

Beans Autowiring/Beans Collaboration:

In general, in spring applications, if we want to inject dependent values in setter method dependency injection or in constructor dependency injection we have to use `<property>` or `<constructor-arg>` tags under `<bean>` tag. If we want to inject simple values like primitive values, string values then we have to use "value" attribute and if we want to inject Secondary data type elements like Objects then we have to use "ref" attribute or we have to use `<ref>` tag in beans configuration file.

In spring applications, if we want to inject dependent bean objects to another bean object automatically without providing `<property>` tags and `<constructor-arg>` tags then we have to use "Autowiring" feature.

"Autowiring" feature of spring framework will make the IOC Container to inject dependent objects to the bean objects automatically on the basis of the properties names or on the basis of properties types without checking `<property>` tags and `<constructor-arg>` tags.

There are four ways to manage autowiring in Spring applications.

- 1.XML Based Autowiring
- 2.Annotation Based Autowiring
- 3.Auto-Discovery[Stereo Types]
- 4.Java Based Autowiring

1.XML Based Auto wiring

In this approach, If we want to provide autowiring in spring applications then we have to use "autowire" attribute in `<bean>` tag

EX:

```
<bean id="--" class="--" autowire="value">
```

Here value may be either of the following "autowiring modes".

1.no

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

2.byName
3.byType
4.constructor

1.no

It is representing "no" autowiring for the beans injection, we must provide explicit configuration for the beans injection.

2.byName

It will provide autowiring on the basis of the properties names. In this autowiring mode, IOC Container will search for dependent bean objects by matching bean properties names with the identity values of the beans configuration in spring configuration file.

3.byType

It will provide autowiring on the basis of the properties data types. In this autowiring mode, IOC Container will identify the dependent bean objects by matching properties data types with the bean data types[class attribute values] in bean configuration.

Note: In Beans configuration file, only one bean definition must be existed with the same type , if we provide more than one bean configuration with the same type in beans configuration file then IOC Container will rise an exception.

4.constructor

It is same as "byType" autowiring mode, but, "byType" autowiring will provide setter method dependency injection and "constructor" autowiring mode will provide constructor dependency injection on the basis of the types.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Example:**Address.java**

```
package com.durgasoft.beans;

public class Address {
    private String hno;
    private String street;
    private String city;
    private String state;

    setXXX()
    getXXX()
}
```

Account.java

```
package com.durgasoft.beans;
public class Account {
    private String accNo;
    private String accName;
    private String accType;
    private long balance;

    setXXX()
    getXXX()
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Employee.java

```
package com.durgasoft.beans;
public class Employee {
    private String eid;
    private String ename;
    private Address eaddr;
    private Account eacc;

    setXXX()
    getXXX()

    public void getEmpDetails(){
        System.out.println("Employee Details");
        System.out.println("-----");
        System.out.println("Employee Id   :"+eid);
        System.out.println("Employee Name  :"+ename);
        System.out.println();
        System.out.println("Employee Address Details");
        System.out.println("-----");
        System.out.println("House Number:"+eaddr.getHno());
        System.out.println("Street       :"+eaddr.getStreet());
        System.out.println("City         :"+eaddr.getCity());
        System.out.println("State        :"+eaddr.getState());
        System.out.println();
        System.out.println("Employee Account Details");
        System.out.println("-----");
        System.out.println("Account Number :"+eacc.getAccNo());
        System.out.println("Account Name   :"+eacc.getAccName());
        System.out.println("Account Type   :"+eacc.getAccType());
        System.out.println("Account Balance:"+eacc.getBalance());
    }
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonline training@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

applicationContext.xml

```
<beans>
  <bean id="eaddr" class="com.durgasoft.beans.Address">
    <property name="hno" value="23/3rt"/>
    <property name="street" value="PS Road"/>
    <property name="city" value="Hyd"/>
    <property name="state" value="Tel"/>
  </bean>
  <bean id="eacc" class="com.durgasoft.beans.Account">
    <property name="accNo" value="abc123"/>
    <property name="accName" value="Durga"/>
    <property name="accType" value="Savings"/>
    <property name="balance" value="20000"/>
  </bean>
  <bean id="emp" class="com.durgasoft.beans.Employee" autowire="byName">
    <property name="eid" value="E-111"/>
    <property name="ename" value="Durga"/>
    <!--
    <property name="eaddr" ref="eaddr"/>
    <property name="eacc" ref="eacc"/>
    -->
  </bean>
</beans>
```

Test.java

```
package com.durgasoft.test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.durgasoft.beans.Employee;
public class Test {
public static void main(String[] args){
ApplicationContext context=new
ClassPathXmlApplicationContext("applicationContext.xml");
Employee emp=(Employee)context.getBean("emp");
```

CONTACT US:**Mobile: +91- 8885 25 26 27**Mail ID: durgasoftonlinetraining@gmail.com**+91- 7207 21 24 27/28**WEBSITE: www.durgasoftonline.com**US NUM: 4433326786****FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,**

```
emp.getEmpDetails();  
}  
}
```

If we want to provide example for "constructor" autowiring then we have to use the following components in the above example

Example:

Address.java

```
package com.durgasoft.beans;  
  
public class Address {  
    private String hno;  
    private String street;  
    private String city;  
    private String state;  
  
    setXXX()  
    getXXX()  
  
}
```

Account.java

```
package com.durgasoft.beans;  
public class Account {  
    private String accNo;  
    private String accName;  
    private String accType;  
    private long balance;  
  
    setXXX()  
    getXXX()  
  
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Employee.java

```
public class Employee {
    private String eid;
    private String ename;
    private Address eaddr;
    private Account eacc;

    public Employee(String eid, String ename, Address eaddr, Account eacc ){
        this.eid=eid;
        this.ename=ename;
        this.eaddr=eaddr;
        this.eacc=eacc;
    }

    public void getEmpDetails(){
        System.out.println("Employee Details");
        System.out.println("-----");
        System.out.println("Employee Id   :"+eid);
        System.out.println("Employee Name  :"+ename);
        System.out.println();
        System.out.println("Employee Address Details");
        System.out.println("-----");
        System.out.println("House Number:"+eaddr.getHno());
        System.out.println("Street       :"+eaddr.getStreet());
        System.out.println("City        :"+eaddr.getCity());
        System.out.println("State       :"+eaddr.getState());
        System.out.println();
        System.out.println("Employee Account Details");
        System.out.println("-----");
        System.out.println("Account NUmber :"+eacc.getAccNo());
        System.out.println("Account Name  :"+eacc.getAccName());
        System.out.println("Account Type  :"+eacc.getAccType());
        System.out.println("Account Balance:"+eacc.getBalance());
    }
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

applicationContext.xml

```
<beans>
    <bean id="eaddr" class="com.durgasoft.beans.Address">
        same as above
    </bean>
    <bean id="eacc" class="com.durgasoft.beans.Account">
        same as above
    </bean>
    <bean id="emp" class="com.durgasoft.beans.Employee"    autowire="constructor">

        <constructor-arg name="eid" value="E-111"/>
        <constructor-arg name="ename" value="Durga"/>
    </bean>
</beans>
```

Test.java

```
package com.durgasoft.test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.durgasoft.beans.Employee;
public class Test {
    public static void main(String[] args){
        ApplicationContext context=new
        ClassPathXmlApplicationContext("applicationContext.xml");
        Employee emp=(Employee)context.getBean("emp");
        emp.getEmpDetails();
    }
}
```

Note: If we want to block any bean object in autowiring then we have to use "autowire-candidate" attribute with "false" value in <bean> tag in beans configuration file.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

EX:

```
<beans>
<bean id="scourse" class="com.durgasoft.beans.Course"
  autore-wire-candidate="false">
  ----
</bean>
<bean id="student" class="com.durgasoft.beans.Student" autore-wire="byType">
  ----
</bean>
</beans>
```

Annotations for Autowiring:

To implement Autowiring in Spring applications with out providing autowiring configuration in spring configuration file , we have to use the following annotations provided by spring framework.

- 1.@Required
- 2.@Autowired
- 3.@Qualifier

1.@Required

This annotation will make IOC Container to inject a particular bean object in another bean object is mandatory. We have to use this annotation at method level, that is, just before setXXX() method. After providing this annotation, if we are not providing the respectiove bean injection then IOC Container will rise an exception .

2.@Autowired

This annotation is able to represent autowiring in bean classes, it will be used at method level, field level and local variables level in constructor dependency injection.

Note: If we provide "required" argument with "false" value in @Autowired annotation then it is not required to use @Required annotationi.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Note: This annotation is following "byType" autowiring internally in spring applications, If we want to use this annotation then we must have only one bean configuration with the respective type in configuration file, if we have more than one bean configuration with the same type then IOC Container will rise an exception.

3.@Qualifier

In the case of "byType" autowiring mode, that is, in the case of @Autowired annotation configuration file must provide only one bean configuration with the respective type, if we provide more than one bean configuration with the same type then IOC Container will rise an exception. In this context, to resolve the ambiguity of beans injection we have to use "@Qualifier" annotation, it will be used to specify a particular bean object among the multiple beans of the same type for injection.

EX: @Qualifier("bean_Identity")

Example:

Student.java

```
package com.durgasoft.beans;

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

public class Student {
    private String sid;
    private String sname;
    private Course scourse;

    public String getSid() {
        return sid;
    }

    public void setSid(String sid) {
        this.sid = sid;
    }
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
}  
public String getSname() {  
    return sname;  
}  
public void setSname(String sname) {  
    this.sname = sname;  
}  
  
public Course getScourse() {  
    return scourse;  
}  
@Autowired(required=true)  
//@Required  
@Qualifier("adv_Java")  
public void setScourse(Course scourse) {  
    this.scourse = scourse;  
}  
public void getStudentDetails(){  
    System.out.println("Student Details");  
    System.out.println("-----");  
    System.out.println("Student Id :"+sid);  
    System.out.println("Student Name :"+sname);  
    System.out.println("Course Details");  
    System.out.println("-----");  
    System.out.println("Course Id :"+scourse.getCid());  
    System.out.println("Course Name :"+scourse.getCname());  
    System.out.println("Course Cost :"+scourse.getCcost());  
}  
}  
  
Course.java  
  
package com.durgasoft.beans;  
public class Course {  
    private String cid;  
    private String cname;  
    private int ccost;
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
    setXXX()  
    getXXX()  
}
```

