**Spring Core Assignment**

1. For Setter Injection Main Class SpringQuestion1Test

Class1:SpringQuestion1

**Package** Spring\_Core;

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

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

**public class** SpringQuestion1

{

@SuppressWarnings("unused")

**Private static** ApplicationContext *context*;

**Public static void** main(String[] args)

{

@SuppressWarnings("resource")

ApplicationContextcontext = **new** ClassPathXmlApplicationContext("file:src/main/resources/Addressbean.xml");

Customer customer = (Customer) context.getBean("Customer");

System.***out***.println(customer);

}

}

Class2:Address

**Package** Spring\_Core;

**Public class** Address

{

**private** String city, street, state, country;

**private int** zip;

**public** String getCity()

{

**return** city;

}

**Public void** setCity(String city)

{

**this**.city = city;

}

**public** String getStreet()

{

**return** street;

}

**Public void** setStreet(String street)

{

**this**.street = street;

}

**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 int** getZip()

{

**return** zip;

}

**Public void** setZip(**int** zip)

{

**this**.zip = zip;

}

}

Xml File:AddressBean

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

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

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

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

<bean id=*"Address"* class=*"Spring\_Core.Address"*>

<property name=*"zip"* value=*"221"*/>

<property name=*"city"* value=*"London"*/>

<property name=*"street"* value=*"Baker Street"*/>

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

<property name=*"country"* value=*"England"*/>

</bean>

<bean id=*"Customer"* class=*"Spring\_Core.Customer"*>

<property name=*"customerId"* value=*"123"*/>

<property name=*"customercontact"* value=*"123456"*/>

<property name=*"customerAddress"* ref=*"Address"*/>

<property name=*"customerName"* value=*"Sherlock Holmes"*/>

</bean>

</beans>

Class:Customer

**Package** Spring\_Core;

**Public class** Customer

{

**Private int** customerId;

**Private int** customercontact;

**private** String customerName;

**private** Address customerAddress;

**public int** getCustomerId()

{

**return** customerId;

}

**Public void** setCustomerId(**int**customerId)

{

**this**.customerId = customerId;

}

**Public int** getCustomercontact()

{

**return** customercontact;

}

**Public void** setCustomercontact(**int** customercontact)

{

**this**.customercontact = customercontact;

}

**public** Address getCustomerAddress()

{

**return** customerAddress;

}

**Public void** setCustomerAddress(Address customerAddress)

{

**this**.customerAddress = customerAddress;

}

**public** String getCustomerName()

{

**return** customerName;

}

**Public void** setCustomerName(String customerName)

{

**this**.customerName = customerName;

}

**public** String toString()

{

**return**"Customer Id: "+getCustomerId()+ ", Customer Contact: "+getCustomercontact()+ ", Customer Address: {City: "+getCustomerAddress().getCity()+", Street: "+getCustomerAddress().getStreet()+", State: "+getCustomerAddress().getState()+", Country: "+getCustomerAddress().getCountry()+", Zip: "+getCustomerAddress().getZip()+"}, Customer Name: "+getCustomerName();

}

}

Class:SpringQuestion1

For Constructor Main Class SpringQuestion1Test

**Package** Spring\_Core;

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

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

**public class** SpringQuestion1

{

@SuppressWarnings("unused")

**Private static** ApplicationContext*context*;

**Public static void** main(String[] args)

{

@SuppressWarnings("resource")

ApplicationContextcontext1 = **new** ClassPathXmlApplicationContext("file:src/main/resources/Customer.xml");

Customer customer2 = (Customer)context1.getBean("Customer1");

System.***out***.println(customer2);

