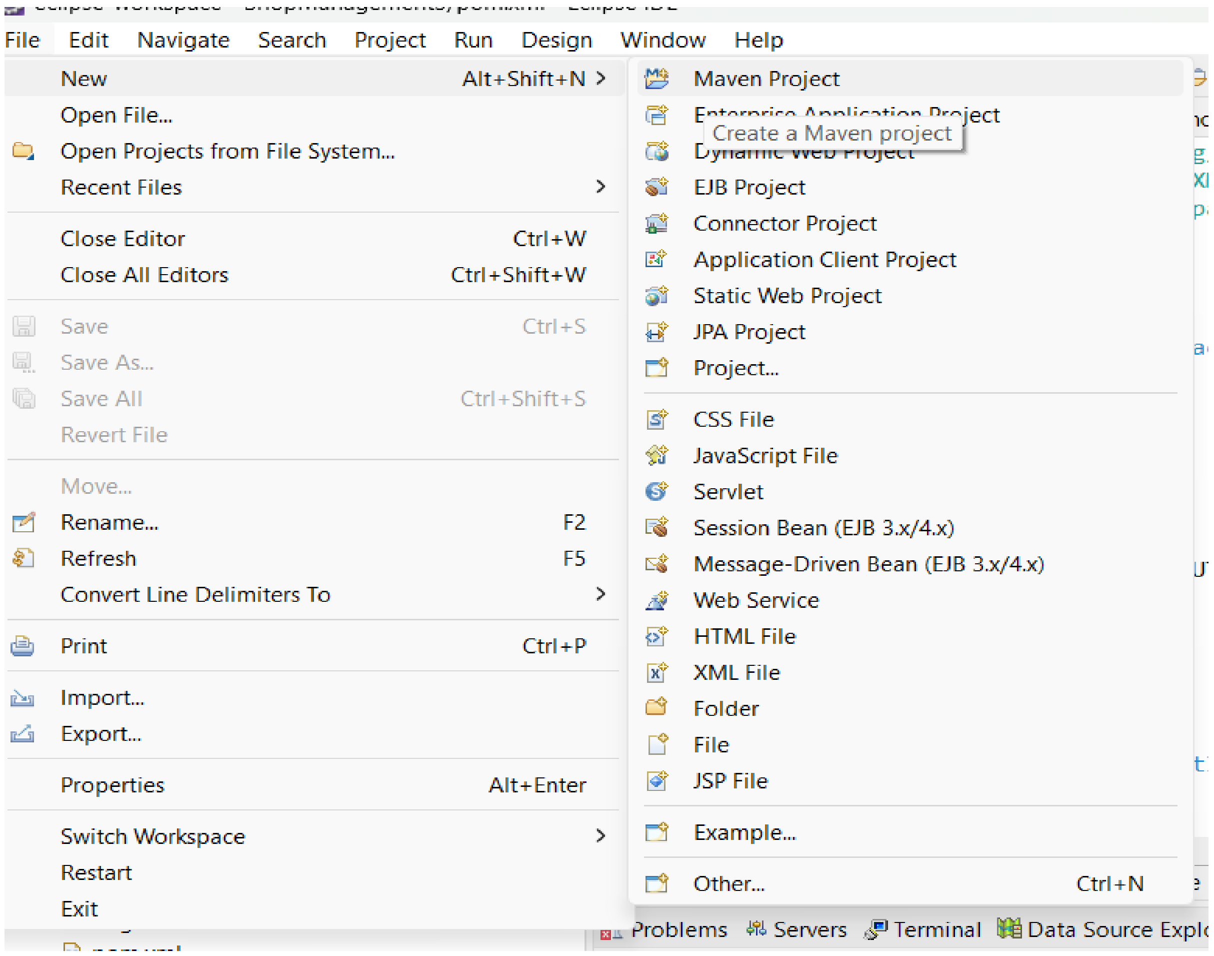
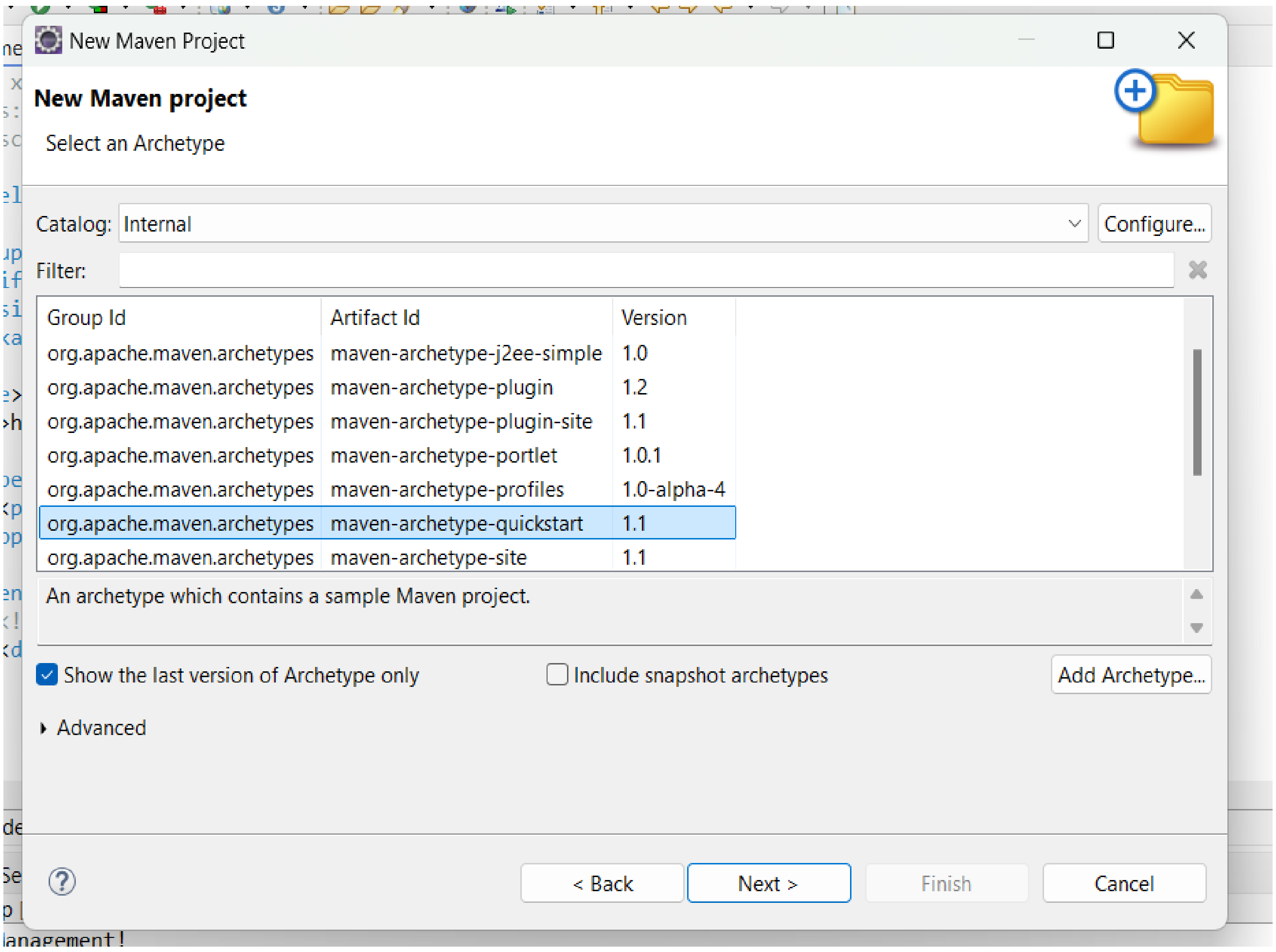
Assignment 03

Name-Pawar Rohit S.