applicationContext.xml

```
<beans ----- >  
  <context:annotation-config/>  
  <bean id="core_Java" class="com.durgasoft.beans.Course">  
    <property name="cid" value="C-111"/>  
    <property name="cname" value="Core Java"/>  
    <property name="ccost" value="10000"/>  
  </bean>  
  <bean id="adv_Java" class="com.durgasoft.beans.Course">  
    <property name="cid" value="C-111"/>  
    <property name="cname" value="Adv Java"/>  
    <property name="ccost" value="20000"/>  
  </bean>  
  <bean id="std" class="com.durgasoft.beans.Student" >  
    <property name="sid" value="S-111"/>  
    <property name="sname" value="Durga"/>  
  </bean>  
</beans>
```

Test.java

```
package com.durgasoft.test;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import com.durgasoft.beans.Student;  
public class Test {  
  public static void main(String[] args)throws Exception {  
    ApplicationContext context=new  
    ClassPathXmlApplicationContext("applicationContext.xml");  
    Student std=(Student)context.getBean("std");  
    std.getStudentDetails();  
  }  
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
}  
}
```

If we want to use @Autowired annotation for constructor dependency injection then we have to use @Autowired annotation just above of the respective constructor and we have to use @Qualifier annotation along with with the Bean parameter in constructor.

Example:

Student.java

```
package com.durgasoft.beans;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.beans.factory.annotation.Required;  
  
public class Student {  
    private String sid;  
    private String sname;  
    private Course scourse;  
  
    @Autowired  
    public Student(String sid, String sname,  
        @Qualifier("adv_Java") Course scourse){  
        this.sid=sid;  
        this.sname=sname;  
        this.scourse=scourse;  
    }  
    public void getStudentDetails(){  
        System.out.println("Student Details");  
        System.out.println("-----");  
        System.out.println("Student Id  :"+sid);  
        System.out.println("Student Name :"+sname);  
        System.out.println("Course Details");  
        System.out.println("-----");  
    }  
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
        System.out.println("Course Id  :"+scourse.getCid());
        System.out.println("Course Name :"+scourse.getCname());
        System.out.println("Course Cost :"+scourse.getCcost());
    }
}
```

Course.java

```
package com.durgasoft.beans;
public class Course {
    private String cid;
    private String cname;
    private int ccost;

    setXXX()
    getXXX()
}
```

applicationContext.xml

```
<beans>
  <context:annotation-config/>
  <bean id="core_Java" class="com.durgasoft.beans.Course">
    <property name="cid" value="C-111"/>
    <property name="cname" value="Core Java"/>
    <property name="ccost" value="10000"/>
  </bean>
  <bean id="adv_Java" class="com.durgasoft.beans.Course">
    <property name="cid" value="C-111"/>
    <property name="cname" value="Adv Java"/>
    <property name="ccost" value="20000"/>
  </bean>
  <bean id="std" class="com.durgasoft.beans.Student" >
    <constructor-arg name="sid" value="S-111"/>
    <constructor-arg name="sname" value="Durga"/>
  </bean>
</beans>
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

</beans>

Test.java

```
package com.durgasoft.test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.durgasoft.beans.Student;
public class Test {
public static void main(String[] args)throws Exception {
ApplicationContext context=new
ClassPathXmlApplicationContext("applicationContext.xml");
Student std=(Student)context.getBean("std");
std.getStudentDetails();
}
}
```

3.Auto-Discovery[Stereo Types]

This mechanism will provide the autowiring beans objects with out using <bean> configuration in configuration file.

To use this mechanism in Spring applications then we have to use the following annotations provided by spring framework in the package "org.springframework.stereotype"

- 1.@Component: It will represent a component which is recognized by Spring Container.
- 2.@Repository: It will represent a class as Model Driven , that is, DAO.
- 3.@Service : It will represent a class as Service class.
- 4.@Controller: It will represent a class as Controller class, it will be used in Spring WEB-MVC Module.

Note: If we want to use these annotations in Spring applications then we must provide the following tag in spring configuration file.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
<context:component-scan base-package="---"/>
```

EX:

```
<context:component-scan base-package="com.durgasoft.service"/>
```

```
<context:component-scan base-package="com.durgasoft.dao"/>
```

```
<context:component-scan base-package="com.durgasoft.controller"/>
```

If we provide the above tag in spring configuration file then IOC Container will scan the specified packages and recognize the classes which are annotated with @Component, @Repository, @Service and @Controller then Container will create bean objects without checking beans configurations in configuration file.