}

}

Class:Address

For Constructor Address.java

**Package** Spring\_Core;

**Public class** Address

{

**private** String city, street, state, country;

**private int** zip;

**public** String getCity()

{

**return** city;

}

**Public void** setCity(String city)

{

**this**.city = city;

}

**public** String getStreet()

{

**return** street;

}

**Public void** setStreet(String street)

{

**this**.street = street;

}

**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 int** getZip()

{

**return** zip;

}

**Public void** setZip(**int** zip)

{

**this**.zip = zip;

}

**Public** Address(String city, String street, String state, String country, **int**zip) {

**super**();

**this**.city = city;

**this**.street = street;

**this**.state = state;

**this**.country = country;

**this**.zip = zip;

}

}

Class:Customer

For Constructor Customer.java

**Package** Spring\_Core;

**Public class** Customer

{

**Private int** customerId;

**Private int** customercontact;

**private** String customerName;

**private** Address customerAddress;

**public** Customer(**int** customerId, **int** customercontact, String customerName, Address customerAddress)

{

**this**.customerId = customerId;

**this**.customercontact = customercontact;

**this**.customerName = customerName;

**this**.customerAddress = customerAddress;

}

**Public int** getCustomerId()

{

**return**customerId;

}

**Public void** setCustomerId(**int** customerId)

{

**this**.customerId = customerId;

}

**Public int** getCustomercontact()

{

**return**customercontact;

}

**Public void** setCustomercontact(**int** customercontact)

{

**this**.customercontact = customercontact;

}

**public** Address getCustomerAddress()

{

**return** customerAddress;

}

**Public void** setCustomerAddress(Address customerAddress)

{

**this**.customerAddress = customerAddress;

}

**public** String getCustomerName()

{

**return** customerName;

}

**Public void** setCustomerName(String customerName)

{

**this**.customerName = customerName;

}

**public** String toString()

{

**return**"Customer Id: "+getCustomerId()+ ", Customer Contact: "+getCustomercontact()+ ", Customer Address: {City: "+getCustomerAddress().getCity()+", Street: "+getCustomerAddress().getStreet()+", State: "+getCustomerAddress().getState()+", Country: "+getCustomerAddress().getCountry()+", Zip: "+getCustomerAddress().getZip()+"}, Customer Name: "+getCustomerName();

}

}

For Constructor Customer.xml

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

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

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

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

<bean id=*"Address"* class=*"Spring\_Core.Address"*>

<constructor-arg index=*"0"* name=*"city"* type=*"java.lang.String"* value=*"London"*/>

<constructor-arg index=*"1"* name=*"street"* type=*"java.lang.String"* value=*"Baker Street"*/>

<constructor-arg index=*"2"*name=*"state"*type=*"java.lang.String"*value=*"World"*/>

<constructor-arg index=*"3"*name=*"country"*type=*"java.lang.String"*value=*"England"*/>

<constructor-arg index=*"4"*name=*"zip"*type=*"int"*value=*"221"*/>

</bean>

<bean id=*"Customer1"*class=*"Spring\_Core.Customer"*>

<constructor-argindex=*"0"*name=*"customerId"*type=*"int"*value=*"123"*/>

<constructor-argindex=*"1"*name=*"customercontact"*type=*"int"*value=*"123456"*/>

<constructor-argindex=*"2"*name=*"customerName"*type=*"java.lang.String"*value=*"Sherlock Holmes"*/>

<constructor-argindex=*"3"*name=*"customerAddress"*ref=*"Address"*/>

</bean>

</beans>

Main Class Question2.java

**Package** Spring\_Core;

**importstatic** org.junit.jupiter.api.Assertions.\*;

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

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

**import** org.junit.jupiter.api.Test;

**class** Question2

{

@SuppressWarnings("unused")

**Private static** ApplicationContext *context*;

**Public static void** main(String[] args)

{

@SuppressWarnings("resource")

ApplicationContextcontext = **new**

ClassPathXmlApplicationContext("Coll.xml");

List list = (List) context.getBean("Collections");

System.***out***.println("List: " + list.getList());

}

}

Question2.java

**Package** Spring\_Core;

**Public class** Question

{

**Private int** questionId;

**private** String question, answers;

**public void** setQuestionId(**int** questionId)

{

**this**.questionId = questionId;

}

**Public void** setQuestion(String question)

{

**this**.question = question;

}

**Public void** setAnswers(String answers)

{

**this**.answers = answers;

}

**public** String toString()

{

**return** "Question ID: "+questionId+ ", Questions: "+question+ ", Answers: "+answers+"\n";

}

}

Class:List.java

package Spring\_Core;

