In this tutorial I am going to list down the important spring annotations used in Spring Core, Spring MVC, Spring Boot, Spring Data JPA, etc. modules that are already published in this blog. This list will be keep updated by me when I am writing tutorial for new annotations. You can bookmark this tutorial for quick reference of most important annotations used in spring applications.

Spring Annotations Reference

ANNOTATIONS	DESCRIPTION
@Component, @Repository and @Service	@Component annotation is the more generalized form that are considered as candidates for auto-detection when using annotation-based configuration and classpath scanning. This annotation extended to more specific forms such as @Controller, @Repository and @Service.
@Autowired	Spring bean dependencies are defined in the XML files, the same can be automatically detected by the Spring container by using the @Autowired annotation. This would eliminate using the XML configurations.
@Qualifier	There may be scenarios when we create more than one bean of the same type and want to wire only one of them with a property. This

	can be controlled using @Qualifier annotation along with the **@Autowired** annotation.
@Required	**@Required** annotation applies to bean property setter methods and enforces required properties
Difference between @Resource, @Autowired and @Inject	This tutorial explains the difference between these three annotations @Resource, @Autowired and @Inject used for injecting the objects.
@Inject and @Named	@Inject and @Named annotations are JSR-330 annotations was introduced in Spring 3 as alternative for the spring annotations @Autowired and @Component.
@Resource, @PostConstruct and @PreDestroy	This tutorial explains the JSR-250 annotations that are introduced in Spring 2.5, which include **@Resource**, **@PostConstruct** and **@PreDestroy** annotations.

@RestController	@RestController annotation is inherited from the @Controller annotation. This annotation is the special version of @Controller annotation for implementing the RESTful Web Services. @RestController has been added as part of the Spring 4 release.
@RequestHeader	@RequestHeader annotation for facilitating us to get the header details easily in our controller class. This annotation would bind the header details with the method arguments, and it can be used inside the methods.
@ControllerAdvice	@ControllerAdvice annotation used for defining the exception handler with specific exception details for each method If any exception thrown on any part of the application will be handled by this component.
@ModelAttribute	@ModelAttribute annotation can be used as the method arguments or before the method declaration. The primary objective of this annotation to bind the request parameters or form fields to an model object.

@Conditional	This tutorial explains one of the new features introduced in spring 4, **conditional annotation type.**
@Query	This spring data series tutorial explains `@Query` annotation and how to create custom query using the `@Query` annotation.
@Profile	This tutorial explain how to enable profiles in your spring application for different environments.
@Configuration	Instead of using the XML files, we can use plain Java classes to annotate the configurations by using the @Configuration annotation. If you annotate a class with @Configuration annotation, it indicates that the class is used for defining the beans using the @Bean annotation.
@PathVariable	We have to use **@PathVariable** for accepting the customized or more dynamic parameters in the request paths.
@Controller, @RequestMapping,	This tutorial explains you some of the key annotations that are related to Spring MVC applications @Controller,

@RequestParam, @SessionAttributes and @InitBinder	@RequestMapping, @RequestParam, @SessionAttributes and @InitBinder
Difference between @RequestParam and @PathVariable	This tutorial explains you what is the difference between the two annotations @RequestParam and @PathVariable.
@RequestMapping	@RequestMapping annotation is used for mapping web requests to a particular handler classes or handler methods.
@Value	This tutorial shows how to load the properties file values using the @Value annotation.
@Import	**@Import** is the annotation used for consolidating all the configurations defined in various configuration files using **@Configuration** annotation.

@Transactional	Use annotation *@Transactional* in order to define a particular method that should be within a transaction.
@SpringBootApplication	This is annotation is the heart of spring boot application. @SpringBootApplication indicates that it is the entry point for the spring boot application.
@EnableAutoConfiguration	This example demonstrates how to use the @EnableAutoConfiguration annotations for auto configuring the spring boot applications.
@EnableCaching	@EnableCaching annotation is the annotation-driven cache management feature in the spring framework. This annotation has been added to the spring since the version 3.1.