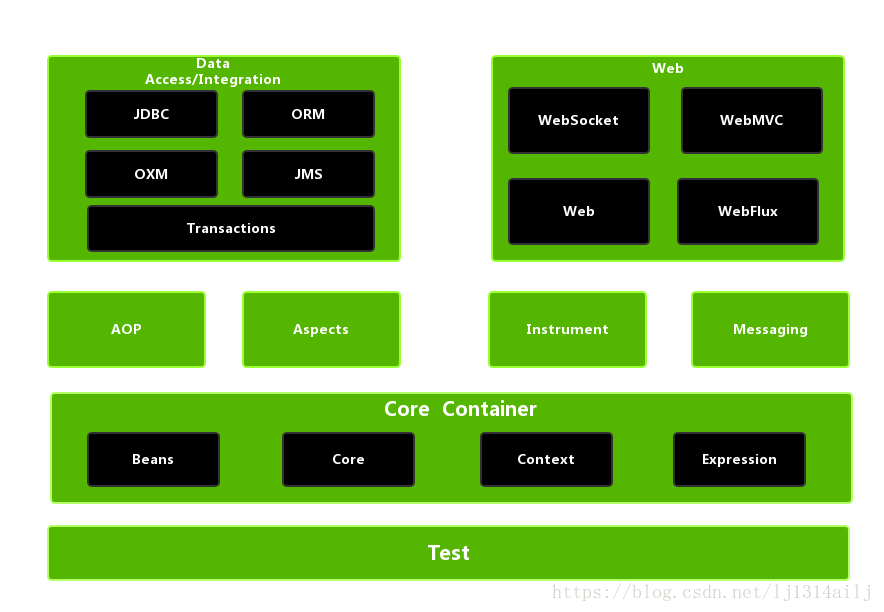
# 绿色农产品质量分析报告

## 1：该项目主要使用的框架



上图是Spring 5的模块结构图，而这些组件被分别整合在核心容器（Core Container）、AOP（Aspect Oriented Programming）、设备支持（Instrument）、数据访问及集成（Data Access）、Web、报文发送（Messaging）、Test等模块。

## 2:关于spring框架的详细介绍

1.核心容器：由spring-beans, spring-core, spring-context和spring-expression 4个模块组成。

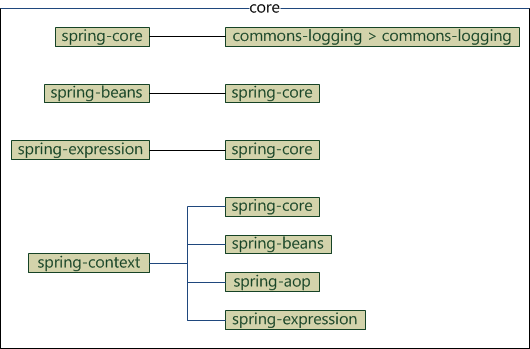
（1）spring-core：控制反转IOC（Inversion of Control）与依赖注入DI（Dependency Injection）的基本实现，控制反转是一种设计思想，即将你设计好的对象交给容器控制，而不是传统的在你的对象内部直接控制。

（2）spring-beans：Bean工厂与bean的装配，Bean Factory接口使用控制反转对应用程序的配置、依赖性规范与实际的应用程序代码进行分离。但是Bean Factory容器实例化后并不会自动实例化Bean，只有当Bean被使用时Bean Factory容器才会对该Bean进行实例化与依赖关系的装配。

（3）spring-context：spring的context上下文，即IOC容器，它扩展了Bean Factory，为它添加了Bean生命周期管理、框架事件体系、资源加载透明化等功能，此外该模块还提供了很多企业级支持，如远程访问、任务调度等。Application Context是该模块的核心接口，它扩展了Bean Factory，Application Context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。

（4）spring-expression：spring表达式语言，是统计表达式语言（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。最出色的要数函数调用和简单字符串的模板函数。

（5）核心容器框架图

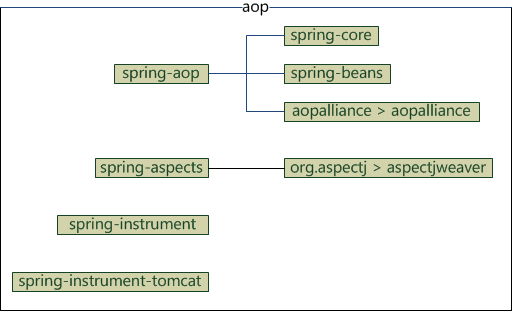


## 3：AOP和设备支持

（1）spring-AOP：面向切面编程，AOP就是在某一个类或方法执行前后打个标记，声明在执行到这里之前要先执行什么，执行完成之后要接着执行什么，插入新的执行方法。在Spring中，它是以JVM的动态代理技术为基础，然后设计一系列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。

（2）spring-aspects：集成AspectJ， 主要是为Spring AOOP提供多种AOP实现方法。

（3）spring-instrument：提供一些类级的工具支持和Class Loader级的实现，用于服务器。该模块是基于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，从而改变一个类的实现。



## 4.数据访问及集成

（1）spring-JDBC：JDBC的支持，主要是提供JDBC模板方式、关系数据库对象化方式、Simple JDBC方式、事务管理来简化JDBC编程，主要实现类JDBC Template、Simple JDBC Template、Named Parameter JDBC Template

（2）spring-tx：事务控制，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。

（3）spring-orm：对象关系映射，集成orm框架，主要集成Hibernate、Java Persistence API(JPA)和Java Data Object(JDO)用于资源管理、数据访问对象（DAO）的实现和事务策略。

（4）spring-oxm：对象xml映射，OXM是Object to XML Mapping的缩写，例如：jaxb, castor xmlbeans,JiBx,XStream等。

（5）spring-jms：java消息服务

（6）spring-messaging：为Spring框架集成一些基础的报文发送应用。

## 5：使用的spring框架进行该项目的优点

1.控制反转将对象的创建交给了spring,简化了开发，降低了代码之间的耦合性和侵入性。

2.方便对程序进行声明式事物管理，我们只需通过配置就可以完成对事物管理。

3.方便集成各种优秀的框架，spring不排斥各种优秀框架，其内部提供了对各种优秀框架如(struts2)等的直接支。

4.方便对程序进行测试，spring对于Junit4的支持，可通过注解方便测试程序。

5.降低了Java EE API的使用难度，JDBC, Java mail,远程调用等，spring对它们进行了封装，使这些API的使用难度大大降低。

6：总之，使用spring框架使得该项目代码更加清晰明了，操作起来方便，项目的整体性能得到了完善，非常好。