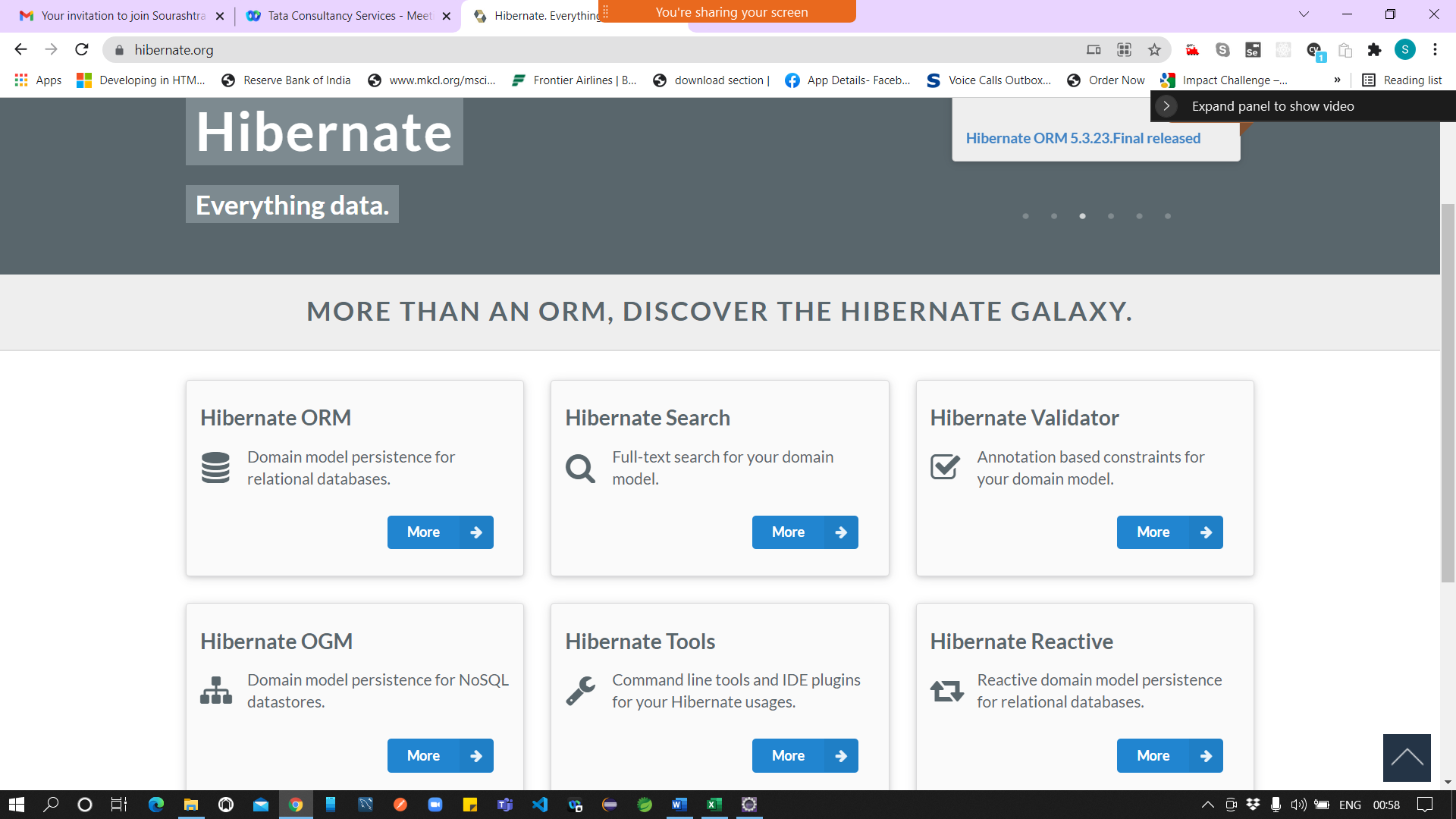
JPA – Hibernate

<https://hibernate.org/>



Hibernate needs a configuration file named “hibernate.cfg.xml”

Java EE – Projects lot of xml configuration

Xml is case and space sensitive.