Roll No-115 Div-B Batch-BT2





New Maven Project

New Maven project

Select an Archetype

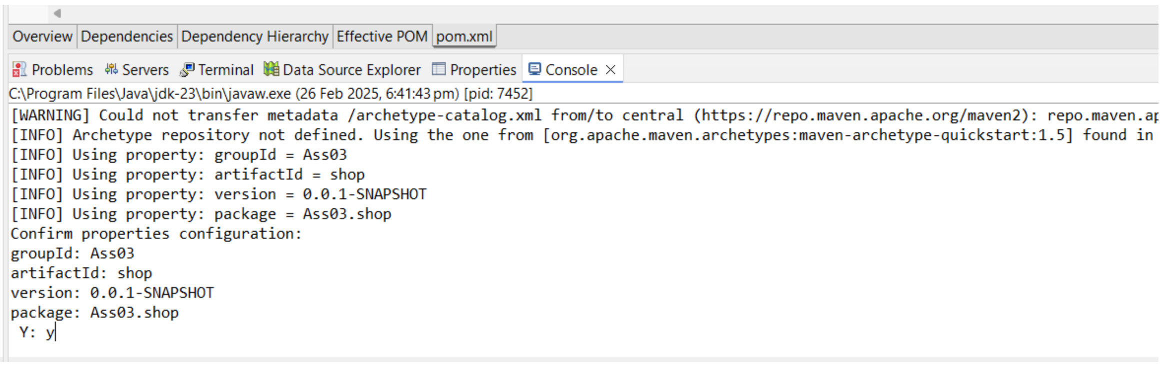
Catalog: Internal

Filter:

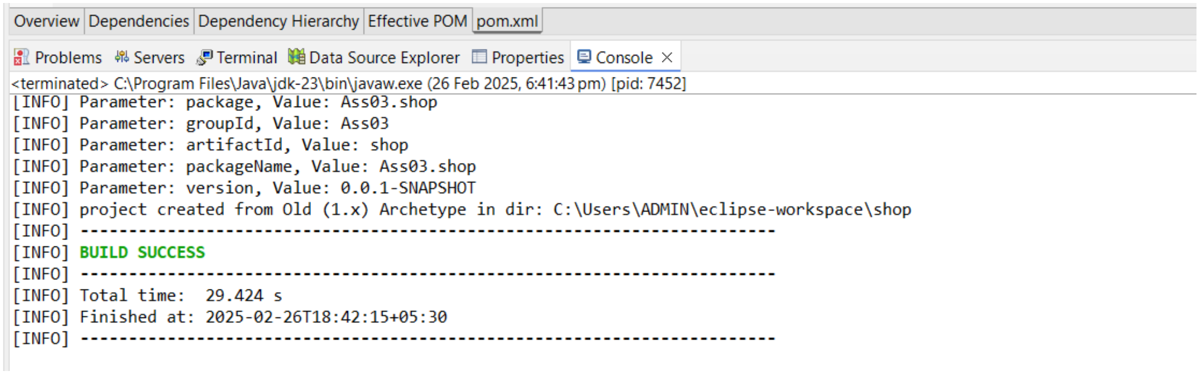
Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-j2ee-simple	1.0
org.apache.maven.archetypes	maven-archetype-plugin	1.2
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
org.apache.maven.archetypes	maven-archetype-site	1.1

An archetype which contains a sample Maven project.

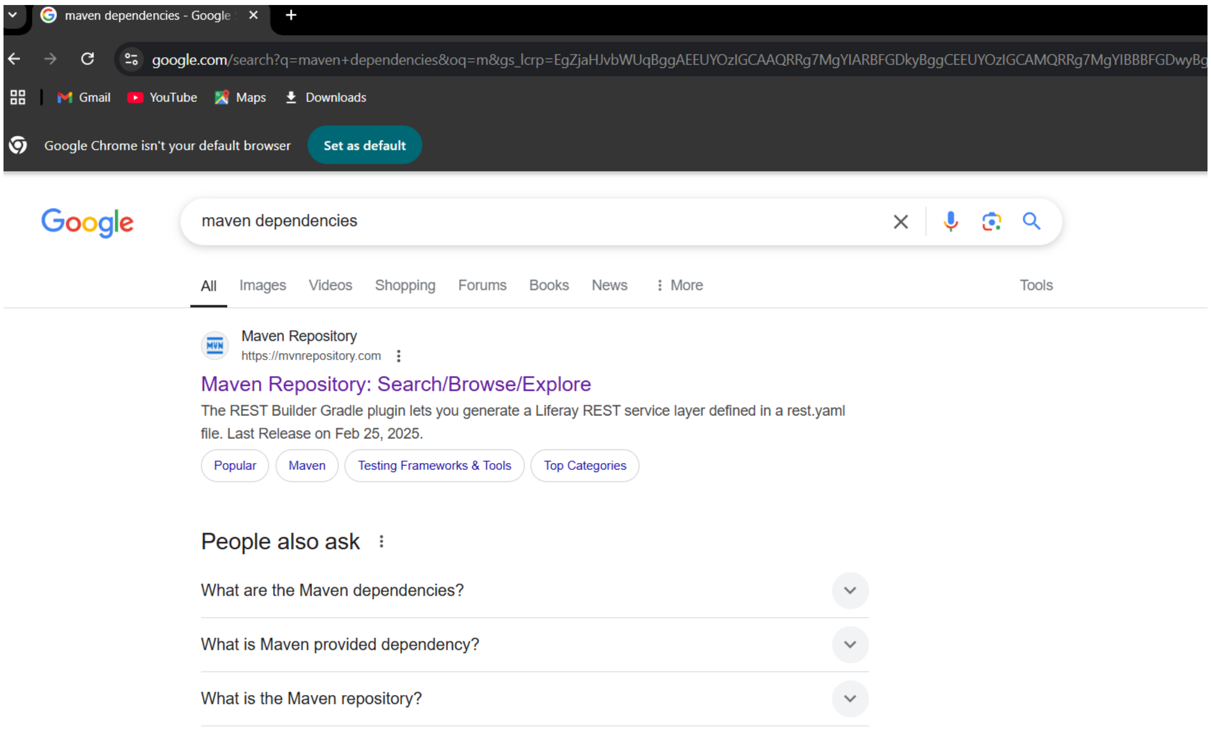
[✓] Show the last version of Archetype only
[ ] Include snapshot archetypes



```
C:\Program Files\Java\jdk-23\bin\javaw.exe (26 Feb 2025, 6:41:43 pm) [pid: 7452]
[WARNING] Could not transfer metadata /archetype-catalog.xml from/to central (https://repo.maven.apache.org/maven2): repo.maven.ap
[INFO] Archetype repository not defined. Using the one from [org.apache.maven.archetypes:maven-archetype-quickstart:1.5] found in
[INFO] Using property: groupId = Ass03
[INFO] Using property: artifactId = shop
[INFO] Using property: version = 0.0.1-SNAPSHOT
[INFO] Using property: package = Ass03.shop
Confirm properties configuration:
groupId: Ass03
artifactId: shop
version: 0.0.1-SNAPSHOT
package: Ass03.shop
 Y: y
```



```
<terminated> C:\Program Files\Java\jdk-23\bin\javaw.exe (26 Feb 2025, 6:41:43 pm) [pid: 7452]
[INFO] Parameter: package, Value: Ass03.shop
[INFO] Parameter: groupId, Value: Ass03
[INFO] Parameter: artifactId, Value: shop
[INFO] Parameter: packageName, Value: Ass03.shop
[INFO] Parameter: version, Value: 0.0.1-SNAPSHOT
[INFO] project created from Old (1.x) Archetype in dir: C:\Users\ADMIN\eclipse-workspace\shop
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  29.424 s
[INFO] Finished at: 2025-02-26T18:42:15+05:30
[INFO] ------------------------------------------------------------------------
```


maven dependencies - Google
google.com/search?q=maven+dependencies&oq=m&gs_lcrp=EgZjaHJvbWUqBggAEEUYOzIGCAAQRRg7MgYIARBFGDkyBggCEEUYOzIGCAMQRRg7MgYIBBBFGDwyBg
Gmail
YouTube
Maps
Downloads
Google Chrome isn't your default browser
Set as default
Google
maven dependencies
All
Images
Videos
Shopping
Forums
Books
News
More
Tools

Maven Repository
https://mvnrepository.com

Maven Repository: Search/Browse/Explore

The REST Builder Gradle plugin lets you generate a Liferay REST service layer defined in a rest.yaml file. Last Release on Feb 25, 2025.

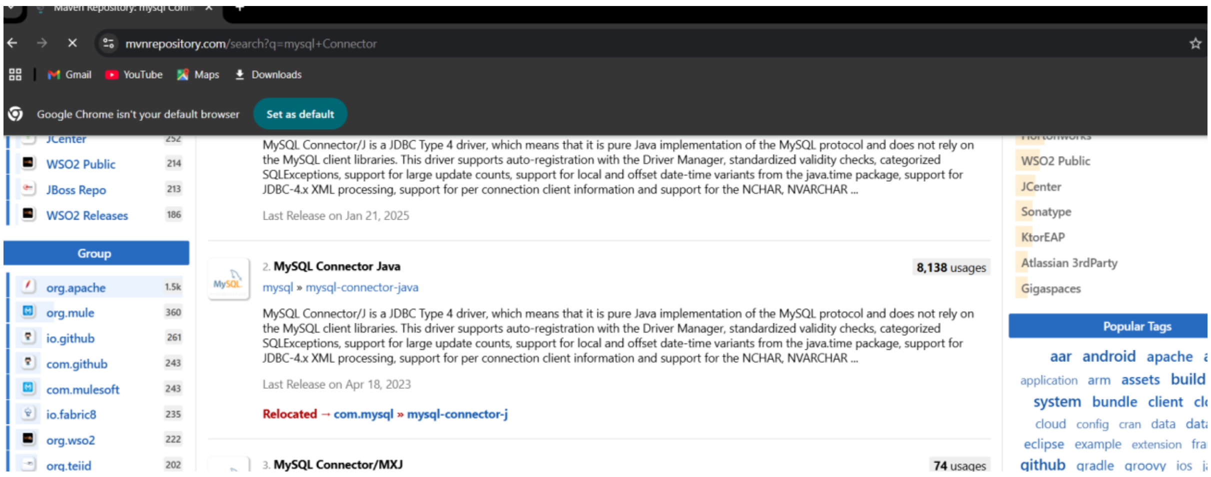
Popular | Maven | Testing Frameworks & Tools | Top Categories

People also ask

What are the Maven dependencies?

