

Frequently Asked annotation In Spring Interview

@SpringBootApplication: Combines `@SpringBootConfiguration`, `@EnableAutoConfiguration`, and `@ComponentScan` annotations to simplify configuration.

@RestController: A combination of `@Controller` and `@ResponseBody`, used to create RESTful web services.

@RequestMapping: Maps HTTP requests to handler methods in controllers, often used for defining routes.

@GetMapping, @PostMapping, @PutMapping, @DeleteMapping, @PatchMapping: Specialized annotations for mapping specific HTTP methods (GET, POST, PUT, DELETE, PATCH) to handler methods.

@Autowired: Automatically injects dependencies in Spring-managed components.

@Component, @Service, @Repository, @Controller: Stereotype annotations for different layers in a Spring application, marking classes as Spring-managed components.

@Configuration: Marks a class as a source of Spring bean definitions.

@Bean: Indicates that a method produces a bean to be managed by the Spring container.

@Transactional: Marks a method or class as transactional, managing transactions automatically.

@EnableAutoConfiguration: Enables Spring Boot's auto-configuration mechanism to automatically configure the application based on its dependencies.

@PropertySource: Adds a property file to the Spring environment.

@Value: Injects values from properties files or environment variables into fields.

@Profile: Defines beans that should be active only for specific profiles.

@PathVariable: Binds a method parameter to a URI template variable.

@RequestParam: Binds a method parameter to a query parameter in a URL.

@ExceptionHandler: Handles exceptions thrown by controller methods.

@ResponseStatus: Sets the HTTP status code in the response.

@CrossOrigin: Enables Cross-Origin Resource Sharing (CORS) for a method or class.

@Valid: Used for method parameter validation with JSR-303/JSR-380 annotations.

@Cacheable, **@CachePut**, **@CacheEvict**: Annotations for caching in Spring.