<https://docs.jboss.org/hibernate/orm/5.5/quickstart/html_single/>

*Example 4. Obtaining the*org.hibernate.SessionFactory

protected void setUp() throws Exception {

// A SessionFactory is set up once for an application!

final StandardServiceRegistry registry = new StandardServiceRegistryBuilder()

.configure() // configures settings from hibernate.cfg.xml

.build();

try {

sessionFactory = new MetadataSources( registry ).buildMetadata().buildSessionFactory();

}

catch (Exception e) {

// The registry would be destroyed by the SessionFactory, but we had trouble building the SessionFactory

// so destroy it manually.

StandardServiceRegistryBuilder.destroy( registry );

}

}

*Example 5. Saving entities*

Session session = sessionFactory.openSession();

session.beginTransaction();

session.save( new Event( "Our very first event!", new Date() ) );

session.save( new Event( "A follow up event", new Date() ) );

session.getTransaction().commit();

session.close();

*Example 6. Obtaining a list of entities*

session = sessionFactory.openSession();

session.beginTransaction();

List result = session.createQuery( "from Event" ).list();

for ( Event event : (List<Event>) result ) {

System.out.println( "Event (" + event.getDate() + ") : " + event.getTitle() );

}

session.getTransaction().commit();

session.close();

*Example 8. Identifying the identifier property*

@Id

@GeneratedValue(generator="increment")

@GenericGenerator(name="increment", strategy = "increment")

public Long getId() {

return id;

}

*Example 9. Identifying basic properties*

public String getTitle() {

return title;

}

@Temporal(TemporalType.TIMESTAMP)

@Column(name = "EVENT\_DATE")

public Date getDate() {

return date;

}

*Example 10. persistence.xml*

<persistence xmlns="http://java.sun.com/xml/ns/persistence"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/persistence http://java.sun.com/xml/ns/persistence/persistence\_2\_0.xsd"

version="2.0">

<persistence-unit name="org.hibernate.tutorial.jpa">

...

</persistence-unit>

</persistence>

*Example 11. Obtaining the javax.persistence.EntityManagerFactory*

protected void setUp() throws Exception {

sessionFactory = Persistence.createEntityManagerFactory( "org.hibernate.tutorial.jpa" );

*Example 12. Saving (persisting) entities*

EntityManager entityManager = sessionFactory.createEntityManager();

entityManager.getTransaction().begin();

entityManager.persist( new Event( "Our very first event!", new Date() ) );

entityManager.persist( new Event( "A follow up event", new Date() ) );

entityManager.getTransaction().commit();

entityManager.close();

*Example 13. Obtaining a list of entities*

entityManager = sessionFactory.createEntityManager();

entityManager.getTransaction().begin();

List<Event> result = entityManager.createQuery( "from Event", Event.class ).getResultList();

for ( Event event : result ) {

System.out.println( "Event (" + event.getDate() + ") : " + event.getTitle() );

