Hibernate Session API

Presented by Hosein Zare
Twitter: @zare88r

Session

- این سرویس رابط بین برنامه ی جاوا وhibernate میباشد
- چرخه حیات یک session از شروع تا پایان یک تراکنش منطقی خلاصه میشود
- کاربردهای اصلی Session شامل ایجاد ، به روز رسانی ، حذف و خواندن اطلاعات میباشد

Session

- هر instance از هر entity برای session سه وضعیت میتواند داشته باشد:
  - Transient –
    - تا به حال ذخیره سازی نشده و به Session هم مرتبط نشده است
  - Persistent –
    - به یک Session مرتبط است
  - Detached –
    - قبلاً persistent بوده هم اکنون از Sessionجدا شده است.

Session

- متود های ذخیره سازی (persistent کردن وضعیت entity ها)
  - save() – (*INSERT*)
  - persist() – (*INSERT*)
  - saveOrUpdate() – (*UPDATE*)
  - merge() – (*UPDATE*)
- متود حذف (transient کردن وضعیت entity ها)
  - delete() – (*DELETE*)

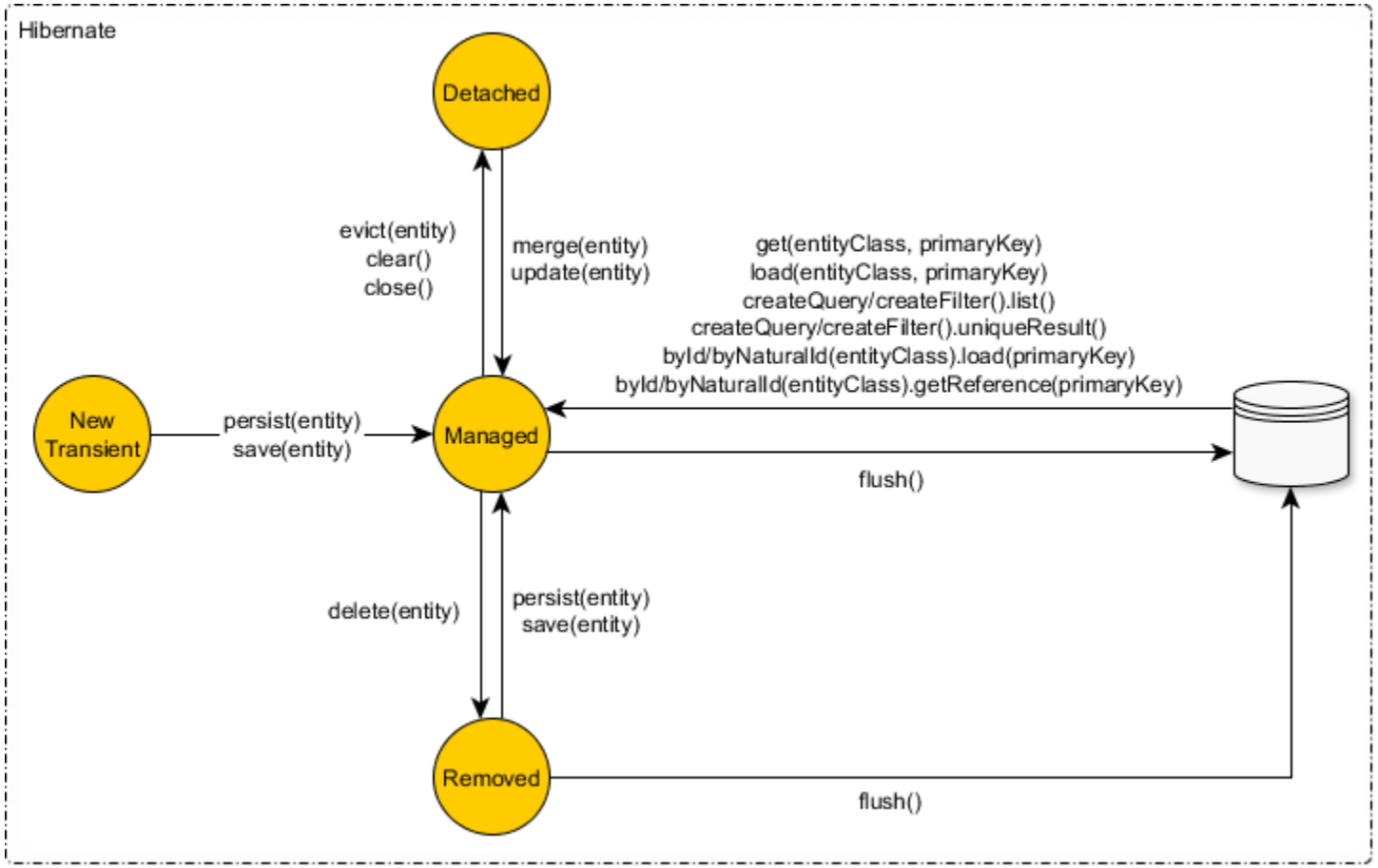
Session

- متود های واکشی اطلاعات با کلید ها (persistent کردن وضعیت entity ها)
  - find() – get()
  - load() –
  - byNaturalId().get() –

