**这些比较常用的注入**

**使用注入之前需要在Spring配置文件中**

**<context:component-scan base-package="com.blog"></context:component-scan>**

**@Service**

**Service层注解**



**@Repositroy**

**Dao层注解**



**@Controller**

**控制层注解**



**前提：**

**如果是SSH整合，则需要导入Struts2-Spring-plugs.jar包**

**并且在web.xml中配置spring监听器**

<!-- 指定多个配置文件 -->

<context-param>

<!-- 参数名为contextConfigLocation -->

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/classes/applicationContext.xml</param-value>

</context-param>

<!-- 使用ContextLoaderListener 初始化spring容器 -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<filter>

<filter-name>openSessionInVieFilter</filter-name>

<filter-class>org.springframework.orm.hibernate5.support.OpenSessionInViewFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>openSessionInVieFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

**还有配置struts的过滤器**

<filter>

<filter-name>struts2</filter-name>

<!-- <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class> -->

<filter-class>com.blog.filter.UeditorFilter</filter-class>

</filter>

<filter-mapping>

<filter-name>struts2</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

**@Trancational**

**事务层注解**



**前提在Spring配置文件中**

<bean id=*"transactionManager"* class=*"org.springframework.orm.hibernate5.HibernateTransactionManager"*>

<property name=*"sessionFactory"* ref=*"sessionFactory"* />

</bean>

<tx:annotation-driven transaction-manager=*"transactionManager"*/>

**@Component**

**当不知道是什么容器的时候可以使用**



**@Resource**

**和**

**@Autowired**

**效果差不多**

**可以省去set方法注入**

