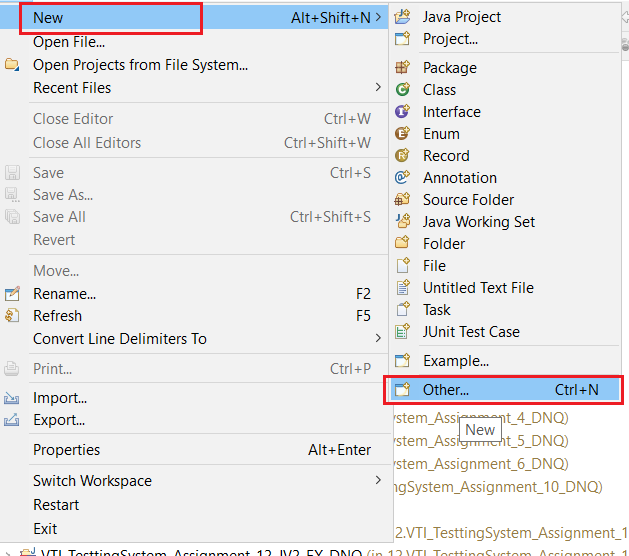
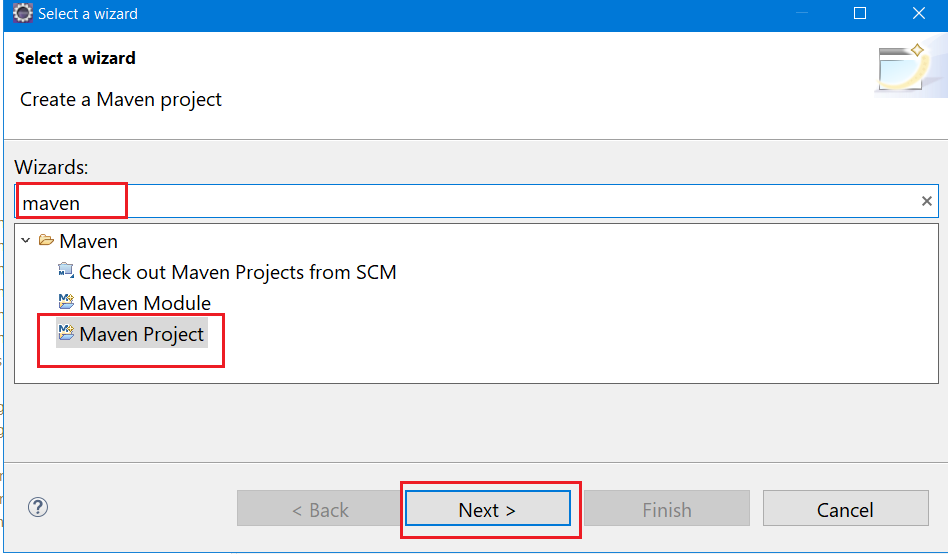
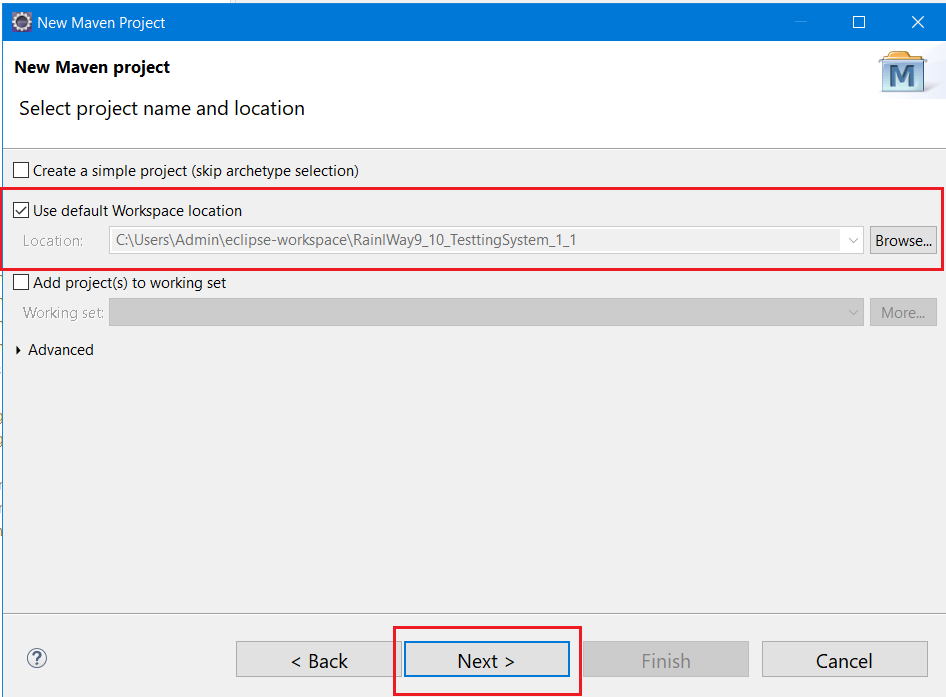
Tạo ứng dụng test kết nối tới DB TestingSystem bằng Hibernate.