Session

- متود هایی که وضعیت detached را به persistent میبرند
  - update()
  - saveOrUpdate()
  - lock()
  - replicate()

Session


Hibernate
Detached
evict(entity)
clear()
close()
merge(entity)
update(entity)
get(entityClass, primaryKey)
load(entityClass, primaryKey)
createQuery/createFilter().list()
createQuery/createFilter().uniqueResult()
byId/byNaturalId(entityClass).load(primaryKey)
byId/byNaturalId(entityClass).getReference(primaryKey)
New Transient
persist(entity)
save(entity)
Managed
flush()
delete(entity)
persist(entity)
save(entity)
Removed
flush()

HQL

- هایبرنت از یک زبان قدرتمند شبیه به SQL جهت بازیابی اطلاعات استفاده میکند
- در این زبان به جای نام جداول نام entity ها و به جای نام ستون ها از فیلد ها استفاده میشود
- کلید واژه ها case sensitive نیستند ولی نام entity و فیلد ها اینطور باید دقیق ذکر شوند

HQL

- نمونه ها
  - `from Cat`
  - `from Cat as cat`
  - `from Cat cat`
  - `from Formula, Parameter`
  - `from Formula as form, Parameter as param`

HQL - Associations and joins

- نمونه ها

```
• from Cat as cat
    inner join cat.mate as mate
    left outer join cat.kittens as kitten
• from Cat as cat
    inner join fetch cat.mate
    left join fetch cat.kittens child
    left join fetch child.kittens
```

HQL - Select

- نمونه ها

```
• from Cat as cat
    inner join cat.mate as mate
    left outer join cat.kittens as kitten
• from Cat as cat
    inner join fetch cat.mate
    left join fetch cat.kittens child
    left join fetch child.kittens
```

HQL - Select

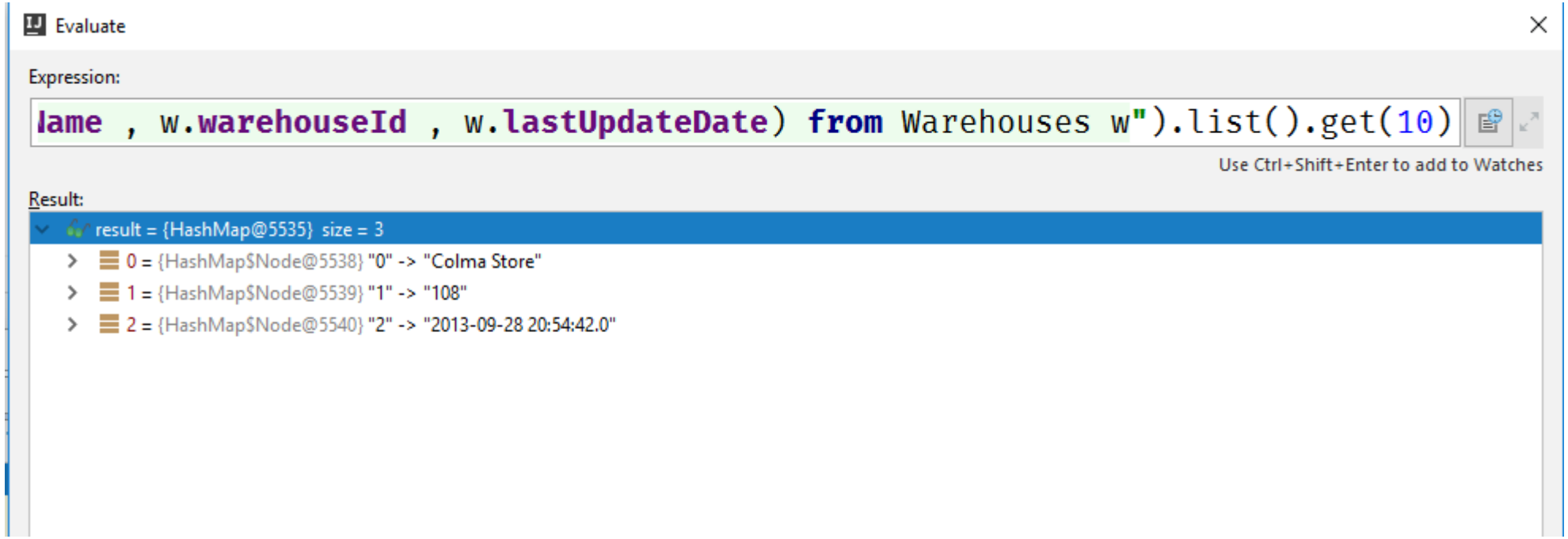
- نمونه ها

```
• select mate from Cat as cat inner join cat.mate as mate
• select cat.mate from Cat cat
• select cat.name from DomesticCat cat  where cat.name like 'fri%'
• select cust.name.firstName from Customer as cust
• select w.warehouseName , w.warehouseId from Warehouses w
• select new list(w.warehouseName , w.warehouseId) from Warehouses w
• select new Family(mother, mate, offspr)
      from DomesticCat as mother join mother.mate as mate left join
      mother.kittens as offspr
```

