Name:Shweta Gund

Theory Questions:

Q1. what is the @component and @Controller.?

Ans:-1.@Component is an annotation that allows Spring to automatically detect our custom beans. In other words, without having to write any explicit code, Spring will: Scan our application for classes annotated with @Component. Instantiate them and inject any specified dependencies into them.

2.@Controller this marks a class as a Spring Web MVC controller. Beans marked with it are automatically imported into the Dependency Injection container.

Q2.@RequestMapping,@Required,@Qualifier,@Autowired @Temporal,@Entity@RequestBody @RestController @Query @PathVariable Annotation, explain?

Ans:1.@RequestMapping: It is used to map the web requests. It has many optional elements like consumes, header, method, name, params, path, produces, and value. We use it with the class as well as the method.

2.@Required: It applies to the bean setter method. It indicates that the annotated bean must be populated at configuration time with the required property, else it throws an exception BeanInitilizationException.

3.@Autowired: Spring provides annotation-based auto-wiring by providing @Autowired annotation. It is used to autowire spring bean on setter methods, instance variable, and constructor. When we use @Autowired annotation, the spring container auto-wires the bean by matching data-type.

4.@Bean: It is a method-level annotation. It is an alternative of XML <bean> tag. It tells the method to produce a bean to be managed by Spring Container.

5.@PathVariable annotation can be used to handle template variables in the request URI mapping, and set them as method parameters.

6.@RestController annotation in order to simplify the creation of RESTful web services.

7.@RequestBody annotation is applicable to handler methods of Spring controllers. This annotation indicates that Spring should deserialize a request body into an object. This object is passed as a handler method parameter.

8.@Query annotation — its value attribute contains the JPQL or SQL to execute.

9.@Entity annotation specifies that the class is an entity and is mapped to a database table.

10.@Temporal annotation may be used in conjunction with the Basic annotation, the Id annotation, or the ElementCollection annotation (when the element collection value is of such a temporal type.

Q3.What are the Stereotype annotations and define MVC flow.

Ans:-1.A Spring MVC is a Java framework which is used to build web applications.It follows the Model-View-Controller design pattern.It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.

2.A Spring MVC provides an elegant solution to use MVC in spring framework by the help of DispatcherServlet. Here, DispatcherServlet is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views.

3.The Model-View-Controller (MVC) is an architectural pattern that separates an application into three main logical components: the model, the view, and the controller

04.Diffrence between application.properties & YML fiile.

Ans:-1.application.properties:-1.It doesn’t really actually have a spec. The closest thing it has to a spec is actually the javadoc.

2.Human Readable

3.Supports key/val, but doesn’t support values beyond the string

4.It is primarily used in java

2.yml :-1.Human Readable (both do quite well in human readability)

2.Supports key/val, basically map, List and scalar types (int, string etc.)

3.Its usage is quite prevalent in many languages like Python, Ruby, and Java

05.What are the scopes in Spring Framework?

ans:-1.singleton:-This scopes the bean definition to a single instance per Spring IoC container (default).

2.prototype:-This scopes a single bean definition to have any number of object instances.

3.request:-This scopes a bean definition to an HTTP request. Only valid in the context of a web-aware Spring ApplicationContext.

4.session:-This scopes a bean definition to an HTTP session. Only valid in the context of a web-aware Spring ApplicationContext.

5.global-session:-This scopes a bean definition to a global HTTP session. Only valid in the context of a web-aware Spring ApplicationContext.

Q6. What is cascading and what are different types of cascading?

Ans:-To establish a dependency between related entities, JPA provides javax.persistence.CascadeType enumerated types that define the cascade operations.These cascading operations can be defined with any type of mapping i.e. One-to-One, One-to-Many,Many-to-One, Many-to-Many.

1.PERSIST - In this cascade operation, if the parent entity is persisted then all its related entity will also be persisted.

2.MERGE - In this cascade operation, if the parent entity is merged then all its related entity will also be merged.

3.DETACH - In this cascade operation, if the parent entity is detached then all its related entity will also be detached.

4.REFRESH - In this cascade operation, if the parent entity is refreshed then all its related entity will also be refreshed.

5.REMOVE - In this cascade operation, if the parent entity is removed then all its related entity will also be removed.

ALL - In this case, all the above cascade operations can be applied to the entities related to parent entity.

Q7.what is the ioc container and autowiring?

Ans:-The IoC container is responsible to instantiate, configure and assemble the objects.The IoC container gets informations from the XML file and works accordingly.

