Content

Spring Core

Spring Data Integration

Spring WEB

Spring Other Important Modules

Prerequisite of course

Core java

JDBC

Servlet & JSP

Important Web and Database related terms

What is Spring?

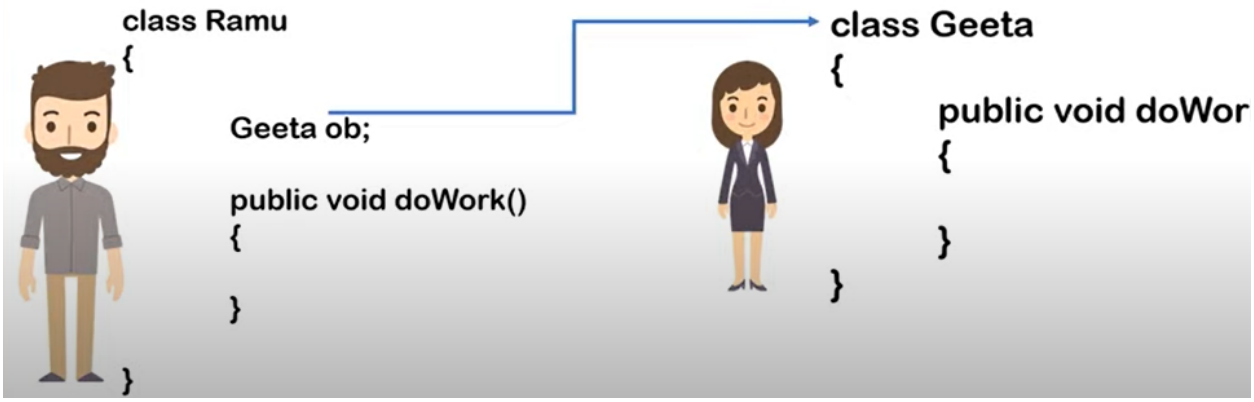
Spring is a Dependency Injection framework to make java application loosely coupled.

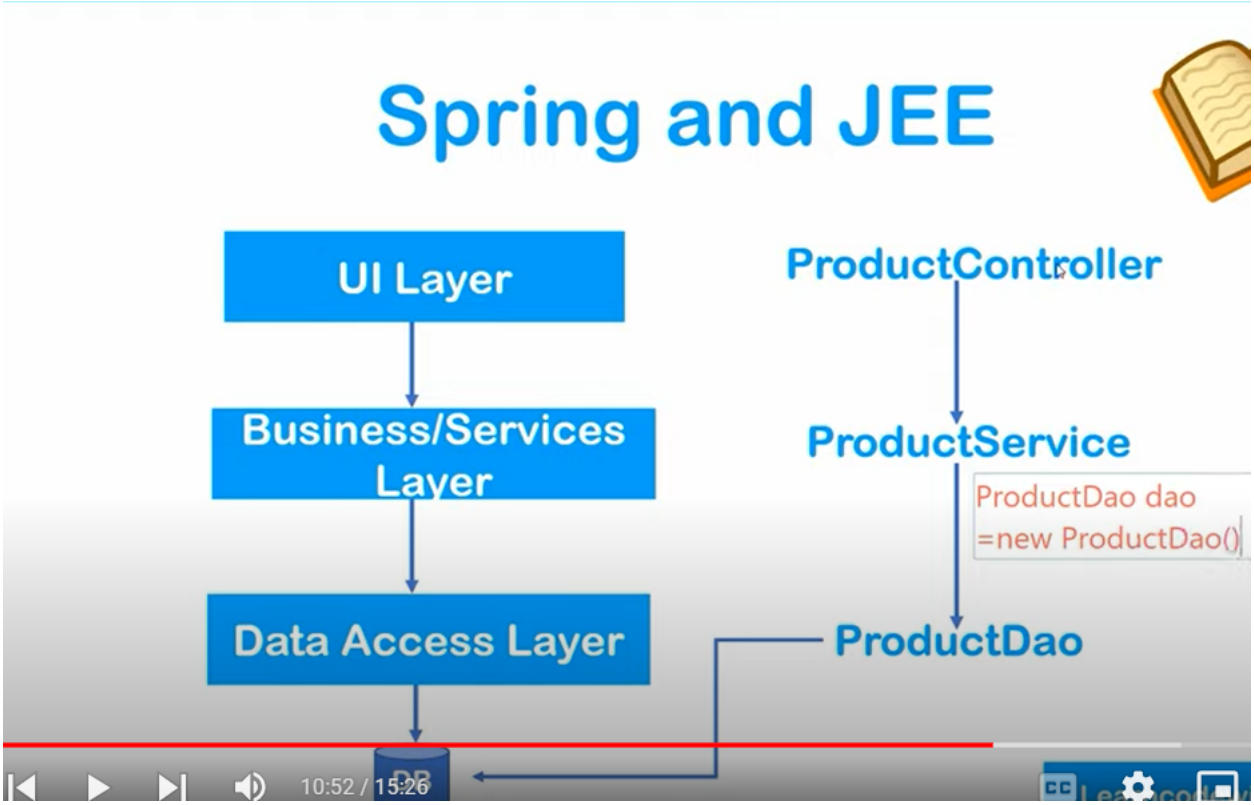
Spring framework makes the easy development of JavaEE application.

