

Hibernate Second Level Cache Parent Ve Collection İşlemleri

ehcache.xml Ayarları

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ehcache xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3       xsi:noNamespaceSchemaLocation="ehcache.xsd"
4       updateCheck="true" monitoring="autodetect"
5       dynamicConfig="true">
6   <diskStore path="java.io.tmpdir"/>
7   <defaultCache
8       maxElementsInMemory="1"
9       eternal="false"
10      timeToIdleSeconds="120"
11      timeToLiveSeconds="120"
12      overflowToDisk="true"
13      />
14
15   <cache name="mucayufa.hibernate.reverseEngineering.Admin"
16         maxElementsInMemory="500"
17         eternal="true"
18         timeToIdleSeconds="100"
19         timeToLiveSeconds="100"
20         overflowToDisk="false"
21         />
22
23   <cache
24         name="mucayufa.hibernate.reverseEngineering.Admin.articles"
25         maxElementsInMemory="450"
26         eternal="false"
27         timeToLiveSeconds="600"/>
28
29   <cache name="mucayufa.hibernate.reverseEngineering.Article"
30         maxElementsInMemory="500"
31         eternal="false"
32         timeToIdleSeconds="100"
33         timeToLiveSeconds="100"
34         overflowToDisk="false"
35         />
36 </ehcache>
```

Not: Ehcache ile ilgili detaylı bilgi için [tıklayınız](http://www.softwarevol.com/en/tutorial/Hibernate-Ehcache-Usage) (<http://www.softwarevol.com/en/tutorial/Hibernate-Ehcache-Usage>)

Collection'u da cache'e almak için Admin.articles tarzında kullanmak gereklidir. Burada articles Admin entity içerisindeki collection değişkenidir.

Entity Sınıfı

```
1 @Entity
2 @Table(name = "admin"
3       , catalog = "softwarevol"
4       )
5 @org.hibernate.annotations.Cache(usage =CacheConcurrencyStrategy.NONSTRICT_READ_WRITE)
6 public class Admin implements java.io.Serializable {
7     @OneToMany(cascade = CascadeType.ALL, fetch = FetchType.LAZY, mappedBy = "admin")
8     @org.hibernate.annotations.Cache(usage =CacheConcurrencyStrategy.READ_WRITE)
9     public List<Article> getArticles() {
10         return this.articles;
11     }
12 }
```

Test İşlemleri

```
1 private void useSecondLevelCache() {
2     Session session1 = HibernateUtil.getSessionFactory().getCurrentSession();
3     session1.beginTransaction();
4     Admin admin=(Admin)session1.load(Admin.class,(short)1);
5     System.out.println(admin.getArticles().get(0).getTitle());
6
7     session1.getTransaction().commit();
8     Session session2 = HibernateUtil.getSessionFactory().getCurrentSession();
9     session2.beginTransaction();
10    admin=(Admin)session2.load(Admin.class,(short)1);
11
12    admin.setName("denem2e");
13    session2.update(admin);
14
15    System.out.println(admin.getArticles().get(0).getTitle());
```

```
16     session1.getTransaction().commit();
17
18     Session session3 = HibernateUtil.getSessionFactory().getCurrentSession();
19     session3.beginTransaction();
20
21     admin=(Admin)session3.load(Admin.class, (short)1);
22     System.out.println(admin.getArticles().get(0).getTitle());
23     session3.getTransaction().commit();
24 }
```

İlk başta(**session1**'de) admin nesnesi veritabanından çekilir. Admin nesnesi ile birlikte Article nesnelere de veritabanından çekileceği için veritabanı sorgusu gerçekleşir. **session2** de ise admin ve ona bağlı article nesnelere, hibernate **second-level cache**'den getirir. Ayrıca bu session süresinde, admin nesnesi update olduğu için, hibernate tekrar veritabanından admin nesnesini çeker ve **second-level cache**'teki admin nesnesini günceller. Güncelleme esnasında article nesnelere tekrar veritabanından çekilmez; çünkü sadece admin nesnesi güncellenmiştir. **session3**'de hibernate, **second-level cache**'den verileri çeker.

Yukarıdaki örnekte parent entity güncelleniyor. Eğer collection güncelleme(yeni entity ekleme, silme, güncelleme) yapılmış olsa idi şu makaledeki gibi işlem yapmak gerekecekti: http://planet.jboss.org/post/collection_caching_in_the_hibernate_second_level_cache (http://planet.jboss.org/post/collection_caching_in_the_hibernate_second_level_cache)