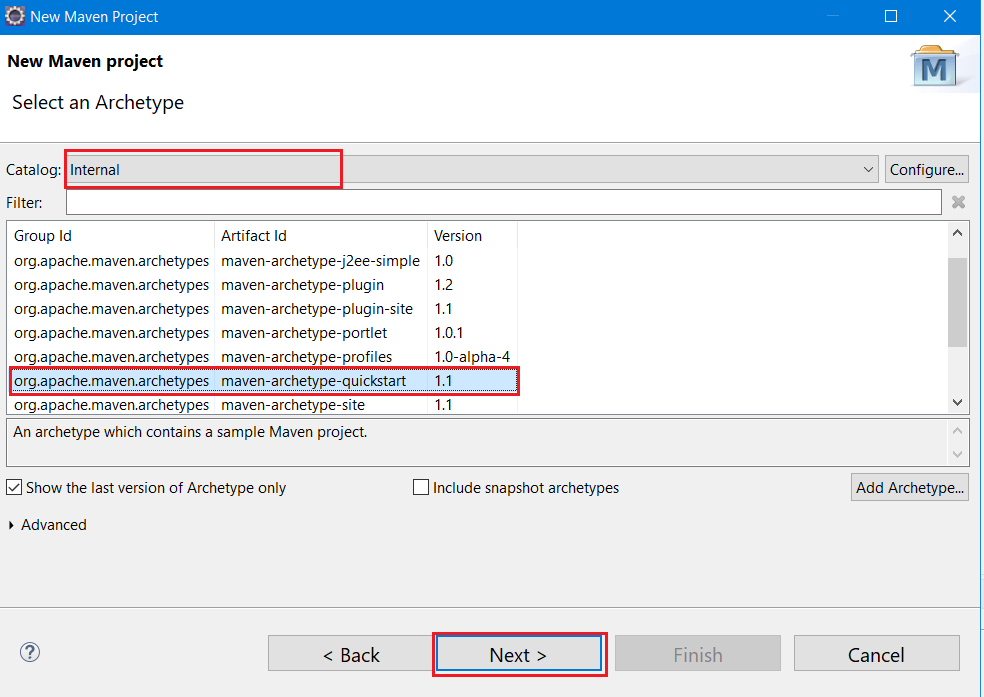


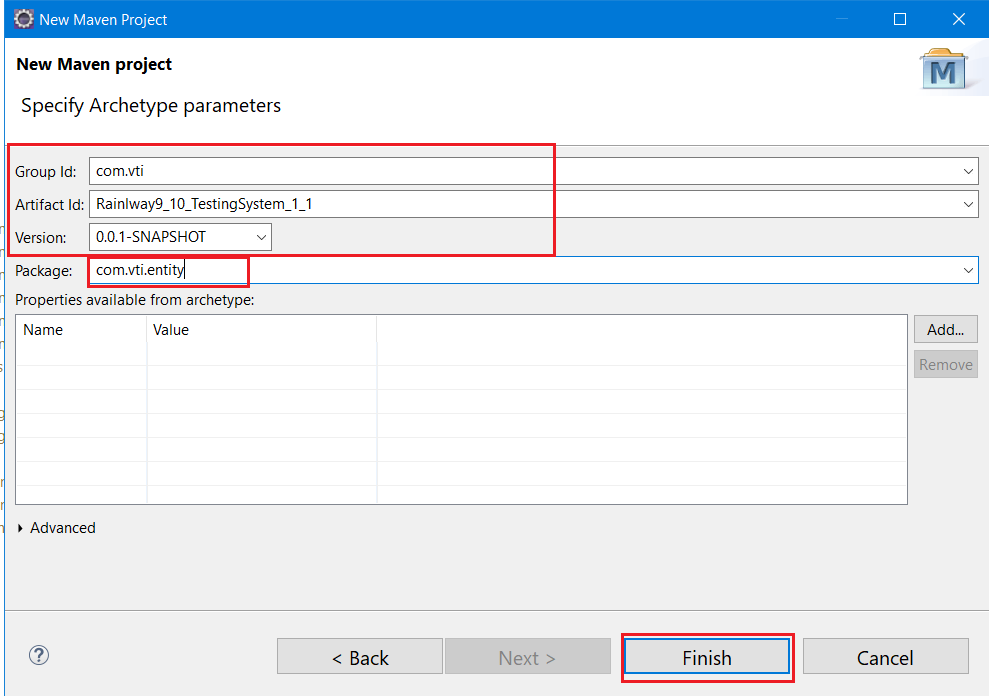
# **Tạo** Project maven**: RainlWay9\_10\_TesttingSystem\_1\_1**

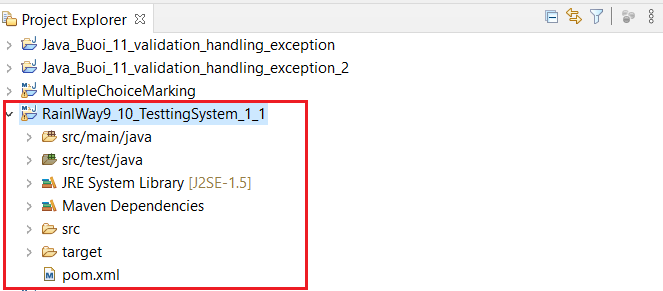