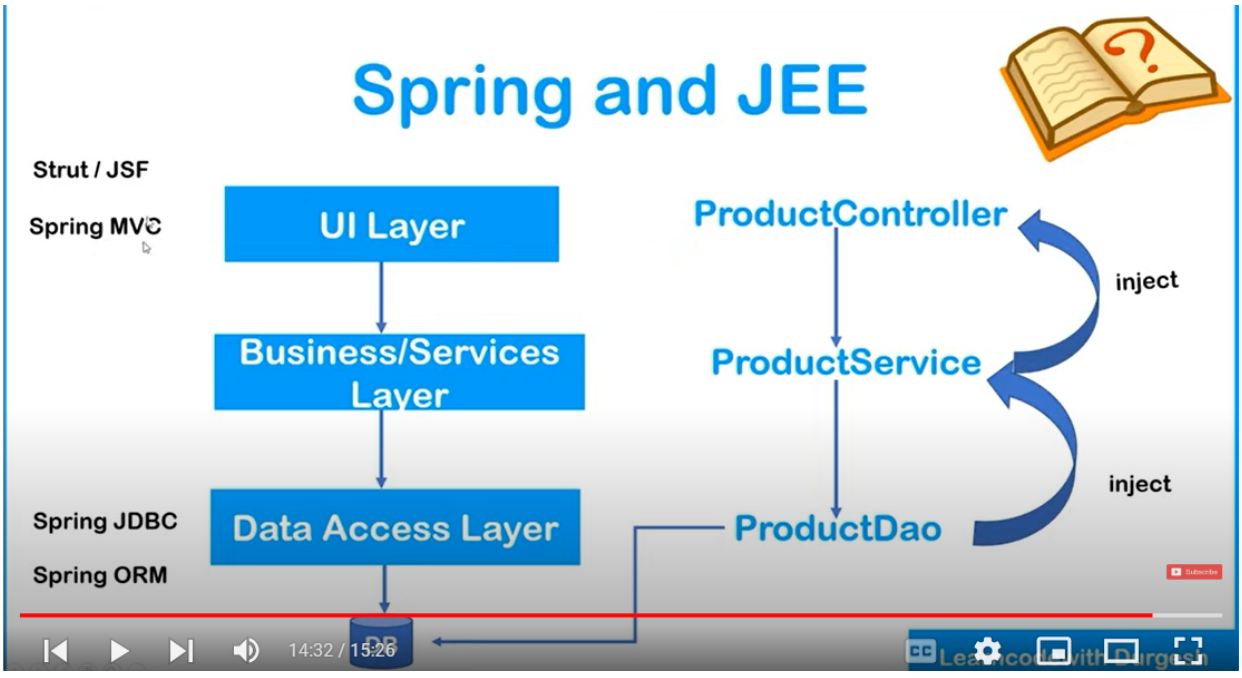
It was developed by Rod Johnson in 2003

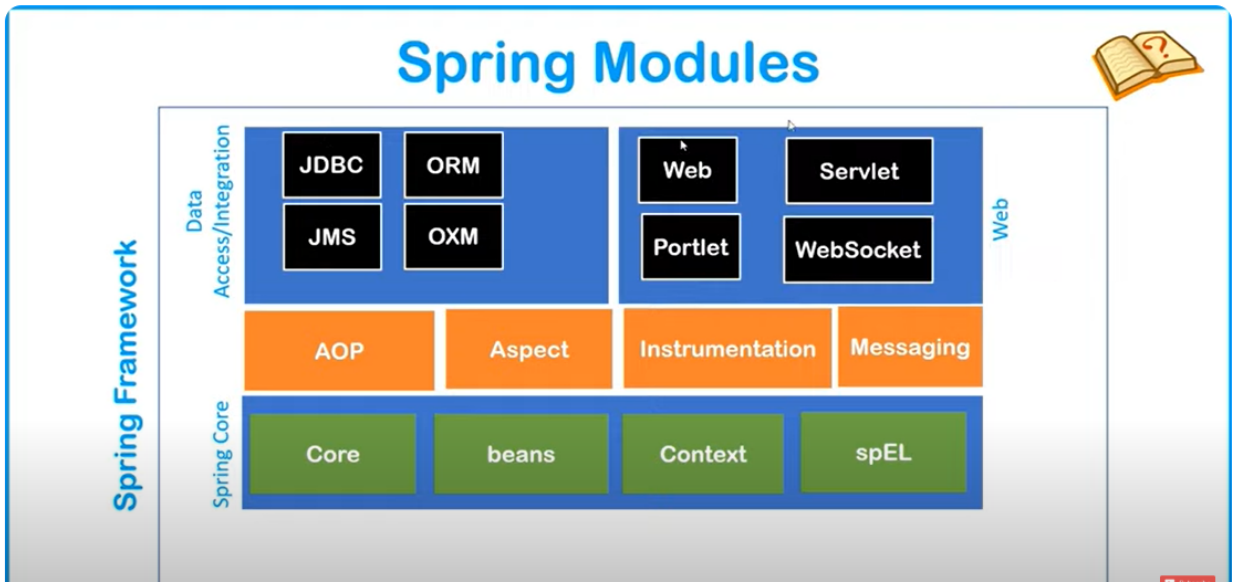
Dependency Injection

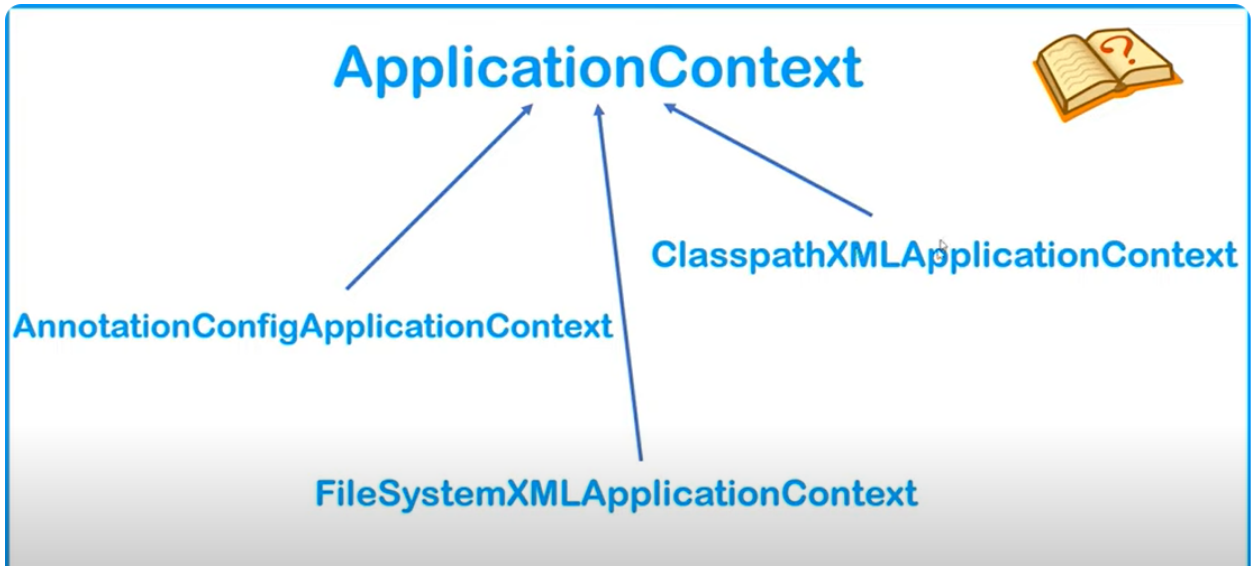
It is design pattern




Spring and JEE
UI Layer
Business/Services Layer
Data Access Layer
DB
ProductController
ProductService
ProductDao dao =new ProductDao()
ProductDao
10:52 / 15:26


Spring and JEE
Strut / JSF
Spring MVC
UI Layer
Business/Services Layer
Spring JDBC
Data Access Layer
Spring ORM
DB
ProductController
ProductService
ProductDao
inject
inject
14:32 / 15:26


Spring Modules
Spring Framework
Data Access/Integration
JDBC
ORM
JMS
OXM
Web
Servlet
Portlet
WebSocket
Web
AOP
Aspect
Instrumentation
Messaging
Spring Core
Core
beans
Context
spEL


ApplicationContext
ClasspathXMLApplicationContext
AnnotationConfigApplicationContext
FileSystemXMLApplicationContext

Dependency Injection

```
class Student
{
    int id;
    String name;
    Address address;
}
```

```
class Address
{
    String street;
    String city;
    String state;
