

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！



Java私塾 《深入浅出学Spring3》 ——系列精品教程

10101010101010101010101010101

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

整体课程概览

- n 系统学习Spring3的核心开发知识，循序渐进
- n 第一章：Spring入门
包括：是什么、能干什么、有什么、各个组成部分的功能
- n 第二章：IoC/DI开发
包括：IoC/DI的概念、基本思想、基本实现方式、容器和Bean的基本原理、依赖注入的实现、Bean的作用域、ApplicationContext、Resource等
- n 第三章：AOP开发
包括：AOP是什么、有什么和能干什么、基本思想、Schema风格的实现、@AspectJ的支持、声明风格的选择、Spring AOP APIs、ProxyFactoryBean、设计上的考虑等
- n 第四章：对JDBC和ORM的支持
包括：DAO的支持、使用JDBC进行数据访问、使用ORM进行数据访问
- n 第五章：Spring中的事务
包括：事务的解决方案、声明式事务、编程式事务
- n 第六章：Struts2 + Spring3 + Hibernate3 整合开发
- n 第七章：Spring的表达式语言

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程概览

n Spring入门

包括：是什么、能干什么、有什么、各个组成部分的功能

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！



《深入浅出学Spring3开发》——系列精品教程

第一章：Spring入门

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

认识Spring-1

n Spring是什么

Spring是一种轻量级的、非侵入式的Java/JavaEE应用框架

n Spring背景

Rod Johnson

《Expert One-on-One J2EE Design and Development》（2002）这本书的中文译本为《J2EE 设计开发编程指南》（2003）

半年后，一个新的Java Framework发布，同样出自Rod Johnson 的手笔，这自然引起了大家极大的兴趣，这就是SpringFramework。SpringFramework 实际上是《Expert One-on-One J2EE Design and Development》一书中所阐述的设计思想的具体实现。

《Expert One-on-One J2EE Development without EJB》（2004）

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

认识Spring-2

n Spring目标

- 让J2EE开发变的简单、易用 (框架的好处)
- 业务应用“不”依靠 Spring API (非侵入性)
- 集成已存在的成熟应用解决方案 (如Hibernate, 不重复发明轮子)
- 利用面向对象的优秀技术 (面向接口而非实现编程, 组合优于继承)
- 促进好的编程习惯 (测试驱动开发, 测试简单, 面向接口编程)
- 让测试业务应用变得简单、快速 (无侵入性, 测试容易)

n Spring优点

- 有效地组织你的中间层对象
- 消除各式各样的配置文件
- 将面对接口编程做到实处
- 自由、不依赖框架
- 将单元测试变的简单
- 统一的数据存储方式
- 只选择你需要的

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供, 更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring能干什么

n Spring能干什么

笼统说：Spring能作为一个超级的”粘合”平台，把很多的技术有效整合到一起，形成一个整体，以更好的解决问题，并使系统结构更优良、性能更出众。

（体现体系结构设计的：组件式、装配式、插拔式、复用性、无侵入式等）

具体点说，Spring能干的事情很多，最基本的有以下几条：

- 1: IoC/DI
- 2: AOP
- 3: 对JDBC和ORM的支持
- 4: 服务(典型如：事务)
- 5: 各种技术的整合（典型如：Struts2+Spring3+Hibernate3整合，还有很多其他的技术整合，如EJB、JMS等）
- 6: Web应用框架（SpringMVC）
- 7: 更多技术，如：Spring的Security、Spring的Web Flow、Spring的WebService等等

真正高质量培训 签订就业协议

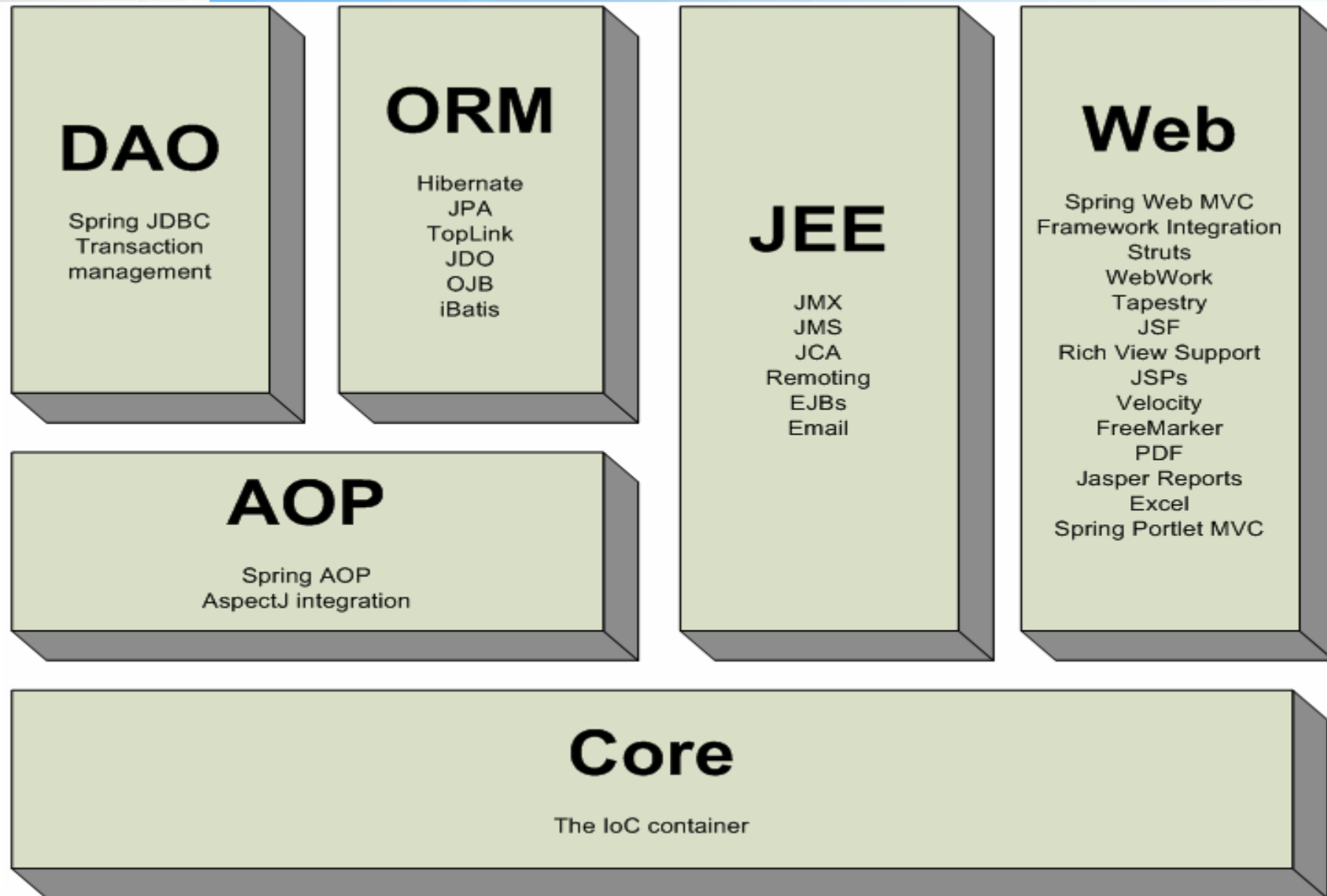
网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程



真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-2

n 上图是Spring框架2.5版时官方给出的概述图

n Core包

Core封装包是框架的最基础部分，提供IoC和依赖注入特性。这里的基础概念是BeanFactory，它提供对Factory模式的经典实现来消除对程序性单例模式的需要，并真正地允许你从程序逻辑中分离出依赖关系和配置。

构建于Core封装包基础上的 Context封装包，提供了一种框架式的对象访问方法，有些象JNDI注册器。Context封装包的特性得自于Beans封装包，并添加了对国际化（I18N）的支持（例如资源绑定），事件传播，资源装载的方式和Context的透明创建，比如说通过Servlet容器。

n DAO包

DAO提供了JDBC的抽象层，它可消除冗长的JDBC编码和解析数据库厂商特有的错误代码。还提供了一种比编程性更好的声明性事务管理方法，不仅仅是实现了特定接口，而且对所有的POJOs（plain old Java objects）都适用。

n ORM包

ORM 封装包提供了常用的“对象/关系”映射APIs的集成层。其中包括JPA、JDO、Hibernate 和 iBatis 。利用ORM封装包，可以混合使用所有Spring提供的特性进行“对象/关系”映射，如前边提到的简单声明性事务管理。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-3

n AOP包

Spring的 AOP 封装包提供了符合 AOP 联盟规范的面向方面的编程实现，让你可以定义如方法拦截器和切入点，从逻辑上讲，可以减弱代码的功能耦合，清晰的被分离开。而且，利用源码级的元数据功能，还可以将各种行为信息合并到你的代码中。

n Web包

Spring中的Web包提供了基础的针对Web开发的集成特性，例如多方文件上传，利用Servlet listeners进行IoC容器初始化和针对Web的application context。这个包使Spring可与其他框架结合。

n MVC包

Spring中的 MVC 封装包提供了Web应用的MVC实现。Spring的MVC框架并不是仅提供一种传统的实现，它提供了一种清晰的分离模型，在领域模型代码和web form之间。并且，还可以借助Spring框架的其他特性。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

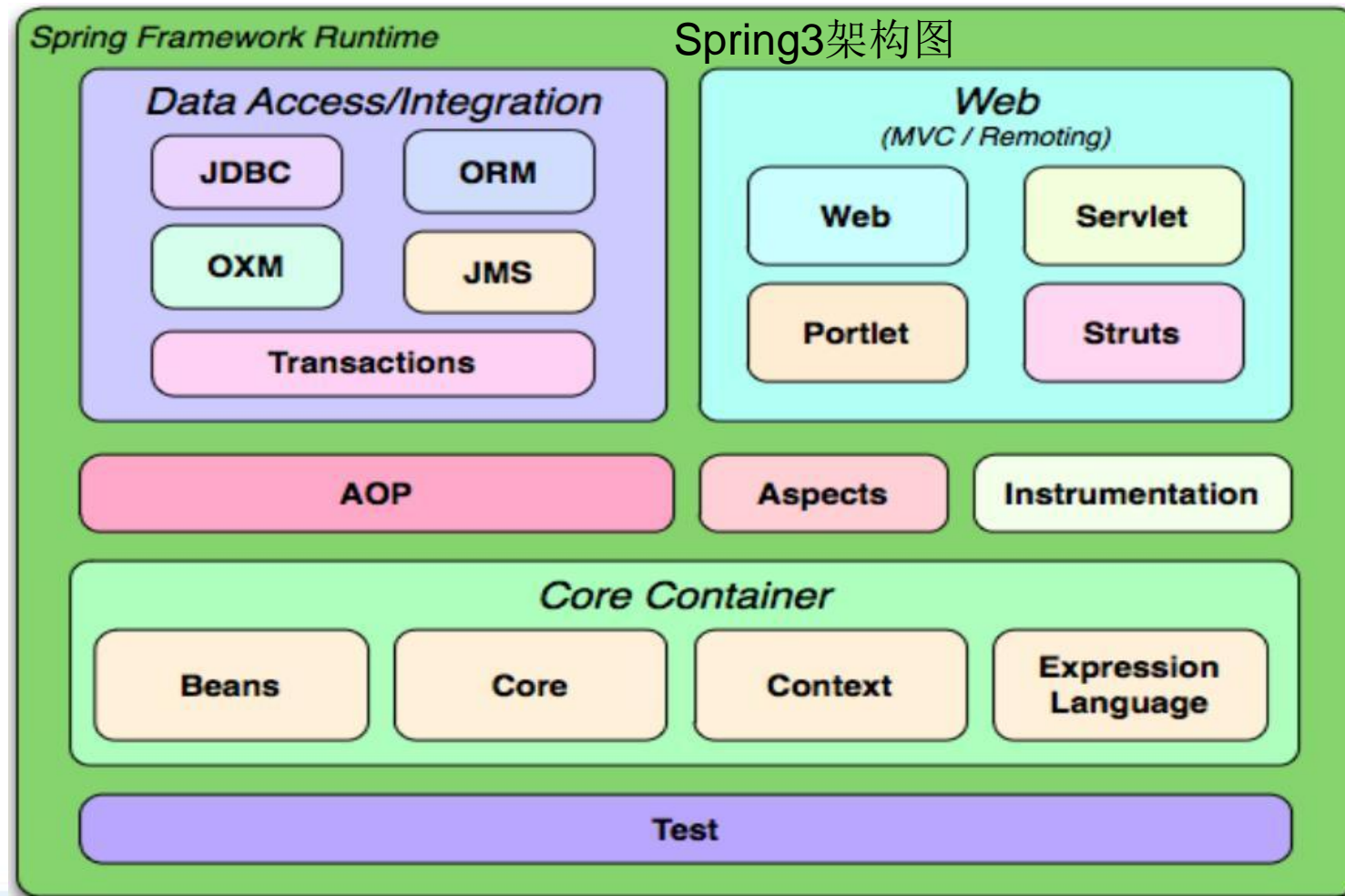
咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-4