Example:

AccountDao.java

```
package com.durgasoft.dao;
```

```
import com.durgasoft.dto.Account;
```

```
public interface AccountDao {  
    public String create(String accNo, String accName, String accType, int balance);  
    public String search(String accNo);  
    public Account getAccount(String accNo);  
    public String update(String accNo, String accName, String accType, int balance);  
    public String delete(String accNo);  
}
```

AccountDaoImpl.java

```
package com.durgasoft.dao;
```

```
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import org.springframework.stereotype.Repository;

import com.durgasoft.dto.Account;

import oracle.jdbc.pool.OracleDataSource;

//@Repository("accDao")
@Component("accDao")
public class AccountDaoImpl implements AccountDao {
    String status = "";
    @Autowired(required=true)
    private OracleDataSource dataSource;

    @Override
    public String create(String accNo, String accName, String accType, int balance) {
        try {
            Connection con = dataSource.getConnection();
            PreparedStatement pst = con.prepareStatement("select * from
account where accNo=?");
            pst.setString(1, accNo);
            ResultSet rs = pst.executeQuery();
            boolean b = rs.next();
            if(b == true) {
                status="existed";
            }else {
                pst = con.prepareStatement("insert into account
values(?,?,?,?)");
                pst.setString(1, accNo);
                pst.setString(2, accName);
                pst.setString(3, accType);
                pst.setInt(4, balance);
                pst.executeUpdate();
                status="success";
            }
        } catch (Exception e) {
            status="error";
        }
    }
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
    }
    } catch (Exception e) {
        status = "failure";
        e.printStackTrace();
    }
    return status;
}

@Override
public String search(String accNo) {
    try {
        Connection con = dataSource.getConnection();
        PreparedStatement pst = con.prepareStatement("select * from
account where accNo = ?");
        pst.setString(1, accNo);
        ResultSet rs = pst.executeQuery();
        boolean b = rs.next();
        if(b == true) {
            status =
"[ACCNO:"+rs.getString("ACCNO")+",ACCNAME:"+rs.getString("ACCNAME")+",ACCTYP
E:"+rs.getString("ACCTYPE")+",BALANCE:"+rs.getInt("BALANCE")+"]";
        }else {
            status = "Account Not Existed";
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return status;
}

@Override
public Account getAccount(String accNo) {
    Account acc = null;
    try {
        Connection con = dataSource.getConnection();
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
PreparedStatement pst = con.prepareStatement("select * from
account where accNO = ?");
pst.setString(1, accNo);
ResultSet rs = pst.executeQuery();
boolean b = rs.next();
if(b == true) {
    acc = new Account();
    acc.setAccNo(rs.getString("ACCNO"));
    acc.setAccName(rs.getString("ACCNAME"));
    acc.setAccType(rs.getString("ACCTYPE"));
    acc.setBalance(rs.getInt("BALANCE"));
} else {
    acc = null;
}
} catch (Exception e) {
    e.printStackTrace();
}
return acc;
}
@Override
public String update(String accNo, String accName, String accType, int balance) {
    try {
        Connection con = dataSource.getConnection();
        PreparedStatement pst = con.prepareStatement("update account set
ACCNAME = ?, ACCTYPE = ?, BALANCE = ? where ACCNO = ?");
        pst.setString(1, accName);
        pst.setString(2, accType);
        pst.setInt(3, balance);
        pst.setString(4, accNo);
        pst.executeUpdate();
        status = "success";
    } catch (Exception e) {
        status = "failure";
        e.printStackTrace();
    }
    return status;
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
}

@Override
public String delete(String accNo) {
    try {
        Connection con = dataSource.getConnection();
        PreparedStatement pst = con.prepareStatement("select * from
account where accNO = ?");
        pst.setString(1, accNo);
        ResultSet rs = pst.executeQuery();
        boolean b = rs.next();
        if(b == true) {
            pst = con.prepareStatement("delete from account where accNo
= ?");

            pst.setString(1, accNo);
            pst.executeUpdate();
            status = "success";
        }else {
            status = "notexisted";
        }
    } catch (Exception e) {
        status = "failure";
        e.printStackTrace();
    }
    return status;
}
}
```

AccountService.java

package com.durgasoft.service;

import com.durgasoft.dto.Account;

public interface AccountService {

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,


```
public String createAccount(String accNo, String accName, String accType, int
balance);
public String searchAccount(String accNo);
public Account getAccount(String accNo);
public String updateAccount(String accNo, String accName, String accType, int
balance);
public String deleteAccount(String accNo);
}
```

AccountServiceImpl.java

```
package com.durgasoft.service;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import com.durgasoft.dao.AccountDao;
import com.durgasoft.dto.Account;
```

```
@Service("accService")
```

```
public class AccountServiceImpl implements AccountService {
```

```
    @Autowired(required=true)
```

```
    private AccountDao dao;
```

```
    @Override
```

```
    public String createAccount(String accNo, String accName, String accType, int
balance) {
```

```
        return dao.create(accNo, accName, accType, balance);
```

```
    }
```

```
    @Override
```

```
    public String searchAccount(String accNo) {
```

```
        return dao.search(accNo);
```

```
    }
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
@Override
public Account getAccount(String accNo) {

    return dao.getAccount(accNo);
}
@Override
public String updateAccount(String accNo, String accName, String accType, int
balance) {

    return dao.update(accNo, accName, accType, balance);
}

@Override
public String deleteAccount(String accNo) {

    return dao.delete(accNo);
}
}
```

