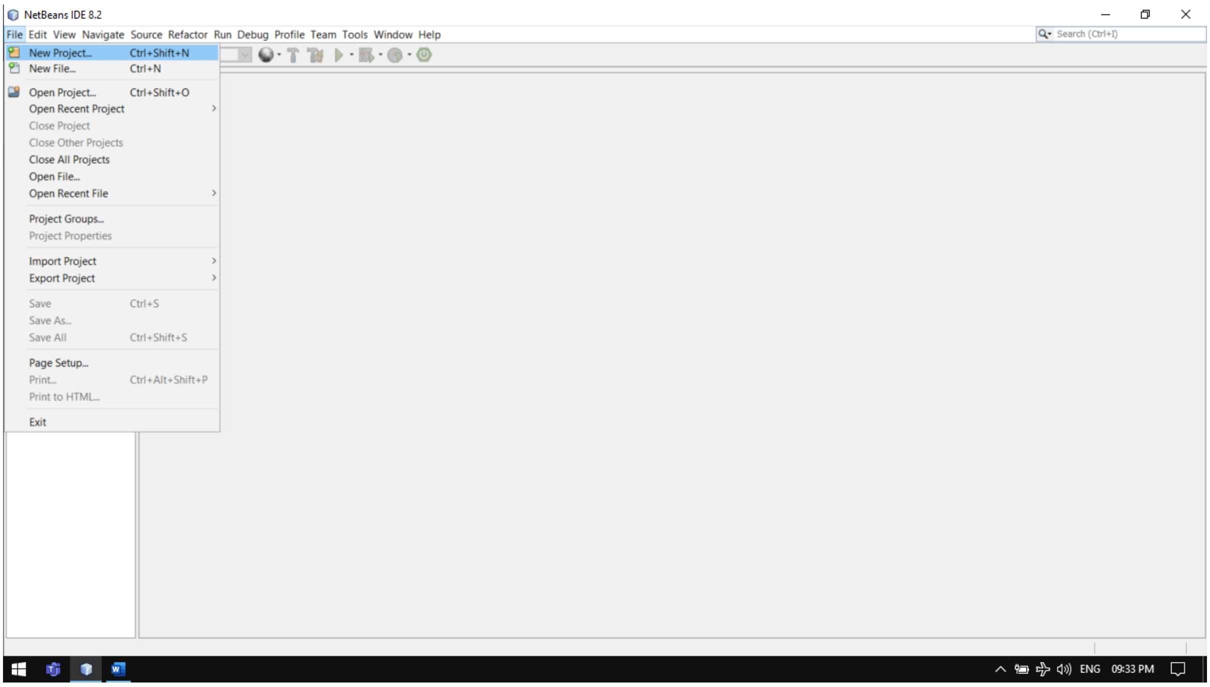
Hibernate Framework

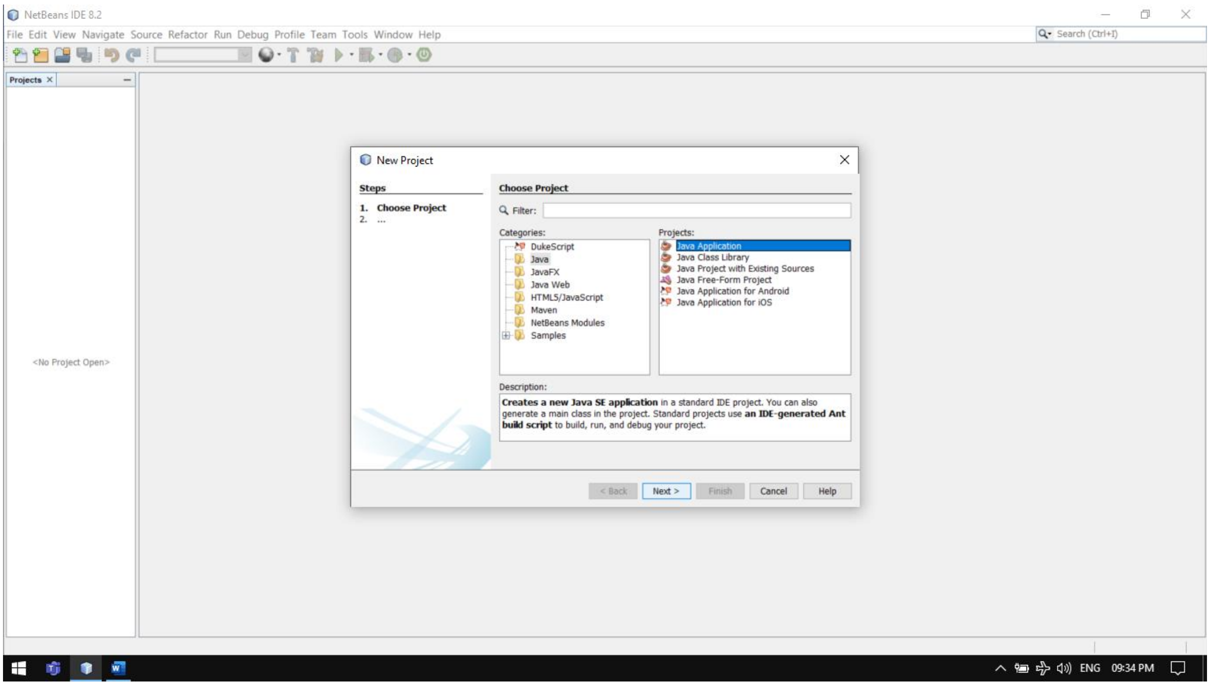
Written by : Samundar Singh

Getting Started:

Step 1:   Select new project



Step 2: Choose Java Application.



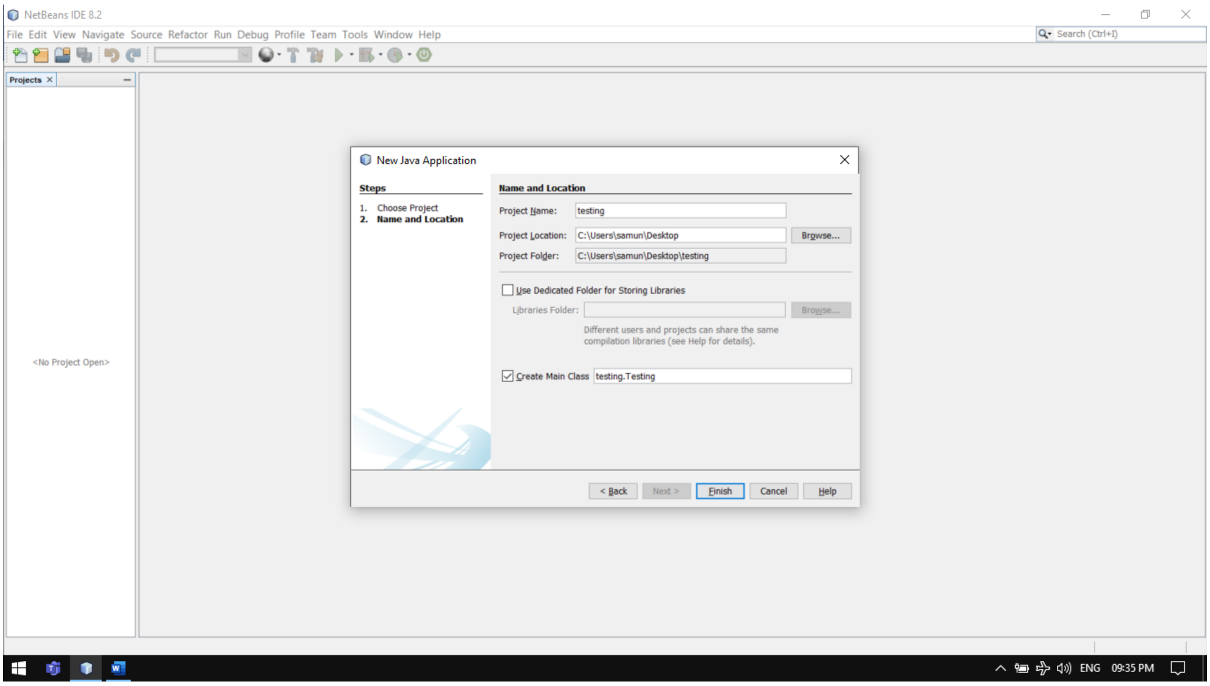
Categories: Java

Projects: Java Application

Description:

Creates a new Java SE application in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

Step 3: Enter Name then Select finish.



