept-url pattern="/listAccounts.html" access="isAuthenticated()" />

<intercept-url pattern="/post.html"access="hasAnyRole('supervisor','teller')" />

<intercept-url pattern="/\*\*" access="denyAll" />

<form-login />

</http>

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

Answer:

Session management is the process of securely handling multiple request to a web-based application or service from a single user or entity. The session management implementation specifies the process for sharing and continually exchanging the session ID between the user and the web application.

If we want to implement session management in spring,

1. Create Spring Boot project from Spring Initializer.
2. Add Spring Session jdbc dependency in pom.xml.
3. Add spring jdbc properties in application.properties.
4. Create rest end points to save, destroy/invalidate session.

QUESTION 6: Why we need Exception Handling?

Answer:

Exceptions are the object thrown at runtime and they can stop the program and make the program inefficient. While we text our codes we need to handle them to make our program work efficiently. There are many exceptions such as NullPointerException, ArrayOutofBoundException…etc. If we handle them, our program works efficiently and it is necessary.

QUESTION 7: Explain what is AuthenticationManager in Spring security?

Answer:

It is an interface to provide authenticatiton. An Authentication manager can return 3 things;

1. Return an Authentication (normally with authenticated=true) if it can verify that the input represents a valid principal.
2. Throw an AuthenticationException if it believes that the input represents an invalid principal.
3. Return null if it cannot decide.

Question 8: What is Spring Security Filter Chain?

Answer:

Spring security build on chain of filters, which will intercept the request, detect (absence of) authentication, redirect to authentication entry point or pass the request to authorization service, and eventually let the request either hit the servlet or throw security exception (unauthenticated or unauthorized). DelegatingFitlerProxy glues these filters together. To perform their tasks, these filter access services such as UserDetailsService and AuthenticationManager.

* Key filters in the chain are (in the order)
* SecurityContextPersistenceFilter (restores Authentication from JSESSIONID)
* UsernamePasswordAuthenticationFilter (performs authentication)
* ExceptionTranslationFilter (catch security exceptions from FilterSecurityInterceptor)
* FilterSecurityInterceptor (may throw authentication and authorization exceptions)

QUESTION 9: What are the differences between OAuth2 and JWT?

Answer:

JWT is a JSON based format of a security token which is basically a base64 url**-**encoded string which is used as a means of transferring secure content between two applications. They are used to secure request data in Web APIs. These are included in authorization HTTP headers as part of the bearer authentication scheme.

OAuth2 is the version 2 of the OAuth protocol. It can be referred to as a authorization framework as well.  OAuth as it name suggests is simply a standard for authorization. OAuth2 uses HTTPS for communication between the client and the authorization server because of confidential data for example client credentials. passing between the two applications.

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

Answer:

We can implement method-level security by placing the @PreAuthorize annotation on controller methods. To enable the security in spring boot:

<dependency>

<groupId>org.springframework.security</groupId>

<artifactId>spring-security-config</artifactId>

</dependency>

Than we need to enable global method security by using these codes:

@Configuration

@EnableGlobalMethodSecurity(

prePostEnabled = true,

securedEnabled = true,

jsr250Enabled = true)

public class MethodSecurityConfig

extends GlobalMethodSecurityConfiguration {

}

To secure our service layer by, for example, restricting which roles are able to execute a particular method and test it using dedicated method-level security test support we use method security.

Question 11: What Proxy means and how and where can be used?

Answer:

Proxy servers are software systems located between a private computer or end-user devices and a desired destination. Proxy is something that people have been using for a long time to protect itself from data leaks, spoofing, ad frauds and other such negative effects.

In addition, because of the proxy, it is possible to get rid of online monitoring. While the proxy server collects responses from web servers, it transmits it to you, providing enhanced levels of privacy as your real IP address is not revealed. And this can be used for bad purposes like entering the illegal websites.

Question 12: What is Wrapper Class and where can be used?

Answer:

Wrapper class is a class that provides to use premitive types in different ways. We use them because:

1. If we want to use Primitive data types as an object.
2. In java.util package we can only implement with classes and we can use wrapper classes that way.
3. For data structures such as ArrayList and Vector, we can use primitive types through wrapper classes.
4. We can use it to create a necessary object for multithreading synchronization.

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

Answer:

* SSL is known as Secure Socket Layer. It is a security protocol which is made by NetScape. It’S functions are:

1. Providing privacy and reliability
2. It allows the communication between the server and the client to be encrypted for the integrity and confidentiality of the information (data protection) during the information transfer on the network.

SSL performs authentication over a structure provided by keys referred to as “Public Key” and “Private Key”. These keys work in harmony with each other and carry out the encryption / verification process between the parties (client and server). A value between 40 bits and 128 bits is used for encryption operations. This value will vary according to SSL types and shows how strong the encryption method is. The higher the key length used, the stronger the encryption.

* TLS is known as Transport Layer Security. It is also made by NetScape. We can say that SSL is a pioneer to TLS.

TLS consists of two layers:

1. TLS Record Protocol
2. TLS Handshake Protocol

Handshake Protocol handles authentication of servers and users. Before the Handshake Protocol data communication, encryption algorithms and encryption keys are allowed, while the Record Protocol ensures the security of the connection.

Differences:

* At the security point, both SSL and TLS offer almost equal security. However, while SSL starts the process with security and moves directly to secure data communication, TLS sends an insecure boot message to the server. If a handshake does not occur between the client and the server over this message (message authentication), the connection is not established. If handshake is provided, a secure connection is established and data communication is provided.
* If the client does not have an SSL certificate, the TLS protocol may pass a "No Certificate" message. In SSL, there is no need for additional notification.
* TLS transmits the certificate authentication message in handshake, while SSL transmits the authentication message in a more complex process.
* TLS uses "done" messages from the client and server along with the PRF output for the "done message". SSL, on the other hand, generates an "end message" similar to key generation.

Question 14: Why do you need the intercept-url?

Answer:

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.