

Spring

七大模块

- Core { IoC Container, Events, Resources, i18n, Validation, Data Binding, Type Conversion, SpEL, AOP.
- Testing { Mock Objects, TestContext Framework, Spring MVC Test, WebTestClient.
- Data Access { Transactions, DAO Support, JDBC, R2DBC, O/R Mapping, XML Marshalling.
- Web Servlet { Spring MVC, WebSocket, SockJS, STOMP Messaging.
- Web Reactive { Spring WebFlux, WebClient, WebSocket
- Integration { Remoting, JMS, JCA, JMX, Email, Tasks, Scheduling, Caching.
- Languages { Kotlin, Groovy, Dynamic Languages.

常用模块

Bean加载过程

- 实例化
- 属性赋值
- 初始化
 - 检查Aware相关接口并设置相关依赖
 - BeanPostProcessor前置处理
 - 若实现InitiazingBean接口，则调用afterPropertiesSet方法
 - 若配置自定义init-method方法则执行
 - BeanPostProcessor后置处理
- 销毁注册回调接口
 - 注册Destruction相关回调接口
 - 是否实现DisposableBean接口
 - 是否实现自定义destroy-method

AOP

- 接口类型 { 默认使用JDK Proxy
- 非接口类型 { 默认使用CGLIB

SpringBoot

- 两大核心 {
 - 自动化配置 { 简化配置核心基于 Configuration, EnableXX, Condition
 - spring-boot-starter { 脚手架核心整合各种第三方类库, 协同工具
- 自动化配置注解
 - 核心注解 { @SpringBootApplication
 - SpringBoot 应用标注在某个类上说明这个类是 SpringBoot 的主配置类, SpringBoot 就会运行这个类的 main 方法来启动 SpringBoot 项目。
 - @SpringBootConfiguration
 - @EnableAutoConfiguration
 - @AutoConfigurationPackage
 - @Import({AutoConfigurationImportSelector.class})
 - 加载所有 META-INF/spring.factories 中存在的配置类 (类似 SpringMVC 中加载所有 converter)
 - 条件化配置 {
 - @ConditionalOnBean
 - @ConditionalOnClass
 - @ConditionalOnMissingBean
 - @ConditionalOnProperty
 - @ConditionalOnResource
 - @ConditionalOnSingleCandidate
 - @ConditionalOnWebApplication
- Starter {
 - spring.factories
 - additional-metadata
 - 自定义 Configuration 类

JDBC

- 定义了数据库交互接口 {
 - DriverManager
 - Connection
 - Statement
 - ResultSet

Mybatis/Hibernate

- Mybatis 优点: 原生 SQL (XML 语法), 直观, 对 DBA 友好。
- Hibernate 优点: 简单场景不用写 SQL (HQL、Cretiria、SQL) 。
- Mybatis 缺点: 繁琐, 可以用 MyBatis-generator、MyBatis-Plus 之类的插件。
- Hibernate 缺点: 对 DBA 不友好。

JPA

- JPA 的全称是 Java Persistence API, 即 Java 持久化 API, 是一套基于 ORM 的规范, 内部是由一系列的接口和抽象类构成。JPA 通过 JDK 5.0 注解描述对象-关系表映射关系, 并将运行期的实体对象持久化到数据库中。核心 EntityManager

Spring事务管理

- 编程式事务管理
- 声明式事务管理: 事务管理器+AOP