**\*\*\*\*\*\*\*\*\*\*\*\*\*Spring EL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*City.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.spel;

public class City {

private String name;

private int population;

private String state;

public String getState() {return state; }

public void setState(String state) { this.state = state; }

public String getName() { return name; }

public int getPopulation() { return population; }

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

public void setPopulation(int population) { this.population = population; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CityList.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.spel;

import java.util.ArrayList;

public class CityList {

private ArrayList<City> cityList;

public ArrayList<City> getCityList() { return cityList; }

public void setCityList(ArrayList<City> cityList) { this.cityList = cityList; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*Course.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.spel;

public class Course {

private String name;

private String trainer;

public String getName() { return name; }

public String getTrainer() { return trainer; }

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

public void setTrainer(String trainer) { this.trainer = trainer; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Employee.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.spel;

import java.util.ArrayList;

public class Employee {

private int eid;

private String enm;

private double esl;

private int eage;

private City eCity;

public City geteCity() { return eCity; }

public void seteCity(City eCity) { this.eCity = eCity; }

public int getEage() { return eage; }

public void setEage(int eage) { this.eage = eage; }

private City ecity;

private ArrayList<Course> crsList;

public int getEid() { return eid; }

public String getEnm() { return enm; }

public double getEsl() { return esl; }

public City getEcity() { return ecity; }

public ArrayList<Course> getCrsList() { return crsList; }

public void setEid(int eid) { this.eid = eid; }

public void setEnm(String enm) { this.enm = enm; }

public void setEsl(double esl) { this.esl = esl; }

public void setEcity(City ecity) { this.ecity = ecity; }

public void setCrsList(ArrayList<Course> crsList) { this.crsList = crsList; }}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*Person.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.spel;

public class Person {

private String firstName;

private String lastName;

private int age;

public void setFirstName(String firstName) { this.firstName = firstName; }

public String getFirstName() { return firstName; }

public void setLastName(String lastName) { this.lastName = lastName; }

public String getLastName() { return lastName; }

public String getPersonName() { return firstName + " " + lastName; }

public int getAge() { return age; }

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

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SpelClient.java\*\*\*\*\*\*\*\*\*\*\***

package com.igate.spel;

import java.util.ArrayList;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class SpelClient {

public static void main(String[] args) {

// TODO Auto-generated method stub

ApplicationContext appContext = new ClassPathXmlApplicationContext("SpEL.xml");

Employee e=(Employee)appContext.getBean("employee");

System.out.println(e.getEid());

System.out.println(e.getEnm());

System.out.println(e.getEsl());

System.out.println(e.getEage());

City city=e.geteCity();

System.out.println(city.getName());

CityList cL=(CityList)appContext.getBean("cList");

ArrayList<City> cList=cL.getCityList();

for(City c:cList)

{ System.out.println(c.getName()+" "+c.getPopulation()+" "+c.getState()); }

}}

**\*\*\*\*\*\*\*\*\*Spel.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:util=*"http://www.springframework.org/schema/util"*

xmlns:p=*"http://www.springframework.org/schema/p"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/util*