Account.java

```
-----
package com.durgasoft.dto;

public class Account {
    private String accNo;
    private String accName;
    private String accType;
    private int balance;

    public String getAccNo() {
        return accNo;
    }

    public void setAccNo(String accNo) {
        this.accNo = accNo;
    }
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
}  
public String getAccName() {  
    return accName;  
}  
public void setAccName(String accName) {  
    this.accName = accName;  
}  
public String getAccType() {  
    return accType;  
}  
public void setAccType(String accType) {  
    this.accType = accType;  
}  
public int getBalance() {  
    return balance;  
}  
public void setBalance(int balance) {  
    this.balance = balance;  
}  
}
```

applicationContext.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"  
    xsi:schemaLocation="http://www.springframework.org/schema/beans  
        http://www.springframework.org/schema/beans/spring-beans.xsd  
        http://www.springframework.org/schema/context  
        http://www.springframework.org/schema/context/spring-context.xsd">  
  
    <context:component-scan base-package="com.durgasoft.service"/>  
    <context:component-scan base-package="com.durgasoft.dao"/>
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
<bean id="dataSource" class="oracle.jdbc.pool.OracleDataSource">
    <property name="URL" value="jdbc:oracle:thin:@localhost:1521:xe"/>
    <property name="user" value="system"/>
    <property name="password" value="durga"/>
</bean>
```

```
</beans>
```

Test.java

```
package com.durgasoft.test;

import java.io.BufferedReader;
import java.io.InputStreamReader;

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

import com.durgasoft.dao.AccountDao;
import com.durgasoft.dto.Account;
import com.durgasoft.service.AccountService;

public class Test {

    public static void main(String[] args)throws Exception {
        ApplicationContext context = new
ClassPathXmlApplicationContext("applicationContext.xml");
        AccountService accService =
(AccountService)context.getBean("accService");
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

        while(true) {
            System.out.println();
            System.out.println("Account Operations Menu");
            System.out.println("1.Create Account");
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
System.out.println("2.Search Account");
System.out.println("3.Update Account");
System.out.println("4.Delete Account");
System.out.println("5.Exit");
System.out.print("Your Option :");
int option = Integer.parseInt(br.readLine());
String status = "";
String accNo = "", accName = "", accType = "";
int balance = 0;
switch(option) {
case 1:
    System.out.print("Account Number :");
    accNo = br.readLine();
    System.out.print("Account Name :");
    accName = br.readLine();
    System.out.print("Account Type :");
    accType = br.readLine();
    System.out.print("Balance :");
    balance = Integer.parseInt(br.readLine());

    status = accService.createAccount(accNo, accName, accType,
balance);
    if(status.equals("success")) {
        System.out.println("Account Created Successfully");
    }
    if(status.equals("failure")){
        System.out.println("Account Creation Failure");
    }
    if(status.equals("existed")) {
        System.out.println("Account Existed Already");
    }
    break;
case 2:
    System.out.print("Account Number :");
    accNo = br.readLine();
    status = accService.searchAccount(accNo);
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
System.out.println("Account Details :"+status);
break;
case 3:
    System.out.print("Account Number :");
    accNo = br.readLine();
    Account acc = accService.getAccount(accNo);
    if(acc == null) {
        System.out.println("Status :Account Not Existed");
    }else {
        Account acc_New = new Account();
        acc_New.setAccNo(accNo);
        System.out.print("Account Name : Old Value
:"+acc.getAccName()+" New Value :");
        String accName_New = br.readLine();
        if(accName_New == null || accName_New.equals("")) {
            acc_New.setAccName(acc.getAccName());
        }else {
            acc_New.setAccName(accName_New);
        }
        System.out.print("Account Type : Old Value
:"+acc.getAccType()+" New Value :");
        String accType_New = br.readLine();
        if(accType_New == null || accType_New.equals("")) {
            acc_New.setAccType(acc.getAccType());
        }else {
            acc_New.setAccType(accType_New);
        }
        System.out.print("Account Balance : Old Value
:"+acc.getBalance()+" New Value :");
        String bal = br.readLine();
        if(bal == null || bal.equals("")) {
            acc_New.setBalance(acc.getBalance());
        }else {
            int balance_New = Integer.parseInt(bal);
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,


```
acc_New.setBalance(balance_New);
    }
    status = accService.updateAccount(acc_New.getAccNo(),
acc_New.getAccName(), acc_New.getAccType(), acc_New.getBalance());
    if(status.equals("success")) {
        System.out.println("Account Updated
Successfully");
    }
    if(status.equals("failure")) {
        System.out.println("Account Updation Failure");
    }
}
break;
case 4:
    System.out.print("Account Number :");
    accNo = br.readLine();
    status = accService.deleteAccount(accNo);
    if(status.equals("success")) {
        System.out.println("Account Deleted Successfully");
    }
    if(status.equals("failure")) {
        System.out.println("Account Deletion Failure");
    }
    if(status.equals("notexisted")) {
        System.out.println("Account Not Existed");
    }
    break;
case 5:
    System.out.println("***** ThankQ for Using Account
Operations App*****");
    System.exit(0);
    break;
default:
    System.out.println("Enter Number from 1,2,3,4 and 5");
    break;
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
    }  
    }  
}
```

4.Java Based Autowiring

AccountDao.java

```
-----  
package com.durgasoft.dao;  
  
import com.durgasoft.dto.Account;  
  
public interface AccountDao {  
    public String create(String accNo, String accName, String accType, int balance);  
    public String search(String accNo);  
    public Account getAccount(String accNo);  
    public String update(String accNo, String accName, String accType, int balance);  
    public String delete(String accNo);  
}
```