HQL - Select

- نمونه ها

- `select new map(w.warehouseName , w.warehouseId , w.lastUpdateDate) from Warehouses w`



HQL - Aggregate

- نمونه ها

```
• select avg(cat.weight), sum(cat.weight), max(cat.weight),
  count(cat) from Cat cat

• select cat.weight + sum(kitten.weight)
from Cat cat join cat.kittens kitten group by cat.id, cat.weight

• select distinct cat.name from Cat cat

• select count(distinct cat.name), count(cat) from Cat cat
```

HQL - where

- نمونه ها

```
select avg(cat.weight), sum(cat.weight), max(cat.weight),
count(cat) from Cat cat

select cat.weight + sum(kitten.weight)
from Cat cat join cat.kittens kitten group by cat.id, cat.weight

select distinct cat.name from Cat cat

select count(distinct cat.name), count(cat) from Cat cat

from Cat where name='Fritz'

select foo from Foo foo, Bar bar where foo.startDate = bar.date

from Cat cat where cat.mate.name is not null
```

HQL - where

- نمونه ها

```
from Foo foo where foo.bar.baz.customer.address.city is not null

select cat, mate from Cat cat, Cat mate where cat.mate = mate

from AuditLog log, Payment payment
 where log.item.class = 'Payment' and log.item.id = payment.id

from DomesticCat cat where cat.name in ( 'Foo', 'Bar', 'Baz' )

from DomesticCat cat where cat.name not between 'A' and 'B'

from Cat cat where cat.alive = true

from Cat cat where cat.kittens.size > 0

from Cat cat where cat.kittens.size > 0
from Cat cat where size(cat.kittens) > 0
```

HQL - order by

- نمونه ها

```
from DomesticCat cat
order by cat.name asc, cat.weight desc, cat.birthdate
```

HQL - group by

- نمونه ها

```
select foo.id, avg(name), max(name)
from Foo foo join foo.names name
group by foo.id
```

```
select cat
from Cat cat
    join cat.kittens kitten
group by cat.id, cat.name, cat.other, cat.properties
having avg(kitten.weight) > 100
order by count(kitten) asc, sum(kitten.weight) desc
```

HQL - Subqueries

- نمونه ها

```
from Cat as fatcat
where fatcat.weight > (select
avg(cat.weight) from DomesticCat cat)


from Cat as cat
where not exists (
    from Cat as mate where mate.mate = cat
)
```

Session Query

- نمونه ها

```
List<Warehouses> warehousesList=  session.createQuery("from Warehouses w" +
          " where lower(w.warehouseName) like concat('%',:paramName,'%') " )
          .setFirstResult(0)
          .setMaxResults(10)
          .setParameter("paramName","it")
          .list();
```

```
Long count = (Long) session.createQuery("select count(w) from Warehouses w" +
        " where lower(w.warehouseName) like concat('%',:paramName,'%') ")
        .setParameter("paramName", "it")
        .getSingleResult();
```

https://docs.jboss.org/hibernate/orm/3.3/reference/en/html/queryhql.html

Named Query

- نمونه ها

```
@NamedQueries(
        @NamedQuery(
                name = "get_person_by_name",
                query = "select p from Person p where name = :name"
                )
        )

Query query = entityManager.createNamedQuery( "get_person_by_name" );
```

Criteria

- با استفاده از org.hibernate.Criteria میتوانید یک query را با یک api تولید کنید
- Session در واقع یک factory برای تولید criteria میباشد

```
Criteria crit = sess.createCriteria(Cat.class);
crit.setMaxResults(50);
List cats = crit.list()
```

