Parul® University

NAAC A++

Enterprise Programming using JAVA

Chapter-4: Hibernet (ORM)

Prof. ARNIKA PATEL
Assistant Professor
Department of CSE

Parul® University

NAAC A++

INDEX

Content

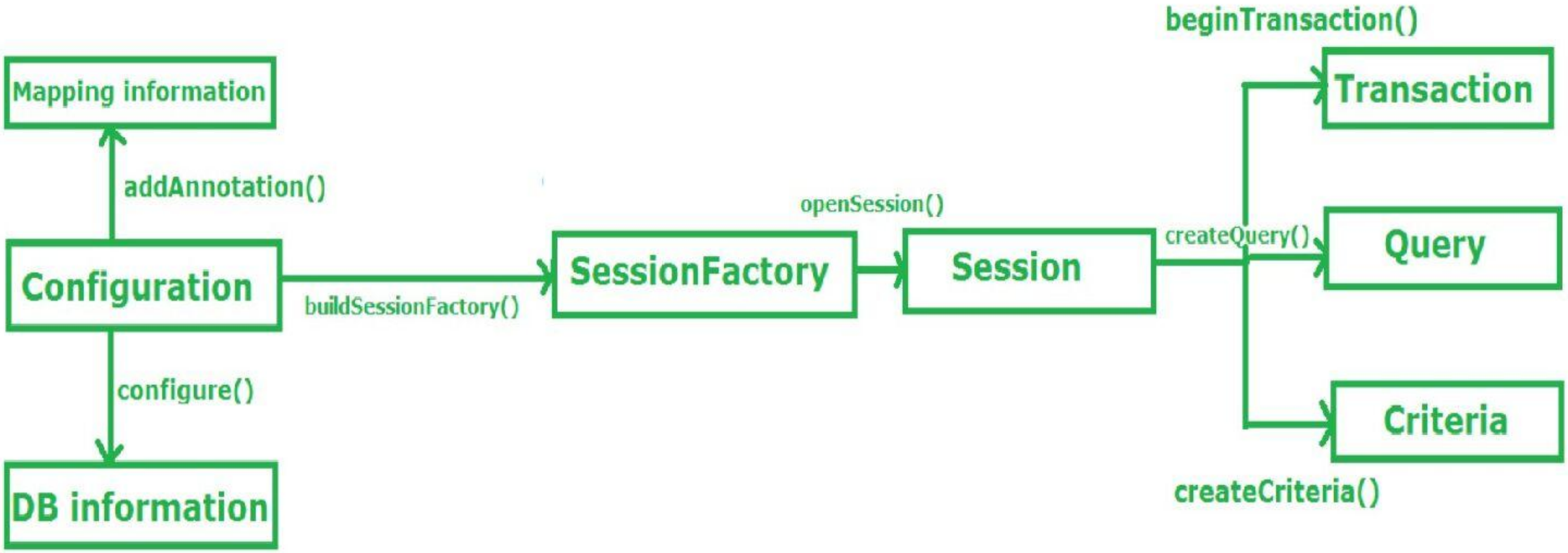
1. Introduction to Hibernet..... 3
2. Hibernet Architecture 4
3. JPA..... 16
4. JPA vs. Hibernet..... 18

Hibernet

Hibernate is a framework which is used to develop persistence logic which is independent of Database software.

In JDBC to develop persistence logic we deal with primitive types. Whereas Hibernate framework we use Objects to develop persistence logic which are independent of database software.

Hibernet Architecture



Hibernet Architecture

Configuration:

- Configuration is a class which is present in org.hibernate.cfg package. It activates Hibernate framework. It reads both configuration file and mapping files.
- It checks whether the config file is syntactically correct or not.
- If the config file is not valid then it will throw an exception. If it is valid then it creates a meta-data in memory and returns the meta-data to object to represent the config file.

Hibernet Architecture

Configuration:

- It activate Hibernate Framework

  Configuration cfg=new Configuration();

- It read both cfg file and mapping files

  cfg.configure();

Hibernet Architecture

SessionFactory:

- SessionFactory is an Interface which is present in org.hibernate package and it is used to create Session Object.
- It is immutable and thread-safe in nature.

- buildSessionFactory() method gathers the meta-data which is in the cfg Object. From cfg object it takes the JDBC information and create a JDBC Connection.