真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-5

- n Spring3总共包含了大约20个模块，这些模块又被组织成为：
核心容器部分、数据访问/集成部分、Web部分、AOP部分和测试支持几个部分。
- n 核心容器部分(Core Container)又包含如下部分
 - 1: Core和Beans模块提供了Spring最基础的功能，提供IoC和依赖注入特性。这里的基础概念是BeanFactory，它提供对Factory模式的经典实现来消除对程序性单例模式的需要，并真正地允许你从程序逻辑中分离出依赖关系和配置。
 - 2: Context模块基于Core和Beans来构建，它提供了用一种框架风格的方式来访问对象，有些像JNDI注册表。Context封装包继承了beans包的功能，还增加了国际化（I18N），事件传播，资源装载，以及透明创建上下文，例如通过servlet容器，以及对大量JavaEE特性的支持，如EJB、JMX。核心接口是ApplicationContext。
 - 3: Expression Language，表达式语言模块，提供了在运行期间查询和操作对象图的强大能力。支持访问和修改属性值，方法调用，支持访问及修改数组、容器和索引器，命名变量，支持算数和逻辑运算，支持从Spring 容器获取Bean，它也支持列表投影、选择和一般的列表聚合等。

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>
咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-6

- n** 数据访问/集成部分(Data Access/Integration)又包含如下部分
- 1: JDBC模块，提供对JDBC的抽象，它可消除冗长的JDBC编码和解析数据库厂商特有的错误代码。
 - 2: ORM模块，提供了常用的“对象/关系”映射APIs的集成层。其中包括JPA、JDO、Hibernate 和 iBatis 。利用ORM封装包，可以混合使用所有Spring提供的特性进行“对象/关系”映射，如简单声明性事务管理 。
 - 3: OXM模块，提供一个支持Object和XML进行映射的抽象层，其中包括JAXB、Castor、XMLBeans、JiBX和XStream。
 - 4: JMS模块，提供一套“消息生产者、消费者”模板用于更加简单的使用JMS，JMS用于用于在两个应用程序之间，或分布式系统中发送消息，进行异步通信。
 - 5: Transaction模块，支持程序通过简单声明性事务管理，只要是Spring管理对象都能得到Spring管理事务的好处，即使是POJO，也可以为他们提供事务。

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>
咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-7

n Web部分又包含如下部分

- 1: Web模块，提供了基础的web功能。例如多文件上传、集成IoC容器、远程过程访问、以及Web Service支持，并提供一个RestTemplate类来提供方便的Restful services访问
- 2: Web-Servlet模块，提供了Web应用的Model-View-Controller (MVC) 实现。Spring MVC框架提供了基于注解的请求资源注入、更简单的数据绑定、数据验证等及一套非常易用的JSP标签，完全无缝与Spring其他技术协作。
- 3: Web-Struts模块， 提供了对Struts集成的支持，这个功能在Spring3.0里面已经不再推荐了，建议你迁移应用到使用Struts2.0或Spring的MVC。
- 4: Web-Portlet模块，提供了在Portlet环境下的MVC实现

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-8

n AOP部分又包含如下部分

- 1: AOP模块，提供了符合AOP 联盟规范的面向方面的编程实现，让你可以定义如方法拦截器和切入点，从逻辑上讲，可以减弱代码的功能耦合，清晰的被分离开。而且，利用源码级的元数据功能，还可以将各种行为信息合并到你的代码中。
- 2: Aspects模块，提供了对AspectJ的集成。
- 3: Instrumentation模块，提供一些类级的工具支持和ClassLoader级的实现，可以在一些特定的应用服务器中使用

n Test部分又包含如下部分

- 1: Test模块，提供对使用JUnit和TestNG来测试Spring组件的支持，它提供一致的ApplicationContexts并缓存这些上下文，它还能提供一些mock对象，使得你可以独立的测试代码。

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

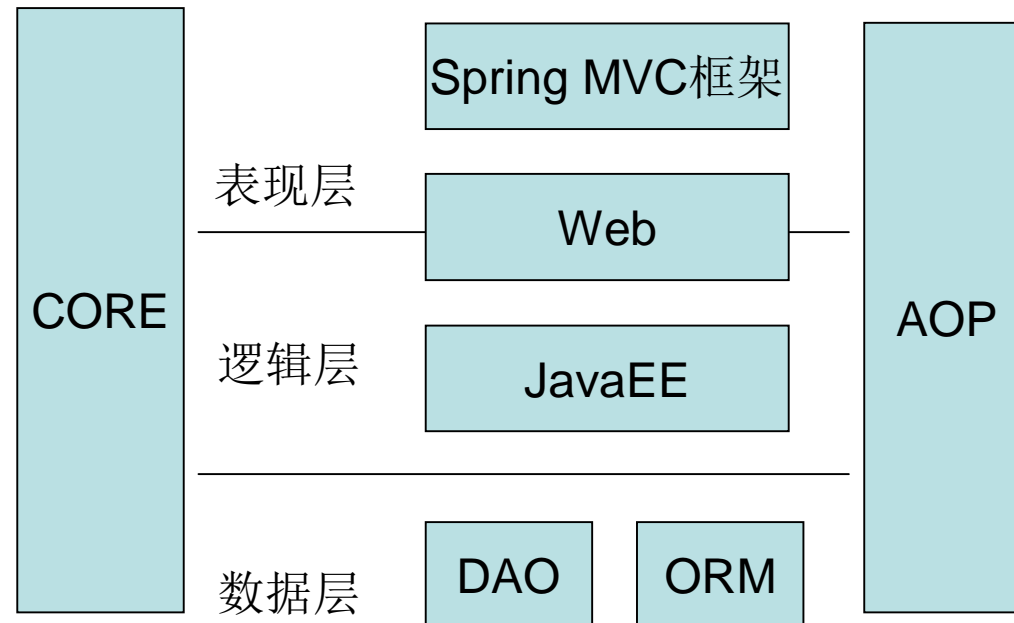
咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Spring有什么-9



真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程小结

n Spring是什么、能干什么、有什么

n 作业：复习和掌握这些理论知识

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程概览

n IoC/DI入门

包括：IoC/DI的概念、理解、基本思想的演变

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！



《深入浅出学Spring3开发》——系列精品教程

第二章：IoC/DI开发

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

IoC/DI概念

n IoC——Inversion of Control，控制反转

在Java开发中，IoC意味着将你设计好的类交给系统去控制，而不是在你的类内部控制。IoC是一种让服务消费者不直接依赖于服务提供者的组件设计方式，是一种减少类与类之间依赖的设计原则。

n DI——Dependency Injection（依赖注入）

即组件之间的依赖关系由容器在运行期决定，形象的来说，即由容器动态的将某种依赖关系注入到组件之中。

依赖注入的目标并非为软件系统带来更多的功能，而是为了提升组件重用的概率，并为系统搭建一个灵活、可扩展的平台。通过依赖注入机制，我们只需要通过简单的配置，而无需任何代码就可指定目标需要的资源，完成自身的业务逻辑，而不用关心具体的资源来自何处、由谁实现。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

IoC/DI理解

1: 控制反转:

谁控制谁？控制什么？为何叫反转（对应于正向）？哪些方面反转了？为何需要反转？

2: 依赖:

什么是依赖（按名词理解，按动词理解）？谁依赖于谁？为什么需要依赖？依赖什么东西？

3: 注入:

谁注入于谁？注入什么东西？为何要注入？

4: 依赖注入和控制反转是同一概念吗？

5: 参与者都有哪些？

6: IoC/DI是什么？能做什么？怎么做？用在什么地方？

还不能完全回答和理解，没有关系，先来看看IoC/DI的基本思想演变，然后再回头来回答这些问题

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

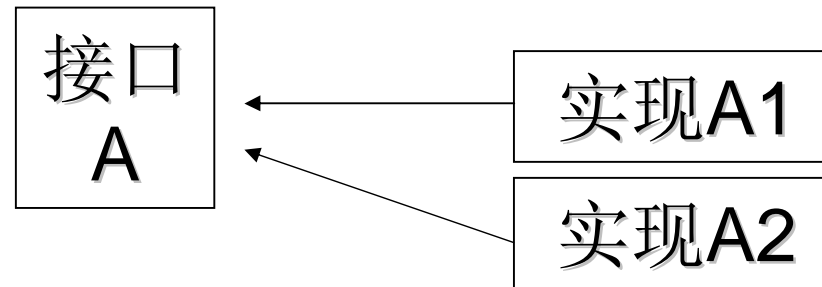
本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

IoC/DI基本思想-1

n Java基本阶段



在C类里面需要使用接口A，最原始的写法是：

A a = new A1(); 或 A a = new A2();
然后通过a来调用接口的方法。

对C来说是主动实例化对象，直接获取依赖

请问：这种方法好吗？有什么问题？

缺点：更换实现需要重新编译源代码

很难更换实现、难于测试

耦合实例生产者和实例消费者

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

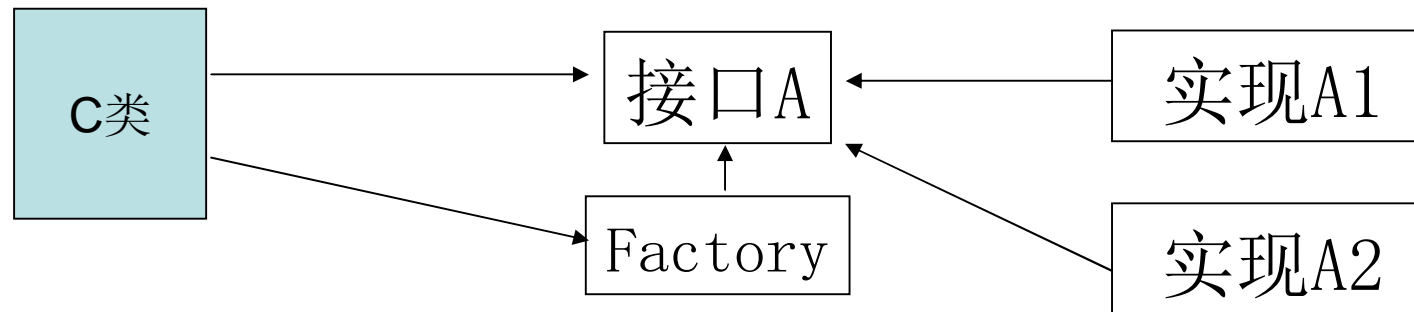
本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

