

Spring 概念

概念：

一、Spring是什么

Spring是一种多层的J2EE应用程序框架，其核心就是**提供一种新的机制管理业务对象及其依赖关系**。

二、为什么要使用Spring

1. 降低组件之间的耦合度,实现软件各层之间的解耦。 Controller -> Service -> DAO
2. 可以使用容器提供的众多服务，如：事务管理服务、消息服务等等。当我们使用容器管理事务时，开发人员就不再需要手工控制事务.也不需处理复杂的事务传播。
3. 容器提供单例模式支持，开发人员不再需要自己编写实现代码。
4. 容器提供了AOP技术，利用它很容易实现如权限拦截、运行期监控等功能。
5. 容器提供的众多辅作类，使用这些类能够加快应用的开发，如：

JdbcTemplate、 HibernateTemplate。

6. Spring对于主流的应用框架提供了集成支持，如：集成mybatis, Hibernate、JPA、Struts等，这样更便于应用的开发。

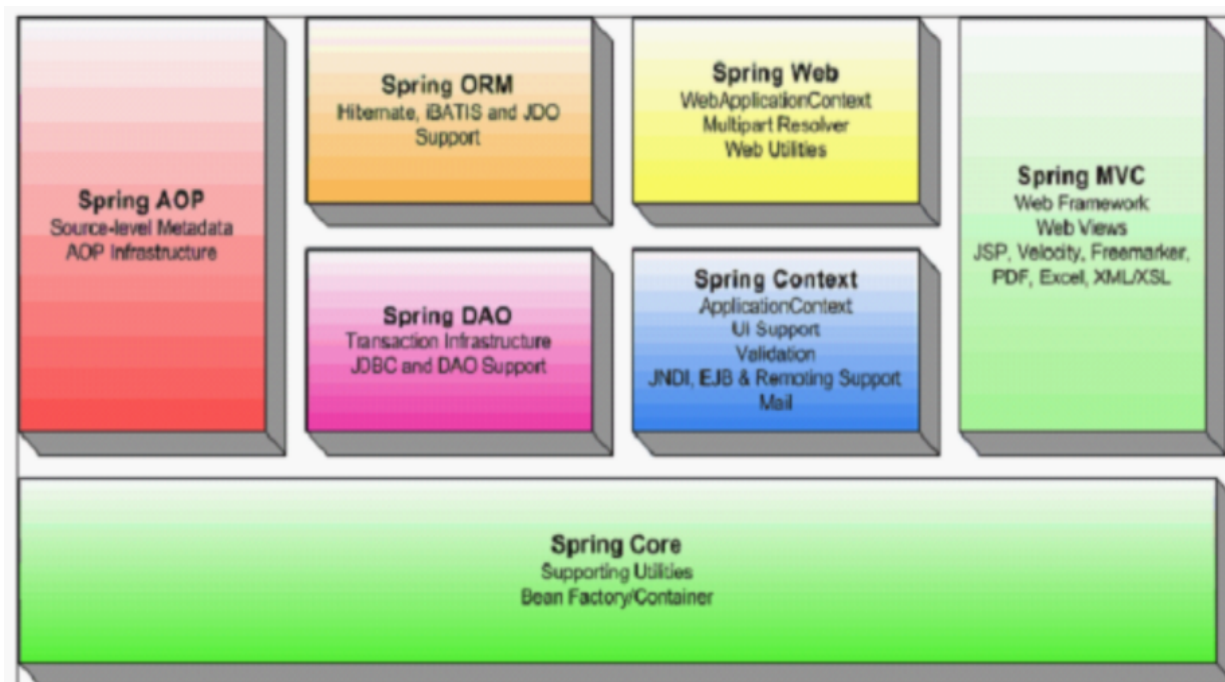
三、Spring的核心及其组成

1. 核心就是提供一种新的机制**管理业务对象及其依赖关系**。IoC (Inversion of Control 控制反转) / DI (Dependency Injection依赖注入)，AOP (面向切面编程)。

1.1 **IoC**: Inversion of Control (控制反转)：就是把要在程序中实例化的对象配置到文件中,在程序中不用new来产生，而是让容器通过配置文件返回一个给你，好处就是当需要更改业务逻辑时，方便替换接口的不同实现类。

1.2 **AOP**: Aspect Oriented Programming (面向切面编程) 用来在系统中提升业务的分离，把日志、安全、事务等东西和核心业务分离开,甚至核心业务都不知道它们的存在。基本实现就是对相关的方法进行拦截，添加所需的处理动作。

2. Spring的七大组成模块



四、Spring的特点

1. 轻量级：轻量级是针对重量级容器（EJB）来说的，Spring的核心包就不到1M大小，而使用Spring的核心包所需的资源也很小，所以可以在小型设备中使用。

2. 非入侵式：Spring目标是一个非入侵式的服务框架。原因是所有框架都是提供大量的功能供用户去使用，从而简化开发时间和成本，但由于大量的使用了框架的API，使应用程序和框架发生了大量的依赖性，无法从框架中独立出来，更加无法使程序组件在其他程序 中使用。

3. 容器：

3.1 Spring提供了容器功能，容器可以管理对象的生命周期、对象与对象间的关系、我们可以通过编写XML来设置对象关系和初始值，这样容器在启动之后，所有的对象都直接可以使用，不用编写任何编码来产生对象。

3.2 Spring有两种不同的容器：

3.2.1 Bean工厂

(org.springframework.beans.factory.BeanFactory) 是最简单的容器（是Spring基本的IOC容器），提供了基础的依赖注入支持

3.2.2 应用上下文

(org.springframework.context.ApplicationContext) 建立在Bean工厂基础之上，提供了系统构架服务

4. IoC / DI:

4.1 Spring最核心的概念就是IoC（控制反转，所谓控制反转就是应用本身不负责依赖对象的创建及维护，依赖对象的创建及维护是由外部容器负责的。这样控制权就由应用转移到了外部容器，控制权的转移就是所谓反转。），而他的另一个名字就是DI（依赖

注入，所谓依赖注入就是指：在运行期，由外部容器动态地将依赖对象注入到组件中。
）。

4.2 使用Spring我们不必在程序中维护对象的依赖关系，只要在XML中设定，Spring容器会自己根据相关的配置去产生他们之间的关系，相当于所有的关系都是在容器运行的时候注入的，而本身他们的关系是没有的。

5. AOP:

5.1 Spring最被人重视的另一个方面就是对AOP的支持，其实AOP只是Spring支持的一个子容器

5.2 AOP在一个服务流程中插入与业务逻辑无关的系统服务逻辑，如：日志、权限验证、异常处理、事务等，而把这些独立出来设计成一个对象，这样的对象称为Aspect。

6. 持久层：Spring提供了对持久层的整合，如对JDBC的使用加以封装简化，提供事务功能，对ORM工具如Hibernate等的整合。

7. Web框架整合：Spring也提供了Web的MVC解决方案，可以很好的善用IOC和AOP的功能，可以轻松的替换掉View层，如使用JPS，也可以使用熟悉的web容器，如struts技术与spring整合。