What is Maven provided dependency?

What is the Maven repository?


Maven Repository: mysql Conn
mvnrepository.com/search?q=mysql+Connector
Gmail
YouTube
Maps
Downloads
Google Chrome isn't your default browser
Set as default
Group
Popular Tags
aar android apache
application arm assets build
system bundle client
cloud config cran data
eclipse example extension
github gradle groovy ios

JCenter	252
WSO2 Public	214
JBoss Repo	213
WSO2 Releases	186

Group	
org.apache	1.5k
org.mule	360
io.github	261
com.github	243
com.mulesoft	243
io.fabric8	235
org.wso2	222
org.teiid	202

MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries. This driver supports auto-registration with the Driver Manager, standardized validity checks, categorized SQLExceptions, support for large update counts, support for local and offset date-time variants from the java.time package, support for JDBC-4.x XML processing, support for per connection client information and support for the NCHAR, NVARCHAR ...

Last Release on Jan 21, 2025

2. **MySQL Connector Java** — **8,138** usages

mysql » mysql-connector-java

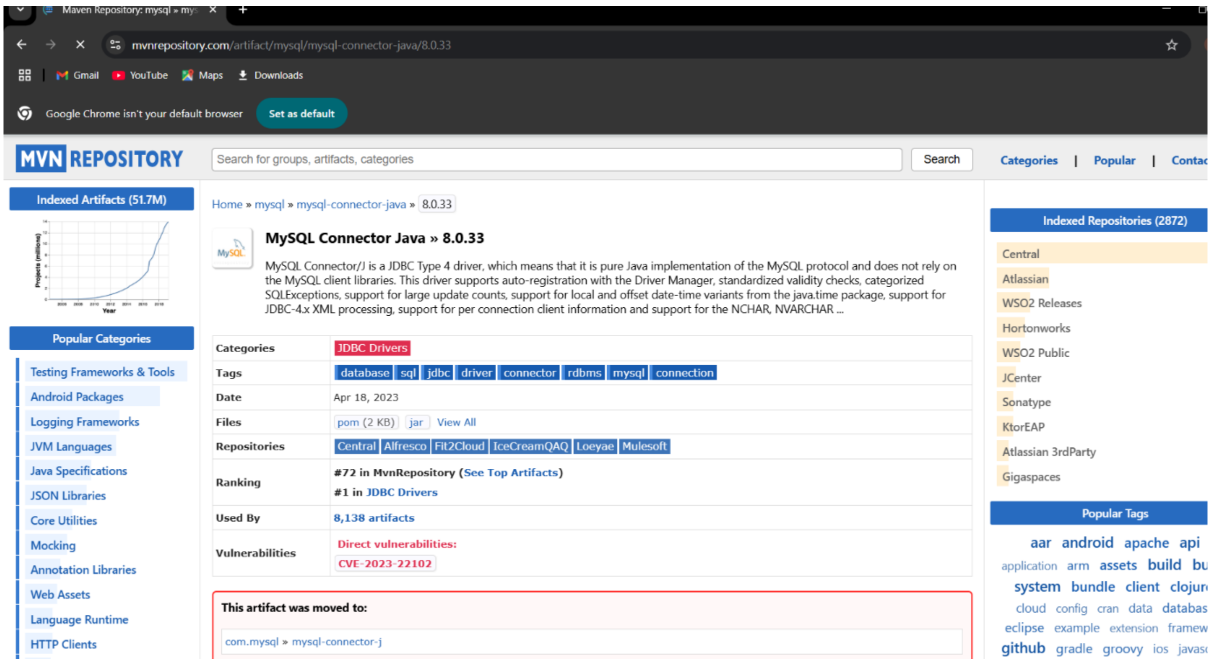
MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries. This driver supports auto-registration with the Driver Manager, standardized validity checks, categorized SQLExceptions, support for large update counts, support for local and offset date-time variants from the java.time package, support for JDBC-4.x XML processing, support for per connection client information and support for the NCHAR, NVARCHAR ...

Last Release on Apr 18, 2023

Relocated → com.mysql » mysql-connector-j

3. **MySQL Connector/MXJ** — **74** usages

WSO2 Public
JCenter
Sonatype
KtorEAP
Atlassian 3rdParty
Gigaspaces



Maven Repository: mysql » my
mvnrepository.com/artifact/mysql/mysql-connector-java/8.0.33
Gmail
YouTube
Maps
Downloads
Google Chrome isn't your default browser
Set as default
MVN REPOSITORY
Search for groups, artifacts, categories
Search
Categories | Popular | Contac
Indexed Artifacts (51.7M)
Popular Categories
Testing Frameworks & Tools
Android Packages
Logging Frameworks
JVM Languages
Java Specifications
JSON Libraries
Core Utilities
Mocking
Annotation Libraries
Web Assets
Language Runtime
HTTP Clients
Indexed Repositories (2872)
Central
Atlassian
WSO2 Releases
Hortonworks
WSO2 Public
JCenter
Sonatype
KtorEAP
Atlassian 3rdParty
Gigaspaces
Popular Tags
aar android apache api
application arm assets build bu
system bundle client cloju
cloud config cran data databas
eclipse example extension framew
github gradle groovy ios javas

Home » mysql » mysql-connector-java » 8.0.33

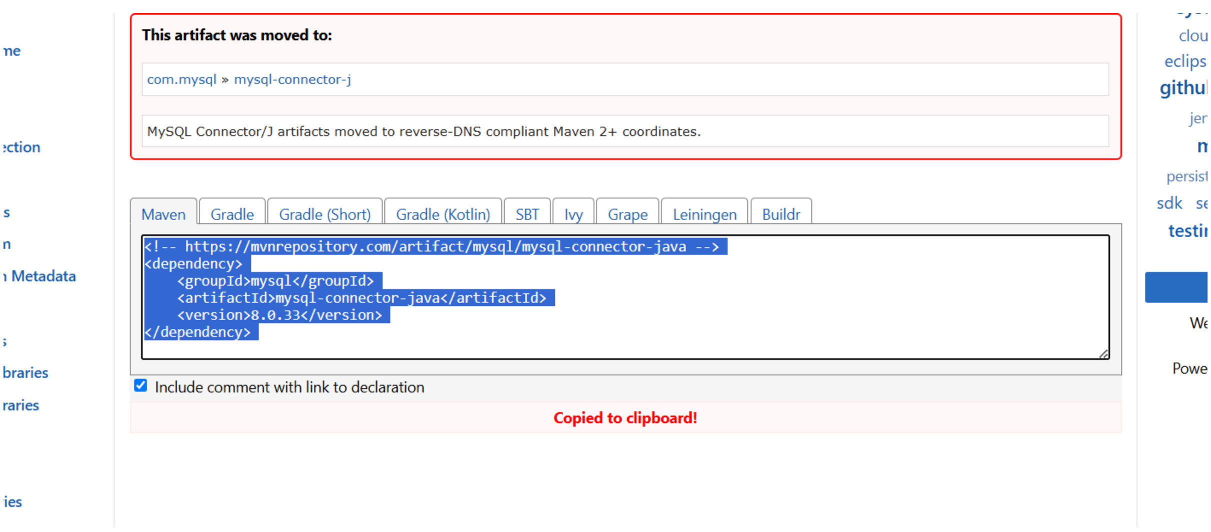
MySQL Connector Java » 8.0.33

MySQL Connector/J is a JDBC Type 4 driver, which means that it is pure Java implementation of the MySQL protocol and does not rely on the MySQL client libraries. This driver supports auto-registration with the Driver Manager, standardized validity checks, categorized SQLExceptions, support for large update counts, support for local and offset date-time variants from the java.time package, support for JDBC-4.x XML processing, support for per connection client information and support for the NCHAR, NVARCHAR ...

