上云-Spring Framework

- day01-Spring-Xml配置
  - Spring简介
    - Spring 各版本文档地址
    - Modules 模块介绍
  - IOC
    - bean 的配置
      - id/class属性
      - name属性
    - 容器创建
      - ClassPathXmlApplicationContext
      - FileSystemXmlApplicationContext
    - 容器获取bean
      - 根据id获取
      - 根据id和类型
      - 根据类型获取
    - 设计模式
      - 单例模式
        - 懒汉式
        - 饿汉式
        - 静态工厂
      - 工厂模式
        - 实例工厂
    - bean 实例化
      - 构造方法
      - 静态工厂
      - 实例工厂
      - 实现 FactoryBean
  - DI
    - DI依赖注入
      - 构造器注入
      - setter注入
      - Collections
    - 自动装配
    - bean scopes
      - singleton scope
      - prototype scope
      - Request scope
      - Session scope
    - bean 生命周期
      - Initialization callbacks
      - Destruction callbacks
      - BeanPostProcessor
  - Container Extension Points
    - Xml 配置数据源对象Druid
      - 硬编码
      - 占位符软编码
        - 加载properties
      - 注意事项
    - 容器层次结构
    - 容器扩展点
- day02-Spring-注解配置
  - 配置 bean 扫描
  - IOC
    - 注解配置 bean
      - 使Component
        - @Controller
        - @Service
        - @Repository
    - Xml 配置注解包扫描
    - 配置类
      - @Configuration
      - 包扫描注解@ComponentScan
    - bean scopes
      - @Scope
    - bean 生命周期
      - @PostConstruct
      - @PreDestroy
  - DI
    - 注解实现按照类型注入
      - @Autowired
    - 注解实现按照名称注入
      - @Autowired
      - @Qualifier
    - 简单数据类型注入
      - @Value
    - 注解读取properties配置文件
      - 注解加载properties配置文件
        - @PropertySource
      - @Value("${name}")
    - 注解配置第三方bean Druid
      - @Bean
      - @Import
  - lombok
    - 添加插件
    - 常用注解
      - @Getter @Setter
      - @NoArgsConstructor
      - @AllArgsConstructor
      - @ToString
      - @NotNull
      - @RequiredArgsConstructor
      - @Data
      - @Slf4j
  - Spring 整合第三方
    - Spring整合Junit
    - Spring整合JdbcTemplate
- day03-Spring-AOP
  - 简介
    - 概述
    - 作用
    - 代理模式
      - 静态代理
      - 动态代理
        - jdk 动态代理
        - cglib 动态代理
  - aop 概念
    - 案例分析
      - 新建类做创造、新、改、查方法
      - 实现事务操作功能
    - 基本概念
      - 连接点(JoinPoint)
      - 切入点(Pointcut)
      - 通知(Advice)
      - 切面(Aspect)
      - 通知类
    - 核心概念
      - 目标对象(Target)
      - 代理(Proxy)
  - 通过注解方式实现AOP
    - 依赖导入
    - @EnableAspectJAutoProxy
    - @Aspect
    - @Pointcut
    - 各种通知
      - @Before
      - @After
      - @AfterReturning
      - @AfterThrowing
      - @Around
      - 各种通知执行顺序
    - 切入点表达式
      - 语法
    - 获取通知相关信息
      - 获取连接点信息
      - 获取目标方法的返回值
      - 获取异常
  - xml 配置AOP
  - aop 事务管理
    - spring 整合 Mybatis
      - 添加依赖
      - 创建单元测试
      - 添加代码生成器
      - 添加 db.properties
      - 创建Mybatis配置类并配置SqlSessionFactory
      - 创建 Service
      - 测试运行
    - Spring事务简介
      - 概述
      - 转账案例需求分析
    - 基于注解事务
      - @EnableTransactionManagement
      - @Transactional
      - 配置事务管理器
      - Spring 事务角色
        - 事务管理员
        - 事务协调员
      - @Transactional 属性
        - readOnly
          - 设置只读
        - timeout
          - 设置超时时间
        - rollbackFor
          - 设置回滚异常
        - rollbackForClassName
          - 设置回滚异常，值是异常的类全名字符串
        - noRollbackFor
          - 设置事务不回滚异常
        - noRollbackForClassName
          - 设置事务不回滚异常，值是异常的类全名字符串
        - isolation
          - 设置事务的隔离级别
        - propagation
    - 基于xml事务