New Java Application

Steps

1. Choose Project
2. Name and Location

Name and Location

Project Name: testing

Project Location: C:\Users\samun\Desktop

Project Folder: C:\Users\samun\Desktop\testing

☐ Use Dedicated Folder for Storing Libraries

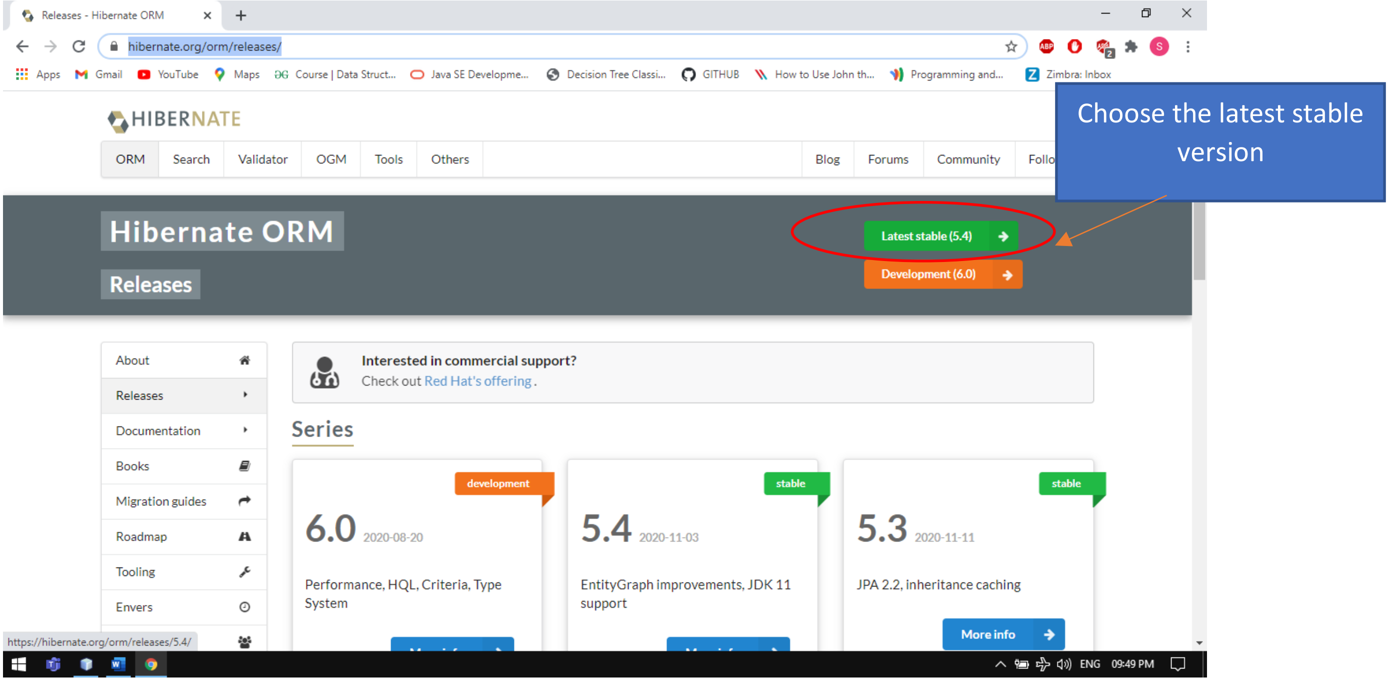
Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class testing.Testing

Step 4: Here we have to download some jar files of hibernate.

Visit website https://hibernate.org/orm/releases



Hibernate ORM

Releases

Latest stable (5.4)

Development (6.0)

Interested in commercial support?
Check out Red Hat's offering.

Series

development
6.0 2020-08-20
Performance, HQL, Criteria, Type System

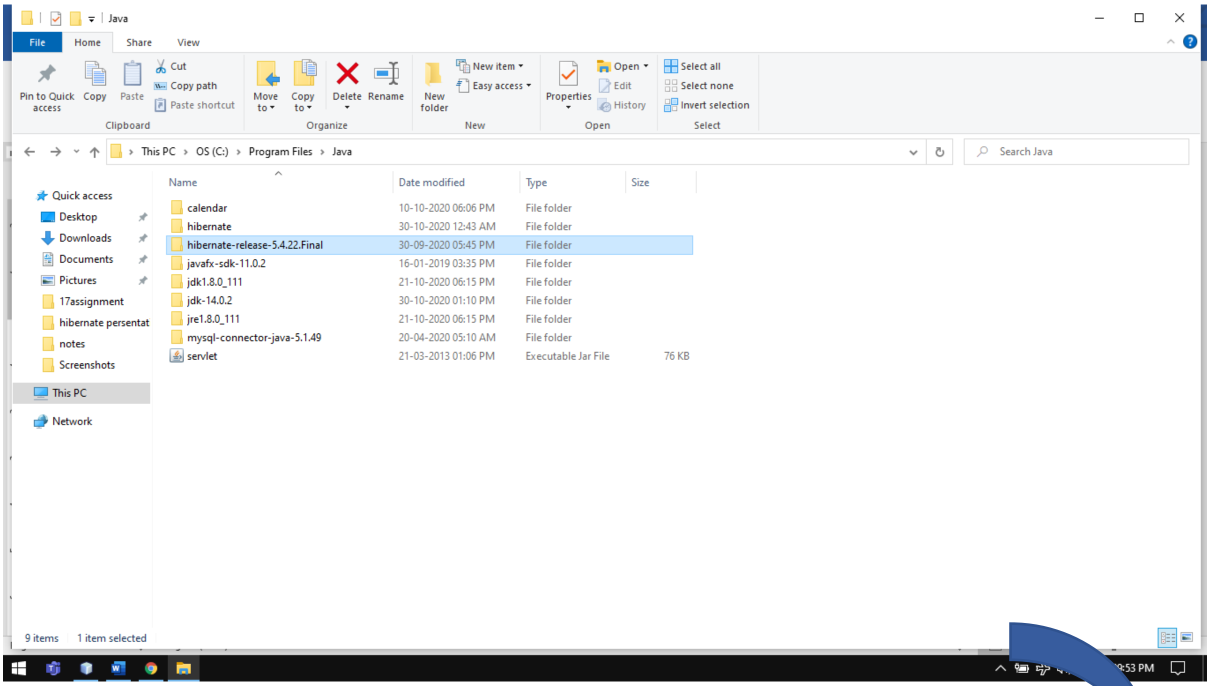
stable
5.4 2020-11-03
EntityGraph improvements, JDK 11 support

