Flashcards

# Flashcard 1

Question: @SpringBootApplication

Answer: A meta-annotation that combines three other annotations: @Configuration, @EnableAutoConfiguration, and @ComponentScan. This annotation is typically placed on the main class of a Spring Boot application and serves as the entry point for the application.

# Flashcard 2

Question: @Controller

Answer: A fundamental component in Spring MVC (Model-View-Controller) architecture. It is used to define a class as a controller, which handles HTTP requests. Typically, methods in a @Controller class return a view name, which is resolved by a view resolver to a physical view template (e.g., a JSP page). It's suitable for applications that render HTML views and follow the traditional multi-page web application model.

# Flashcard 3

Question: @RestController

Answer: A specialized version of @Controller introduced in Spring 4. It is used to define a class as a controller that handles HTTP requests, but it is specifically designed for RESTful web services. Methods in a @RestController class return the data directly (e.g., JSON or XML) instead of relying on a view resolver to render HTML views. It's suitable for applications that expose APIs and return data in a format like JSON or XML.

# Flashcard 4

Question: @Service

Answer: Used to indicate that a class is a service in the business logic layer. It is typically applied to service classes that contain business logic, perform operations, and coordinate the interaction between different components.

# Flashcard 5

Question: @Repository

Answer: Used to indicate that a class is a Data Access Object (DAO) in the persistence layer. It is typically applied to classes that access a database or perform data access operations, and it helps in translating any platform-specific exceptions to Spring's DataAccessException.

# Flashcard 6

Question: @Component

Answer: Use @Service for service classes in the business logic layer. Use @Repository for DAO classes in the persistence layer. Use @Component as a generic stereotype when a more specific one is not applicable.

# Flashcard 7

Question: @Autowired

Answer: Used to automatically inject dependencies into a Spring bean. It is a form of dependency injection that allows Spring to resolve and inject collaborating beans into your bean.

# Flashcard 8

Question: @Qualifier

Answer: Used in conjunction with @Autowired to specify which bean should be injected when there are multiple candidates of the same type. This is especially useful when you have more than one bean of the same type in the Spring context, and you want to explicitly specify which one should be injected.

# Flashcard 9

Question: @Entity

Answer: Used to define a data entity managed by the ORM (Object-Relational Mapping) system. It maps the class fields to database tables/columns while providing CRUD operations.

# Flashcard 10

Question: @Transactional

Answer: Defines the scope of a single database transaction. Applied to methods or classes to indicate that the annotated method or all methods in the class should be wrapped with a transaction.