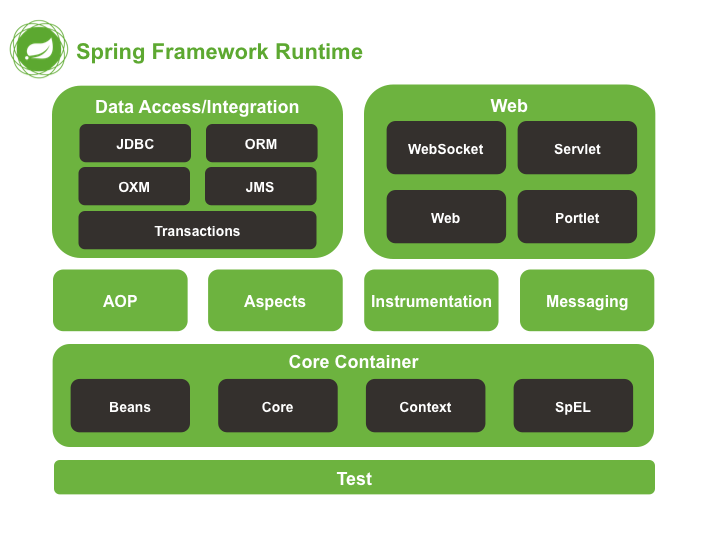
# spring5的体系架构

Spring 总共大约有 20 个模块， 由 1300 多个不同的文件构成。 而这些组件被分别整合在核心容器（Core Container） 、 AOP（Aspect Oriented Programming） 和设备支持（Instrmentation） 、数据访问及集成（Data Access/Integeration） 、 Web、 报文发送（Messaging） 、 Test， 6 个模块集合中。 以下是 Spring 5 的模块结构图：



spring5基本模块.png

组成 Spring 框架的每个模块集合或者模块都可以单独存在， 也可以一个或多个模块联合实现。 每个模  
块的组成和功能如下：

**1.核心容器：**

由 spring-beans、spring-core、spring-context 和 spring-expression（Spring Expression Language, SpEL） 4 个模块组成。

spring-beans 和 spring-core 模块是 Spring 框架的核心模块， 包含了控制反转（Inversion of Control, IOC） 和依赖注入（Dependency Injection, DI） 。 BeanFactory 接口是 Spring 框架中的核心接口， 它是工厂模式的具体实现。 BeanFactory 使用控制反转对应用程序的配置和依赖性规范与实际的应用程序代码进行了分离。 但 BeanFactory 容器实例化后并不会自动实例化 Bean， 只有当 Bean被使用时 BeanFactory 容器才会对该 Bean 进行实例化与依赖关系的装配。

·spring-context 模块构架于核心模块之上， 他扩展了 BeanFactory， 为它添加了 Bean 生命周期控制、 框架事件体系以及资源加载透明化等功能·。 此外该模块还提供了许多企业级支持， 如邮件访问、远程访问、 任务调度等， ApplicationContext 是该模块的核心接口，它是 BeanFactory 的超类， 与BeanFactory 不同， ApplicationContext 容器实例化后会自动对所有的单实例 Bean 进行实例化与依赖关系的装配， 使之处于待用状态。

spring-expression 模块是统一表达式语言（EL） 的扩展模块， 可以查询、 管理运行中的对象，同时也方便的可以调用对象方法、 操作数组、 集合等。 它的语法类似于传统 EL， 但提供了额外的功能，最出色的要数函数调用和简单字符串的模板函数。 这种语言的特性是基于 Spring 产品的需求而设计，他可以非常方便地同 Spring IOC 进行交互。

**2.AOP 和设备支持**

由spring-aop、 spring-aspects 和 spring-instrument3 个模块组成。

spring-aop 是 Spring 的另一个核心模块， 是 AOP 主要的实现模块。 作为继 OOP 后， 对程序员影响最大的编程思想之一， AOP 极大地开拓了人们对于编程的思路。 在 Spring 中， 他是以 JVM 的动态代理技术为基础， 然后设计出了一系列的 AOP 横切实现， 比如前置通知、 返回通知、 异常通知等， 同时，Pointcut 接口来匹配切入点， 可以使用现有的切入点来设计横切面， 也可以扩展相关方法根据需求进  
行切入。

spring-aspects 模块集成自 AspectJ 框架， 主要是为 Spring AOP 提供多种 AOP 实现方法。

spring-instrument 模块是基于 JAVA SE 中的"java.lang.instrument"进行设计的， 应该算是AOP 的一个支援模块， 主要作用是在 JVM 启用时， 生成一个代理类， 程序员通过代理类在运行时修改类的字节， 从而改变一个类的功能， 实现 AOP 的功能。 在分类里， 我把他分在了 AOP 模块下， 在 Spring 官方文档里对这个地方也有点含糊不清， 这里是纯个人观点。