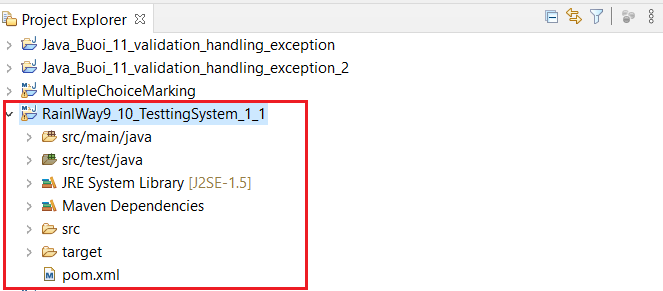






# **Tạo** khung **chương trình:**

* Tạo mới project: RainlWay9\_10\_TesttingSystem\_1\_1



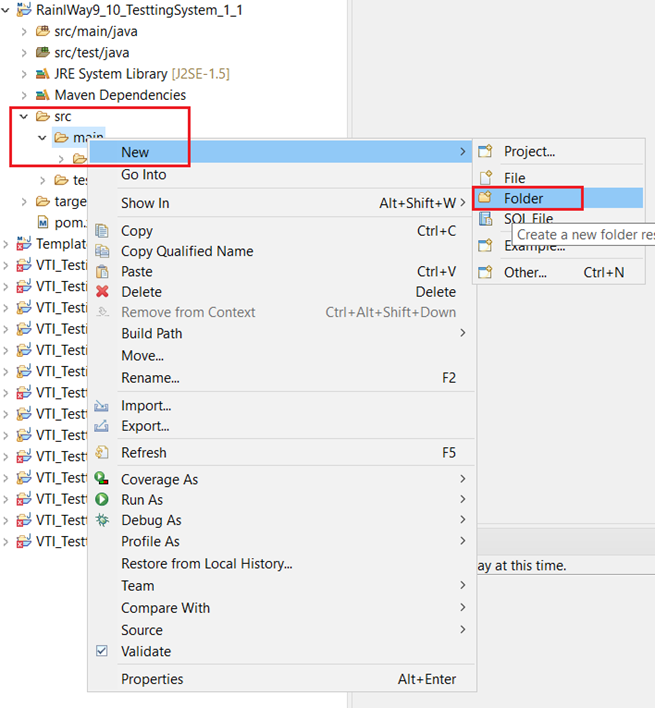
# **Thêm các cấu hình cho file: pom.xml**

