**Spring Collection Injection**

In our previous examples, we have seen that we can inject single value using **value** attribute also

We can inject a Bean using **ref** attribute. But what about the Collection attribute or plural values?

Spring allow us to pass plural value , so it has four different types of attributes by which we can pass Collection values.

**Here are the list of elements**

|  |  |
| --- | --- |
| <list> | By this tag, we can inject list values. As you know list accepts duplicate so it allows duplicate. |
| <set> | By this tag, we can inject set values. As you know set does not accept duplicate so it does not allow duplicate. |
| <map> | By this tag, we can inject name value pair. As you know map accepts name /value pair and it can be any data type. |
| <props> | By this tag, we can inject name value pair but here both are Strings. |

**Now we will take an Example where we can use all the collection Elements.**

We will take a simple example of Person where I declare person’s visited place as List.Hobby as Set , Education as Map and relative as Properties.

**1.Collection.xml :**

We define a bean called Person , It has four collection peroperties

1. Visited Place : which is list , Please note that ,as visitedPlace is a list it contains duplicate value.

2. hobbySet : which is Set , Please note that as hobbySet is a Set , although we provide duplicate values but on outcome it contains only unique values.

3. educationMap : which is a Map , Please note that we set education standards as Key and organization name as a value.

4. relativeProperty : Which is Property, Property is special type of Map whose key and Value both are String. Here we treat relationship as Key and name as a value.

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="person" class="com.example.collection.Person">

<property name="visitedPlaceList">

<list>

<value>Delhi</value>

<value>Mumbai</value>

<value>Kolkata</value>

<value>Kolkata</value>

</list>

</property>

<property name="hobbySet">

<set>

<value>Photography</value>

<value>Surfing</value>

<value>Rafting</value>

<value>Rafting</value>

</set>

</property>

<property name="educationMap">

<map>

<entry key="10" value="Scottish Chruch Scool"/>

<entry key="12" value="BD GOVT School"/>

<entry key="14" value="Techno India"/>

</map>

</property>

<property name="relativeProperty">

<props>

<prop key="Father">Samir Kumar Mitra</prop>

<prop key="Mother">Late Subhra Mitra</prop>

<prop key="Spouse">Swatika Mitra</prop>

</props>

</property>

</bean>

</beans>

**2. Person Java Class:** Create Person java class.

package com.example.collection;

import java.util.List;

import java.util.Map;

import java.util.Properties;

import java.util.Set;

public class Person {

List<String> visitedPlaceList;

Set<String> hobbySet;

Map<Integer,String> educationMap;

Properties relativeProperty;

public List<String> getVisitedPlaceList() {

return visitedPlaceList;

}

public void setVisitedPlaceList(List<String> visitedPlaceList) {

this.visitedPlaceList = visitedPlaceList;

}

public Set<String> getHobbySet() {

return hobbySet;

}

public void setHobbySet(Set<String> hobbySet) {

this.hobbySet = hobbySet;

}

public Map<Integer, String> getEducationMap() {

return educationMap;

}

public void setEducationMap(Map<Integer, String> educationMap) {

this.educationMap = educationMap;

}

public Properties getRelativeProperty() {

return relativeProperty;

}

public void setRelativeProperty(Properties relativeProperty) {

this.relativeProperty = relativeProperty;

}

@Override

public String toString() {

return "Person [visitedPlaceList=" + visitedPlaceList + ", hobbySet="

+ hobbySet + ", educationMap=" + educationMap

+ ", relativeProperty=" + relativeProperty + "]";

}

}

**3. Main.java class:** Get the Person bean from Application context and print it properties.

package com.example.collection;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {

public static void main(String[] args) {

ApplicationContext ctx = new ClassPathXmlApplicationContext("configFiles/collection.xml");

Person person =(Person) ctx.getBean("person");

System.out.println(person);

}

}

**Output:**

Person [visitedPlaceList=[Delhi, Mumbai, Kolkata, Kolkata], hobbySet=[Photography, Surfing, Rafting], educationMap={10=Scottish Chruch Scool, 12=BD GOVT School, 14=Techno India}, relativeProperty={Father=Samir Kumar Mitra, Mother=Late Subhra Mitra, Spouse=Swatika Mitra}]

Please note that we are doing setter injection here. Also in hobbySet, we have provided duplicate value but it takes unique values.