Spring Boot Interview Questions & Answers

Chapter 14: Spring Boot – Annotations

Q: Why are annotations important in Spring Boot?

Annotations in Spring Boot reduce boilerplate code and provide declarative programming. Instead of writing lengthy configurations, you can enable features with simple annotations.

Use-Case / Example: Using @RestController removes the need to write both @Controller and @ResponseBody separately.

Q: What does @SpringBootApplication do?

It is a combination of three annotations:
- @Configuration → marks the class as a configuration class.
- @EnableAutoConfiguration → enables Spring Boot's auto-configuration.
- @ComponentScan → enables component scanning in the base package.

Use-Case / Example: A single @SpringBootApplication annotation is enough to bootstrap a Spring Boot application.

Q: What is the difference between @Controller, @RestController, and @ResponseBody?

- @Controller → used for MVC apps to return views.
- @ResponseBody → tells Spring to return JSON/XML instead of a view.
- @RestController → shorthand that combines both.

Use-Case / Example: For a REST API that returns JSON, use @RestController. For a web app that returns JSP or Thymeleaf templates, use @Controller.

Q: What does @Autowired do?

@Autowired automatically injects a bean where it is needed. It reduces manual object creation and helps achieve dependency injection.

Use-Case / Example: In a BookController, adding @Autowired to a BookService field injects the service automatically.

Q: What is the role of @Service, @Repository, and @Component?

- @Component → generic stereotype for any Spring bean.
- @Service → used for service layer classes (business logic).
- @Repository → used for DAO layer, with added exception translation.

Use-Case / Example: In a banking system:
- AccountService is annotated with @Service.
- AccountRepository is annotated with @Repository.

Q: What does @Configuration and @Bean mean?

- @Configuration marks a class that contains bean definitions.
- @Bean marks a method that returns a bean to be managed by Spring.

Use-Case / Example:

```
@Configuration
public class AppConfig {
@Bean
public ModelMapper modelMapper() {
return new ModelMapper();
}
}
```

Q: How is @Value used in Spring Boot?

@Value is used to inject values from application.properties or environment variables into fields.

Use-Case / Example:

```
@Value("${app.name}")
private String appName;
```

This injects the value of app.name property into the variable.

Q: What does @EnableAutoConfiguration do?

It tells Spring Boot to automatically configure beans based on dependencies in the classpath.

Use-Case / Example: If you add spring-boot-starter-data-jpa, Spring Boot auto-configures a DataSource and JPA-related beans.

Q: What is @Qualifier and when do you use it?

@Qualifier is used when multiple beans of the same type exist, and you need to specify which one should be injected.

Use-Case / Example: If both PayPalService and StripeService implement PaymentService, you can specify @Qualifier("paypalService") to inject the correct one.

Q: What does @Lazy annotation do?

By default, Spring initializes beans eagerly at startup. @Lazy tells Spring to initialize a bean only when it is first needed.

Use-Case / Example: In a reporting application, a heavy ReportGenerator bean can be marked as @Lazy, so it loads only when reports are actually requested.