stable
5.3 2020-11-11
JPA 2.2, inheritance caching

Choose the latest stable version

Written by : Samundar Singh

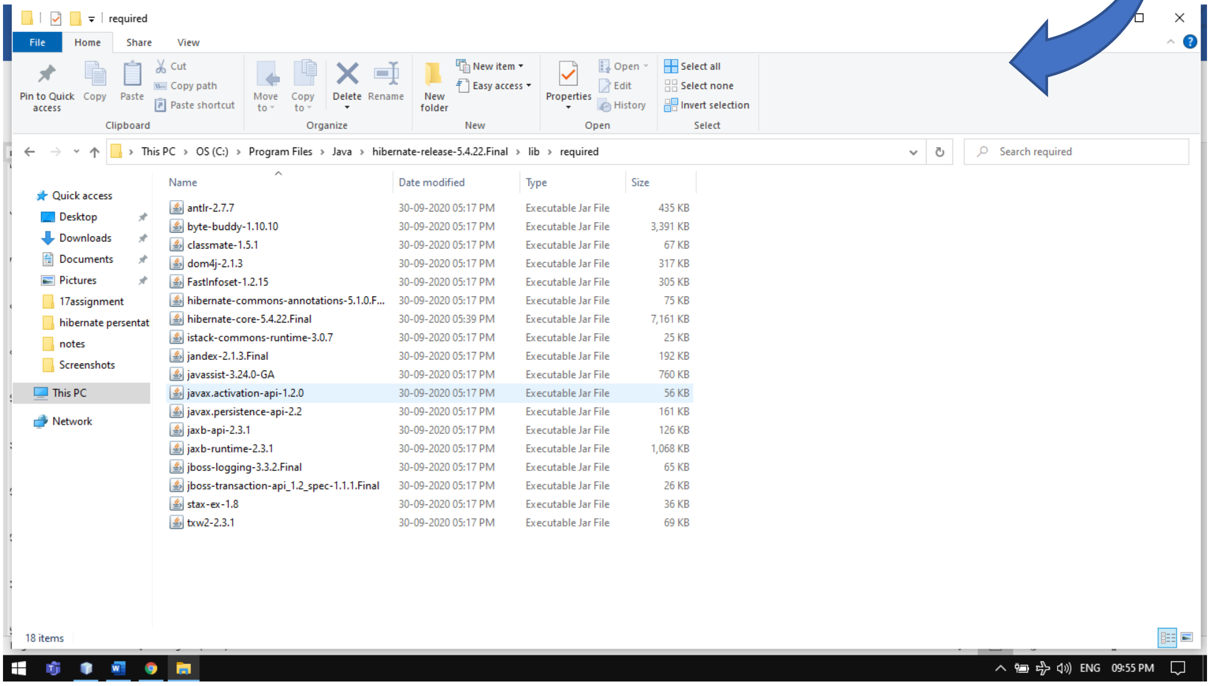
Step 5: Download it and Unzip it will be shown like this



Name	Date modified	Type	Size
calendar	10-10-2020 06:06 PM	File folder	
hibernate	30-10-2020 12:43 AM	File folder	
hibernate-release-5.4.22.Final	30-09-2020 05:45 PM	File folder	
javafx-sdk-11.0.2	16-01-2019 03:35 PM	File folder	
jdk1.8.0_111	21-10-2020 06:15 PM	File folder	
jdk-14.0.2	30-10-2020 01:10 PM	File folder	
jre1.8.0_111	21-10-2020 06:15 PM	File folder	
mysql-connector-java-5.1.49	20-04-2020 05:10 AM	File folder	
servlet	21-03-2013 01:06 PM	Executable Jar File	76 KB

For simple project we will need very few files so

Go :> hibernate-release-5.4.22.Final\lib\required

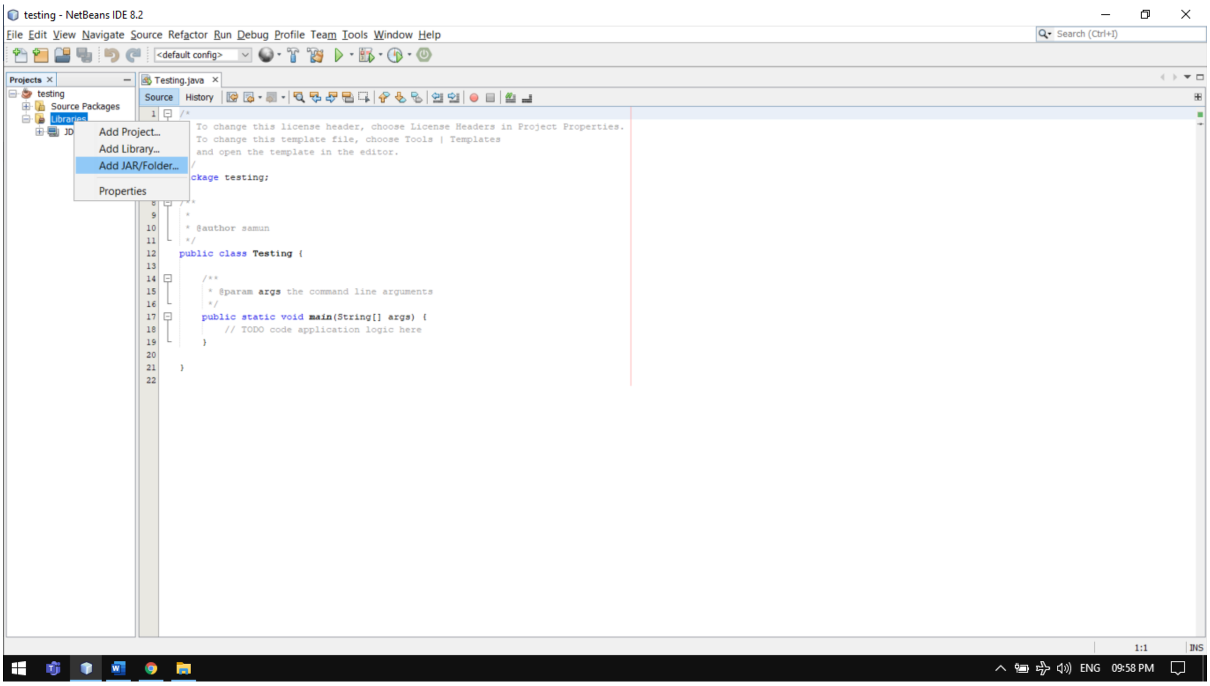


Name	Date modified	Type	Size
antlr-2.7.7	30-09-2020 05:17 PM	Executable Jar File	435 KB
byte-buddy-1.10.10	30-09-2020 05:17 PM	Executable Jar File	3,391 KB
classmate-1.5.1	30-09-2020 05:17 PM	Executable Jar File	67 KB
dom4j-2.1.3	30-09-2020 05:17 PM	Executable Jar File	317 KB
FastInfoset-1.2.15	30-09-2020 05:17 PM	Executable Jar File	305 KB
hibernate-commons-annotations-5.1.0.F...	30-09-2020 05:17 PM	Executable Jar File	75 KB
hibernate-core-5.4.22.Final	30-09-2020 05:39 PM	Executable Jar File	7,161 KB
istack-commons-runtime-3.0.7	30-09-2020 05:17 PM	Executable Jar File	25 KB
jandex-2.1.3.Final	30-09-2020 05:17 PM	Executable Jar File	192 KB
javassist-3.24.0-GA	30-09-2020 05:17 PM	Executable Jar File	760 KB
javax.activation-api-1.2.0	30-09-2020 05:17 PM	Executable Jar File	56 KB
javax.persistence-api-2.2	30-09-2020 05:17 PM	Executable Jar File	161 KB
jaxb-api-2.3.1	30-09-2020 05:17 PM	Executable Jar File	126 KB
jaxb-runtime-2.3.1	30-09-2020 05:17 PM	Executable Jar File	1,068 KB
jboss-logging-3.3.2.Final	30-09-2020 05:17 PM	Executable Jar File	65 KB
jboss-transaction-api_1.2_spec-1.1.1.Final	30-09-2020 05:17 PM	Executable Jar File	26 KB
stax-ex-1.8	30-09-2020 05:17 PM	Executable Jar File	36 KB
txw2-2.3.1	30-09-2020 05:17 PM	Executable Jar File	69 KB

