**\*\*\*\*\*\*\*VaishaliSpring4.0BootProject\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*Address\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.bean;

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

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

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

import org.springframework.stereotype.Component;

//@Component("offAdd")

public class Address

{

// @Value("MS")

private String state;

// @Value("India")

private String country;

//@Value("67877")

private String zipcode;

public Address() { super(); }

//@Autowired

//@Qualifier("city1")

private City city;

@Override

public String toString() {

return "Address [state=" + state + ", country=" + country

+ ", zipcode=" + zipcode + ", myCity=" + city + "]";

}

public Address(String state, String country, String zipcode, City city) {

super();

this.state = state;

this.country = country;

this.zipcode = zipcode;

this.city = city;

}

public String getState() { return state; }

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

public String getCountry() { return country; }

public void setCountry(String country) { this.country = country; }

public String getZipcode() { return zipcode; }

public void setZipcode(String zipcode) { this.zipcode = zipcode; }

public City getCity() { return city; }

public void setCity(City city) { this.city = city; } }

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

package com.cg.bean;

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

import org.springframework.stereotype.Component;

@Component("city1")

//@Component //autowire byType

public class City

{

public City() { super(); }

@Value("Pune")

private String name;

@Value("3000000")

private int population;

public City(String name, int population) {

super();

this.name = name;

this.population = population;

}

public String getName() { return name; }

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

public int getPopulation() { return population; }

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

@Override

public String toString() {

return "City [name=" + name + ", population=" + population + "]";

} }

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CurrencyConvertorImpl .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.bean;

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

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

import org.springframework.stereotype.Component;

@Component("currencyConObj1")

public class CurrencyConvertorImpl implements ICurrencyConvertor

{

private int exchangeRate;

private String countryName;

@Override

public String toString() {

return "CurrencyConvertorImpl [exchangeRate=" + exchangeRate

+ ", countryName=" + countryName + "]";

}

@Override

public int yenToRupees(int yen)

{ return (exchangeRate\*yen); }

@Override

public double dollerToRupees(double dollers) { return (exchangeRate\*dollers);}

public CurrencyConvertorImpl(int exchangeRate)

{ super(); this.exchangeRate = exchangeRate; }

public CurrencyConvertorImpl() { super(); }

@Autowired

public CurrencyConvertorImpl(@Value("68")int exchangeRate,

@Value("India")String countryName)

{

super();

this.exchangeRate = exchangeRate;

this.countryName = countryName;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\* CustomDateEditorRegistrar\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.bean;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.beans.PropertyEditorRegistrar;

import org.springframework.beans.PropertyEditorRegistry;

import org.springframework.beans.propertyeditors.CustomDateEditor;

import org.springframework.stereotype.Component;

@Component

public class CustomDateEditorRegistrar implements

PropertyEditorRegistrar

{

@Override

public void registerCustomEditors(

PropertyEditorRegistry registry)

{

registry.registerCustomEditor(

Date.class,

new CustomDateEditor(

new SimpleDateFormat("dd-MM-yyyy"),

false));

}

}

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

package com.cg.bean;

import java.sql.Date;

import javax.annotation.Resource;

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

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

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

import org.springframework.context.annotation.Configuration;

import org.springframework.stereotype.Component;

@Component("empObj")

@Configuration

public class Employee

{

@Value("#{T(java.lang.Math).random()\*100}")

private int eid;

@Value("#{per.firstName}")

private String enm;

@Value("#{per.getAge()}")

private int eage;

@Autowired

//@Qualifier("homeAdd")

@Resource(name="homeAdd")

private Address empAddress;

@Value("getDoj()")

Date doj;

@Value("03-02-2013")

java.util.Date dob;

public Date getDoj() { return doj; }

public void setDoj(Date doj) { this.doj = doj; }

public java.util.Date getDob() { return dob; }

public void setDob(java.util.Date dob) {

this.dob = dob;

}

public Address getEmpAddress() { return empAddress; }

public void setEmpAddress(Address empAddress) {

this.empAddress = empAddress;

}

@Value("100000")

private double esl;

@Override

public String toString() {

return "Employee [eid=" + eid + ", enm=" + enm + ", eage=" + eage

+ ", empAddress=" + empAddress + ", doj=" + doj + ", dob="

+ dob + ", esl=" + 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 int getEage() { return eage; }

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

public double getEsl() { return esl; }

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

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ICurrencyConvertor.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.cg.bean;

**public** **interface** ICurrencyConvertor

{

**public** **double** dollerToRupees(**double** dollers);

**public** **int** yenToRupees(**int** yen);

}

**\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\* IGreet\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**package** com.cg.bean;

**public** **interface** IGreet

{ **public** String greetMe();

