



[Home](#) » [Hibernate](#) » Sự khác nhau giữa load() và get() trong Hibernate

Sự khác nhau giữa load() và get() trong Hibernate

Posted on Tháng Mười Một 21, 2017

Sự khác nhau giữa load() và get() trong Hibernate

1. Method load()

- Chỉ sử dụng method load() khi bạn chắc chắn rằng đối tượng tồn tại trong database
- Method load() sẽ ném ra 1 exception nếu đối tượng không tìm thấy trong database
- Method load() chỉ trả về 1 đối tượng giả (proxy object) nó chỉ lấy dữ liệu từ database ra khi cần tới.

* Proxy Object là 1 đối tượng giả, nó chỉ có id, các thuộc tính khác không được khởi tạo, ví dụ khi bạn để FETCH_TYPE = LAZY khi mapping thì nó cũng chỉ trả về 1 proxy object.

2. Method get()

- Nếu bạn không chắc chắn rằng đối tượng có tồn tại trong database không thì hãy dùng get()
- Method get() sẽ trả về null nếu không tìm thấy đối tượng trong database
- Method get() sẽ truy xuất vào database ngay lập tức để lấy đối tượng thực đang tồn tại.

Ví dụ 1:

Thực hiện get() đối tượng role có id = 1 và load() đối tượng user có id = 1000 (đối tượng này ko tồn tại trong database)

```
SessionFactory factory = HibernateUtils.getSessionFactory();
Session session = factory.openSession();
Users user = (Users) session.load(Users.class, 1000);
Role role = (Role) session.get(Role.class, 1);
```

Kết quả: chỉ có đối tượng role là được truy xuất từ database, còn đối tượng user ko được truy xuất

```
Hibernate: select role0_.id as id1_6_0_, role0_.delete_flag as delete_f2_6_0_, role0_.nam
```



FAQ, HIBERNATE

Ví dụ 2:

Thực hiện load() đối tượng user có id = 1000 và in ra email của user này (đối tượng này không tồn tại trong database)

```
SessionFactory factory = HibernateUtils.getSessionFactory();
Session session = factory.openSession();
Users user = (Users) session.load(Users.class, 1000);
System.out.println(user.getEmail());
```

Kết quả: Khi lấy email thì nó sẽ truy xuất tới database, nhưng vì đối tượng không tồn tại nên xảy ra exception *org.hibernate.ObjectNotFoundException*

```
Hibernate: select users0_.id as id1_10_0_, users0_.create_date as create_d2_10_0_, users0_
Exception in thread "main" org.hibernate.ObjectNotFoundException: No row with the given i
    at org.hibernate.internal.SessionFactoryImpl$1$1.handleEntityNotFound(SessionFactoryImp
    at org.hibernate.proxy.AbstractLazyInitializer.checkTargetState(AbstractLazyInitializer
    at org.hibernate.proxy.AbstractLazyInitializer.initialize(AbstractLazyInitializer.java:
    at org.hibernate.proxy.AbstractLazyInitializer.getImplementation(AbstractLazyInitializ
    at org.hibernate.proxy.pojo.javassist.JavassistLazyInitializer.invoke(JavassistLazyInit
    at com.mysql.entities.Users_$$_jvst75a_9.getEmail(Users_$$_jvst75a_9.java)
    at com.mysql.dao.Demo.main(Demo.java:171)
```

Ví dụ 3:

Thực hiện load() đối tượng user có id = 1 (Có tồn tại trong database) và in ra email của nó;

get() đối tượng role có id = 1000 (không tồn tại trong database) và in ra role name của nó:

```
SessionFactory factory = HibernateUtils.getSessionFactory();
Session session = factory.openSession();
Users user = (Users) session.load(Users.class, 1);
System.out.println(user.getEmail());

Role role = (Role) session.get(Role.class, 1000);
if (role == null) {
    System.out.println("role with id = 1000 is null");
} else {
    System.out.println(role.getName());
}
```

Kết quả:

- Đối tượng user tồn tại nên nó sẽ truy xuất và in ra email
- Đối tượng role không tồn tại nên trả về null

```
Hibernate: select users0_.id as id1_10_0_, users0_.create_date as create_d2_10_0_, users0_
```

Eclipse sau khi xóa

AOP là gì? Aspect Oriented Programming trong Java

Dependency Injection (DI) là gì? Code ví dụ bằng Java

Cấu hình giới hạn kích thước file upload cho Tomcat

Phân biệt save, persist, update, merge, saveOrUpdate trong hibernate

Sự khác nhau giữa merge với saveOrUpdate trong Hibernate

Hibernate Batch Processing là gì? Batch Processing trong Hibernate

Sự khác nhau giữa load() và get() trong Hibernate

Code ví dụ hibernate annotation @CreationTimestamp, @UpdateTimestamp (thời gian tạo/sửa)

Code ví dụ Hibernate annotation @Version (Hibernate Locking Version)

Locking trong Hibernate, so sánh Optimistic lock với Pessimistic lock

ORM là gì? Tổng quan về ORM Framework

Sự khác nhau giữa String, StringBuffer, StringBuilder?

JPA là gì? Sự khác nhau giữa JPA với Hibernate

Một số câu hỏi hay gặp khi phỏng vấn vị trí lập trình Java

Code ví dụ Hibernate ID tự tăng

```
0a@gmail.com

Hibernate: select role0_.id as id1_6_0_, role0_.delete_flag as delete_f2_6_0_, role0_.nam
role with id = 1000 is null
```

This entry was posted in [FAQ](#), [Hibernate](#) and tagged [faq](#), [hibernate](#). Bookmark the [permalink](#).