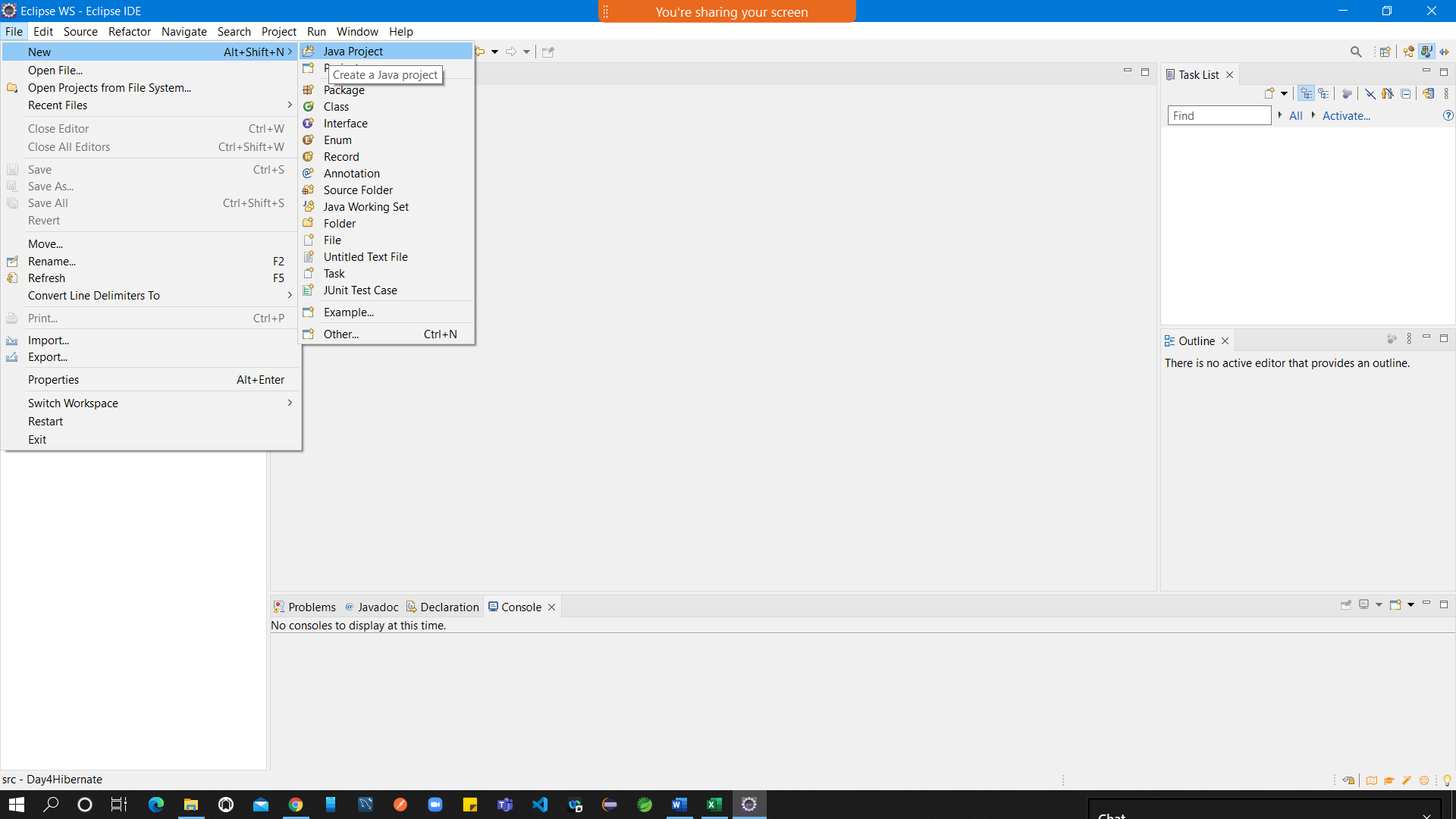
}

entityManager.getTransaction().commit();

entityManager.close();

