# Spring基础框架

## 名词解析

1. jms：java message service Java消息服务，允许应用程序组件基于JavaEE平台创建、发送、接收和读取消息。
2. jmx：java management extensions，java管理扩展，是一个为应用程序、设备、系统等植入管理功能的框架。
3. oxm：object xml mapping Java 对象（几乎总是一个 plain old Java object，或简写为 POJO）和 XML 文档之间来回转换。
4. jndl：java naming and directory interface java命名和文件接口。
5. ejb：enterprise javabean 定义了一个用于开发基于组件的企业多重应用程序的标准，分别是会话Bean（Session Bean），实体Bean（Entity Bean）和消息驱动Bean（MessageDriven Bean）。
6. JPA：在EJB3.0推出以后，实体Bean被单独从EJB分了出来，形成了新的规范JPA。
7. Aop：面向切面编程
8. Ioc：控制反转
9. DI依赖注入

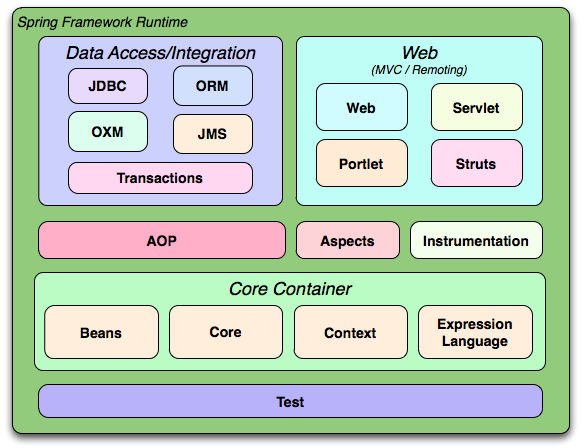
## Spring的相关概念

概念：Spring是一个管理对象关系容器

1. 轻量级，相对于重量级容器EJB，Spring的核心包不到1M，且核心包所需要的资源少，所以可以在小型设备中使用。
2. 非侵入时：
   1. 侵入式：需要继承框架的特定类或接口，使程序对框架产生依赖，不能脱离框架使用。例如Struts的action。
   2. 非侵入时：不需要继承框架的特定类或接口，程序对框架没有依赖.例如Spring
3. 容器：管理对象的关系，对象的生命周期，通过编写xml设置对象的初始值和关系。
4. 依赖反转（IOC）：对象之间的关系不是有对象本身来维护，而是交由Spring来维护。把对象之间的关系写入xml中或用注解标注，Spring在加载xml文件或者扫描注解的时候确定类与类之间的依赖关系。这个依赖关系时在运行时加载到Spring容器中时决定的，而不是在编码时决定的。
5. Aop：就是代理模式。  
   参考：<http://www.cnblogs.com/zhaozihan/p/5953063.html>  
   <https://my.oschina.net/huangyong/blog/161338>  
   Spring事务管理<http://blog.csdn.net/trigl/article/details/50968079>