Categories	JDBC Drivers
Tags	database sql jdbc driver connector rdbms mysql connection
Date	Apr 18, 2023
Files	pom (2 KB) jar View All
Repositories	Central Alfresco Fit2Cloud IceCreamQAQ Loeyae Mulesoft
Ranking	#72 in MvnRepository (See Top Artifacts) #1 in JDBC Drivers
Used By	8,138 artifacts
Vulnerabilities	Direct vulnerabilities: CVE-2023-22102

This artifact was moved to:

com.mysql » mysql-connector-j



This artifact was moved to:

com.mysql » mysql-connector-j

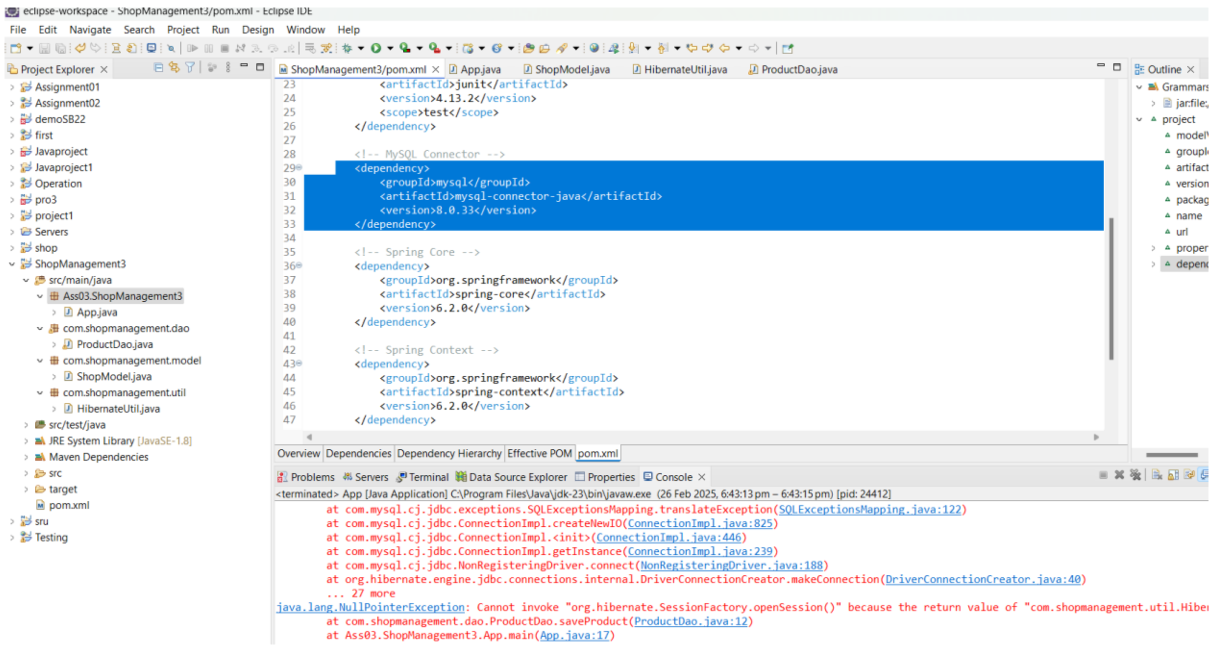
MySQL Connector/J artifacts moved to reverse-DNS compliant Maven 2+ coordinates.

Maven | Gradle | Gradle (Short) | Gradle (Kotlin) | SBT | Ivy | Grape | Leiningen | Buildr

```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
</dependency>
```

☑ Include comment with link to declaration

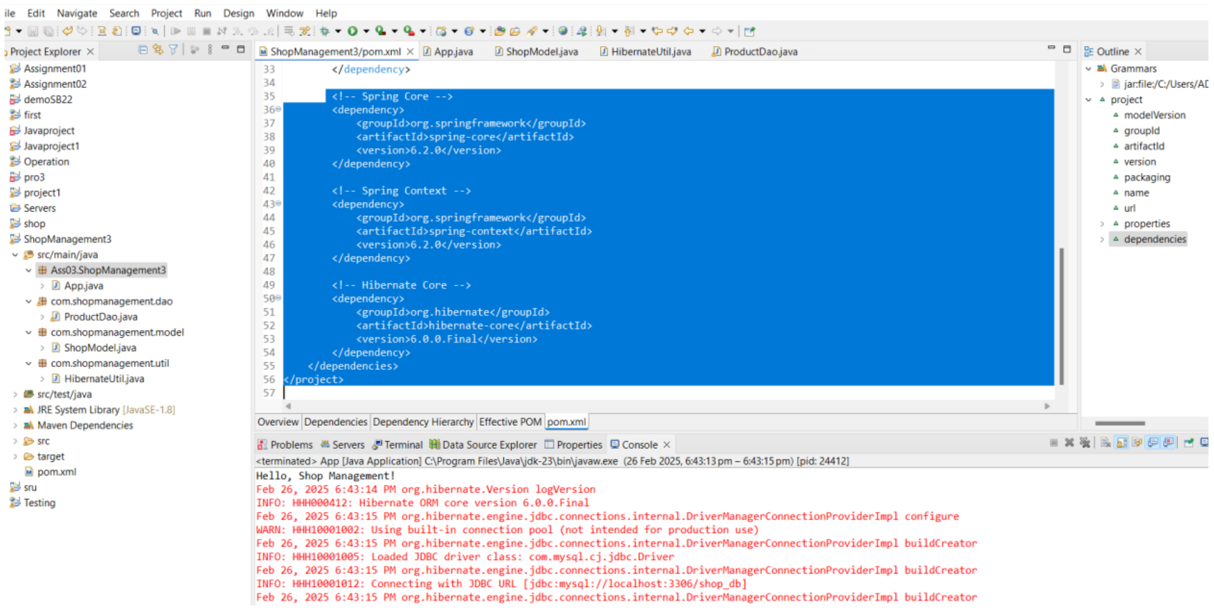
Copied to clipboard!



```
            <artifactId>junit</artifactId>
            <version>4.13.2</version>
            <scope>test</scope>
        </dependency>

        <!-- MySQL Connector -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>

        <!-- Spring Core -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>6.2.0</version>
        </dependency>

        <!-- Spring Context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>6.2.0</version>
        </dependency>
```

```
<terminated> App [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe  (26 Feb 2025, 6:43:13 pm – 6:43:15 pm) [pid: 24412]
	at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122)
	at com.mysql.cj.jdbc.ConnectionImpl.createNewIO(ConnectionImpl.java:825)
	at com.mysql.cj.jdbc.ConnectionImpl.<init>(ConnectionImpl.java:446)
	at com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:239)
	at com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:188)
	at org.hibernate.engine.jdbc.connections.internal.DriverConnectionCreator.makeConnection(DriverConnectionCreator.java:40)
	... 27 more
java.lang.NullPointerException: Cannot invoke "org.hibernate.SessionFactory.openSession()" because the return value of "com.shopmanagement.util.Hibe
	at com.shopmanagement.dao.ProductDao.saveProduct(ProductDao.java:12)
	at Ass03.ShopManagement3.App.main(App.java:17)
```

```
<dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
```

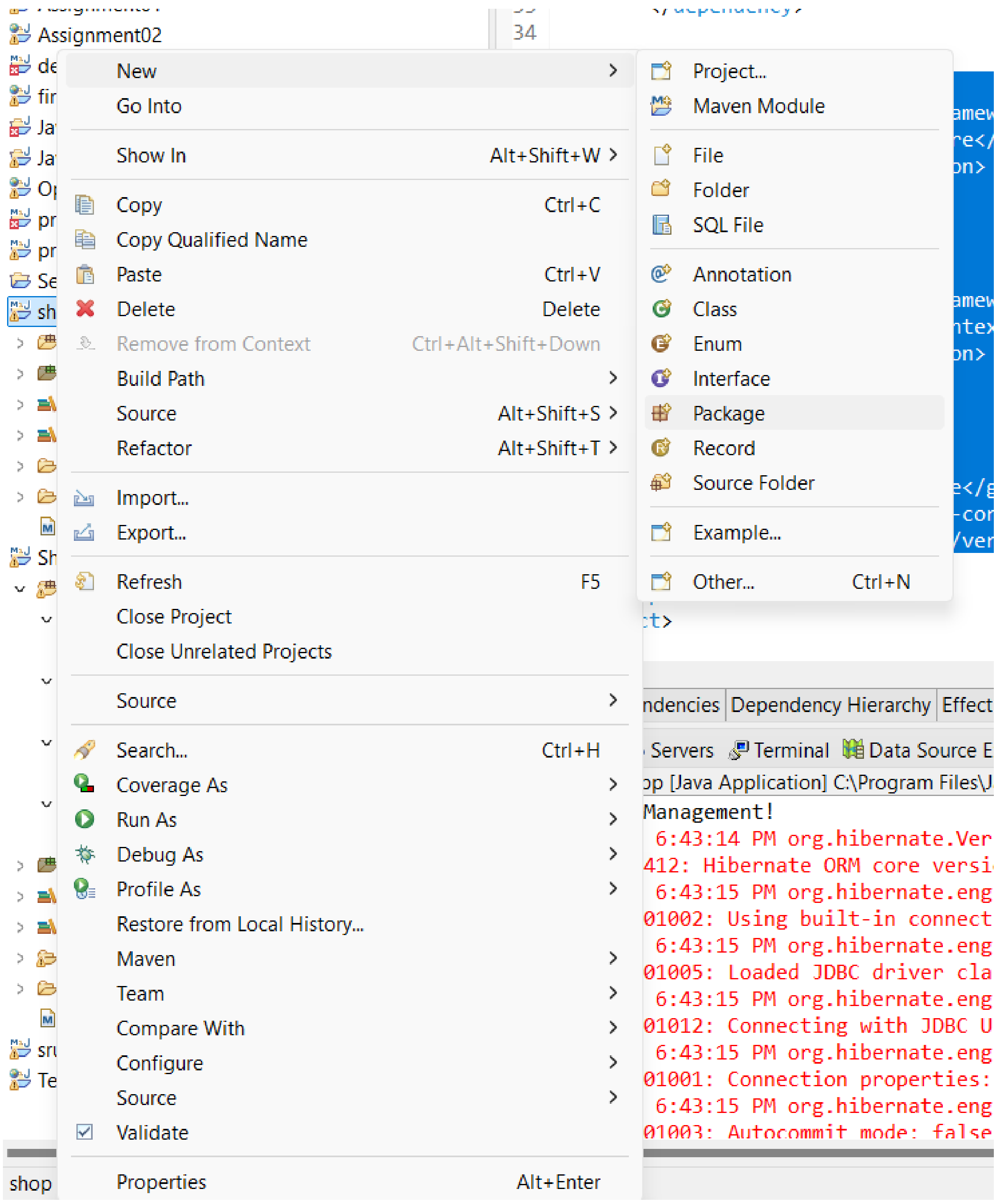


```
        </dependency>

        <!-- Spring Core -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>6.2.0</version>
        </dependency>

        <!-- Spring Context -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>6.2.0</version>
        </dependency>

        <!-- Hibernate Core -->
        <dependency>
            <groupId>org.hibernate</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>6.0.0.Final</version>
        </dependency>
    </dependencies>
</project>
```