Here we are including only these file(as per requirement add optional jar file it’s totally upto you)

Written by : Samundar Singh

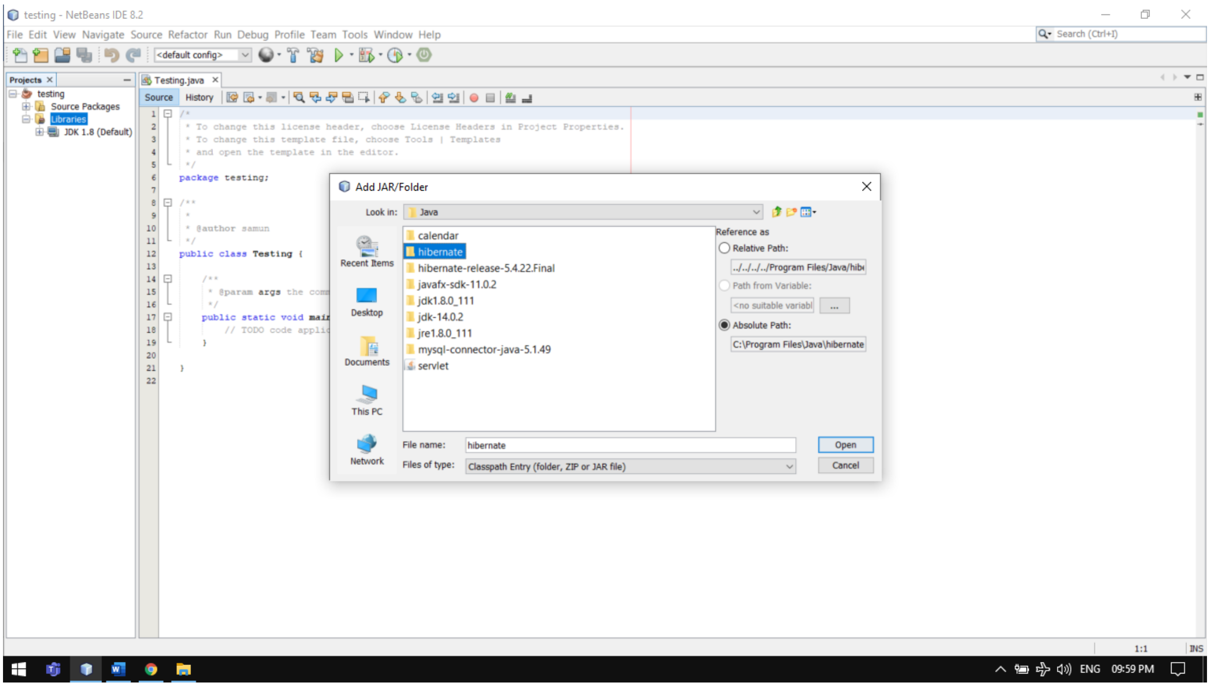
Step 6: Add the above jar file to your project .



Add Project...
Add Library...
Add JAR/Folder...
Properties

```
/*
 To change this license header, choose License Headers in Project Properties.
 To change this template file, choose Tools | Templates
 and open the template in the editor.
/
ckage testing;

/**
 *
 * @author samun
 */
public class Testing {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Select open option.



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package testing;

/**
 *
 * @author samun
 */
public class Testing {

    /**
     * @param args the comm
     */
    public static void mair
        // TODO code applic
    }

}
```

Add JAR/Folder

Look in: Java

- calendar
- hibernate
- hibernate-release-5.4.22.Final
- javafx-sdk-11.0.2
- jdk1.8.0_111
- jdk-14.0.2
- jre1.8.0_111
- mysql-connector-java-5.1.49
- servlet

Reference as

○ Relative Path: ../../../../Program Files/Java/hibe

○ Path from Variable: <no suitable variabl ...

● Absolute Path: C:\Program Files\Java\hibernate

File name: hibernate

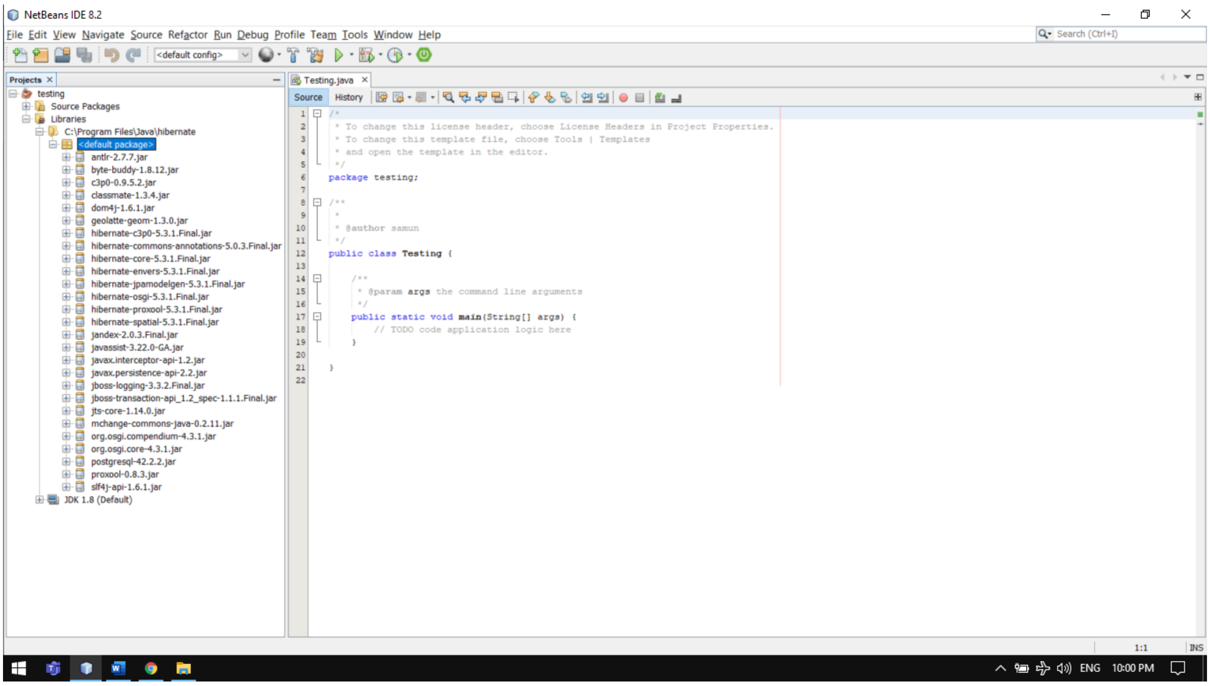
Files of type: Classpath Entry (folder, ZIP or JAR file)