IoC/DI基本思想-2

n Factory阶段



上述方法进化到使用工厂模式——Factory
在C类里面需要使用接口A，应用工厂模式的写法是：

```
A a = Factory.createA();
```

然后通过a来调用接口的方法。

对C来说是被动实例化对象，间接获取依赖，但对工厂类来说是主动的

请问：这种方法好吗？有什么问题？

缺点：更换实现需要重新编译源代码
很难更换实现、难于测试

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

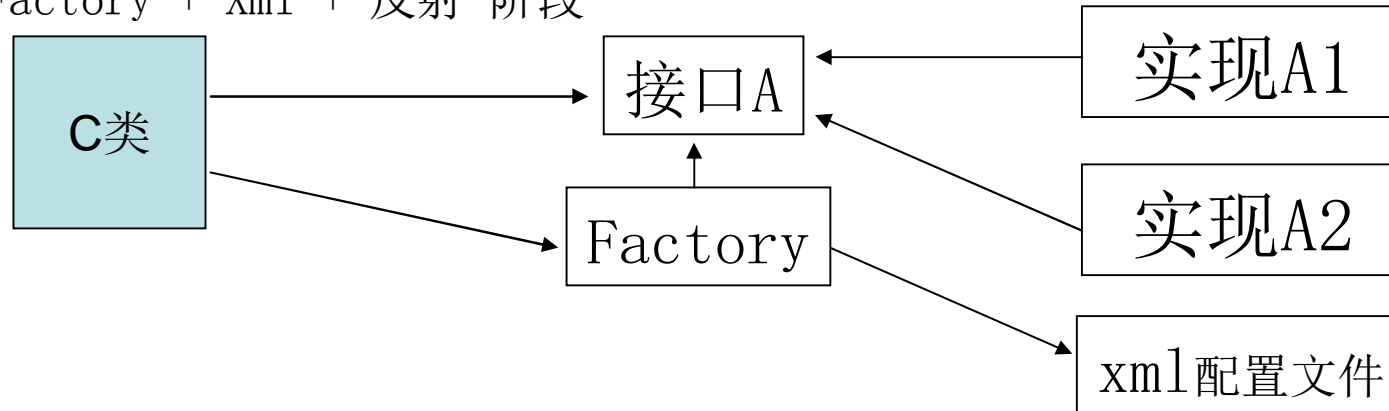
本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

IoC/DI基本思想-3

n Factory + Xml + 反射 阶段



n 继续进化到使用工厂模式加配置文件——Factory+xml+反射
在C类里面需要使用接口A，应用工厂模式的写法是：

```
A a = Factory.createA();
```

然后通过a来调用接口的方法。在工厂类里使用配置文件来决定要实例化的具体类。
对C类说是被动创建对象，间接获取依赖，对工厂来说也是被动的

在这种情况下，还会遇到很多的问题，比如：如何实例化带参数的类，如何在调用的方法传递值等等。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

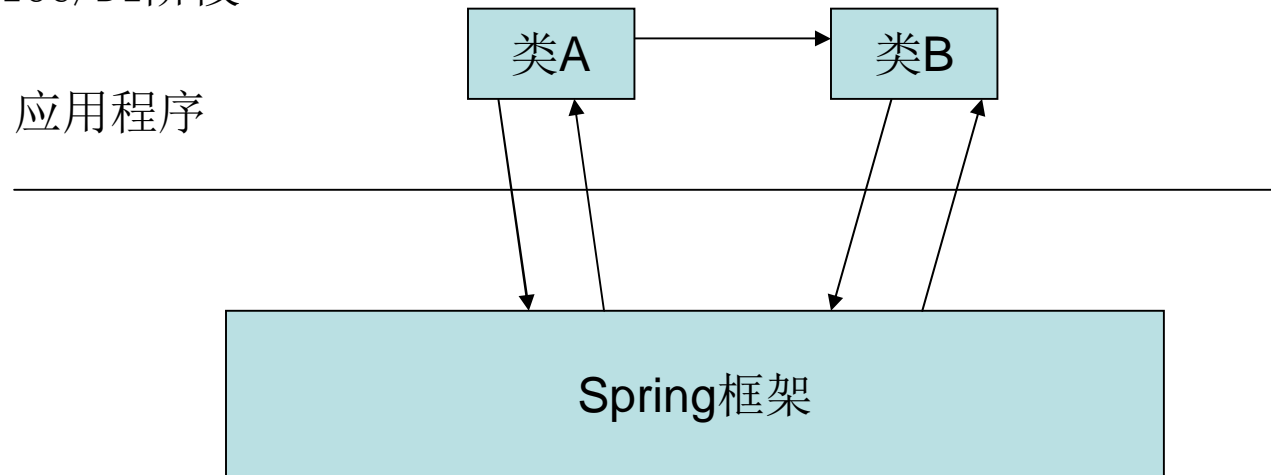
本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

IoC/DI基本思想-4

n IoC/DI阶段



继续进化，就到了IoC/DI的层次了，在这种设计里面，出现了IoC/DI容器，容器对IoC/DI涉及的元素进行整体控制，并提供更多更好的通用服务。

此时在C类里面需要使用接口A，以Spring为例：

```
A a = BeanFactory.getBean("XX");
```

然后通过a来调用接口的方法。

此时由容器来创建对象和装配对象，并管理对象生命周期。对于应用程序而言，就是被动实例化和被动接受依赖了。

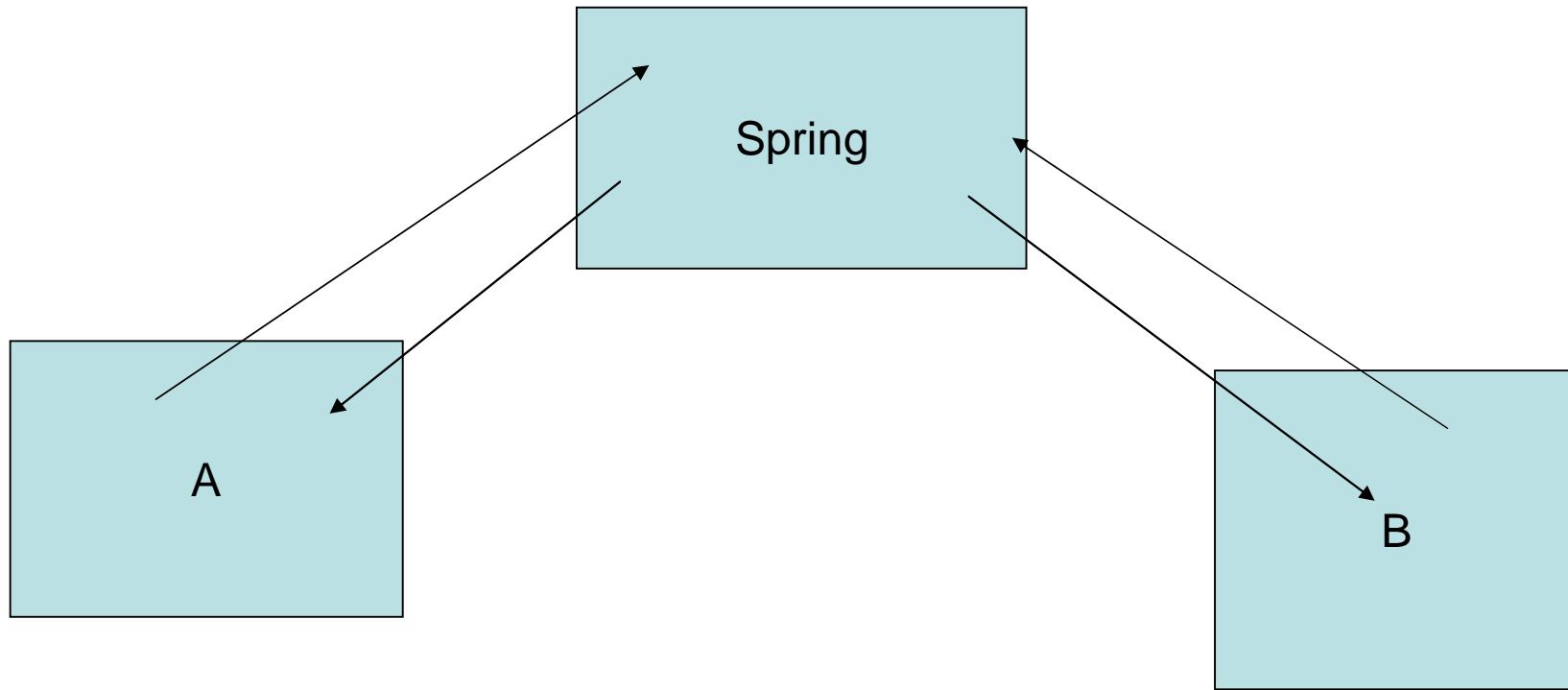
真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！





《深入浅出学Spring3开发》——系列精品教程

IoC/DI基本思想-5

n IoC容器

简单的理解就是：实现IoC思想，并提供对象创建、对象装配以及对象生命周期管理的软件就是IoC容器。

n IoC理解

- 1: 应用程序无需主动new对象；而是描述对象应该如何被创建即可
IoC容器帮你创建，即被动实例化；
- 2: 应用程序不需要主动装配对象之间的依赖关系，而是描述需要哪个服务
IoC容器会帮你装配（即负责将它们关联在一起），被动接受装配；
- 3: 主动变被动，体现好莱坞法则：别打电话给我们，我们会打给你
- 4: 体现迪米特法则（最少知识原则）：应用程序不知道依赖的具体实现，只知道需要提供某类服务的对象（面向接口编程）；并松散耦合，一个对象应当对其他对象有尽可能少的了解，不和陌生人（实现）说话
- 5: 是一种让服务消费者不直接依赖于服务提供者的组件设计方式，是一种减少类与类之间依赖的设计原则。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

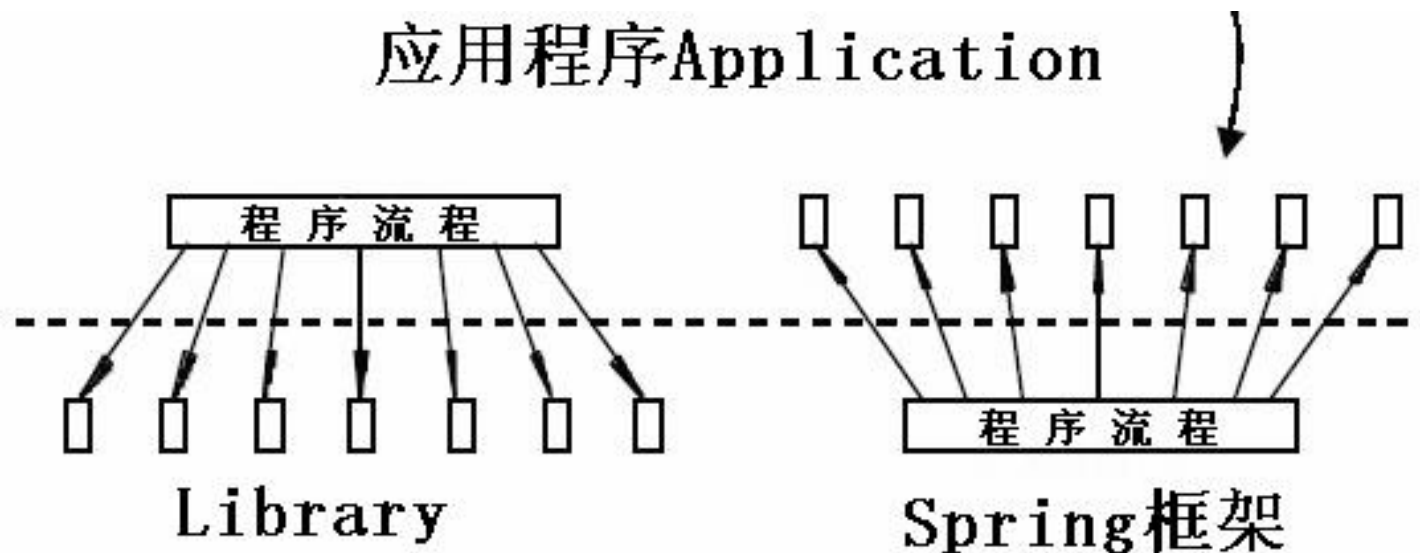
回答前面的问题

这个时候再回头看看第一部分里面提出的问题。

思考：

- 1: IoC/DI等同于工厂吗？
- 2: IoC/DI跟以前的方式有什么不一样？

领会：主从换位的思想



真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

IoC/DI基本思想-6

n 使用IoC/DI容器开发需要改变的思路

- 1: 应用程序不主动创建对象，但要描述创建它们的方式。
- 2: 在应用程序代码中不直接进行服务的装配，但要描述哪一个组件需要哪一项服务，由容器负责将这些装配在一起。

