**SPRING**

* Open eclipse oxygen given by trainer
* Create new -> File -> Maven Project
* Select Default Workspace location -> next
* Select archetype -> quickstart -> next
* Give
  + Group Id: com.hexa
  + Artifact Id : BeanInjection
* Set the JDk for the Project
* Set the compiler 1.8
* Go to pom.xml
* Add the following data to the pom.xml

<properties>

<maven.compiler.source> 1.8 </maven.compiler.source>

<maven.compiler.target> 1.8 </maven.compiler.target>

</properties>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>4.3.2.RELEASE</version>

</dependency>

* Create a new folder under src -> main and give its name “resources” and copy the beans.xml provided by the trainer

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

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

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/tx*

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

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

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

</beans>

* In src/main/java create three packages
  + Com.hexa.client
  + Com.hexa.services
  + Com.hexa.dao
* In Dao package create an interface Idao

**package** com.hexa.dao;

**public** **interface** IDao {

**public** String viewData();

}

* Create a new class JdbcDaoImpl in dao package

**package** com.hexa.dao;

**public** **class** JdbcDaoImpl **implements** IDao{

**public** JdbcDaoImpl () {

System.***out***.println("Jdbc Dao Constructor");

}

@Override

**public** String viewData() {

**return** "jdbc";

}

}

Create a interface Iser in service package, create class SerImpl

**package** com.hexa.service;

**public** **interface** ISer {

**public** String getData();

}

**import** com.hexa.dao.IDao;

**public** **class** SerImpl **implements** ISer{

**private** IDao dao;

**public** SerImpl() {

System.***out***.println("Service Constructor fires");

}

// Setter is for spring IOC to do dependency Injection

**public** **void** setDao(IDao dao) {

System.***out***.println("Dao is Injected");

**this**.dao = dao;

}

@Override

**public** String getData() {

// **TODO** Auto-generated method stub

**return** dao.viewData();

}

}

In bean.xml add the following lines

<bean id = *"myser"* class=*"com.hexa.service.SerImpl"* >

<property name=*"dao"* ref = *"mydao"*></property> <!-- when dao is given here it will not call the field it will call the set dao method -->

</bean>

<bean id = *"myDao"* class=*"com.hexa.dao.JdbcDaoImpl"* ></bean>

In com.hexa.client create a new class ClientA

**package** com.hexa.client;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.hexa.service.ISer;

**public** **class** ClientA {

**private** **static** ApplicationContext *ctx*;

**static** {

*ctx* = **new** ClassPathXmlApplicationContext("beans.xml");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("Inside Main");

ISer ser = *ctx*.getBean("myser",ISer.**class**);

System.***out***.println(ser.getData());

System.***out***.println("-----Demand for the second time----");

ser = *ctx*.getBean("myser",ISer.**class**);

System.***out***.println(ser.getData());

}

}

o/P:

Inside Main

Service Constructor fires

Jdbc Dao Constructor

Dao is Injected

jdbc

-----Demand for the second time----

Jdbc

If lazy-init is set to false in beans.xml

Service Constructor fires

Jdbc Dao Constructor

Dao is Injected

Inside Main

jdbc

-----Demand for the second time----

Jdbc

If scope is set as ‘Prototype’ it ll create instance for each demand

<bean id = *"myser"* class=*"com.hexa.service.SerImpl"* scope=*"prototype"*>

<property name=*"dao"* ref = *"myDao"*></property> <!-- when dao is given here it will not call the field it will call the set dao method -->

</bean>

<bean id = *"myDao"* class=*"com.hexa.dao.JdbcDaoImpl"* scope=*"prototype"*></bean>

O/P:

Inside Main

Service Constructor fires

Jdbc Dao Constructor

Dao is Injected

jdbc

-----Demand for the second time----

Service Constructor fires

Jdbc Dao Constructor

Dao is Injected

Jdbc

If scope changed to singleton as below

<bean id = *"myser"* class=*"com.hexa.service.SerImpl"* scope=*"singleton"*>

<property name=*"dao"* ref = *"myDao"*></property> <!-- when dao is given here it will not call the field it will call the set dao method -->