Open

Cancel

Written by : Samundar Singh

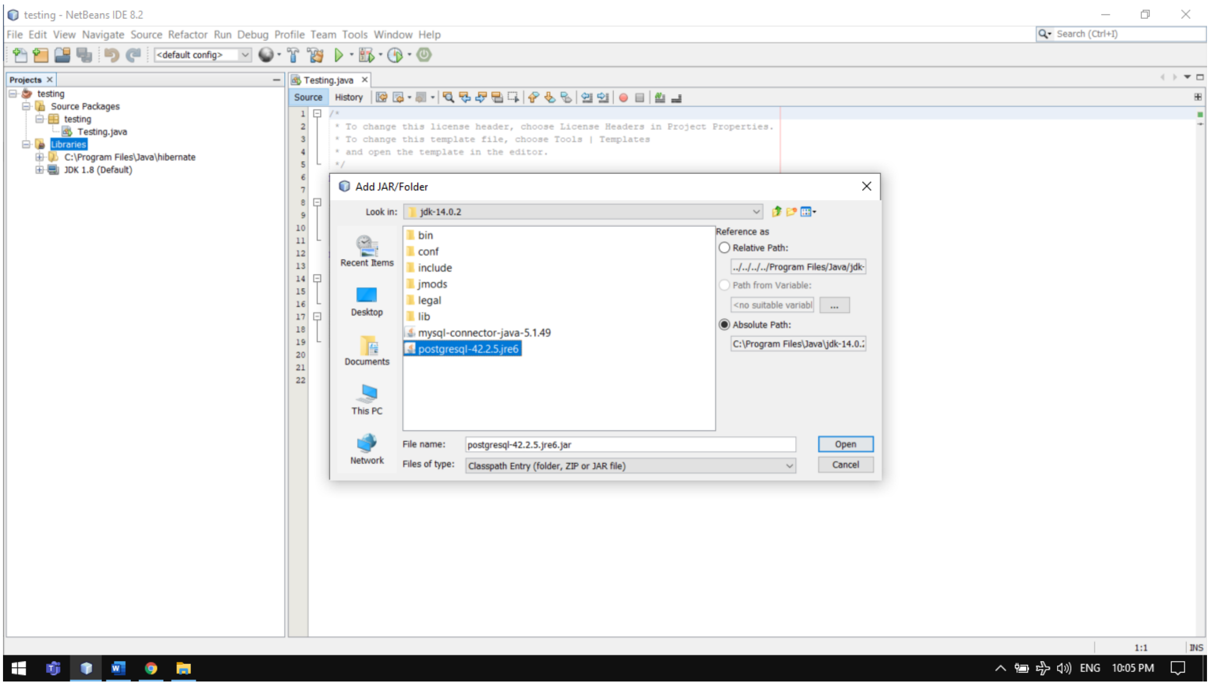
After including all the jar file it will be shown like this.



```
testing
  Source Packages
  Libraries
    C:\Program Files\Java\hibernate
      <default package>
        antlr-2.7.7.jar
        byte-buddy-1.8.12.jar
        c3p0-0.9.5.2.jar
        classmate-1.3.4.jar
        dom4j-1.6.1.jar
        geolatte-geom-1.3.0.jar
        hibernate-c3p0-5.3.1.Final.jar
        hibernate-commons-annotations-5.0.3.Final.jar
        hibernate-core-5.3.1.Final.jar
        hibernate-envers-5.3.1.Final.jar
        hibernate-jpamodelgen-5.3.1.Final.jar
        hibernate-osgi-5.3.1.Final.jar
        hibernate-proxool-5.3.1.Final.jar
        hibernate-spatial-5.3.1.Final.jar
        jandex-2.0.3.Final.jar
        javassist-3.22.0-GA.jar
        javax.interceptor-api-1.2.jar
        javax.persistence-api-2.2.jar
        jboss-logging-3.3.2.Final.jar
        jboss-transaction-api_1.2_spec-1.1.1.Final.jar
        jts-core-1.14.0.jar
        mchange-commons-java-0.2.11.jar
        org.osgi.compendium-4.3.1.jar
        org.osgi.core-4.3.1.jar
        postgresql-42.2.2.jar
        proxool-0.8.3.jar
        slf4j-api-1.6.1.jar
    JDK 1.8 (Default)
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package testing;

/**
 *
 * @author samun
 */
public class Testing {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Step 7: Download the jdbc jar file of the database that u are going to use and include it in same manner as we include the hibernate jar file which is mentioned above.



```
testing
  Source Packages
    testing
      Testing.java
  Libraries
    C:\Program Files\Java\hibernate
    JDK 1.8 (Default)
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

Look in: jdk-14.0.2

- bin
- conf
- include
- jmods
- legal
- lib
- mysql-connector-java-5.1.49
- postgresql-42.2.5.jre6

Reference as

○ Relative Path: ../../../../Program Files/Java/jdk-

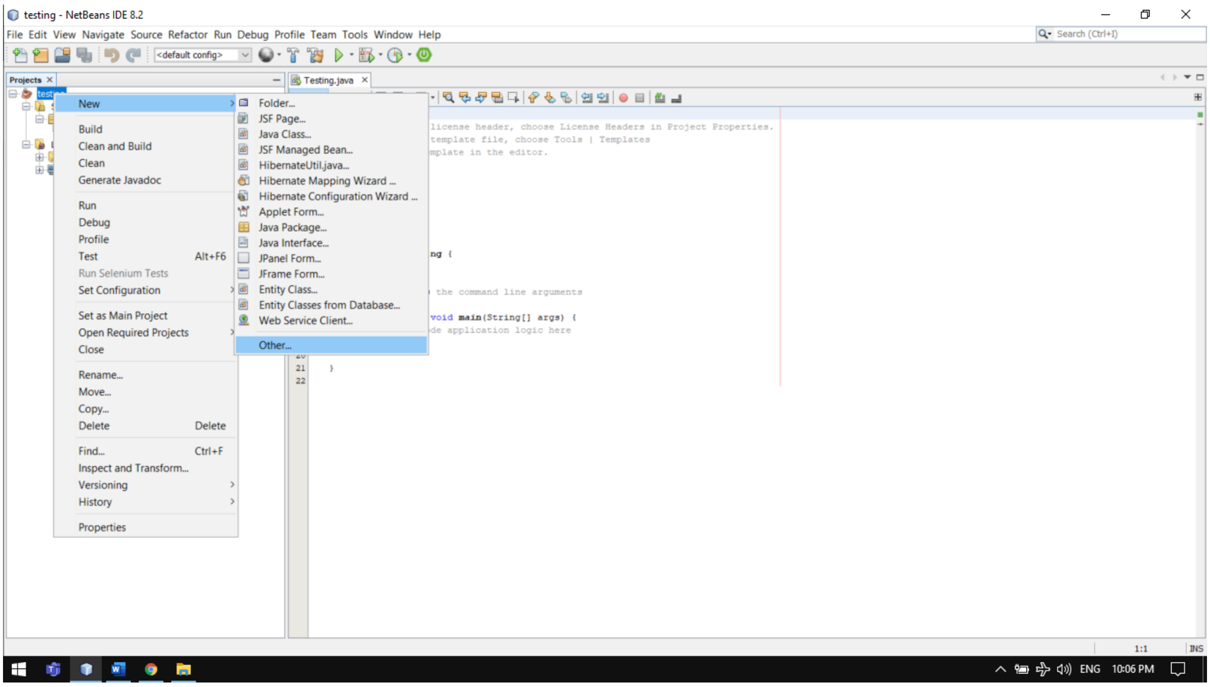
○ Path from Variable: <no suitable variabl ...

