

Hibernate-Working with Collection-part 2 [adding primary key

This post is in continuation to our previous post where we have added a List of address. Now we wish to define a Primary key so that we can put an index on the table generated by Hibernate. so for that the data type of the Address should be the one that supports indexes.

ArrayList is a good option as it supports indexes,so we will change our address from Set to Collection. We will be using the interface Collection and not the ArrayList while defining the listOfAddresses as `private Collection<Address> listOfAddresses = new ArrayList<Address>();`

next generate the getters and setters.

Next is we need to define the primary key configuration and the way to do that is by using the `@CollectionId` annotation.

`@CollectionId` is Hibernate Specific and its not from JPA.

Below are the attributes of `@CollectionId`

```
@CollectionId(columns = { @Column }, generator = "", type = @Type)
```

Here:

`@Column` is the will be used for then name of primary key column.

`generator` would be the squence generator strategy and

`@Type` will be the data type of the column.

For generator we will have to add `@GenericGenerator`

```
@GenericGenerator(name = "sequence-gen", strategy = "sequence")
```

"sequence-gen" is the one of the generator that hibernate provide with strategy as "sequence"

So will all the annotation it will look like below

```
@ElementCollection
```

```
@JoinTable(name="USER_ADDRESS",
```

```
joinColumns=@JoinColumn(name="USER_ID")
```

```
)
```

```
@GenericGenerator(name="sequence-gen",strategy="sequence")
```

```
@CollectionId(columns = { @Column(name="ADDRESS_ID") }, generator = "sequence-gen", type = @Type(type="long"))
```

```
private Collection<Address> listOfAddresses = new ArrayList<Address>();
```

UserDetails.java

```
package com.technicalstack.dto;
```

```
import java.util.ArrayList;
```

```
import java.util.Collection;
```

```
import java.util.HashSet;
```

```
import java.util.Set;
```

```
import javax.persistence.Column;
```

```
import javax.persistence.ElementCollection;
```

```
import javax.persistence.Embedded;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.JoinColumn;
```

```
import javax.persistence.JoinTable;
import javax.persistence.Table;

import org.hibernate.annotations.CollectionId;
import org.hibernate.annotations.GenericGenerator;
import org.hibernate.annotations.Type;

@Entity
@Table (name="USER_DETAILS")
public class UserDetails {

    @Id @GeneratedValue
    private int userId;

    @Column (name="USER_NAME")
    private String name;

    /*@Embedded
    private Address address;

    public Address getAddress() {
        return address;
    }
    public void setAddress(Address address) {
        this.address = address;
    }*/

    @ElementCollection
    @JoinTable(name="USER_ADDRESS",
        joinColumns=@JoinColumn(name="USER_ID")
    )
    @GenericGenerator(name="sequence-gen",strategy="sequence")
    @CollectionId(columns = { @Column(name="ADDRESS_ID") }, generator = "sequence-gen", type = @Type(type="long"))
    private Collection<Address> listOfAddresses = new ArrayList<Address>();

    public Collection<Address> getListOfAddresses() {
        return listOfAddresses;
    }
    public void setListOfAddresses(Collection<Address> listOfAddresses) {
        this.listOfAddresses = listOfAddresses;
    }
    public int getUserId() {
        return userId;
    }
    public void setUserId(int userId) {
        this.userId = userId;
    }
}
```

```
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
  
}
```

HibernateTest.java

```
package com.technicalstack.dto;  
  
import javax.persistence.GeneratedValue;  
import javax.persistence.Id;  
  
import org.hibernate.Session;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class HibernateTest {  
  
    public static void main(String[] args) {  
        UserDetails user1 = new UserDetails();  
        // user1.setUserId(1); as we are using @Id @GeneratedValue so we don't need to set it.  
        user1.setName("Shailesh");  
  
        Address add1 = new Address();  
        add1.setCity("City1");  
        add1.setState("Maharashtra");  
        add1.setPincode("440011");  
        add1.setStreet("Street1");  
        user1.getListOfAddresses().add(add1);  
  
        Address add2 = new Address();  
        add2.setCity("City2");  
        add2.setState("Maharashtra");  
        add2.setPincode("440025");  
        add2.setStreet("Stree2");  
        user1.getListOfAddresses().add(add2);  
  
        SessionFactory sessionFactory = new Configuration().configure().buildSessionFactory();  
        Session session = sessionFactory.openSession();  
        session.beginTransaction();  
  
        session.save(user1);  
  
        session.getTransaction().commit();  
        session.close();  
    }  
}
```

```
}  
  
}
```

Now when we run the code we will get below output.

```
Hibernate: alter table USER_ADDRESS drop constraint FK3ndjyd9yl6efpjb7jx0voo2yk  
Hibernate: drop table if exists USER_ADDRESS cascade  
Hibernate: drop table if exists USER_DETAILS cascade  
Hibernate: drop sequence hibernate_sequence  
Hibernate: create sequence hibernate_sequence start 1 increment 1  
Hibernate: create table USER_ADDRESS (USER_ID int4 not null, USER_CITY varchar(255), USER_PINCODE varchar(255),  
USER_STATE varchar(255), USER_STREET varchar(255), ADDRESS_ID int8 not null, primary key (ADDRESS_ID))  
Hibernate: create table USER_DETAILS (userId int4 not null, USER_NAME varchar(255), primary key (userId))  
Hibernate: alter table USER_ADDRESS add constraint FK3ndjyd9yl6efpjb7jx0voo2yk foreign key (USER_ID) references  
USER_DETAILS  
Sep 26, 2016 6:43:45 AM org.hibernate.tool.hbm2ddl.SchemaExport execute  
INFO: HHH000230: Schema export complete  
Hibernate: select nextval ('hibernate_sequence')  
Hibernate: insert into USER_DETAILS (USER_NAME, userId) values (?, ?)  
Hibernate: select nextval ('hibernate_sequence')  
Hibernate: select nextval ('hibernate_sequence')  
Hibernate: insert into USER_ADDRESS (USER_ID, ADDRESS_ID, USER_CITY, USER_PINCODE, USER_STATE,  
USER_STREET) values (?, ?, ?, ?, ?, ?)  
Hibernate: insert into USER_ADDRESS (USER_ID, ADDRESS_ID, USER_CITY, USER_PINCODE, USER_STATE,  
USER_STREET) values (?, ?, ?, ?, ?, ?)  
Notice line 6 in above image
```

```
Hibernate: create table USER_ADDRESS (USER_ID int4 not null, USER_CITY varchar(255), USER_PINCODE varchar(255),  
USER_STATE varchar(255), USER_STREET varchar(255), ADDRESS_ID int8 not null, primary key (ADDRESS_ID))
```

This will insert a primary key column in Database

Output pane						
	Data	Output	Explain	Messages	History	
	user_id	user_city	user_pincode	user_state	user_street	address_id
	integer	character varying(255)	character varying(255)	character varying(255)	character varying(255)	bigint
1	1	City1	440011	Maharashtra	Street1	2
2	1	City2	440025	Maharashtra	Stree2	3