

07 Spring_配置详解_复杂类型注入

Spring

一:实体类

```
1. public class Collection {
2.     private Object[] obj;
3.     private List list;
4.     private Map map;
5.     private Properties pr;
6.     //对应的get set toString方法
7. }
```

二:数组

1、单个值

1、配置文件

2、如数组中只有一个值|对象可以使用 value=""|ref 的方式注入

```
1. <!-- 数组注入
2.     |-如数组中只有一个值可以使用value=""的方式注入
3.     -->
4. <bean name="coll01" class="com.spring.bean.Collection">
5.     <!--
6.     |-如数组中只有一个值|对象可以使用 value=""|ref 的方式注入-->
7.     <property name="obj" value="小王"></property>
8. </bean>
```

3、测试

```
1. @org.junit.Test
```

```

2.     public void getVoid02() {
3.         //1.创建容器对象
4.         ClassPathXmlApplicationContext classPath= new
ClassPathXmlApplicationContext("applicationContext.xml");
5.         //2.找容器要对象
6.         Collection bean = (Collection)classPath.getBean("coll01");
7.         //3.打印对象单列为true
8.         System.out.println("bean:"+bean);
9.     }

```

2、多个值

1、配置文件

```

1.     <bean name="coll01" class="com.spring.bean.Collection">
2.         <!-- 如多个注入值
3.             value: 属性方式注入
4.             ref: 对象方式注入
5.         -->
6.         <property name="obj">
7.             <array>
8.                 <value>小张</value>
9.                 <value>小陈</value>
10.                <ref bean="car"/>
11.            </array>
12.        </property>
13.    </bean>
14.
15.    <bean name="car" class="com.spring.bean.Car">
16.        <property name="name" value="奔驰" ></property>
17.        <property name="clor" value="红色"></property>
18.    </bean>

```

2、测试同上

三:List|Set集合

1、配置文件

```

1. <bean name="coll01" class="com.spring.bean.Collection">
2.     <!-- 如多个注入值
3.         value: 属性方式注入
4.         ref: 对象方式注入
5.     -->
6.         <property name="list">
7.             <list>
8.                 <value>jack</value>
9.                 <value>小陈</value>
10.                <ref bean="car" />
11.            </list>
12.        </property>
13. </bean>
14.
15. <bean name="car" class="com.spring.bean.Car">
16.     <property name="name" value="奔驰"></property>
17.     <property name="clor" value="红色"></property>
18. </bean>

```

2、测试同上

四:Map

1、配置文件

```

1. <bean name="coll01" class="com.spring.bean.Collection">
2.     <!-- 如多个注入值
3.         value: 属性方式注入
4.         key-ref|value-ref: 对象方式注入
5.     -->
6.         <property name="map">
7.             <map>
8.                 <entry key="name" value="小张"></entry>
9.                 <entry key="car" value-ref="car"></entry>
10.                <entry key-ref="car" value="car"></entry>
11.            </map>
12.        </property>
13. </bean>

```

```
14.
15.     <bean name="car" class="com.spring.bean.Car">
16.         <property name="name" value="奔驰"></property>
17.         <property name="clor" value="红色"></property>
18.     </bean>
```

2、测试同上

四:Properties

1、配置文件

```
1.     <bean name="coll01" class="com.spring.bean.Collection">
2.         <property name="pr">
3.             <props>
4.                 <prop
5. key="driverClass">com.jdbc.mysql.Driver</prop>
6.                 <prop key="userName">root</prop>
7.             </props>
8.         </property>
9.     </bean>
10.
11.     <bean name="car" class="com.spring.bean.Car">
12.         <property name="name" value="奔驰"></property>
13.         <property name="clor" value="红色"></property>
14.     </bean>
```

2、测试同上

五:null值注入

1、使用< null/>元素显示的注入一个null值

```
1.     <bean class="ExampleBean">
2.         <property name="email">
```

```
3.         <null/>
4.     </property>
5. </bean>
```