</bean>

<bean id = *"myDao"* class=*"com.hexa.dao.JdbcDaoImpl"* scope=*"singleton"*></bean>

o/p:

Service Constructor fires

Jdbc Dao Constructor

Dao is Injected

Inside Main

jdbc

-----Demand for the second time----

Jdbc

**String Property Injection**

* File new-maven Proj
* Group Id: com.hexa
* ArtifactId: StringPopInjection
* Set jdk and compiler settings
* Create folder resources and put the beans.xml file onto it
* Add spring in pom.xml
* Create file hexa.txt

**Hexa.txt**

1001,ram,78

1004,tom,45

1003,ravi,98

1005,suresh,56

1002,peter,67

* Create 4 packages com.hexa.client,dao,entity,service
* Create an student entity class under entity package

**package** com.hexa.entity;

**public** **class** Student {

**private** **int** sId;

**private** String sName;

**private** **double** marks;

**public** Student(**int** sId, String sName, **double** marks) {

**super**();

**this**.sId = sId;

**this**.sName = sName;

**this**.marks = marks;

}

**public** **int** getsId() {

**return** sId;

}

**public** String getsName() {

**return** sName;

}

**public** **double** getMarks() {

**return** marks;

}

@Override

**public** String toString() {

**return** sId + " " + sName + " " + marks;

}

}

Under Dao package create EmpDao Interface with one method

**package** com.hexa.dao;

**import** java.util.List;

**import** com.hexa.entity.Student;

**public** **interface** EmpDao {

**public** List<Student> getStudents();

}

**package** com.hexa.dao;

**import** java.io.BufferedReader;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.util.ArrayList;

**import** java.util.List;

**import** com.hexa.entity.Student;

**public** **class** FileDaoImpl **implements** EmpDao{

**private** String fName;

**public** **void** setfName(String fName) {

**this**.fName = fName;

}

@Override

**public** List<Student> getStudents() {

List<Student> lst = **new** ArrayList<>();

Student stu = **null**;

// Java 1.7 feature

**try**(BufferedReader br = **new** BufferedReader(**new** FileReader(fName))) {

String line = **null**;

**while** ((line = br.readLine()) != **null**) {

String[] values = line.split(",");

stu = **new** Student(Integer.*parseInt*(values[0]), values[1], Integer.*parseInt*(values[3]));

lst.add(stu);

}

} **catch**(Exception e) {

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

}

**return** lst;

}

}

In com.hexa.service create interface anmd its implementation

**package** com.hexa.service;

**import** java.util.List;

**import** com.hexa.entity.Student;

**public** **interface** ISer {

**public** List<Student> orderByMarks();

}

**package** com.hexa.service;

**import** java.util.Comparator;

**import** java.util.List;

**import** java.util.stream.Collectors;

**import** com.hexa.dao.EmpDao;

**import** com.hexa.entity.Student;

**public** **class** SerImpl **implements** ISer{

**private** EmpDao dao;

**public** **void** setDao(EmpDao dao) {

**this**.dao = dao;

}

@Override

**public** List<Student> orderByMarks() {

Comparator<Student> c = (s1,s2) -> {

Integer m1 = s1.getMarks();

Integer m2 = s2.getMarks();

**return** m1.compareTo(m2);

};

List<Student> lst = dao.getStudents().stream().sorted(c).collect(Collectors.*toList*());

**return** lst;

}

}

**Bean.xml**

<bean id = *"myser"* class=*"com.hexa.service.SerImpl"*>

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

</bean>

<bean id = *"myDao"* class=*"com.hexa.dao.FileDaoImpl"*>

<property name=*"fName"* value = *"hexa.txt"*></property> <!-- value only for string and wrapper type -->

</bean>

**Client Code:**

**package** com.hexa.client;

**import** java.util.List;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.hexa.entity.Student;

**import** com.hexa.service.ISer;

**public** **class** ClientA {

**private** **static** ApplicationContext *ctx*;

**static** {

*ctx* = **new** ClassPathXmlApplicationContext("beans.xml");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("Inside Main");

ISer ser = *ctx*.getBean("myser",ISer.**class**);

List<Student> lst = ser.orderByMarks();

lst.forEach(System.***out***::println);