    String country;
}
```

Setter Injection

```
class Student
{
        id,name,address

        setId(id){ }

        setName(name){ }

        setAddress(address){ }

}
```

```
class Address
{
        street,city,state,country
         setStreet(street)
         setCity(city)
         setState(state)
         setCountry(country)
}
```

Constructor Injection

```
class Student
{
        String id,name,address;

        Student(id,name,address)
        {


        }
}
```

IoC Container

```
class Address
{
  String street,city,state,country;
  Address(street,city,state,country)
  {

  }
```

Address

Configuration File

Class ABC ----- beans

XML

Where we declare beans and its dependency

Data Types(Dependencies)

1)Primitive DataTypes

Byte , short , char , int , float , double , long , boolean

2)Collection Type

List, Set , Map and Properties

3)Reference Type

Other class Object

Softwares:
1=>Eclipse/Netbeans/IntelliJ
2=>TomcatServer
3=>Mysql for database
4=>Sqlyog , workbench or phpmyadmin for mysql gui

1-Create maven project
2-Adding dependencies =>spring core , spring context
3-creating beans
4-creating configuration file=>config.xml
5-Setting Injection
6-Main class : which can pull the object and use

```
Primitive Types:
<property name="">
    <value>value</value>
</property>

<property name="" value="" />


<bean p:property:Value />

++++++++++++++++++++++++++++++++++++++++++++++++++
```

Collection Types:

List

```
<bean>
    <property name="">

     <list>
        <value>10</value>
        <value>1023</value>
        <value>12350</value>
        <value>1052</value>
        <null/>
     </list>

    </property>
</bean>
```

Set

```
<bean>
    <property name="">

     <set>
        <value>10</value>
        <value>1023</value>
        <value>12350</value>
        <value>1052</value>
     </set>

    </property>
</bean>
```

Map

```
<bean>
    <property name="">

     <map>
        <entry key="java" value="2month" />
        <entry key="python" value="1month" />

     </map>

    </property>

</bean>
```

