**JAVA SPRING**

Annotation

* + 1. **@Bean:--**

This is used before method. The @Bean annotation works with @Configuration to create Spring beans.  The method annotated with this annotation works as the bean ID, and it creates and returns the actual bean.

* + 1. **@Required:--**

The @Required annotation apply to bean property setter methods. Specifies that a particular property must be injected or else it will fail.

* + 1. **@AutoWired:--**

When we have to apply bean property Setter Method, Non-Setter Method, constructor and Properties. Item which use with Autowired annotation should not have to be public . When you use @Autowired on fields and pass the values for the fields using the property name, Spring will automatically assign the fields with the passed values.

* + 1. **@Configuration:--**

. A Java class annotated with @Configuration is a configuration by itself and will have methods to instantiate and configure the dependencies. This annotation is used on classes that define beans.

* + 1. **@component:--**

This annotation is used on classes to indicate a Spring component. Annotate your other components just like REST resource classes

* + 1. **@Controller:--**

We use controller class to annotating that particular class will work as a controller.

* + 1. **@Service:--**

Usually we write our business logic in service class so for annonate the service class We us @Service annotation

* + 1. **@RequestMapping:--**

We use the @RequestMapping spring annotation to map URLs onto an entire class or a particular handler method. It specify the path onto a controller

* + 1. **@GetMapping:--**

This annotation is used for mapping HTTP GET requests onto specific handler methods. @GetMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.GET).

* + 1. **@PostMapping:--**

This annotation is used for mapping HTTP POST requests onto specific handler methods. @PostMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.POST).

* + 1. **@PutMapping:--**

This annotation is used for mapping HTTP PUT requests onto specific handler methods. @PutMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.PUT).

* + 1. **@PatchMapping:--**

This annotation is used for mapping HTTP PATCH requests onto specific handler methods. @PatchMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.PATCH).

* + 1. **@DeleteMapping:--**

This annotation is used for mapping HTTP DELETE requests onto specific handler methods. @DeleteMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.DELETE).

* + 1. **@RequestParam:--**

We can use the @RequestParam annotation to retrieve the URL parameter and map it to the method argument. The@RequestParam annotation is used to bind request parameters to a method parameter in your controller.

* + 1. **@Qualifier:--**

The @Qualifier annotation along with @Autowired can be used to remove the confusion by specifiying which exact bean will be wired. When you need more control of the dependency injection process, @Qualifier can be used.