Hibernate XML Configuration Implementation Walkthrough

1. Object-Relational Mapping in Hibernate XML Configuration File

   Hibernate enables mapping of Java classes to database tables via XML configuration files, primarily using:

   hibernate.cfg.xml – Main configuration file.
   .hbm.xml files – Class-to-table mapping files.

   Example of Mapping File (Employee.hbm.xml):

   ```
   <hibernate-mapping>
     <class name="Employee" table="EMPLOYEE">
       <id name="id" column="ID">
         <generator class="native"/>
       </id>
       <property name="firstName" column="FIRST_NAME"/>
       <property name="lastName" column="LAST_NAME"/>
       <property name="salary" column="SALARY"/>
     </class>
   </hibernate-mapping>
   ```

   <class>: maps the Java class to a table.
   <id>: defines the primary key.
   <property>: maps fields to table columns.

2. Hibernate Core Components and End-to-End Operations

   Hibernate uses several key interfaces and methods for ORM operations:

   a. SessionFactory
      - A thread-safe (immutable) factory of `Session` objects.
      - Created once per application using **Configuration().configure().buildSessionFactory().**
      - Heavyweight object; should be shared across application.

   b. Session
      - Lightweight and non-thread-safe object.
      - Represents a single unit of work with the database.
      - Provides APIs to create, read, update, and delete objects.
        Example:
        Session session = sessionFactory.openSession();

   c. Transaction
      - Represents a unit of work.
      - Handles ACID properties.
      - Always use transactions for data-altering operations.

d. beginTransaction()
   - Starts a new transaction.
   - Returns a Transaction object.

   Transaction tx = session.beginTransaction();
e. commit()
   - Commits the current transaction, making changes permanent in the database.

   tx.commit();
f. rollback()
   - Rolls back the current transaction in case of failure.

   tx.rollback();
g. session.save(Object entity)
   - Saves a new entity object to the database.
   - Returns the generated identifier.

   session.save(employee);
h. session.createQuery("from Entity").list()
   - Creates an HQL (Hibernate Query Language) query.
   - list() returns all results in a List.

   List<Employee> list = session.createQuery("from Employee").list();
i. session.get(Class, Serializable id)
   - Fetches an entity by primary key.
   - Returns null if no matching row found.

   Employee e = session.get(Employee.class, 1);
j. session.delete(Object entity)
   - Deletes a persistent object.
   - Entity must be attached to the session.

   session.delete(employee);