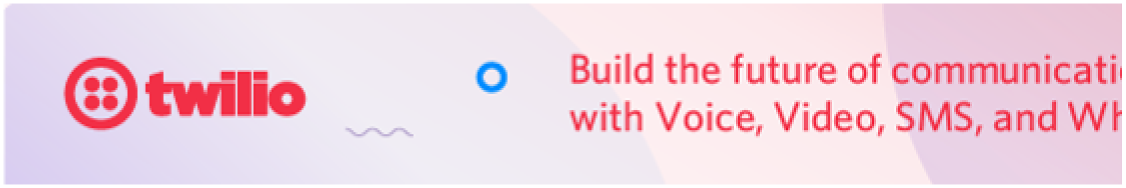
twilio
Build the future of communicati
with Voice, Video, SMS, and Wh

Clear Hibernate 2nd level cache after manually DB update

Ask Question

Shortly, I have an entity mapped to view in DB (Oracle) with enabled 2nd level Cache (read only strategy) -- ehcache.

If I manually update some column in DB -- cache will not be updated.

I did not find any ways to do this. Only if updates will be done through Hibernate entity.

May I somehow implement this feature?

Maybe Job to monitor table (or view)? Or maybe there is some method to notify Hibernate about change in DB in concrete table.

Thanks for future answers!

java oracle hibernate caching ehcache

asked Oct 28 '16 at 9:28

Artem Vereschaka
25 1 8

is no automagic version of such functionality - either don't change DB manually or do an function in your app to refresh cache on "administrator" (yours) request, you can also restart your app - it should force the cache to be reloaded. – Krzysztof Cichocki Oct 28 '16 at 9:52

3 Answers

According to Hibernate JavaDoc, you can use `org.hibernate.Cache.evictAllRegions()` :

> evictAllRegions() Evict all data from the cache.

Using Session and SessionFactory:

```
Session session = sessionFac

if (session != null) {
    session.clear(); // inte
}

Cache cache = sessionFactory

if (cache != null) {
    cache.evictAllRegions();
}
```

1) If you need update only one entity (if directly from db you will update only certain entities) not whole session, you can use

> evictEntityRegion(Class entityClass) Evicts all

you can use this method that evicts all entities from 2nd level cache (we can expose this method to admins through JMX or other admin tools):

```
/**
  * Evicts all second level c
  * needed when an external a
  */
public void evict2ndLevelCac
    try {
        Map<String, ClassMet
sessionFactory.getAllClassMe
        Cache cache = sessic
        for (String entityNa
            logger.info("Evi
            cache.evictEntit
        }
    } catch (Exception e) {
        logger.logp(Level.SE
evicting 2nd level hibernate
    }
}
```

3) here another apporche described here for postgresql+hibernate, I think similar you can do for Oracle like this

edited May 23 '17 at 11:53

Community ♦
1 1

answered Oct 28 '16 at 9:37

Daniyar
390 2 20

I do not need to clear all the cache. I need to update only cache related to one entity. – Artem Vereschaka Oct 28 '16 at 9:38

then you should use session.refresh(entity) as Vimal wrote – Daniyar Oct

Artem Vereschaka Oct 28 '16 at 9:41

look on this, I hope it will help you this is about postgresql but i think it also possible in oracle stackoverflow.com/questions/13258976/… – Daniyar Oct 28 '16 at 9:46

This might work with some scheduler. Thanks for links and such a detailed answer. – Artem Vereschaka Oct 28 '16 at 10:49

You can use `session.refresh()` method to reload the objects which are currently held in session.

Read object loading for more detail.

answered Oct 28 '16 at 9:34

Vimal Bera
8,359 3 17 40

Each time when I am trying to use my entity -- call refresh method? Is there any sense to use cache if I will call refresh each time on access? – Artem Vereschaka Oct 28 '16 at 9:36

You will find here the way to control the second level cache:

answered Oct 28 '16 at 9:38

mvera
604 6 19