

Spring 概述
 Spring 简史
 Spring 概述
Spring 项目快速搭建
Spring 基础配置
 基础原则
 依赖注入
 java 配置
 AOP

Spring 概述

Spring 简史

- spring 1.x 时代, xml 配置。
- spring 2.x 时代, 注解配置。
- spring 3.x 时代, Java 配置。

Spring 概述

1. spring 是模块化的:
 - 核心容器
 - AOP
 - Messaging
 - Web
 - Data Access / integration
2. spring 生态: 根据自己项目的需要可以选择使用相应的项目。

Spring 项目快速搭建

- 使用 Maven 来管理项目的依赖、编译、文档等信息。

Spring 基础配置

基础原则

- 使用 POJO 进行轻量级和最小侵入式开发。
- 通过依赖注入和基于接口编程实现松耦合。
- 通过 AOP 和默认习惯实现声明式编程。
- 使用 AOP 和 template 减少模式化代码。

依赖注入

- 依赖注入是指用**容器负责创建对象和维护对象间的依赖关系**, 而不是对象本身。
- 依赖注入的目的是**解耦**
- 声明 Bean 的注解
 - @Component 没有明确角色。
 - @Service 在业务逻辑层 (service 层) 使用。

- @Repository 在数据访问层使用。
 - @Controller 在展现层使用。
- 注入 Bean 的注解
 - @Autowired: spring 提供的注解。
 - @Inject: JSR-330 提供的注解。
 - @Resource: JSR-250 提供的注解。
- 示例见包 highlight_spring4.ch1.di
 - 使用 @Service 声明当前类是 spring 管理的一个 bean。
 - 使用 @Autowired 将实体 bean 注入当前某个类。
 - 用 @Configuration 声明当前类是配置类。
 - 用 @ComponentScan("包名") 将包下所有声明为 bean 的注册为 bean。

java 配置

- @Configuration 声明当前类是配置类。
- 用 @Bean 注解在方法上，声明当前方法返回一个 Bean。
- 全局配置使用Java配置（如数据库相关配置、MVC相关配置）。
- 业务Bean的配置使用注解配置（@Service、@Component、@Repository、@Controller）。
- 示例见包 highlight_spring4.ch1.javaconfig

AOP

- 目的：解耦。
- 此处不理解，待补充。
- 示例见包 highlight_spring4.ch1.aop