● Absolute Path: C:\Program Files\Java\jdk-14.0.2

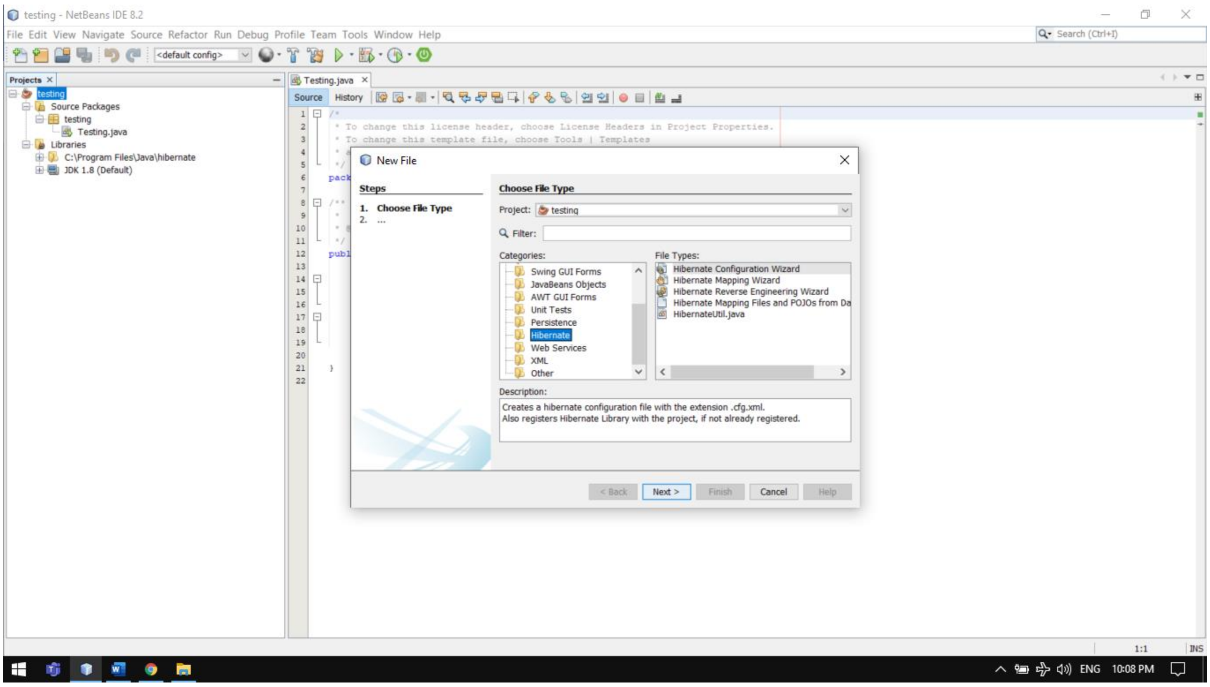
File name: postgresql-42.2.5.jre6.jar

Files of type: Classpath Entry (folder, ZIP or JAR file)

Step 8: Now our project is configured Properly.to include other files we have to manually include it one by one.



Step 9: Chose hibernate from the menu.



Project: testing

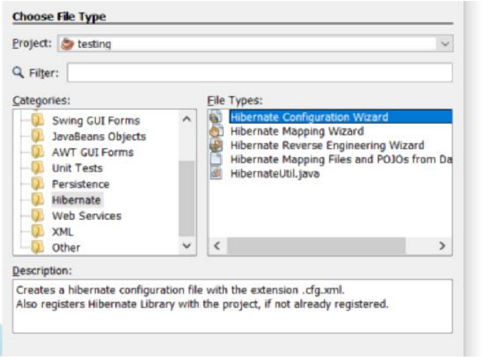
Filter:

Description:

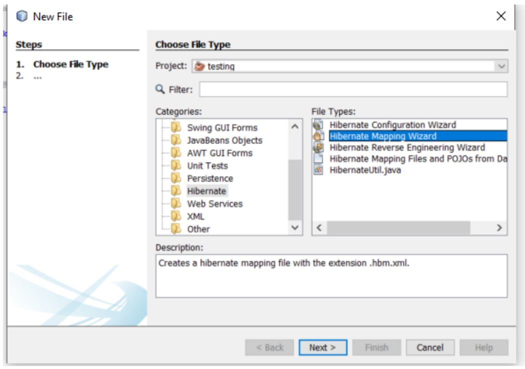
Creates a hibernate configuration file with the extension .cfg.xml.
Also registers Hibernate Library with the project, if not already registered.

Written by : Samundar Singh

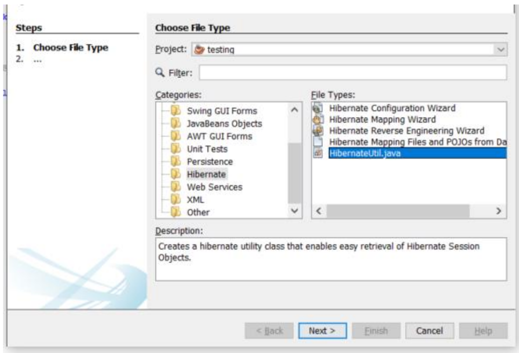
Step 10: From here include the files one by one in project, the list of files are:



Creates a hibernate configuration file with the extension .cfg.xml.
Also registers Hibernate Library with the project, if not already registered.

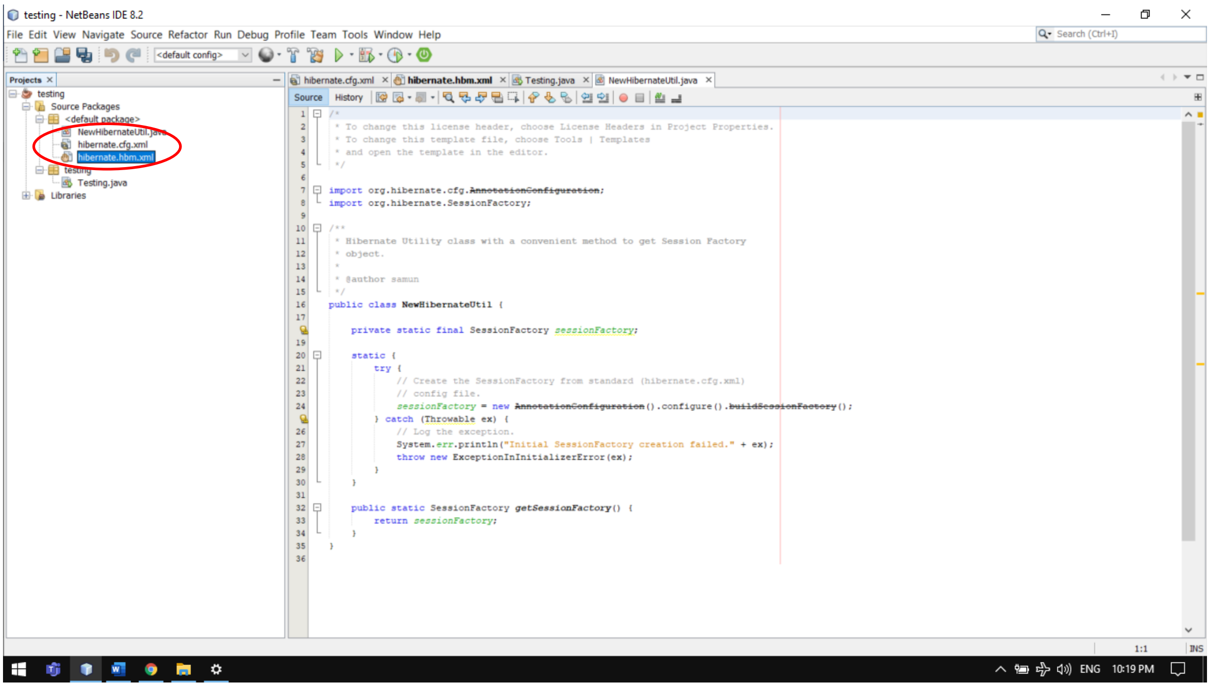


Creates a hibernate mapping file with the extension .hbm.xml.



