

# Spring Boot Annotations - Interview FAQ

---

## 1. Core Annotations - Mostly Asked in Every Interview

### @SpringBootApplication

- **Purpose:** Marks the main class of a Spring Boot application.
- **Explanation:** Combines @Configuration, @EnableAutoConfiguration, and @ComponentScan.
- **Example:**

```
@SpringBootApplication
public class MySpringBootApplication {
    public static void main(String[] args) {
        SpringApplication.run(MySpringBootApplication.class, args);
    }
}
```

### @Configuration

- **Purpose:** Indicates that the class can be used by Spring IoC container as a source of bean definitions.
- **Example:**

```
@Configuration
public class AppConfig {
    @Bean
    public MyService myService() {
        return new MyService();
    }
}
```

### @ComponentScan

- **Purpose:** Automatically discovers and registers beans in specified packages.
- **Example:**

```
@ComponentScan(basePackages = "com.example.myapp")
public class AppConfig {}
```

### @EnableAutoConfiguration

- **Purpose:** Enables Spring Boot's auto-configuration mechanism.
- **Example:**

```
@EnableAutoConfiguration
public class AppConfig {}
```

---

## 2. Bean Annotations

### @Component

- **Purpose:** Marks a Java class as a Spring-managed component.
- **Example:**

```
@Component
public class MyComponent {}
```

### @Service

- **Purpose:** Specialized @Component annotation, indicating a service layer.
- **Example:**

```
@Service
public class MyService {}
```

### @Repository

- **Purpose:** Specialized @Component annotation for the persistence layer.
- **Example:**

```
@Repository
public class MyRepository {}
```

### @Bean

- **Purpose:** Declares a Spring bean in @Configuration classes.
- **Example:**

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

---

## 3. Web Layer Annotations

### @RestController

- **Purpose:** Combination of @Controller and @ResponseBody.
- **Example:**

```
@RestController
@RequestMapping("/api")
public class MyController {
    @GetMapping("/hello")
    public String sayHello() {
        return "Hello, World!";
    }
}
```

### @RequestMapping

- **Purpose:** Maps HTTP requests to handler methods.

- **Example:**

```
@RequestMapping("/api")
```

### **@GetMapping, @PostMapping, @PutMapping, @DeleteMapping**

- **Purpose:** Specialized annotations for HTTP methods.
- **Example:**

```
@GetMapping("/users")
public List<User> getUsers() {
    return userService.getAllUsers();
}
```

---

## **4. Dependency Injection Annotations**

### **@Autowired**

- **Purpose:** Automatically injects dependencies.
- **Example:**

```
@Autowired
private MyService myService;
```

### **@Qualifier**

- **Purpose:** Specifies which bean to inject when multiple candidates exist.
- **Example:**

```
@Autowired
@Qualifier("specificBeanName")
private MyService myService;
```

### **@Value**

- **Purpose:** Injects values from properties files.
- **Example:**

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

---

## **5. Conditional Annotations**

### **@ConditionalOnProperty**

- **Purpose:** Configures beans based on environment properties.
- **Example:**

```
@Bean
@ConditionalOnProperty(name = "app.feature.enabled", havingValue = "true")
public MyFeatureBean myFeatureBean() {
    return new MyFeatureBean();
}
```

### @ConditionalOnMissingBean

- **Purpose:** Defines a bean only if another specific bean is missing.
- **Example:**

```
@Bean
@ConditionalOnMissingBean
public MyDefaultBean myDefaultBean() {
    return new MyDefaultBean();
}
```

---

## 6. Transaction and Validation Annotations

### @Transactional

- **Purpose:** Marks a method or class as transactional.
- **Example:**

```
@Transactional
public void saveUser(User user) {
    userRepository.save(user);
}
```

### @Valid & @Validated

- **Purpose:** Validates request bodies or method arguments.
- **Example:**

```
@PostMapping("/users")
public ResponseEntity<?> createUser(@Valid @RequestBody User user) {
    return ResponseEntity.ok(userService.save(user));
}
```

---

## 7. Spring Boot Testing Annotations

### @SpringBootTest

- **Purpose:** Provides a Spring context for integration testing.
- **Example:**

```
@SpringBootTest
public class MyApplicationTests {
    @Test
    public void contextLoads() {
    }
}
```