**3.数据访问及集成**

由 spring-jdbc、 spring-tx、 spring-orm、 spring-jms 和 spring-oxm 5 个模块组成。

spring-jdbc 模块是 Spring 提供的 JDBC 抽象框架的主要实现模块， 用于简化 Spring JDBC。主要是提供 JDBC 模板方式、 关系数据库对象化方式、 SimpleJdbc 方式、 事务管理来简化 JDBC 编程， 主要实现类是 JdbcTemplate、 SimpleJdbcTemplate 以及 NamedParameterJdbcTemplate。

spring-tx 模块是 Spring JDBC 事务控制实现模块。 使用 Spring 框架， 它对事务做了很好的封装，通过它的 AOP 配置， 可以灵活的配置在任何一层； 但是在很多的需求和应用， 直接使用 JDBC 事务控制还是有其优势的。 其实， 事务是以业务逻辑为基础的； 一个完整的业务应该对应业务层里的一个方法；如果业务操作失败， 则整个事务回滚； 所以， 事务控制是绝对应该放在业务层的； 但是， 持久层的设计则应该遵循一个很重要的原则： 保证操作的原子性， 即持久层里的每个方法都应该是不可以分割的。 所以， 在使用 Spring JDBC 事务控制时， 应该注意其特殊性。

spring-orm 模块是 ORM 框架支持模块， 主要集成 Hibernate, Java Persistence API (JPA) 和Java Data Objects (JDO) 用于资源管理、 数据访问对象(DAO)的实现和事务策略。

spring-jms 模块（Java Messaging Service） 能够发送和接受信息， 自 Spring Framework 4.1以后， 他还提供了对 spring-messaging 模块的支撑。

spring-oxm 模块主要提供一个抽象层以支撑 OXM（OXM 是 Object-to-XML-Mapping 的缩写， 它是一个 O/M-mapper， 将 java 对象映射成 XML 数据， 或者将 XML 数据映射成 java 对象） ， 例如： JAXB,Castor, XMLBeans, JiBX 和 XStream 等。

**4.Web**

由 spring-web、 spring-webmvc、 spring-websocket 和 spring-webflux4 个模块组成。

spring-web 模块为 Spring 提供了最基础 Web 支持， 主要建立于核心容器之上， 通过 Servlet 或者 Listeners 来初始化 IOC 容器， 也包含一些与 Web 相关的支持。

spring-webmvc 模 块 众 所 周 知 是 一 个 的 Web-Servlet 模 块 ， 实 现 了 Spring MVC（model-view-Controller） 的 Web 应用。

spring-websocket 模块主要是与 Web 前端的全双工通讯的协议。

spring-webflux 是一个新的非堵塞函数式 Reactive Web 框架， 可以用来建立异步的， 非阻塞，事件驱动的服务， 并且扩展性非常好。

**5.报文发送**

即 spring-messaging模块。

spring-messaging 是从 Spring4 开始新加入的一个模块， 主要职责是为 Spring 框架集成一些基础的报文传送应用。

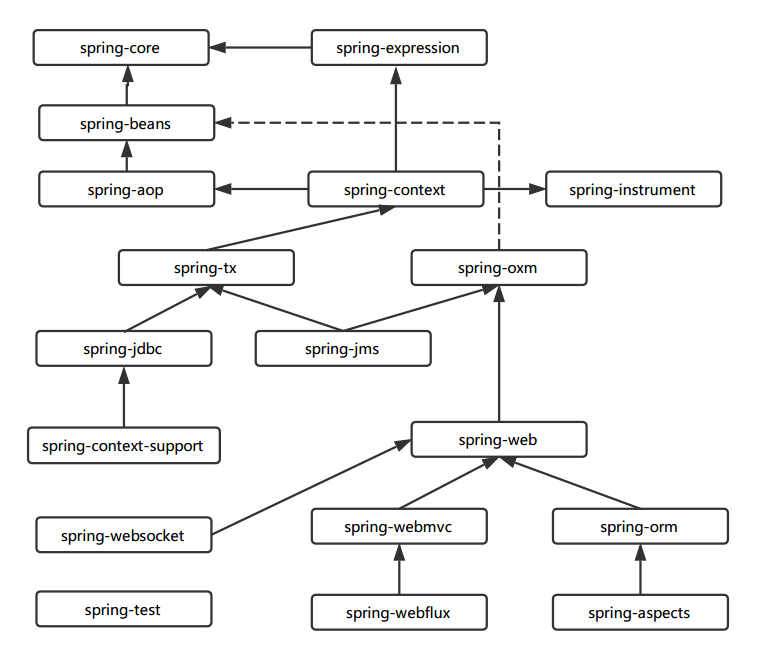
**6.Test：**

即 spring-test 模块。

spring-test 模块主要为测试提供支持的， 毕竟在不需要发布（程序） 到你的应用服务器或者连接到其他企业设施的情况下能够执行一些集成测试或者其他测试对于任何企业都是非常重要的。

**7.Spirng 各模块之间的依赖关系**

该图是 Spring5 的包结构， 可以从中清楚看出 Spring 各个模块之间的依赖关系。



spring5模块依赖关系图.png