Welcome
To

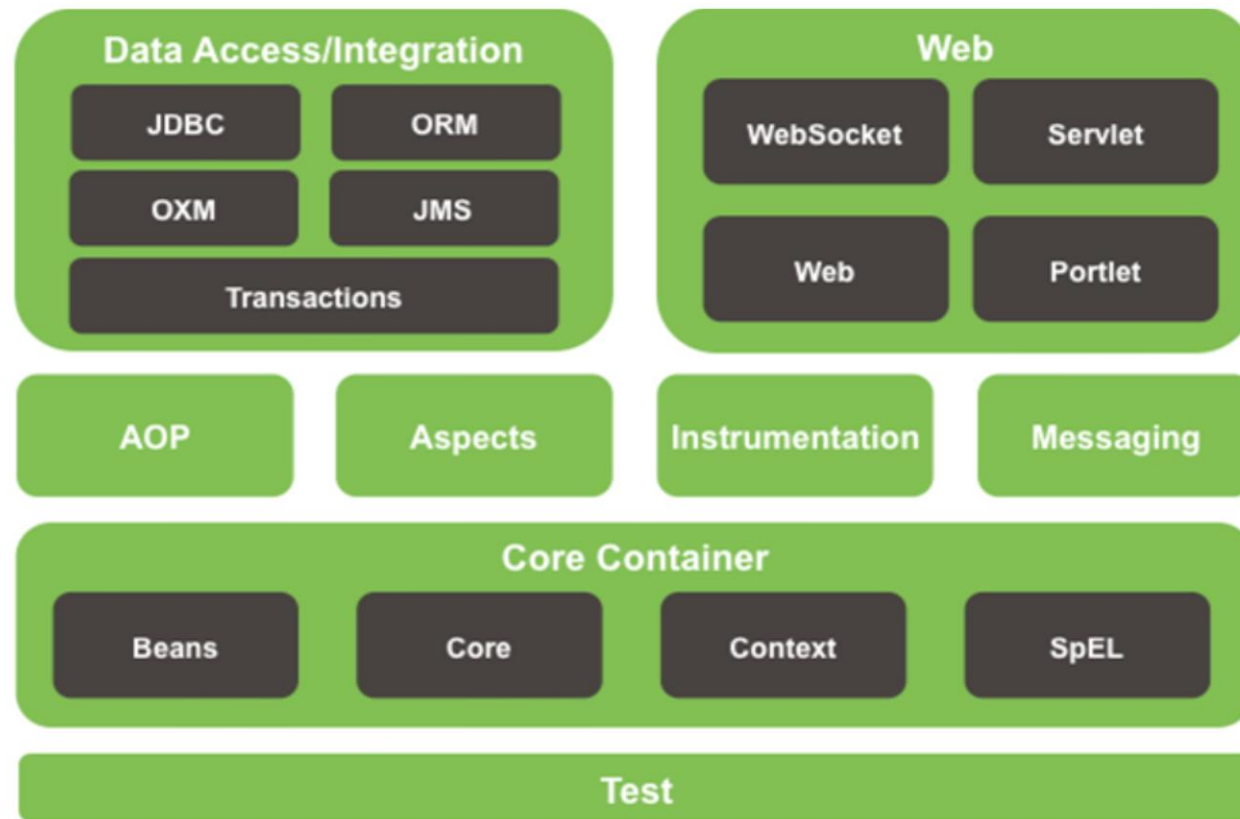# Spring Boot & Microservices

Introduction

❖ **Java Based Open Source Framework**

❖ **Enterprise Application Development**

❖ **Modularized Framework**

❖ **Simplicity of POJOS**

❖ **Dependency Injection & Inversion of Control (IOC)**

❖ **Lightweight Framework**

Spring Architecture

❖ **Spring Core**

❖ **Spring Web**

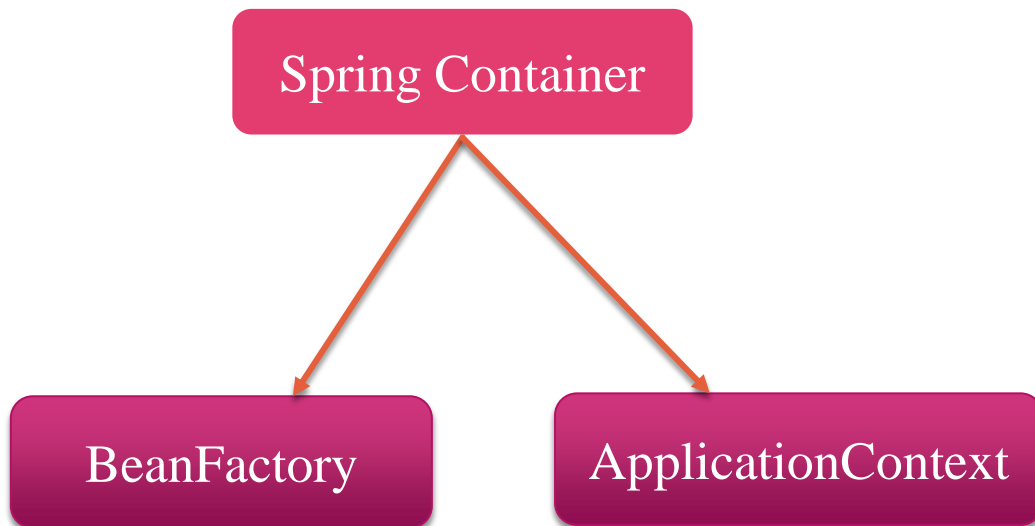❖ **Spring DAO**

❖ **Spring AOP**

Spring Core

- ❖ Spring Core Module is base Module for remaining Modules of Spring Framework.

- ❖ Spring Core is the central part of spring framework to work with spring container. It manages how the beans are created, configured in a spring application.

- ❖ This module makes spring application as "**Light -Weight**" by providing loose coupling between the objects.

- ❖ This module provides services like **Dependency Injection(DI),** Email, I18N, AOP programming etc.

Spring Container

❖ Container is a software application (or) Java Class that can take care of the whole life cycle of given resource.

❖ Spring container is nothing but a Java Class which is provided by spring framework but it is not .exe file (or) .bat file (or) not a setup file.

Spring Container

BeanFactory   ApplicationContext

✓ Bean Factory can only provides "**Dependency Injection**"

✓ Application Context can provide "**Dependency Injection, AOP, I18N,EventHandling**" etc.,

✓ BeanFactory & ApplicationContext are interfaces from Spring library.

Spring Container Classes

