

# Spring Web Development

Build web application using jdbc and webmvc

zzy.hit@gmail.com

May 27, 2013

- Spring Jdbc
- Listener/Servlet/Filter
- Spring WebMVC

## IoC - Inversion of Control

## 定律一

高层不应依赖低层，模块都必须抽象

## 定律一

高层不应依赖低层，模块都必须抽象

now, save to database?

## 定律一

高层不应依赖低层，模块都必须抽象

now, save to database?

- 高层逻辑通常是抽象业务，应该具有可重用性
- 不应依赖底层某个模块实现

## 定律二

实现必须依赖抽象，而不是抽象依赖实现

## 定律二

实现必须依赖抽象，而不是抽象依赖实现

now, save to database?



# IoC (Inversion of Control)

# IoC (Inversion of Control)

FileWriter? DatabaseWriter? NetWriter? CompositeWriter...

## 定律三

应用不应依赖容器，容器应服务于应用

- 业务逻辑不做修改可迁移
- 容器不侵入应用

## The Dependency Inversion Principle - Uncle Bob

# 举个例子



Spring的核心是一个轻量级的IoC (Inversion of Control) 容器。用它构建的应用可以达到模块间低耦合，可测试，最终使得整个系统结构简化，易于维护。

- 轻量级(Lightweight)  
体积小(核心不足1M)、资源少、无侵入(Nonintrusive)
- 容器(Container)  
管理组件生命周期、状态、依赖关系等
- IoC (Inversion of Control)  
通过配置维护组件依赖关系，无需编码

除此之外，Spring的目标是实现一个全方位的整合框架，其下由多个子框架组合，子框架之间彼此独立，并可以使用其他框架方案代替。

- AOP框架

Spring神器之一(Asspect-Oriented Programming)

- 持久层

如JDBC、ORM(Hibernate、iBatis)、事务处理等

- Web框架

Spring提供了自己的Web框架，同时你也可以用其他框架来代替，如Struts、Webwork等



# Spring Framework



- id/name, class,
- constructor-arg, properties, value/ref/value-ref
- abstract/parent

XmlBeanFactory

- init-method - InitializingBean
- destroy-method - DisposableBean
- lazy-init

factory-bean, factory-method



`scope(singleton/prototype/request/session/global),`



# Annotation

- what is annotation
- @Service, @Resource(Autowired)

# Configuration (xml vs annotation)

# AOP - Aspect Oriented Programming

DBCodeBeanListLoader vs MappedCodeBeanListLoader

-EOF-