Spring Container – IOC – inversion Of Control

mybean

Configuration

Bean

Sort2

Business Layer

Look up for mybean

myBean.sort(…)

Client

Main()

dependencies

JPAImpl

JdbcImpl

Sort2 (bean)

sort()

Sort1 (bean)

sort()

injects

Spring

1. IOC Container
2. LightWeighted
3. DI – Dependency Injection
4. Loosely Coupled Components

Applications

1. Standalone apps
2. Web applications ( MVC) – tomcat
3. Rest Services
4. Integerate with hibernate, JPA(ORM) ,Jdbc,XML ….

Configuration file

1. Configure bean
2. Constructor args / properties
3. Add security
4. Logs
5. XML file or (parsers)
6. Java class

Steps

1. Create a maven project
   1. Archetype – quick start
2. Add the spring dependency- pom.xml
3. beans.xml – (configuration)
4. Class called HelloWorld
   1. sayHello()
5. add bean configuration to beans.xml

Beans.xml

App.java

HelloWorld

Create an interface Operator that has operate(int x,int b)

With two implementations AddOperator and MultiplyOperator

PrintInterface

print(result)

OperatorDemo

App2.java

Beans.xml

AddOperator

Operator

operate(..)

MultiplyOperator

MyFileWriter (writes to a file)

ConsoleWriter ( write to console)

bean1

hello message 1

new message 1

<……>-servlet.xml

Controller

(java class)

web.xml

Servlet

DispatcherServlet

(Spring mvc)

Web

UI

Bean2

1. Create web app – archetype – web app – org.apache.maven
2. Link the tomcat
3. Web.xml
4. Spring bean – spring-servlet.xml