Steps

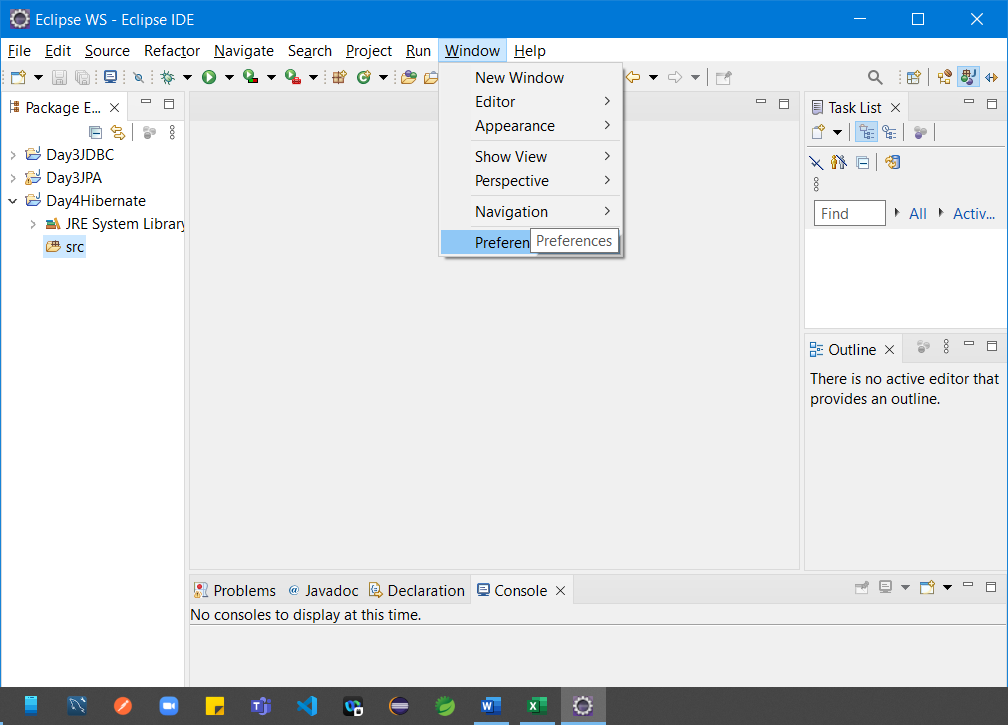
1. Create a new Java Project in Eclipse



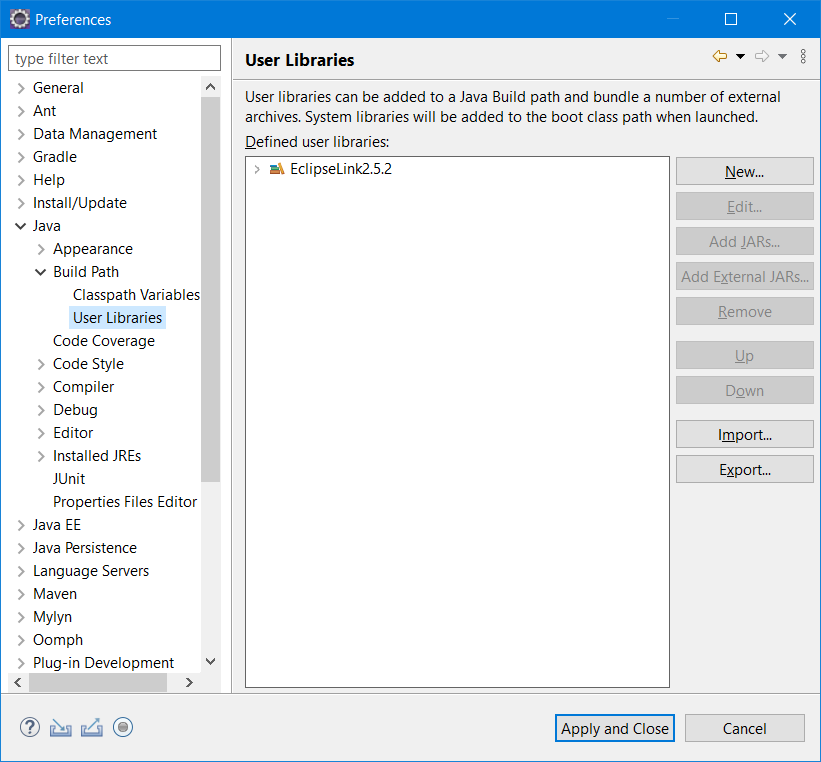
1. Download the Hibernate JARS from official site

<https://hibernate.org/orm/releases/5.5/>

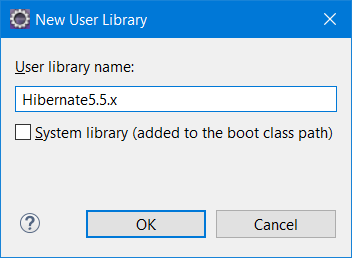
1. Extract the content
2. Create a new user library in Eclipse for Hibernate (open Window -Preferences)



