Tuesday, January 4, 2022 10:34 AM

Spring Framework --- Beans are the pojo classes

Maven , Gradle , Ant --- Build Tools !!!

Supports --- compilations and packaging of the project !!

Libraries --- Complexity !!!
  Interdependency
  Versions
  Compatibility of versions

For Compilation ---------
Find proper versions, find all dependencies , download all jars , add the jars in the class path

Packaging ---- WAR file / Jar file

____

Maven Repository (Storage) for JAR files
  Global Repository = mvnrepository.com
  Local repository = On your machine the maven downloads the required jars
    C:\Users\LOGINUNAME\.m2\repository

____

Maven folder structure ---
  USUALLY we use the following folder structure - our packages are directly in src----
  src
    study (package)
      SomeClass.java

   bin
     study (package)
        SomeClass.class

____

  Maven folder structure
  src
   main (folder)
     java (folder)
       study (package)
         SomeClass.java
     resources
         Non java setup files

  Compiled code and the jar/war files are created in
   target (folder)
     Packages
        .class files
     x.jar
     x.war
____

For maven configuration = POM.xml POM project object model
  What does this pom.xml
    Group descriptor = every project in maven has
    Artifact name = file name
    Version = 1.0

    Packaging type (jar /war)
    JDK version

```
<dependencies>  ( We will give jars to be used  )
        <dependency> </dependency>
        <dependency> </dependency>
        <dependency> </dependency>
        <dependency>
           <groupId>mysql</groupId>
           <artifactId>mysql-connector-java</artifactId>
           <version>8.0.27</version>
        </dependency>

        ….
</dependencies>
```

Ex1 --- Create a Hello World Spring Framework application

1. New spring starter project
2. Write a main class
3. Write a beans.xml file
4. Bean class ---- MessageBean
5. Get the bean using the spring context !!!

Context = CONTAINER /Application Context

Spring **Container** manages spring components !!!
Spring components are **BEANS**
We communicate with the container using XML

DTD = Document Type Definition
WHEN we use any xml ---- it has a schema (structure)--- what are the names of the tags
What are the attributes of the tags , what are the sub tags , which attributes are compulsory/optional
How many times a subtag can be used inside the tag !!!!

This schema is defined in earlier versions ---- <! DocType "url \\................\xyz.dtd ""url \\................\abc.dtd " >
Late versions do not use dtd for schema definition - they use **NAME SPACES**
< xml**ns src="url" ……………..xsd" >**
XSD : xml schema definition

Initialize the Application Context of the Spring Container !!!!!

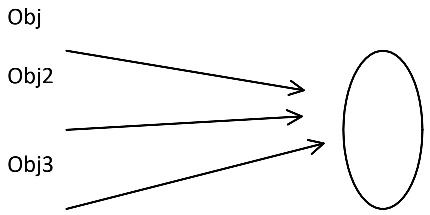
```
Interface ApplicationContext
            |
            |
            |
      Class ClassPathXMLApplicationContext  implements ApplicationContext
            This will get bean info through beans.xml

      Class AnnotationConfigApplicationContext  implements ApplicationContext
            This will get bean info through annotations
```

SPRING container is doing the following jobs ---------------

1. Reads the XML file
2. Creates the bean mentioned in the file , attaches the bean instance with a bean id
3. Calls the property setters of the bean }}}} DI }} Dependency Injection }} setting the properties of the bean !!!

SPRING container uses different techniques to create bean objects ----
spring bean factory by default creates a SINGLETON object !!! Only one instance of the bean will live in the bean context !!

