Spring常用注解大全

# ****Spring部分****

### **1.声明bean的注解**

****@Component**** 组件，没有明确的角色

****@Service**** 在业务逻辑层使用（service层）

****@Repository****在数据访问层使用（dao层）

****@Controller**** 在展现层使用，控制器的声明（C）

### **2.注入bean的注解**

@Autowired：由Spring提供

@Inject：由JSR-330提供

@Resource：由JSR-250提供

都可以注解在set方法和属性上，推荐注解在属性上（一目了然，少写代码）。

### **3.java配置类相关注解**

@Configuration 声明当前类为配置类，相当于xml形式的Spring配置（类上）

@Bean 注解在方法上，声明当前方法的返回值为一个bean，替代xml中的方式（方法上）

@Configuration 声明当前类为配置类，其中内部组合了@Component注解，表明这个类是一个bean（类上）

@ComponentScan 用于对Component进行扫描，相当于xml中的（类上）

@WishlyConfiguration 为@Configuration与@ComponentScan的组合注解，可以替代这两个注解

### **4.切面（AOP）相关注解**

Spring支持AspectJ的注解式切面编程。

****@Aspect**** 声明一个切面（类上）   
使用****@After、@Before、@Around****定义建言（advice），可直接将拦截规则（切点）作为参数。

****@After****在方法执行之后执行（方法上）   
****@Before**** 在方法执行之前执行（方法上）   
****@Around**** 在方法执行之前与之后执行（方法上）

****@PointCut****声明切点   
在java配置类中使用@EnableAspectJAutoProxy注解开启Spring对AspectJ代理的支持（类上）

### **5.@Bean的属性支持**

****@Scope**** 设置Spring容器如何新建Bean实例（方法上，得有@Bean）   
其设置类型包括：

Singleton （单例,一个Spring容器中只有一个bean实例，默认模式）,   
Protetype （每次调用新建一个bean）,   
Request （web项目中，给每个http request新建一个bean）,   
Session （web项目中，给每个http session新建一个bean）,   
GlobalSession（给每一个 global http session新建一个Bean实例）

****@StepScope**** 在Spring Batch中还有涉及

****@PostConstruct**** 由JSR-250提供，在构造函数执行完之后执行，等价于xml配置文件中bean的initMethod

****@PreDestory**** 由JSR-250提供，在Bean销毁之前执行，等价于xml配置文件中bean的destroyMethod

### **6.@Value注解**

****@Value****为属性注入值（属性上）   
支持如下方式的注入：   
》注入普通字符

@Value("Michael Jackson")

String name;

》注入操作系统属性

@Value("#{systemProperties['os.name']}")

String osName;

》注入表达式结果

@Value("#{ T(java.lang.Math).random() \* 100 }")

String randomNumber;

》注入其它bean属性

@Value("#{domeClass.name}")

String name;

》注入文件资源

@Value("classpath:com/hgs/hello/test.txt")

String Resource file;

》注入网站资源

@Value("http://www.cznovel.com")

Resource url;

》****注入配置文件****

@Value("${book.name}")

String bookName;

注入配置使用方法：   
① 编写配置文件（test.properties）

book.name=《三体》

② @PropertySource 加载配置文件(类上)

@PropertySource("classpath:com/hgs/hello/test/test.propertie")

③ 还需配置一个PropertySourcesPlaceholderConfigurer的bean。

### **7.环境切换**

@Profile 通过设定Environment的ActiveProfiles来设定当前context需要使用的配置环境。（类或方法上）

****@Conditional****Spring4中可以使用此注解定义条件话的bean，通过实现Condition接口，并重写matches方法，从而决定该bean是否被实例化。（方法上）

### **8.异步相关**

****@EnableAsync**** 配置类中，通过此注解开启对异步任务的支持，叙事性AsyncConfigurer接口（类上）

****@Async**** 在实际执行的bean方法使用该注解来申明其是一个异步任务（方法上或类上*所有的方法都将异步*，需要@EnableAsync开启异步任务）

### **9.定时任务相关**

****@EnableScheduling**** 在配置类上使用，开启计划任务的支持（类上）

****@Scheduled**** 来申明这是一个任务，包括cron,fixDelay,fixRate等类型（方法上，需先开启计划任务的支持）

### ****10.@Enable\*注解说明****

这些注解主要用来开启对xxx的支持。   
****@EnableAspectJAutoProxy**** 开启对AspectJ自动代理的支持

****@EnableAsync**** 开启异步方法的支持

****@EnableScheduling****开启计划任务的支持

****@EnableWebMvc**** 开启Web MVC的配置支持

****@EnableConfigurationProperties****开启对@ConfigurationProperties注解配置Bean的支持

****@EnableJpaRepositories**** 开启对SpringData JPA Repository的支持

****@EnableTransactionManagement**** 开启注解式事务的支持

****@EnableTransactionManagement****开启注解式事务的支持

****@EnableCaching**** 开启注解式的缓存支持

11.测试相关注解

****@RunWith**** 运行器，Spring中通常用于对JUnit的支持

****@ContextConfiguration**** 用来加载配置ApplicationContext，其中classes属性用来加载配置类

@RunWith(SpringJUnit4ClassRunner.class)

@ContextConfiguration(classes={TestConfig.class})

public class KjtTest {

private static Logger logger = LoggerFactory.getLogger("KjtTest");

@Autowired

Service service;

@Test

public void test() {

}

}