

Spring Boot Annotations

Here's a table summarizing commonly used Spring Boot annotations, their purpose, and where they are applied:

Annotation	Description	Usage Scope
@SpringBootApplication	Combines @Configuration, @EnableAutoConfiguration, and @ComponentScan. It marks the main class.	Class (Main Application)
@RestController	Combines @Controller and @ResponseBody to create RESTful web services.	Class
@Controller	Marks a class as a web controller.	Class
@Service	Indicates a service layer component.	Class
@Repository	Indicates a repository or DAO layer component.	Class
@Component	Marks a general-purpose Spring-managed bean.	Class
@Autowired	Automatically injects dependencies into beans.	Fields, Constructors, Methods
@Value	Injects values from application properties or environment variables.	Fields, Constructors, Methods
@Bean	Declares a Spring bean within a @Configuration class.	Method
@Configuration	Indicates a class that defines Spring configuration.	Class
@EnableAutoConfiguration	Enables Spring Boot's auto-configuration feature.	Class
@ComponentScan	Scans for Spring components, configurations, and beans in specified packages.	Class
@RequestMapping	Maps HTTP requests to handler methods of REST controllers.	Class, Method
@GetMapping, @PostMapping, @PutMapping, @DeleteMapping	Specialized annotations for HTTP methods.	Method
@PathVariable	Binds a URI template variable to a method parameter.	Method Parameter
@RequestParam	Binds a query parameter to a method parameter.	Method Parameter

@RequestBody	Binds the body of a request to a method parameter.	Method Parameter
@ResponseBody	Binds the method return value to the HTTP response body.	Method
@Entity	Marks a class as a JPA entity (table in a database).	Class
@Table	Specifies the table name in the database for a JPA entity.	Class
@Id	Marks a field as the primary key for a JPA entity.	Field
@GeneratedValue	Specifies the generation strategy for primary keys.	Field
@Column	Maps a field to a database column and allows customization of column properties.	Field
@Transactional	Indicates that a method or class is transactional.	Method, Class

These annotations are foundational for developing and managing Spring Boot applications efficiently.