}

}

**Output:**

Inside Main

1004 tom 45

1005 suresh 56

1002 peter 67

1001 ram 78

1003 ravi 98

**Constructor Injection**

Inorder to obtain data from jdbc the above code has to be modified as follows

package com.hexa.dao;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import java.util.Properties;

import javax.swing.text.StyledEditorKit.ForegroundAction;

import com.hexa.entity.Student;

import com.mysql.jdbc.PreparedStatement;

public class JdbcDaoImpl implements EmpDao{

private Properties jprops;

public JdbcDaoImpl(Properties props) throws ClassNotFoundException {

this.jprops = props;

Class.forName(jprops.getProperty("driver"));

}

@Override

public List<Student> getStudents() {

List<Student> lst = new ArrayList<>();

Student stu;

String url = jprops.getProperty("url");

String uname = jprops.getProperty("user");

String password = jprops.getProperty("passwd");

try(Connection con = DriverManager.getConnection(url,uname,password)){

String sql = "Select \* from student";

java.sql.PreparedStatement st = con.prepareStatement(sql);

ResultSet rs = st.executeQuery();

while(rs.next()) {

stu = new Student(rs.getInt("stu\_id"), rs.getString("stu\_name"), rs.getInt("stu\_marks"));

lst.add(stu);

}

} catch(Exception e) {

}

return lst;

}

}

Bean.xml

<bean id=*"myser"* class=*"com.hexa.service.SerImpl"* lazy-init=*"true"*>

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

</bean>

<bean id=*"filedao"* class=*"com.hexa.dao.FileDaoImpl"* lazy-init=*"true"*>

<property name=*"fName"* value=*"hexa.txt"*></property> <!-- value only for string and wrapper type -->

</bean>

<bean id=*"jdbcdao"* class=*"com.hexa.dao.JdbcDaoImpl"* lazy-init=*"true"*>

<constructor-arg>

<props>

<prop key=*"driver"*>com.mysql.jdbc.Driver</prop>

<prop key=*"url"*>jdbc:mysql://localhost:3306/test</prop>

<prop key=*"user"*>root</prop>

<prop key=*"passwd"*>root</prop>

</props>

</constructor-arg>

</bean>

Pom.xml

<dependency>

<groupId>mysql</groupId>

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

<version>5.1.6</version>

</dependency>

**Another Example of injection**

* **create** a maven project. Do all the required settings initially
* create another file named beans2.xml along with bean.xml file
* create the 4 required packages

**package** com.hexa.dao;

**import** com.hexa.entity.Account;

**public** **interface** IDao {

**public** Account getAccount(**int** accountId);

}

**package** com.hexa.dao;

**import** java.util.Map;

**import** com.hexa.entity.Account;

**public** **class** DaoImpl **implements** IDao{

**private** Map<Integer,Account> amap;

**public** **void** setAmap(Map<Integer, Account> amap) {

**this**.amap = amap;

}

@Override

**public** Account getAccount(**int** accountId) {

**return** amap.get(accountId);

}

}

**package** com.hexa.service;

**public** **interface** ISer {

**public** **boolean** trasferFund(**int** from,**int** to,**int** doubleAmt);

}

**package** com.hexa.service;

**import** com.hexa.dao.IDao;

**import** com.hexa.entity.Account;

**public** **class** SerImpl **implements** ISer {

**private** IDao dao;

// Setter for IOC to do Dependency Injection

**public** **void** setDao(IDao dao) {

**this**.dao = dao;

}

@Override

**public** **boolean** trasferFund(**int** from, **int** to, **int** doubleAmt) {

Account acc1 = dao.getAccount(from);

Account acc2 = dao.getAccount(to);

**if** (acc1 == **null** || acc2 == **null**) {

**return** **false**;

}

**if** (acc1.getBalance() < doubleAmt) {

**return** **false**;

}

acc1.setBalance(acc1.getBalance() - doubleAmt);

acc2.setBalance(acc2.getBalance() + doubleAmt);