也就是说：所有的组件都是被动的，组件初始化和装配都由容器负责，应用程序只是在获取相应的组件后，实现应用的功能即可。

n 提醒一点

IoC/DI是思想，不是纯实现技术。IoC是框架共性，只是控制权的转移，转移到框架，所以不能因为实现了IoC就叫IoC容器，而一般除了实现了IoC外，还具有DI功能的才叫IoC容器，因为容器除了要负责创建并装配组件关系，还需要管理组件生命周期。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程小结

n IoC/DI入门

包括：IoC/DI的概念、理解、基本思想的演变

n 作业：复习和掌握这些理论知识，尤其是领会基本思想的演变。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程概览

n IoC/DI入门

包括：HelloWorld

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！



《深入浅出学Spring3开发》——系列精品教程

第二章：IoC/DI开发

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

HelloWorld-1

n 工具准备

- 1: Eclipse + Jdk6.0 , 示例用的Eclipse是Eclipse Java EE IDE for Web Developers, Version: Helios Service Release 1
- 2: spring-framework-3.1.0.M2-with-docs.zip

n 构建环境

- 1: 在Eclipse里面新建一个工程，设若名称是Spring3test
- 2: 把发行包里面的dist下面的jar包都添加到Eclipse里面
- 3: 根据Spring的工程来获取Spring需要的依赖包，在联网的情况下，通过Ant运行 projects/build-spring-framework/build.xml，会自动去下载所需要的jar包，下载后的包位于projects/ivy-cache/repository下面。
- 4: 为了方便，把这些jar包也添加到Eclipse里面

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

HelloWorld-2

n 开发接口

```
public interface HelloApi {  
    public String helloSpring3(int a);  
}
```

n 开发实现类

```
public class HelloImpl implements HelloApi{  
    public String helloSpring3(int a){  
        System.out.println("hello Spring3==="+a);  
        return "Ok, a="+a;  
    }  
}
```

n 配置文件

- 1: 在src下面新建一个文件叫applicationContext.xml
- 2: 在Spring发行包里面搜索一个例子，比如使用：
projects\org.springframework.context\src\test\java\org\springframework\jmx下的
applicationContext.xml，先把里面的配置都删掉，留下基本的xml定义和根元素就可以了，
它是一个DTD版的，而且还是2.0版的。
- 3: 建议使用Spring3的Schema版本，示例如下：

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

HelloWorld-3

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="
http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context-3.0.xsd
http://www.springframework.org/schema/aop
    http://www.springframework.org/schema/aop/spring-aop-3.0.xsd
http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx-3.0.xsd
">
.....
</beans>
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

HelloWorld-4

4: 配置applicationContext.xml如下:

```
<bean name="helloBean" class="cn.javass.Spring3.hello.HelloImpl"></bean>
```

n 编写客户端如下:

```
package cn.javass.Spring3.hello;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Client {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext(
            new String[] {"applicationContext.xml"});
        HelloApi api = (HelloApi)context.getBean("helloBean");
        String s = api.helloSpring3(3);
        System.out.println("the s="+s);
    }
}
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

HelloWorld-5

n 审视和结论

- 1: 所有代码中（除测试代码之外），并没有出现Spring的任何组件。
- 2: 客户代码（这里就是我们的测试代码）仅仅面向接口编程，而无需知道实现类的具体名称。同时，我们可以很简单的通过修改配置文件来切换具体的底层实现类。

结论

- 1: 首先，我们的组件并不需要实现框架指定的接口，因此可以轻松的将组件从Spring脱离，甚至不需要任何修改（这在基于EJB架实现的应用中是难以想象的）。
- 2: 其次，组件间的依赖关系减少，极大改善了代码的可重用性和可维护性
- 3: 面向接口编程

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Bean和容器

n 什么是Spring中的Bean

在Spring中，那些组成应用的主体及由Spring IoC容器所管理的对象被称之为bean。简单地讲，bean就是由Spring容器初始化、装配及被管理的对象，除此之外，bean就没有特别之处了(与应用中的其他对象没有什么区别)。而bean定义以及bean相互间的依赖关系将通过配置元数据来描述。

n 为什么使用Bean这个名字

使用‘bean’这个名字而不是‘组件’(component)或‘对象’(object)的动机源于Spring框架本身(部分原因则是相对于复杂的EJB而言的)。

n Spring的IoC容器

org.springframework.beans.factory.BeanFactory是Spring IoC容器的实际代表者，IoC容器负责容纳bean，并对bean进行管理。

Spring IoC容器将读取配置元数据；并通过它对应用中各个对象进行实例化、配置以及组装。通常情况下我们使用简单直观的XML来作为配置元数据的描述格式。在XML配置元数据中我们可以对那些我们希望通过Spring IoC容器管理的bean进行定义。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

BeanFactory和ApplicationContext

n IoC/DI是Spring最核心的功能之一，Spring框架所提供的众多功能之所以能成为一个整体正是建立在IoC的基础之上

n BeanFactory和ApplicationContext

org.springframework.beans及org.springframework.context包是Spring IoC容器的基础。BeanFactory提供的高级配置机制，使得管理任何性质的对象成为可能。

ApplicationContext是BeanFactory的扩展，功能得到了进一步增强，比如更易与Spring AOP集成、消息资源处理(国际化处理)、事件传递及各种不同应用层的context实现(如针对web应用的WebApplicationContext)。

n 接口选择之惑

在实际应用中，用户有时候不知道到底是选择BeanFactory接口还是ApplicationContext接口。但是通常在构建JavaEE应用时，使用ApplicationContext将是更好的选择，因为它不仅提供了BeanFactory的所有特性，同时也允许使用更多的声明方式来得到我们想要的功能。

简而言之，BeanFactory提供了配制框架及基本功能，而ApplicationContext则增加了更多支持企业核心内容的功能。ApplicationContext完全由BeanFactory扩展而来，因而BeanFactory所具备的能力和行为也适用于ApplicationContext。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

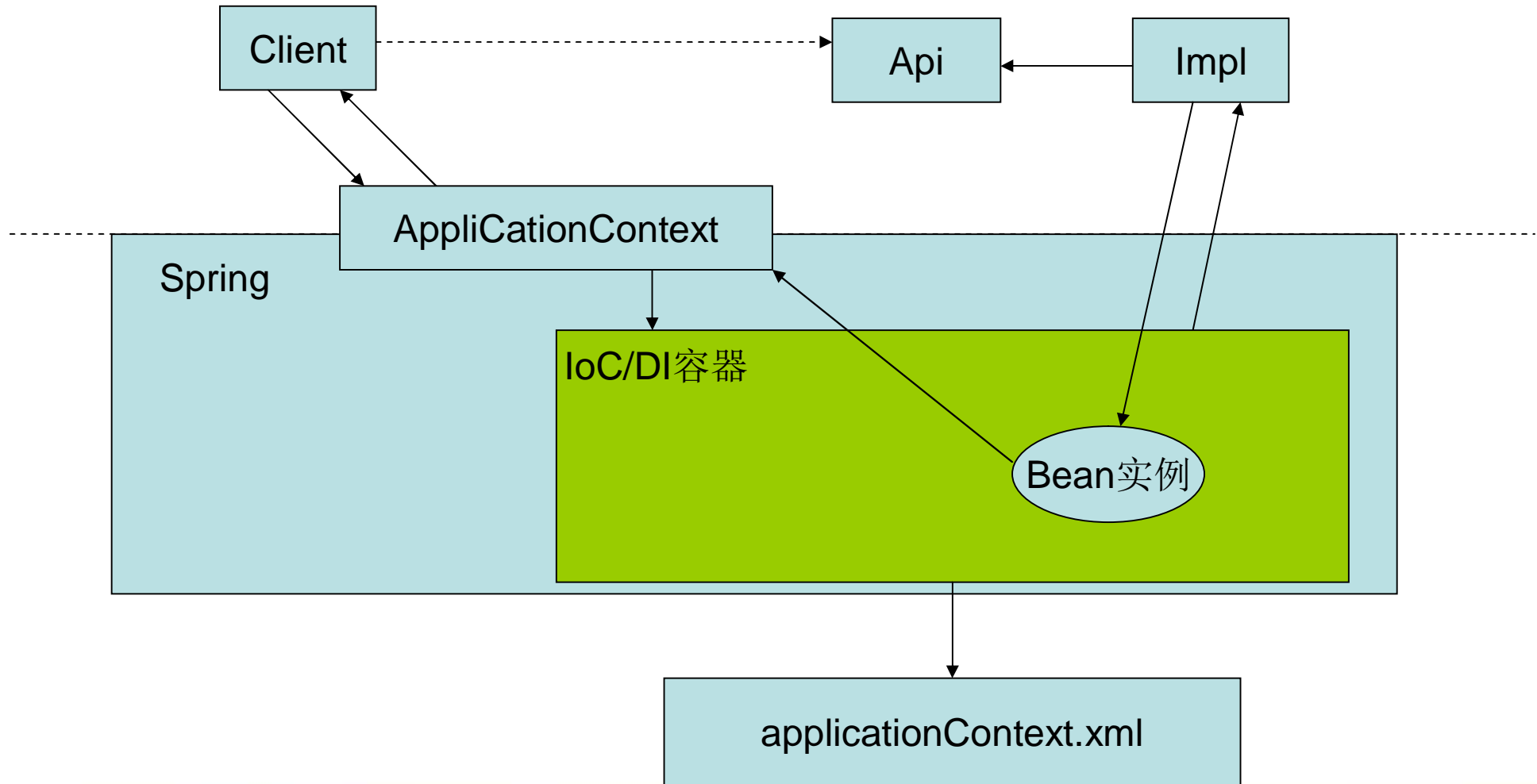
咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

HelloWorld的基本调用过程



真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程小结

- n 掌握IoC/DI的HelloWorld的开发
- n 理解HelloWorld的基本调用过程

- n 作业：
 - 1: 复习和掌握这些理论知识
 - 2: 按照课程讲述，进行HelloWorld的开发
 - 3: 通过开发去实际体会以下HelloWorld的基本调用过程

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程概览

n IoC/DI开发

包括：基本的配置使用

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！



《深入浅出学Spring3开发》——系列精品教程

第二章：IoC/DI开发

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

实例化容器

Spring IoC容器的实例化非常简单，如下面的例子：

1：第一种：

