1. 注入(spring.jar、commons-logging.jar)
2. 注入類別

|  |
| --- |
| 配置bean-config.xml |
| <!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN" "http://www.springframework.org/dtd/spring-beans-2.0.dtd" >  <beans>  <bean id=*"Cat1"* class=*"com.willy.model.CatModel"*>  <property name=*"name"* value=*"cat1"*></property>  <property name=*"color"* value=*"white"*></property>  <property name=*"age"* value=*"5"*></property>  </bean>  </beans> |
| 建立Model POJO Bean |
| 引入Bean |
| ApplicationContext ac=**new** ClassPathXmlApplicationContext("xml位置");  Ex: "com/willy/test/spring/bean-config.xml"  CatModel cat=(CatModel) ac.getBean("Cat1");  System.***out***.println(cat.getName()); |

1. 構造注入

|  |
| --- |
| 配置好一個Bean🡪Cat1 |
| 配置bean-config.xml |
| <bean id=*"constructorCat"* class=*"com.willy.test.spring.ConstructorInjection"*>  <constructor-arg ref=*"Cat1"*></constructor-arg>  </bean> |
| 在被注入類別以構造函數承接 |
| CatModel cat;  **public** ConstructorInjection(CatModel cat) {  **super**();  **this**.cat = cat;  } |
| 被建立時即自動載入Bean |

1. Scope(Singleton/Prototype)

|  |  |
| --- | --- |
| Singleton | 每次被引用指向同一個記憶體位置 |
| Prototype | 每次引用在新的記憶體位置創造新的物件 |
| <bean id="Cat2" scope="prototype" class="com.willy.model.CatModel"> | |



1. 配置List、Map

|  |
| --- |
| 配置好兩個Bean🡪Cat1 |
| 配置Map、List的Bean |
| 配置bean-config.xml |
| <bean id=*"collection"* class=*"com.willy.model.CollectionCat"*>  <property name=*"list"*>  <list>  <ref bean=*"Cat1"*/>  <ref bean=*"Cat2"*/>  </list>  </property>  <property name=*"map"*>  <map>  <entry>  <key><value>firstKey</value></key>  <ref bean=*"Cat1"*/>  </entry>  <entry>  <key><value>secondKey</value></key>  <ref bean=*"Cat2"*/>  </entry>  </map>  </property>  </bean> |

1. AOP
2. 建立Interface
3. 建立Imp
4. 建立AOP(MethodBeforeAdvice,AfterReturningAdvice

,MethodInterceptor,ThrowsAdvice)

1. 設定Bean-config.xml

--建立一個實體Bean

<bean id=*"test"* class=*"com.willy.model.TestServiceImpl"*>

<property name=*"name"* value=*"willy"*></property>

</bean>

--建立一個AOP Bean

<bean id=*"MyMethodBeforeAdvice"* class=*"com.willy.model.MyMethodBeforeAdvice"*></bean>

--建立一個ProxyBean

<bean id=*"proxyFactoryBean1"* class=*"org.springframework.aop.framework.ProxyFactoryBean"*>

-+-設置要proxy的Interface

<property name=*"proxyInterfaces"*>

<list>

<value>com.willy.interfaces.TestService</value>

<value>com.willy.interfaces.TestService1</value>

</list>

</property>

-+-設置AOP類別

<property name=*"interceptorNames"*>

<list>

<value>MyMethodBeforeAdvice</value>

</list>

</property>

-+-引入被AOP的實體Bean

<property name=*"target"* ref=*"test"*></property>

</bean>