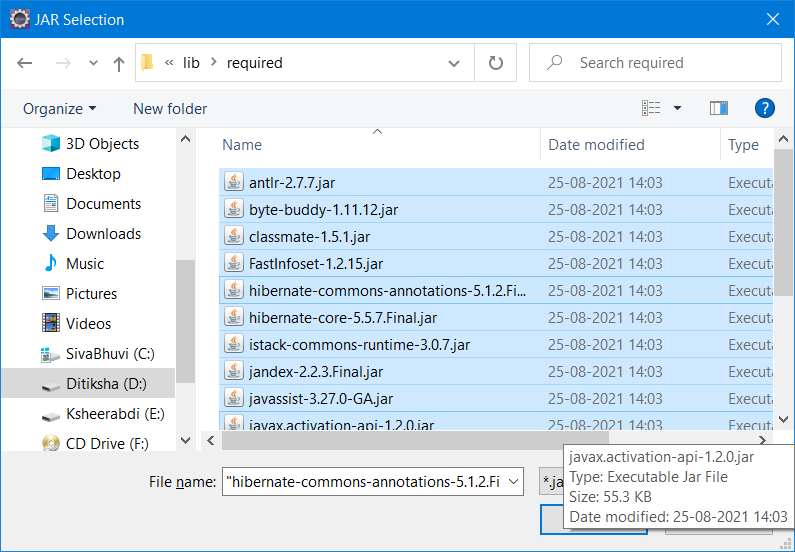
1. Select “User Libraries” under Java -> Build Path and click “New” button



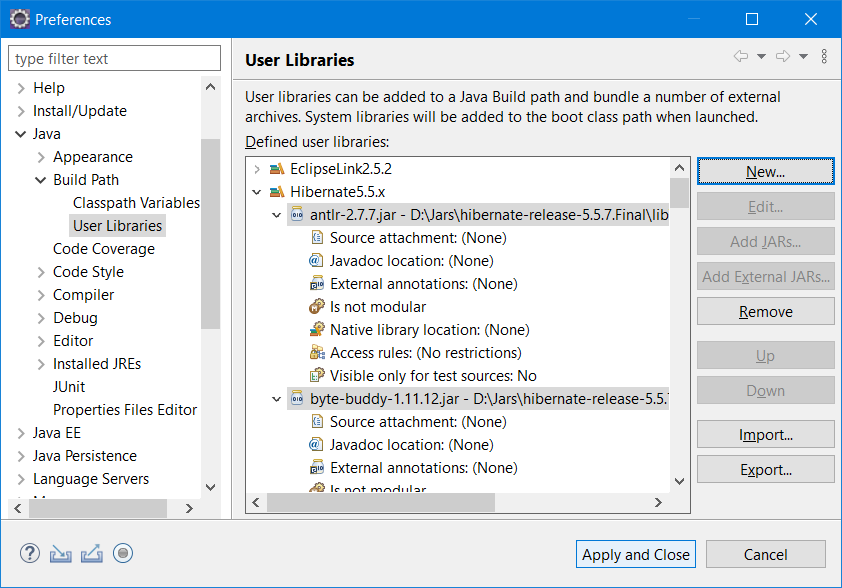
1. Enter the name of library as “Hibernate5.5.x” and click “ok” button.