**import** java.util.ArrayList;

**import** java.util.Map;

**import** java.util.Set;

**public** **class** List

{

**private** ArrayList<Object>list;

**private** Set<Object>set;

**private** Map<Object, Object>map;

**public** ArrayList<Object>getList()

{

**return** list;

}

**public** **void** setList(ArrayList<Object> list)

{

**this**.list = list;

}

**public** Set<Object>getSet()

{

**return** set;

}

**public** **void** setSet(Set<Object> set)

{

**this**.set = set;

}

**public** Map<Object, Object>getMap()

{

**return** map;

}

**public** **void** setMap(Map<Object, Object> map)

{

**this**.map = map;

}

}

Xml File:Coll.xml

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

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

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

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

xmlns:task=*"http://www.springframework.org/schema/task"*xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"Collections"* class=*"Spring\_Core.List"*>

<property name=*"list"*>

<list>

<ref bean=*"answers"*/>

<ref bean=*"answers1"*/>

<ref bean=*"answers2"*/>

</list>

</property>

</bean>

<bean id=*"answers"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*”Teja"*/>

<property name=*"question"* value=*"What's your name?"*/>

<property name=*"questionId"* value=*"123"*/>

</bean>

<bean id=*"answers1"*class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Ritu"*/>

<property name=*"question"* value=*"What's your name?"*/>

<property name=*"questionId"* value=*"456"*/>

</bean>

<bean id=*"answers2"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Manu"*/>

<property name=*"question"* value=*"What's your name?"*/>

<property name=*"questionId"* value=*"789"*/>

</bean>

</beans>

Main Class Question2.java

**Package** Spring\_Core;

**import static** org.junit.jupiter.api.Assertions.\*;

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

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

**import** org.junit.jupiter.api.Test;

**class** Question2

{

@SuppressWarnings("unused")

**Private static** ApplicationContext *context*;

**Public static void** main(String[] args)

{

@SuppressWarnings("resource")

ApplicationContextcontext = **new** ClassPathXmlApplicationContext("Coll.xml");

List set = (List) context.getBean("Collections");

System.***out***.println("Set : " + set.getSet());

}

}

Xml File:Coll.xml

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

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

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

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

xmlns:task=*"http://www.springframework.org/schema/task"*xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"Collections"* class=*"Spring\_Core.List"*>

<property name=*"set"*>

<set>

<ref bean=*"answers"*/>

<ref bean=*"answers1"*/>

<ref bean=*"answers2"*/>

</set>

</property>

</bean>

<bean id=*"answers"* class=*"Spring\_Core.Question"*>

<property name=*"answers"* value=*"Teja"*/>

<property name=*"question"* value=*"What's your name?"*/>

<property name=*"questionId"* value=*"123"*/>

</bean>

<bean id=*"answers1"*class=*"Spring\_Core.Question"*>

<property name=*"answers"*value=*"Ritu"*/>

<property name=*"question"*value=*"What's your name?"*/>

<property name=*"questionId"*value=*"456"*/>

</bean>

<bean id=*"answers2"*class=*"Spring\_Core.Question"*>

<property name=*"answers"*value=*"Manu"*/>

<property name=*"question"*value=*"What's your name?"*/>

<property name=*"questionId"*value=*"789"*/>

</bean>

</beans>

Main Class Question2.java

**Package** Spring\_Core;

**import static** org.junit.jupiter.api.Assertions.\*;

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

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

**import** org.junit.jupiter.api.Test;

**class** Question2

{

@SuppressWarnings("unused")

**Private static** ApplicationContext*context*;

**Public static void** main(String[] args)

{

@SuppressWarnings("resource")

ApplicationContextcontext = **new**

ClassPathXmlApplicationContext("Coll.xml");

List map = (List) context.getBean("Collections");

System.***out***.println("Map : " + map.getMap());

}

}

Xml File:Coll.xml

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

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

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

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

xmlns:task=*"http://www.springframework.org/schema/task"*xsi:schemaLocation=*"http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.2.xsd http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.2.xsd http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-3.2.xsd http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.2.xsd http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.2.xsd"*>