*http://www.springframework.org/schema/util/spring-util-3.0.xsd"*>

<context:annotation-config/>

<context:component-scan base-package=*"com.igate.spel"*/>

<util:list id=*"cities"* >

<bean class=*"com.igate.spel.City"*

p:name=*"Chicago"* p:state=*"IL"* p:population=*"100000"*/>

<bean class=*"com.igate.spel.City"*

p:name=*"Atlanta"* p:state=*"GA"* p:population=*"200002"*/>

<bean class=*"com.igate.spel.City"*

p:name=*"Dallas"* p:state=*"TX"* p:population=*"300000"*/>

</util:list>

<bean id=*"employee"* class=*"com.igate.spel.Employee"*>

<property name=*"eid"* value=*"#{10\*20}"* />

<!--<property name="eid" value="#{T(java.lang.Math).random() \* 100}"/>

<property name="enm" value="#{person.firstName}" /> -->

<property name=*"enm"* value=*"#{person.getLastName()}"* />

<property name=*"esl"* value=*"#{12345.56}"* />

<property name=*"eage"* value=*"#{person.age}"* />

<property name=*"eCity"* value=*"#{cities[2]}"* />

</bean>

<bean id=*"person"* class=*"com.igate.spel.Person"*>

<property name=*"firstName"* value=*"mohan"* />

<property name=*"lastName"* value=*"kumar"* />

<property name=*"age"* value=*"28"* />

</bean>

<bean id=*"city1"* class=*"com.igate.spel.City"*>

<property name=*"name"* value=*"Bangalore"* />

<property name=*"population"* value=*"6000000"* />

<property name=*"state"* value=*"Karnatake"* />

</bean>

<bean id=*"city2"* class=*"com.igate.spel.City"*>

<property name=*"name"* value=*"Mysore"* />

<property name=*"population"* value=*"1000000"* />

<property name=*"state"* value=*"Karnatake"* />

</bean>

<bean id=*"city3"* class=*"com.igate.spel.City"*>

<property name=*"name"* value=*"Pune"* />

<property name=*"population"* value=*"5000000"* />

<property name=*"state"* value=*"Maharashtra"* />

</bean>

<bean id=*"city4"* class=*"com.igate.spel.City"*>

<property name=*"name"* value=*"Chennai"* />

<property name=*"population"* value=*"8000000"* />

<property name=*"state"* value=*"Tamilnadu"* />

</bean>

<bean id=*"cList"* class=*"com.igate.spel.CityList"*>

<property name=*"cityList"*>

<list>

<value>#{city1}</value>

<value>#{city2}</value>

<value>#{city3}</value>

<value>#{city4}</value>

</list>

</property>

</bean>

</beans>

\*\*\*\*\*\*\*\*\*\*\*\*Output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Jun 23, 2015 11:31:42 AM org.springframework.context.support.AbstractApplicationContext prepareRefresh

INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@1010be6: startup date [Tue Jun 23 11:31:42 IST 2015]; root of context hierarchy

Jun 23, 2015 11:31:42 AM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions

INFO: Loading XML bean definitions from class path resource [SpEL.xml]

200

kumar

12345.56

28

Dallas

Bangalore 6000000 Karnatake

Mysore 1000000 Karnatake

Pune 5000000 Maharashtra

Chennai 8000000 Tamilnadu

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*src/User.Properties\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

username = ilearn

password = edude

userid=1001

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*src/userbean.xml\*\*\*\*\*\*\*\*\*\*\*\*\***

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:util=*"http://www.springframework.org/schema/util"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd*

*http://www.springframework.org/schema/util*

*http://www.springframework.org/schema/util/spring-util-3.0.xsd"*>

<context:component-scan base-package=*"training.spel1"* />

<util:properties id=*"userprops"* location=*"classpath:user.properties"* />

<bean id=*"u"* class=*"training.spel1.UserBean"*>

<property name=*"userId"* value=*"#{userprops.userid}"*/>

<property name=*"username"* value=*"#{userprops.username}"*/>

<property name=*"password"* value=*"#{userprops.password}"*/>

</bean>

<!--

Note: To use SpEL in annotation, you must register your component

via annotation (See the bean class). If you register your bean in XML

and define @Value in Java class, the @Value will fail to execute.

-->

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*UserBean.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package training.spel1;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.stereotype.Component;

@Component("user")

public class UserBean {

public int getUserId() { return userId; }

public void setUserId(int userId) { this.userId = userId; }

@Value("#{userprops.userid}")

private int userId;

@Value("#{userprops.username}")

//@Value("majrul")

private String username;

@Value("#{userprops.password}")

//@Value("majrul123")

private String password;

public String getPassword() { return password; }

public void setPassword(String password) { this.password = password; }

public String getUsername() { return username; }

public void setUsername(String username) { this.username = username; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*UserClient1.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package training.spel1;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class UserClient1 {

public static void main(String args[]) throws Exception {

ApplicationContext ctx = new ClassPathXmlApplicationContext("userbean.xml");

UserBean user = (UserBean) ctx.getBean("u");

//user.setUsername("Chennai");

System.out.println(user.getUserId());

System.out.println(user.getUsername());

System.out.println(user.getPassword());

}

}

**\*\*\*\*\*\*\*\*\*\*\*\* Output\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Jun 23, 2015 11:36:07 AM org.springframework.context.support.AbstractApplicationContext prepareRefresh