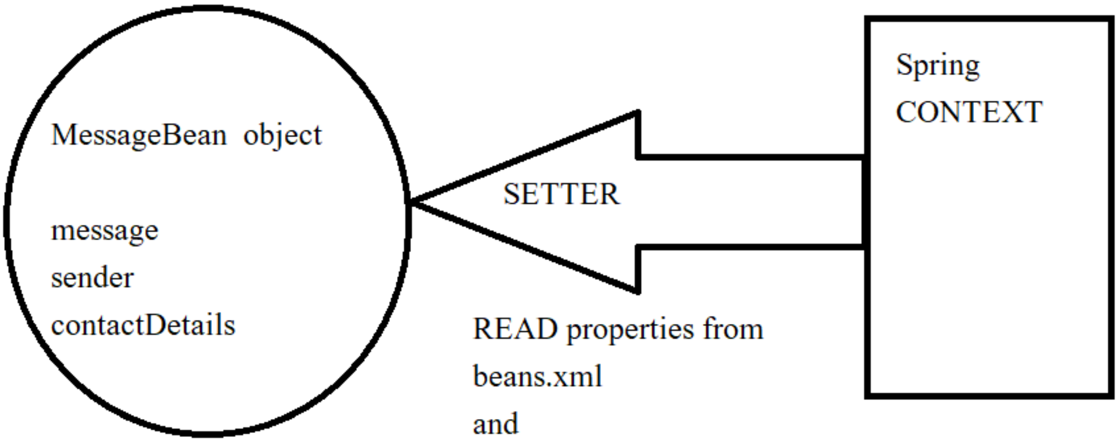


Spring can use a PROTOTYPE bean factory that creates one bean instance for every getBean call !!!!!

If YOU create object of the BEAN using **new** then the purpose of using SPRING is DEFEATED !!!
You must get the beans from SPRING context --- that way they will be **managed** by spring

DI = Dependency Injection

Calling the setter methods of beans and setting its properties known as Dependancy Injection.
this is done by Spring Container



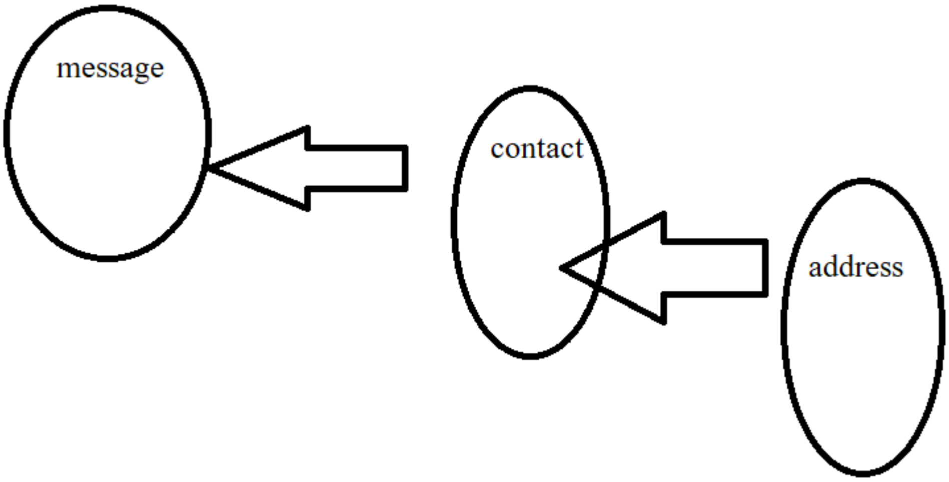
READ properties from
beans.xml
and
Inject those in the message
bean objext

Write a ContactDetails Bean
phonNumber
Email

MessageBean has one more property contact Details!!!!

Add a property Address in contact details !!

AddressBean ----------- area,city,state,country, pin;



Strings and primitives don’t have a bean
Other types have beans declared in the xml !!!

___

Spring Framework Can get configurations of the bean using 3 WAYS -----------

1. Using xml
2. Using Java Config
3. Using Annotations

beans.xml in resources use for mapping

Separate class is written for Configuration

Autowired
Context.scan("package name")
@Configuration

___

HW -- create 3 projects for XMLExample , JavaConfigExample , AnnotationConfigExample
Using spring starter project

Create following beans ---
PersonBean ---- name, AddressBean address , CardsBean cards

AddressBean --- area, city…………….
CardsBean ---- adhaar, pan, passport ,…..

In the main
Access all the three beans separately using getbean and print values
CardsBean is prototype
AddressBean is Singleton
PersonBean is prototype