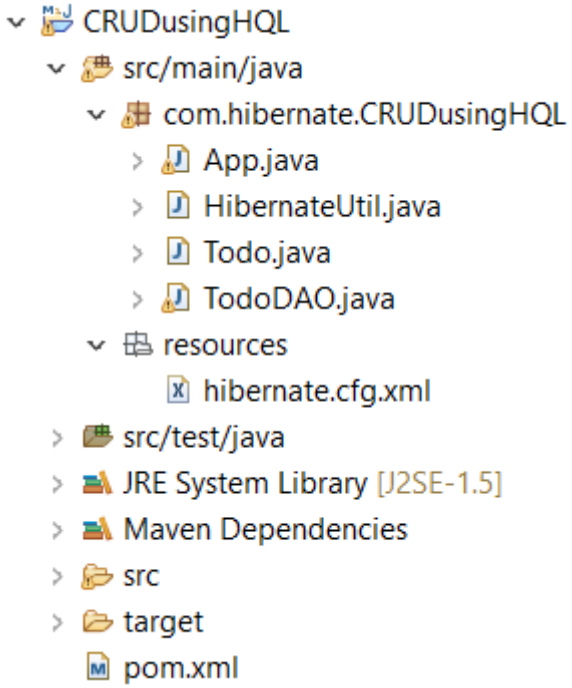
Project Structure



Step 1: Create Maven Project

1. Open your IDE (e.g., IntelliJ IDEA, Eclipse).
2. Create a new Maven project.
    - Group ID: com.hibernate
    - Artifact ID: CRUDusingHQL

Step 2: Add the below Dependencies to pom.xml

1. mysql-connector-j
2. hibernate-core (hibernate-core)
3. jakarta.persistence-api (jakarta.persistence)
4. javax.validation (javax.validation)

Step-3 Create the below Classes

1. Create the Todo class in the com.hibernate.CRUDusingHQL package.
2. Create the TodoDAO class in the com.hibernate.CRUDusingHQL package.

Class : Todo

```
package com.hibernate.CRUDusingHQL;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Todo {
        @Id
```

```
	@GeneratedValue(strategy = GenerationType.IDENTITY)
	private int id;
	private String title;
	private String description;
	private boolean completed;

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

	public String getTitle() { return title; }
	public void setTitle(String title) { this.title = title;}

	public String getDescription() { return description; }
	public void setDescription(String description) { this.description = description; }

	public boolean getCompleted() { return completed; }
	public void setCompleted(boolean completed) { this.completed = completed; }
}
```

Class : TodoDAO

```
package com.hibernate.CRUDusingHQL;

import java.util.*;

import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

public class TodoDAO {

	Session session;
	Transaction transaction = null;

	public TodoDAO() { session = HibernateUtil.getSessionFactory().openSession(); }

	public void saveTodo(Todo todo) {
		try {
			transaction = session.beginTransaction();

			session.save(todo);
			transaction.commit();

		} catch (Exception e) {
			System.out.println("Error :" + e.getMessage());
			if (transaction != null) { transaction.rollback(); }
			e.printStackTrace();
		}
```

```
}

public void updateTodo(Todo todo) {
	try {
		transaction = session.beginTransaction();
		session.update(todo);
		transaction.commit();
	} catch (Exception e) {
		if (transaction != null) { transaction.rollback(); }
		e.printStackTrace();
	}
}

public List<Todo> getTodos() {
	List<Todo> alltodos = new ArrayList<Todo>();
	try {
		transaction = session.beginTransaction();

		Query<Todo> query = session.createQuery("from Todo", Todo.class);
		System.out.println(query.list().size());
		alltodos = query.list();
	} catch (Exception e) {
		System.out.println("Error :" + e.getMessage());
	}
	return alltodos;
}

public void deleteTodo(int id) {
	try {
		transaction = session.beginTransaction();
		Todo todo = session.get(Todo.class, id);
		if (todo != null) {
			session.delete(todo);
			System.out.println("Todo is deleted");
		}
		else
			System.out.println("Todo does not exists");
		transaction.commit();
	} catch (Exception e) {
		if (transaction != null) { transaction.rollback(); }
		e.printStackTrace();
```

```
            }
        }

        public Todo getTodoByID(int id) {
            Todo todo = new Todo();
            try {
                todo = session.get(Todo.class, id);
            } catch (Exception e) {
                e.printStackTrace();
            }
            return todo;
        }
}
```