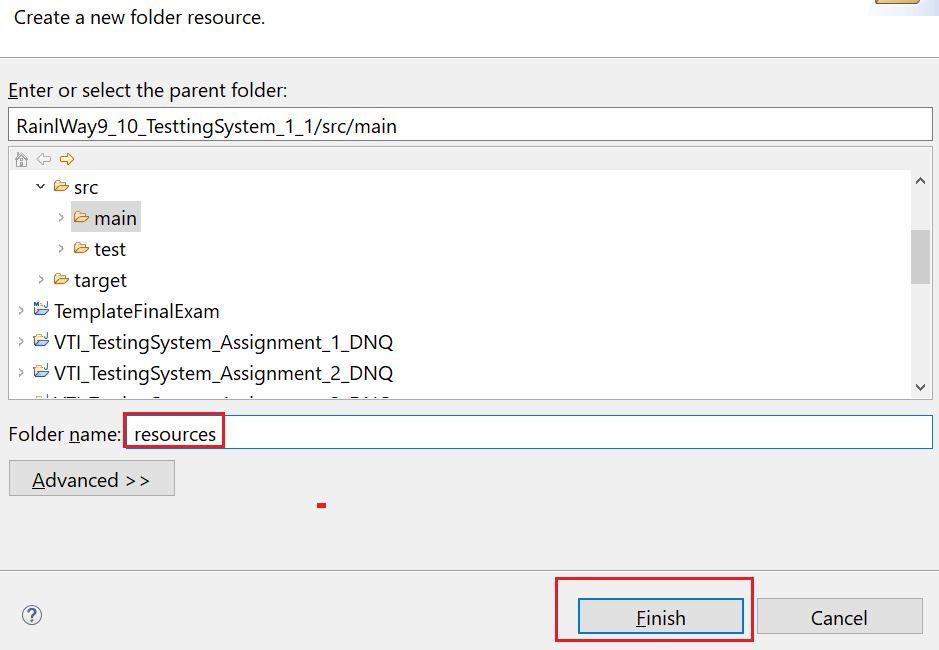
|  |
| --- |
| <dependency>  <groupId>org.hibernate</groupId>  <artifactId>hibernate-core</artifactId>  <version>5.4.17.Final</version>  </dependency>  <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  <version>8.0.20</version>  </dependency> |

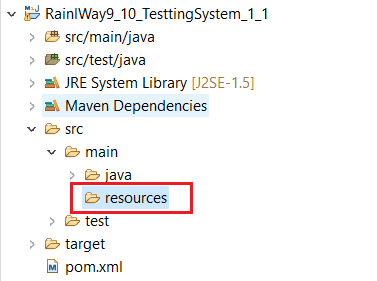


# **Thêm các cấu hình cho file: hibernate.cfg.xml trong resources**

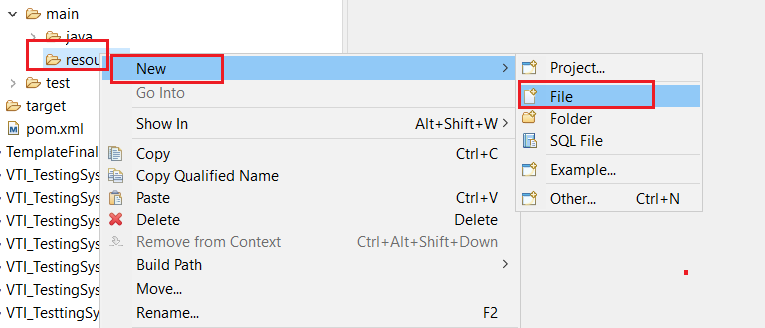
* Tạo mới folder resources, đường dẫn **src.main.resources**. chú ý viết đúng tên folder.