❖ BeanFactory(I) and its implementation Classes.
   1) XMLBeanFactory
   2) SimpleJndiBeanFactory
   3) DefaultListableBeanFactory

❖ ApplicationContext(I) and its implementation Classes.
   1) ClassPathXmlApplicationContext
   2) FileSystemXmlApplicationContext
   3) XMLWebApplicationContext

❖ Activating the Spring container in spring application is nothing but creating object an implementation class for either of any Spring containers.

❖ Most frequently used implementation class are "XMLBeanFactory, ClassPathXMlApplicationContext" in spring application development.

❖ We can't activate the Servlet Container, JSP Container by creating objects for certain classes so they are heavy weight containers these containers will activate during server starup time itself.

Spring Configuration

❖ Spring Container creates spring bean object, performs life cycle operations, inject dependencies and finally destroys spring bean object.

❖ Every Spring bean in Spring application has to be configure in **Spring configuration file**.

❖ Spring configuration file is nothing but an xml file created by programmer/Developer. It should be configured every spring bean in spring application.

anyname.xml

❖ We can provide any file name to Spring configuration file i.e., anyname.xml.

XML

Spring Bean Configuration

❖ We can create more than one spring configuration file in spring application.

❖ Spring configuration File is a simple xml file with beans and their dependencies configuration.