[← Code ví dụ hibernate annotation @CreationTimestamp, @UpdateTimestamp \(thời gian tạo/sửa\)](#)
[Hibernate Batch Processing là gì? Batch Processing trong Hibernate →](#)

[\(@GeneratedValue,
@GenericGenerator\)](#)

[Code ví dụ Hibernate
@ElementCollection, lưu dữ liệu
dạng list](#)

[Code ví dụ Hibernate
@EmbeddedId, @Embeddable, Id
gồm nhiều column](#)

[Code ví dụ Hibernate tự sinh ID
dạng text, String](#)

[Code ví dụ Hibernate FetchType =
LAZY \(Lazy loading\)](#)

[orphanRemoval là gì? Code ví dụ
Hibernate orphanRemoval = true](#)

[Code ví dụ Hibernate cascade,
annotation @Cascade](#)

[Code ví dụ Hibernate FetchType =
EAGER \(Eager loading\)](#)

[Code ví dụ Hibernate
@Enumerated, lưu dữ liệu dạng
Enum](#)

[So sánh sự khác nhau giữa
@ElementCollection và
@OneToMany](#)

[Code ví dụ Hibernate @OneToOne
– Quan hệ một – một](#)

[Code ví dụ Hibernate
@ManyToMany – Quan hệ nhiều
nhiều](#)

[Code ví dụ Hibernate One To Many
\(@OneToMany, @ManyToOne\)](#)

[hibernate.dialect là gì – Các loại
SQL Dialects trong hibernate](#)

[Hibernate configuration – Các
thông tin cấu hình hibernate](#)

Code ví dụ Hibernate Pagination,
phân trang trong hibernate

Code ví dụ với JPA Callbacks
method @PrePersist, @PreUpdate,
@PostRemove

So sánh Hibernate Criteria với
HQL, HSQL /JPQL

Code ví dụ Hibernate Criteria
(Hibernate Criteria Queries vs
Restrictions)

Code ví dụ Hibernate Named
Query (annotation @NamedQuery,
@NameQueries)

Sự khác nhau giữa openSession()
và getCurrentSession() trong
Hibernate

So sánh sự khác nhau Hibernate
Session với JPA EntityManager

Code ví dụ truy vấn Hibernate với
EntityManager,
EntityManagerFactory

Code ví dụ Hibernate Session,
SessionFactory (MySQL + Maven +
Eclipse)

Giải thích các annotation trong
Hibernate (code ví dụ)

Series Hibernate: Phần 5 Truy vấn
cơ sở dữ liệu bằng hibernate

Phần 4: Hibernate tạo ra các class
Entity từ các bảng

Series Hibernate: Phần 3 Cài đặt
jboss tool (hibernate tool: công cụ
tạo lớp thực thể từ bảng của
database)

Series Hibernate: Phần 2 Kết nối

[Cơ sở dữ liệu bằng eclipse](#)

[Series Hibernate: Phần 1 Giới thiệu về hibernate framework.](#)

[Hibernate FetchType là gì? Phân biệt FetchType Lazy với Eager](#)

[Sự khác nhau giữa Cascade REMOVE/DELETE với orphanRemoval = true](#)

[So sánh sự khác nhau giữa @OneToOne với @ManyToOne Hibernate](#)

[Hướng dẫn tự học Hibernate Framework bằng tiếng việt](#)

[Lập trình hướng đối tượng là gì? Ưu, nhược điểm](#)

[Lập trình cấu trúc là gì? Ưu nhược điểm.](#)

[Lập trình tuyến tính là gì? Ưu nhược điểm.](#)

[So sánh lập trình cấu trúc với hướng đối tượng](#)

[Cascade trong JPA, Hibernate là gì? Các loại CascadeType](#)

[Servlet Container là gì? Web Server là gì?](#)

[So sánh Statement với PreparedStatement, CallableStatement trong JDBC – Java](#)

[strictfp là gì, Từ khóa strictfp trong Java, ví dụ](#)

[Connection pool là gì? Khái niệm connection pool trong database](#)

[Java 2 là gì? Các version/Phiên bản](#)

[của Java](#)

[Phân biệt giữa Java ME, Java SE và Java EE](#)

[So sánh sự khác nhau giữa J2ME, J2SE và J2EE](#)

CHUYÊN MỤC

[Algorithm](#)

[Apache](#)

[Apache JMeter](#)

[Apache Kafka](#)

[AWS](#)

[C/C++](#)

[CDI](#)

[Clean Code](#)

[Demo](#)

[Design Pattern](#)

[Docker](#)

[Eclipse](#)

[Elasticsearch](#)

[Excel](#)

[FAQ](#)

[Framework](#)

[Freemaker](#)

[FreeMarker](#)

[Gradle](#)

[Hibernate](#)

[HttpClient](#)

[HttpComponents](#)

[Install](#)

IntelliJ IDEA

Java

Java Basic

Java Core

Java8

JavaScript

jooq

JSF

JSP-Servlet

JUnit

Library

Linux

Maven

MongoDB

MySQL

Network Programming

Node.js

OOP

PostgreSQL

PrimeFaces

Principle

Python

quartz

Redis

SDKMan

Security

SocketCluster

Spring

Spring Boot

Spring Core

Spring Data

Spring Hibernate

Spring JDBC

Spring MVC

Spring Security

Thymeleaf

Tomcat

Uncategorized

Web Service

WebSocket

Wordpress

QC ▾

vimeo

If a
butterfly
came out
early,

Keep it private

PROTECTED BY
DMCA