SessionFactory factory=cfg.buildSessionFactory();

Hibernet Architecture

Session:

- Session is an interface which is present in org.hibernate package. Session object is created based upon SessionFactory object i.e. factory.
- It opens the Connection/Session with Database software through Hibernate Framework.
- It is a light-weight object and it is not thread-safe.
- Session object is used to perform CRUD operations.

Session session = factory.openSession();

Hibernet Architecture

Session:

- **openSession()** is a method provided by the SessionFactory that creates and returns a new Session instance.
- This session is not bound to any transaction or context and is independent of any ongoing transactions in the application.
- We can also use getCurrentSession, that returns a Session bound to the current context, which is usually managed by a transaction manager or a framework like Spring.

Session session = sessionFactory.getCurrentSession();

Hibernet Architecture

Transaction:

- Transaction object is used whenever we perform any operation and based upon that operation there is some change in database.
- Transaction object is used to give the instruction to the database to make the changes that happen because of operation as a permanent by using commit() method.

```
Transaction tx=session.beginTransaction();
tx.commit();
```

Hibernet Architecture

Query:

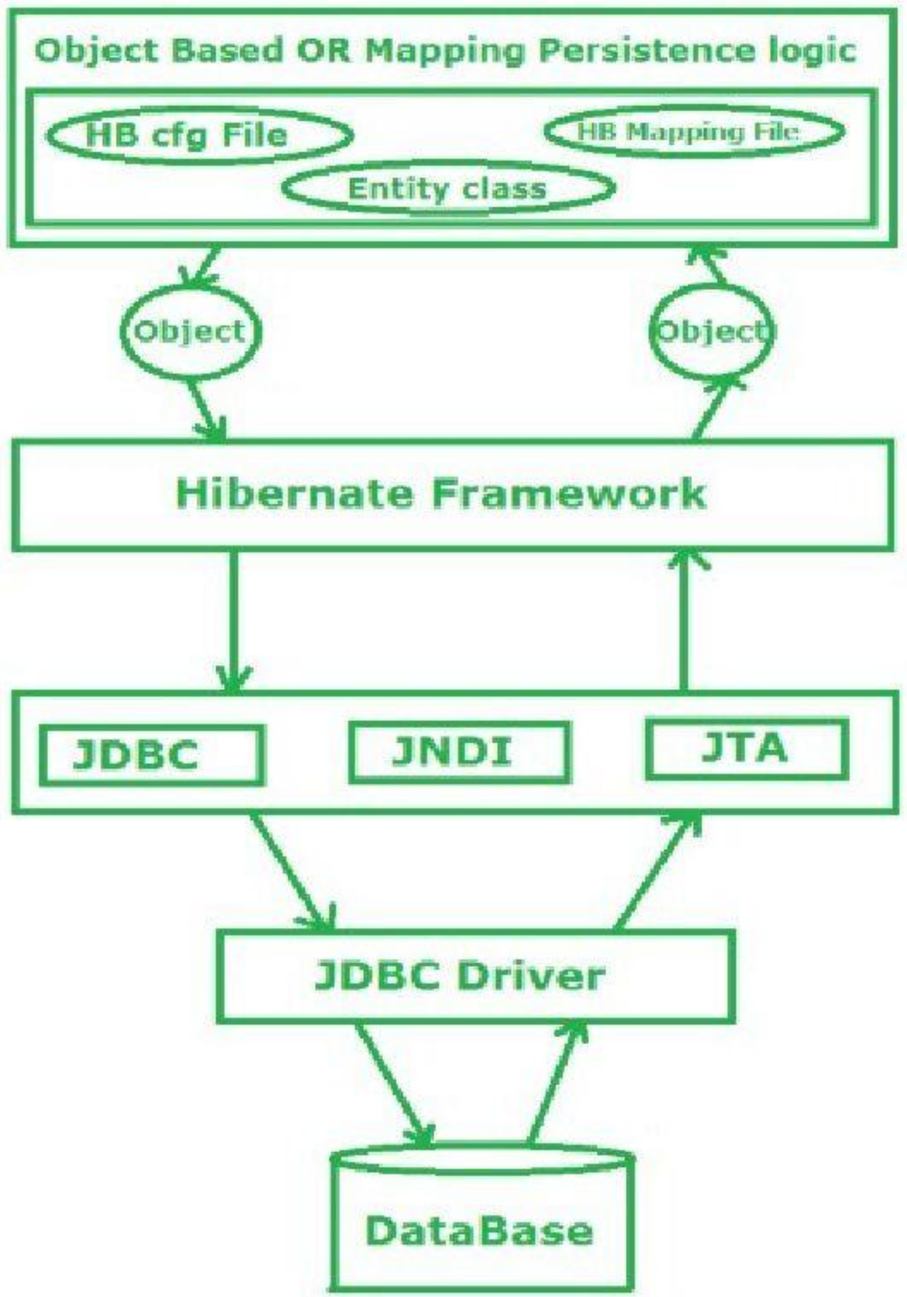
- Query is an interface that present inside org.hibernate package.
- A Query instance is obtained by calling Session.createQuery().
- This interface exposes some extra functionality beyond that provided by Session.iterate() and Session.find():
    1. A particular page of the result set may be selected by calling setMaxResults(), setFirstResult().
    2. Named query parameters may be used.