Spring Bean Configuration

```
Class Employee{
    ............
    ............
    ............
}
```

```
Class Customer{
    ............
    ............
    ............
}
```

```xml
<beans xlmns ="...........................
                ..........................
                .......................">

    <!-- configuring the Employee class -->
    <bean id="emp" class="com.ait.Employee">
        ..........................................
        ..........................................
    </bean>


    <!-- configuring the Customer class -->
    <bean id="emp" class="com.ait.Customer">
        ..........................................
        ..........................................
    </bean>

<beans>
```
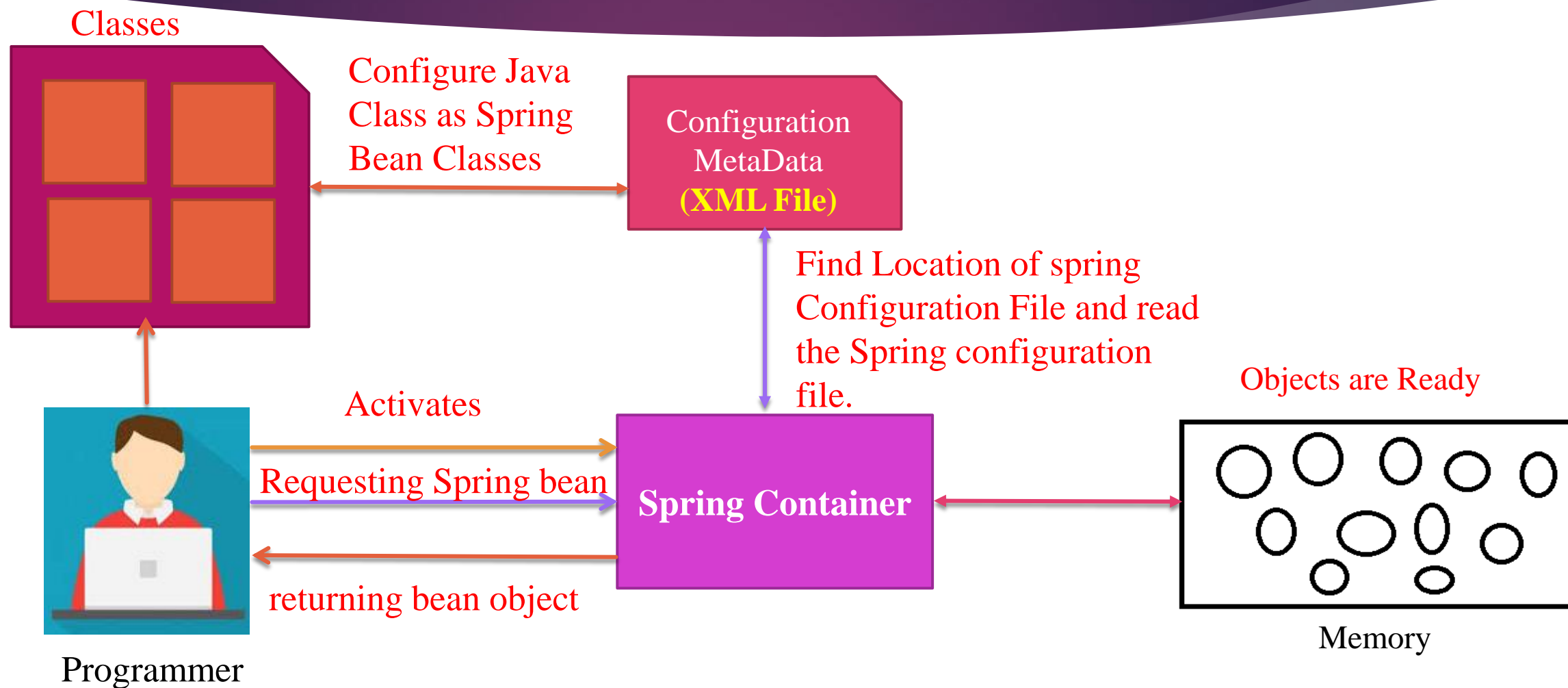
Spring Bean

❖ Spring bean is a key concept of the Spring Framework.

❖ Spring beans are nothing but Simple Java Class which doesn't extends (or) implements third party related classes & interface.

❖ Spring beans are always managed by Spring IOC Container mean that Till Object Creation to Object Destruction will be taking care of everything by Spring Framework.

❖ We can represent the Java Class as Spring Bean below ways
1) XML Configuration
2) By Using Stereo Type Annotation(@Component, @Service, @Repository, @Controller)
3) Java Based Configuration (@Bean)

❖ Every Spring Bean in Spring framework represented as "Singleton" by default.

Basic Flow

Classes

Configure Java Class as Spring Bean Classes

Configuration MetaData **(XML File)**

Find Location of spring Configuration File and read the Spring configuration file.

Objects are Ready

Activates

Requesting Spring bean

Spring Container

returning bean object

Memory

Programmer

Q & A Session

For More Information : https://ashokitech.com/