

SSM框架注解大全

Spring Framework + Spring MVC + MyBatis + Spring Boot + Lombok + Knife4j + Spring Validation

- **Spring Framework**

是Spring生态的核心，包含2个核心特性：

- 第1个：IoC控制反转，是一种编程思想，反转资源的获取方向，把创建对象和管理对象交由框架管理；
- 第2个：AOP面向切面编程，是一种编程泛型，提高代码的可维护性、可重用性和可扩展性，降低代码的耦合度，集中管理横切关注点的代码，提高开发效率；

- **Spring MVC**

Spring MVC 是Spring框架的一个重要组成部分，提供了一种MVC即模型视图控制器的设计模式实现方式，将应用程序的控制器层、模型层、视图层分离，使得开发人员能够更加高效地构建Web应用程序。

- **MyBatis**

MyBatis是一个数据持久层框架，提供了简单易用的接口，通过对象关系映射让开发者能够更轻松地操作关系型数据库；

主要通过使用SQL语句和配置文件来实现，消除了几乎所有的JDBC代码和参数的手动设置，以及对结果集的检索。

- **Spring Boot**

Spring Boot 是一个快速搭建工程环境的脚手架，核心特性有：

- 起步依赖：创建项目时，会默认添加基础依赖，简化我们自己查找依赖的过程。
- 自动配置：在构建Spring应用时，只需添加相应的场景依赖，Spring Boot就会根据添加的场景依赖自动进行配置，在无须额外手动添加配置的情况下快速构建出一个独立的Spring应用。
- 内嵌Web服务：SpringBoot工程支持内嵌的web服务，可以将tomcat这样的服务直接嵌套到web依赖中，并且可以自动将项目打包，并在项目运行时部署到服务器中。

- **Lombok**

Lombok是一个Java库，主要作用有两个：

- 可以通过注解自动生成 `setter` `getter` `toString` 等方法，减少重复代码，提高开发效率；
- 通过 `@Slf4j` 注解使用其提供的日志功能，可以有效的帮助开发者调试程序和管理服务器资源消耗。

- **Knife4j**

`Knife4j` 主要功能为：

- 接口测试：内置了接口测试功能，可以在文档界面上进行接口的测试和调试；
- 生成离线API文档：通过注解和页面操作，可以直接生成API文档，无需额外的配置和编写文档，提高了开发效率。

- **Spring Validation**

`Spring Validation` 是一个数据验证框架，用于对Java对象进行校验；

可通过注解的方式对数据进行校验，避免开发人员手动校验数据的同时，又提高了系统的安全性和可靠性，有效地避免非法参数和恶意攻击。

	注解	所属框架	作用
1	@Component	Spring	添加在类上，标记当前类是组件类，可以通过参数配置Spring Bean名称
2	@Controller	Spring	添加在类上，标记当前类是控制器组件类，用法同 @Component
3	@Service	Spring	添加在类上，标记当前类是业务逻辑组件类，用法同 @Component
4	@Repository	Spring	添加在类上，标记当前类是数据访问组件类，用法同 @Component
5	@Configuration	Spring	添加在类上，仅添加此注解的类才被视为配置类，通常不配置注解参数
6	@ComponentScan	Spring	添加在配置类上，开启组件扫描。 如果没有配置包名，则扫描当前配置类所在的包， 如果配置了包名，则扫描所配置的包及其子孙包
7	@Value	Spring	添加在属性上，或添加在被Spring调用的方法的参数上，用于读取 Environment 中的属性值，为对象的属性或方法的参数注入值
8	@Autowired	Spring	添加在属性上，使得Spring自动装配此属性的值 添加在构造方法上，使得Spring自动调用此构造方法 添加在Setter方法上，使得Spring自动调用此方法
9	@Qualifier	Spring	添加在属性上，或添加在方法的参数上，配合自动装配机制，用于指定需要装配的Spring Bean的名称
10	@Resource	Spring	此注解是 javax 包中的注解，添加在属性上，使得Spring自动装配此属性的值，通常不推荐使用此注解
11	@Scope	Spring	添加在组件类上，或添加在已经添加了 @Bean 注解的方法上， 用于指定作用域，注解参数为 singleton（默认）时为“单例”，注解参数为 prototype 时为“非单例”
12	@Configuration	Spring	添加在配置类上，用于代替传统的xml配置文件， Spring会在启动时自动加载带有 @Configuration 注解的类

