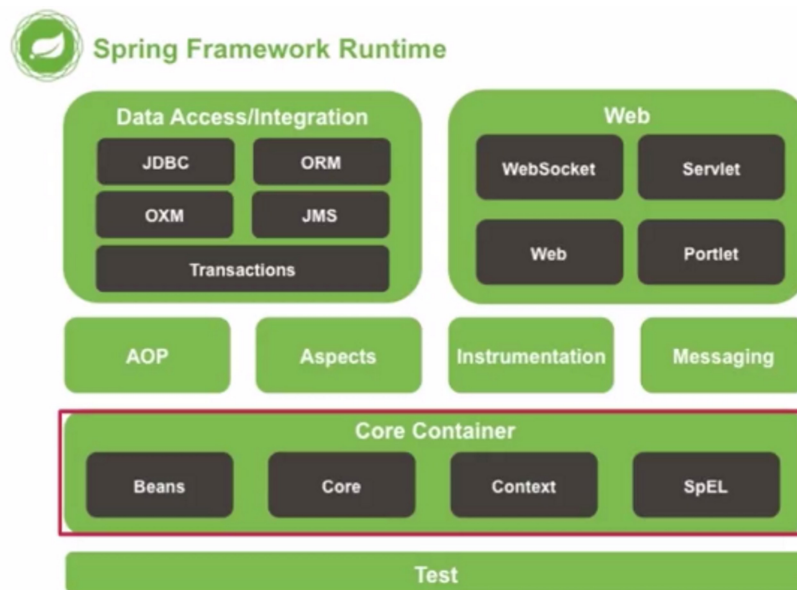


# Spring

2017年5月6日 15:59

## Spring 简介

- 1.Spring控制反转IoC面向切面AOP
- 2.面向切面编程的丰富支持，分离应用的业务逻辑与系统服务进行内聚性开发
- 3.包含并管理应用对象的配置和生命周期，这个意义上是一中容器
- 4.将简单的组件配置，组合成为复杂的应用，即框架
- 5.特点 简单，快捷，方便
- 6.核心框架



### 7.Spring的作用：

- 1.容器
- 2.多种技术的支持 ( JMS , MQ , Unit Test )
- 3.AOP
- 4.众多方便应用的辅助类
- 5.主流的应用框架 ( Hibernate )

## Spring注入方式

# Spring注入

- Spring注入是指在启动Spring容器加载bean配置的时候，完成对变量的赋值行为
- 常用的两种注入方式
  - 设值注入
  - 构造注入

创建过程中，对其赋值

设值注入：对属性或者set方法进行配置

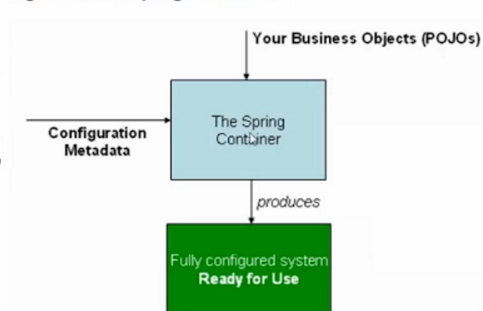
构造注入：一定要有构造器constructor-arg

## IOC控制反转

### 什么是IOC

- **IOC：控制反转**，控制权的转移，应用程序本身不负责依赖对象的创建和维护，而是由外部容器负责创建和维护
- **DI（依赖注入）是其一种实现方式**
- **目的**：创建对象并且组装对象之间的关系

Figure 4.1. The Spring IoC container



## Bean

### 专题二 Bean

- Bean配置项
- Bean的作用域
- Bean的生命周期
- Bean的自动装配
- Resources&ResourceLoader

### Bean的配置项

ID 唯一标识 必须！！ } 找到 加载bean的两种方式  
Class 需要实例化的类

Scope 范围 作用域

Constructor arguments 构造器的参数

Properties 属性

Autiwring mode 自动装配

Lazy-initialization mode 懒加载

Initialization destruction mmethod 初始化和销毁的方法

## Bean的作用域

- **singleton** : 单例 , 指一个Bean容器中只存在一份
- **prototype** : 每次请求 ( 每次使用 ) 创建新的实例 , destroy 方式不生效
- **request** : 每次http请求创建一个实例且仅在当前request内有效
- **session** : 同上 , 每次http请求创建 , 当前session内有效
- **global session** : 基于portlet的web中有效 ( portlet定义了 global session ) , 如果是在web中 , 同session

默认方式

## Bean的生命 周期

1.初始化 : 实现init-method方法 如下两种

- **初始化**

- 实现org.springframework.beans.factory.InitializingBean接口 , 覆盖afterPropertiesSet方法
- 配置init-method

```
<bean id="exampleInitBean" class="examples.ExampleBean" init-method="init"/>
```

```
public class ExampleBean {  
    public void init() {  
        // do some initialization work  
    }  
}
```

2.销毁 :

- **销毁**

- 实现org.springframework.beans.factory.DisposableBean接口 , 覆盖destroy方法
- 配置destroy-method

```
<bean id="exampleInitBean" class="examples.ExampleBean" destroy-method="cleanup"/>
```

```
public class ExampleBean {  
    public void cleanup() {  
        // do some destruction work (like releasing pooled connections)  
    }  
}
```

- **Spring中提供了一些以Aware结尾的接口，实现了Aware接口的bean在被初始化之后，可以获取相应资源**
- **通过Aware接口，可以对Spring相应资源进行操作（一定要慎重）**
- **为对Spring进行简单的扩展提供了方便的入口**

### Bean的自动装配

- 1.No：不做什么操作
- 2.byname：根据属性名自动装配，查找属性完全一致的bean
- 3.bytype：如果容器中存在一个属性相同的，则自动装配，如果有多个，那么出现异常
- 4.Constructor：与bytype类似，不同的是他应用构造参数，如果没有找到则抛出异常