References :

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

<https://www.geeksforgeeks.org/how-to-create-a-spring-bean-in-3-different-ways/?ref=asr3>

<https://www.digitalocean.com/community/tutorials/spring-annotations>

<https://docs.spring.io/spring-framework/reference/core/beans/java/configuration-annotation.html>

@configuration annotation

<https://www.geeksforgeeks.org/spring-configuration-annotation-with-example/?ref=asr9>

use of componentscan and configuration annotation

https://www.geeksforgeeks.org/spring-componentscan-annotation-with-example/

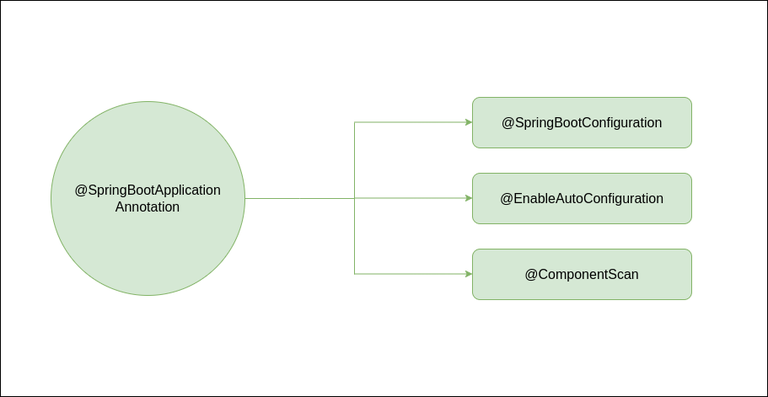
use of @Bean annotation only

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

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

**1. @SpringBootApplication Annotation**

This annotation is used to mark the main class of a Spring Boot application. It encapsulates **@SpringBootConfiguration**, **@EnableAutoConfiguration**, and **@ComponentScan**annotations with their default attributes.



@SpringBootApplication

*// Class*

**public** **class** **DemoApplication** {

*// Main driver method*

**public** **static** void main(String[] args)

{

SpringApplication.run(DemoApplication.class, args);

}

}

**@component annotation:**

It indicates that an annotated class is a “spring bean/component “

The @component annotation tells spring container to automatically create

A spring bean.

https://www.geeksforgeeks.org/spring-component-annotation-with-example/