https://www.google.com/search?q=difference+between+controller+and+restcontroller&rlz=1C1GGRV\_enIN1009IN1009&oq=difference+between+controller+&aqs=chrome.0.0i512j69i57j0i512l8.8861j0j7&sourceid=chrome&ie=UTF-8

Spring

**What is difference between @service and @component?**

@Component is a generic stereotype for any Spring-managed component. **@Service annotates classes at the service layer.** **@Repository annotates classes at the persistence layer, which will act as a database repository**.

<https://www.baeldung.com/spring-component-repository-service>

**Why do we use @autowired annotation?**

The @Autowired annotation **provides more fine-grained control over where and how autowiring should be accomplished**. The @Autowired annotation can be used to autowire bean on the setter method just like @Required annotation, constructor, a property or methods with arbitrary names and/or multiple arguments.

Starting with Spring 2.5, the framework introduced annotations-driven Dependency Injection. The main annotation of this feature is @Autowired. It **allows Spring to resolve and inject collaborating beans into our bean**.

<https://www.tutorialspoint.com/spring/spring_autowired_annotation.htm>

https://www.baeldung.com/spring-autowire

**What is @RequestMapping in Spring?**

One of the most important annotations in spring is the @RequestMapping Annotation which is **used to map HTTP requests to handler methods of MVC and REST controllers**. In Spring MVC applications, the DispatcherServlet (Front Controller) is responsible for routing incoming HTTP requests to handler methods of controllers.

<https://www.geeksforgeeks.org/spring-requestmapping-annotation-with-example/>

**What is the use of @Bean annotation? Why do we need it?**

**Why should you use constructor based injection and avoid @autowired on private fields?**

Spring Boot

**What is the difference between @configuration and @EnableAutoConfiguration in Spring boot?**

**@EnableAutoConfiguration**: enable Spring Boot's auto-configuration mechanism. @ComponentScan: enable @Component scan on the package where the application is located (see the best practices)

@Configuration: allow to register extra beans in the context or import additional configuration classes.

<https://docs.spring.io/spring-boot/docs/2.0.x/reference/html/using-boot-using-springbootapplication-annotation.html>