**Steps to create the spring application**

Pre Requisites

JDK 1.8

MAVEN 3.8.1

IDE Eclipse || STS

STEP-1:

Create new Maven Project

STEP-2:

Pom.XML

Properties and Dependencies to be used in the project updation

<properties>

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

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

<spring.version>5.3.18</spring.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- Spring ORM -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

</dependency>

</dependencies>

STEP-3 : Installing Dependencies for the first time in the cmd prompt

Mvn clean install (.m2 local repo)

STEP-4 :

Create the appropriate packages (model, main, service, config etc)

STEP -5:

Create the POJO files in the model package

**package** com.model;

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

**private** **int** studentId;

**private** String studentName;

**private** **float** studentMark;

**public** Student() {

**super**();

System.***out***.println("Student Model Constructor");

}

**public** Student(**int** studentId, String studentName, **float** studentMark) {

**super**();

System.***out***.println("Student Model 3 param Constructor");

**this**.studentId = studentId;

**this**.studentName = studentName;

**this**.studentMark = studentMark;

}

**public** **int** getStudentId() {

**return** studentId;

}

**public** **void** setStudentId(**int** studentId) {

**this**.studentId = studentId;

}

**public** String getStudentName() {

**return** studentName;

}

**public** **void** setStudentName(String studentName) {

**this**.studentName = studentName;

}

**public** **float** getStudentMark() {

**return** studentMark;

}

**public** **void** setStudentMark(**float** studentMark) {

**this**.studentMark = studentMark;

}

}

STEP-6:

Create the Config.XML file with the appropriate beans that are written in the model POJOs

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

<bean id=*"student"* class=*"com.model.Student"*>

<property name=*"studentId"* value = *'${aaa}'*></property>

<property name=*"studentName"* value = *'${bbb}'*></property>

<property name=*"studentMark"* value = *'${ccc}'*></property>

</bean>

<bean id=*"student1"* class=*"com.model.Student"*>

<property name=*"studentId"* value = *'${aa}'*></property>

<property name=*"studentName"* value = *'${bb}'*></property>

<property name=*"studentMark"* value =*'${cc}'*></property>

</bean>

<bean id=*"trainer"* class=*"com.model.Trainer"* autowire=*"byName"*>

<property name=*"trainerId"* value=*"1587"*></property>

<property name=*"trainerName"* value=*"JKD"*></property>

<property name=*"trainerDept"* value=*"Training"*></property>

<!-- <property name="students">

<list>

<ref bean = "student"/>

<ref bean = "student1"/>

<ref bean = "student1"/>

<ref bean = "student1"/>

</list>

<set>

<ref bean = "student"/>

<ref bean = "student1"/>

</set></property> -->

</bean>

<bean

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

<property name=*"locations"*>

<list>

<value>classpath\*:/myproperties/abc.properties</value>

<value>classpath\*:/myproperties/efg.properties</value>

</list>

</property>

</bean>

</beans>

STEP-6:

Create the main and the call the container or Bean factory

package com.main;

import java.util.Set;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.model.Student;

import com.model.Trainer;

public class StudentDemo1Main {

public static void main(String[] args) {

ApplicationContext applicationContext = new ClassPathXmlApplicationContext("StudentConfig.xml");

Student student = (Student) applicationContext.getBean("student");

System.out.println(student.getStudentId());

System.out.println(student.getStudentName());

System.out.println(student.getStudentMark());

Trainer trainer = (Trainer) applicationContext.getBean("trainer");

System.out.println(trainer.getTrainerDept());

System.out.println(trainer.getStudent().getStudentId());

}

}