Step 4: Create a resources package inside the src/man/java folder.
Create a XML file named hibernate.cfg.xml with the following content inside the resources folder.

```
<?xml version="1.0" encoding="UTF-8"?>
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class"> com.mysql.cj.jdbc.Driver </property>
        <property name="hibernate.connection.url">
            jdbc:mysql://localhost:3306/hibernateDB
        </property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">password</property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="show_sql">true</property>
        <property name="format_sql">true</property>
        <property name="hbm2ddl.auto">create </property>
        <mapping class="com.hibernate.CRUDusingHQL.Todo"></mapping>
    </session-factory>
</hibernate-configuration>
```

Note: Ensure the MySQL database hibernateDB exists, and replace the username and password with your actual MySQL credentials.

Step 5: Create the HibernateUtil Class

Create the HibernateUtil class in the com.hibernate.one_to_one_example package:

```
package com.hibernate.one_to_one_example;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
	private static final SessionFactory sessionFactory = buildSessionFactory();
	private static SessionFactory buildSessionFactory() {
		SessionFactory sessionFactory = null;
		try {
			Configuration configuration = new Configuration();
			configuration.configure("resources/hibernate.cfg.xml");
			sessionFactory = configuration.buildSessionFactory();
		}
		catch (Exception e) { e.printStackTrace(); }
		return sessionFactory;
	}

	public static SessionFactory getSessionFactory()  {
		return sessionFactory;
	}
}
```

Step-6 Now add the below lines of code into **App.java** class file.

Please note each and every data operation call one by one (Un comment those code which you want to run such as Add todo, show all, update todo, select todo by id, and delete todo by id.

```
package com.hibernate.CRUDusingHQL;

import org.hibernate.Transaction;
import java.util.*;
public class App
{
   public static void main( String[] args )
   {
	TodoDAO tododao = new TodoDAO();

	//----------------------------------------
	//Add new record
	//----------------------------------------
	Transaction tran = null;
	try {
```

```
            Todo todo = new Todo();
            todo.setId(1);
            todo.setTitle("Complete the LAB work");
            todo.setDescription("Hibernate crud operation using HQL");
            todo.setCompleted(true);

                    tododao.saveTodo(todo);
    }
    catch (Exception e) {
            System.out.println("Error :" + e.getMessage());
    }

    //-------------------------------------------
    // Select all todos
    //-------------------------------------------
    /*
List<Todo> todos = tododao.getTodos();
System.out.println(todos.size());
for(Todo todo : todos) {
    System.out.println(todo.getId() + "\t" + todo.getTitle());
}
*/

    //-------------------------------------------
    //Update the todo
    //-------------------------------------------
    /*
    try {
            int id = 1;
            Todo todo = tododao.getTodoByID(id);
            if(todo != null) {
                    todo.setId(id);
                    todo.setTitle("New Complete the LAB Work");
                    todo.setDescription("Hibernate crud operation using HQL");
                    todo.setCompleted(false);
                    tododao.updateTodo(todo);
            }
            else
                    System.out.println("Todo does not exists!");
    }
    catch (Exception e) {
```

```
            System.out.println("Error :" + e.getMessage());
        }
        */

        //-----------------------------------------
        //Select todo by id
        //-----------------------------------------
        /*
        try {
            int id = 1;
            Todo todo = tododao.getTodoByID(id);
            if(todo != null) {
                System.out.println("Id        :" + todo.getId());
                System.out.println("Tital     :" + todo.getTitle());
                System.out.println("Description:" + todo.getDescription());
                System.out.println("Status    :" + todo.getCompleted());
            }
            else
                System.out.println("Todo does not exists!");
        }
        catch(Exception ex) {
            System.out.println("Error :" + ex.getMessage());
        }
        */

        //-----------------------------------------
        //Delete the todo id
        //-----------------------------------------
        //tododao.deleteTodo(10);
    }
}
```

Step-7 Right click on App.java file and Run the application