Springs – Framework of frameworks

Introduction to spring

Spring is an open-source framework for building enterprise java applications.  
Springs aims to simplify the complexity of java application development process by offering a framework that includes technologies such as:

* Aspect Oriented programming
* Dependency Injection
* Plain old object (POJO)

Why Springs?

* Even with all these technologies, spring is a light weight framework that can be used to create scalable, secure and robust enterprise web application.
* The spring framework is also the base that powers all the other spring-based project such as

1.Spring Boot

2.Spring Cloud

3.Spring Graph QL etc...

IOC Containers

Core Container (bean Factory)  
J2EE Containers (Application Context)  
MVC Containers (Web Application Container)

IOC Container

IOC (Inversion of Control) container is responsible to instantiate, configure and assemble the objects.

The IOC Container gets information from the xml file and works accordingly.

The Main task performed by the IOC container are:

* To instantiate the application class.
* To configure the object
* To assemble the dependencies between the objects.

Two types  
Bean factory  
Application Context

POJO (Plain old java Object) and bean class difference  
Note:  
POJO can also have other than private fields ie, POJO can be public, protected, private and default whereas JAVA bean can only have private fields.  
POJO may or may not have a constructor but java beans should have a no argument constructor.

POJO and bean classes are used to define objects for improved readability.

All java Beans are POJOs but all POJOs are not JAVA Beans

Diagram in notes 1.1

Diagram in notes 1.2

TO create spring

Create Maven Project – Simple project - Dependencies to add - Spring context - 5.3.18

What is dependency Injection?

In software engineering, dependency injection is a design pattern in which an object or function receives other objects or functions that it depends on. A form of inversion of control, dependency injection aims to separate the concerns of constructing objects and using them, leading to loosely coupled programs.

Dependencies Injection types.

Variable- only possible thru application context

Constructor- can be done using both interfaces

Property injection - Setters- can be done using both interfaces

Bean if we call twice the object address will be same and it will return same object if we want two different address or different object then set scope as prototype by default it is singleton

Use Scope=” prototype”

Class Loading

Instantiation

Init method

Destroy method – it is not present in Application Context to we have to use child class of application context.

Ie,ClasspathxmlApplicationContext.close();

To use init() and destroy methods we have to mention it in bean in xml

Bean definition for annotation  
Annotation – Component above class from sterotype

Autowired – this will create reference

@component(value=””)

@Scope(“”)

@value to set values to variables, setters and constructor we can use that.

EJB container belongs to application server. It will work for homogeneous type of application, it won’t work for heterogeneous type of applications.

Spring MVC  
-----------------