```
Resource resource = new FileSystemResource("beans.xml");  
BeanFactory factory = new XmlBeanFactory(resource);
```

2：第二种：

```
ClassPathResource resource = new ClassPathResource("beans.xml");  
BeanFactory factory = new XmlBeanFactory(resource);
```

3：第三种：

```
ApplicationContext context = new ClassPathXmlApplicationContext(  
    new String[] {"applicationContext.xml", "applicationContext-  
    part2.xml"});  
// of course, an ApplicationContext is just a BeanFactory  
BeanFactory factory = (BeanFactory) context;
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

多配置文件

n 读取多个配置文件

第一种方法：

为了加载多个XML文件生成一个ApplicationContext实例，可以将文件路径作为字符串数组传给ApplicationContext构造器。而bean factory将通过调用bean definition reader从多个文件中读取bean定义。通常情况下，Spring团队倾向于上述做法，因为这样各个配置并不会查觉到它们与其他配置文件的组合。

第二种方法：

使用一个或多个的<import/>元素来从另外一个或多个文件加载bean定义。所有的<import/>元素必须放在<bean/>元素之前以完成bean定义的导入。让我们看个例子：

```
<beans><import resource="services.xml"/>
<import resource="/resources/messageSource.xml"/>
<import resource="/resources/themeSource.xml"/>
<bean id="bean1" class="..." />
<bean id="bean2" class="..." />
</beans>
```

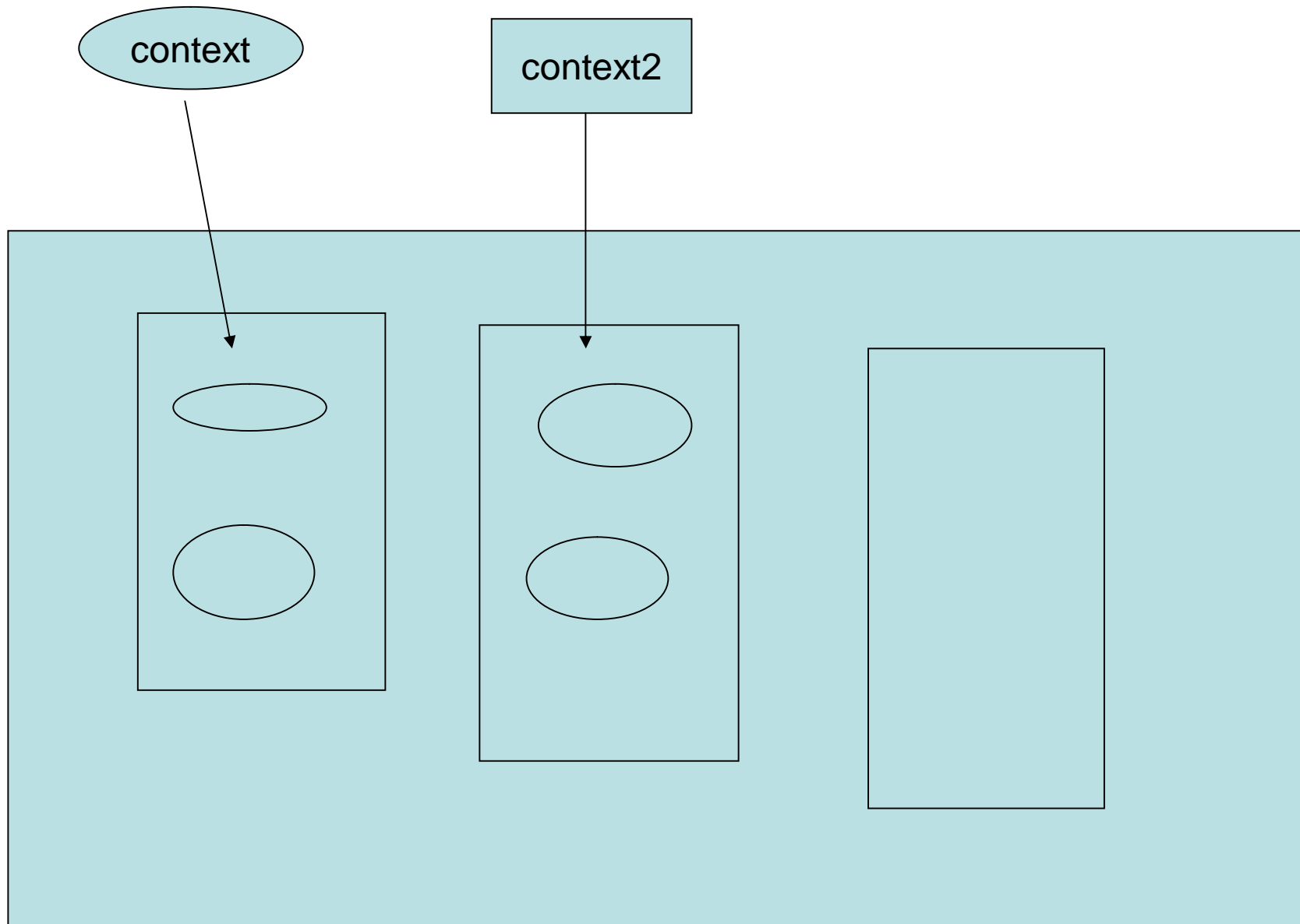
真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！





《深入浅出学Spring3开发》——系列精品教程

配置文件的配置内容

n 配置文件中常见的配置内容

在IoC容器内部，bean的定义由BeanDefinition 对象来表示，该定义将包含以下信息：

- 1: **全限定类名**：这通常就是已定义bean的实际实现类。如果通过调用static factory方法来实例化bean，而不是使用常规的构造器，那么类名称实际上就是工厂类的类名。
- 2: **bean行为的定义**，即创建模式（prototype还是singleton）、自动装配模式、依赖检查模式、初始化以及销毁方法。这些定义将决定bean在容器中的行为。
- 3: **用于创建bean实例的构造器参数及属性值**。比如使用bean来定义连接池，可以通过属性或者构造参数指定连接数，以及连接池大小限制等。
- 4: **bean之间的关系**，即协作（或者称依赖）。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

命名Bean

n Bean的命名

每个bean都有一个或多个id(或称之为标识符或名称，在术语上可以理解成一回事)，这些id在当前IoC容器中必须唯一。

当然也可以为每个bean定义一个name，但是并不是必须的，如果没有指定，那么容器将为其生成一个惟一的name。对于不指定name属性的原因我们会在后面介绍(比如内部bean就不需要)。

n Bean命名的约定

bean的命名采用标准的Java命名约定，即小写字母开头，首字母大写间隔的命名方式。如accountManager、accountService等等。

对bean采用统一的命名约定将会使配置更加简单易懂。而且在使用Spring AOP，这种简单的命名方式将会令你受益匪浅。

n Bean的别名

一个Bean要提供多个名称，可以通过alias属性来加以指定，示例如下：

```
<alias name="fromName" alias="toName"/>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

实例化Bean-1

n 容器如何实例化Bean

当采用XML描述配置元数据时，将通过<bean/>元素的class属性来指定实例化对象的类型。class属性主要有两种用途：在大多数情况下，容器将直接通过反射调用指定类的构造器来创建bean(这有点类似于在Java代码中使用new操作符)；在极少数情况下，容器将调用类的静态工厂方法来创建bean实例，class属性将用来指定实际具有静态工厂方法的类(至于调用静态工厂方法创建的对象类型是当前class还是其他的class则无关紧要)。

n 用构造器来实例化Bean，前面的实例就是

n 使用静态工厂方法实例化

采用静态工厂方法创建bean时，除了需要指定class属性外，还需要通过factory-method属性来指定创建bean实例的工厂方法，示例如下：

```
<bean id="exampleBean"  
      class="examples.ExampleBean2"  
      factory-method="createInstance"/>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

实例化Bean-2

n 使用实例工厂方法实例化

使用此机制，class属性必须为空，而factory-bean属性必须指定为当前（或其祖先）容器中包含工厂方法的bean的名称，而该工厂bean的工厂方法本身必须通过factory-method属性来设定，并且这个方法不能是静态的，示例如下：

```
<bean id="exampleBean" factory-bean="myFactoryBean" factory-  
method="createInstance"/>
```

n 使用容器

从本质上讲，BeanFactory仅仅只是一个维护bean定义以及相互依赖关系的高级工厂接口。使用getBean(String)方法就可以取得bean的实例；BeanFactory提供的方法极其简单。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

依赖注入介绍

n 依赖注入（DI）背后的基本原理

是对象之间的依赖关系（即一起工作的其它对象）只会通过以下几种方式来实现：**构造器的参数、工厂方法的参数，或给由构造函数或者工厂方法创建的对象设置属性。**

因此，容器的工作就是创建bean时注入那些依赖关系。相对于由bean自己来控制其实例化、直接在构造器中指定依赖关系或则类似服务定位器（Service Locator）模式这3种自主控制依赖关系注入的方法来说，控制从根本上发生了倒转，这也正是控制反转IoC名字的由来。

n 应用依赖注入（DI）的好处、

应用DI原则后，代码将更加清晰。而且当bean自己不再担心对象之间的依赖关系（以及在何时何地指定这种依赖关系和依赖的实际类是什么）之后，实现更高层次的**松耦合**将易如反掌。

n 依赖注入（DI）基本的实现方式

DI主要有两种注入方式，即**Setter注入**和 **构造器注入**。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

setter注入

n 通过调用无参构造器或无参static工厂方法实例化bean之后，调用该bean的setter方法，即可实现基于setter的DI。 示例如下：

n 示例——Java类

```
public class HelloImpl implements HelloApi{
    private String name = "";
    public void setName(String name){
        this.name = name;
    }
    public String helloSpring3(int a){
        System.out.println("hello Spring3==="+a+", name="+name);
        return "Ok, a="+a;
    }
}
```

n 示例——配置文件

```
<bean name="helloBean" class="cn.javass.Spring3.hello.HelloImpl">
    <property name="name"><value>javass Spring3</value></property>
</bean>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

构造器注入

n 示例——Java类

```
public class HelloImpl implements HelloApi {  
    private String name = "";  
    public HelloImpl(String name) {  
        this.name = name;  
    }  
    public String helloSpring3(int a) {  
        System.out.println("hello Spring3===" + a + ", name=" + name);  
        return "Ok, a=" + a;  
    }  
}
```

n 示例——配置文件

```
<bean name="helloBean" class="cn.javass.Spring3.hello.HelloImpl">  
    <constructor-arg><value>javass Spring3</value></constructor-arg>  
</bean>
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

构造器参数解析-1

n 默认解析方式

构造器参数将根据类型来进行匹配。如果bean定义中的构造器参数类型明确，那么bean定义中的参数顺序就是对应构造器参数的顺序

n 构造器参数类型匹配

可以使用type属性来显式的指定参数所对应的简单类型。例如：

```
<bean id="exampleBean" class="examples.ExampleBean">  
  <constructor-arg type="int" value="7500000"/>  
  <constructor-arg type="java.lang.String" value="42"/>  
</bean>
```

n 构造器参数的索引

使用index属性可以显式的指定构造器参数出现顺序。例如：

```
<bean id="exampleBean" class="examples.ExampleBean">  
  <constructor-arg index="0" value="7500000"/>  
  <constructor-arg index="1" value="42"/>  
</bean>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

构造器参数解析-2

n 构造器参数的名称

在Spring3里面, 可以使用构造器参数的名称来直接赋值。例如:

```
<bean id="exampleBean" class="examples.ExampleBean">  
    <constructor-arg name="years" value="7500000"/>  
    <constructor-arg name="ultimateanswer" value="42"/>  
</bean>
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供, 更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-1

n 直接量(基本类型、Strings类型等)

<value/>元素通过字符串来指定属性或构造器参数的值。JavaBean属性编辑器将把字符串从java.lang.String类型转化为实际的属性或参数类型。示例：

```
<bean id="myDataSource" class="org.apache.commons.dbcp.BasicDataSource">
  <property name="driverClassName">
    <value>oracle.jdbc.driver.OracleDriver</value>
  </property>
  <property name="url">
    <value>jdbc:oracle:thin:@localhost:1521:orcl</value>
  </property>
  <property name="username"> <value>test</value> </property>
  <property name="password" value="test"/>
</bean>
```

Value可以做为子元素或者是属性使用。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-2

n idref元素

idref元素用来将容器内其它bean的id传给<constructor-arg/> 或 <property/>元素，同时提供错误验证功能。idref元素和<value>差不多，只是传递一个字符串，用来方便xml检查。示例如下：

```
<bean id="theTargetBean" class="..." />
<bean id="theClientBean" class="...">
    <property name="targetName"> <idref bean="theTargetBean" />
</property>
</bean>
```

上述bean定义片段完全地等同于（在运行时）以下的片段

```
<bean id="theTargetBean" class="..." />
<bean id="client" class="...">
    <property name="targetName"> <value>theTargetBean</value> </property>
</bean>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-3

n idref元素 续

第一种形式比第二种更可取的主要原因是，使用idref标记允许容器在部署时验证所被引用的bean是否存在。而第二种方式中，传给client bean的targetName属性值并没有被验证。任何的输入错误仅在client bean实际实例化时才会被发现（可能伴随着致命的错误）。如果client bean是prototype类型的bean，则此输入错误（及由此导致的异常）可能在容器部署很久以后才会被发现。

如果被引用的bean在同一XML文件内，且bean名字就是bean id，那么可以使用local属性，此属性允许XML解析器在解析XML文件时来对引用的bean进行验证，示例如下：

```
<property name="targetName">
    <idref local="theTargetBean"/>
</property>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>
咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-4

n 引用其它的bean（协作者） ——ref元素

尽管都是对另外一个对象的引用，但是通过id/name指向另外一个对象却有三种不同的形式，不同的形式将决定如何处理作用域及验证。

1: 第一种形式也是最常见的形式是使用<ref/>标记指定目标bean，示例：

```
<ref bean= "someBean" />
```

2: 第二种形式是使用ref的local属性指定目标bean，它可以利用XML解析器来验证所引用的bean是否存在同一文件中。示例：

```
<ref local="someBean"/>
```

3: 第三种方式是通过使用ref的parent属性来引用当前容器的父容器中的bean，并不常用。示例：

```
<bean id="accountService" class="com.foo.SimpleAccountService"> </bean>
```

```
<bean id= "accountService" <-- 注意这里的名字和parent的名字是一样的-->  
class="org.springframework.aop.framework.ProxyFactoryBean">
```

```
<property name="target"><ref parent="accountService"/> </property>  
</bean>
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-5

n 内部Bean

所谓的内部bean (inner bean) 是指在一个bean的<property/>或<constructor-arg/>元素中使用<bean/>元素定义的bean。内部bean定义不需要有id或name属性，即使指定id 或 name属性值也将会被容器忽略。示例：

```
<bean id="outer" class="...">
  <property name="target">
    <bean class="com.mycompany.Person">
      <property name="name" value="Fiona Apple"/>
      <property name="age" value="25"/>
    </bean>
  </property>
</bean>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程小结

n IoC/DI开发

包括：基本的配置使用

- n 作业：复习和掌握这些理论知识，一定要动手去配置试验一下，会有助于加深理解和掌握。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程概览

n IoC/DI开发

包括：基本配置、自动装配、依赖检查、Bean的作用域。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！



《深入浅出学Spring3开发》——系列精品教程

第二章：IoC/DI开发

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-6

- n** 通过<list/>、<set/>、<map/>及<props/>元素可以定义和设置与Java Collection类型对应List、Set、Map及Properties的值，示例如下：

```
<bean id="moreComplexObject" class="example.ComplexObject">
  <property name="adminEmails">
    <props>
      <prop key="administrator">admin@somecompany.org</prop>
      <prop key="support">support@somecompany.org</prop>
    </props>
  </property>
  <property name="someList">
    <list>
      <value>a list element followed by a reference</value>
      <ref bean="myDataSource" />
    </list>
  </property>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-7

```
<property name="someMap">
    <map>
        <entry>
            <key> <value>yup an entry</value> </key>
            <value>just some string</value>
        </entry>
        <entry>
            <key> <value>yup a ref</value> </key>
            <ref bean="myDataSource" />
        </entry>
    </map>
</property>
<property name="someSet">
    <set>
        <value>just some string</value>
        <ref bean="myDataSource" />
    </set>
</property>
</bean>
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

集合合并-1

n 可以定义parent-style和child-style的<list/>、<map/>、<set/>或<props/>元素，子集合的值从其父集合继承和覆盖而来；也就是说，父子集合元素合并后的值就是子集合中的最终结果，而且子集合中的元素值将覆盖父集全中对应的值。

```
<beans>
<bean id="parent" abstract="true" class="example.ComplexObject">
  <property name="adminEmails">
    <props>
      <prop key="administrator">administrator@somecompany.com</prop>
      <prop key="support">support@somecompany.com</prop>
    </props>
  </property>
</bean>
<bean id="child" parent="parent" class="example.Child">
  <property name="adminEmails">
    <props merge="true">
      <prop key="sales">sales@somecompany.com</prop>
      <prop key="support">support@somecompany.co.uk</prop>
    </props>
  </property>
</bean>
</beans>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

集合合并-2

- n 在上面的例子中，childbean的adminEmails属性的<props/>元素上使用了merge=true属性。当child bean被容器实际解析及实例化时，其 adminEmails将与父集合的adminEmails属性进行合并。
- n 注意到这里子bean的Properties集合将从父<props/>继承所有属性元素。同时子bean的support值将覆盖父集合的相应值
- n 不同的集合类型是不能合并（如map和 list是不能合并的），否则将会抛出相应的Exception。merge属性必须在继承的子bean中定义，而在父bean的集合属性上指定的merge属性将被忽略
- n 在JDK5以上的版本里，Spring支持强类型集合



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-8

n <null/>用于处理null值，Spring会把属性的空参数当作空字符串处理。

1: 以下的xml片断将email属性设为空字符串。

```
<bean class="ExampleBean">
    <property name="email"><value></value></property>
</bean>
```

这等同于Java代码： exampleBean.setEmail(“”)。

2: 而null值则可以使用<null>元素可用来表示。例如：

```
<bean class="ExampleBean">
    <property name="email"><null/></property>
</bean>
```

上述的配置等同于Java代码： exampleBean.setEmail(null)。

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-9

n 针对常见的value值或bean的引用，Spring提供了简化格式用于替代<value/>和<ref/>元素。如下：

```
<property name="myProperty">
  <value>hello</value>
</property>
<property name="myProperty">
  <ref bean="myBean">
</property>
<entry>
  <key>
    <ref bean="myKeyBean" />
  </key>
  <ref bean="myValueBean" />
</entry>
```

完全等同于

```
<property name="myProperty" value="hello"/>
<property name="myProperty" ref="myBean"/>
<entry key-ref="myKeyBean" value-ref="myValueBean"/>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-10

n 强调一点

只有`<ref bean=“xxx”>`元素的简写形式，没有`<ref local=“xxx”>`的简写形式。也就是说`<property name=“myProperty” ref=“myBean” />`里面的ref是相当于`<ref bean=“”>`的形式。

n 组合属性名称

当设置bean的组合属性时，除了最后一个属性外，只要其他属性值不为null，组合或嵌套属性名是完全合法的。例如，下面bean的定义：

```
<bean id="foo" class="foo.Bar">
  <property name="fred.bob.sammy" value="123" />
</bean>
```

表示foo bean有个fred属性，此属性有个bob属性，而bob属性又有个sammy属性，最后把sammy属性设置为123。为了让此定义能工作，foo的fred属性及fred的bob属性在bean被构造后都必须非空，否则将抛出NullPointerException异常。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-11

n depends-on，用于当前bean初始化之前显式地强制一个或多个bean被初始化。
示例：

```
<bean id="beanOne" class="ExampleBean" depends-on="manager"/>
<bean id="manager" class="ManagerBean" />
```

若需要表达对多个bean的依赖，可以在' depends-on' 中将指定的多个bean名字用分隔符进行分隔，分隔符可以是逗号、空格及分号等。下面的例子中使用了' depends-on' 来表达对多个bean的依赖。

```
<bean id="beanOne" class="ExampleBean" depends-on="manager, accountDao">
  <property name="manager" ref="manager" />
</bean>
<bean id="manager" class="ManagerBean" />
<bean id="accountDao" class="x.y.jdbc.JdbcAccountDao" />
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>
咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

配置参数说明-12

n 延迟初始化bean

1: ApplicationContext实现的默认行为就是在启动时将所有singleton bean提前进行实例化，这样可能会增大资源的消耗，但会加快程序的运行速度。

2: 可以将bean设置为延迟实例化。在XML配置文件中，延迟初始化将通过<bean/>元素中的lazy-init属性来进行控制。例如：

```
<bean id="lazy" class="com.foo.ExpensiveToCreateBean" lazy-init="true">
</bean>
```

3: 在容器层次中通过在<beans/>元素上使用' default-lazy-init' 属性来控制延迟初始化也是可能的。如下面的配置：

```
<beans default-lazy-init="true">
</beans>
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

自动装配-1

n Spring IoC容器可以自动装配（autowire）相互协作bean之间的关联关系。

模式	说明
no	不使用自动装配。必须通过ref元素指定依赖，这是默认设置。由于显式指定协作者可以使配置更灵活、更清晰，因此对于较大的部署配置，推荐采用该设置。而且在某种程度上，它也是系统架构的一种文档形式。
byName	根据属性名自动装配。此选项将检查容器并根据名字查找与属性完全一致的bean，并将其与属性自动装配。例如，在bean定义中将autowire设置为by name，而该bean包含master属性（同时提供setMaster(..)方法），Spring就会查找名为master的bean定义，并用它来装配给master属性。
byType	如果容器中存在一个与指定属性类型相同的bean，那么将与该属性自动装配。如果存在多个该类型的bean，那么将会抛出异常，并指出不能使用byType方式进行自动装配。若没有找到相匹配的bean，则什么事都不发生，属性也不会被设置。如果你不希望这样，那么可以通过设置dependency-check="objects"让Spring抛出异常。
constructor	与byType的方式类似，不同之处在于它应用于构造器参数。如果在容器中没有找到与构造器参数类型一致的bean，那么将会抛出异常。
autodetect	通过bean类的自省机制（introspection）来决定是使用constructor还是byType方式进行自动装配。如果发现默认的构造器，那么将使用byType方式。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

自动装配-2

n 自动装配的优缺点

1: 优点:

(1) 自动装配能显著减少配置的数量。

(2) 自动装配可以使配置与java代码同步更新。例如，如果你需要给一个java类增加一个依赖，那么该依赖将被自动实现而不需要修改配置。

2: 缺点:

(1) 尽管自动装配比显式装配更神奇，但是，正如上面所提到的，Spring会尽量避免在装配不明确的时候进行猜测，因为装配不明确可能出现难以预料的结果，而且Spring所管理的对象之间的关联关系也不再能清晰的进行文档化。