1. Emp .java a>private Properties prop; b>Add getter and setter . c>Add prop to the constructor. 2. collectionConfig.xml < property name ="prop"> <props> <prop key="name">OJ</prop> <prop key="Industry">I.T.</prop> </props> </property> 3. test.java System.out.println(emp1.getProp());

```
<bean class="com.springcore.ref.B" name="bref">
    <property name="y" value="90" />
</bean>

<bean class="com.springcore.ref.A" name="aref" p:x="33" p:ob-ref="bref" />

<!--<property name="x" value="12" />

    <property name="ob"> <ref bean="bref" /> </property>

    <property name="ob" ref="bref" />
</bean>
```

Assignment-2 <beans> <bean class="ConsInjection_Exercise.Certi" name="Cert"> <constructor-arg value="Java"/> </bean> <bean class="ConsInjection_Exercise.PersonInfo" name="po"> <constructor-arg value="Raj"/> <constructor-arg value="504"/> <constructor-arg ref="Cert"/> <constructor-arg > <list> <value>Java</value> <value>Python</value> <value>SpringBoot</value> </list> </constructor-arg> </bean> </beans>

```
</bean>

<bean class="com.springcore.ci.Person" name="person">
    <constructor-arg value="Durgesh" />
    <constructor-arg value="12" />
    <constructor-arg ref="cer" />
</bean>

<bean class="com.springcore.ci.Adition" name="add" >
    <constructor-arg value="12" type="int" index="1" />
    <constructor-arg value="34" type="int" index="0" />
</bean>
```

Attribute : index

Life Cycle Methods

Spring provide two important methods to every bean

public void init()

public void destroy()

Configure Technique

- Xml
- Spring Interface
- Annotation

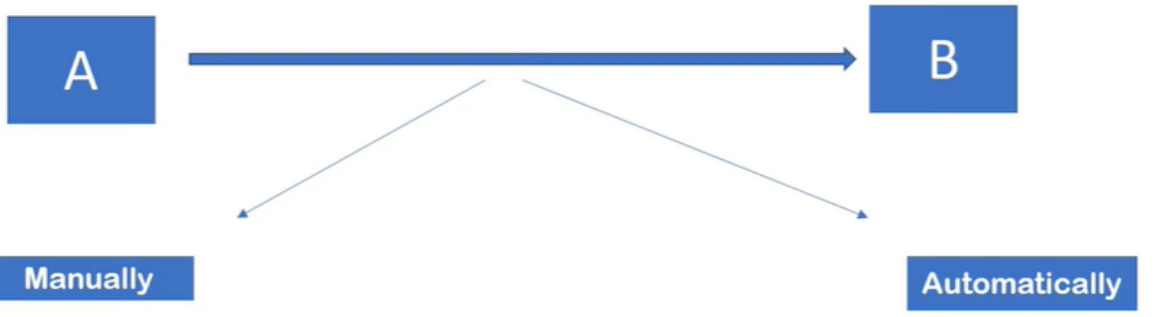
Using Annotations

Implementing Bean LifeCycle
using Annotations
@PostConstruct
@PreDestroy

```
21
22      <bean
23          class="org.springframework.context.annotation.CommonAnnotationBeanPostProcessor" />
24
25  </beans>
```

