**Spring Annotation**

1. **Spring BASIC ANNOTATION**

[**1.@Configuration**](mailto:1.@Configuration)

➢ Used for configuring a class by a bean

Method.

➢ It is a class level annotation.

[2.@Bean](mailto:2.@Bean)

➢ Method level annotation, used Indicate

a method as a bean configuration.

➢ We use @Bean annotation inside the

@Configuration annotation.

[3.@Component](mailto:3.@Component)

➢This annotation that allows Spring to detect

our custom beans automatically**.**

[4.@Component](mailto:4.@Component)Scan

➢ @ComponentScan annotation is used to

specify packages for spring to scan for

annotated components.

[5.@Scope](mailto:5.@Scope)

➢ We will use @Scope annotation will be used

to define scope type.

a.@Scope (“singleton”)

b.@Scope(“prototype”)

[6.@Autowired](mailto:6.@Autowired)

➢ Autowiring feature of spring framework

enables you to inject the object dependency

implicitly.

➢ Autowiring can't be used to inject primitive

and string values. It works with reference only

7.@Qualifier

➢ By using the @Qualifier annotation, we can

eliminate the issue of which bean needs to

be injected

8@Primary

➢ @Primary can use to decide which bean to

inject when ambiguity is present regarding

dependency injection

1. **Spring DATA ANNOTATION**

[9.@Entity](mailto:9.@Entity)

➢ An entity class is a Java class that

represents a row in a database table.

**10.@Table**

➢ Specifies the primary table for the

annotated entity

**11.@Column**

➢ Specifies the mapped colmn for a

persistent/POJO property or field.

[**12.@Id**](mailto:12.@Id)

➢ The field or property to which the @Id

annotation is apply that properties is

know as the Primary key of DB table.

**13.@EnableJpaRepositories("flipkart.\*")**

➢ Indicates where the repositories classes

are present

➢ We will write the @EnableJpaRepositories

In top of class with @Configuration

[**14.@Bean**](mailto:14.@Bean)**(“entityManagerFactory”)**

➢ To indicate this method is the entity-

ManagerFactory interface.

[**15.@Bean(transcationManager)**](mailto:15.@Bean(transcationManager))

➢ Integrates the JPA provider with the

Spring transaction mechanism

[**16.@Query**](mailto:16.@Query)

➢ Used for allow to write the SQL Query

In the pre-define Repository interface.

[**17.@Param**](mailto:17.@Param)

➢ Used to set the prepared statement name

Position as the SQL table fields

[**18.@Transactional**](mailto:18.@Transactional)

➢ If want to DML operation (Insert, update,

Delete ) then it allow to perform operation

Easly.

[**19.@Modifying**](mailto:19.@Modifying)

➢ The @Modifying annotation is a powerful

tool that can be used to update, delete,

and insert data in the database

**3. Spring Web MVC ANNOTATION**

**20.@Controller**

➢ @Controller annotation indicates that the

annotated class is a controller. It is a

specialization of @Component and is

autodetected through class path /

component scanning.

➢ If we define @Controller at the class level

Then we must define @ResponseBody at

The Controller method level.

➢ If we don’t pass and we pass URL that of

Method then it will throw 405 error

[**21.@ResponseBody**](mailto:21.@ResponseBody)

➢This indicates a method return value should

be bound to the web response that means

Http-Response body.

[**22.@RestController**](mailto:22.@RestController)

➢ Used to annotated a class as the Controller

class then we also pass the @RestCotroller

in the situation we don’t @ResponseBody

in Controller method level

[**23.@ResquestMapping**](mailto:23.@ResquestMapping)

➢This Annotation for mapping web requests

onto methods in request-handling classes

[**24.@GetMapping**](mailto:24.@GetMapping)

➢This annotation is acts as mapping the Get

Request of web Request into the controller

Method to handle GET mode

[**25.@PostMapping**](mailto:25.@PostMapping)

➢This annotation is acts as mapping the Post

Request of web Request into the controller

Method to handle POST mode

[**26.@PutMapping**](mailto:26.@PutMapping)

➢This annotation is acts as mapping the Put

Request of web Request into the controller

Method to handle PUT mode

[**27.@DeleteMapping**](mailto:27.@DeleteMapping)

➢This annotation is acts as mapping the Delete

Request of web Request into the controller

Method to handle DELETE mode

28.@