	注解	所属框架	作用
13	<code>@Aspect</code>	Spring	<p>添加在切面类上，属于 AOP 的注解</p> <p>用于告诉Spring容器将该类识别为一个切面，它定义了切入点(Pointcut)和通知(Advice)，切入点定义了应用程序中哪些方法或代码块应该被拦截和执行额外的逻辑，通知包括前置通知、返回通知、环绕通知、异常通知和最终通知</p>
14	<code>@Before</code>	Spring	前置通知，添加在切面类方法上，在目标方法执行之前执行的通知
15	<code>@After</code>	Spring	后置通知，添加在切面类方法上，在目标方法最终执行之后执行的通知
16	<code>@AfterReturning</code>	Spring	返回通知，添加在切面类方法上，在目标方法正常返回之后执行的通知
17	<code>@AfterThrowing</code>	Spring	异常通知，添加在切面类方法上，在目标方法抛出异常之后执行的通知
18	<code>@Around</code>	Spring	环绕通知，添加在切面类方法上，将目标方法封装起来，在目标方法调用之前和之后执行自定义的行为，通常使用 <code>try ... catch ... finally</code> 包裹
19	<code>@Mapper</code>	Mybatis	添加在Mapper接口上，用于标记此接口是Mybatis的Mapper接口
20	<code>@MapperScan</code>	Mybatis	添加在配置类上，用于指定Mapper接口的根包，Mybatis将根据此根包执行扫描，以找到各Mapper接口
21	<code>@Select</code>	Mybatis	<p>添加在Mapper接口的抽象方法上，可以通过此注解直接配置此抽象方法对应的SQL语句（不必将SQL语句配置在XML文件中），用于配置 <code>SELECT</code> 类的SQL语句，但是，非常不推荐这种做法</p>
22	<code>@Insert</code>	Mybatis	同上，用于配置 <code>INSERT</code> 类的SQL语句
23	<code>@Update</code>	Mybatis	同上，用于配置 <code>UPDATE</code> 类的SQL语句
24	<code>@Delete</code>	Mybatis	同上，用于配置 <code>DELETE</code> 类的SQL语句
25	<code>@Param</code>	Mybatis	<p>添加在接口方法参数前，</p> <p>用于传递多个参数给SQL语句，在Mapper接口的方法中定义多个参数时，需要使用 <code>@Param</code> 注解来标识每个参数的名字，这样Mybatis就可以将参数值绑定到SQL语句中对应的参数上</p>

	注解	所属框架	作用
26	<code>@SpringBootApplication</code>	Spring Boot	添加在类上，用于标记此类是Spring Boot的启动类，每个Spring Boot项目应该只有1个类添加了此注解
27	<code>@SpringBootTest</code>	Spring Boot	添加在类上，用于标记此类是加载Spring环境的测试类
28	<code>@Test</code>	Spring Boot	添加在方法上，用于标记此方法是Spring环境的测试方法
29	<code>@RequestMapping</code>	Spring MVC	添加在类上，也可以添加在处理请求的方法上，通常用于配置请求路径
30	<code>@ResponseBody</code>	Spring MVC	添加在方法上，标记此方法是“响应正文”的，添加在类上，标记此类中所有方法都是“响应正文”的
31	<code>@RestController</code>	Spring MVC	添加在类上，标记此类是一个“响应正文”的控制器类
32	<code>@GetMapping</code>	Spring MVC	添加在方法上，是将请求方式限制为 GET 的 <code>@RequestMapping</code>
33	<code>@PostMapping</code>	Spring MVC	添加在方法上，是将请求方式限制为 POST 的 <code>@RequestMapping</code>
34	<code>@DeleteMapping</code>	Spring MVC	添加在方法上，是将请求方式限制为 DELETE 的 <code>@RequestMapping</code>
35	<code>@PutMapping</code>	Spring MVC	添加在方法上，是将请求方式限制为 PUT 的 <code>@RequestMapping</code>
36	<code>@PathVariable</code>	Spring MVC	添加在请求参数上，用于标记此参数的值来自URL中的占位符，如果URL中的占位符名称与方法的参数名称不同，需要配置此注解参数来指定URL中的占位符名称
37	<code>@RequestBody</code>	Spring MVC	添加在请求参数上，用于标记此参数必须是对象格式的参数，如果未添加此注解，参数必须是FormData格式的
38	<code>@RequestParam</code>	Spring MVC	添加在请求参数上，用于将请求参数值绑定到控制器方法的参数上，同时方便使用 Knife4j 测试
39	<code>@ExceptionHandler</code>	Spring MVC	添加在方法上，标记此方法是处理异常的方法，可以通过配置注解参数来指定需要处理的异常类型，如果没有配置注解参数，所处理的异常类型取决于方法的参数列表中的异常类型
40	<code>@ControllerAdvice</code>	Spring MVC	添加在类上，标记此类中特定的方法将作用于每次处理请求的过程中