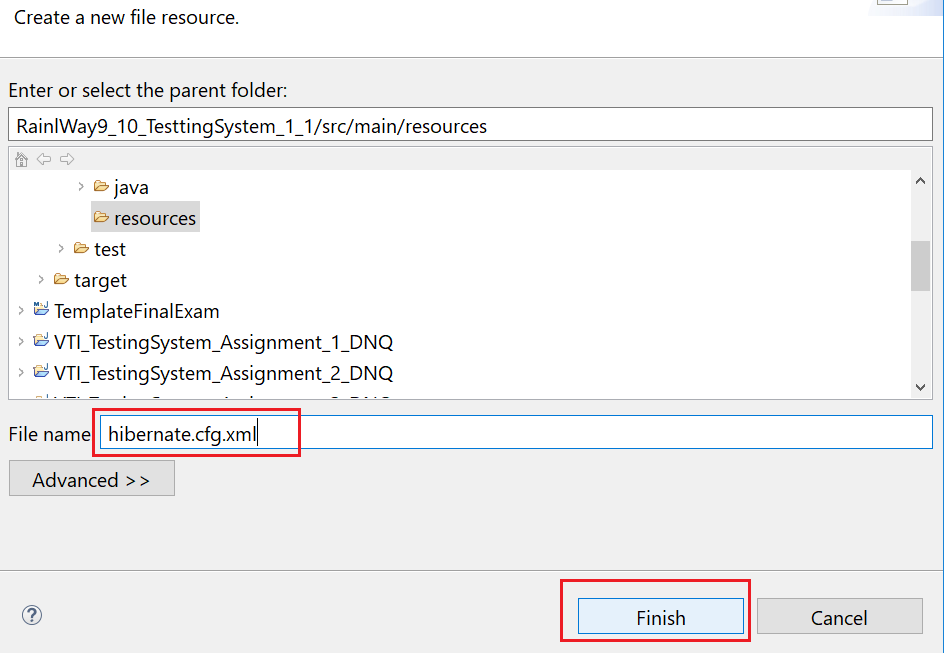


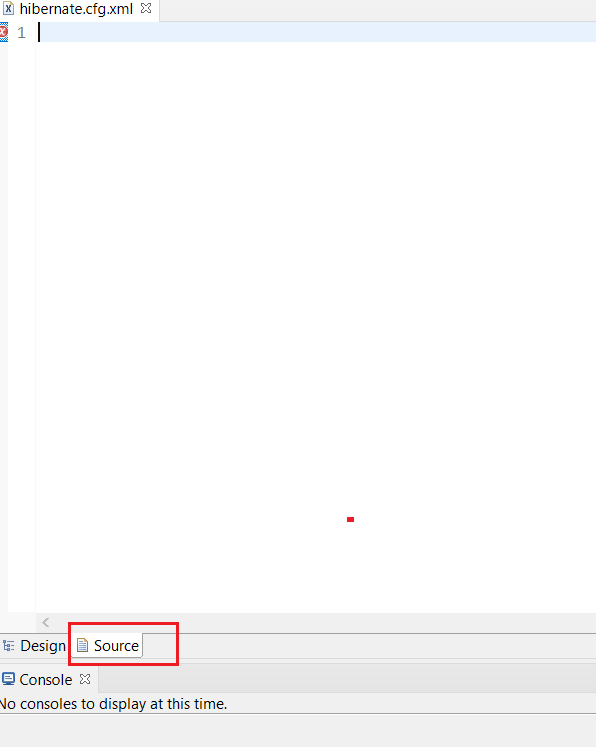




* Tạo mới file: **hibernate.cfg.xml** trong folder vừa tạo.



