Hibernate – III

Collections Mapping

Table of Contents

Introduction

- Hibernate permits collection mapping as entity type.
- Hibernate mapping element used for mapping a collection depends upon the type of the interface. For example, at <**set**> element is used for mapping properties of type Set.
- Apart from <**set**>, there is also <**list**>, <**map**>, <**bag**>,<**SortedSet**> and <**SortedMap**>
  - ✓ java.util.**List** − java.util.ArrayList is used to store value. Preserves the position with an index column
  - ✓ **Bag** semantics − java.util.ArrayList is used to store value however the position is not preserved.
  - ✓ java.util.**Set** − java.util.HashSet is used to store value.
  - ✓ java.util.**SortedSet** − java.util.TreeSet is used to store value.
  - ✓ java.util.**Map** − java.util.HashMap is used to store value.
  - ✓ java.util.**SortedMap** − java.util.TreeMap is used to store value.

Note- Collections from one/many-to-many associations between types so there can be:

1. Value Type Collections
2. Embeddable Type Collections
3. Entity Collections

   Here we are discussing only Value Type Collections, other will discuss in Association Mapping.

Java.util.List Mapping

```
package com.Biditvats.domain;

import java.util.List;

public class Customer {
	private Long id;
	private String firstName;
	private String lastName;
	private String email;
	private List<Long> mobiles;

	public Customer() {
		//Do Nothing
	}

	//Getters and Setters
}
```

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

<!DOCTYPE hibernate-mapping PUBLIC
  "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
  "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
```

```
<hibernate-mapping>
 <class name="com.Biditvats.domain.Customer" table="CUSTOMER_MASTER3">
        <id name="id" column="customer_id">
            <generator class="identity" />
        </id>

 <property name="firstName" column="FIRST_NAME" />
 <property name="lastName" column="LAST_NAME" />
 <property name="email" column="EMAIL" />

 <list name="mobiles" table="CUSTOMER_MOBILES" cascade="all">
        <key column="customer_id" />
        <index column="MOBILE_iNDEX" type="int" />
        <element column="MOBILE" type="long" />
 </list>

 </class>
</hibernate-mapping>


package com.Biditvats.domain;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="CUSTOMER_TEMP")
public class Customer {

 @Id
 @GeneratedValue(strategy=GenerationType.IDENTITY)
 @Column(name="CUSTOMER_ID")
 private Long id;

 @Column(name="FIRST_NAME")
 private String firstName;

 @Column(name="LAST_NAME")
 private String lastName;

 @Column(name="EMAIL")
 private String email;

  @ElementCollection
 @Column(name="MOBILE")
 @OrderColumn(name="MOBILE_INDEX", nullable=false)
 private List<Long> mobiles;

 public Customer() {
   System.out.println("Customer object is created");
 }
```

```
	//Getters and Setters

	}
```

Bag Mapping

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

<!DOCTYPE hibernate-mapping PUBLIC
   "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
   "http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name="com.Biditvats.domain.Customer" table="CUSTOMER_MASTER4">
		<id name="id" column="customer_id">
			<generator class="identity" />
		</id>

	<property name="firstName" column="FIRST_NAME" />
	<property name="lastName" column="LAST_NAME" />
	<property name="email" column="EMAIL" />

	<bag name="mobiles" table="CUSTOMER_MOBILES2" cascade="all">
		<key column="customer_id" />
		<element column="MOBILE" type="long" />
	</bag>

</class>
</hibernate-mapping>
```

Java.util.Set Mapping

```
package com.Biditvats.domain;

import java.util.Set;

public class Product {
	private Long id;
	private String name;
	private String model;
	private String brand;
	private String category;
	private Double price;
	private Set<String> colors;

	public Product() {
		//Do Nothing
	}

	//Getters and Setters
}
```

```
<hibernate-mapping>
<class name="com.Biditvats.domain.Product" table="PRODUCT_MASTER2">
	<id name="id" column="product_id">
		<generator class="identity" />
	</id>

	<property name="name" column="NAME" />
	<property name="model" column="MODEL" />
	<property name="brand" column="BRAND" />
	<property name="category" column="CATEGORY" />
	<property name="price" column="PRICE" />

	<!-- Set of value Mapping -->
	<set name="colors" table="PRODUCT_COLORS2" cascade="all">
		<key column="product_id" /> <!-- Foreign Key -->
		<element column="COLOR" type="string" />

	</set>

</class>
</hibernate-mapping>
```

Java.util.SortedSet Mapping

A SortedSet that relies on the natural sorting order given by the child element Comparable implementation Logic must be annotated with the @SortNatural Hibernate annotation.

```
@Entity(name = "Person")
public static class Person {

	@Id
	private Long id;
	@OneToMany(cascade = CascadeType.ALL)
	@SortNatural
	private SortedSet<Phone> phones = new TreeSet<>();

	public Person() {
	}

	public Person(Long id) {
		this.id = id;
	}

	public Set<Phone> getPhones() {
		return phones;
	}
}


@Entity(name = "Phone" )
public static class Phone implements Comparable<Phone> {

	@Id
	private Long id;
```

```
        private String type;
        @NaturalId
        @Column(name= "number")
        private String number;

        public Phone() {
        }

        //Getters and Setters

        @Override
        public int compareTo(Phone o) {
                return number.compareTo( o.getNumber() );
        }

        @Override
        public boolean equals(Object o) {
                if( this == o ) {
                        return true;
                }
           if( 0 == null || getClass() != o.getClass() ) {
                        return false;
                }
                Phone phone = (Phone) o;
                return Objects.equals( number , phone.number );
        }

        @Override
        public int hashCode() {
                return Objects.hash( number );
        }
}
```

Java.util.Map Mapping

```
    package com.Biditvats.domain;

    import java.util.Map;

    public class Product {
        private Long id;
        private String name;
        private String model;
        private String brand;
        private String category;
        private Double price;
        private Map<String , String> props;

        public Product() {
                //Do Nothing
        }

        //Getters and Setters
    }
```

```
<hibernate-mapping>
<class name="com.Biditvats.domain.Product" table="PRODUCT_MASTER3">
	<id name="id" column="product_id">
		<generator class="identity" />
	</id>

	<property name="name" column="NAME" />
	<property name="model" column="MODEL" />
	<property name="brand" column="BRAND" />
	<property name="category" column="CATEGORY" />
	<property name="price" column="PRICE" />

	<!-- Map is indexed collection, so we required map-key with type attribute  -->
	<map name="props" table="PRODUCT_PROPS3" cascade="all">
		<key column="product_id" /> <!-- key defines the foreign Key -->
		<map-key column="PROP_KEY" type="string" />
		<element column="PROP_VALUE" type="string" /> <!-- type is required here -->
	</map>

</class>
</hibernate-mapping>
```