AccountDaoImpl.java

```
package com.durgasoft.dao;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Component;  
import org.springframework.stereotype.Repository;  
  
import com.durgasoft.dto.Account;
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
import oracle.jdbc.pool.OracleDataSource;

//@Repository("accDao")
@Component("accDao")
public class AccountDaoImpl implements AccountDao {
    String status = "";
    @Autowired(required=true)
    private OracleDataSource dataSource;

    @Override
    public String create(String accNo, String accName, String accType, int balance) {
        try {
            Connection con = dataSource.getConnection();
            PreparedStatement pst = con.prepareStatement("select * from
account where accNo=?");
            pst.setString(1, accNo);
            ResultSet rs = pst.executeQuery();
            boolean b = rs.next();
            if(b == true) {
                status="existed";
            }else {
                pst = con.prepareStatement("insert into account
values(?,?,?,?)");
                pst.setString(1, accNo);
                pst.setString(2, accName);
                pst.setString(3, accType);
                pst.setInt(4, balance);
                pst.executeUpdate();
                status="success";
            }
        } catch (Exception e) {
            status = "failure";
            e.printStackTrace();
        }
        return status;
    }
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
}

@Override
public String search(String accNo) {
    try {
        Connection con = dataSource.getConnection();
        PreparedStatement pst = con.prepareStatement("select * from
account where accNo = ?");
        pst.setString(1, accNo);
        ResultSet rs = pst.executeQuery();
        boolean b = rs.next();
        if(b == true) {
            status =
"[ACCNO:"+rs.getString("ACCNO")+",ACCNAME:"+rs.getString("ACCNAME")+",ACCTYP
E:"+rs.getString("ACCTYPE")+",BALANCE:"+rs.getInt("BALANCE")+"]";
        }else {
            status = "Account Not Existed";
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return status;
}

@Override
public Account getAccount(String accNo) {
    Account acc = null;
    try {
        Connection con = dataSource.getConnection();
        PreparedStatement pst = con.prepareStatement("select * from
account where accNO = ?");
        pst.setString(1, accNo);
        ResultSet rs = pst.executeQuery();
        boolean b = rs.next();
        if(b == true) {
            acc = new Account();
        }
    }
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
        acc.setAccNo(rs.getString("ACCNO"));
        acc.setAccName(rs.getString("ACCNAME"));
        acc.setAccType(rs.getString("ACCTYPE"));
        acc.setBalance(rs.getInt("BALANCE"));
    }else {
        acc = null;
    }
} catch (Exception e) {
    e.printStackTrace();
}
return acc;
}
@Override
public String update(String accNo, String accName, String accType, int balance) {
    try {
        Connection con = dataSource.getConnection();
        PreparedStatement pst = con.prepareStatement("update account set
ACCNAME = ?, ACCTYPE = ?, BALANCE = ? where ACCNO = ?");
        pst.setString(1, accName);
        pst.setString(2, accType);
        pst.setInt(3, balance);
        pst.setString(4, accNo);
        pst.executeUpdate();
        status = "success";
    } catch (Exception e) {
        status = "failure";
        e.printStackTrace();
    }
    return status;
}

@Override
public String delete(String accNo) {
    try {
        Connection con = dataSource.getConnection();
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
PreparedStatement pst = con.prepareStatement("select * from
account where accNO = ?");
pst.setString(1, accNo);
ResultSet rs = pst.executeQuery();
boolean b = rs.next();
if(b == true) {
    pst = con.prepareStatement("delete from account where accNo
= ?");
    pst.setString(1, accNo);
    pst.executeUpdate();
    status = "success";
}else {
    status = "notexisted";
}
} catch (Exception e) {
    status = "failure";
    e.printStackTrace();
}
return status;
}
```

AccountService.java

```
package com.durgasoft.service;

import com.durgasoft.dto.Account;

public interface AccountService {
    public String createAccount(String accNo, String accName, String accType, int
balance);
    public String searchAccount(String accNo);
    public Account getAccount(String accNo);
    public String updateAccount(String accNo, String accName, String accType, int
balance);
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,


```
public String deleteAccount(String accNo);  
}
```

AccountServiceImpl.java

```
package com.durgasoft.service;  
  
import com.durgasoft.dto.Account;  
  
public interface AccountService {  
    public String createAccount(String accNo, String accName, String accType, int  
balance);  
    public String searchAccount(String accNo);  
    public Account getAccount(String accNo);  
    public String updateAccount(String accNo, String accName, String accType, int  
balance);  
    public String deleteAccount(String accNo);  
}
```

AccountServiceImpl.java

```
package com.durgasoft.service;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import com.durgasoft.dao.AccountDao;  
import com.durgasoft.dto.Account;  
  
@Service("accService")  
public class AccountServiceImpl implements AccountService {  
  
