Spring概况

<http://spring.io/>

<http://projects.spring.io/spring-framework/>

Spring作为一个轻量级的J2EE的开源开发框架，为企业级的复杂应用的开发提供了一套简单轻量级的解决方案。但也不止应用于企业应用了，还有互联网应用，服务端应用等等。Spring为复杂的JavaEE开发带来了春天。

框架的特点

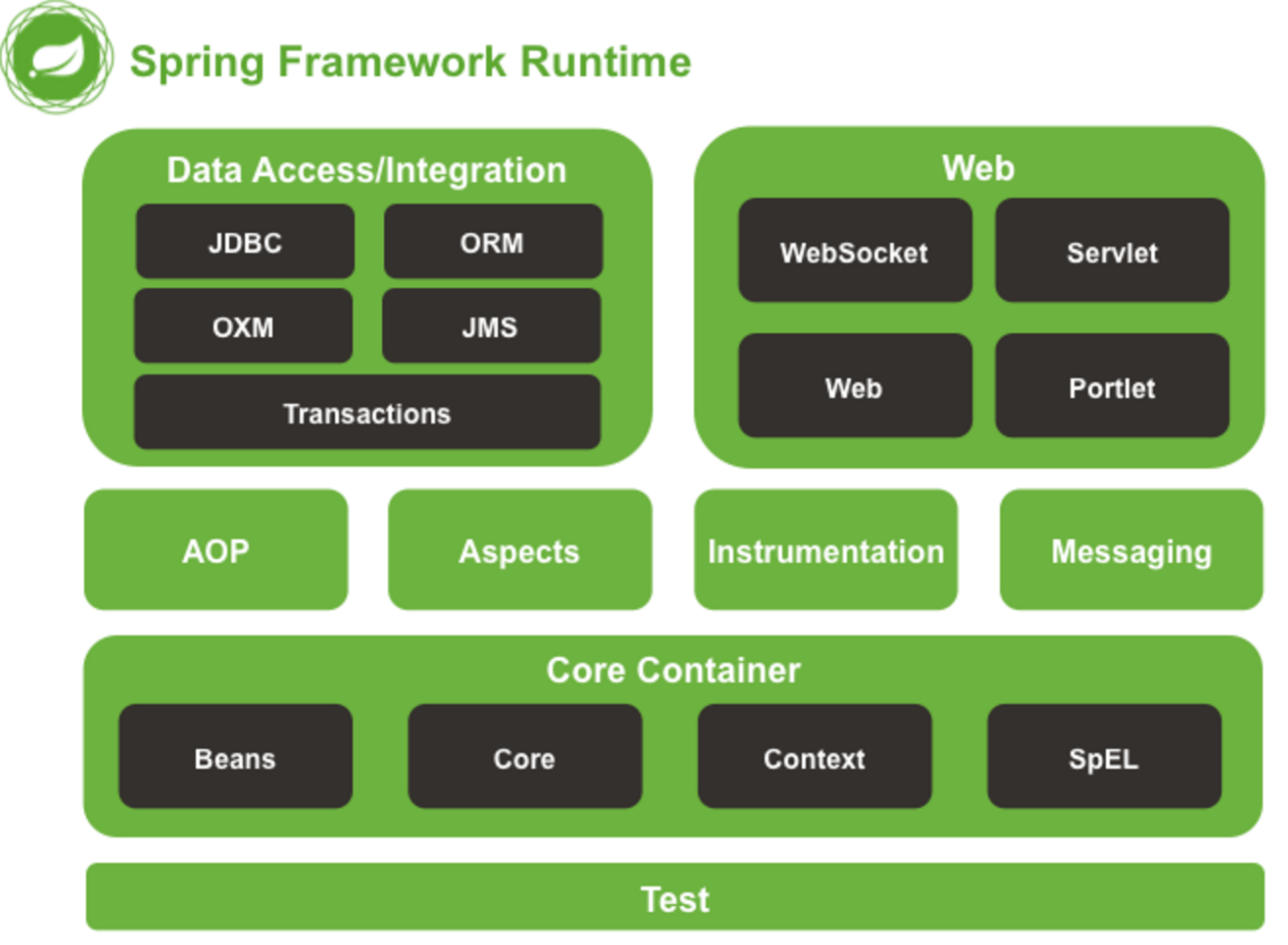
1. 半成品
2. 封装了特定的处理流程和控制逻辑
3. 成熟的、不断升级改进的软件

框架与类库的区别

1. 框架一般是封装了逻辑、高内聚的，类库这是松散的工具组合
2. 框架专注于某一领域，类库则是更通用的

Spring最初是一个轻量级的控制反转(IoC)和面向切面(AOP)的容器框架。

1. 从大小与开销方面而言Spring都是轻量的。
2. 通过控制反转IoC的技术达到松耦合的目的。
3. 提供了面向切面AOP编程的支持，允许通过分离应用的业务逻辑与系统级服务进行内聚性的开发。比如，商品管理中商品的入库、出库就是业务逻辑；系统服务指比如入库和出库都要进行统一的日志记录、事务管理与具体业务无关，无论什么业务都会涉及到。
4. 包含并管理应用对象的配置和生命周期，这个意义上是一种容器。
5. 将简单的组件配置、组合成为复杂的应用，这个意义上是框架。对主流应用框架(Hibernate, Struts等)提供了良好的支持。



Spring框架的主要优势之一就是其分层架构，允许开发者选择哪一个组件。同时，也为J2EE开发提供了一个集成的框架。

Spring框架使用基本的JavaBeans来代替EJB, 并且提供了更多企业级应用的功能。Spring致力于提供统一的方法来管理业务对象。Spring是全面的和模块化的，分层结构使得可以选择它孤立的任何一个部分。它的设计从底部帮助你编写易于测试的代码。

其他优点：

* 低侵入式设计，代码污染极低
* Write Once, Run Anywhere
* Dependency Injection有效的降低了耦合度，提高了组件之间的解耦
* AOP提供了通用任务的集中管理，从而更好的复用。任务比如：安全，事务，日志等
* ORM和DAO机制提供了与第三方持久层框架的良好整合，简化了对数据库的访问
* 高度开放性，并不强制。该特点贯穿于表现层，业务层，持久层。
* 可以有效的组织中间层对象。
* 使用统一的配置文件，消除使用各种不同格式的属性定制文件的需要，在整个工程应用中可以通过一致的方法来配置。
* 促进良好的编程习惯，减少编程代价。
* 易于简单的单元测试。提供了对JUnit的支持，通过注解来测试。
* 使EJB成为一种备选。
* 为数据存取提供了一致的框架。

主要特点：

* 方便解耦，简化开发。 通过Spring提供的IoC容器，可以将对象之间的依赖关系交由Spring来控制。避免硬编码，造成的程序过渡的耦合。用户可以不再为单实例模式类，属性文件解析等很底层的需求来编写代码。可以更加专注上层的应用。
* AOP编程的支持，用户可以实现面向切面的编程，许多不能用传统OOP实现的功能都可以用AOP编程来轻松面对。
* 声明式事务的支持
* 方便程序测试
* 方便集成各种优秀框架，比如Struts, Hibernate等。
* 降低JavaEE API的使用难度。比如：JDBC, Java Mail, 远程调用等。
* Spring的源码是经典学习范例

Spring之IoC (控制反转)

Spring框架中的核心模块之一，也是系统中其他组建模块和应用开发的基础。

Spring之AOP (面向切面)

Spring框架中的核心模块之一，也是系统中其他组建模块和应用开发的基础。