# Pom.xml:

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.FlyAway.project</groupId>

<artifactId>flyawayapp</artifactId>

<version>1.0</version>

<packaging>war</packaging>

<properties>

<maven.compiler.source>11</maven.compiler.source>

<maven.compiler.target>11</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>3.1.0</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>javax.servlet.jsp-api</artifactId>

<version>2.3.1</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.30</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.4.32.Final</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<dependency>

<groupId>taglibs</groupId>

<artifactId>standard</artifactId>

<version>1.1.2</version>

</dependency>

</dependencies>

<!--below configuration will allow us to not want web.xml file

packaging should be war -->

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-war-plugin</artifactId>

<version>3.3.1</version>

<configuration>

<!--we dont need web.xml file, web.xml is very old way -->

<failOnMissingWebXml>false</failOnMissingWebXml>

</configuration>

</plugin>

</plugins>

<resources>

<resource>

<directory>src/main/java</directory>

<filtering>true</filtering>

</resource>

</resources>

</build>

</project>

# HibernateUtils.java:

package flyawayapp.utils;

import org.hibernate.SessionFactory;

import org.hibernate.boot.Metadata;

import org.hibernate.boot.MetadataSources;

import org.hibernate.boot.registry.StandardServiceRegistry;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

// this is one time activity

public class HibernateUtils {

private static StandardServiceRegistry standardServiceRegistry;

// it is a factory for your session

private static SessionFactory sessionFactory;

public static SessionFactory getSessionFactory() {

// session factory is an object, is used to create sessionobject which is like connection object in jdbc

// through which we get session object, without this we wont be able to communicate with database

if ( sessionFactory == null ) {

try {

// load hibernate.cfg.xml and build

// we need to follow these levels for security purpose

standardServiceRegistry = new StandardServiceRegistryBuilder().configure().build();

// once registry is created

MetadataSources metadataSources = new MetadataSources(standardServiceRegistry);

Metadata metadata = metadataSources.getMetadataBuilder().build();

// session is the starting point which is created using sessionFactory

sessionFactory = metadata.getSessionFactoryBuilder().build();

}catch ( Exception e) {

e.printStackTrace();

if ( standardServiceRegistry != null ) {

StandardServiceRegistryBuilder.destroy(standardServiceRegistry);

}

}

}

return sessionFactory;

}

}