INFO: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@1010be6: startup date [Tue Jun 23 11:36:07 IST 2015]; root of context hierarchy

Jun 23, 2015 11:36:07 AM org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions

INFO: Loading XML bean definitions from class path resource [userbean.xml]

Jun 23, 2015 11:36:07 AM org.springframework.core.io.support.PropertiesLoaderSupport loadProperties

INFO: Loading properties file from class path resource [user.properties]

1001

ilearn

edude

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JDBC\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**13)\*\*\*\*\*** **013-Spring4JEmpJdbcProject\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Employee.java\*\*\*\*\*\*\*\*\*\*\***

package com.igate.jdbc.bean;

public class Employee {

private int eid;

private String enm;

private double esl;

public int getEid() { return eid; }

public void setEid(int eid) { this.eid = eid; }

public String getEnm() { return enm; }

public void setEnm(String enm) { this.enm = enm; }

public double getEsl() { return esl; }

public void setEsl(double esl) { this.esl = esl; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EmployeeDao.java\*\*\*\*\*\*\*\*\***

package com.igate.jdbc.dao;

import java.util.List;

import com.igate.jdbc.bean.Employee;

public interface EmployeeDao {

public int getCount();

public String getEmployeeName(int eid);

public int insertRec(int eid,String enm,double esl);

public int updateRec(int eid,String enm,double esl);

public List getAll();

public Employee getEmpByEid(int eid) ;

public List<Employee> getEmployeeList() ;

}

**\*\*\*\*\*\*\*\*\*\*EmployeeDaoImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.jdbc.dao;

import java.util.List;

import org.springframework.jdbc.core.JdbcTemplate;

import com.igate.jdbc.bean.Employee;

public class EmployeeDaoImpl implements EmployeeDao{

JdbcTemplate jdbcTemplate;

public JdbcTemplate getJdbcTemplate() { return jdbcTemplate; }

public void setjdbcTemplate(JdbcTemplate jdbcTemplate) {this.jdbcTemplate = jdbcTemplate; }

public int getCount()

{

int count=0;

count=jdbcTemplate.queryForObject("SELECT COUNT(\*) FROM employees",Integer.class);

return count;

}

public String getEmployeeName(int eid)

{

String sql = "SELECT ename FROM employees WHERE eid=?";

String name=getJdbcTemplate().queryForObject(sql,String.class,eid);

return name;

}

public int insertRec(int eid,String enm,double esl)

{

String sql="INSERT INTO employees VALUES(?,?,?)";

Object[] params=new Object[]{eid,enm,esl};

return jdbcTemplate.update(sql, params);

}

public int updateRec(int eid,String enm,double esl)

{

String sql="UPDATE employees SET ename=?,esal=? WHERE eid=?";

Object[] params = new Object[]{enm,esl,eid};

int update=jdbcTemplate.update(sql, params);

return update;

}

public List getAll()

{

Object[] params=new Object[]{new Double(40000.00)};

List list=jdbcTemplate.queryForList("SELECT \* FROM employees WHERE esal > ?",params);

return list;

}

// to retrieve the an employees given eid

public Employee getEmpByEid(int eid) {

String sql = "SELECT eid,ename,esal FROM employees WHERE eid=?";

Employee employees= (Employee) jdbcTemplate.queryForObject(sql,new EmployeeRowMapper(),eid);

return employees;

}

public List<Employee> getEmployeeList()

{

String sql = "SELECT \* FROM employees where esal>?";

List<Employee> eList =jdbcTemplate.query(sql,new EmployeeRowMapper(),4000);

return eList;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EmployeeRowMapper.java\*\*\*\*\*\*\*\*\*\*\***

package com.igate.jdbc.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import com.igate.jdbc.bean.Employee;

public class EmployeeRowMapper implements RowMapper {

@Override

public Object mapRow(ResultSet rs, int line) throws SQLException {

Employee e = new Employee();

e.setEid(rs.getInt(1));

e.setEnm(rs.getString(2));

e.setEsl(rs.getDouble(3));

return e;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EmployeeService.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.jdbc.service;

import java.util.List;

import com.igate.jdbc.bean.Employee;

public interface EmployeeService {

public int getCount();

public String getEmployeeName(int eid);

public int insertRec(int eid,String enm,double esl);

public int updateRec(int eid,String enm,double esl);

public List getAll();

public Employee getEmpByEid(int eid) ;

public List<Employee> getEmployeeList() ;

