Bean

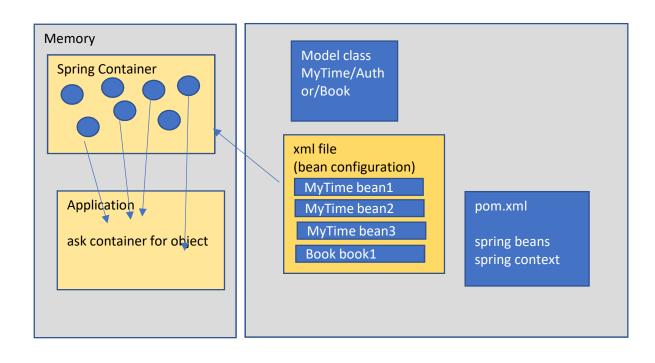
It is an object created in spring container Ready to use object By injecting all dependencies

Container

Area in RAM to load all beans defined in configuration

Beans which are loaded into container can be used by application directly

Beans load to container from configuration before using



Steps to create spring application, define beans by injecting values thru setters

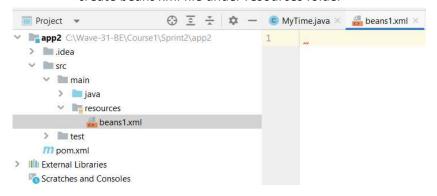
- 1 Create maven project
- 2 add required dependencies

```
25
           <!-- https://mvnrepository.com/artifact/org.springframework/spring-beans -->
26
           <dependency>
27
             <groupId>org.springframework
28
             <artifactId>spring-beans</artifactId>
29
             <version>5.3.23
30
           </dependency>
31
32
           <!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->
33
           <dependency>
34
             <groupId>org.springframework
35
             <artifactId>spring-context</artifactId>
36
             <version>5.3.23
37
           </dependency>
```

3 Define model class (MyTime)

```
MyTime.java
       package org.example.model;
3
       public class MyTime {
 4
           private int hours, minutes, seconds;
5
           public MyTime() {...}
8
           public MyTime(int h, int m, int s) {...}
13
           public int getHours() { return hours; }
16
           public void setHours(int hours) { this.hours = hours; }
19
           public int getMinutes() { return minutes; }
           public void setMinutes(int minutes) { this.minutes = minutes; }
25
           public int getSeconds() { return seconds; }
           public void setSeconds(int seconds) { this.seconds = seconds; }
28
31
32
           @Override
33 01
           public String toString() { return hours + " : " + minutes + " : " + seconds; }
36
```

4 Define beans in xml configuration file create resources folder under main create beans xml file under resources folder



```
💿 MyTime.java 🗶 🍶 beans1.xml 🗴 🏻 💣 MyTimeBeanApp1.java 🗴   🎁 pom.xml (app2) 🔀
        <beans xmlns="http://www.springframework.org/schema/beans"</pre>
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:context="http://www.springframework.org/schema/context"
 3
               xsi:schemaLocation="http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd
 5
           http://www.springframework.org/schema/context
 6
           http://www.springframework.org/schema/context/spring-context.xsd">
 7
 8
 9
            <!-- define beans here -->
10
                                                                         executes default
11
            <!-- define mytime bean and inject values thru setters --
                                                                         constructor
12
            <bean class="org.example.model.MyTime" iu= time1">
13
                property name="hours" value="30"/>
                cproperty name="minutes" value="45"/>
14
                cproperty name="seconds" value="56"/>
15
                                                                      Injecting
16
            </bean>
                                                                      dependency values
17
                                                                      thru setters
18
       </beans>
```

```
10
         <!-- define beans here -->
11
         <!-- define mytime bean and inject values thru setters -->
12
         <bean class="org.example.model.MyTime" id="time1">
13
            cproperty name="hours" value="30"/> <!-- calls setHours(30) -->
14
            calls setMinutes(45) -->
15
            16
         </bean>
17
18
         <!-- define mytime bean and inject values thru setters -->
19
         <bean class="org.example.model.MyTime" id="time2">
20
            cproperty name="hours" value="10"/>
21
            property name="minutes" value="25"/>
22
            property name="seconds" value="30"/>
23
       </bean>
```