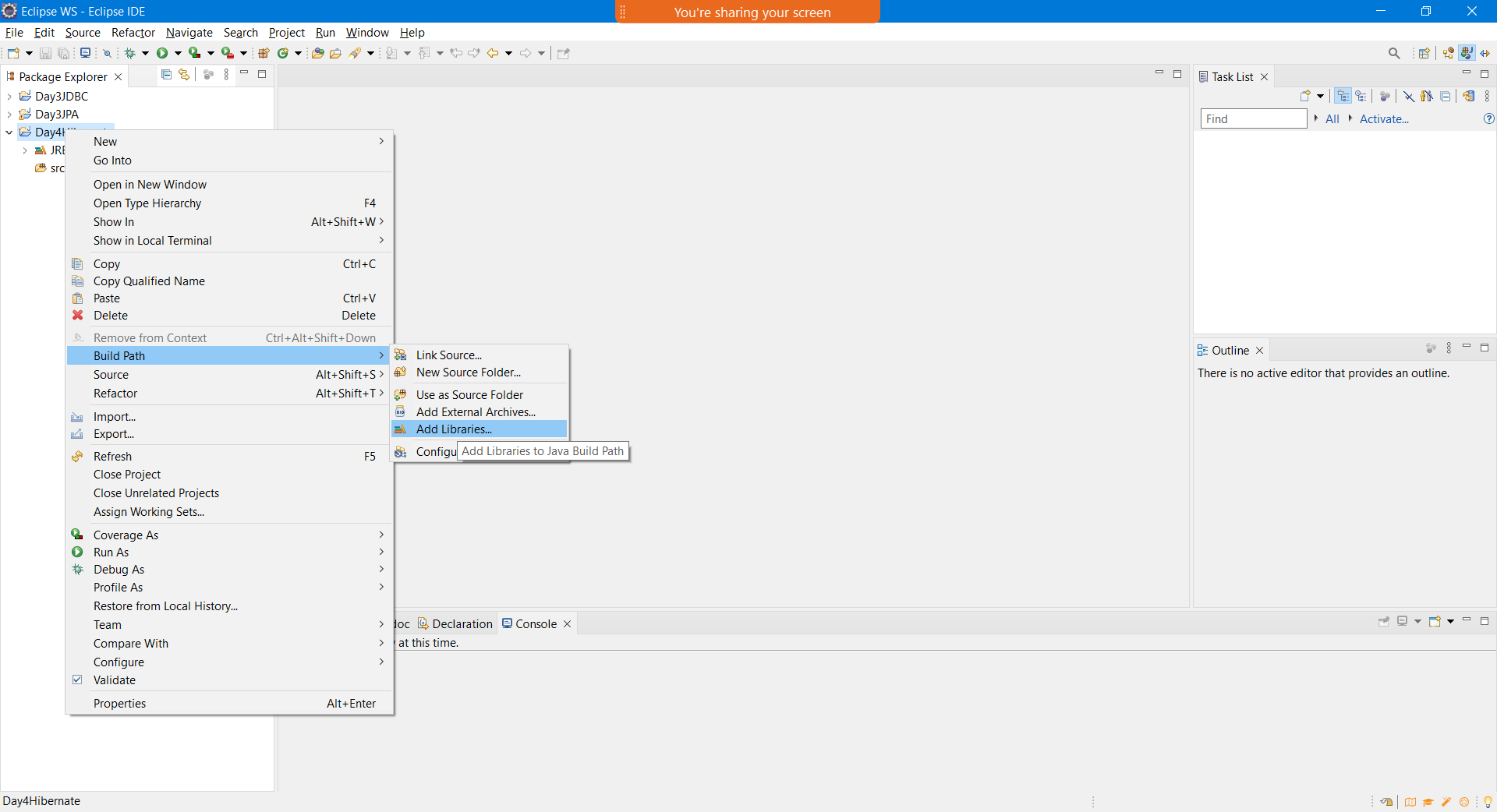
1. Select the newly added user library and click “Add External Jars” button.
2. Select all the required jars from the extracted content. (<extracted\_folder>\hibernate-release-5.5.7.Final\lib\required ) and click “open” button.