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*EmployeeServiceImpl.java\*\*\*\*\*\*\*\*\***

package com.igate.jdbc.service;

import java.util.List;

import com.igate.jdbc.bean.Employee;

import com.igate.jdbc.dao.EmployeeDao;

public class EmployeeServiceImpl implements EmployeeService {

EmployeeDao dao;

public void setDao(EmployeeDao dao) { this.dao = dao; }

@Override

public List getAll() {

return dao.getAll(); }

@Override

public int getCount() { return dao.getCount(); }

@Override

public Employee getEmpByEid(int eid) { return dao.getEmpByEid(eid); }

@Override

public List<Employee> getEmployeeList() { return dao.getEmployeeList(); }

@Override

public String getEmployeeName(int eid) {return dao.getEmployeeName(eid); }

@Override

public int insertRec(int eid, String enm, double esl) {return dao.insertRec(eid, enm, esl); }

@Override

public int updateRec(int eid, String enm, double esl) {return dao.updateRec(eid, enm, esl); }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*jdbc.xml\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean id=*"properties"*

class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"location"*>

<value>oracle.properties</value>

</property>

</bean>

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"${db.driver}"* />

<property name=*"url"* value=*"${db.url}"* />

<property name=*"username"* value=*"${db.user}"* />

<property name=*"password"* value=*"${db.password}"* />

</bean>

<bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<constructor-arg ref=*"dataSource"* />

</bean>

<bean id=*"employeeDao"* class=*"com.igate.jdbc.dao.EmployeeDaoImpl"*>

<property name=*"jdbcTemplate"* ref=*"jdbcTemplate"* />

</bean>

<bean id=*"employeeService"* class=*"com.igate.jdbc.service.EmployeeServiceImpl"*>

<property name=*"dao"* ref=*"employeeDao"* />

</bean>

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*oracle.properties\*\*\*\*\*\*\*\***

db.driver=oracle.jdbc.driver.OracleDriver

db.url=jdbc:oracle:thin:@127.0.0.1:1521:XE

db.user=system

db.password=root

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*JdbcClient.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.jdbc.pl;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import org.springframework.dao.DataAccessException;

import org.springframework.dao.DuplicateKeyException;

import org.springframework.dao.EmptyResultDataAccessException;

import com.igate.jdbc.bean.Employee;

import com.igate.jdbc.service.EmployeeService;

public class JdbcClient {

public static void main(String[] args) {

ApplicationContext bf = new ClassPathXmlApplicationContext("jdbc.xml");

EmployeeService service = (EmployeeService) bf

.getBean("employeeService");

//execute queryForInt

int count=service.getCount();

System.out.println("No of Employees in the table :"+count);

try

{

String name=service.getEmployeeName(1101);

System.out.println("Name "+name);

}

catch (EmptyResultDataAccessException e)

{

System.out.println("The Employee ID is invalid");

} catch (DataAccessException e) {

System.out.println(e.getMessage());

}

//execute update (insert new record) int

try

{

int insert=service.insertRec(1105,"test",55555.55);

if(insert>0)

System.out.println("Insert Success");

else

System.out.println("Insert Failed");

}catch (DuplicateKeyException e)

{

System.out.println("Employee Already Exist");

} catch (DataAccessException e)

{

System.out.println(e.getMessage());

}

List list=service.getAll();

if(list!=null)

{

System.out.println("List of Records");

for(Object emp:list) {

//Employee e=(Employee)emp;

System.out.println(emp);

}

}

List<Employee> eList=service.getEmployeeList();

if(eList!=null)

{

System.out.println("List of Employees"); for(Employee e:eList)

System.out.println(e.getEid()+" "+e.getEnm()+" "+e.getEsl());

}

//execute update (modify)

int update=service.updateRec(1102, "Shiv Kumar", 77777.77); if(update>0)

System.out.println("Update Success"); else

System.out.println("Update Failed");

//execute QueryForList

try

{

Employee employee=service.getEmpByEid(101);

System.out.println("Employee Info");

System.out.println("EID :"+employee.getEid());

System.out.println("ENM :"+employee.getEnm());