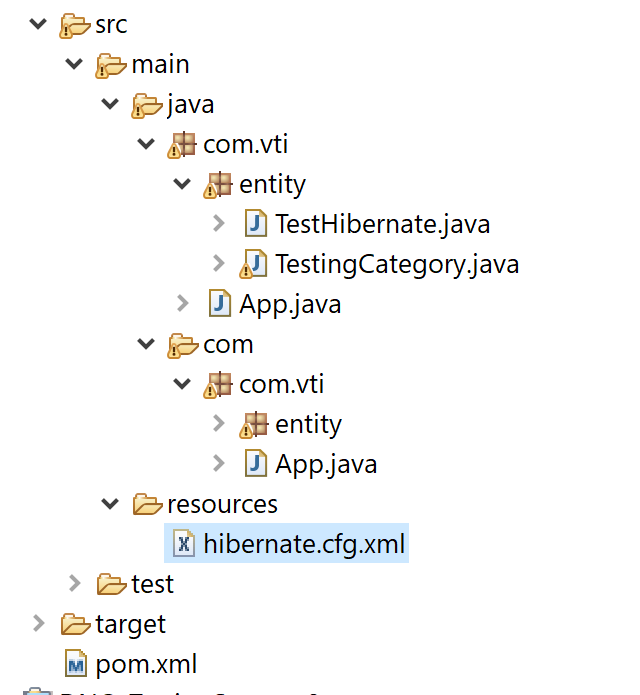




* Copy nội dung bên dưới vào file vừa tạo.

|  |
| --- |
| <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE hibernate-configuration PUBLIC  "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  <hibernate-configuration>  <session-factory>  <!-- Database connection settings -->  <property name=*"connection.driver\_class"*>com.mysql.cj.jdbc.Driver</property>  <property name=*"connection.url"*>jdbc:mysql://localhost:3306/TestingSystem</property>  <property name=*"connection.username"*>root</property>  <property name=*"connection.password"*>root</property>  <!-- format code SQL -->  <property name=*"show\_sql"*>true</property>  <property name=*"hibernate.format\_sql"*>true</property>  <property name=*"connection.pool\_size"*>10</property>  <!-- other -->  <property name=*"hibernate.connection.characterEncoding"*>utf8</property>  </session-factory>  </hibernate-configuration> |





Tạo DB TestingSystem để Demo:

|  |
| --- |
| -- create database  DROP DATABASE IF EXISTS TestingSystem;  CREATE DATABASE TestingSystem;  USE TestingSystem;  -- create table 1: Department  DROP TABLE IF EXISTS Testing\_Category;  CREATE TABLE Testing\_Category(  id TINYINT UNSIGNED AUTO\_INCREMENT PRIMARY KEY,  `name` NVARCHAR(50) NOT NULL UNIQUE KEY  );  INSERT INTO Testing\_Category(`name`)  VALUES  ('Category 1'),  ('Category 2'),  ('Category 3'),  ('Category 4'),  ('Category 5'),  ('Category 6'),  ('Category 7'); |

# **Tạo Class TestingCategory trong package com.vti.entity:**

|  |
| --- |
| **package** com.vti.entity;  **import** java.io.Serializable;  **import** javax.persistence.Column;  **import** javax.persistence.Entity;  **import** javax.persistence.GeneratedValue;  **import** javax.persistence.GenerationType;  **import** javax.persistence.Id;  **import** javax.persistence.Table;  **import** net.bytebuddy.dynamic.loading.ClassReloadingStrategy.Strategy;  @Entity  @Table(name ="Testing\_Category", catalog = "TestingSystem")  **public** **class** TestingCategory **implements** Serializable {  @Column(name = "id")  @Id  @GeneratedValue(strategy = GenerationType.IDENTITY)  **private** **short** id;  @Column(name = "name", length = 50, nullable = **false**, unique = **true**)  **private** String name;    **public** TestingCategory() {  **super**();  }  /\*\*  \* **@return** the id  \*/  **public** **short** getId() {  **return** id;  }  /\*\*  \* **@param** id the id to set  \*/  **public** **void** setId(**short** id) {  **this**.id = id;  }  /\*\*  \* **@return** the name  \*/  **public** String getName() {  **return** name;  }  /\*\*  \* **@param** name the name to set  \*/  **public** **void** setName(String name) {  **this**.name = name;  }      } |