Hibernet Architecture

Criteria:

- Criteria is a simplified API for retrieving entities by composing Criterion objects.
- The Session is a factory for Criteria. Criterion instances are usually obtained via the factory methods on Restrictions.

Hibernet Architecture Workflow



Hibernet Architecture Workflow

Stage I: In first stage, we will write the persistence logic to perform some specific operations to the database with the help of Hibernate Configuration file and Hibernate mapping file. And after that we create an object of the particular class on which we wrote the persistence logic.

Stage II: In second stage, our class which contains the persistence logic will interact with the hibernate framework where hibernate framework gives some abstraction do perform some task. Now here the picture of java class is over. Now Hibernate is responsible to perform the persistence logic with the help of layers which is below of Hibernate framework or we can say that the layers which are the internal implementation of Hibernate.

Hibernet Architecture Workflow

Stage III: In third stage, our hibernate framework interact which JDBC, JNDI, JTA etc to go to the database to perform that persistence logic.

Stage IV & V: In fourth & fifth stage, hibernate is interact with Database with the help of JDBC driver.

JPA

- JPA stands for Java Persistence API (Application Programming Interface).
- It was initially released on 11 May 2006. It is a Java specification that provides functionality and standards for ORM tools.
- It is used to examine, control, and persist data between Java objects and relational databases. It is regarded as a standard technique for Object Relational Mapping.

JPA

- **Key Features of JPA**
  - It is a lightweight persistence API.
  - Supports object-relational mapping with simple annotations.
  - Provides support for polymorphism and inheritance.
  - Allows dynamic and named queries.
  - Enables seamless switching between different ORM implementations.

JPA vs. Hibernet

The main difference between JPA and Hibernate is that,

- JPA is a specification that defines how ORM tools should function.
- Hibernate is an actual implementation of JPA that provides ORM capabilities.

JPA vs. Hibernet

JPA	Hibernate
JPA is described in **javax.persistence** package.	Hibernate is described in **org.hibernate** package.
It describes the handling of relational data in Java applications.	Hibernate is an Object-Relational Mapping (ORM) tool that is used to save Java objects in the relational database system.
It is not an implementation. It is only a Java specification.	Hibernate is an implementation of JPA. Hence, the common standard which is given by JPA is followed by Hibernate.
It is a standard API that permits to perform database operations.	It is used in mapping Java data types with SQL data types and database tables.
As an object-oriented query language, it uses Java Persistence Query Language (JPQL) to execute database operations.	As an object-oriented query language, it uses Hibernate Query Language (HQL) to execute database operations.
To interconnect with the entity manager factory for the persistence unit, it uses the **EntityManagerFactory** interface. Thus, it gives an entity manager.	To create Session instances, it uses the **SessionFactory** interface.
To make, read, and remove actions for instances of mapped entity classes, it uses the **EntityManager** interface. This interface interconnects with the persistence condition.	To make, read, and remove actions for instances of mapped entity classes, it uses **Session** interface. It acts as a runtime interface between a Java application and Hibernate.

PPT Content Resources Reference Sample:

1. **Book Reference**

   Jim Farley, William Crawford, David Flanagan. Java Enterprise in a Nutshell, O'Reilly

2. **Book Reference**

   Rocha, R., Purificação, J. (2018). Java EE 8 Design Patterns and Best Practices: Build Enterprise-ready Scalable Applications with Architectural Design Patterns. Germany: Packt Publishing..

3. **Website Reference**

   https://www.scribd.com/document/268349254/Java-8-Programming-Black-Book .

4. **Sources**

   https://developers.redhat.com/topics/enterprise-java

5. **Article**

   https://www.researchgate.net/publication/276412369_Advanced_Java_Programming

Parul® University | NAAC GRADE A++

https://paruluniversity.ac.in/