    @Autowired(required=true)  
    private AccountDao dao;  
    @Override
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
public String createAccount(String accNo, String accName, String accType, int
balance) {

    return dao.create(accNo, accName, accType, balance);
}

@Override
public String searchAccount(String accNo) {

    return dao.search(accNo);
}

@Override
public Account getAccount(String accNo) {

    return dao.getAccount(accNo);
}

@Override
public String updateAccount(String accNo, String accName, String accType, int
balance) {

    return dao.update(accNo, accName, accType, balance);
}

@Override
public String deleteAccount(String accNo) {

    return dao.delete(accNo);
}
}
```

Account.java

package com.durgasoft.dto;

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
public class Account {  
    private String accNo;  
    private String accName;  
    private String accType;  
    private int balance;  
  
    public String getAccNo() {  
        return accNo;  
    }  
    public void setAccNo(String accNo) {  
        this.accNo = accNo;  
    }  
    public String getAccName() {  
        return accName;  
    }  
    public void setAccName(String accName) {  
        this.accName = accName;  
    }  
    public String getAccType() {  
        return accType;  
    }  
    public void setAccType(String accType) {  
        this.accType = accType;  
    }  
    public int getBalance() {  
        return balance;  
    }  
    public void setBalance(int balance) {  
        this.balance = balance;  
    }  
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

AccountConfig.java

```
package com.durgasoft.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import com.durgasoft.dao.AccountDao;
import com.durgasoft.dao.AccountDaoImpl;
import com.durgasoft.service.AccountService;
import com.durgasoft.service.AccountServiceImpl;

import oracle.jdbc.pool.OracleDataSource;

@Configuration
public class AccountConfig {
    @Bean
    public OracleDataSource dataSource() {
        OracleDataSource dataSource = null;
        try {
            dataSource = new OracleDataSource();
            dataSource.setURL("jdbc:oracle:thin:@localhost:1521:xe");
            dataSource.setUser("system");
            dataSource.setPassword("durga");
        } catch (Exception e) {
            e.printStackTrace();
        }
        return dataSource;
    }

    @Bean
    public AccountService accService() {
        AccountService accService = new AccountServiceImpl();
        return accService;
    }
}
```

CONTACT US:**Mobile: +91- 8885 25 26 27****+91- 7207 21 24 27/28****US NUM: 4433326786**Mail ID: durgasoftonlinetraining@gmail.comWEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
@Bean
public AccountDao dao() {
    AccountDao dao = new AccountDaoImpl();
    return dao;
}
}
```

Drawbacks with Autowiring:

1. Autowiring is less exact than normal wiring.
2. Autowiring will not provide configuration metadata to the documentation tools to prepare Documentations.
3. Autowiring will increase confusion when we have more than one bean object of the same type in IOC Container.
4. In Spring applications, if we provide both explicit wiring and autowiring both at a time to a bean object injection then explicit wiring overrides autowiring configurations.
5. In Spring configuration files, Explicit wiring will improve readability when compared with autowiring.
6. In Spring applications, if we have more and more no of bean objects to inject there it is suggestible to use explicit wiring when compared with autowiring.
7. Autowiring is applicable for only bean objects injection, it is not applicable for simple values injection like primitive values, string values,....

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Method Injection

In Spring applications, by default, all the objects are singleton objects provided by IOC Container[ApplicationContext] . In Spring application , as part of dependency injection both container object and contained object are having same scope then application may not get any problem, if container object and contained objects are having different scopes then application may get problem.

EX: In Spring applications, if we inject Course object in Student Object , where if provide Singleton scope to Student object and Prototype scope to Course object then For every request for Student object single Student Object is created , along with single student object single Course object is created with out checking its scope "prototype" which we provided in spring configuration file, it is voilating spring scopes rules and regulations.

Student.java

```
public class Student{
-----
private Course scounrse;
-----
setXXX()
getXXX()
}
Course.java
-----
public class Course{
-----
}
```

spring_beans_config.xml

```
<beans>
<bean id="std" class="com.durgasoft.beans.Student" scope="singleton">
-----
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,


```
<property name="scourse" ref="scourse"/>
</bean>
<bean id="scourse" class="com.durgasoft.beans.Course" scope="prototype">
-----
</bean>
</beans>
```

Test.java

```
public class Test{
public static void main(String[] args){
ApplicationContext context=new ClasspathXmlApplicationContext("/com/
durgasoft/cfgs/spring_beans_config.xml");
Student std1=(Student)context.getBean("std");
Student std2=(Student)context.getBean("std");
System.out.println(std1);// a111
System.out.println(std2);// a111
System.out.println(std1.getScourse());// b111
System.out.println(std2.getScourse());// b111
}
}
```

To overcome the above problem Spring has provided no of solutions, where one of the solution is "Method Injection".

In Spring Framework, Method injection is available in two forms.

- 1.Lookup Method Injection
- 2.Arbitrary Method Replacement

1.Lookup Method Injection

In case of Look Method injection, we will declare an abstract class as a factory class and an abstract method as a factory method then we will give an intemation to IOC Container about to generate a sub class for the abstract class and an implementation for the abstract method dynamically.

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

In this context, IOC Container will prepare dynamic sub classes for abstract factory class and return that objects to the test application as per the requirement.

In Spring Framework, IOC Container will provide dynamic sub classes by using CGLib third party library which is managed by Spring framework internally.

To give an intimation to the IOC Container about the sub classes generation and implementation for Factory method by providing configuration details in spring configuration file.

To achieve the above requirement, we have to use "<lookup-method>" in beans configuration.

Example:

Account.java

