**Spring\_Interview\_Questions**

**Q1) What is spring security**

**A)** for spring application security purpose, we should use the security. in my project we are using JWT token.

**Q2) how we are authenticating by using JWT token.**

**A)** first, we will call POST/authenticate api with username and password

then server-side validation should happen if the user valid it will generate JWT token to the user. If the user not valid then it will show unauthorized error (401 error-code)

if suppose user trying to get the data then we need to send the JWT token along with request.

if the token valid then we will get the response.

**Q3) how to create and send JWT token**

**A)** we can define random key, key-store, key-store password in properties file.

then we can create java class for JWT configurations and we can call the properties file by using @PropertySource annotation then we can use the random key in java class.

suppose if we want to use the fields from properties file in java class we can use @Value annotation.

For Creating JWT token: JWT.create().withHeader().withIssuer().withIssuedAt().withExpiredAt().withSubject().sign()

**Q4) Spring Working flow**

**A)**

* Once we submit the request through client
* Dispatcher Servlet traps the request
* Then Dispatcher servlet sends the request to handler mapper
* Handler mapper will check the URI in controller class if it is matched any handler method, it will send the method signature and class name to the handler mapper
* Handler mapper can share the details (method signature and class name) to dispatcher servlet
* Then dispatcher servlet directly connects with controller class with the help of method signature
* And it will execute the business logic then return the LVN to the dispatcher servlet
* The dispatcher servlet sends the LVN name to view resolver to get the physical view component
* View resolver shares the physical view components details to dispatcher servlet
* Then dispatcher servlet connects with physical view component for response
* Physical view component shares the response to dispatcher servlet with requested format
* Dispatcher servlet sends the response to client

**Q5) What is reflection API**

**A)** Reflection is an API that is used to examine or modify the behaviour of the methods, classes and interfaces at runtime. The required classes for reflection are provided by under Java.lang.reflection package.

Reflection API gives us information about the class to which object belongs and also the methods of the class that can executed by using which object.

Through reflection API, we can invoke method at runtime irrespective of the access specifier used with them.

**Q6) What is spring IOC container**

**A)** Spring IOC is the mechanism to achieve loose-coupling between objects dependencies.

To achieve loose-coupling and dynamic binding of the objects at runtime, objects dependencies are injected by another assembler object, that object is called IOC container.

The IOC container is responsible to instantiate, configure and assemble the objects. The IOC container get the information from the XML file and working accordingly.

Bean factory is the root interface for spring IOC container and application context is the child interface of bean factory

IOC Container is a software program who manages whole life cycle of given resource(class) end to end (from object creation to object destroy).

Ex-1: JSP container manages the life cycle of JSP components

EX-2: EJB container manages the life cycle of EJB components

Ex-3: Spring container manages the life cycle of spring beans/spring components

**Q7) What is spring bean**

**A)** Spring bean is nothing but any object in the spring framework that we initialize through spring container is called spring bean.

Any java pojo class can be a spring bean if it’s configured to be initialized via container by providing configuration meta data information.

Spring bean can be a pre-defined class or user-defined class or third party supplied class as well.

**Q8) Spring bean configuration**

**A)** Spring bean provides three ways to configure beans to be used.

a.) Annotation based configuration - @Service or @Component or @Repository

b.) XML based configuration – by using <bean> tag

c.) Java based configuration - @Configuration or @Bean or @ComponentScan

**Q9) What is spring bean scope**

**A)** Bean Scope refers to the lifecycle of a bean, visibility of a bean, how long does the bean live, how many instances are created, how is the bean shared?

There are five scopes defined for spring bean, Bean's default scope is a singleton.

**a.) Singleton:** Only one instance of the bean will be created for each container. This is the default scope for the spring bean, while using this scope make sure bean doesn’t have shared instance variable. It may leads the data inconsistency issue.

**b.) Prototype:** A new instance will be created every time the bean is requested.

**c.) Request:** This is same as prototype scope, however it is to be used for web application.

**d.) Session:** A new bean will be created for each HTTP session by the container

**e.) Global-Session:** This is used to create a global session bean for global HTTP sessions

**Q10) What are the different ways to achieve the dependency injection**

**A)** We can achieve dependence injection by using three ways.

i.) Constructor based dependence injection

ii.) Setter based dependence injection

iii.) Field or property-based dependence injection

**Q11) What is Spring**

**A)** Spring is a light weight, open source, loosely coupled, non-invasive, aspect-oriented Java/J2ee application framework to develop all types of java, jee applications by getting abstraction on multiple java, jee technologies.

**Q12) What is the different between EJB and Spring**

**A)**

|  |  |
| --- | --- |
| **EJB** | **Spring** |
| It can inject anything in the container including EJB data source, JM resource and JPA resource | It can inject anything including list, map and JNDI resource |
| It supports only JTA transaction manager | It supports multiple transaction manager such as JTA, JDBC and hibernate |
| It tightly integrated to JPA | It supports various persistence technologies such as JDBC, hibernate, JPA and IBatis |
| It provide simple scheduling through EJB time service | We need to add and configure quartz for scheduling |
| It supports both declarative and programmatic security through JAAS | It provides declarative security through the spring configuration |
| It provides AOP limited support through interceptor | It provides AOP robust support through Aspect |

**Q13) What is the difference between HTML form tags and Spring mvc supplied form tags**

**A)**

|  |  |
| --- | --- |
| **HTML form Tags** | **Spring MVC Supplied form Tags** |
| These tags are given by W3C (world wide web consortium) | These tags are given by spring mvc framework (pivotal team) |
| These tags support one way data binding when they are used in spring mvc apps (from data to model class obj) | These tags support two way data binding when they are used in spring mvc apps (from data to model class obj and vice-versa) |
| Default request method for the form tag is GET | Default request method for the from is POST |
| These tags are not strictly typed | These tags are strictly typed |
| These tags are not case-sensitive | These tags are very much case-sensitive |
| Can be used in HTML programming and JSP programming | Can be used only in JSP programming |

Q14) Why spring framework

A)