

## Hibernate-Working with Collection-part 1 [@ElementCollection, @JoinTable, @JoinColumn]

Now Suppose we have a requirement that we need to save a collection.

For example save all the address all the addresses where a user has lived,so this will have a collection of addresses.

Now in our example we will remove Address as attribute and add Set<Address> listOfAddresses to hold collection and remember to initialize it with HashSet() as below in our UserDetails.java.

@ElementCollection

To use collection in our class we have to annotate the attribute with @ElementCollection  
package com.technicalstack.dto;

```
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.Table;
```

@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

```
public Set<Address> listOfAddresses = new HashSet();
```

```
public Set<Address> getListOfAddresses() {
    return listOfAddresses;
}
```

```
public void setListOfAddresses(Set<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;
}
}
```

Next we have to add the details in our HibernateTest.java for UserDetails user1.

```
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("Pune");
        add1.setState("Maharashtra");
        add1.setPincode("440011");
        add1.setStreet("MG-Road");
        user1.getListOfAddresses().add(add1);
```

```
        Address add2 = new Address();
        add2.setCity("Mumbai");
        add2.setState("Maharashtra");
        add2.setPincode("440025");
        add2.setStreet("Meera-Road");
        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 run the application and observe the output.

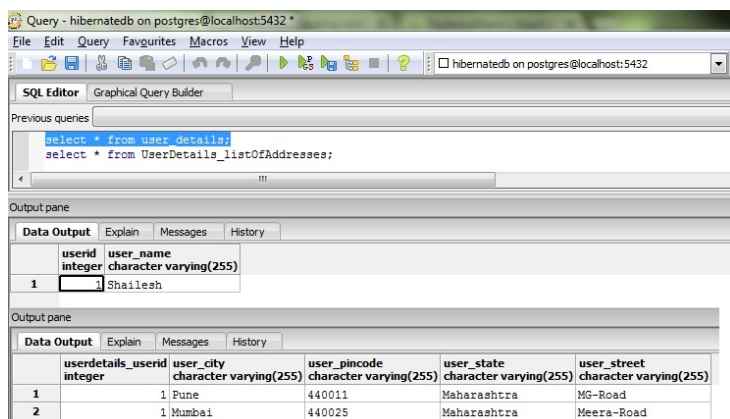
Output:

```
Hibernate: drop table if exists UserDetails_listOfAddresses cascade
Hibernate: drop sequence hibernate_sequence
Hibernate: create sequence hibernate_sequence start 1 increment 1
Hibernate: create table USER_DETAILS (userId int4 not null, USER_NAME varchar(255), primary key (userId))
Hibernate: create table UserDetails_listOfAddresses (UserDetails_userId int4 not null, USER_CITY varchar(255),
USER_PINCODE varchar(255), USER_STATE varchar(255), USER_STREET varchar(255))
Hibernate: alter table UserDetails_listOfAddresses add constraint FK6wx73ocb6j0mymo1jkc0o3wm foreign key
(UserDetails_userId) references USER_DETAILS
Sep 25, 2016 8:23:00 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: insert into UserDetails_listOfAddresses (UserDetails_userId, USER_CITY, USER_PINCODE, USER_STATE,
USER_STREET) values (?, ?, ?, ?, ?)
Hibernate: insert into UserDetails_listOfAddresses (UserDetails_userId, USER_CITY, USER_PINCODE, USER_STATE,
USER_STREET) values (?, ?, ?, ?, ?)
```

Explanation:

At line 4 and 10, notice first hibernate has inserted in the table USER\_DETAILS and then it created a new table UserDetails\_listOfAddresses and has inserted UserDetails\_userId as foreign key and has inserted the Address details in this table.

Entries in DataBase:



Next we will see use of annotation @JoinTable:

#### @JoinTable

Here we have seen that Hibernate has created the default table UserDetails\_listOfAddresses which is not very friendly so now we will how to customize the name of the table created by Hibernate for the Addresses  
We will have to use another annotation @JoinTable(name="USER\_ADDRESS")

#### @ElementCollection

```
@JoinTable(name="USER_ADDRESS")
public Set<Address> listOfAddresses = new HashSet();
```

Now when you rerun the application you will find hibernate has created the USER\_ADDRESS table and has inserted the records in it.

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 (UserDetails\_userId int4 not null, USER\_CITY varchar(255), USER\_PINCODE varchar(255), USER\_STATE varchar(255), USER\_STREET varchar(255))

Hibernate: create table USER\_DETAILS (userId int4 not null, USER\_NAME varchar(255), primary key (userId))

Hibernate: alter table USER\_ADDRESS add constraint FK30p76h3gg2si436waw9bt9e9n foreign key (UserDetails\_userId) references USER\_DETAILS

Sep 25, 2016 9:21:40 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: insert into USER\_ADDRESS (UserDetails\_userId, USER\_CITY, USER\_PINCODE, USER\_STATE, USER\_STREET) values (?, ?, ?, ?, ?)

Hibernate: insert into USER\_ADDRESS (UserDetails\_userId, USER\_CITY, USER\_PINCODE, USER\_STATE, USER\_STREET) values (?, ?, ?, ?, ?)

#### @JoinColumn

here if you notice even in the generated new table the primary key the default UserDetails\_userId which is again needs to be customised. So lets see how to do that.

#### @ElementCollection

```
@JoinTable(name="USER_ADDRESS",joinColumns=@JoinColumn(name="USER_ID"))
)
public Set<Address> listOfAddresses = new HashSet();
```

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))

Hibernate: create table USER\_DETAILS (userId int4 not null, USER\_NAME varchar(255), primary key (userId))

Hibernate: alter table USER\_ADDRESS add constraint FK3ndjyd9yl6efpb7jx0voo2yk foreign key (USER\_ID) references USER\_DETAILS

Sep 25, 2016 9:56:58 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: insert into USER\_ADDRESS (USER\_ID, USER\_CITY, USER\_PINCODE, USER\_STATE, USER\_STREET) values (?, ?, ?, ?)

Hibernate: insert into USER\_ADDRESS (USER\_ID, USER\_CITY, USER\_PINCODE, USER\_STATE, USER\_STREET) values (?, ?, ?, ?)