

Annotations in Spring Framework

1. @Controller:

By giving this annotation, we tell the Spring Framework that this class is a Controller. For Spring Framework to scan all the Annotations, we need to give the base package path in the spring configuration file. Also, it serves as a specialization for @Component

2. @Component:

It indicated that the annotated class is a Component and will be found while classpath scanning

3. @Service

Indicated that the annotated class is a Service. It is a specialization of @Component. It was originally defined as an operation that is offered as an interface and stands alone in the model with no encapsulated state.

4. @Repository

It indicated that the annotated class is a Repository. It is defined as a mechanism for performing CRUD operations.

5. @InitBinder

It identifies methods that initialize WebDataBinder and populates commands. InitBinder methods should not have return values and they are usually declared as void. It takes WebDataBinder as an argument. A WebDataBinder extends DataBinder and is used to bind web request objects to Java beans.

6. @RequestMapping

It is used for mapping web requests onto specific handler classes/methods. The methods with this annotation can have very flexible signatures and they depend on the specific Controller model.

7. @RequestParam

Indicates that the method parameter should be bound to a web request parameter.

8.@Sessionattributes

`@Sessionattributes` is used for passing the values across the different pages through the session. In simple terms, `@SessionAttributes` is replacement or simplified implementation for storing the values in the session object.

9. `@RequestBody`

Indicates that the method parameter should be bound to the web request body. The body of the request is passed through an `HttpMessageConverter` to resolve the method argument depending on the content type of the request.

10. `@ResponseBody`

Annotation that indicates a method return value should be bound to the web response body.

11. `@RestController`

A convenience annotation that is itself annotated with `@Controller` and `@ResponseBody`.

Types that carry this annotation are treated as controllers where `@RequestMapping` methods assume `@ResponseBody` semantics by default.

12. `@PathVariable`

It is one of the path parameters. They provide access to URI template variables.

13. `@MatrixVariable`

It gives annotated parameters for access to name-value pairs located in URI path segments.

Example: `@MatrixVariable(value="xyz", pathVar="storeId",)`

14. `@ModelAttribute`

Annotation that maps an argument or a return value of a method to a model attribute exposed to a web view. It is supported for Controller classes with `@RequestMapping` methods.

Can also be used to expose reference data to a web view through annotating accessor methods in a controller class with `@RequestMapping` methods. Such accessor methods are allowed to have any arguments that `@RequestMapping` methods support, returning the model attribute value to expose.

15. `@ExceptionHandler`

Annotation for handling exceptions in specific handler classes and methods. The return types of the method can be ModelAndView object / View object/ Model object/ String value/ Map object etc. The arguments could be Session object/ InputStream/ OutputStream/ Locale/ Request or response objects etc