

# Mapping Map in collection mapping using xml file

Hibernate allows you to map Map elements with the RDBMS. As we know, list and map are index-based collections. In case of map, index column works as the key and element column works as the value.

## Example of Mapping Map in collection mapping using xml file

You need to create following pages for mapping map elements.

- Question.java
- question.hbm.xml
- hibernate.cfg.xml
- StoreTest.java
- FetchTest.java

### Question.java

```
package com.javatpoint;

import java.util.Map;

public class Question {
    private int id;
    private String name,username;
    private Map<String,String> answers;

    public Question() {}
    public Question(String name, String username, Map<String, String> answers) {
        super();
        this.name = name;
        this.username = username;
        this.answers = answers;
    }
    public int getId() {
        return id;
    }
}
```

```
public void setId(int id) {  
    this.id = id;  
}  
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public String getUsername() {  
    return username;  
}  
public void setUsername(String username) {  
    this.username = username;  
}  
public Map<String, String> getAnswers() {  
    return answers;  
}  
public void setAnswers(Map<String, String> answers) {  
    this.answers = answers;  
}  
}
```

### question.hbm.xml

```
<?xml version='1.0' encoding='UTF-8'?>  
<!DOCTYPE hibernate-mapping PUBLIC  
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"  
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">  
  
<hibernate-mapping>  
  
<class name="com.javatpoint.Question" table="question736">  
<id name="id">  
<generator class="native"></generator>  
</id>
```

```
<property name="name"></property>
<property name="username"></property>

<map name="answers" table="answer736" cascade="all">
<key column="questionid"></key>
<index column="answer" type="string"></index>
<element column="username" type="string"></element>
</map>
</class>

</hibernate-mapping>
```

### hibernate.cfg.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<!-- Generated by MyEclipse Hibernate Tools.           -->
<hibernate-configuration>

    <session-factory>
        <property name="hbm2ddl.auto">update</property>
        <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
        <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
        <property name="connection.username">system</property>
        <property name="connection.password">oracle</property>
        <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>

        <mapping resource="question.hbm.xml"/>
    </session-factory>

</hibernate-configuration>
```

### StoreTest.java

```
package com.javatpoint;
```

```
import java.util.HashMap;
import org.hibernate.*;
import org.hibernate.cfg.*;
public class StoreTest {
    public static void main(String[] args) {
        Session session=new Configuration().configure().buildSessionFactory().openSession();
        Transaction tx=session.beginTransaction();

        HashMap<String,String> map1=new HashMap<String,String>();
        map1.put("java is a programming language","John Milton");
        map1.put("java is a platform","Ashok Kumar");

        HashMap<String,String> map2=new HashMap<String,String>();
        map2.put("servlet technology is a server side programming","John Milton");
        map2.put("Servlet is an Interface","Ashok Kumar");
        map2.put("Servlet is a package","Rahul Kumar");

        Question question1=new Question("What is java?","Alok",map1);
        Question question2=new Question("What is servlet?","Jai Dixit",map2);

        session.persist(question1);
        session.persist(question2);

        tx.commit();
        session.close();
        System.out.println("successfully stored");
    }
}
```

### FetchTest.java

```
package com.javatpoint;
import java.util.*;
import org.hibernate.*;
import org.hibernate.cfg.*;
public class FetchTest {
```

```
public static void main(String[] args) {  
    Session session=new Configuration().configure().buildSessionFactory().openSession();  
  
    Query query=session.createQuery("from Question ");  
    List<Question> list=query.list();  
  
    Iterator<Question> iterator=list.iterator();  
    while(iterator.hasNext()){  
        Question question=iterator.next();  
        System.out.println("question id:"+question.getId());  
        System.out.println("question name:"+question.getName());  
        System.out.println("question posted by:"+question.getUsername());  
        System.out.println("answers.....");  
        Map<String,String> map=question.getAnswers();  
        Set<Map.Entry<String,String>> set=map.entrySet();  
  
        Iterator<Map.Entry<String,String>> iteratoranswer=set.iterator();  
        while(iteratoranswer.hasNext()){  
            Map.Entry<String,String> entry=(Map.Entry<String,String>)iteratoranswer.next();  
            System.out.println("answer name:"+entry.getKey());  
            System.out.println("answer posted by:"+entry.getValue());  
        }  
    }  
    session.close();  
}
```

download this hibernate example(developed using MyEclipse IDE)

[← prev](#)[next →](#)

Share  0