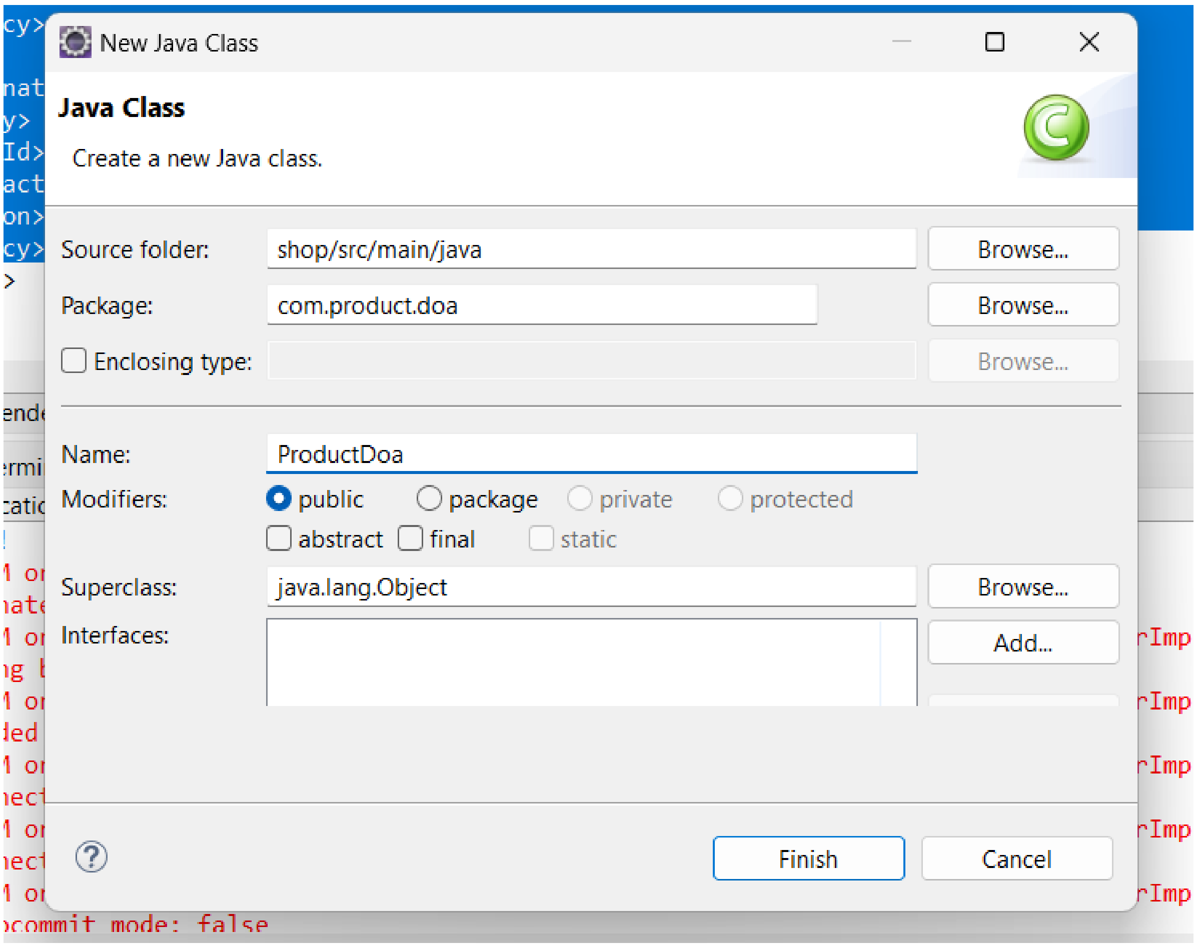
```
<terminated> App [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe  (26 Feb 2025, 6:43:13 pm – 6:43:15 pm) [pid: 24412]
Hello, Shop Management!
Feb 26, 2025 6:43:14 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version 6.0.0.Final
Feb 26, 2025 6:43:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using built-in connection pool (not intended for production use)
Feb 26, 2025 6:43:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: Loaded JDBC driver class: com.mysql.cj.jdbc.Driver
Feb 26, 2025 6:43:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001012: Connecting with JDBC URL [jdbc:mysql://localhost:3306/shop_db]
Feb 26, 2025 6:43:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
```

```
<dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
```

```xml
    <version>6.2.0</version>
</dependency>

<!-- Spring Context -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>6.2.0</version>
</dependency>

<!-- Hibernate Core -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>6.0.0.Final</version>
</dependency>
```



```
op [Java Application] C:\Program Files\J
Management!
 6:43:14 PM org.hibernate.Ver
412: Hibernate ORM core versi
 6:43:15 PM org.hibernate.eng
01002: Using built-in connect
 6:43:15 PM org.hibernate.eng
01005: Loaded JDBC driver cla
 6:43:15 PM org.hibernate.eng
01012: Connecting with JDBC U
 6:43:15 PM org.hibernate.eng
01001: Connection properties:
 6:43:15 PM org.hibernate.eng
01003: Autocommit mode: false
```


New Java Class

Java Class

Create a new Java class.

Source folder: shop/src/main/java

Package: com.product.doa

Enclosing type: [ ]

Name: ProductDoa

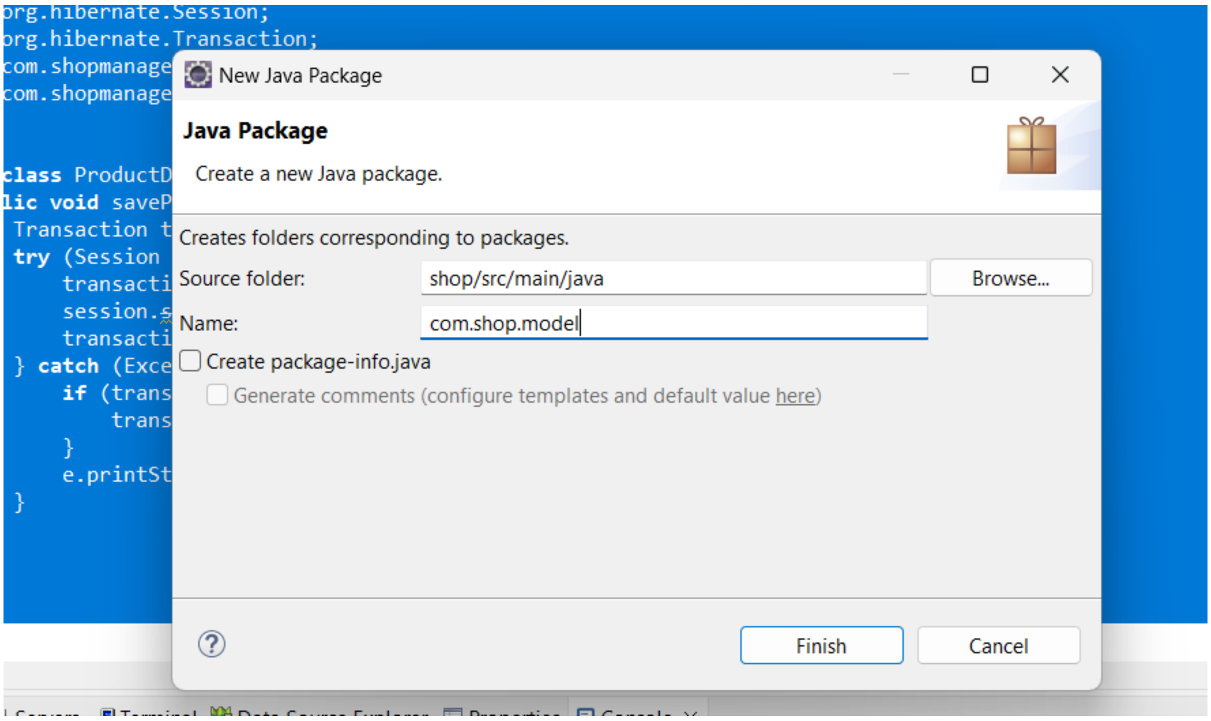
Modifiers: [✓ public] package private protected

