

## 1. HibernateWithJavaProjectXMLBase

a. Create new Java Project

b. Add all jar files

c. Create two packages.

    one for entities - com.entities

    one for main class - com.driver

d. Create POJO/Bean/Model/Entity class CustomerDetails in com.entities

-----CustomerDetails-----

```
public class CustomerDetails {  
    private int cid;  
    private String cname;  
    private String caddress;  
    public String getCName() {  
        return cname;  
    }  
    public void setCName(String cname) {  
        this.cname = cname;  
    }  
    public int getCid() {  
        return cid;  
    }  
    public void setCid(int cid) {  
        this.cid = cid;  
    }  
    public String getCAddress() {  
        return caddress;  
    }  
    public void setCAddress(String caddress) {  
        this.caddress = caddress;  
    }  
}
```

-----  
e. add code in main method

```
public static void main(String[] args) {  
    CustomerDetails obj = new CustomerDetails();  
    Configuration conf = new Configuration();  
    conf.configure();  
    SessionFactory sf = conf.buildSessionFactory();  
    Session session = sf.openSession();  
    Transaction trans = session.beginTransaction();  
    obj.setCname("Paras");  
    obj.setCaddress("Bangalore");  
    session.save(obj);  
    trans.commit();  
}
```

-----  
f. In src add Hibernate configuration file.

name and extension should be

"hibernate.cfg.xml"

Template -->

Code for hibernate.cfg.xml

```
-----  
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE hibernate-configuration SYSTEM  
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  
<hibernate-configuration>  
    <session-factory>  
        <property name =  
"hibernate.dialect">org.hibernate.dialect.SQLServerDialect</property>  
        <property name =  
"hibernate.connection.driver_class">com.microsoft.sqlserver.jdbc.SQLServerDriver</property>  
  
        <!-- Assume test is the database name -->
```

```

        <property name =
"hibernate.connection.url">jdbc:sqlserver://localhost;databaseName=emp;instanceName=SQLEXPRESS2019;</property>

        <property name = "hibernate.connection.username">sa</property>
        <property name = "hibernate.connection.password">password_123</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <!-- List of XML mapping files -->
        <mapping resource = "CustomerDetails.hbm.xml"/>
    </session-factory>
</hibernate-configuration>

```

#### g. hibernate-mapping

Now, we will have to map our pojo class in order to create table in database.

"CustomerDetails.hbm.xml"

```

-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD//EN"
"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
<class name = "com.entities.CustomerDetails" table = "CusomerDetails">

    <meta attribute = "class-description">
This class contains the Customer detail.
    </meta>

    <id name = "cid" type = "int" column = "cid">
<generator class="native"/>
    </id>

```

```

        <property name = "cname" column = "first_name" type = "string"/>
        <property name = "caddress" column = "last_name" type = "string"/>

    </class>

</hibernate-mapping>

```

## 2. HibernateWithJavaProjectAnnotationBase

a. Create new project

b. Add all jar files

c. Create two packages.

one for entities - com.entities

one for main class - com.driver

d. Create POJO/Bean/Model/Entity class CustomerDetails in com.entities

-----CustomerDetails-----

```
import javax.persistence.Column;
```

```
import javax.persistence.Entity;
```

```
import javax.persistence.GeneratedValue;
```

```
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
```

```
import javax.persistence.Table;
```

```
@Entity
```

```
@Table(name="CustDetails")
```

```
public class CustomerDetails {
```

```
@Id
```

```
@GeneratedValue(strategy = GenerationType.AUTO)
```

```
@Column(name = "CustId")
```

```
private int cid;
```

```
@Column(name = "CustName")
```

```
private String cname;
```

```
@Column(name = "CustAddress")
```

```

private String caddress;

public String getCName() {
    return cname;
}

public void setCName(String cname) {
    this.cname = cname;
}

public int getCid() {
    return cid;
}

public void setCid(int cid) {
    this.cid = cid;
}

public String getCAddress() {
    return caddress;
}

public void setCAddress(String caddress) {
    this.caddress = caddress;
}
}

```

-----

e. add code in main method

```

public static void main(String[] args) {
    CustomerDetails obj = new CustomerDetails();
    Configuration conf = new Configuration();
    conf.configure();
    SessionFactory sf = conf.buildSessionFactory();
    Session session = sf.openSession();
    Transaction trans = session.beginTransaction();
    obj.setCName("Paras");
    obj.setCAddress("Bangalore");
}

```

```

        session.save(obj);

        trans.commit();
    }

```

f. In src add Hibernate configuration file.

name and extension should be

"hibernate.cfg.xml"

Template -->

Code for hibernate.cfg.xml

```

-----

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-configuration SYSTEM

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

    <session-factory>

        <property name =
"hibernate.dialect">org.hibernate.dialect.SQLServerDialect</property>

        <property name =
"hibernate.connection.driver_class">com.microsoft.sqlserver.jdbc.SQLServerDriver</property>

        <!-- Assume test is the database name -->

        <property name =
"hibernate.connection.url">jdbc:sqlserver://localhost;databaseName=emp;instanceName=SQLEXPRESS2019;</property>

        <property name = "hibernate.connection.username">sa</property>

        <property name = "hibernate.connection.password">password_123</property>

        <property name="hibernate.hbm2ddl.auto">update</property>

        <!-- List of XML mapping files -->

        <mapping class = "com.entities.CustomerDetails"/>

    </session-factory>

</hibernate-configuration>

-----

```

### 3. HibernateWithMavenProjectXML/AnnotationBase

- a. create maven project
- b. add dependencies for hibernate
- c. add dependency for SQL server
- d. use code from previous demos.

```
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.6.14.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc -->
<dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <version>11.2.2.jre18</version>
</dependency>

</dependencies>
```

---