There are two types of IoC containers.:-

1.BeanFactory,2.ApplicationContext

Autowiring feature of spring framework enables you to inject the object dependency implicitly.It internally uses setter or constructor injection. Autowiring can't be used to inject primitive and string values. It works with reference only.

Q8.What is difference between hibernate and jpa? and What is Hibernate cache?.

Ans:- 1.JPA- Java Persistence API (JPA) defines the management of relational data in the Java applications.It is just a specification. Various ORM tools implement it for data persistence.It is defined in javax.persistence package.The EntityManagerFactory interface is used to interact with the entity manager factory for the persistence unit. Thus, it provides an entity manager.

2.Hibernate -

Hibernate is an Object-Relational Mapping (ORM) tool which is used to save the state of Java object into the database.

It is one of the most frequently used JPA implementation.

It is defined in org.hibernate package.

It uses SessionFactory interface to create Session instances.

Q10.what is difference between Named Queries and Criteria Queries?

Ans:- A named query is a statically defined query with a predefined unchangeable query string. Using named queries instead of dynamic queries may improve code organization by separating the JPQL query strings from the Java code.

A query criterion is an expression that Access compares to query field values to determine whether to include the record that contains each value.

Q11.What is Spring Profile?

Ans:-We use the @Profile annotation — we are mapping the bean to that particular profile; the annotation simply takes the names of one (or multiple) profiles.

We annotate that bean with a dev profile, and it will only be present in the container during development.

In production, the dev simply won't be active:

@Component

@Profile("dev")

public class DevDatasourceConfig

Declare Profiles in XML:

<beans profile="dev">

<bean id="devDatasourceConfig"

class="org.baeldung.profiles.DevDatasourceConfig" />

</beans>

Programmatically via ConfigurableEnvironment:

@Autowired

private ConfigurableEnvironment env;

...

env.setActiveProfiles("someProfile");

Q12.List Annotations from springboot?

ans:-1.@Required: It applies to the bean setter method. It indicates that the annotated bean must be populated at configuration time with the required property, else it throws an exception BeanInitilizationException.

2.@Autowired: Spring provides annotation-based auto-wiring by providing @Autowired annotation. It is used to autowire spring bean on setter methods, instance variable,

and constructor. When we use @Autowired annotation, the spring container auto-wires the bean by matching data-type.

3.@Configuration: It is a class-level annotation. The class annotated with @Configuration used by Spring Containers as a source of bean definitions.

4.@Bean: It is a method-level annotation. It is an alternative of XML <bean> tag. It tells the method to produce a bean to be managed by Spring Container.

5.@Controller: The @Controller is a class-level annotation. It is a specialization of @Component. It marks a class as a web request handler. It is often used to serve web pages.By default, it returns a string that indicates which route to redirect. It is mostly used with @RequestMapping annotation.

6.@Service: It is also used at class level. It tells the Spring that class contains the business logic.

7.@Repository: It is a class-level annotation. The repository is a DAOs (Data Access Object) that access the database directly. The repository does all the operations related to the database.

8.@RequestMapping: It is used to map the web requests. It has many optional elements like consumes, header, method, name, params, path, produces, and value. We use it with the class as well as the method.

9.@GetMapping: It maps the HTTP GET requests on the specific handler method. It is used to create a web service endpoint that fetches It is used instead of using: @RequestMapping(method = RequestMethod.GET)

10.@PostMapping: It maps the HTTP POST requests on the specific handler method. It is used to create a web service endpoint that creates It is used instead of using: @RequestMapping(method = RequestMethod.POST)

11.@PutMapping: It maps the HTTP PUT requests on the specific handler method. It is used to create a web service endpoint that creates or updates It is used instead of using: @RequestMapping(method = RequestMethod.PUT)

12.@DeleteMapping: It maps the HTTP DELETE requests on the specific handler method. It is used to create a web service endpoint that deletes a resource. It is used instead of using:@RequestMapping(method = RequestMethod.DELETE)

13.@RequestBody: It is used to bind HTTP request with an object in a method parameter. Internally it uses HTTP MessageConverters to convert the body of the request. When we annotate a method parameter with @RequestBody, the Springframework binds the incoming HTTP request body to that parameter.

14.@RestController: It can be considered as a combination of @Controller and @ResponseBody annotations. The @RestController annotation is itself annotated with the @ResponseBody annotation. It eliminates the need for annotating each method with @ResponseBody.

Q14.What is Junit and Annotations used in Junit.?

Ans:- **JUnit Annotations** is a special form of syntactic meta-data that can be added to Java source code for better code readability and structure. Variables, parameters, packages, methods and classes can be annotated.