[ ] abstract [ ] final [ ] static

Superclass: java.lang.Object

Interfaces:

```
package com.shopmanagement.dao;

import org.hibernate.Session;
import org.hibernate.Transaction;
import com.shopmanagement.util.HibernateUtil;
import com.shopmanagement.model.ShopModel;


public class ProductDao {
        public void saveProduct(ShopModel product) {
         Transaction transaction = null;
         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
             transaction = session.beginTransaction();
             session.~~save~~(product);
             transaction.commit();
         } catch (Exception e) {
             if (transaction != null) {
                 transaction.rollback();
             }
             e.printStackTrace();
         }
     }


}
```


New Java Package

Java Package

Create a new Java package.

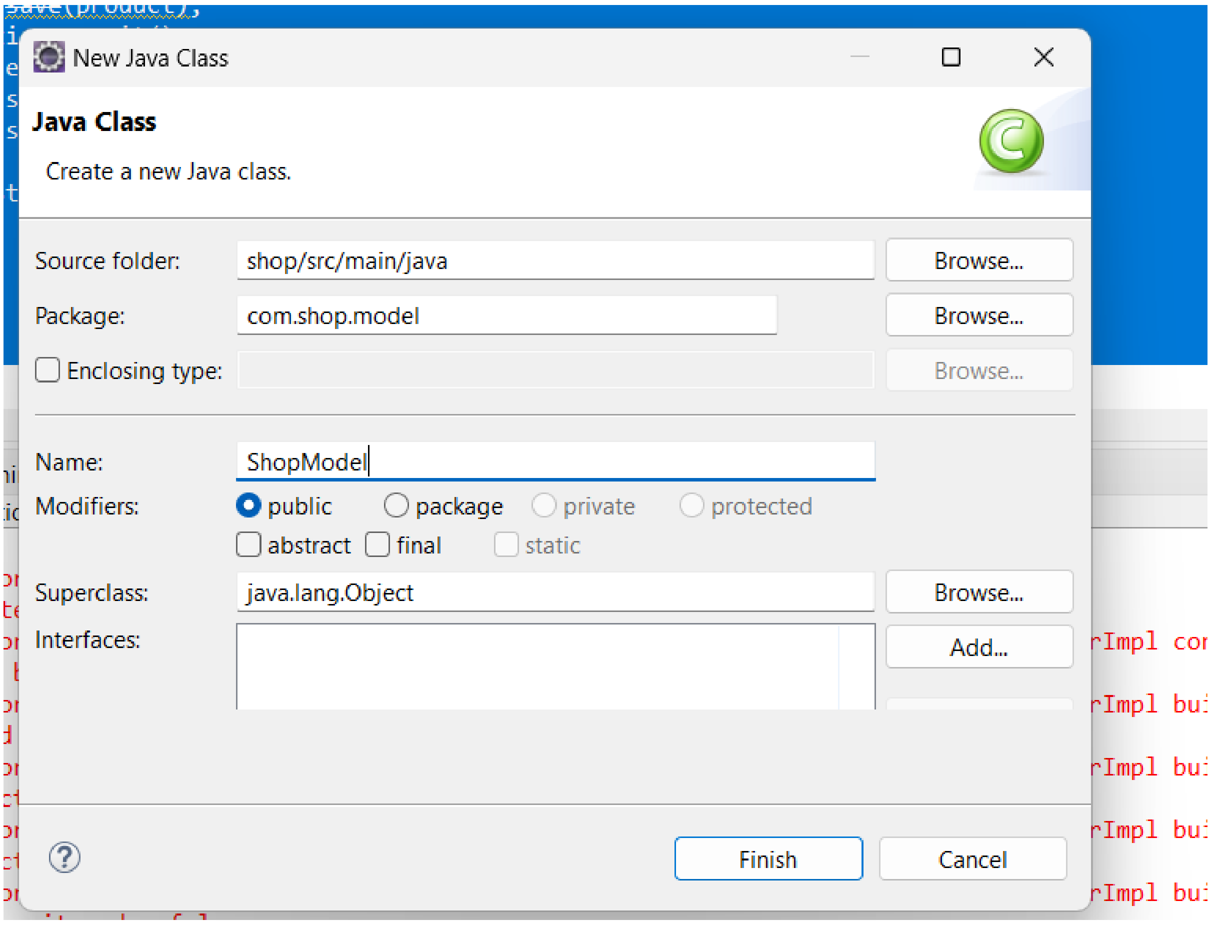
Creates folders corresponding to packages.

Source folder: shop/src/main/java

Name: com.shop.model

☐ Create package-info.java

☐ Generate comments (configure templates and default value here)


New Java Class

Java Class

Create a new Java class.

Source folder: shop/src/main/java

Package: com.shop.model

☐ Enclosing type:

Name: ShopModel

Modifiers: [✓ public] ○ package ○ private ○ protected

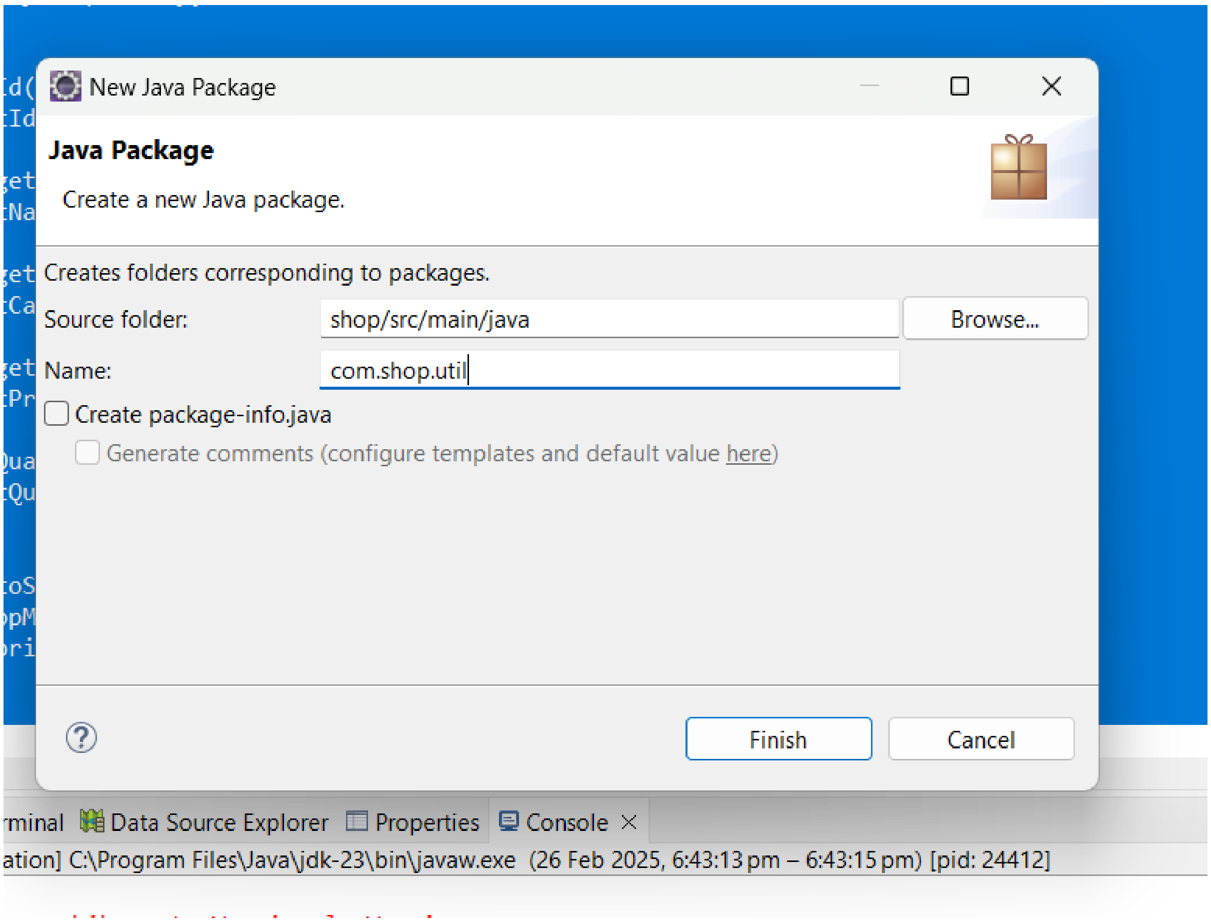
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object

Interfaces:

```
package com.shopmanagement.model;

import jakarta.persistence.*;
```

```
@Entity
@Table(name = "shop_products")
public class ShopModel {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    private String name;
    private String category;
    private double price;
    private int quantity;

    public ShopModel() {}

    public ShopModel(String name, String category, double price, int quantity) {
        this.name = name;
        this.category = category;
        this.price = price;
        this.quantity = quantity;
    }

    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 getCategory() { return category; }
    public void setCategory(String category) { this.category = category; }

    public double getPrice() { return price; }
    public void setPrice(double price) { this.price = price; }

    public int getQuantity() { return quantity; }
    public void setQuantity(int quantity) { this.quantity = quantity; }

    @Override
    public String toString() {
        return "ShopModel [id=" + id + ", name=" + name + ", category=" + category
+
               ", price=" + price + ", quantity=" + quantity + "]";
    }
}
```



New Java Package

Java Package

Create a new Java package.

