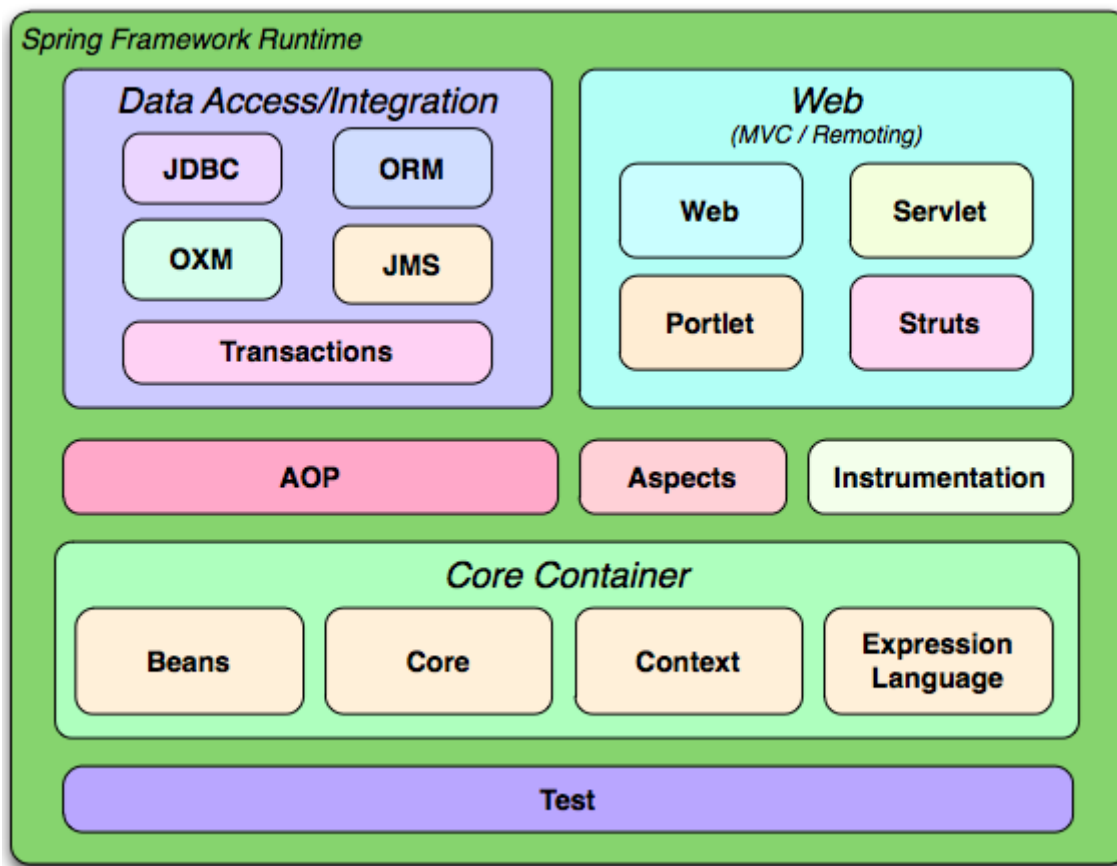


前言

- 前言
 - 1. 三种思想
 - 2. BeanFactory
 - 3. ApplicationContext
 - 4. BeanFactory与ApplicationContext的继承

1. 三种思想

- IoC，控制反转，强调的是将原来程序中创建Bean的权利反转给第三方
- DI，依赖注入，强调的是Bean之间的关系，这种关系由第三方负责去设置
- AOP，面向切面编程，功能的横向抽取，主要的实现方式就是Proxy



2. BeanFactory

核心的Bean工厂

开发步骤：

1. 引入BeanFactory，导入Spring
2. 定义Bean，定义UserService接口及其实现UserServiceImpl
3. 定义配置清单，创建beans.xml配置文件，将UserServiceImpl的信息配置到该xml中

4. 测试

3. ApplicationContext

ApplicationContext称为Spring容器，内部封装了BeanFactory，比BeanFactory功能更丰富强大，使用ApplicationContext进行开发时，xml配置文件的名称习惯写成applicationContext.xml

BeanFactory与ApplicationContext的关系

1. BeanFactory是Spring的早期接口，称为Spring的Bean工厂，ApplicationContext是后期更高级的接口，称之为Spring容器
2. ApplicationContext在BeanFactory的基础上对功能进行了扩展，例如：监听功能、国际化功能等。BeanFactory的API更偏向底层，ApplicationContext的API大多是对这些API的封装
3. Bean创建的主要逻辑和功能都被封装在BeanFactory中，ApplicationContext不仅集成了BeanFactory，而且ApplicationContext内部还维护着BeanFactory的引用，所以ApplicationContext与BeanFactory既有集成关系，又有融合关系
4. Bean的初始化机制不同，原始BeanFactory是在首次调用getBean时才进行Bean的创建，而ApplicationContext则是配置文件加载，容器一创建就将Bean都实例化好

4. BeanFactory与ApplicationContext的继承

- ApplicationContext中维护的是一个DefaultListableBeanFactory

只在Spring基础环境下，常用的ApplicationContext有：

- ClassPathApplicationContext，加载类路径配置文件
- FileSystemApplicationContext，加载磁盘路径下的配置文件
- AnnotationConfigApplicationContext，加载注解配置类

当导入spring-web包时，会有另外一套WebApplicationContext