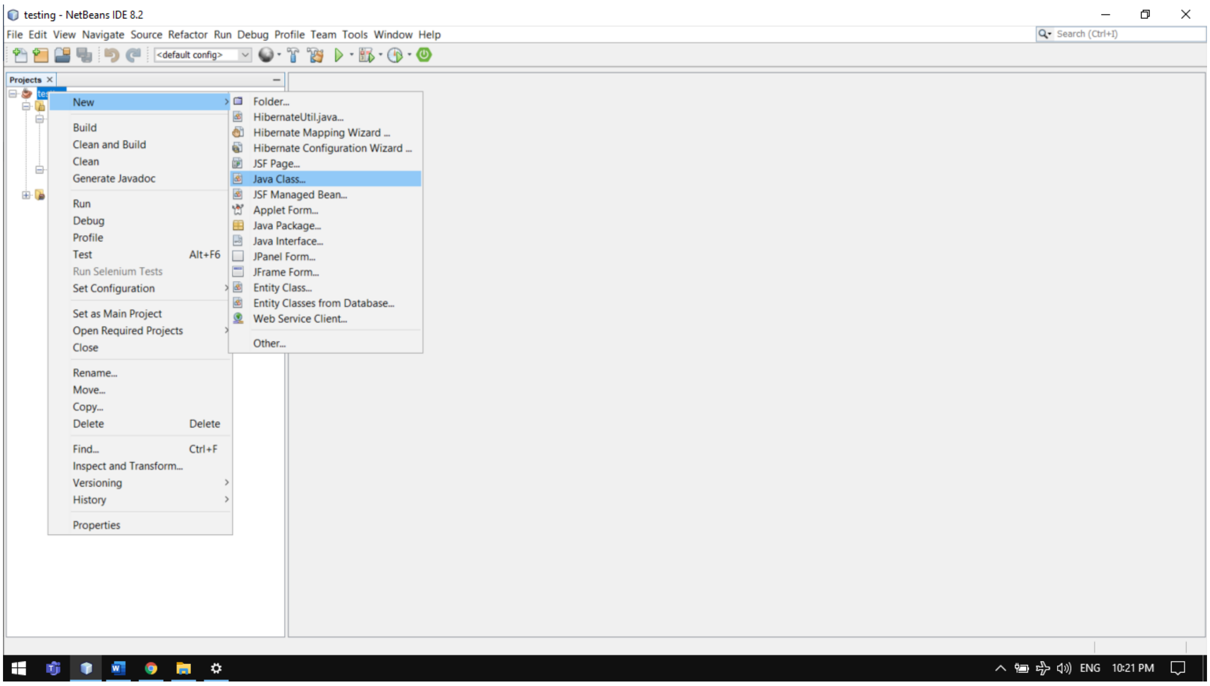
Creates a hibernate utility class that enables easy retrieval of Hibernate Session Objects.

Go through all the setup Wizards and include them in project

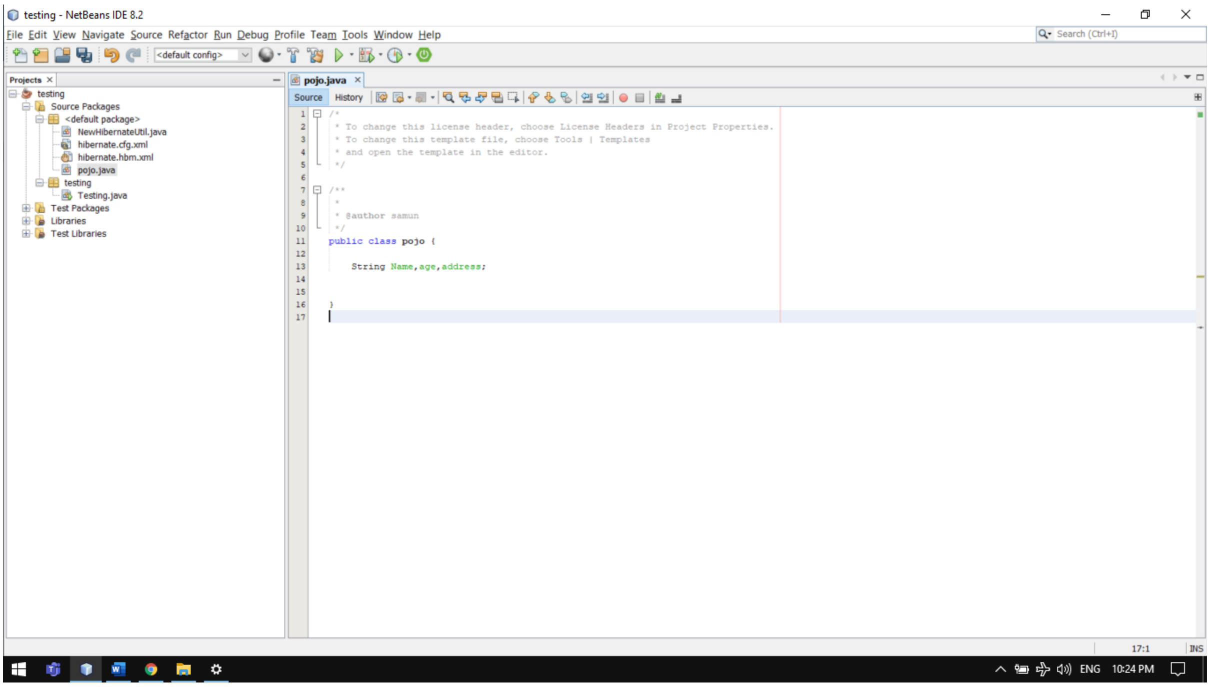


```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

import org.hibernate.cfg.AnnotationConfiguration;
import org.hibernate.SessionFactory;

/**
 * Hibernate Utility class with a convenient method to get Session Factory
 * object.
 *
 * @author samun
 */
public class NewHibernateUtil {

    private static final SessionFactory sessionFactory;

    static {
        try {
            // Create the SessionFactory from standard (hibernate.cfg.xml)
            // config file.
            sessionFactory = new AnnotationConfiguration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            // Log the exception.
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

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

```

Step 11: Here I am creating a POJO file manually

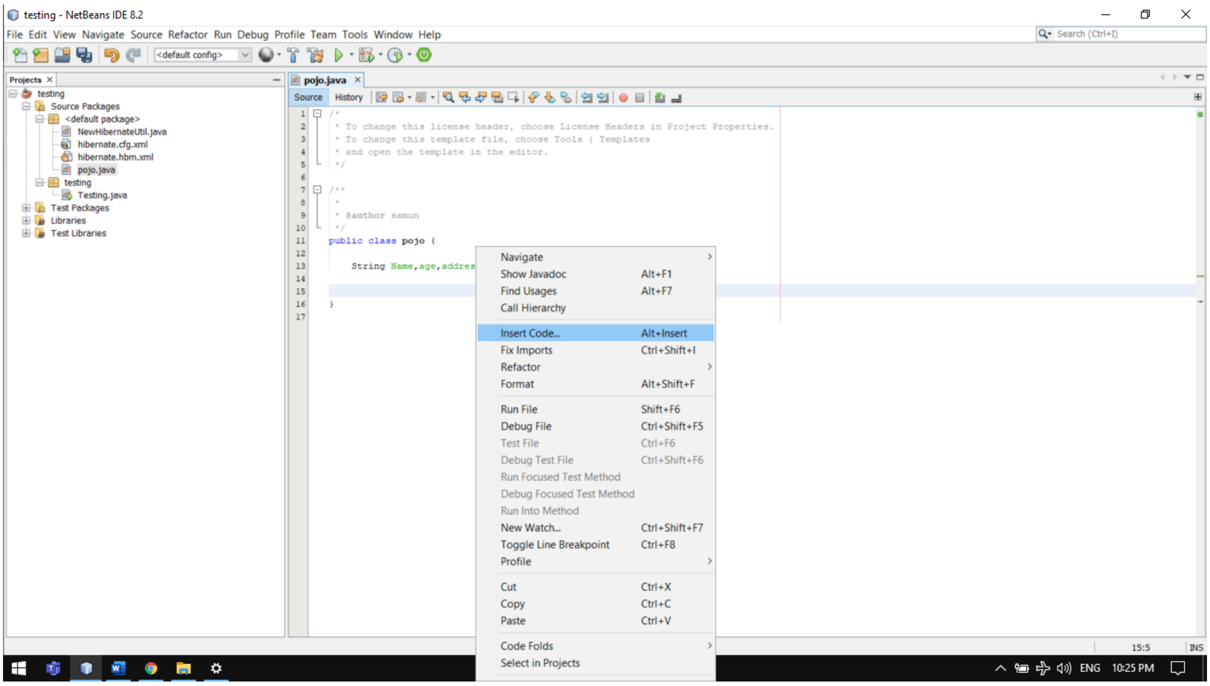


Step 12: After creating file simply mentioned the data that you want to map with database