**return** **true**;

}

}

**package** com.hexa.entity;

**public** **class** Account {

**private** **int** aId;

**private** String cname;

**private** **double** balance;

**public** **int** getaId() {

**return** aId;

}

**public** String getCname() {

**return** cname;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

// used by spring IOC for dependency injection

**public** Account(**int** aId, String cname, **double** balance) {

**super**();

**this**.aId = aId;

**this**.cname = cname;

**this**.balance = balance;

}

@Override

**public** String toString() {

**return** aId + " " + cname + " " + balance;

}

}

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

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

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/tx*

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

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

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

<bean id=*"myser"* class=*"com.hexa.service.SerImpl"*>

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

</bean>

<bean id=*"myDao"* class=*"com.hexa.dao.DaoImpl"*>

<property name=*"amap"* >

<map>

<entry key=*"1001"* value-ref=*"acc1"*></entry>

<entry key=*"1002"* value-ref=*"acc2"*></entry>

</map>

</property>

</bean>

</beans>

Beans2.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:aop=*"http://www.springframework.org/schema/aop"*

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

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/tx*

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

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

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

<bean id=*"acc1"* class=*"com.hexa.entity.Account"* >

<constructor-arg value =*"1001"*></constructor-arg>

<constructor-arg value =*"Tom"*></constructor-arg>

<constructor-arg value =*"10000"*></constructor-arg>

</bean>

<bean id=*"acc2"* class=*"com.hexa.entity.Account"* >

<constructor-arg value =*"1002"*></constructor-arg>

<constructor-arg value =*"Ram"*></constructor-arg>

<constructor-arg value =*"4000"*></constructor-arg>

</bean>

</beans>

**package** com.hexa.client;

**import** java.util.List;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.hexa.dao.IDao;

**import** com.hexa.service.ISer;

**public** **class** ClientA {

**private** **static** ApplicationContext *ctx*;

**static** {

*ctx* = **new** ClassPathXmlApplicationContext("beans.xml","beans2.xml");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("Inside Main");

ISer ser = *ctx*.getBean("myser",ISer.**class**);

**boolean** res = ser.trasferFund(1001, 1002, 2000);

System.***out***.println("Fud transferred " + res);

IDao dao = *ctx*.getBean("myDao",IDao.**class**);

System.***out***.println(dao.getAccount(1001));

System.***out***.println(dao.getAccount(1002));

}

}

O/P:

Inside Main

Fud transferred true

1001 Tom 8000.0

1002 Ram 6000.0

Another Example

**package** com.hexa.service;

**import** java.util.Set;

**public** **interface** ISer {

Set<String> getCities();

}

package com.hexa.service;

import java.io.BufferedReader;

import java.io.FileReader;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.List;

import java.util.Set;

public class SerImpl implements ISer{

private List<String> fnames;

public void setFnames(List<String> fnames) {

this.fnames = fnames;

}

@Override

public Set<String> getCities() {

Set<String> set = new HashSet<String>();

for(String fname : fnames) {

set.addAll(readData(fname));

}

return set;

}

public List<String> readData(String fname){

List<String> lst = new ArrayList<>();

String line = null;

String arr[] = null;

try(BufferedReader br = new BufferedReader(new FileReader(fname))) {

while((line = br.readLine()) != null ) {

lst.add(line);

}

} catch (Exception e) {

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

}

return lst;

}

}

**package** com.hexa.client;

**import** java.util.List;

**import** java.util.Set;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** com.hexa.service.ISer;

**public** **class** ClientA {

**private** **static** ApplicationContext *ctx*;

**static** {

*ctx* = **new** ClassPathXmlApplicationContext("beans.xml");

}

**public** **static** **void** main(String[] args) {

System.***out***.println("Inside Main");

ISer ser = *ctx*.getBean("myser",ISer.**class**);

Set<String> cities = ser.getCities();

cities.forEach(System.***out***::println);

}

}

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

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

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/tx*

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

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

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

<bean id=*"myser"* class=*"com.hexa.service.SerImpl"*>

<property name=*"fnames"*>

<list>

<value>hexa1.txt</value>

<value>hexa2.txt</value>

<value>hexa3.txt</value>

</list>

</property>

</bean>

</beans>

O/P:

Inside Main

coimbatore

trichy

mumbai

bangalore

hyderabad

noida

pune

chennai

delhi