<bean id=*"Collections"*class=*"Spring\_Core.List"*>

<property name=*"map"*>

<map>

<entry key=*"1"*value-ref=*"answers"*/>

<entry key=*"2"*value-ref=*"answers1"*/>

<entry key=*"3"*value-ref=*"answers2"*/>

</map>

</property>

</bean>

<bean id=*"answers"*class=*"Spring\_Core.Question"*>

<property name=*"answers"*value=*"Teja"*/>

<property name=*"question"*value=*"What's your name?"*/>

<property name=*"questionId"*value=*"123"*/>

</bean>

<bean id=*"answers1"*class=*"Spring\_Core.Question"*>

<property name=*"answers"*value=*"Ritu"*/>

<property name=*"question"*value=*"What's your name?"*/>

<property name=*"questionId"*value=*"456"*/>

</bean>

<bean id=*"answers2"*class=*"Spring\_Core.Question"*>

<property name=*"answers"*value=*"Manu"*/>

<property name=*"question"*value=*"What's your name?"*/>

<property name=*"questionId"*value=*"789"*/>

</bean>

</beans>

2(D): Create a Test Class with main() method, get Question bean from ApplicationContext object and print question and its answers.

Class:BeanClass

**package** test;

**public** **class** BeanClass

{

**int** questionid;

String question;

String answers;

**public** **int** getQuestionid()

{

**return** questionid;

}

**public** **void** setQuestionid(**int** questionid)

{

**this**.questionid = questionid;

}

**public** String getQuestion()

{

**return** question;

}

**public** **void** setQuestion(String question)

{

**this**.question = question;

}

**public** String getAnswers()

{

**return** answers;

}

**public** **void** setAnswers(String answers)

{

**this**.answers = answers;

}

}

XML FILE:Beans.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-4.3.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd"

**default**-lazy-init="true">

<bean id="myid" **class**="test.BeanClass">

<property name="questionid" value="1"></property>

<property name="question" value="Java Spring"></property>

<property name="answer" value="Java Spring"></property>

</bean>

</beans>

Class:Tester

**package** test;

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

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

**public** **class** Tester

{

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

{

ApplicationContext context=**new** ClassPathXmlApplicationContext("Beans.xml");

BeanClass beanClass=(BeanClass) context.getBean("myid");

beanClass.getName();

}

}

4.Example of @Controller, @Service, @Repository, @Autowired, @Configuration and @Bean

Modify the above application,use annotations and java based configuration.

5. Write a program to demonstrate use of @Resource, @inject, @Required annontations

Class:Employee

**package** org.spring.app;

**import** javax.annotation.Resource;

**import** org.springframework.beans.factory.annotation.Required;

**public** **class** Employee

{

**private** String id;

**private** String name;

**private** String address;

@Required

**public** **void** setName(String name)

{

**this**.name = name;

}

**public** String getName()

{

**return** name;

}

@Inject

**private** Address address;

@Resource(name="mycompany")

**private** Company company;

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** Company getCompany() {

**return** company;

}

**public** **void** setCompany(Company company) {

**this**.company = company;

}

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

@Override

**public** String toString()

{

**return** "Employee [id=" + id + ", name=" + name + ", company=" + company.toString() + "]";

}

}

Class:AppMain

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

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

**import** pojo.Employee;

**public** **class** AppMain {

@SuppressWarnings("resource")

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

ApplicationContext ac = **new** ClassPathXmlApplicationContext("resource-annotation.xml");

Employee emp = ac.getBean("myemployee", Employee.**class**);

System.***out***.println(emp.toString());

}

}

Xml File:resource-annotation.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">

<!-- To activate the '@Resource' annotation in the spring framework -->

<context:annotation-config />

<bean id="mycompany" **class**="com.spring.pojo.Company">

<property name="name" value="Test Pvt. Ltd." />

<property name="location" value="India" />

</bean>

<bean id="myemployee" **class**="com.spring.pojo.Employee">

<property name="id" value="123456" />

<property name="name" value="Charlotte O' Neil" />

</bean>

<bean

**class**="org.springframework.beans.factory.

annotation.AutowiredAnnotationBeanPostProcessor">

</bean>

<bean **class**="org.springframework.beans.factory.annotation.RequiredAnnotationBeanPostProcessor">

</bean>

</beans>

7. Write a Java program to demonstrate SPEL(Spring Expression Language)

Test.java

**package** com.maven7;

**import** org.springframework.expression.Expression;

**import** org.springframework.expression.ExpressionParser;

**import** org.springframework.expression.spel.standard.SpelExpressionParser;

**public class** Test {

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

/\*

ExpressionParser parser = new SpelExpressionParser();