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\* NewYearWishGreetImpl.java\*\*\*\*\*\*\*\*\***

package com.cg.bean;

import javax.annotation.PostConstruct;

import javax.annotation.PreDestroy;

import org.springframework.beans.BeansException;

import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.BeanFactoryAware;

import org.springframework.beans.factory.BeanNameAware;

import org.springframework.beans.factory.DisposableBean;

import org.springframework.beans.factory.InitializingBean;

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

import org.springframework.stereotype.Component;

@Component("obj1")

public class NewYearWishGreetImpl

implements IGreet,BeanNameAware ,BeanFactoryAware ,

DisposableBean ,InitializingBean

{

@Value("Vaishali")

String name;

@Value("1999")

String year;

public String getYear() {

return year;

}

public void setYear(String year) {

this.year = year;

System.out.println("\*\*\*\*\*\*\*\*setYear() called and year initialized\*\*\*");

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

System.out.println("\*\*\*\*\*\*\*\*setName() called and name initialized\*\*\*");

}

public NewYearWishGreetImpl()

{

System.out.println("\*\*\*\*NewYearWishGreetImpl Bean Instantiated\*\*\* ");

}

@Override

public String greetMe()

{

return name +" Happy New Year "+year;

}

@Override

public void setBeanFactory(BeanFactory bf)

throws BeansException {

System.out.println("setBeanFactory callled \*\*\*\*\*");

}

@Override

public void setBeanName(String bn) {

System.out.println("setBeanName callled \*\*\*\*\*"+bn);

}

@PostConstruct

public void initGreetAfterCreation()

{

System.out.println(" \*\*After bean instantiated "+

"some initialization can be performed to get it to a usable state "+

" that time initGreetAfterCreation is called\*\*\*\*\*\*\*\*\*\*\*\*");

}

@PreDestroy

public void destroyGreetBeforeDeletion()

{

System.out.println(" \*\*When the bean is removed from"+

" the container, some cleanup may be required"+

" that time destroyGreetBeforeDeletion is called\*\*\*\*\*\*\*\*\*\*\*\*");

}

@Override

public void destroy() throws Exception {

System.out.println("bean is destroied from container\*\*\*\*\*\*");

}

@Override

public void afterPropertiesSet() throws Exception {

System.out.println(" This Is called After All PropertiesSet "); }}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NewYearWishPostProcessor.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.bean;

import org.springframework.beans.BeansException;

import org.springframework.beans.factory.config.BeanPostProcessor;

public class NewYearWishPostProcessor implements BeanPostProcessor {

@Override

public Object postProcessAfterInitialization(Object bean,

String beanName)

throws BeansException {

System.out.println("AfterInitialization : " + beanName);

Class cls=bean.getClass();

String type=cls.getName();

System.out.println(type);

return bean; // you can return any other object as well

}

@Override

public Object postProcessBeforeInitialization(Object bean, String beanName)

throws BeansException {

System.out.println("BeforeInitialization : " + beanName);

return bean; // you can return any other object as well

}

}

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

package com.cg.bean;

import java.util.ArrayList;

import javax.annotation.Resource;

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

import org.springframework.stereotype.Component;

@Component("per")

public class Person

{

@Value("Vaishali")

private String firstName;

@Value("Srivastava")

private String lastName;

@Value("80")

private int age;

@Resource(name="getAddList")

private ArrayList addList;

public int getAge() { return age; }

public ArrayList getAddList() { return addList; }

public void setAddList(ArrayList addList) { this.addList = addList; }

@Override

public String toString() {

return "Person [firstName=" + firstName + ", lastName=" + lastName

+ ", age=" + age + ", addList=" + addList + "]";

}

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

public String getFirstName() { return firstName; }

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

public String getLastName() { return lastName; }

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

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SpringConfig.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.bean;

import java.beans.PropertyEditor;

import java.beans.PropertyEditorSupport;

import java.util.\*;

import org.springframework.beans.PropertyEditorRegistrar;

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

import org.springframework.beans.factory.config.CustomEditorConfigurer;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;

@Configuration

@ComponentScan(basePackages="com.cg")

public class SpringConfig

{

@Bean

public City city2()

{

City myCity=new City("Gaziabad",500000);

return myCity;

}

@Bean(name="city3")

public City getCity3()

{

City myCity=new City("Kolkata",1000000);

return myCity;

}

@Bean

public Address homeAdd()

{

Address hA=new Address("UP","India","86777",new City("Gaziabad",45000));

return hA;

}

@Bean

public List<Address> getAddList()

{

City hometown=new City("Nagpur",40000);

Address homeA=new Address("MS","INDIA","600087",hometown);

City edutown=new City("Ramtek",20000);