	注解	所属框架	作用
41	<code>@RestControllerAdvice</code>	Spring MVC	添加在类上，是 <code>@ControllerAdvice</code> 和 <code>@ResponseBody</code> 的组合注解
42	<code>@Data</code>	Lombok	添加在类上，将在编译期生成此类中所有属性的 Setter、Getter 方法，及 <code>hashCode()</code> 、 <code>equals()</code> 、 <code>toString()</code> 方法
43	<code>@Setter</code>	Lombok	添加在类上，将在编译期生成此类中所有属性的 Setter 方法，也可以添加在类的属性上，将在编译期生成此属性的 Setter 方法
44	<code>@Getter</code>	Lombok	添加在类上，将在编译期生成此类中所有属性的 Getter 方法，也可以添加在类的属性上，将在编译期生成此属性的 Getter 方法
45	<code>@ToString</code>	Lombok	添加在类上，将在编译期生成基于此类中所有属性的 <code>toString()</code> 方法
46	<code>@NoArgConstructor</code>	Lombok	添加在类上，将在编译期生成此类的无参数构造方法
47	<code>@AllArgsConstructor</code>	Lombok	添加在类上，将在编译期生成基于此类中所有属性的全参构造方法
48	<code>@Api</code>	Knife4j	添加在控制器类上，通过此注解的 <code>tags</code> 属性配置 API 文档中的模块名称
49	<code>@ApiOperation</code>	Knife4j	添加在控制器类中处理请求的方法上，用于配置业务名称
50	<code>@ApiModelProperty</code>	Knife4j	添加在封装的请求参数类型中的属性上，用于配置请求参数的详细说明，包括：名称、数据类型、是否必须等
51	<code>@ApiImplicitParam</code>	Knife4j	添加在控制器类中处理请求的方法上，用于配置请求参数的详细说明，包括：名称、数据类型、是否必须等
52	<code>@ApiImplicitParams</code>	Knife4j	添加在控制器类中处理请求的方法上，如果需要通过 <code>@ApiImplicitParam</code> 注解配置的参数超过 1 个，则必须将多个 <code>@ApiImplicitParam</code> 注解作为此注解的参数
53	<code>@ApiIgnore</code>	Knife4j	添加在请求参数上，用于标记 API 文档中将不关心此参数
54	<code>@Validated</code>	Spring Validation	添加在方法的参数上，标记此参数需要经过 Validation 框架的检查；添加在类上，并结合方法上的检查注解（例如 <code>@NotNull</code> 等）实现对未封装的参数的检查

	注解	所属框架	作用
55	@NotNull	Spring Validation	添加在需要被检查的参数上，或添加在需要被检查的封装类型的属性上，用于配置“不允许为null”的检查规则
56	@NotEmpty	Spring Validation	使用位置同 @NotNull，用于配置“不允许为空字符串”的检查规则
57	@NotBlank	Spring Validation	使用位置同 @NotNull，用于配置“不允许为空白”的检查规则
58	@Pattern	Spring Validation	使用位置同 @NotNull，用于配置正则表达式的检查规则
59	@Range	Spring Validation	使用位置同 @NotNull，用于配置“数值必须在某个取值区间”的检查规则