## 架构图

### 架构图一

* 
* Spring官方架构图
* **核心容器（Core Container）**由beans、core、context和el表达式组成。  
  core和bean模块提供了框架的基本部分，包括IoC和依赖注入特性。BeanFactory是工厂模式的一个成熟的实现。它消除了对编程单例的需求，并允许您将依赖项的配置和规范从实际的程序逻辑中分离。  
  context构建于核心和bean模块提供的坚实基础之上，它是一种以框架方式访问对象的方法，类似于JNDI注册表。context模块从bean模块继承其特性，并添加对国际化的支持(例如，使用资源包)、事件传播、资源加载和上下文透明的创建，例如，servlet容器。context模块还支持Java EE特性，如EJB、JMX和基本远程控制。ApplicationContext接口是上下文模块的焦点。  
  Expression language模块提供了一种强大的表达式语言，用于在运行时查询和操作对象图。它是JSP 2.1规范中指定的统一表达式语言(统一EL)的扩展。该语言支持设置和获取属性值、属性分配、方法调用、访问数组的上下文、集合和索引器、逻辑和算术运算符、命名变量，以及根据Spring的IoC容器的名称检索对象。它还支持列表投影和选择以及常见的列表聚合。
* **数据访问和集成（Data Access/Integeration）**Data Access/Integeration由JDBC，orm，oxm，jms和事务组成。JDBC模块提供了一个JDBC抽象层，它消除了进行乏味的JDBC编码和解析数据库供应商特定的错误代码的需要。Spring可以直接编写jdbc代码进行实现，也可以使用jdbcTemplate和jdbcDaoSupport简化数据库操作。  
  ORM模块为流行的对象-关系映射api提供了集成层，包括JPA、JDO、Hibernate和iBatis。使用ORM包，您可以将所有这些o/r映射框架与Spring提供的所有其他特性结合使用，例如前面提到的简单的声明性事务管理特性。  
  OXM模块提供了一个抽象层，支持JAXB、Castor、XMLBeans、JiBX和XStream等框架的object/xml映射实现，可以整合这些框架。  
  Transaction模块支持对实现特殊接口和所有pojo(plain old Java objects)类进行编程和声明性事务管理。Spring事务管理有两种：一种是声明式事务管理（3种），1、基于TransactionProxyFactoryBean的方式（很少使用），2、基于AspectJ的方式，即<tx:advice>（常使用），3、基于注解的方式（经常使用）；一种是编程式事务管理（1种），使用TransactionTemplate或者PlatformTransactionManager。
* **Web模块** web模块包括web，web servlet，web Struts，web portlet模块  
  Spring的Web模块提供了基本的面向Web的集成特性，比如多部件文件上传功能，以及使用servlet侦听器和面向Web的应用程序上下文的IoC容器的初始化。它还包含Spring远程支持的web相关部分。  
  web servlet模块包含用于web应用程序的Spring模型-视图-控制器(MVC)实现。Spring的MVC框架提供了域模型代码和web表单之间的清晰分离，并集成了Spring框架的所有其他特性。  
  web Struts模块包含用于在Spring应用程序中集成一个经典Struts web层的支持类。注意，在Spring 3.0中，这种支持已经被弃用。考虑将您的应用程序迁移到Struts 2.0及其Spring integration或Spring MVC解决方案。  
  web portlet模块提供了在portlet环境中使用的MVC实现，并反映了web servlet模块的功能。
* **AOP and** **Instrumentation** Spring的AOP模块提供了一种允许您定义的面向方面的编程实现，，例如，方法-拦截器和切入点，可以彻底分离那些应该分离的功能代码。使用源代码级别的元数据功能，您还可以将行为信息合并到您的代码中，以类似于.NET的方式  
  独立的aspects模块提供了与AspectJ的集成

Instrumentation模块提供了类工具支持和类加载器实现，以便在某些应用服务器中使用。  
采用了面向切面编程来实现很多基础但是与业务逻辑无关的功能的解耦,比如：事务管理、日志、权限验证.....

* **Test**测试模块支持使用JUnit或TestNG对Spring组件进行测试。它提供了Spring applicationcontext和这些上下文缓存的一致加载。它还提供了模拟对象，您可以使用这些对象来单独测试您的代码。

Spring框架是一个分层架构，它有7个模块，分为三层，如图所示。底层是Spring core，这个是Spring核心模块，是其他六个模块的基础。Spring上层依赖下层，需要和下层组合使用。Spring每个模块（或者与下层组合成的组件）都可以单独使用，也可以和其他模块或组件联合使用。

### 架构图二



网上架构图

* **核心容器**：Spring core是一个装载bean的容器，是Spring框架的基础，提供Spring框架的基本功能。她的主要组件是beanFactory，他是工厂模式的实现。Beanfactory使用控制反转（IOC）模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。
* **Spring上下文** ：Spring上下文是一个配置文件，向Spring提供配置信息。Spring上下文包括企业服务，例如：JNDI，EJB，国际化，电子邮件，校验和调度功能。
* **Spring AOP**：通过配置管理特性，Spring aop直接将面向切面的编程集中到Spring框架中，所以，很容易地使Spring管理的任何对象支持aop。Spring aop为基于Spring应用程序中的对象提供了事务管理服务。通过aop，不用通过EJB，就可以将声明式事务管理集成到易用程序中。
* **Spring DAO**：Spring DAO抽象层提供了有意义的异常层次结构。可以用该结构管理异常处理和不同数据库提供商抛出的异常信息。异常层次结构简化了错误处理，并且极大地降低了需要编写的异常代码数量（如打开和关闭连接）。Spring dao面向jdbc的异常遵循通用的DAO异常层次结构。对JDBC进行封装，通过jdbcTemplate简化
* **Spring ORM**：spring框架引入了若干个orm框架，从而提供了orm的对象管理工具，如JDO，hibernate和ibatis。所有这些都遵循Spring的通用事务和DAO异常层次结构。
* **Spring WEB**：web模块上下文建立在应用程序上下文之上，为给予web的应用程序提供了上下文。所以，Spring框架支持与jakarta Struts的集成。Web还简化了处理多部分请求以及将参赛绑定到域对象的工作。
* **Spring MVC**：MVC框架是一个全功能的构建web应用程序的MVC实现。通过策略接口，mvc框架变为高度可配置的，MVC容纳了大量视图技术，其中包括JSP，Velocity，Tiles和POI。

## 参考

各个框架架构图<http://www.cnblogs.com/littleCode/p/3926791.html>

Spring框架学习<https://www.ibm.com/developerworks/cn/java/wa-spring1/>

Spring官方文档