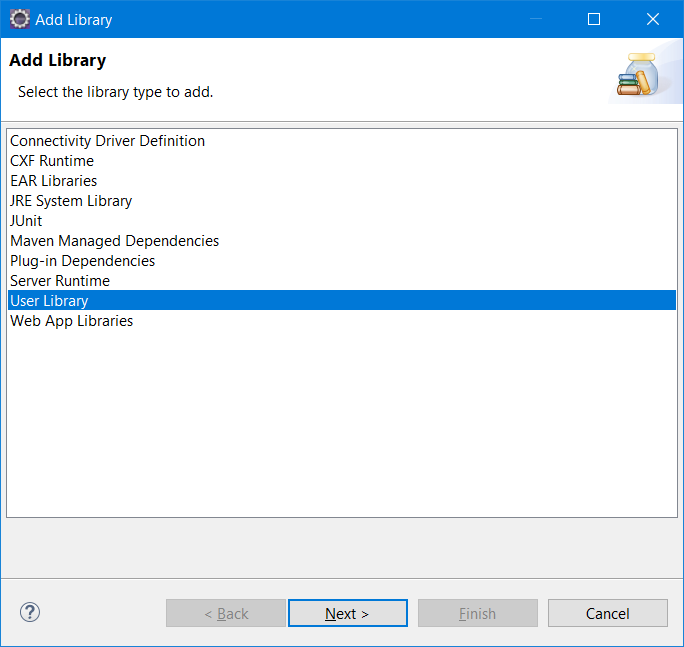


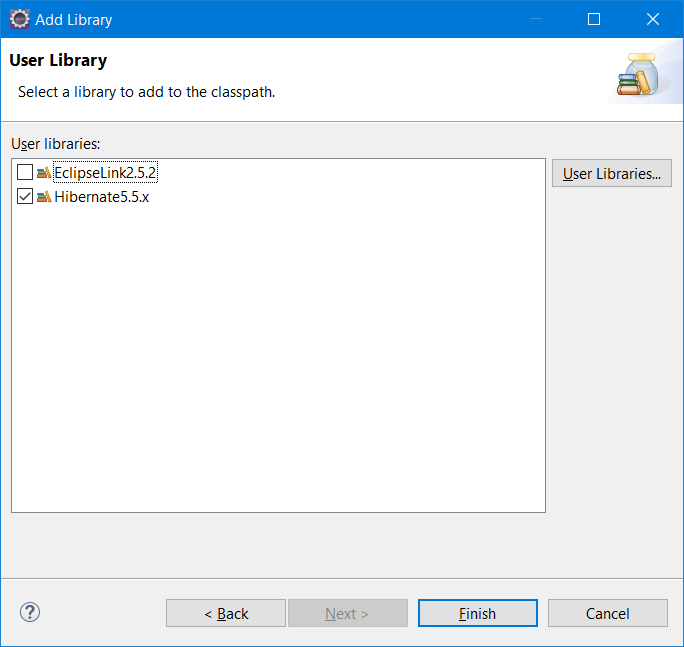
1. Click on “Apply & Close”



1. Add this newly created user library to our project.







Right click project -> New -> Other -> File-> xml-> hibernate.cfg.xml (Enter)

Paste the following content

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 5.3//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-5.3.dtd">

<hibernate-configuration>

<session-factory>

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

<property name=*"dialect"*>org.hibernate.dialect.Oracle9Dialect</property>

<property name=*"connection.url"*>jdbc:oracle:thin:@localhost:1521:xe</property>

<property name=*"connection.username"*>system</property>

<property name=*"connection.password"*>jtp</property>

<property name=*"connection.driver\_class"*>oracle.jdbc.driver.OracleDriver</property>

<mapping resource=*"employee.hbm.xml"* />

</session-factory>

</hibernate-configuration>

Updated hibernate.cfg.xml file

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 5.3//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-5.3.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hbm2ddl.auto"*>create</property>

<property name=*"dialect"*>org.hibernate.dialect.MySQLDialect</property>

<property name=*"connection.url"*>jdbc:mysql://localhost:3306/tcs</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>root</property>

<property name=*"connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>

<mapping class=*"com.tcs.hibernate.entity.Student"*/>

</session-factory>

</hibernate-configuration>

HQL – hibernate Query Lang (DB Independent Query)

Here we use the Entity Bean class name rather than using DB\_table name.