System.out.println("ESL :"+employee.getEsl());

}catch (EmptyResultDataAccessException e)

{

System.out.println("The Employee ID is invalid");

} catch (DataAccessException e) {

System.out.println(e.getMessage());

}

/\*List<Employee> eList = service.getEmployeeList();

System.out.println("List of Employees");

for (Employee e : eList)

System.out.println(e.getEid() + " ||||" + e.getEnm() + " "

+ e.getEsl());\*/

}

}

**14)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **014-Spring4JEmpDaocAnnoBootProfileProject\*\*\*\*\*\*\*\*\*\*\*\*\***

**…………………………………………………… EmployeeDaoImpl.java………………………………………………..**

package com.igate.jdbc.dao;

import java.util.List;

import javax.annotation.PostConstruct;

import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

import org.springframework.stereotype.Repository;

import com.igate.jdbc.bean.Employee;

@Repository("dao")

public class EmployeeDaoImpl extends JdbcDaoSupport implements EmployeeDao{

@Autowired

private DataSource myDataSource;

@PostConstruct

private void initialize() {

setDataSource(myDataSource);

}

public int getCount()

{

int count=0;

count=getJdbcTemplate().queryForObject("SELECT COUNT(\*) FROM employees",Integer.class);

return count;

}

public String getEmployeeName(int eid)

{

String sql = "SELECT ename FROM employees WHERE eid=?";

String name=getJdbcTemplate().queryForObject(sql,String.class,eid);

return name;

}

public int insertRec(int eid,String enm,double esl)

{

String sql="INSERT INTO employees VALUES(?,?,?)";

Object[] params=new Object[]{eid,enm,esl};

return getJdbcTemplate().update(sql, params);

}

public int updateRec(int eid,String enm,double esl)

{

String sql="UPDATE employees SET ename=?,esal=? WHERE eid=?";

Object[] params = new Object[]{enm,esl,eid};

int update=getJdbcTemplate().update(sql, params);

return update;

}

public List getAll()

{

Object[] params=new Object[]{new Double(40000.00)};

List list=getJdbcTemplate().queryForList("SELECT \* FROM employees WHERE esal > ?",params);

return list;

}

// to retrieve the an employee given eid

public Employee getEmpByEid(int eid) {

Employee emp=null;

int count=getJdbcTemplate().queryForObject("SELECT COUNT(\*) FROM employees WHERE eid=?", Integer.class,eid);

if(count>0)

{

String sql = "SELECT eid,ename,esal FROM employees WHERE eid=?";

emp= (Employee) getJdbcTemplate().queryForObject(sql,new EmployeeRowMapper(),eid);

}

return emp;

}

public List<Employee> getEmployeeList()

{

String sql = "SELECT \* FROM employees WHERE esal>?";

List<Employee> eList =getJdbcTemplate().query(sql,new EmployeeRowMapper(),4000);

return eList; }}

**……………………………………… DatabaseConfig.java……………………………………**

package com.igate.jdbc.utility;

import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Profile;

import org.springframework.context.annotation.PropertySource;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.stereotype.Component;

@Component

@PropertySource("classpath:oracle.properties")

@Profile("dev")

public class DatabaseConfig {

@Value("${db.driver}")

public String driver;

@Value("${db.url}")

public String url;

@Value("${db.user}")

public String username;

@Value("${db.password}")

public String password;

@Bean

@Qualifier("myDataSource")

public DataSource createDataSource()

{

DriverManagerDataSource dms=new DriverManagerDataSource();

dms.setDriverClassName(driver);

//dms.setDriverClassName("oracle.jdbc.driver.OracleDriver");

dms.setUrl(url);

dms.setUsername(username);

dms.setPassword(password);

return dms;

}

}

**……………………………JdbcClient.java……………………………………………………………**

package com.igate.jdbc.pl;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.core.env.AbstractEnvironment;

import com.igate.jdbc.service.EmployeeService;

@ComponentScan("com.igate")

@EnableAutoConfiguration

public class JdbcClient {

public static void main(String[] args) {

System.setProperty("spring.profiles.active", "dev");

ApplicationContext context = SpringApplication.run(JdbcClient.class, args);

EmployeeService service = (EmployeeService) context.getBean("service");

//execute queryForInt

int count =service.getCount();

