Hibernate Interview Questions and Answers

What are the Core interfaces are of Hibernate framework?

Answer :: <span style="border-collapse: separate; color: rgb(0, 0, 0); font-family: 'times new roman'; font-size: 16px; font-style: normal; font-variant: normal; font-weight: normal; letter-spacing: normal; line-height: normal; orphans: 2; text-indent: 0px; text-transform: none; white-space: normal; widows: 2; word-spacing: 0px;" class="Apple-style-span">
<p align="justify" style="font-size: 13px; font-family: 'Trebuchet MS',Verdana,Arial,Helvetica,sans-serif; color: rgb(65, 65, 65); margin-bottom: 4px; margin-top: 4px; padding-left: 20px;" class="content">The five core interfaces are used in just about every Hibernate application. Using these interfaces, you can store and retrieve persistent objects and control transactions.
<p align="justify" style="font-size: 13px; font-family: 'Trebuchet MS',Verdana,Arial,Helvetica,sans-serif; color: rgb(65, 65, 65); margin-bottom: 4px; margin-top: 4px; padding-left: 20px;" class="content">
<ul style="font-size: 13px; font-family: 'Trebuchet MS',Verdana,Arial,Helvetica,sans-serif; color: rgb(65, 65, 65); list-style-image: url(http://www.developersbook.com/images/redarrow.gif); list-style-position: outside; margin-top: 0em; margin-bottom: 0em;">

- Session interface
- SessionFactory interface
- Configuration interface
- Transaction interface
- Query and Criteria interfaces

.... Read More

Why do you need ORM tools like hibernate?

Answer :: The main advantage of ORM like hibernate is that it shields developers from messy SQL. Apart from this, ORM provides following benefits:

Improved productivity

- High-level object-oriented API
- Less Java code to write
- No SQL to write

Improved performance

- Sophisticated caching
- Lazy loading
- Eager loading

Improved maintainability

A lot less code to write

Improved portability

ORM framework generates database-specific SQL for you
.... Read More

What are the ORM levels?
Answer :: The ORM levels are:

Pure relational (stored procedure.)
Light objects mapping (JDBC)
Medium object mapping
Full object Mapping (composition,inheritance, polymorphism, persistence by reachability)

.... Read More

What does ORM consists of ?
Answer :: An ORM solution consists of the followig four pieces:

API for performing basic CRUD operations
API to express queries refering to classes
Facilities to specify metadata
Optimization facilities : dirty checking,lazy associations fetching
.... Read More

What is ORM?
Answer :: ORM stands for object/relational mapping. ORM is the automated persistence of objects in a Java application to the tables in a relational database. .... Read More

How does Hibernate distinguish between transient (i.e. newly instantiated) and detached objects?
Answer ::

Hibernate uses the "version" property, if there is one.
If not uses the identifier value. No identifier value means a new object. This does work only for Hibernate managed surrogate keys. Does not work for natural keys and assigned (i.e. not managed by Hibernate) surrogate keys.
Write your own strategy with Interceptor.isUnsaved().
.... Read More

What are the pros and cons of detached objects?
Answer :: <u>Pros: </u>

When long transactions are required due to user think-time, it is the best practice to break the long transaction up into two or more transactions. You can use detached objects from the first transaction to carry data all the way up to the presentation layer. These detached objects get modified outside a transaction and later on re-attached to a new transaction via another session.

<u>Cons </u>

In general, working with detached objects is quite cumbersome, and better to not clutter up the session with them if possible. It is better to discard them and re-fetch them on subsequent requests. This approach is not only more portable but also more efficient because - the objects hang around in Hibernate's cache anyway.

Also from pure rich domain driven design perspective it is recommended to use DTOs (DataTransferObjects) and DOs (DomainObjects) to maintain the separation between Service and UI tiers. .... Read More

What are the benefits of detached objects?
Answer :: Detached objects can be passed across layers all the way up to the presentation layer without having to use any DTOs (Data Transfer Objects). You can later on re-attach the detached objects to another session. .... Read More

What is a Session? Can you share a session object between different theads?
Answer :: Session is a light weight and a non-threadsafe object (No, you cannot share it between threads) that represents a single unit-of-work with the database. Sessions are opened by a SessionFactory and then are closed when all work is complete. Session is the primary interface for the persistence service. A session obtains a database connection lazily (i.e. only when required). To avoid creating too many sessions ThreadLocal class can be used as shown below to get the current session no matter how many times you make call to the currentSession() method.

```
public class HibernateUtil {

public static final ThreadLocal local = new ThreadLocal();

public static Session currentSession() throws HibernateException {
Session session = (Session) local.get();
//open a new session if this thread has no session
if(session == null) {
session = sessionFactory.openSession();
local.set(session);
}
return session;
}
}
```

It is also vital that you close your session after your unit of work completes.
<u>Note:</u> Keep your Hibernate Session API handy. .... Read More

What is a SessionFactory? Is it a thread-safe object?
Answer :: SessionFactory is Hibernate's concept of a single datastore and is threadsafe so that many threads can access it concurrently and request for sessions and immutable cache of compiled mappings for a single database. A SessionFactory is usually only built once at startup. SessionFactory should be wrapped in some kind of singleton so that it can be easily accessed in an application code.

SessionFactory sessionFactory = new Configuration().configure().buildSessionfactory(); .... Read More

How will you configure Hibernate?
Answer :: The configuration files hibernate.cfg.xml (or hibernate.properties) and mapping files *.hbm.xml are used by the Configuration class to create (i.e. configure and bootstrap hibernate) the SessionFactory, which in turn creates the Session instances. Session instances are the primary interface for the persistence service.

" hibernate.cfg.xml (alternatively can use hibernate.properties): These two files are used to configure the hibernate sevice (connection driver class, connection URL, connection username, connection password, dialect etc). If both files are present in the classpath then hibernate.cfg.xml file overrides the settings found in the hibernate.properties file.

" Mapping files (*.hbm.xml): These files are used to map persistent objects to a relational database. It is the best practice to store each object in an individual mapping file (i.e mapping file per class) because storing large number of persistent classes into one mapping file can be difficult to manage and maintain. The naming convention is to use the same name as the persistent (POJO) class name. For example Account.class will have a mapping file named Account.hbm.xml. Alternatively hibernate annotations can be used as part of your persistent class code instead of the *.hbm.xml files. .... Read More