Address eduA=new Address("MS","INDIA","678",edutown);

List<Address> addList=new ArrayList<Address>();

addList.add(homeA);

addList.add(eduA);

return addList;

}

@Bean

public static PropertySourcesPlaceholderConfigurer propertyConfigInDev() {

return new PropertySourcesPlaceholderConfigurer();

}

@Bean

public CustomEditorConfigurer customEditorConfigurer1 ()

{

CustomEditorConfigurer cec = new CustomEditorConfigurer();

cec.setPropertyEditorRegistrars(

new PropertyEditorRegistrar[]{

new CustomDateEditorRegistrar()});

return cec;

}

@Bean//(name="sqlDateObj")

public CustomEditorConfigurer customEditorConfigurer2 ()

{

Map<Class<?>, Class<? extends PropertyEditor>> customEditors =

new HashMap<Class<?>, Class<? extends PropertyEditor>>(1);

customEditors.put(java.sql.Date.class, SQLDateEditor.class);

CustomEditorConfigurer configurer = new CustomEditorConfigurer();

configurer.setCustomEditors(customEditors);

return configurer;

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SQLDateEditor.java\*\*\*\*\*\*\*\*\*\*\***

package com.cg.bean;

import java.beans.PropertyEditorSupport;

import java.sql.Date;

public class SQLDateEditor extends PropertyEditorSupport

{

public SQLDateEditor() { }

public void setAsText(String text) {

if(text.equalsIgnoreCase("getDoj()"))

{

Date date = new Date(System.currentTimeMillis());

setValue(date);

}

else

{

setValue(Date.valueOf(text));

}

}

public String getAsText() {

return getValue().toString();

}

}

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

package com.cg.bean;

import java.util.ArrayList;

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

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.PropertySource;

import org.springframework.context.annotation.PropertySources;

import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;

import org.springframework.stereotype.Component;

@Component("urObj")

@PropertySource("classpath:user.properties")

public class User

{

@Override

public String toString() {

return "User [userName=" + userName + ", password=" + password

+ ", userLocs=" + userLocs + "]";

}

@Value("${unm}")

private String userName;

@Value("${pwd}")

private String password;

private ArrayList<String> userLocs;

public ArrayList getUserLocs() { return userLocs; }

public void setUserLocs(ArrayList userLocs) { this.userLocs = userLocs; }

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; }

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TestSpelClientDemo .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.ui;

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.support.ClassPathXmlApplicationContext;

import com.cg.bean.Address;

import com.cg.bean.Employee;

import com.cg.bean.Person;

import com.cg.bean.User;

@ComponentScan(basePackages="com.cg")

public class TestSpelClientDemo {

public static void main(String[] args)

{

ApplicationContext ctx=SpringApplication.run(TestSpelClientDemo.class, args);

Address a1=(Address)ctx.getBean("homeAdd");

System.out.println(a1);

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

Person tempPer=(Person)ctx.getBean("per");

System.out.println(tempPer);

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

Employee tempEmp=(Employee)ctx.getBean("empObj");

System.out.println(tempEmp);

}

}

**\*\*\*\*\*\*\*\*\*\*\*\* TestSpringCurrClient .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.ui;

import org.springframework.boot.SpringApplication;

import org.springframework.context.\*;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.cg.bean.ICurrencyConvertor;

import com.cg.bean.IGreet;

import com.cg.bean.NewYearWishGreetImpl;

@ComponentScan("com.cg")

public class TestSpringCurrClient

{

public static void main(String[] args)

{

ApplicationContext ctx=SpringApplication.run(TestSpringCurrClient.class, args);

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

IGreet greet1=(IGreet)ctx.getBean("obj1");

System.out.println(greet1.greetMe());

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

ICurrencyConvertor currC1= (ICurrencyConvertor)ctx.

getBean("currencyConObj1");

System.out.println(" 50 Doller Is ? "+

currC1.dollerToRupees(50) + " RS. ");

/\*ICurrencyConvertor currC2= (ICurrencyConvertor)ctx.

getBean("currencyConObj2");

System.out.println(" 30 Yen Is ? "+

currC2.yenToRupees(30) + " RS. ");\*/

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TestUserClient .java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

package com.cg.ui;

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.context.support.ClassPathXmlApplicationContext;

import com.cg.bean.User;

@ComponentScan("com.cg")

public class TestUserClient {

public static void main(String[] args)

{

ApplicationContext ctx=SpringApplication.run(TestUserClient.class, args);

User u=(User)ctx.getBean("urObj");

System.out.println(" My User Id Id : "+

u.getUserName());

System.out.println(" My User Password Is : "+

u.getPassword());

}

}

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*src/userprop.properties\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

unm=CG

pwd=CG