System.out.println("No of Employees in the table :"+count);

}

}

**15)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*** **015-Spring4JEmpDaoProfileProject\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**…………….** EmployeeDaoImpl.java…………….

package com.igate.jdbc.dao;

import java.util.List;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

import com.igate.jdbc.bean.Employee;

public class EmployeeDaoImpl extends JdbcDaoSupport implements EmployeeDao{

public int getCount()

{

int count=0;

count=getJdbcTemplate().queryForObject("SELECT COUNT(\*) FROM employees",Integer.class);

return count;

}

public String getEmployeeName(int eid)

{

String sql = "SELECT ename FROM employees WHERE eid=?";

String name=getJdbcTemplate().queryForObject(sql,String.class,eid);

return name;

}

public int insertRec(int eid,String enm,double esl)

{

String sql="INSERT INTO employees VALUES(?,?,?)";

Object[] params=new Object[]{eid,enm,esl};

return getJdbcTemplate().update(sql, params);

}

public int updateRec(int eid,String enm,double esl)

{

String sql="UPDATE employees SET ename=?,esal=? WHERE eid=?";

Object[] params = new Object[]{enm,esl,eid};

int update=getJdbcTemplate().update(sql, params);

return update;

}

public List getAll()

{

Object[] params=new Object[]{new Double(40000.00)};

List list=getJdbcTemplate().queryForList("SELECT \* from employees WHERE esal > ?",params);

return list;

}

// to retrieve the an employee given eid

public Employee getEmpByEid(int eid) {

String sql = "SELECT eid,ename,esal FROM employees WHERE eid=?";

Employee employee= (Employee) getJdbcTemplate().queryForObject(sql,new EmployeeRowMapper(),eid);

return employee;

}

public List<Employee> getEmployeeList()

{

String sql = "SELECT \* FROM employees WHERE esal>?";

List<Employee> eList =getJdbcTemplate().query(sql,new EmployeeRowMapper(),4000);

return eList;

}

}

…………JdbcClient.java……………….

package com.igate.jdbc.pl;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.igate.jdbc.bean.Employee;

import com.igate.jdbc.service.EmployeeService;

public class JdbcClient {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.setProperty("spring.profiles.active", "oracle1");

ApplicationContext bf = new ClassPathXmlApplicationContext("jdbc.xml");

EmployeeService service = (EmployeeService) bf

.getBean("employeeService");

/\*

\* //execute queryForInt int count=service.getCount();

\* System.out.println("No of Employees in the table :"+count);

\*

\*

\* String name=service.getEmployeeName(101);

\* System.out.println("Name "+name);

\*

\*

\*

\* //execute update (insert new record) int

\* insert=service.insertRec(1105,"test",55555.55); if(insert>0)

\* System.out.println("Insert Success"); else

\* System.out.println("Insert Failed");

\*

\* List list=service.getAll(); System.out.println("List of Records");

\* for(Object emp:list) { //Employee e=(Employee)emp;

\* System.out.println(emp); }

\*

\*

\* List<Employee> eList=service.getEmployeeList();

\* System.out.println("List of Employees"); for(Employee e:eList)

\* System.out.println(e.getEid()+" "+e.getEnm()+" "+e.getEsl());

\*

\*

\* //execute update (modify) int update=service.updateRec(1102,

\* "Shiv Kumar", 77777.77); if(update>0)

\* System.out.println("Update Success"); else

\* System.out.println("Update Failed");

\*

\*

\*

\*

\* //execute QueryForList List list=service.getAll();

\* System.out.println("List of Records"); for(Object emp:list)

\* System.out.println(emp);

\*

\*

\* Employee employee=service.getEmpByEid(101);

\* System.out.println("Employee Info");

\* System.out.println("EID :"+employee.getEid());

\* System.out.println("ENM :"+employee.getEnm());

\* System.out.println("ESL :"+employee.getEsl());

\*/

List<Employee> eList = service.getEmployeeList();

System.out.println("List of Employees");

for (Employee e : eList)

System.out.println(e.getEid() + " ||||" + e.getEnm() + " "

+ e.getEsl());

}

}

……………….jdbc.xml…………………….