```
package com.durgasoft.beans;
```

```
public interface Account {  
    public void create();  
    public void search();  
    public void update();  
    public void delete();  
}
```

CurrentAccount.java

```
package com.durgasoft.beans;
```

```
public class CurrentAccount implements Account {  
    public void create() {  
        System.out.println("Current Account is Created");  
    }  
    public void search() {  
        System.out.println("Current Account is Identified");  
    }  
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
public void update() {  
    System.out.println("Current Account is Updated");  
}  
public void delete() {  
    System.out.println("Current Account is Deleted");  
}  
}
```

SavingsAccount.java

```
package com.durgasoft.beans;  
public class SavingsAccount implements Account {  
    public void create() {  
        System.out.println("Savings Account is created");  
    }  
    public void search() {  
        System.out.println("Savings Account is identified");  
    }  
    public void update() {  
        System.out.println("Savings Account is Updated");  
    }  
    public void delete() {  
        System.out.println("Savings Account is Deleted");  
    }  
}
```

AccountFactory.java

```
package com.durgasoft.factory;  
import com.durgasoft.beans.Account;  
public abstract class AccountFactory {  
    public abstract Account getAccount();  
}
```

spring_beans_config.xml

```
<beans>
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
<bean id="savingsAccount" class="com.durgasoft.beans.SavingsAccount" />
<bean id="currentAccount" class="com.durgasoft.beans.CurrentAccount" />
<bean id="savingsAccountFactory"
class="com.durgasoft.factory.AccountFactory">
    <lookup-method name="getAccount" bean="savingsAccount"/>
</bean>
<bean id="currentAccountFactory" class="com.durgasoft.factory.AccountFactory">
    <lookup-method name="getAccount" bean="currentAccount"/>
</bean>
</beans>
```

Test.java

```
package com.durgasoft.test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.durgasoft.beans.CurrentAccount;
import com.durgasoft.beans.SavingsAccount;
import com.durgasoft.factory.AccountFactory;
public class Test {
public static void main(String[] args) {
ApplicationContext context=new
ClassPathXmlApplicationContext("/com/durgasoft/cfgs/spring_beans_config.xml");
AccountFactory
savingsAccountFactory=(AccountFactory)context.getBean("savingsAccountFactory");
SavingsAccount
savings_Account=(SavingsAccount)savingsAccountFactory.getAccount();
savings_Account.create();
savings_Account.search();
savings_Account.update();
savings_Account.delete();
System.out.println();

AccountFactory
currentAccountFactory=(AccountFactory)context.getBean("currentAccountFactory");
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
CurrentAccount
current_Account=(CurrentAccount)currentAccountFactory.getAccount();
current_Account.create();
current_Account.search();
current_Account.update();
current_Account.delete();
}
}
```

Method Replacement:

It is one type of method injection, where we will give an intimation to IOC Container about to replace the existed method implementation with some other method implementation.

To achieve Method Replacement, we have to declare an user defined bean class with a method whose implementation we want to replace and we will provide new implementation for that method by declaring a class and by implementing "MethodReplacer" interface.

MethodReplacer interface contains reimplement(-----) method, here we must provide our new implementation in reimplement() method.

Example:

Course.java

```
package com.durgasoft.beans;
public class Course {
    private String cid;
    private String cname;
    private int ccost;
    setXXX()
    getXXX()
    public void getCourseDetails(){
        System.out.println("Course Details");
    }
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

```
        System.out.println("-----");
        System.out.println("Course Id :"+cid);
        System.out.println("Course Name :"+cname);
        System.out.println("Course Cost :"+ccost);
    }
}
```

NewImpl.java

```
package com.durgasoft.beans;
import java.lang.reflect.Method;
import org.springframework.beans.factory.support.MethodReplacer;
public class NewImpl implements MethodReplacer {
    public Object reimplement(Object arg0, Method arg1, Object[] arg2) throws Throwable
    {
        System.out.println("Course Details");
        System.out.println("-----");
        System.out.println("Course Id :C-222");
        System.out.println("Course Name :.NET");
        System.out.println("Course Cost :20000");
        return null;
    }
}
```

spring beans config.xml

```
<beans>
<bean id="newImpl" class="com.durgasoft.beans.NewImpl"/>
<bean id="course" class="com.durgasoft.beans.Course">
    <property name="cid" value="C-111"/>
    <property name="cname" value="Java"/>
    <property name="ccost" value="10000"/>
    <replaced-method name="getCourseDetails" replacer="newImpl"/>
</bean>
</beans>
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,

Test.java

```
package com.durgasoft.test;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.durgasoft.beans.Course;
public class Test {
public static void main(String[] args)throws Exception {
ApplicationContext context=new
ClassPathXmlApplicationContext("/com/durgasoft/cfgs/spring_beans_config.xml");
Course crs=(Course)context.getBean("course");
crs.getCourseDetails();
}
}
```

CONTACT US:

Mobile: +91- 8885 25 26 27



+91- 7207 21 24 27/28



US NUM: 4433326786

Mail ID: durgasoftonlinetraining@gmail.com

WEBSITE: www.durgasoftonline.com

FLAT NO: 202, HMDA MYTRIVANUM, AMEERPET, HYDERABAD.,