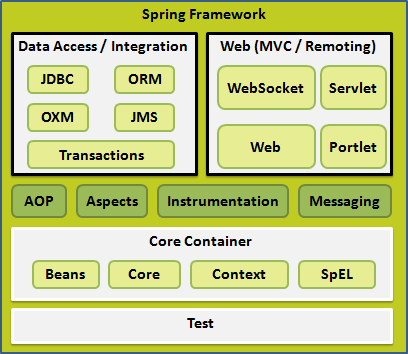
# Spring 体系结构

Spring框架提供了约20个模块，可根据应用实现要求进行选择。

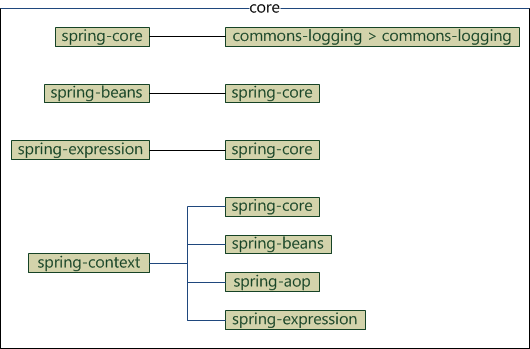


# 核心容器

核心容器由 **spring-core，spring-beans，spring-context，spring-context-support和spring-expression**（SpEL，Spring 表达式语言，Spring Expression Language）等模块组成，它们的细节如下：

* **spring-core**模块提供了框架的基本组成部分，包括 IoC 和依赖注入功能。
* **spring-beans** 模块提供 BeanFactory，工厂模式的微妙实现，它移除了编码式单例的需要，并且可以把配置和依赖从实际编码逻辑中解耦。
* **context**模块建立在由 **core**和 **beans** 模块的基础上建立起来的，它以一种类似于 JNDI 注册的方式访问对象。Context 模块继承自 Bean 模块，并且添加了国际化（比如，使用资源束）、事件传播、资源加载和透明地创建上下文（比如，通过 Servlet 容器）等功能。Context 模块也支持 Java EE 的功能，比如 EJB、JMX 和远程调用等。**ApplicationContext**接口是 Context 模块的焦点。
* **spring-context-support**提供了对第三方集成到 Spring 上下文的支持，比如缓存（EhCache, Guava, JCache）、邮件（JavaMail）、调度（CommonJ, Quartz）、模板引擎（FreeMarker, JasperReports, Velocity）等。
* **spring-expression**模块提供了强大的表达式语言，用于在运行时查询和操作对象图。它是 JSP2.1 规范中定义的统一表达式语言的扩展，支持 set 和 get 属性值、属性赋值、方法调用、访问数组集合及索引的内容、逻辑算术运算、命名变量、通过名字从 Spring IoC 容器检索对象，还支持列表的投影、选择以及聚合等。

**Spring以上容器完整依赖关系如图**



## 数据访问/集成

数据访问/集成层包括 JDBC，ORM，OXM，JMS 和事务处理模块，它们的细节如下：（注：JDBC=Java Data Base Connectivity，ORM=Object Relational Mapping，OXM=Object XML Mapping，JMS=Java Message Service）

* **JDBC** 模块提供了 JDBC 抽象层，它消除了冗长的 JDBC 编码和对数据库供应商特定错误代码的解析。
* **ORM** 模块提供了对流行的对象关系映射 API 的集成，包括 JPA、JDO 和 Hibernate 等。通过此模块可以让这些 ORM 框架和 spring的其它功能整合，比如前面提及的事务管理。
* **OXM** 模块提供了对 OXM 实现的支持，比如 JAXB、Castor、XML Beans、JiBX、XStream 等。
* **JMS** 模块包含生产（produce）和消费（consume）消息的功能。从 Spring 4.1 开始，集成了 spring-messaging 模块。
* **事务**模块为实现特殊接口类及所有的 POJO 支持编程式和声明式事务管理。（注：编程式事务需要自己写 beginTransaction()、commit()、rollback() 等事务管理方法，声明式事务是通过注解或配置由 spring 自动处理，编程式事务力度更细）

## Web

Web 层由 Web，Web-MVC，Web-Socket 和 Web-Portlet 组成，它们的细节如下：

* **Web** 模块提供面向 web 的基本功能和面向 web 的应用上下文，比如多部分（multipart）文件上传功能、使用 Servlet 监听器初始化 IoC 容器等。它还包括 HTTP 客户端以及 Spring 远程调用中与 web 相关的部分。
* **Web-MVC** 模块为 web 应用提供了模型视图控制（MVC）和 REST Web服务的实现。Spring 的 MVC 框架可以使领域模型代码和 web 表单完全地分离，且可以与 Spring 框架的其它所有功能进行集成。
* **Web-Socket** 模块为 WebSocket-based 提供了支持，而且在 web 应用程序中提供了客户端和服务器端之间通信的两种方式。
* **Web-Portlet** 模块提供了用于 Portlet 环境的 MVC 实现，并反映了 spring-webmvc 模块的功能。（暂未了解）

# 其他

还有其他一些重要的模块，像 [AOP](https://www.w3cschool.cn/wkspring/izae1h9w.html)，Aspects，Instrumentation，Web 和测试模块，它们的细节如下：

* **AOP** 模块提供了面向方面（切面）的编程实现，允许你定义方法拦截器和切入点对代码进行干净地解耦，从而使实现功能的代码彻底的解耦出来。使用源码级的元数据，可以用类似于.Net属性的方式合并行为信息到代码中。
* **Aspects** 模块提供了与 **AspectJ** 的集成，这是一个功能强大且成熟的面向切面编程（AOP）框架。
* **Instrumentation** 模块在一定的应用服务器中提供了类 instrumentation 的支持和类加载器的实现。（暂未了解）
* **Messaging** 模块为 STOMP 提供了支持作为在应用程序中 WebSocket 子协议的使用。它也支持一个注解编程模型，它是为了选路和处理来自 WebSocket 客户端的 STOMP 信息。（暂未了解）
* **测试**模块支持对具有 JUnit 或 TestNG 框架的 Spring 组件的测试。