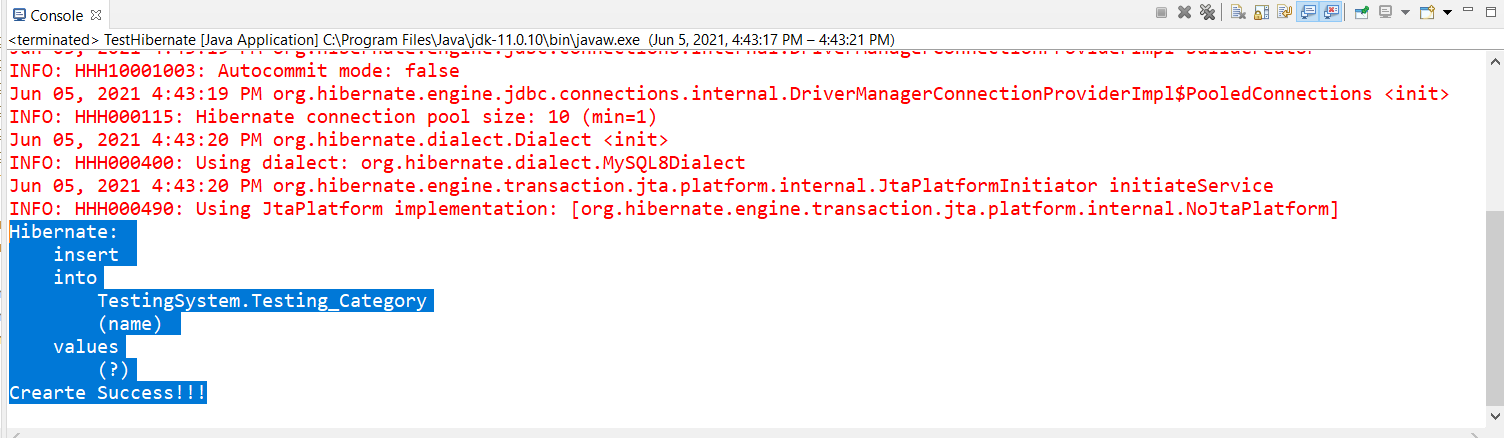
****

# **Tạo Class TestHibernate.java trong package com.vti.entity**

|  |
| --- |
| **package** com.vti.frontend;  **import** java.util.List;  **import** org.hibernate.Session;  **import** org.hibernate.SessionFactory;  **import** org.hibernate.boot.registry.StandardServiceRegistryBuilder;  **import** org.hibernate.cfg.Configuration;  **import** org.hibernate.query.Query;  **import** org.hibernate.service.ServiceRegistry;  **import** com.vti.entity.TestingCategory;  **public** **class** TestHibernate {  **public** **static** **void** main(String[] args) {  Session session = **null**;  **try** {  session = *buildSessionFactory*().openSession();  session.beginTransaction();  // Insert dữ liệu:  TestingCategory testingcategory = **new** TestingCategory();  testingcategory.setName("Category Test");  session.save(testingcategory);  System.***out***.println("Crearte Success!!!");  // Select dữ liệu cơ bản:  Query<TestingCategory> query = session.createQuery("FROM TestingCategory");  List<TestingCategory> listResul = query.list();  **for** (TestingCategory testingCategory2 : listResul) {  System.***out***.println(testingCategory2.getId() + " " + testingCategory2.getName());  }  session.getTransaction().commit();  } **finally** {  **if** (session != **null**) {  session.close();  }  }  }  **private** **static** SessionFactory buildSessionFactory() {  Configuration configuration = **new** Configuration();  configuration.configure("hibernate.cfg.xml");  configuration.addAnnotatedClass(TestingCategory.**class**);  ServiceRegistry serviceRegistry = **new** StandardServiceRegistryBuilder()  .applySettings(configuration.getProperties()).build();  **return** configuration.buildSessionFactory(serviceRegistry);  }  } |

****

# **Chạy chương trình để kiểm tra:**

****