Spring介绍

# 介绍

Spring是一个轻量级的，用于简化开发复杂的Java企业级应用的框架。Spring的目标就是简化Java开发，所以它提供了应用程序基础方面的支持。其目的就是让开发人员能够专注于应用业务逻辑层面的开发，而不是为了写大量冗余的模板代码（如jdbc）浪费大量时间。而且，Spring是模块化的，你可以在任何的Java应用中使用自己需要的模块，所以大部分Java应用开发都能从Spring中获益。

Spring的提供4个关键的策略：

* 使用轻量级和最小化侵入的POJO来开发
* 使用DI（依赖注入）和面向接口编程来解耦
* 通过切面和共用的约定来实现声明式编程
* 通过切面和模板消除样板文件

# DI (IOC) 和 AOP

依赖注入（Dependency Injection）和面向切面编程（Aspect-oriented Programming）是Spring的核心技术，整个Spring大家庭都依赖于这两种编程技术（特别是依赖注入）。

## 依赖注入（也称控制反转）

依赖注入（也叫控制反转 Inversion of Control）是一种设计模式，它让对象不自己创建和管理依赖，而是通过注入的方式。依赖就是一个可被使用的对象（一个service），注入即把依赖传入依赖于该服务的对象（一个client）的动作。依赖注入是对象变得易于管理和测试，实现了代码间的解耦，达到松耦合的目的。

## 面向切面编程

面向切面编程就是通过切面把系统中的关注点模块化的技术。关注点就是系统中跨多个模块的功能点（比如事务管理、日志记录等）独立出来模块化，从而使业务逻辑各部分分离，达到松耦合，实现代码的可重用性，提高开发效率。

# Spring模块

Spring是由很多模块组成的，Spring 4就有20个不同的模块，可以被分为以下几组。



Spring核心容器：核心容器是Spring最基础的部分，管理Beans如何创建、配置和管理。主要由BeanFactory构成，提供依赖注入功能。BeanFactory是工厂模式的实现，它把配置文件，依赖性规范和实际的业务逻辑代码分开，实现解耦。建立在BeanFactory之上，Spring提供多种上下文的实现方式，每种方式都提供不同的配置Spring的方法。除了BeanFactory和应用上下文，核心容器还包括许多企业服务，例如email，JNDI查找，EJB集成和任务调度（scheduling）。

Spring AOP 模块：Spring提供丰富的面向切面编程的支持在AOP模块中，在开发你自己的切面时作为基础存在。使用AOP，你可以把系统层面的关注点（比如事务控制、安全和日志等）从业务代码中提取出来，从而实现代码的解耦，模块的分离。

数据存取和集成：Spring JDBC和DAO模块让你从大量的样板文件（比如建立连接，创建Statement，处理ResultSet）编写中解脱出来，使代码保持整洁和简单。该模块在各数据库提供商的异常错误信息之上提供了有意义的异常处理层，再也不需要去破译含糊和专有的SQL错误信息了。 建立在Spring DAO模块之上，Spring ORM模块提供了方便的方式在主流ORM框架上建立DAO层，比如Hibernate、Java Persistence API（JPA）、Java Data Object和iBATIS SQL Maps。这些ORM框架和JDBC都支持Spring通用事务管理。 该模块包括Java Message Service（JMS）和object-to-XML功能。

Web：MVC（Model-View-Controller）模式是一种广为接受的用来建立用户接口与业务逻辑分离的web应用的方式。Spring MVC框架提供主要模型和web表单之间清晰的分离，并且可与其它spring框架提供的功能很好集成。

除了提供spring mvc功能，该模块还包括基本的web方面集成的功能，例如多文件上传，使用servlet监听器初始化IoC容器和一个web相关的应用上下文。而且，它也包括Http client功能和spring远程支持web相关的部分。

而且Spring也是第一梯队支持Restful API的框架。

测试：意识到测试在应用开发中的重要性，spring提供了模块专门用于测试Spring应用。Spring Test模块支持使用JUnit和TestNG对Spring组件进行单元测试和集成测试。该模块提供spring应用上下文的加载和缓存，所以可在测试中使用上下文中的beans。以便你隔离的测试代码，该模块还提供mock（虚假）对象来实现该功能。