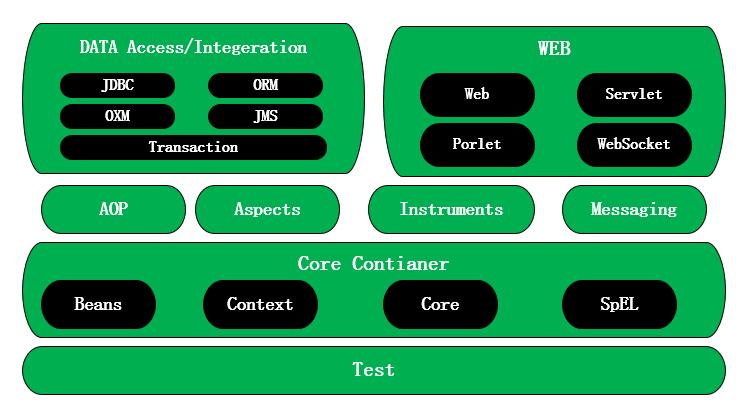
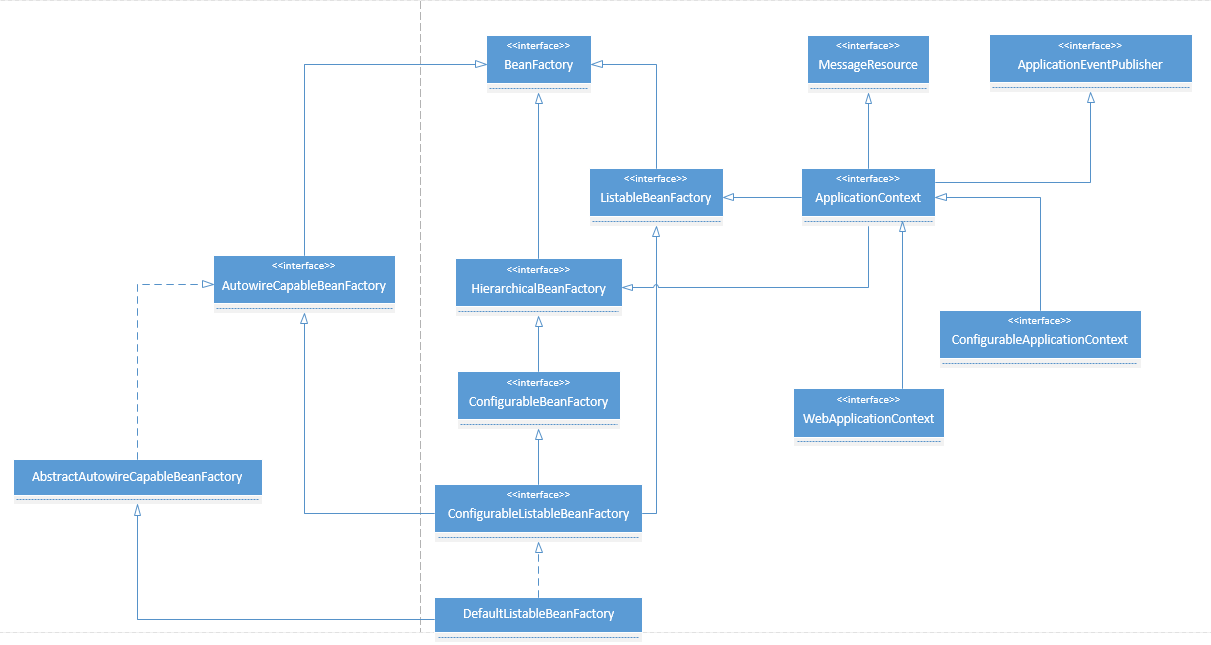
**SpringFramework**

1. **Spring体系结构**

Spring核心容器(IOC)：提供Spring框架的基本功能，管理着Spring应用中bean的创建、配置和管理。核心容器的主要组件BeanFactory和ApplicationContext，都可以看成容器的具体表现形式。

Spring提供的基本IOC容器的接口和定义的基础之上，spring通过定义BeanDefinition来管理各种spring对象之间的依赖关系。BeanDefinition抽象bean的定义，对于IOC容器来说，BeanDefinition就是对依赖反转模式管理依赖关系的的数据抽象，也是容器实现依赖反转功能的核心数据结构。

IOC容器设计的主要源码在Spring-beans和Spring-context这两个包中，主要接口设计如下：



第一条设计主线是：BeanFactory作为顶层接口类，它的三个子接口为ListableBeanFactory(说明这些bean是可列表的)，HierarchicalBeanFactory(表示这些bean是有继承关系的，每个bean有可能有父类bean，增加getParentBeanFactory()功能后，使得BeanFactory具备了双亲IOC容器的管理功能)，AutowireCapableBeanFactory(定义bean的自动装配规则)，默认实现类是DefaultListableBeanFactory。

第二条设计主线是：以ApplicationContext应用上下文为核心的设计。涉及的主要接口类有BeanFactory🡪ListableBeanFactory🡪ApplicationContext🡪 ConfigurableApplicationContext(WebApplicationContext)，ApplicationContext继承MessageSource(支持信息源，实现国际化)，ResourcePatternResolver(访问资源)，ApplicationEventPublisher(支持响应时间)

BeanFactory和FactoryBean的区别：

BeanFactory是IOC容器的编程抽象，主要是管理对象的创建、配置和建立对象之间的依赖

FactoryBean是工厂bean，可以看成是一个抽象工厂，对FactoryBean的调用返回的是工厂产生的bean(使用该容器时不会返回factory bean本身，而是返回其生成的对象)，所有的Factory bean都实现了FactoryBean接口，spring包括了大部分通用资源和服务的抽象(JNDI查询的处理，对代理对象的处理，对事务性的处理，对RMI代理的处理)

1. **Spring Bean的生命周期**