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean id=*"properties"*

class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"location"*>

<value>oracle.properties</value>

</property>

</bean>

<beans profile=*"oracle"*>

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"${db.driver}"* />

<property name=*"url"* value=*"${db.url}"* />

<property name=*"username"* value=*"${db.user}"* />

<property name=*"password"* value=*"${db.password}"* />

</bean>

<bean id=*"employeeDao"* class=*"com.igate.jdbc.dao.EmployeeDaoImpl"*>

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

<bean id=*"employeeService"* class=*"com.igate.jdbc.service.EmployeeServiceImpl"*>

<property name=*"dao"* ref=*"employeeDao"* />

</bean>

</beans>

<beans profile=*"oracle1"*>

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"oracle.jdbc.driver.OracleDriver"* />

<property name=*"url"* value=*"jdbc:oracle:thin:@127.0.0.1:1521:XE"* />

<property name=*"username"* value=*"system"* />

<property name=*"password"* value=*"root"* />

</bean>

<bean id=*"employeeDao"* class=*"com.igate.jdbc.dao.EmployeeDaoImpl"*>

<property name=*"dataSource"* ref=*"dataSource"* />

</bean>

<bean id=*"employeeService"* class=*"com.igate.jdbc.service.EmployeeServiceImpl"*>

<property name=*"dao"* ref=*"employeeDao"* />

</bean>

</beans>

</beans>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*User.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.bean;

public class User {

private String username;

private String password;

public String getUsername() { return username; }

public void setUsername(String username) { this.username = username; }

public String getPassword() { return password; }

public void setPassword(String password) { this.password = password; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UserDaoImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.List;

import javax.annotation.PostConstruct;

import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.jdbc.core.RowMapper;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

import org.springframework.stereotype.Repository;

import com.igate.bean.User;

@Repository("userDao")

public class UserDaoImpl extends JdbcDaoSupport implements UserDaoIntf {

@Autowired

private DataSource myDataSource;

@PostConstruct

private void initialize() {

setDataSource(myDataSource);

@Override

public List<User> displayUsers() {

List<User> users=getJdbcTemplate().query("select \* from users", new UserRowMapper());

return users;

}

}

**\*\*\*\*\*\*\* UserDaoIntf.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.dao;

import java.util.List;

import com.igate.bean.User;

public interface UserDaoIntf { public List<User> displayUsers();}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UserRowMapper.java\*\*\*\*\*\*\*\*\*\***

package com.igate.dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import org.springframework.jdbc.core.RowMapper;

import com.igate.bean.User;

public class UserRowMapper implements RowMapper {

public Object mapRow(ResultSet rs, int index) throws SQLException {

User user=new User();

user.setUsername(rs.getString(1));

user.setPassword(rs.getString(2));

return user;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Example.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.pl;

import java.util.List;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.igate.bean.User;

import com.igate.service.UserServiceIntf;

public class Example {

public static void main(String s[]) {

ApplicationContext ctx = new ClassPathXmlApplicationContext(

"config.xml");

UserServiceIntf ui = (UserServiceIntf) ctx.getBean("userService");

List<User> l = ui.displayUsers();

for (User u : l) {

System.out.println(u.getUsername());

}

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UserServiceImpl.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.igate.bean.User;

import com.igate.dao.UserDaoIntf;

@Service("userService")

public class UserServiceImpl implements UserServiceIntf {

@Autowired

UserDaoIntf userDao;

@Override

public List<User> displayUsers() { return userDao.displayUsers(); }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* UserServiceIntf.java\*\*\*\*\*\*\*\*\*\*\*\***

package com.igate.service;

import java.util.List;

import com.igate.bean.User;

public interface UserServiceIntf { public List<User> displayUsers();

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*config.xml\*\*\*\*\*\*\*\*\*\*\*\*\***

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xmlns:aop=*"http://www.springframework.org/schema/aop"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<context:component-scan base-package=*"com.igate"* />

<bean id=*"myDataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"oracle.jdbc.driver.OracleDriver"* />