محدود کردن خروجی - Criteria

```
List cats = sess.createCriteria(Cat.class)
        .add( Restrictions.like("name", "Fritz%") )
        .add( Restrictions.between("weight", minWeight, maxWeight) )
        .list();
```

```
List cats = sess.createCriteria(Cat.class)
        .add( Restrictions.in( "name", new String[] { "Fritz", "Izi", "Pk" } ) )
        .add( Restrictions.disjunction()
                .add( Restrictions.isNull("age") )
                .add( Restrictions.eq("age", new Integer(0) ) )
                .add( Restrictions.eq("age", new Integer(1) ) )
                .add( Restrictions.eq("age", new Integer(2) ) )
        )
    ).list();
```

محدود کردن خروجی - Criteria

```
Property age = Property.forName("age");
List cats = sess.createCriteria(Cat.class)
        .add(Restrictions.disjunction()
        .add(age.isNull())
        .add(age.eq(new Integer(0)))
        .add(age.eq(new Integer(1)))
        .add(age.eq(new Integer(2)))
            )
        .add(Property.forName("name").in(new String[]{"Fritz","Izi","Pk"}))
        .list();
```

```
List cats = sess.createCriteria(Cat.class)
            .add(Restrictions.sqlRestriction("lower({alias}.name) like lower(?)", "Fritz%“
                                , Hibernate.STRING))
            .list();
```

Criteria – Order by

```
List cats = sess.createCriteria(Cat.class)
            .add(Restrictions.like("name", "F%")
                    .addOrder(Order.asc("name").nulls(NullPrecedence.LAST))
                    .addOrder(Order.desc("age"))
                    .setMaxResults(50)
                    .list();
```

```
List cats = sess.createCriteria(Cat.class)
            .add(Property.forName("name").like("F%"))
            .addOrder(Property.forName("name").asc())
            .addOrder(Property.forName("age").desc())
            .setMaxResults(50)
            .list();
```

Criteria – Associations

```
List cats = sess.createCriteria(Cat.class)
            .add( Restrictions.like("name", "F%") )
            .createCriteria("kittens")
            .add( Restrictions.like("name", "F%") )
            .list();
```

```
List cats = sess.createCriteria(Cat.class)
            .createAlias("kittens", "kt")
            .createAlias("mate", "mt")
            .add( Restrictions.eqProperty("kt.name", "mt.name") )
            .list();
```

Criteria – Associations

```
List cats = sess.createCriteria(Cat.class)
            .createCriteria("kittens", "kt")
            .add( Restrictions.eq("name", "F%") )
            .setResultTransformer(Criteria.ALIAS_TO_ENTITY_MAP)
            .list();
Iterator iter = cats.iterator();
while ( iter.hasNext() ) {
    Map map = (Map) iter.next();
    Cat cat = (Cat) map.get(Criteria.ROOT_ALIAS);
    Cat kitten = (Cat) map.get("kt");
}
```

```
List cats = session.createCriteria( Cat.class )
            .createAlias("mate", "mt", Criteria.LEFT_JOIN, Restrictions.like("mt.name", "good%") )
            .addOrder(Order.asc("mt.age"))
            .list();
```

Criteria – Associations

```
List cats = sess.createCriteria(Cat.class)
            .add( Restrictions.like("name", "Fritz%") )
            .setFetchMode("mate", FetchMode.EAGER)
            .setFetchMode("kittens", FetchMode.EAGER)
            .list();
```

Criteria – Example queries

```
Cat cat = new Cat();
cat.setName('kittyCat');
cat.setColor(Color.BLACK);

List results = session.createCriteria(Cat.class)
        .add( Example.create(cat) )
        .list();
```

Criteria – Projections, aggregation and grouping

```
List results = session.createCriteria(Cat.class)
            .setProjection(Projections.rowCount())
            .add(Restrictions.eq("color", Color.BLACK))
            .list();


List results = session.createCriteria(Cat.class)
                .setProjection(Projections.projectionList()
                        .add(Projections.rowCount())
                        .add(Projections.avg("weight"))
                        .add(Projections.max("weight"))
                        .add(Projections.groupProperty("color"))
                )
.list();
```

Criteria – Detached queries

```
DetachedCriteria avgWeightForName = DetachedCriteria.forClass(Cat.class, "cat2")
        .setProjection( Property.forName("weight").avg() )
        .add( Property.forName("cat2.name").eqProperty("cat.name") );
session.createCriteria(Cat.class, "cat")
        .add( Property.forName("weight").gt(avgWeightForName) )
        .list();
```