Java Assignment 20/06/2024

Name: Syedanwar Hanzahusen Mashalkar

==========

1. Create the Persistent Class:

This class will represent a table in our database.

```
package com.asignment;

import javax.persistence.*;

@Entity
@Table(name = "STUDENT")
public class Student {
   @Id
      @Column(name = "ID")
   private int id;

   @Column(name = "NAME")
   private String name;

   @Column(name = "EMAIL")
   private String email;

   public Student() {}

   public int getId() {
      return id;
   }

   public void setId(int id) {
      this.id = id;
   }

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String getEmail() {
      return email;
   }

   public void setEmail(String email) {
      this.email = email;
```

```
    }
}
```

2. Create the Mapping File for Persistent Class

```
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
   <class name="com.example.Student" table="STUDENT">
      <id name="id" column="ID">
         <generator class="native"/>
      </id>
      <property name="name" column="NAME"/>
      <property name="email" column="EMAIL"/>
   </class>
</hibernate-mapping>
```

3. Create the Configuration File

The Hibernate configuration file will include the database connection settings and Hibernate properties.

```
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
   <session-factory>
      <!-- Database connection settings -->
      <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
      <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/yourdatabase</property>
      <property name="hibernate.connection.username">yourusername</property>
      <property name="hibernate.connection.password">yourpassword</property>

      <!-- JDBC connection pool settings -->
      <property name="hibernate.c3p0.min_size">5</property>
      <property name="hibernate.c3p0.max_size">20</property>
      <property name="hibernate.c3p0.timeout">300</property>
      <property name="hibernate.c3p0.max_statements">50</property>
      <property name="hibernate.c3p0.idle_test_period">3000</property>

      <!-- SQL dialect -->
      <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

      <!-- Enable Hibernate's automatic session context management -->
      <property name="hibernate.current_session_context_class">thread</property>

      <!-- Echo all executed SQL to stdout -->
      <property name="hibernate.show_sql">true</property>

      <!-- Drop and re-create the database schema on startup -->
      <property name="hibernate.hbm2ddl.auto">update</property>

      <!-- Annotated classes -->
      <mapping class="com.example.Student"/>
   </session-factory>
```

```
</hibernate-configuration>
```

4. Create the Class to Retrieve or Persist the Object

This class will use Hibernate to save and retrieve `Student` objects.

```
package com.assignment;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class StudentManager {
   private static SessionFactory factory;

   public static void main(String[] args) {
      try {
         factory = new Configuration().configure().buildSessionFactory();
      } catch (Throwable ex) {
         System.err.println("Failed to create sessionFactory object." + ex);
         throw new ExceptionInInitializerError(ex);
      }

      StudentManager manager = new StudentManager();

      // Create a new student
      Integer studentID1 = manager.addStudent("John Doe", "john.doe@example.com");

      // Retrieve student
      manager.getStudent(studentID1);
   }

   /* Method to CREATE a student in the database */
   public Integer addStudent(String name, String email) {
      Session session = factory.openSession();
      Integer studentID = null;
      try {
         session.beginTransaction();
         Student student = new Student();
         student.setName(name);
         student.setEmail(email);
         studentID = (Integer) session.save(student);
         session.getTransaction().commit();
      } catch (Exception e) {
         if (session.getTransaction() != null) session.getTransaction().rollback();
         e.printStackTrace();
      } finally {
         session.close();
      }
      return studentID;
   }
```

```
    /* Method to READ a student */
    public void getStudent(Integer studentID) {
       Session session = factory.openSession();
       try {
          session.beginTransaction();
          Student student = (Student) session.get(Student.class, studentID);
          if (student != null) {
             System.out.println("Name: " + student.getName());
             System.out.println("Email: " + student.getEmail());
          }
          session.getTransaction().commit();
       } catch (Exception e) {
          if (session.getTransaction() != null) session.getTransaction().rollback();
          e.printStackTrace();
       } finally {
          session.close();
       }
    }
}
```

5. Load the JAR File

```
<dependencies>
    <!-- Hibernate dependencies -->
    <dependency>
        <groupId>org.hibernate</groupId>
        <artifactId>hibernate-core</artifactId>
        <version>5.4.10.Final</version>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.17</version>
    </dependency>
</dependencies>
```