<property name=*"url"* value=*"jdbc:oracle:thin:@127.0.0.1:1521:XE"* />

<property name=*"username"* value=*"system"* />

<property name=*"password"* value=*"root"* />

</bean>

</beans>

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**017)\*\*\*\*\*\*\*\*\*\*\* 017--Spring4JEmpDaoSupportAnnoBootLambda\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**…………………….Employee.java……………………………………………………………………………………………..**

package com.igate.jdbc.bean;

public class Employee {

private int eid;

private String enm;

private double esl;

public int getEid() { return eid; }

public void setEid(int eid) { this.eid = eid; }

public String getEnm() { return enm; }

public void setEnm(String enm) { this.enm = enm; }

public double getEsl() { return esl; }

public void setEsl(double esl) { this.esl = esl; }

@Override

public String toString() {return "Employee [eid=" + eid + ", enm=" + enm + ", esl=" + esl + "]"; }

}

**……………………………………………………………..………….EmployeeDao.java……………………………………………..**

package com.igate.jdbc.dao;

import java.util.List;

import com.igate.jdbc.bean.Employee;

public interface EmployeeDao { public List<Employee> getAllEmp();}

**……………………………………………………………………..…..EmployeeDaoImpl.java…………………**

package com.igate.jdbc.dao;

import java.util.List;

import javax.annotation.PostConstruct;

import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.jdbc.core.support.JdbcDaoSupport;

import org.springframework.stereotype.Repository;

import com.igate.jdbc.bean.Employee;

@Repository("dao")

public class EmployeeDaoImpl extends JdbcDaoSupport implements EmployeeDao{

@Autowired

private DataSource myDataSource;

@PostConstruct

private void initialize()

{setDataSource(myDataSource); }

public List<Employee> getAllEmp()

{

String sql="SELECT \* FROM employees";

List<Employee> empList =getJdbcTemplate().

query(sql, (rs,rowNum) ->{

Employee emp = new Employee();

emp.setEid(rs.getInt(1));

emp.setEnm(rs.getString(2));

emp.setEsl(rs.getDouble(3));

return emp;

});

return empList;

}

}

**....................................................................EmployeeService.java…………………………………..………**

package com.igate.jdbc.service;

import java.util.List;

import com.igate.jdbc.bean.Employee;

public interface EmployeeService {public List<Employee> getAllEmp();}

**……………………………………………………... EmployeeServiceImpl.java……………..**

package com.igate.jdbc.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.igate.jdbc.bean.Employee;

import com.igate.jdbc.dao.EmployeeDao;

@Service("service")

public class EmployeeServiceImpl implements EmployeeService {

@Autowired

EmployeeDao dao;

public void setDao(EmployeeDao dao) { this.dao = dao; }

public List<Employee> getAllEmp() { return dao.getAllEmp(); }

}

**…………………………………………………………….……DatabaseConfiguration.java………….**

package com.igate.jdbc.utility;

import javax.sql.DataSource;

import org.springframework.beans.factory.annotation.Qualifier;

import org.springframework.beans.factory.annotation.Value;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Profile;

import org.springframework.context.annotation.PropertySource;

import org.springframework.jdbc.datasource.DriverManagerDataSource;

import org.springframework.stereotype.Component;

@Component

@PropertySource("classpath:oracle.properties")

@Profile("dev")

public class DatabaseConfig {

@Value("${db.driver}")

public String driver;

@Value("${db.url}")

public String url;

@Value("${db.user}")

public String username;

@Value("${db.password}")

public String password;

@Bean

@Qualifier("myDataSource")

public DataSource createDataSource()

{

DriverManagerDataSource dms=new DriverManagerDataSource();

dms.setDriverClassName(driver);

//dms.setDriverClassName("oracle.jdbc.driver.OracleDriver");

dms.setUrl(url);

dms.setUsername(username);

dms.setPassword(password);

return dms;

}

}

**…………………………………………………………………JdbcClient.java………………………..**

package com.igate.jdbc.pl;

import java.util.List;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.core.env.AbstractEnvironment;

import com.igate.jdbc.bean.Employee;

import com.igate.jdbc.service.EmployeeService;

@ComponentScan("com.igate")

@EnableAutoConfiguration

public class JdbcClient {

public static void main(String[] args) {

System.setProperty("spring.profiles.active", "dev");

ApplicationContext context = SpringApplication.run(JdbcClient.class, args);

EmployeeService service = (EmployeeService) context.getBean("service");

//execute queryForInt

List<Employee> eList=service.getAllEmp();

System.out.println("Employees in the table :"+eList);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Thank You\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***