Expression exp = parser.parseExpression("'Hello Maven'");

String message = (String)exp.getValue();

System.out.println(message);

\*/

/\*

ExpressionParser parser = new SpelExpressionParser();

Expression exp = parser.parseExpression("'Welcome To'.concat('SpringExpression')");

String message = (String) exp.getValue(); System.out.println(message);

\*/

/\*

ExpressionParser parser = new SpelExpressionParser();

Expression exp = parser.parseExpression("'Hello Abhi'.bytes");

byte[] bytes = (byte[]) exp.getValue();

for(int i=0;i<bytes.length;i++) {

System.out.print(bytes[i]+" ");

}

\*/

/\*

ExpressionParser parser = new SpelExpressionParser();

Expression exp =

parser.parseExpression("new String('hello world').toUpperCase()");

String message = exp.getValue(String.class); System.out.println(message);

\*/

ExpressionParser parser = **new** SpelExpressionParser();

Expression exp = parser.parseExpression("'Hello Abhi'.bytes.length");

**int**length = (Integer) exp.getValue();

System.***out***.println(length);

}

}

Pom.xml

<projectxmlns="http://maven.apache.org/POM/4.0.0"xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.maven7</groupId>

<artifactId>SPELexpression</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.3.15</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-expression -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-expression</artifactId>

<version>5.3.15</version>

</dependency>

</dependencies>

</project>

OUTPUT:

Hello Maven

Concat : Welcome To SpringExpression

StringtoByte : 72 101 108 108 111 32 87 111 114 108 100

UpperCase : HELLO WORLD

Length : 10

8. Write a java program todemonstrate InitializingBean and DisponsableBean.

TRY different Ways:

(usinginit-method and Destroy method xml config file )

(Use @PostConstruct and @preDestroy)

**1 Way:**

**Class 1:CustomerService**

**Package** QuestionEight;

**Import** org.springframework.beans.factory.DisposableBean;

**Import** org.springframework.beans.factory.InitializingBean;

**Public class** CustomerService **implements** InitializingBean, DisposableBean

{

**private** String msg;

**public** String getMsg()

{

**Return** msg;

}

**Public void** setMsg(String msg)

{

**this**.msg = msg;

}

**Public void** destroy() **throws** Exception

{

System.***out***.println("Spring Container is destroy! Customer clean up");

}

**Public void** afterPropertiesSet() **throws** Exception

{

System.***out***.println("Init method after properties are set : " + msg);

}}

**Class 2: Snippet**

**Package** QuestionEight;

**import** org.springframework.context.ConfigurableApplicationContext;

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

**public class** Snippet

{

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

{

ConfigurableApplicationContext context = **new** ClassPathXmlApplicationContext(**new** String[] { "SpringCustomer.xml" });

CustomerServicecust = (CustomerService) context.getBean("customerService");

System.***out***.println(cust);

context.close();

}

}

**XML FILE**

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

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

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

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

*http://www.springframework.org/schema/beans/spring-beans-2.5.xsd"*>

<bean id=*"customerService"* class=*"QuestionEight.CustomerService"*>

<property name=*"msg"* value=*"i'm property message"*/>

</bean>

</beans>

OUTPUT:

Init method after properties are set :i'm property message

Spring Container is destroy! Customer clean up

**2nd Way:**

**Class1: MyBean**

**Package** EightB;

**import** javax.annotation.PostConstruct;

**import** javax.annotation.PreDestroy;

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

**import** org.springframework.stereotype.Component;

@Component

**Public class** MyBean {

**public void** MyBean()

{

System.***out***.println("MyBean instance created");

}

@PostConstruct

**Private void** init()

{

System.***out***.println("Verifying Resources");

}

@PreDestroy

**Private void** shutdown()

{

System.***out***.println("Shutdown All Resources");

}

**Public void** close()

{

System.***out***.println("Closing All Resources");

}

}

**Class2:MyConfiguration**

**Package** EightB;

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

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

**import** org.springframework.context.annotation.Scope;

@Configuration

**Public class** MyConfiguration

{

@Bean

@Scope(value = "singleton")