Creates folders corresponding to packages.

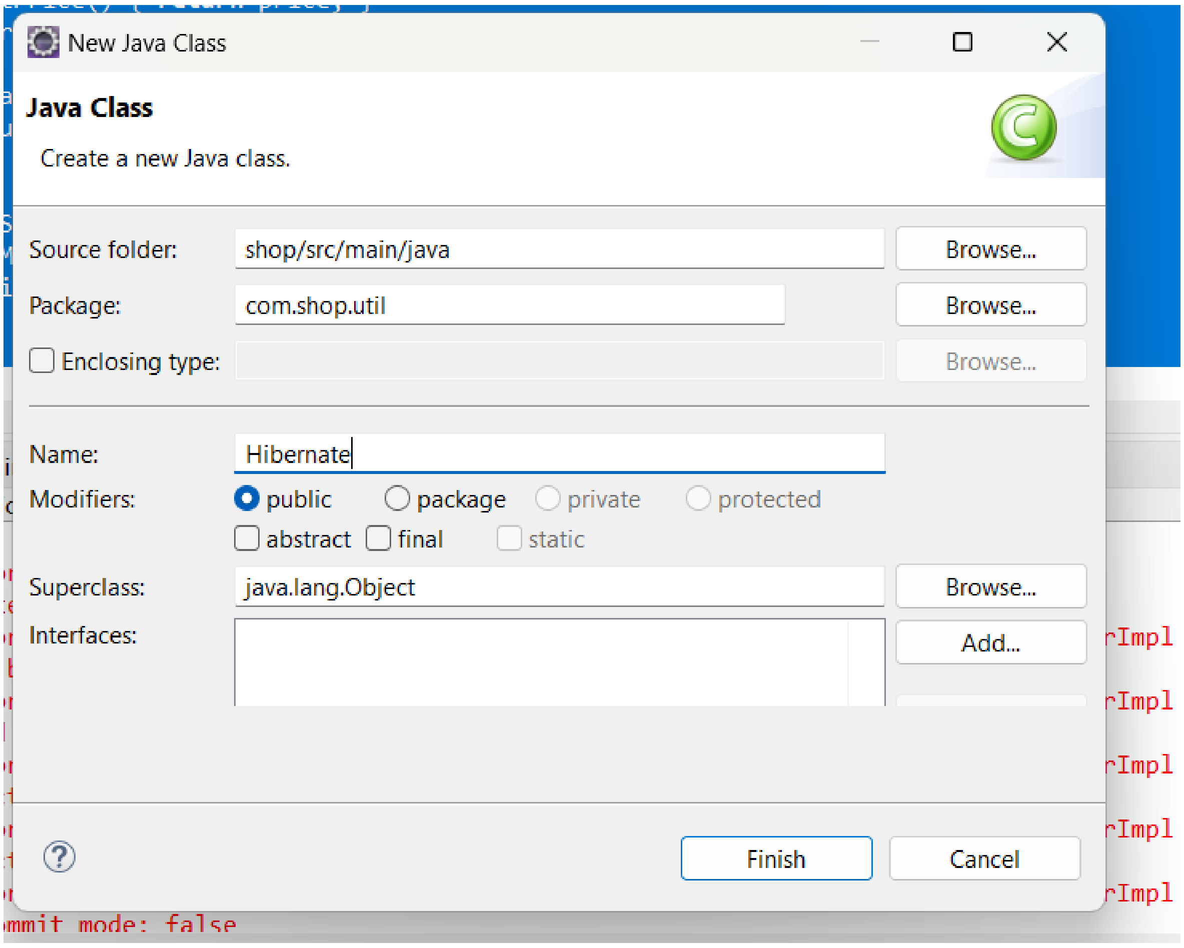
Source folder: shop/src/main/java

Name: com.shop.util

☐ Create package-info.java

☐ Generate comments (configure templates and default value here)

```
ation] C:\Program Files\Java\jdk-23\bin\javaw.exe  (26 Feb 2025, 6:43:13 pm – 6:43:15 pm) [pid: 24412]
```


New Java Class

Java Class

Create a new Java class.

Source folder: shop/src/main/java

Package: com.shop.util

Enclosing type:

Name: Hibernate

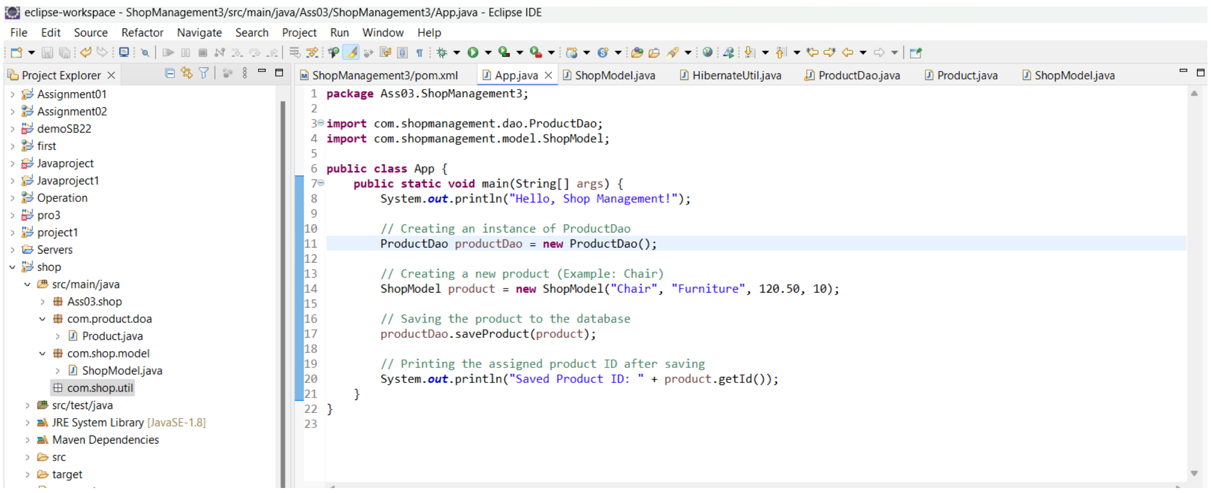
Modifiers: [✓ public] package private protected

abstract final static

Superclass: java.lang.Object

Interfaces:

