Cascade – JPA & Hibernate annotation common mistake

By mkyong (https://www.mkyong.com/author/mkyong/) | February 6, 2010 | Updated : August 30, 2012 | Viewed : 269,378 times +569 pv/w

Often times, developers are mix used the JPA and Hibernate annotation together, it will caused a very common mistake – JPA cascade type annotation is not working in Hibernate?

During the code review section, i find out many Java developers are not aware of this mistake and causing the program failed to execute the cascade operation to the related entities. I will take this one-to-many hibernate example (http://www.mkyong.com/hibernate/hibernate-one-to-many-relationship-example/) for the demonstration.

Mistake

In the one-to-many example, many developers declared the JPA cascade options as following :

```
...
@Entity
@Table(name = "stock", catalog = "mkyong", uniqueConstraints = {
        @UniqueConstraint(columnNames = "STOCK_NAME"),
        @UniqueConstraint(columnNames = "STOCK_CODE") })
public class Stock implements java.io.Serializable {
    ...
    private Set<StockDailyRecord> stockDailyRecords =
                                          new HashSet<StockDailyRecord>(0);

    @OneToMany(fetch = FetchType.LAZY,
       cascade = {CascadeType.PERSIST,CascadeType.MERGE },
       mappedBy = "stock")
    public Set<StockDailyRecord> getStockDailyRecords() {
    return this.stockDailyRecords;
    }

    public void setStockDailyRecords(Set<StockDailyRecord> stockDailyRecords) {
    this.stockDailyRecords = stockDailyRecords;
    }
    ...
```

Save it with Hibernate session.

```
stockDailyRecords.setStock(stock);
stock.getStockDailyRecords().add(stockDailyRecords);

session.save(stock);
session.getTransaction().commit();
```

What the code trying to do is when you save a “stock”, it will save the associated stockDailyRecords as well. Everything look fine, but **THIS IS NOT WORKING**, the cascade options will not execute and save the stockDailyRecords. Can you spot the mistake?

Explanation

Look in the code, **@OneToMany** is from JPA , it expected a JPA cascade – **javax.persistence.CascadeType**. However when you save it with Hibernate session, **org.hibernate.engine.Cascade** will do the following checking …

```
if ( style.doCascade( action ) ) {
    cascadeProperty(
          persister.getPropertyValue( parent, i, entityMode ),
      types[i],
                  style,
          anything,
          false
    );
}
```

The Hibernate save process will causing a **ACTION_SAVE_UPDATE** action, but the JPA will pass a **ACTION_PERSIST** and **ACTION_MERGE**, it will not match and causing the cascade failed to execute.

@see source code

- **org.hibernate.engine.Cascade**
- **org.hibernate.engine.CascadeStyle**
- **org.hibernate.engine.CascadingAction**

Solution

Delete the JPA cascade – **javax.persistence.CascadeType**, replace it with Hibernate cascade – **org.hibernate.annotations.Cascade**, with **CascadeType.SAVE_UPDATE**.

```
...
@Entity
@Table(name = "stock", catalog = "mkyong", uniqueConstraints = {
        @UniqueConstraint(columnNames = "STOCK_NAME"),
        @UniqueConstraint(columnNames = "STOCK_CODE") })
public class Stock implements java.io.Serializable {
    ...
    private Set<StockDailyRecord> stockDailyRecords =
                                              new HashSet<StockDailyRecord>(0);

    @OneToMany(fetch = FetchType.LAZY, mappedBy = "stock")
    @Cascade({CascadeType.SAVE_UPDATE})
    public Set<StockDailyRecord> getStockDailyRecords() {
    return this.stockDailyRecords;
    }

    public void setStockDailyRecords(Set<StockDailyRecord> stockDailyRecords) {
    this.stockDailyRecords = stockDailyRecords;
    }
    ...
```

Now, it work as what you expected.

Conclusion

It look like an incompatible issue between JPA and Hibernate cascade annotation, if Hibernate is a JPA implementation, what may causing the misunderstand in between?

Tags : cascade (https://www.mkyong.com/tag/cascade/) hibernate (https://www.mkyong.com/tag/hibernate/) jpa (https://www.mkyong.com/tag/jpa/)

Share this article on

Twitter (https://twitter.com/intent/tweet?text=Cascade - JPA & Hibernate annotation common mistake&url=https://www.mkyong.com/hibernate/cascade-jpa-hibernate-annotation-common-mistake/&via=mkyong) Facebook (https://www.facebook.com/sharer/sharer.php?u=https://www.mkyong.com/hibernate/cascade-jpa-hibernate-annotation-common-mistake/) Google+ (https://plus.google.com/share?url=https://www.mkyong.com/hibernate/cascade-jpa-hibernate-annotation-common-mistake/)

About the Author

mkyong

Founder of Mkyong.com (http://mkyong.com), love Java and open source stuff. Follow him on Twitter (https://twitter.com/mkyong), or befriend him on Facebook (http://www.facebook.com/java.tutorial) or Google Plus