(2) 对于那些根据Spring配置文件生成文档的工具来说，自动装配将会使这些工具没法生成依赖信息。

(3) 当根据类型进行自动装配的时候，容器中可能存在多个bean定义跟自动装配的setter方法和构造器参数类型匹配。虽然对于数组、集合以及Map，不存在这个问题，但是对于单值依赖来说，就会存在模棱两可的问题。如果bean定义不唯一，装配时就会抛出异常

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

自动装配-3

n 将bean排除在自动装配之外

- 1: <bean/>元素的 `autowire-candidate` 属性可被设为 `false`，这样容器在查找自动装配对象时将不考虑该bean。
- 2: 另一个做法就是使用对bean名字进行模式匹配来对自动装配进行限制。其做法是在<beans/>元素的 '`default-autowire-candidates`' 属性中进行设置。

比如，将自动装配限制在名字以 '`Repository`' 结尾的bean，那么可以设置为 "`*Repository`"。对于多个匹配模式则可以使用逗号进行分隔。注意，如果在bean定义中的 '`autowire-candidate`' 属性显式的设置为 '`true`' 或 '`false`'，那么该容器在自动装配的时候优先采用该属性的设置，而模式匹配将不起作用。

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

依赖检查

- n Spring除了能对容器中bean的依赖设置进行检查外。还可以检查bean定义中实际属性值的设置，当然也包括采用自动装配方式设置属性值的检查。
- n 有几种不同的使用模式，在xml配置文件中，可以在bean定义中为dependency-check属性使用以下几种值

模式	说明
none	没有依赖检查，如果bean的属性没有值的话可以不用设置。
simple	对于原始类型及集合（除协作者外的一切东西）执行依赖检查
object	仅对协作者执行依赖检查
all	对协作者，原始类型及集合执行依赖检查

- n 缺省为none

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Bean的作用域-1

n singleton

在每个Spring IoC容器中一个bean定义对应一个对象实例，在读取配置文件创建IoC容器的时候就会根据配置初始化singleton的Bean实例

n prototype

一个bean定义对应多个对象实例

n request

在一次HTTP请求中，一个bean定义对应一个实例；即每次HTTP请求将会有各自的bean实例，它们依据某个bean定义创建而成。该作用域仅在基于web的Spring ApplicationContext情形下有效。

n session

在一个HTTP Session中，一个bean定义对应一个实例。该作用域仅在基于web的Spring ApplicationContext情形下有效。

n global session

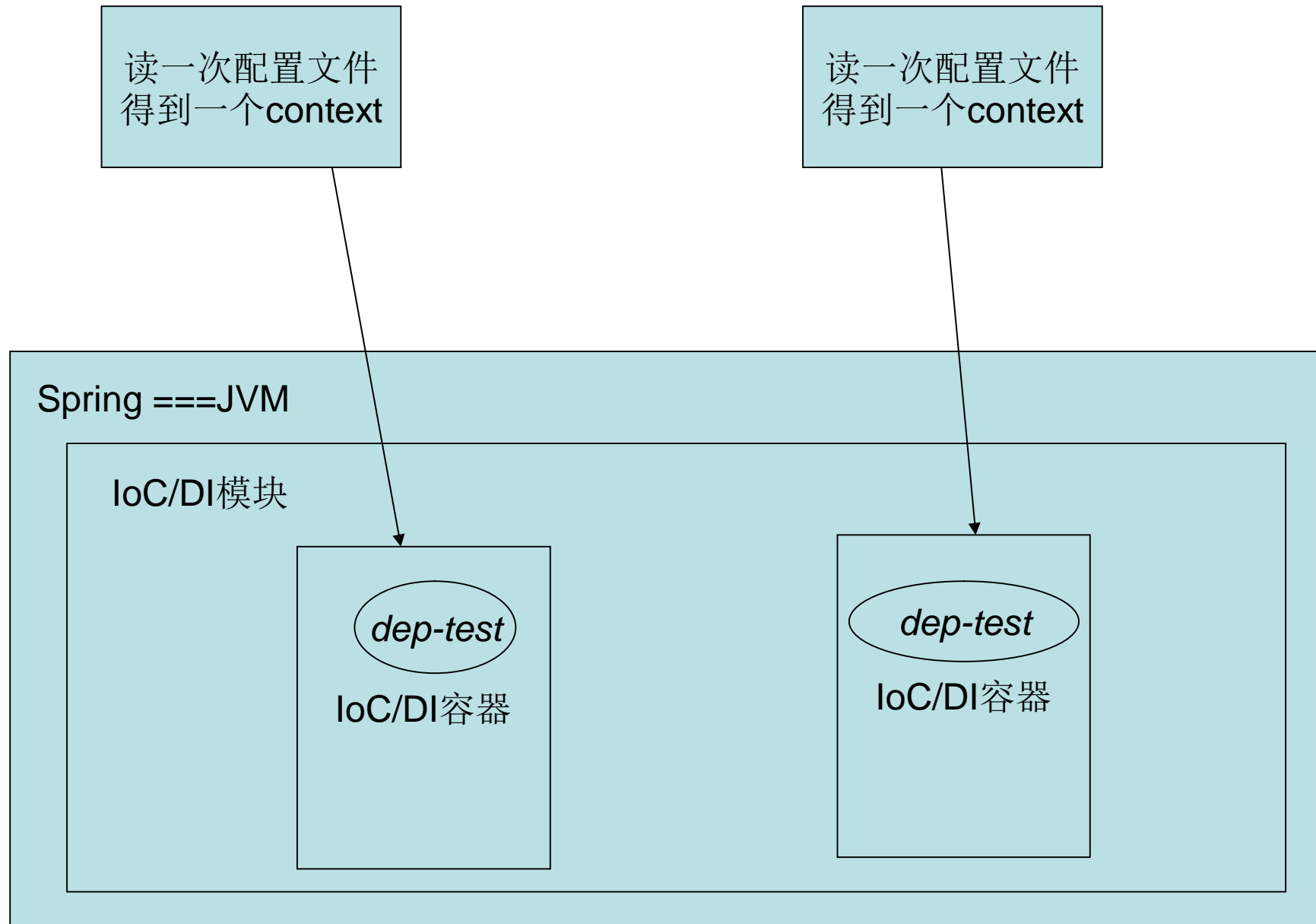
在一个全局的HTTP Session中，一个bean定义对应一个实例。典型情况下，仅在使用portlet context的时候有效。该作用域仅在基于web的Spring ApplicationContext情形下有效。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送





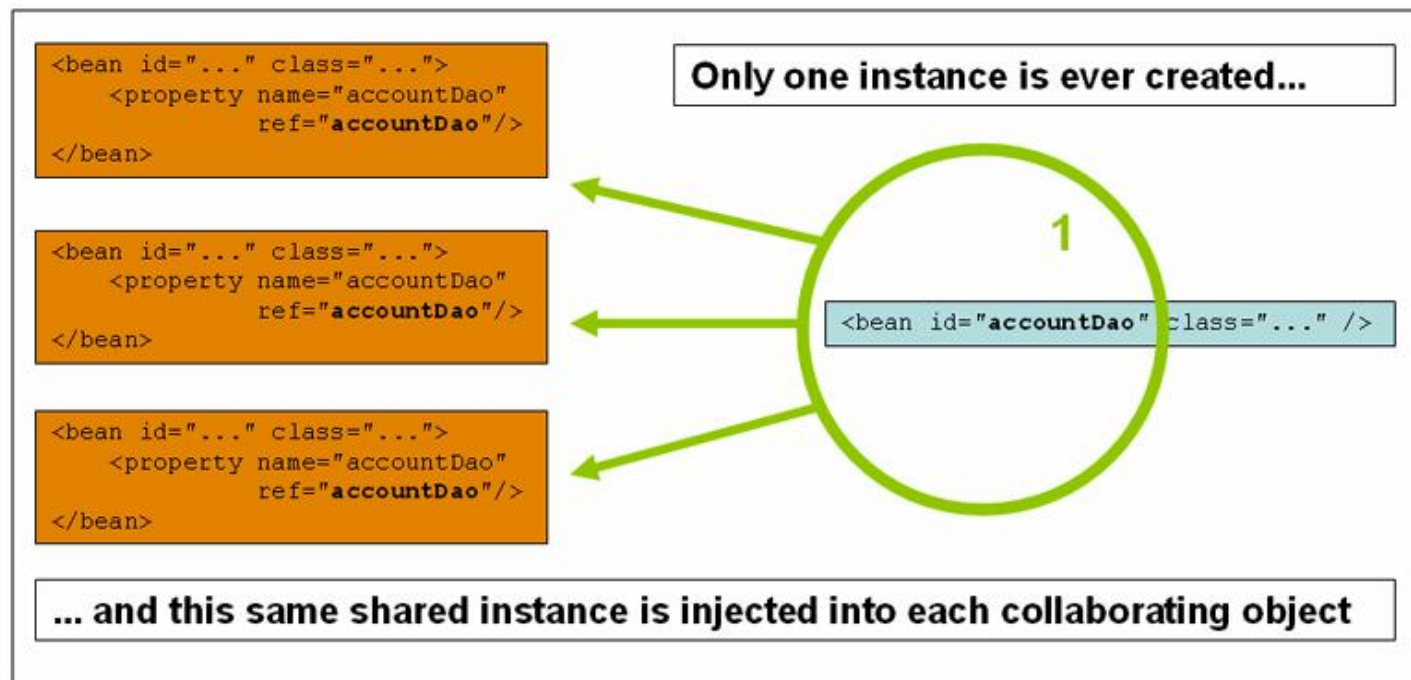
《深入浅出学Spring3开发》——系列精品教程

Bean的作用域-2

n Singleton作用域

一个bean定义设置为singleton时，Spring IoC容器只会创建该bean定义的唯一实例。

- n 注意Spring的singleton概念与GoF模式一书中定义的Singleton模式是完全不同的。经典的GoF Singleton模式的对象范围是指在每一个ClassLoader中指定class创建的实例有且仅有一个。而Spring的singleton作用域应该描述成一个container对应一个bean实例最为贴切



