Setter Injection with Non-String Map (having dependent Object) Example

1. [Setter Injection with Non-String Map](http://www.javatpoint.com/spring-tutorial-setter-injection-with-non-string-map)

In this example, we are using **map** as the answer that have Answer and User. Here, we are using key and value pair both as an object. Answer has its own information such as answerId, answer and postedDate, User has its own information such as userId, username, emailId.

Like previous examples, it is the example of forum where **one question can have multiple answers**.

**Question.java**

This class contains three properties, getters & setters and displayInfo() method to display the information.

1. **package** com.javatpoint;
2. **import** java.util.Iterator;
3. **import** java.util.Map;
4. **import** java.util.Set;
5. **import** java.util.Map.Entry;
7. **public** **class** Question {
8. **private** **int** id;
9. **private** String name;
10. **private** Map<Answer,User> answers;
12. //getters and setters

15. **public** **void** displayInfo(){
16. System.out.println("question id:"+id);
17. System.out.println("question name:"+name);
18. System.out.println("Answers....");
19. Set<Entry<Answer, User>> set=answers.entrySet();
20. Iterator<Entry<Answer, User>> itr=set.iterator();
21. **while**(itr.hasNext()){
22. Entry<Answer, User> entry=itr.next();
23. Answer ans=entry.getKey();
24. User user=entry.getValue();
25. System.out.println("Answer Information:");
26. System.out.println(ans);
27. System.out.println("Posted By:");
28. System.out.println(user);
29. }
30. }
31. }

**Answer.java**

1. **package** com.javatpoint;
3. **import** java.util.Date;
5. **public** **class** Answer {
6. **private** **int** id;
7. **private** String answer;
8. **private** Date postedDate;
9. **public** Answer() {}
10. **public** Answer(**int** id, String answer, Date postedDate) {
11. **super**();
12. **this**.id = id;
13. **this**.answer = answer;
14. **this**.postedDate = postedDate;
15. }
17. **public** String toString(){
18. **return** "Id:"+id+" Answer:"+answer+" Posted Date:"+postedDate;
19. }
20. }

**User.java**

1. **package** com.javatpoint;
3. **public** **class** User {
4. **private** **int** id;
5. **private** String name,email;
6. **public** User() {}
7. **public** User(**int** id, String name, String email) {
8. **super**();
9. **this**.id = id;
10. **this**.name = name;
11. **this**.email = email;
12. }
14. **public** String toString(){
15. **return** "Id:"+id+" Name:"+name+" Email Id:"+email;
16. }
17. }

**applicationContext.xml**

The **key-ref** and **value-ref** attributes of entry **element** is used to define the reference of bean in the map.

1. <?xml version="1.0" encoding="UTF-8"?>
2. <beans
3. xmlns="http://www.springframework.org/schema/beans"
4. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5. xmlns:p="http://www.springframework.org/schema/p"
6. xsi:schemaLocation="http://www.springframework.org/schema/beans
7. http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
9. <bean id="answer1" **class**="com.javatpoint.Answer">
10. <property name="id" value="1"></property>
11. <property name="answer" value="Java is a Programming Language"></property>
12. <property name="postedDate" value="12/12/2001"></property>
13. </bean>
14. <bean id="answer2" **class**="com.javatpoint.Answer">
15. <property name="id" value="2"></property>
16. <property name="answer" value="Java is a Platform"></property>
17. <property name="postedDate" value="12/12/2003"></property>
18. </bean>
20. <bean id="user1" **class**="com.javatpoint.User">
21. <property name="id" value="1"></property>
22. <property name="name" value="Arun Kumar"></property>
23. <property name="email" value="arun@gmail.com"></property>
24. </bean>
25. <bean id="user2" **class**="com.javatpoint.User">
26. <property name="id" value="2"></property>
27. <property name="name" value="Varun Kumar"></property>
28. <property name="email" value="Varun@gmail.com"></property>
29. </bean>
31. <bean id="q" **class**="com.javatpoint.Question">
32. <property name="id" value="1"></property>
33. <property name="name" value="What is Java?"></property>
34. <property name="answers">
35. <map>
36. <entry key-ref="answer1" value-ref="user1"></entry>
37. <entry key-ref="answer2" value-ref="user2"></entry>
38. </map>
39. </property>
40. </bean>
42. </beans>

**Test.java**

This class gets the bean from the applicationContext.xml file and calls the displayInfo() method to display the information.

1. **package** com.javatpoint;
3. **import** org.springframework.beans.factory.BeanFactory;
4. **import** org.springframework.beans.factory.xml.XmlBeanFactory;
5. **import** org.springframework.core.io.ClassPathResource;
6. **import** org.springframework.core.io.Resource;
8. **public** **class** Test {
9. **public** **static** **void** main(String[] args) {
10. Resource r=**new** ClassPathResource("applicationContext.xml");
11. BeanFactory factory=**new** XmlBeanFactory(r);
13. Question q=(Question)factory.getBean("q");
14. q.displayInfo();
16. }
17. }