



## Component Scanning & Dependency Injection

- `@Component`: Marks a class as a Spring-managed component.
  - `@Service`: Specialized `@Component` for the service layer.
  - `@Repository`: Specialized `@Component` for the persistence layer (adds exception translation).
  - `@Controller`: Specialized `@Component` for web controllers.
  - `@RestController`: Combines `@Controller` and `@ResponseBody` for REST APIs.
  - `@ComponentScan`: Enables scanning for components in specified packages.
  - `@Autowired`: Automatically injects dependencies.
  - `@Qualifier`: Specifies which bean to inject when multiple beans of the same type exist.
  - `@Primary`: Marks a bean as the default for injection.
-



## Configuration & Bean Management

- `@Configuration`: Indicates the class contains Spring bean definitions.
- `@Bean`: Declares a Spring-managed bean.
- `@Lazy`: Defers bean initialization until it is first used.
- `@DependsOn`: Declares dependencies that must be initialized before the bean.
- `@Scope`: Sets the bean scope (`singleton`, `prototype`, etc.).



## Scheduling & Async Execution

- `@EnableScheduling`: Enables support for scheduled tasks.
- `@Scheduled`: Marks a method to run on a schedule (cron, `fixedDelay`).
- `@EnableAsync`: Enables asynchronous method execution.
- `@Async`: Runs a method in a separate thread asynchronously.



## Spring AOP (Aspect-Oriented Programming)

- `@EnableAspectJAutoProxy`: Enables AOP proxy support.
  - `@Aspect`: Marks a class as an aspect.
  - `@Before`: Executes before a matched method.
  - `@After`: Executes after a matched method.
  - `@AfterReturning`: Runs after a method successfully returns.
  - `@AfterThrowing`: Runs if a matched method throws an exception.
  - `@Around`: Wraps a method (runs before and after).
-



## Spring Boot Annotations

- `@SpringBootApplication`: Combines `@Configuration`, `@EnableAutoConfiguration`, `@ComponentScan`.
  - `@EnableAutoConfiguration`: Automatically configures Spring beans based on classpath.
  - `@ConfigurationProperties`: Binds external config (YAML/properties) to a POJO.
  - `@Value`: Injects values from properties files.
  - `@Profile`: Enables beans only under specific profiles.
  - `@ConditionalOnProperty`: Loads bean only if a specific property is set.
-



## Spring MVC & REST

- `@RequestMapping`: Maps HTTP requests to handler methods.
  - `@GetMapping`, `@PostMapping`, `@PutMapping`, `@DeleteMapping`: Shorthand for `@RequestMapping` with HTTP verbs.
  - `@RequestParam`: Binds query parameters.
  - `@PathVariable`: Binds URI template variables.
  - `@RequestBody`: Binds request body to a method parameter.
  - `@ResponseBody`: Binds method return value to the HTTP response body.
-



## Exception Handling

- `@ControllerAdvice`: Global exception handler for controllers.
  - `@ExceptionHandler`: Handles specific exceptions.
  - `@ResponseStatus`: Specifies the HTTP status for a method or exception.
-

## 📌 Spring Cloud & Microservices

- `@EnableDiscoveryClient`: Enables service discovery (Eureka/Consul).
  - `@FeignClient`: Declarative REST client.
  - `@LoadBalanced`: Enables client-side load balancing for `RestTemplate`.
  - `@RefreshScope`: Refreshes beans when config changes.
  - `@EnableConfigServer`: Sets up a Spring Cloud Config Server.
-



## Spring Kafka

- `@EnableKafka`: Enables Kafka support in the application.
  - `@KafkaListener`: Marks a method to consume Kafka messages.
  - `@KafkaHandler`: Handles different message types in a single listener.
- 



## Messaging & WebSockets

- `@EnableWebSocket`: Enables WebSocket configuration.
  - `@MessageMapping`: Maps WebSocket messages to methods (STOMP).
  - `@SendTo`: Sends a response to a message destination.
  - `@SubscribeMapping`: Handles STOMP subscription requests.
  - `@JmsListener`: Listens for JMS messages.
  - `@RabbitListener`: Listens for RabbitMQ messages.
-

## Caching

- `@EnableCaching`: Enables caching support in Spring.
  - `@Cacheable`: Caches the return value of a method.
  - `@CachePut`: Updates cache without skipping method execution.
  - `@CacheEvict`: Clears entries from the cache.
- 

## Spring Security

- `@EnableWebSecurity`: Enables Spring Security configuration.
  - `@PreAuthorize`: Adds method-level security using SpEL.
  - `@Secured`: Restricts access to methods based on roles.
  - `@WithMockUser`: Simulates a user for Spring test security.
-



## Spring Data MongoDB

- `@Document`: Marks a class as a MongoDB document.
  - `@Field`: Maps a field to a MongoDB field name.
  - `@Id`: Identifies the primary key field.
  - `@DBRef`: Links to another document.
-



## Spring Data Cassandra

- `@Table`: Marks a class as a Cassandra table.
  - `@PrimaryKey`: Marks the primary key field.
  - `@Column`: Maps a field to a Cassandra column.
  - `@CassandraType`: Specifies Cassandra-specific type.
- 



## Spring Testing

- `@SpringBootTest`: Loads the full application context for tests.
  - `@DataJpaTest`: Loads JPA components with in-memory DB.
  - `@WebMvcTest`: Loads only web layer (controllers).
  - `@MockBean`: Registers a Mockito mock in Spring context.
  - `@TestConfiguration`: Defines test-specific bean configuration.
- 

## 📌 JPA & Transactions

- `@Entity`: Marks a class as a JPA entity.
- `@Table`: Maps the entity to a DB table.
- `@Column`: Maps a field to a table column.
- `@Id`: Denotes the primary key field.
- `@GeneratedValue`: Strategy for primary key generation.
- `@OneToMany`, `@ManyToOne`: Defines entity relationships.
- `@Transactional`: Declares method/class as transactional.
- `@EnableTransactionManagement`: Enables `@Transactional` support.



## Spring Repository & Query

- `@Repository`: Marks a DAO class.
  - `@EnableJpaRepositories`: Enables scanning for JPA repositories.
  - `@Query`: Declares custom queries (JPQL or native).
  - `@Modifying`: Marks update/delete queries.
- 



## Validation

- `@Valid`: Triggers bean validation on input.
  - `@NotNull`: Validates non-null fields.
  - `@Size`: Validates size constraints.
  - `@Email`: Validates email format.
  - `@Pattern`: Validates string using regex.
  - `@Min, @Max`: Validates numeric range.
-