Java 9

Hibernate 5

**Dependencies in POM**

<dependencies>

<!-- Hibernate 5.2.12 Final dependency-->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.2.12.Final</version>

</dependency>

<!-- Hibernate Jcache -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-ehcache</artifactId>

<version>5.2.12.Final</version>

</dependency>

<!-- Ehcache 3.x -->

<dependency>

<groupId>org.ehcache</groupId>

<artifactId>ehcache</artifactId>

<version>3.4.0</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.10</version>

</dependency>

</dependencies>

**Database**

**Create table**

**Database Name:** cacheexample

CREATE TABLE `EMPLOYEE` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`first\_name` varchar(20) DEFAULT NULL,

`last\_name` varchar(20) DEFAULT NULL,

`salary` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=latin1;

**Hibernate.cfg.xml**

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE hibernate-configuration SYSTEM

"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<property name="hibernate.dialect">

org.hibernate.dialect.MySQLDialect

</property>

<property name="hibernate.connection.driver\_class">

com.mysql.jdbc.Driver

</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/cacheexample</property>

<property name="hibernate.connection.username">

root

</property>

<property name="hibernate.connection.password">

root

</property>

**<property name="hibernate.cache.region.factory\_class">org.hibernate.cache.ehcache.EhCacheRegionFactory </property>**

**<property name="hibernate.cache.use\_second\_level\_cache">true</property>**

<property name="show\_sql">true</property>

<mapping class="Employee" />

</session-factory>

</hibernate-configuration>

**HibernateConnector to create session facotory**

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;

public class HibernateConnector {

private static StandardServiceRegistry standardServiceRegistry;

private static SessionFactory sessionFactory;

public static SessionFactory getSessionFactory(){

try{

standardServiceRegistry = new StandardServiceRegistryBuilder()

.configure()

.build();

// Create MetadataSources

MetadataSources metadataSources = new MetadataSources(standardServiceRegistry);

// Create Metadata

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

// Create SessionFactory

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

return sessionFactory;

}catch (Throwable ex) {

System.err.println("Failed to create sessionFactory object." + ex);

throw new ExceptionInInitializerError(ex);

}

}

}

**Entity class Employee**

import javax.persistence.\*;

import org.hibernate.annotations.CacheConcurrencyStrategy;

@Entity

@Cacheable

@org.hibernate.annotations.Cache(usage = CacheConcurrencyStrategy.READ\_WRITE)

@Table(name = "EMPLOYEE")

public class Employee {

@Id @GeneratedValue

@Column(name = "id")

private int id;

@Column(name = "first\_name")

private String firstName;

@Column(name = "last\_name")

private String lastName;

@Column(name = "salary")

private int salary;

public Employee() {}

public int getId() {

return id;

}

public void setId( int id ) {

this.id = id;

}

public String getFirstName() {

return firstName;

}

public void setFirstName( String first\_name ) {

this.firstName = first\_name;

}

public String getLastName() {

return lastName;

}

public void setLastName( String last\_name ) {

this.lastName = last\_name;

}

public int getSalary() {

return salary;

}

public void setSalary( int salary ) {

this.salary = salary;

}

}

**EmployeeDAO class**

import java.util.Iterator;

import java.util.List;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

public class EmployeeDAO {

SessionFactory factory = HibernateConnector.getSessionFactory();

/\* Method to CREATE an employee in the database \*/

public Integer addEmployee(String fname, String lname, int salary){

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try{

tx = session.beginTransaction();

Employee employee = new Employee();

employee.setFirstName(fname);

employee.setLastName(lname);

employee.setSalary(salary);

employeeID = (Integer) session.save(employee);

tx.commit();

}catch (HibernateException e) {

if (tx!=null) tx.rollback();

e.printStackTrace();

}finally {

session.close();

}

return employeeID;

}

/\* Method to READ all the employees \*/

public void listEmployees( ){

Session session = factory.openSession();

session.beginTransaction();