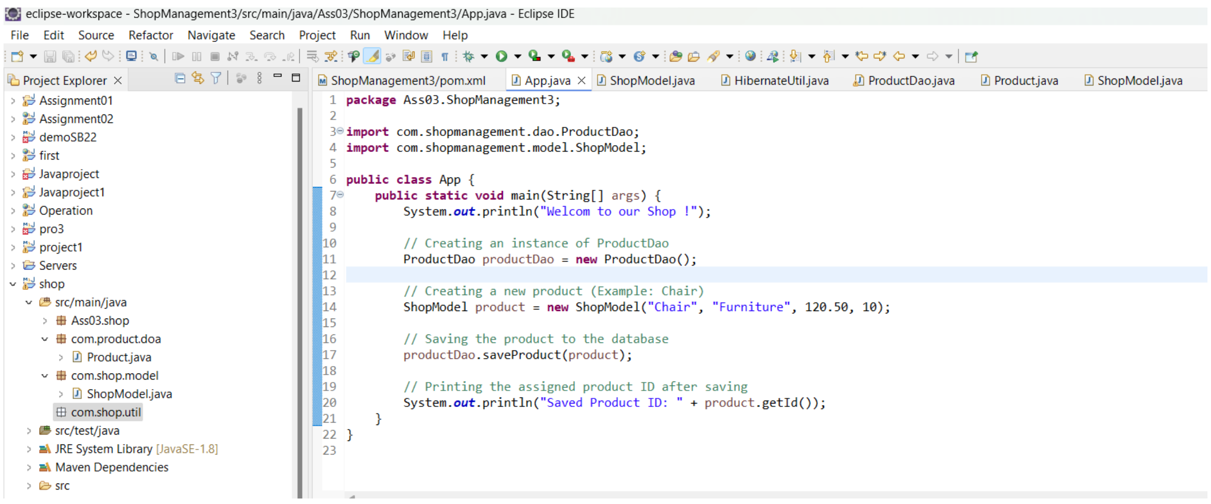
```java
package com.shopmanagement.util;

import java.util.Properties;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.cfg.Environment;
import org.hibernate.service.ServiceRegistry;
import com.shopmanagement.model.ShopModel;

public class HibernateUtil {
    private static SessionFactory sessionFactory;

    public static SessionFactory getSessionFactory() {
        if (sessionFactory == null) {
            try {
                Configuration configuration = new Configuration();

                Properties settings = new Properties();
                settings.put(Environment.DRIVER, "com.mysql.cj.jdbc.Driver");
                settings.put(Environment.URL,
"jdbc:mysql://localhost:3306/shop_db");
                settings.put(Environment.USER, "root");
                settings.put(Environment.PASS, "");
                settings.put(Environment.DIALECT,
"org.hibernate.dialect.MySQL8Dialect");
                settings.put(Environment.SHOW_SQL, "true");
                settings.put(Environment.CURRENT_SESSION_CONTEXT_CLASS, "thread");
                settings.put(Environment.HBM2DDL_AUTO, "update");

                configuration.setProperties(settings);
                configuration.addAnnotatedClass(ShopModel.class);

                ServiceRegistry serviceRegistry = new
StandardServiceRegistryBuilder()
                        .applySettings(configuration.getProperties()).build();
                sessionFactory =
configuration.buildSessionFactory(serviceRegistry);

            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return sessionFactory;
    }
}
```



```
package Ass03.ShopManagement3;

import com.shopmanagement.dao.ProductDao;
import com.shopmanagement.model.ShopModel;

public class App {
    public static void main(String[] args) {
        System.out.println("Hello, Shop Management!");

        // Creating an instance of ProductDao
        ProductDao productDao = new ProductDao();

        // Creating a new product (Example: Chair)
        ShopModel product = new ShopModel("Chair", "Furniture", 120.50, 10);

        // Saving the product to the database
        productDao.saveProduct(product);

        // Printing the assigned product ID after saving
        System.out.println("Saved Product ID: " + product.getId());
    }
}
```

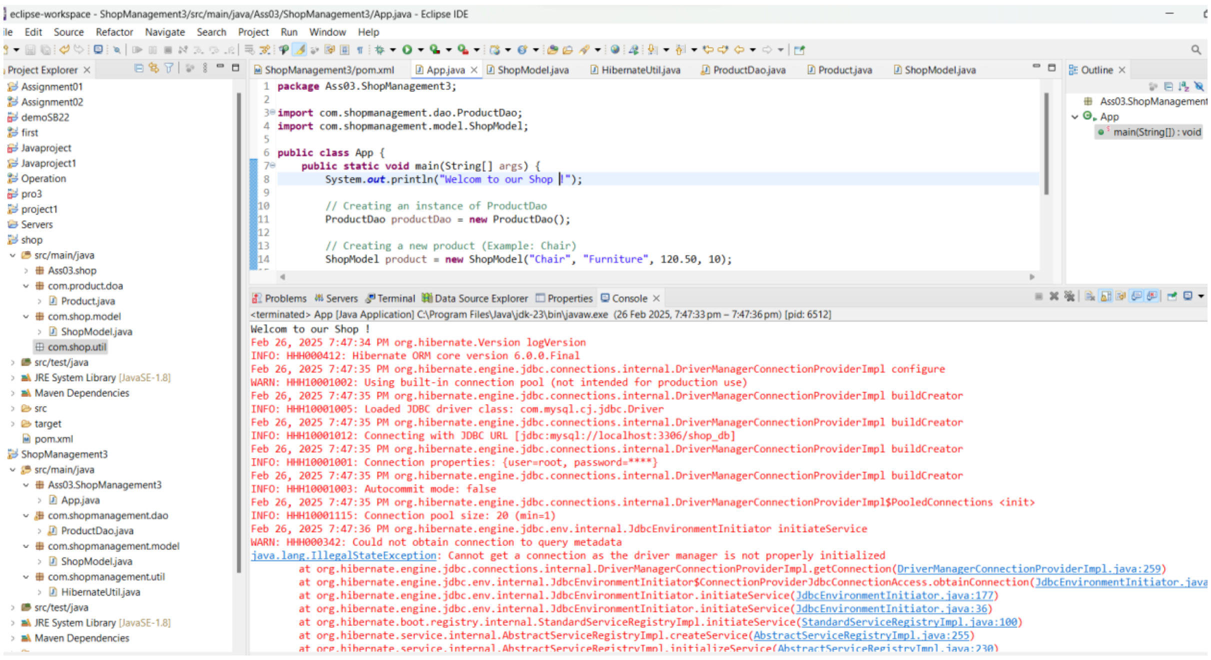
```
package Ass03.ShopManagement3;

import com.shopmanagement.dao.ProductDao;
import com.shopmanagement.model.ShopModel;

public class App {
    public static void main(String[] args) {
        System.out.println("Hello, Shop Management!");

        // Creating an instance of ProductDao
        ProductDao productDao = new ProductDao();

        // Creating a new product (Example: Chair)
        ShopModel product = new ShopModel("Chair", "Furniture", 120.50, 10);
```

```java
        // Saving the product to the database
        productDao.saveProduct(product);

        // Printing the assigned product ID after saving
        System.out.println("Saved Product ID: " + product.getId());
    }
}
```



```java
package Ass03.ShopManagement3;

import com.shopmanagement.dao.ProductDao;
import com.shopmanagement.model.ShopModel;

public class App {
    public static void main(String[] args) {
        System.out.println("Welcom to our Shop !");

        // Creating an instance of ProductDao
        ProductDao productDao = new ProductDao();

        // Creating a new product (Example: Chair)
        ShopModel product = new ShopModel("Chair", "Furniture", 120.50, 10);

        // Saving the product to the database
        productDao.saveProduct(product);

        // Printing the assigned product ID after saving
        System.out.println("Saved Product ID: " + product.getId());
    }
}
```

Output:



```java
package Ass03.ShopManagement3;

import com.shopmanagement.dao.ProductDao;
import com.shopmanagement.model.ShopModel;

public class App {
    public static void main(String[] args) {
        System.out.println("Welcom to our Shop !");

        // Creating an instance of ProductDao
        ProductDao productDao = new ProductDao();

        // Creating a new product (Example: Chair)
        ShopModel product = new ShopModel("Chair", "Furniture", 120.50, 10);
```

```
<terminated> App [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (26 Feb 2025, 7:47:33 pm – 7:47:36 pm) [pid: 6512]
Welcom to our Shop !
Feb 26, 2025 7:47:34 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version 6.0.0.Final
Feb 26, 2025 7:47:35 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using built-in connection pool (not intended for production use)
Feb 26, 2025 7:47:35 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: Loaded JDBC driver class: com.mysql.cj.jdbc.Driver
Feb 26, 2025 7:47:35 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001012: Connecting with JDBC URL [jdbc:mysql://localhost:3306/shop_db]
Feb 26, 2025 7:47:35 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {user=root, password=****}
Feb 26, 2025 7:47:35 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
Feb 26, 2025 7:47:35 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH10001115: Connection pool size: 20 (min=1)
Feb 26, 2025 7:47:36 PM org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator initiateService
WARN: HHH000342: Could not obtain connection to query metadata
java.lang.IllegalStateException: Cannot get a connection as the driver manager is not properly initialized
	at org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl.getConnection(DriverManagerConnectionProviderImpl.java:259)
	at org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess.obtainConnection(JdbcEnvironmentInitiator.java
	at org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator.initiateService(JdbcEnvironmentInitiator.java:177)
	at org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator.initiateService(JdbcEnvironmentInitiator.java:36)
	at org.hibernate.boot.registry.internal.StandardServiceRegistryImpl.initiateService(StandardServiceRegistryImpl.java:100)
	at org.hibernate.service.internal.AbstractServiceRegistryImpl.createService(AbstractServiceRegistryImpl.java:255)
	at org.hibernate.service.internal.AbstractServiceRegistryImpl.initializeService(AbstractServiceRegistryImpl.java:230)
```

INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect

```
WARN: HHH000206: MySQL5Dialect has been deprecated; use org.hibernate.dialect.MySQLDialect
instead

INFO: HHH10001051: Connection obtained from JdbcConnectionAccess


Hibernate: create table Shop1 (productId integer not null auto_increment, 'Product Category'
varchar(255), 'Product Name' varchar(255), primary key (productId)) engine=MyISAM

Hibernate: insert into Shop1 ('Product Category', 'Product Name') values (?, ?)

1
```