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

Transaction Management

- A transaction is a sequence of operation which works as an atomic unit.
- A transaction only completes if all the operations completed successfully.
- A transaction has the Atomicity, Consistency, Isolation and Durability properties (ACID).

Transaction Management

Transaction interface:

Transaction interface provides the facility to define the units of work or transactions.

A transaction is associated with a session.

We have to call beginTransaction()method of Session to start a transaction (Session.beginTransaction()).

Transaction Management

Commonly used methods of Transaction interface:

1. begin(): It starts a new transaction.
Syntax:
public void begin() throws HibernateException

2. commit(): It ends the transaction and flush the associated session.
Syntax:
public void commit() throws HibernateException

Transaction Management

Commonly used methods of Transaction interface:

3. rollback(): It roll back the current transaction.
Syntax:
public void rollback()throws HibernateException

4. setTimeout(int seconds): It set the transaction timeout for any transaction started by a subsequent call to begin() on this instance.
Syntax:
public void **setTimeout(int seconds)** throws HibernateException

Transaction Management

Commonly used methods of Transaction interface:

3. rollback(): It roll back the current transaction.
Syntax:
public void rollback()throws HibernateException

4. setTimeout(int seconds): It set the transaction timeout for any transaction started by a subsequent call to begin() on this instance.
Syntax:
public void **setTimeout(int seconds)** throws HibernateException

Transaction Management

Commonly used methods of Transaction interface:

5. isActive(): It checks that is this transaction still active or not.

Syntax:

public boolean isActive()throws HibernateException

6. wasRolledBack(): It checks that is this transaction roll backed successfully or not.

Syntax:

public boolean wasRolledBack()throws HibernateException

Transaction Management

Commonly used methods of Transaction interface:

5. isActive(): It checks that is this transaction still active or not.

Syntax:

public boolean isActive()throws HibernateException

6. wasRolledBack(): It checks that is this transaction roll backed successfully or not.

Syntax:

public boolean wasRolledBack()throws HibernateException

Transaction Management

Commonly used methods of Transaction interface:

7. wasCommitted(): It checks that is this transaction committed successfully or not.

Syntax:

public boolean wasCommitted()throws HibernateException

8.registerSynchronization(Synchronization synchronization): It register a user synchronization callback for this transaction.

Syntax:

public boolean registerSynchronization(Synchronization synchronization)throws HibernateException

Transaction Management

```
Transaction tx = null;
Session session =
HibernateUtil.getSessionFactory().openSession();
try{
        tx = session.beginTransaction();
        //Perform some operation here
        tx.commit();
}catch (HibernateException e) {
        if(tx!=null){
                tx.rollback();
         } e.printStackTrace();
}finally {
        session.close();
}
```

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/