真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

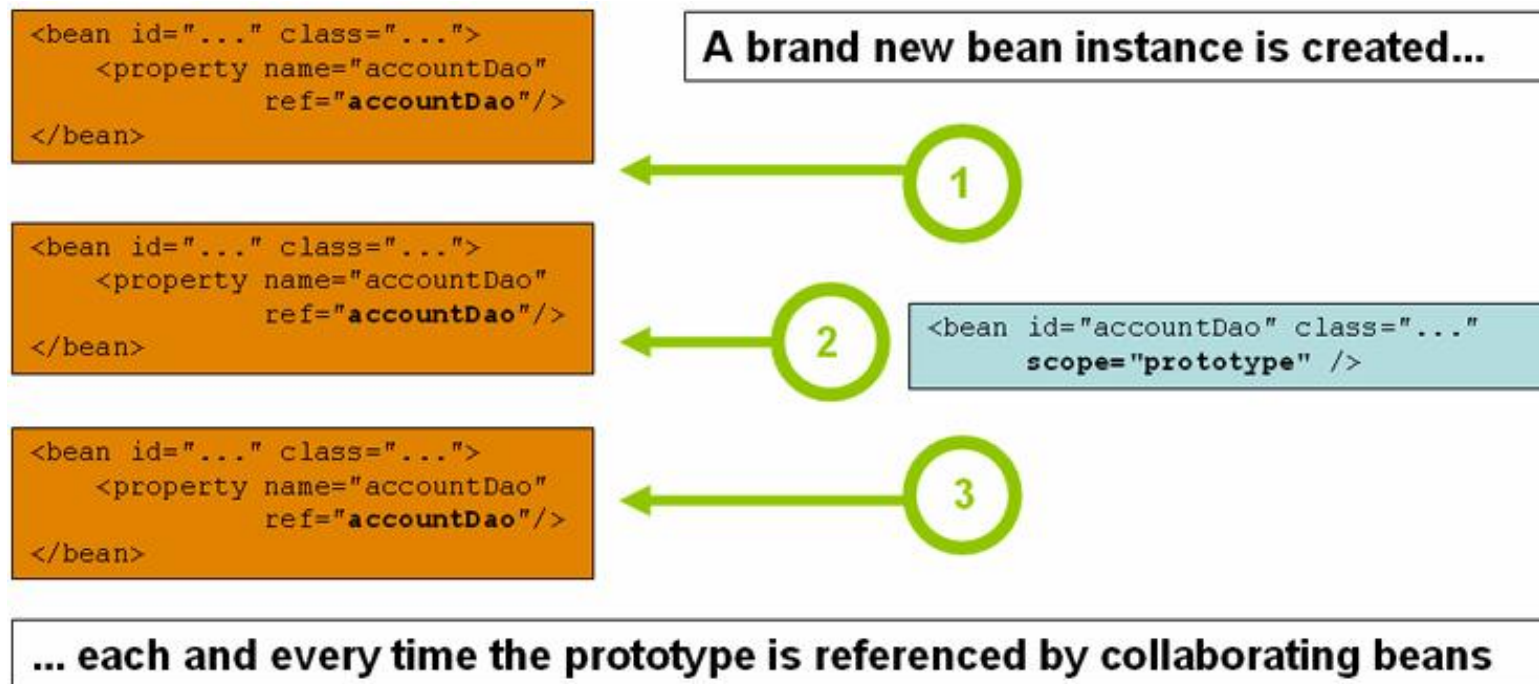


《深入浅出学Spring3开发》——系列精品教程

Bean的作用域-3

n Prototype作用域

Prototype作用域的bean会导致在每次对该bean请求（将其注入到另一个bean中，或者以程序的方式调用容器的getBean()方法）时都会创建一个新的bean实例。根据经验，对有状态的bean应该使用prototype作用域，而对无状态的bean则应该使用singleton作用域。



真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Bean的作用域-4

n request、session以及global session仅在基于web的应用中使用

n 初始化web配置，Servlet 2.4及以上的web容器，如下配置：

```
<web-app>
  <listener>
    <listener-class>org.springframework.web.context.request.RequestContextListener</listener-
    class>
  </listener>
</web-app>
```

n request作用域说明

等同于Http的Request

n session作用域说明

针对某个HTTP Session，Spring容器会根据bean定义创建一个全新的bean实例，且该bean仅在当前HTTP Session内有效。与request作用域一样，你可以根据需要放心的更改所创建实例的内部状态，而别的HTTP Session中创建的实例，将不会看到这些特定于某个HTTP Session的状态变化。当HTTP Session最终被废弃的时候，在该HTTP Session作用域内的bean也会被废弃掉

n global session

等同于标准的HTTP Session

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程小结

n IoC/DI开发

包括：基本配置、自动装配、依赖检查、Bean的作用域。

n 作业：

- 1：复习和掌握这些理论知识。一定要动手去配置试验一下，会有助于加深理解和掌握。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程概览

n IoC/DI开发

包括：方法注入、生命周期以及ApplicationContext的使用。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！



《深入浅出学Spring3开发》——系列精品教程

第二章：IoC/DI开发

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

方法注入-1

- n 问题：比如在调用一个singleton类型bean A的某个方法时，需要引用另一个非singleton (prototype) 类型的bean B，对于bean A来说，容器只会创建一次，这样就没法在需要的时候每次让容器为bean A提供一个新的bean B实例
- n 解决方案：Lookup方法注入，示例如下：
- n Java类

```
public abstract class HelloImpl implements HelloApi{  
    private T2 t2;  
    public String helloSpring3(int a){  
        getT2().t1();  
        System.out.println("hello Spring3==="+a);  
        return "Ok, a="+a;  
    }  
    public abstract T2 getT2();  
}
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>
咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

方法注入-2

```
public class T2 {  
    public void t1(){  
        System.out.println("now in t1");  
    }  
}
```

n 配置文件:

```
<bean id="helloBean" class="cn.javass.Spring3.hello.HelloImpl">  
    <lookup-method name="getT2" bean="t2"/>  
</bean>  
<bean id="t2" class="cn.javass.Spring3.hello.T2"></bean>
```

n Lookup方法注入的内部机制是Spring利用了CGLIB库在运行时生成二进制代码功能，通过动态创建Lookup方法bean的子类而达到复写Lookup方法的目的

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>
咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

Lifecycle

n 初始化回调

有两种方法，如下：

- 1: 实现org.springframework.beans.factory.InitializingBean接口，这种方法不被推荐，因为这样和Spring耦合起来了。可以采用声明式的方法，如下：
- 2: 在Bean定义中指定一个普通的初始化方法，即在XML配置文件中通过指定init-method属性来完成，配置如下：

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

n 析构回调

也有两种方法，如下：

- 1: 实现org.springframework.beans.factory.DisposableBean接口，这种方法不被推荐，因为这样和Spring耦合起来了。可以采用声明式的方法，如下：

```
<bean id="initBean" class="ex.ExampleBean" destroy-method="cleanup"/>
```

n 缺省的初始化和析构方法

配置在beans上，这样就不用每个Bean都配了，如下：

```
<beans default-init-method="init">
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-国际化-1

n context包的核心是ApplicationContext接口。它由BeanFactory接口派生而来，除了提供了BeanFactory所有的功能，还提供了以下的功能：

- 1: MessageSource，提供国际化的消息访问
- 2: 资源访问，如URL和文件
- 3: 事件传播，实现了ApplicationListener接口的bean
- 4: 载入多个（有继承关系）上下文，使得每一个上下文都专注于一个特定的层次，比如应用的web层

n 利用MessageSource实现国际化

- 1: 在配置文件中添加

```
<bean id="messageSource"
      class="org.springframework.context.support.ResourceBundleMessageSource">
  <property name="basenames">
    <list>
      <value>format</value>
      <value>exceptions</value>
    </list>
  </property>
</bean>
```

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-国际化-2

2: 说明: 上述配置是表示在classpath下有两个标准的properties文件。文件格式是标准的properties格式, Spring通过ResourceBundle, 使用JDK中解析消息的标准方式, 来处理任何解析消息的请求。

3: 测试代码:

```
String msg = context.getMessage("testmsg", null, "Default", Locale.CHINA);
```

n 前面getMessage方法的第二个参数是用来从程序中向消息里面传值的, 如下:

消息文件: testmsg=this is a test, {0}, {1}

```
Java类: String msg = context.getMessage("testmsg",  
    new Object[]{"M1", "M2"}, "Default", Locale.CHINA);  
System.out.println("msg="+msg);
```

n 前面getMessage方法的第四个参数是用来指定Locale的

对于国际化 (i18n), Spring中不同的MessageResource实现与JDK标准ResourceBundle中的locale解析规则一样。比如在上面例子中定义的消息Source bean, 如果你想解析British (en-GB) locale的消息, 那么需要创建format_en_GB.properties的资源文件; 中国的如msg_zh_CN.properties。Locale解析通常由应用程序根据运行环境来指定。

真正高质量培训 签订就业协议

网 址: <http://www.javass.cn>

咨询QQ: 460190900

本资料由 私塾在线 <http://sishuok.com> 独家提供, 更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-国际化-3

n 也可以把MessageSource当作资源注入到Bean中，Java类示例如下：

```
public class HelloImpl implements HelloApi{
    private MessageSource ms = null;
    public void setMs(MessageSource ms){
        this.ms = ms;
    }
    public String helloSpring3(int a){
        String msg = this.ms.getMessage("testmsg", null, "Default", Locale.CHINA);
        System.out.println("hello Spring3==="+msg);
        return "Ok, a="+a;
    }
}
```

配置文件示例如下：

```
<bean id="helloBean" class="cn.javass.Spring3.hello.HelloImpl">
    <property name="ms" ref="messageSource"></property>
</bean>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-事件-1

n 事件传播

ApplicationContext中的事件处理是通过ApplicationEvent类和ApplicationListener接口来提供的。如果在上下文中部署一个实现了ApplicationListener接口的bean，那么每当一个ApplicationEvent发布到ApplicationContext时，这个bean就得到通知。Spring提供了三个标准事件，如下：

n ContextRefreshedEvent

当ApplicationContext初始化或刷新时发送的事件。这里的初始化意味着：所有的bean被装载，singleton被预实例化，以及ApplicationContext已就绪可用

n ContextClosedEvent

当使用ApplicationContext的close()方法结束上下文时发送的事件。这里的结束意味着：singleton bean 被销毁

n RequestHandledEvent

一个与web相关的事件，告诉所有的bean一个HTTP请求已经被响应了（也就是在一个请求结束后会发送该事件）。注意，只有在Spring中使用了DispatcherServlet的web应用才能使用

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-事件-2

n 示例，Java类：

```
public class T2 implements ApplicationListener{  
    @Override  
    public void onApplicationEvent(ApplicationEvent arg0) {  
        System.out.println("事件发生了==" + arg0);  
    }  
}
```

n 配置文件

```
<bean id="t2" class="cn.javass.Spring3.hello.T2"></bean>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-访问资源-1

n Resource 接口：

Spring的 Resource 接口是为了提供更强的访问底层资源能力的抽象，典型的是访问文件资源。基本的定义如下：

```
public interface Resource extends InputStreamSource {
    boolean exists();
    boolean isOpen();
    URL getURL() throws IOException;
    File getFile() throws IOException;
    Resource createRelative(String relativePath) throws IOException;
    String getFilename();
    String getDescription();
}

public interface InputStreamSource {
    InputStream getInputStream() throws IOException;
}
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-访问资源-2

n 可以使用ApplicationContext直接访问资源，示例如下：

```
InputStream in =  
    context.getResource("msg_en_GB.properties").getInputStream();  
byte bs[] = new byte[100];  
in.read(bs);  
System.out.println("file content==" + new String(bs));
```

n 也可以向Bean里面注入资源，示例如下：

在Java类当中添加：

```
private Resource rs = null;  
public void setRs(Resource rs) {  
    this.rs = rs;  
}
```

在配置文件中：

```
<bean id="helloBean" class="cn.javass.Spring3.hello.HelloImpl">  
    <property name="rs" value="msg_en_GB.properties"></property>  
</bean>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

ApplicationContext-在Web应用中的实例化

n ApplicationContext能以声明的方式创建，在web.xml中配置如下：

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/daoContext.xml /WEB-
    INF/applicationContext.xml</param-value>
</context-param>
<listener>
  <listener-
    class>org.springframework.web.context.ContextLoaderListener</listener-
    class>
</listener>
```

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程小结

n IoC/DI开发

包括：方法注入、生命周期以及ApplicationContext的使用。

n 作业：

- 1：复习和掌握这些理论知识。一定要动手去配置试验一下，会有助于加深理解和掌握。
- 2：从前面的项目练习中，找一个模块，修改成使用IoC/DI的方式。

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送



《深入浅出学Spring3开发》——系列精品教程

本节课程概览

n AOP入门

包括：是什么、能干什么、AOP基本思想的演变

真正高质量培训 签订就业协议

网 址：<http://www.javass.cn>

咨询QQ：460190900

本资料由 私塾在线 <http://si.shuok.com> 独家提供，更有大量免费在线学习视频独家大放送