Autowiring in Spring

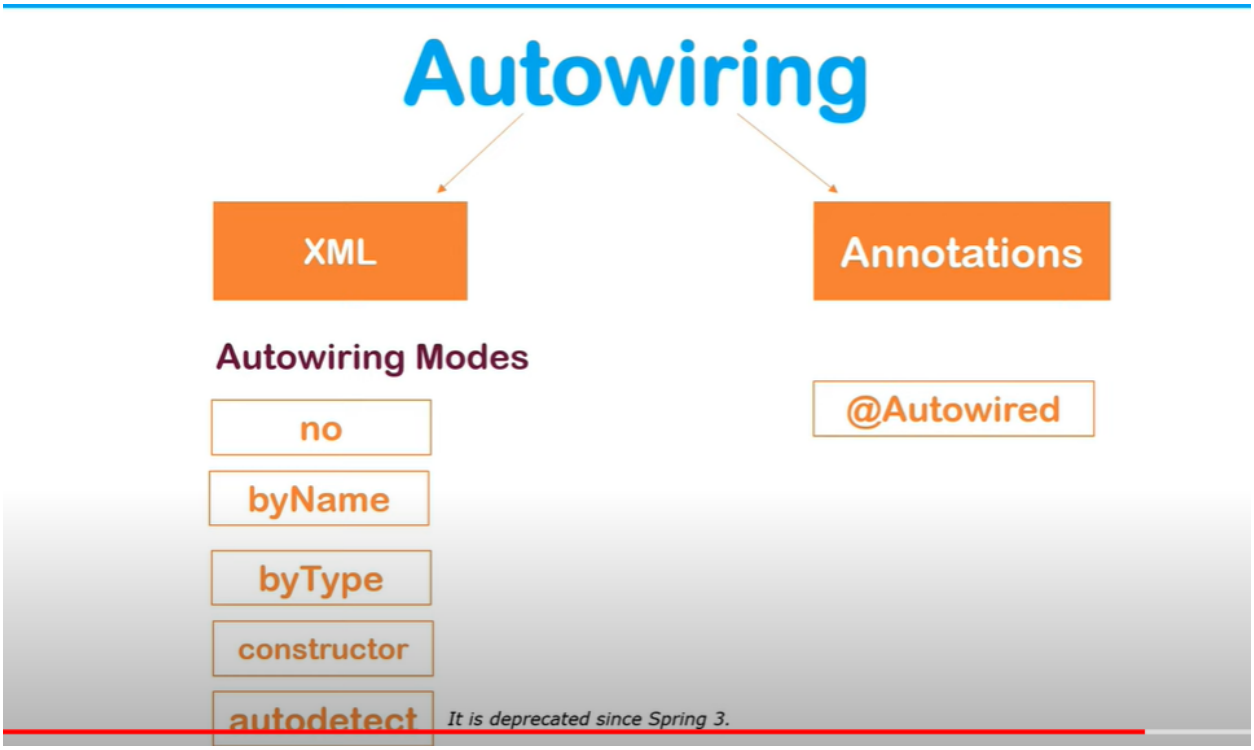
- Feature of spring Framework in which spring container inject the dependencies automatically .
- Autowiring can't be used to inject primitive and string values. It works with reference only.



<ref bean=""/>

Programmer

Spring container



Autowiring Advantages

- Automatic
- less code

Autowiring Disadvantages

- No control of programmer.
- It can't be used for primitive and string values.

```xml
<?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"
    xmlns:p="http://www.springframework.org/schema/p"
    xmlns:util="http://www.springframework.org/schema/util"
    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
    http://www.springframework.org/schema/util
    http://www.springframework.org/schema/util/spring-util.xsd
    ">

    <context:component-scan
        base-package="com.springcore.stereotype" />

    <util:list list-class="java.util.ArrayList" id="temp">
        <value>Lucknow</value>
        <value>Delhi</value>
        <value>Kanpur</value>
    </util:list>


</beans>
```

```java
    @Value("Durgesh Tiwari")
    private String studentName;
    @Value("Lucknow")
    private String city;

    @Value("#{temp}")
```

```java
    public List<String> getAddress() {
        return address;
    }
    public void setAddress(List<String> address) {
        this.address = address;
    }
```

```java
    ApplicationContext con = new ClassPathXmlApplicationContext("com/springcore/stereotype/s
    Student student = con.getBean("ob", Student.class);
    System.out.println(student);
    System.out.println(student.getAddress());
    System.out.println(student.getAddress().getClass().getName());
}
```

Bean Scope

Singleton

prototype

request

session

globalsession

Configure bean scope

```
<bean class=“ ” name=“ ” scope=“ ” />
```

```
@Component
@Scope( “   ”)
Class Student
{


}
```

SpEL Spring Expression Language

Supports Parsing and executing expression with the help of @Value annotations

Expression

@Value (" #{ expression } ")

Classes,Variable,Methods,Constructors and Objects

and symbols

char,numerics,operators,keywords and special symbols which return a value

SpEL

@value("#{11+22}")

@value("#{ 8>6 ? 88 : 55"})

static methods | object methods | variables

How to invoke static method and variable?

T(class).method(param)

How to create objects?

new Object(value)