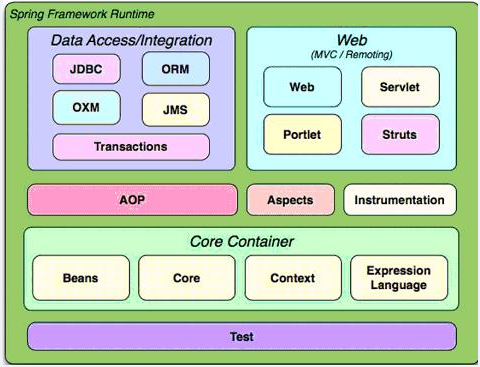
1. Spring是什么

Spring是用于简化程序开发框架，主要用于管理程序中对象间的相互依赖关系，是一个粘合性极强的开发框架。

1. Spring能帮我们做什么
2. spring能根据配置文件创建及组装对象间的依赖关系
3. spring面向切面变成能帮助我们无耦合的实现日志记录、性能统计、安全控制
4. spring管理数据库事务
5. spring提供与第三方数据（hibernate、jpa）访问框架无缝结合，自己也提供一套jdbc访问模板。
6. spring提供与第三方web（Struts,jsf）框架无缝集成，自己也提供医药springmvc web框架
7. spring 能方便与j2ee（javaMail,任务调度）结合
8. 为何需要spring
9. 轻量级容器
10. AOP
11. IOC
12. 简单的
13. 如何学好spring

---------------------------------------------------------------------------------------------------------------------------------

一、Spring基础



* 1. **Core Container**

1. **Core模块：**封装了框架依赖的基础部分，包括资源访问、类型转换及一些常用工具类。
2. **Beans模块：**提供了框架的最底层部分，包括反转控制和依赖注入。其中Bean Factory是容器核心，本质是“工厂模式”的实现，而且无需编程实现“单例设计模式”，单例完全由容器控制，而且提倡面向接口编程，而非面向实现变成；所有应用程序对象及对象关系由框架管理，从而真正把你从程序逻辑中把维护对象之间的依赖关系提取出来，所有这些依赖关系都由BeanFactory来维护。
3. **Context模块:** 以Core和Beans为基础，集成Beans模块功能并添加资源绑定、数据验证、国际化**、**Java EE支持、容器生命周期、事件传播等；核心接口是ApplicationContext。
4. **EL模块:** 提供强大的表达式语言支持，支持访问和修改属性值，方法调用，支持访问及修改数组、容器和索引器，命名变量，支持算数和逻辑运算，支持从Spring 容器获取Bean，它也支持列表投影、选择和一般的列表聚合等。
   1. **AOP、Aspects模块：**
5. **AOP模块：**Spring AOP模块提供了符合 AOP Alliance规范的面向方面的编程（aspect-oriented programming）实现，提供比如日志记录、权限控制、性能统计等通用功能和业务逻辑分离的技术，并且能动态的把这些功能添加到需要的代码中；这样各专其职，降低业务逻辑和通用功能的耦合。
6. **Aspects模块：**提供了对AspectJ的集成，AspectJ提供了比Spring ASP更强大的功能。
   1. **数据访问/集成模块：**该模块包括了JDBC、ORM、OXM、JMS和事务管理
7. 事务模块：该模块用于Spring管理事务，只要是Spring管理对象都能得到Spring管理事务的好处，无需在代码中进行事务控制了，而且支持编程和声明性的事物管理。
8. JDBC模块：提供了一个JBDC的样例模板，使用这些模板能消除传统冗长的JDBC编码还有必须的事务控制，而且能享受到Spring管理事务的好处。
9. ORM模块：提供与流行的“对象-关系”映射框架的无缝集成，包括Hibernate、JPA、Ibatiss等。而且可以使用Spring事务管理，无需额外控制事务。
10. OXM模块：提供了一个对Object/XML映射实现，将java对象映射成XML数据，或者将XML数据映射成java对象，Object/XML映射实现包括JAXB、Castor、XMLBeans和XStream。
11. JMS模块：用于JMS(Java Messaging Service)，提供一套 “消息生产者、消息消费者”模板用于更加简单的使用JMS，JMS用于用于在两个应用程序之间，或分布式系统中发送消息，进行异步通信。
    1. **Web/Remoting模块:** Web/Remoting模块包含了Web、Web-Servlet、Web-Struts、Web-Porlet模块。
12. Web模块：提供了基础的web功能。例如多文件上传、集成IoC容器、远程过程访问（RMI、Hessian、Burlap）以及Web Service支持，并提供一个RestTemplate类来提供方便的Restful services访问。
13. Web-Servlet模块：提供了一个Spring MVC Web框架实现。Spring MVC框架提供了基于注解的请求资源注入、更简单的数据绑定、数据验证等及一套非常易用的JSP标签，完全无缝与Spring其他技术协作。
14. Web-Struts模块：提供了与Struts无缝集成，Struts1.x 和Struts2.x都支持

---------------------------------------------------------------------------------------------------------------------------------

1. IOC基础
2. **IOC是什么**

是一种设计思想。

1. **谁控制谁，控制什么：**IoC 容器控制了对象；控制什么？那就是主要控制了外部资源获取（不只是对象包括比如文件等）
2. **为何是反转，哪些方面反转了：**有反转就有正转，传统应用程序是由我们自己在对象中主动控制去直接获取依赖对象，也就是正转；而反转则是由容器来帮忙创建及注入依赖对象；为何是反转？因为由容器帮我们查找及注入依赖对象，对象只是被动的接受依赖对象，所以是反转；哪些方面反转了？依赖对象的获取被反转了。
3. **IOC能做什么**

有了IoC容器后，把创建和查找依赖对象的控制权交给了容器，由容器进行注入组合对象，所以对象与对象之间是松散耦合，这样也方便测试，利于功能复用，更重要的是使得程序的整个体系结构变得非常灵活

1. **IoC和DI**
2. **谁依赖于谁：**当然是应用程序依赖于IoC容器；
3. **为什么需要依赖：应用程序需要IoC容器来提供对象需要的外部资源；**
4. **谁注入谁：**很明显是IoC容器注入应用程序某个对象，应用程序依赖的对象；
5. **注入了什么：**就是注入某个对象所需要的外部资源（包括对象、资源、常量数据）。
6. IOC容器基本原理
7. **IoC容器的概念**

IoC容器就是具有依赖注入功能的容器，IoC容器负责实例化、定位、配置应用程序中的对象及建立这些对象间的依赖。应用程序无需直接在代码中new相关的对象，应用程序由IoC容器进行组装。在Spring中BeanFactory是IoC容器的实际代表者。

Spring IoC容器如何知道哪些是它管理的对象呢？这就需要配置文件，Spring IoC容器通过读取配置文件中的配置元数据，通过元数据对应用中的各个对象进行实例化及装配。一般使用基于xml配置文件进行配置元数据，而且Spring与配置文件完全解耦的，可以使用其他任何可能的方式进行配置元数据，比如注解、基于java文件的、基于属性文件的配置都可以。