**Public** MyBeanmyBean()

{

**return new** MyBean();

}

}

**Class3:SpringApp**

**Package** EightB;

**import** org.springframework.context.annotation.AnnotationConfigApplicationContext;

**public class** SpringApp {

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

{

AnnotationConfigApplicationContextctx = **new**

AnnotationConfigApplicationContext();

ctx.register(MyConfiguration.**class**);

ctx.refresh();

MyBean mb1 = ctx.getBean(MyBean.**class**);

System.***out***.println(mb1.hashCode());

MyBean mb2 = ctx.getBean(MyBean.**class**);

System.***out***.println(mb2.hashCode());

ctx.close();

}

}

**OUTPUT:**

MyBean instance created

Verifying Resources

2145970759

2145970759

Shutdown All Resources

Closing All Resources

**9. Write a Java program to demonstrate complete Bean Life Cycle .**

**Class1:HelloWorld**

**Public class** HelloWorld

{

**Public void** init() **throws** Exception

{

System.***out***.println("Bean HelloWorld has been "

+ "instantiated and I'm "

+ "the init() method");

}

// This method executes

// when the spring container

// is closed

**publicvoid** destroy() **throws** Exception

{

System.***out***.println( "Container has been closed "

+ "and I'm the destroy() method");

}

}

Class2: Main method

**import** org.springframework.context.ConfigurableApplicationContext;

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

**public class** Client {

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

{

// Loading the Spring XML configuration

// file into the spring container and

// it will create the instance of

// the bean as it loads into container

ConfigurableApplicationContext cap = **new** ClassPathXmlApplicationContext("spring.xml");

// It will close the spring container

// and as a result invokes the

// destroy() method

cap.close();

}

}

XML file

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

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

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

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

<bean id=*"HelloBean"* class=*"springcore\_example.HelloBean"*>

<property name=*"name"* value=*"Manisha"*>

</property>

</bean>

</beans>

**OUTPUT:**

Bean HelloWorld has been instantiated and I'm the init() method

Container has been closed and I'm the destroy() method

**10. Write a Java program to demonstrate ApplicationContextAwareinterface .**

**Class1:Employee**

**Package** AplicationContextAware;

**Public class** Employee

{

**private** String Name;

**public** String getName()

{

**return** Name;

}

**Public void** setName(String name)

{

Name = name;

}

@Override

**public** String toString()

{

**return** "employee [Name=" + Name + "]";

}

}

Class 2: ApplicationContextAware interface is used

**package** org.spring.app;

**import** org.springframework.beans.BeansException;

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

**import** org.springframework.context.ApplicationContextAware;

**public** **class** AppContextAwareImpl **implements** ApplicationContextAware {

**private** ApplicationContextapplicationContext;

**public** **void** setApplicationContext(ApplicationContextapplicationContext) **throws** BeansException

{

System.out.println("Set Application Context method is called by the spring container");

**this**.applicationContext = applicationContext;

}

**public** **void** displayEmployeeDetails()

{

Employee employee = applicationContext.getBean("employee", Employee.**class**);

System.***out***.println("Got employee object from the applicationContext(Spring Container)=" + employee);

System.***out***.println("Is employee object Singleton =" + applicationContext.isSingleton("employee"));

}

}

**Class3:App**

**Package** AplicationContextAware;

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

**public class** App {

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

{

ClassPathXmlApplicationContext applicationContext= **new**

ClassPathXmlApplicationContext("applicationContext.xml");

AppContextAwareImplapplicationContextAwareImpl= applicationContext.getBean("applicationContextAware",AppContextAwareImpl.**class**);

applicationContextAwareImpl.displayEmployeeDetails();

applicationContext.close();

}

}

**XML file**

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

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

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

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

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

<bean id=*"employee"*class=*"AplicationContextAware.Employee"*>

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

</bean>

<beanid=*"applicationContextAware"*class=*"AplicationContextAware.AppContextAwareImpl"*></bean>

</beans>

**OUTPUT:**

Set Application Context method is called by the spring container

Got employee object from the applicationContext(Spring Container)=employee [Name=peter]

Is employee object Singleton =true