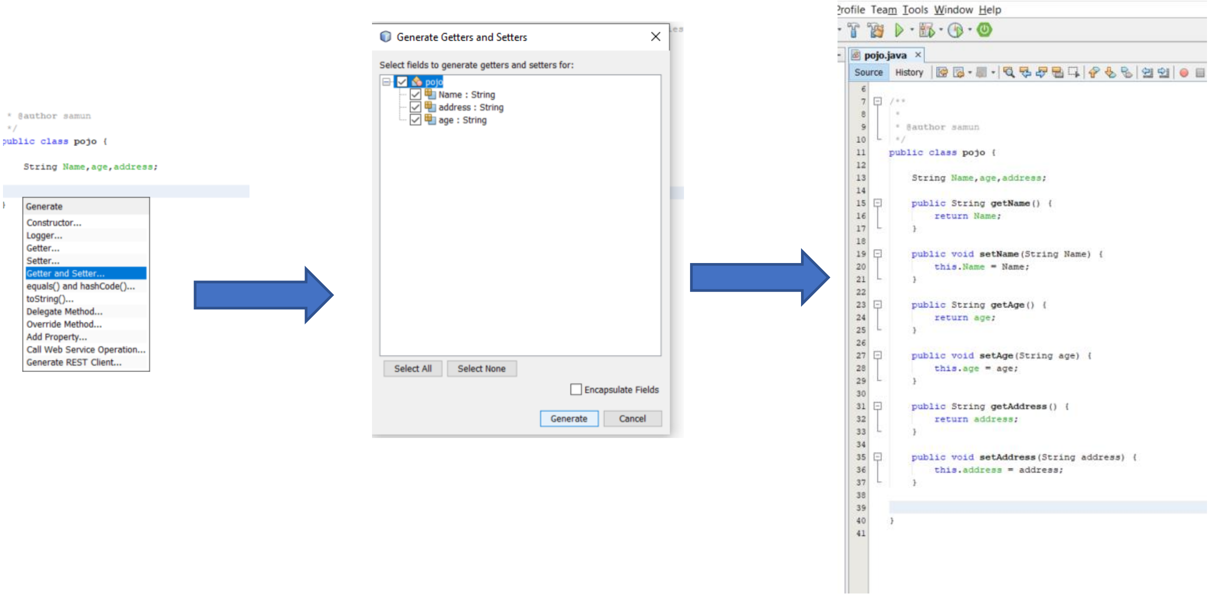


```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author samun
 */
public class pojo {

    String Name,age,address;


}
```

Step 12: After then simply right click choose insert code then getter and setter



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author samun
 */
public class pojo {

    String Name,age,addres

}
```



Select fields to generate getters and setters for:

- ☑ pojo
  - ☑ Name : String
  - ☑ address : String
  - ☑ age : String

□ Encapsulate Fields

```java
/**
 *
 * @author samun
 */
public class pojo {

    String Name,age,address;

    public String getName() {
        return Name;
    }

    public void setName(String Name) {
        this.Name = Name;
    }

    public String getAge() {
        return age;
    }

    public void setAge(String age) {
        this.age = age;
    }

    public String getAddress() {
        return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

}
```

Now your project is ready for the implementation All you need to do just writing some code and make some setting that is mentioned in ppt or in the given file .

Some important notes/terms:

1. What is hibernate?

Ans: Hibernate is an open-source and lightweight ORM tool that is used to store, manipulate, and retrieve data from the database.

2. What is ORM?

Ans: ORM is an stands for Object/Relational mapping. It is a programming strategy to map object with the data stored in the database. It simplifies data creation, data manipulation, and data access.

3. What are persistent classes?

Ans: Classes whose objects are stored in a database table are called as persistent classes (POJO file).

4. What is Session?

Ans: It maintains a connection between the hibernate application and database. It provides methods to store, update, delete or fetch data from the database such as persist (), update (), delete (), load (), get () etc.

5. What is HQL?

Ans: HQL is the acronym of Hibernate Query Language. It is an Object-Oriented Query Language and is independent of the database.

6. What do you mean by Hibernate Configuration File?

Ans: *Hibernate Configuration File* mainly contains database-specific configurations and are used to initialize *SessionFactory*. Some important parts of the Hibernate Configuration File are Dialect information, so that hibernate knows the database type and mapping file or class details.

Hibernate configuration Steps:

1. First, identify the POJOs (Plain Old Java Objects) that have a database representation.
2. Identify which properties of POJOs need to be continued.
3. Annotate each of the POJOs in order to map the Java objects to columns in a database table.
4. Create a database schema using the schema export tool which uses an existing database, or you can create your own database schema.
5. Add Hibernate Java libraries to the application's classpath.
6. Create a Hibernate *XML configuration file* that points to the database and the mapped classes.
7. In the Java application, you can create a Hibernate Configuration object that refers to your XML configuration file.

8. Also, build a Hibernate SessionFactory object from the Configuration object.
9. Retrieve the Hibernate Session objects from the SessionFactory and write down the data access logic for your application (create, retrieve, update, and delete).

7. What do you mean by Hibernate mapping File?

Ans: Hibernate mapping file is used to define the entity bean fields and database table column mappings. We know that JPA annotations can be used for mapping but sometimes XML mapping file comes handy when we are using third party classes and we can't use annotations.

8. What Elements of Hibernate Architecture?

Ans:

SessionFactory

The SessionFactory is a factory of session and client of ConnectionProvider. It holds second level cache (optional) of data. The org.hibernate.SessionFactory interface provides factory method to get the object of Session.

Session

The session object provides an interface between the application and data stored in the database. It is a short-lived object and wraps the JDBC connection. It is factory of Transaction, Query and Criteria. It holds a first-level cache (mandatory) of data. The org.hibernate.Session interface provides methods to insert, update and delete the object. It also provides factory methods for Transaction, Query and Criteria.

Transaction

The transaction object specifies the atomic unit of work. It is optional. The org.hibernate.Transaction interface provides methods for transaction management.

ConnectionProvider

It is a factory of JDBC connections. It abstracts the application from DriverManager or DataSource. It is optional.

TransactionFactory

It is a factory of Transaction. It is optional.

V.V.I

9. What Difference between first level and second level cache in Hibernate?

Ans: The main difference between first level and second level cache in Hibernate is that the first level is maintained at the Session level and accessible only to the Session, while the second level cache is maintained at the SessionFactory level and available to all Sessions. This means, you can use the first level cache to store local data, i.e. the data which is needed by the Session, and you can use the second-level cache to store global data, i.e. something which can be shared across sessions.

10.What is First Level Cache in Hibernate?

As the name suggests, the first-level cache is the first cache hibernate consults before loading an object from the database. It is maintained at Session level, and it's by default enabled.

We know that Session is the interface between Hibernate and Database. We load objects using Session, e.g. by calling get() or load() method or by executing queries.

Example:

When we ask load method to return a Person (an object stored in database) with Id=1, first time it's loaded from database, that too lazily, when any method of that object is called other than getId(), next time if you load the same object then Hibernate doesn't go to Database, instead it **returns the object from first level cache maintained at Session level**.

The data in the first level cache is maintained as long as Session is open, as soon as you close the Session, all data is lost. Next time even if you load the same object, e.g. Person with the same id, Hibernate will again go to Database to load that object, provided Second-level Cache is disabled

10.What is Second Level Cache in Hibernate?

Ans: Unlike first level cache which is accessible only to the session who maintains it, Second-level Cache is accessible to all Sessions.

This means if one Session loads an object, like Person with `id=1` and Second session also loads the same object, only one database call will be made. Next session will get the data from the Second-level Cache.

Means no database call each time.

This also helps in interacting with different databases.