


Home

PUBLIC

 Stack Overflow

Tags

Users

Jobs

TEAMS

+ Create Team

who is owner in association with jpa and hibernate?

Ask Question


I am confused in owner in relationship with hibernate.

what exactly is owner(owner side) in association?

I want to study Mappedby and inverse. please help.

[java](#) [hibernate](#) [jpa](#)

asked Jul 31 '14 at 8:07



[user3543851](#)
66 6

mkyong.com is a good start – [Yehia Awad](#) Jul 31 '14 at 8:11

@яша Stop suggesting random site. Link official docs if you have. – [SURESH ATTA](#) Jul 31 '14 at 8:13

2 Here you go :docs.oracle.com/javaee/5/tutorial/doc/bnbqa.html#bnbqi – [SURESH ATTA](#) Jul 31 '14 at 8:15

1 @SURESHATTA whether it is official or not the main thing is to understand the concept! who cares if the official docs is offering 1000's of pages of philosophic words while what I need are 2-3 pages to help me solve my problem ? – [Yehia Awad](#) Jul 31 '14 at 8:30

@SURESHATTA then why people still trust sites like stackoverflow ?! you believe people in the official documentation do not make mistakes ? – [Yehia Awad](#) Jul 31 '14 at 8:57

the relation to be persisted.

If you are mapping the entities to a relational database (most probably the case) the owning side often can be identified as the entity whose table contains the foreign key.

In the entities themselves `mappedBy` would refer to the owning side and thus is placed on the inverse side of the relation.

In 1:n relations in most cases the owning side is the n side, in n:m relations, 1:1 relations or 1:n with mapping tables you can choose either side, just choose one.

Example:

```
class Thread {
    @OneToMany( mappedBy = "thread" )
    List<Entry> entries;
}

class Entry {
    @ManyToOne
    Thread thread;
}
```

In the example, the owning side would be the `Entry` entity, since you need to change the value of `Entry#thread` to change the thread an entry belongs to. Just adding/removing an entry to/from `Thread#entries` wouldn't make the changes persisted in most cases (`orphanRemoval` and the like would still have an effect if done correctly).

edited Jul 31 '14 at 8:25

answered Jul 31 '14 at 8:19



Thomas

66k 8 86 120