5 in app, create spring container and load beans from configuration file

```
public class MyTimeBeanApp1 {
           public static void main(String[] args) {
9
               // create spring container and load beans from xml file
10
               ApplicationContext container = new ClassPathXmlApplicationContext( configLocation: "beans1.xml");
11
12
               // get a particular bean from container by bean id, and use the bean
13
               MyTime t1 = (MyTime)container.getBean( s: "time1");
14
               System.out.println(t1);
15
           }
16
       MyTimeBeanApp1 ×
       "C:\Program Files\Java\jdk-14.0.1\bin\ja
       MyTime:Default constructor executed....
       30 : 45 : 56
```

Injecting dependencies thru constructor

Example 2 with ref type dependencies

Author class to be defined first Book class to be defined next

```
class Book { // dependent
private string bookname;
private int price
private Author author;
public Book() { } // not assigned value
public Book( String bn, int price, Author auth) { }
Setters
public String toString() { ... }
```

Book HAS_A Author

```
class Author{ // dependency
private string name,address;
public Author() { } // not assigned value
public Author(String nm, String add) { }
Setters
public String toString() { ... }
```

```
<!--
                                                                      28
                                                                                 <!--
10
             Book b = new Book();
                                                                      29
                                                                                     Author a1 = new Authr("xyz", "delhi");
11
             b.setBookName("Network protocols");
                                                                      30
                                                                                     Book b = new Book();
12
             b.setBookPrice(2314);
                                                                      31
                                                                                     b.setBookName("abcd");
13
             b.setAuthor(new Author("Wen", "US"));
                                                                      32
                                                                                     b.setBookPrice(345);
14
         -->
                                                                      33
                                                                                     b.setAuthor(a1);
15
          <!-- define book bean, inject dependencies thru setters -->
                                                                      34
16
          <bean class="org.example.model.Book" id="book1">
17
              35
                                                                                 <bean class="org.example.model.Author" id="a1">
18
              cproperty name="bookPrice" value="2314"/>
                                                                      36
                                                                                     <constructor-arg value="xyz"/>
19
              cproperty name="author">
                                                                      37
                                                                                     <constructor-arg value="delhi"/>
20
                  <!-- define author bean -->
                                                                      38
                                                                                 </bean>
21
                  <bean class="org.example.model.Author">
                                                                      39
                                                                                 <bean class="org.example.model.Book" id="book2">
22
                      <constructor-arg value="Wen"/>
                                                                      40
                                                                                     property name="bookName" value="abcd"/>
23
                     <constructor-arg value="US"/>
                                                                      41
                                                                                     cproperty name="bookPrice" value="345"/>
                  </bean>
24
                                                                      42
                                                                                     cproperty name="author" ref="a1"/>
25
              </property>
                                                                      43
                                                                                 </bean>
26
          </bean>
```

```
7
                                                                  public class BookMainApp1 {
                                                           8
                                                                      public static void main(String[] args) {
                                                           9
                                                                          ApplicationContext container;
                                                           10
                                                                          container = new ClassPathXmlApplicationContext( configLocation: "beans2.xml");
                                                           11
                                                                          Book b1 = (Book)container.getBean( s: "book1");
                                                           12
                                                                          Book b2 = (Book)container.getBean( s: "book2");
                                                           13
                                                                          System.out.println(b1);
                                                           14
                                                                          System.out.println(b2);
                                                           15
                                                          16
                                                                      7
       public class BookMainApp1 {
8
                                                          17
9 >
           public static void main(String[] args) {
10
               ApplicationContext container;
11
               container = new ClassPathXmlApplicationContext( configLocation: "beans2.xml");
12
               Book b1 = (Book)container.getBean( s: "book1");
13
               Book b2 = (Book)container.getBean( s "book2");
14
               System.out.println(b1);
15
               System.out.println(b2);
16
               Author a = container.getBean( s: "a1", Author.class);
17
               System.out.println(a);
18
           }
19
```