List employees = session.createQuery("FROM Employee").list();

employees = session.createQuery("FROM Employee").list();

for (Iterator iterator =

employees.iterator(); iterator.hasNext();){

Employee employee = (Employee) iterator.next();

System.out.print("First Name: " + employee.getFirstName());

System.out.print(" Last Name: " + employee.getLastName());

System.out.println(" Salary: " + employee.getSalary());

}

session.getTransaction().commit();

session.close();

/\* //twice

Session session2 = factory.openSession();

List employees2 = session2.createQuery("FROM Employee").list();

for (Iterator iterator =

employees2.iterator(); iterator.hasNext();){

Employee employee = (Employee) iterator.next();

System.out.print("First Name: " + employee.getFirstName());

System.out.print(" Last Name: " + employee.getLastName());

System.out.println(" Salary: " + employee.getSalary());

}

session2.close();\*/

/\*Session session = factory.openSession();

Transaction tx = null;

try{

tx = session.beginTransaction();

List employees = session.createQuery("FROM Employee").list();

for (Iterator iterator =

employees.iterator(); iterator.hasNext();){

Employee employee = (Employee) iterator.next();

System.out.print("First Name: " + employee.getFirstName());

System.out.print(" Last Name: " + employee.getLastName());

System.out.println(" Salary: " + employee.getSalary());

}

tx.commit();

}catch (HibernateException e) {

if (tx!=null) tx.rollback();

e.printStackTrace();

}finally {

session.close();

}\*/

}

/\* Method to UPDATE salary for an employee \*/

public void updateEmployee(Integer EmployeeID, int salary ){

Session session = factory.openSession();

Transaction tx = null;

try{

tx = session.beginTransaction();

Employee employee =

(Employee)session.get(Employee.class, EmployeeID);

employee.setSalary( salary );

session.update(employee);

tx.commit();

}catch (HibernateException e) {

if (tx!=null) tx.rollback();

e.printStackTrace();

}finally {

session.close();

}

}

/\* Method to DELETE an employee from the records \*/

public void deleteEmployee(Integer EmployeeID){

Session session = factory.openSession();

Transaction tx = null;

try{

tx = session.beginTransaction();

Employee employee =

(Employee)session.get(Employee.class, EmployeeID);

session.delete(employee);

tx.commit();

}catch (HibernateException e) {

if (tx!=null) tx.rollback();

e.printStackTrace();

}finally {

session.close();

}

}

//primary cache, same session calling employee detail twice

public void getEmployee(Integer EmployeeID){

Session session = factory.openSession();

Transaction tx = null;

tx = session.beginTransaction();

Employee employee =

(Employee)session.get(Employee.class, EmployeeID);

Employee employee2 =

(Employee)session.get(Employee.class, EmployeeID);

System.out.println(employee.getFirstName());

tx.commit();

session.close();

}

public void getEmployeeSeconLevelCache(Integer EmployeeID){

Session session = factory.openSession();

Transaction tx = null;

tx = session.beginTransaction();

Employee employee =

(Employee)session.get(Employee.class, EmployeeID);

System.out.println(employee.getFirstName());

tx.commit();

session.close();

Session session2 = factory.openSession();

Employee employee2 =

(Employee)session2.get(Employee.class, EmployeeID);

}

}

**Main class**

import java.util.List;

import java.util.Date;

import java.util.Iterator;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.Transaction;

import org.hibernate.service.ServiceRegistry;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class ManageEmployee {

private static SessionFactory factory;

private static ServiceRegistry serviceRegistry;

public static void main(String[] args) {

EmployeeDAO employeeDao = new EmployeeDAO();

//employeeDao.getEmployee(4);//primary cache

employeeDao.getEmployeeSeconLevelCache(4);

}

}

**Steps to enable second level cache**

Entity class cacheble and mention strategy(read or write)

Mention cache provider in hibernate config file

Mention cache provider dependency in pom file