# MiniDao原理笔记

1. 通过在spring容器中的注解注册minidao的接口，当程序启动时自动扫描examples.dao.\*下的所有有关minidao注解的类，由org.jeecgframework.minidao.factory.MiniDaoBeanFactory进行解析，该类需要继承BeanFactoryPostProcessor，实现其抽象方法postProcessBeanFactory，方法中具体实现：
   1. 在该类中循环传入的minidao配置项，解析配置项查出该配置项对应文件夹下的所有class文件并进行加载
   2. 循环这些class文件类，判断是否是以MiniDao为注解方式，如果是则单独加载一个接口的代理类，将spring容器传入改代理类，并将miniDaoHandler作为拦截器传入，将对应的类注入给spring代理工厂管理，也就是说当这些类的方法在被调用的时候，都会去动态加载miniDaoHandler中的invoke方法。
2. 当运行时，调用了被动态代理过的DAO中的方法时，执行miniDaoHandler中的invoke方法.
   1. 判断被执行的method是否是抽象方法，如果是抽象方法，则不执行MiniDao拦截器；
   2. 如果不是，则继续判断该方法是否是hibernate实体维护的方法，如果是执行hibernate方式维护实体。
   3. 如果不是，则获取该方法的自定义注解（Arguments）,进行对应SQL模板的装载，这里需要注意的是，判断装载SQL模板的两种方式，一种是SQL注解方式，一种是Arguments执行对应参数或者对象。
   4. 解析上一步得到的SQL模板，如果上一步获得的SQL模板是null，则说明不是以SQL注解方式装载，那么需要根据 [接口名\_方法名.sql] 的格式获取SQL模板的路径，然后用Freemarker模板引擎解析该SQL模板。
   5. 组装占位符参数；因为SQL模板中是以 [：参数名] 为规则，这里需要将注解传递的参数替换到对应的占位符中
   6. 通过Spring封装好的JDBC方式执行执行SQL，返回结果