

SPRING BOOT ANNOTATIONS – COMPLETE LIST WITH MEANING

◆ 1. CORE SPRING ANNOTATIONS

@Component

- Marks a class as **Spring-managed bean**
- Base annotation for all stereotypes

```
@Component  
public class UtilityService { }
```

@Autowired

- Used for **Dependency Injection**
- Injects required bean automatically

```
@Autowired  
private StudentService service;
```

@Qualifier

- Used when **multiple beans of same type** exist

```
@Qualifier("studentServiceImpl")
```

@Primary

- Marks default bean when multiple beans exist
-

◆ 2. STEREOTYPE ANNOTATIONS

@Controller

- Used for **MVC web controllers**
 - Returns **view (HTML/JSP)**
-

@RestController

- Combination of:
 - @Controller
 - @ResponseBody
- Returns **JSON/XML response**

@RestController

@Service

- Used for **business logic layer**

@Service

```
public class StudentService { }
```

@Repository

- Used for **DAO layer**
- Handles DB exceptions automatically

@Repository

```
public interface StudentRepo { }
```

◆ 3. SPRING BOOT MAIN ANNOTATION

@SpringBootApplication

- Combination of:
 - @Configuration
 - @EnableAutoConfiguration
 - @ComponentScan

@SpringBootApplication

◆ 4. WEB / REST ANNOTATIONS

@RequestMapping

- Maps URL to controller/method
- Supports all HTTP methods

@RequestMapping("/users")

HTTP METHOD ANNOTATIONS

Annotation	Meaning
<code>@GetMapping</code>	Fetch data
<code>@PostMapping</code>	Create new data
<code>@PutMapping</code>	Update full data
<code>@PatchMapping</code>	Update partial data
<code>@DeleteMapping</code>	Delete data

`@RequestParam`

- Gets **query parameter** from URL

`@RequestParam("name") String name`

`@PathVariable`

- Gets **dynamic value** from URL path

`@PathVariable int id`

`@RequestBody`

- Gets data from **request body (JSON/XML)**

`@RequestBody Student student`

@ResponseBody

- Converts return value to JSON/XML
-

@CrossOrigin

- Allows cross-origin requests (CORS)
-

◆ 5. VALIDATION ANNOTATIONS

@Valid

- Enables validation on request body
-

Common Validation Annotations

Annotation	Meaning
@NotNull	Cannot be null
@NotBlank	Cannot be empty
@Size	Size constraint
@Min	Minimum value
@Max	Maximum value
@Email	Valid email

◆ 6. JPA / HIBERNATE ANNOTATIONS (VERY IMPORTANT)

@Entity

- Marks class as **database table**
-

@Table

- Specifies table name
-

@Id

- Marks primary key
-

@GeneratedValue

- Auto-generates ID

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column

- Maps field to DB column
-

RELATIONSHIP ANNOTATIONS

Annotation	Meaning
@OneToOne	One-to-One
@OneToMany	One-to-Many
@ManyToOne	Many-to-One
@ManyToMany	Many-to-Many

@JoinColumn

- Foreign key column
-

@Transient

- Field not stored in DB
-

◆ 7. TRANSACTION ANNOTATIONS

@Transactional

- Manages transaction (commit/rollback)
 - Used in **Service layer**
-

◆ 8. CONFIGURATION ANNOTATIONS

@Configuration

- Marks configuration class
-

@Bean

- Creates manual bean

```
@Bean  
public ModelMapper mapper() { }
```

@Value

- Reads value from properties file
-

@ConfigurationProperties

- Binds properties to class
-

◆ 9. EXCEPTION HANDLING

@ExceptionHandler

- Handles specific exception
-

@ControllerAdvice

- Global exception handler
-

◆ 10. TESTING ANNOTATIONS

Annotation	Meaning
@SpringBootTest	Full context test
@MockBean	Mock dependency
@WebMvcTest	Controller test

◆ 11. SECURITY (Basic Awareness)

Annotation	Meaning
@EnableWebSecurity	Enable security
@PreAuthorize	Method security

◆ 12. MOST IMPORTANT INTERVIEW LIST (MEMORIZE)

@SpringBootApplication

@RestController

@Service

@Repository

@Autowired

@GetMapping / PostMapping

@RequestBody

@PathVariable

@RequestParam

@Entity

@Id

@GeneratedValue

@Transactional