



(12)发明专利申请

(10)申请公布号 CN 111259067 A
(43)申请公布日 2020.06.09

(21)申请号 202010071423.8

(22)申请日 2020.01.21

(71)申请人 褚晶晶

地址 026000 内蒙古自治区锡林郭勒盟太
仆寺旗宝昌镇太平西大街1栋2排72号

(72)发明人 褚晶晶

(74)专利代理机构 北京盛凡智荣知识产权代理
有限公司 11616

代理人 陈月婷

(51)Int.Cl.

G06F 16/25(2019.01)

G06F 8/30(2018.01)

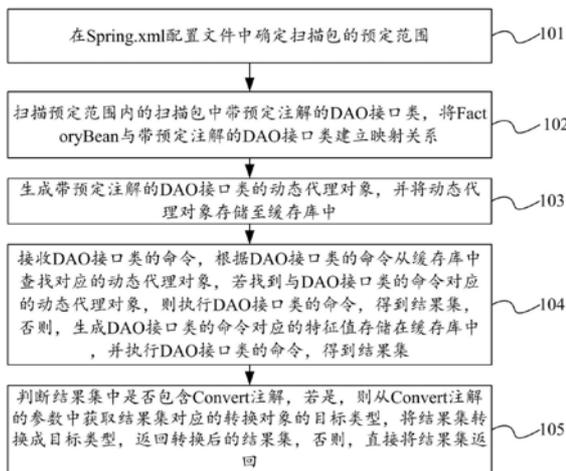
权利要求书2页 说明书12页 附图3页

(54)发明名称

一种基于Spring实现DAO接口的方法、装置及
设备

(57)摘要

本申请属于数据开发领域,公开了一种基于Spring实现DAO接口的方法、装置及设备,通过Spring框架指定扫描包的预定范围,将带预定注解的DAO接口类和FactoryBean建立对应关系,生成相应的动态代理对象存储至缓存库中,若找到该DAO接口类的命令对应的动态代理对象,则执行该命令,若找不到,则根据该命令生成相应的特征值,存储至缓存库中,再执行该命令,最后若该结果集中包含有Convert注解,需要根据Convert注解将结果集转换成相应的目标类型后返回,若该结果集中没有Convert注解,则直接将结果集返回,保证得到的DAO接口能够同时支持SQL和HQL,进行应用构建时,无需将数据整体迁移,不会产生大量的XML配置文件,管理简洁方便,开发人员只需对开发代码进行维护,提高开发效率。



1. 一种基于Spring实现DAO接口的方法,其特征在于,所述方法的步骤包括:

在Spring.xml配置文件中确定扫描包的预定范围;

扫描所述预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean与所述带预定注解的DAO接口类建立映射关系;

生成所述带预定注解的DAO接口类的动态代理对象,并将所述动态代理对象存储至缓存库中;

接收DAO接口类的命令,根据所述DAO接口类的命令从所述缓存库中查找对应的动态代理对象,若找到与所述DAO接口类的命令对应的动态代理对象,则执行所述DAO接口类的命令,得到结果集,否则,生成所述DAO接口类的命令对应的特征值存储在所述缓存库中,并执行所述DAO接口类的命令,得到结果集;

判断所述结果集中是否包含Convert注解,若是,则从Convert注解的参数中获取所述结果集对应的转换对象的目标类型,将所述结果集转换成所述目标类型,返回转换后的结果集,否则,直接将所述结果集返回。

2. 根据权利要求1所述的方法,其特征在于,所述在Spring.xml配置文件中确定扫描包的预定范围,具体包括:

在Spring.xml配置文件中利用BeanDefinitionRegistryPostProcessor确定扫描包的预定范围;

为所述扫描包设定对应的注解名称,其中,设定所述扫描包的初始注解名称为AutoDaoImpl;

为AutoDaoImpl匹配相应的entityClass参数,利用entityClass参数来对应所述带指定注解的DAO接口类的具体操作的实体类对象。

3. 根据权利要求1所述的方法,其特征在于,所述扫描所述预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean与所述带预定注解的DAO接口类建立映射关系,具体包括:

扫描所述预定范围内的扫描包中带预定注解的DAO接口类;

定义ClassPathBeanDefinitionScanner生成BeanDefinition,在BeanDefinition中定义FactoryBean与带预定注解的DAO接口类的映射关系,并为BeanDefinition设置对应的注入模式。

4. 根据权利要求1所述的方法,其特征在于,所述生成所述带预定注解的DAO接口类的动态代理对象,并将所述动态代理对象存储至缓存库中,具体包括:

将所述带预定注解的DAO接口类设置成自定义DAO接口;

利用Proxy对象和所述自定义DAO接口,生成所述自定义DAO接口的动态代理对象,将所述动态代理对象存储至缓存库中。

5. 根据权利要求1所述的方法,其特征在于,所述生成所述DAO接口类的命令对应的特征值存储在所述缓存库中,具体包括:

从所述DAO接口类的命令中获取执行名称,对所述执行名称进行解析,得到对应的返回值;

根据所述执行名称确定对应的执行参数,其中,所述执行参数包括:常用参数和回调函数参数,所述回调函数参数中记录有回调函数参数的索引位置;

根据所述执行名称从数据中查找对应的执行对象；

利用Convert将所述执行对象转换成对应的特定对象类型,其中,所述Convert中设定了各种特征值对应转换的对象类型,以及每种对象类型的转换方案;

根据所述执行名称的命名规则,对所述执行名称进行拆分,得到对应的特征值,将所述特征值存储在所述缓存库中。

6.根据权利要求5所述的方法,其特征在于,所述根据所述执行名称的命名规则,对所述执行名称进行拆分,得到对应的特征值,具体包括:

根据所述命名规则将所述执行名称拆分为前缀、中部、后缀、关键词,将拆分得到的前缀、中部、后缀、关键词作为特征值;

其中,所述前缀包括但不限于下列任一或组合:保存命令、删除命令、修改命令和查询命令;

所述中部为自定义执行业务名称;

所述后缀包括但不限于下列任一或组合:查询所有数据命令、根据ID查询命令、多属性关联查询命令、自定义HQL查询命令、自定义SQL查询命令、自定义SQL逻辑查询命令;

所述关键词为By,用于将前缀、中部和后缀分开。

7.根据权利要求1所述的方法,其特征在于,所述执行所述DAO接口类的命令,得到结果集,具体包括:

从所述缓存库中查找所述DAO接口类的命令对应的目标特征值,根据所述目标特征值确定对应的执行逻辑;

根据所述执行逻辑生成相应的HQL和SQL语句并执行,得到所述结果集。

8.一种基于Spring实现DAO接口的装置,其特征在于,包括:确定模块、扫描模块、生成模块、查找模块和判断模块依次连接;

所述确定模块,用于在Spring.xml配置文件中确定扫描包的预定范围;

所述扫描模块,用于扫描所述预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean与所述带预定注解的DAO接口类建立映射关系;

所述生成模块,用于生成所述带预定注解的DAO接口类的动态代理对象,并将所述动态代理对象存储至缓存库中;

所述查找模块,用于接收DAO接口类的命令,根据所述DAO接口类的命令从所述缓存库中查找对应的动态代理对象,若找到与所述DAO接口类的命令对应的动态代理对象,则执行所述DAO接口类的命令,得到结果集,否则,生成所述DAO接口类的命令对应的特征值存储在所述缓存库中,并执行所述DAO接口类的命令,得到结果集;

所述判断模块,用于判断所述结果集中是否包含Convert注解,若是,则从Convert注解的参数中获取所述结果集对应的转换对象的目标类型,将所述结果集转换成所述目标类型,返回转换后的结果集,否则,直接将所述结果集返回。

9.一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至7中任一项所述的基于Spring实现DAO接口的方法的步骤。

10.一种计算机存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至7中任一项所述的基于Spring实现DAO接口的方法的步骤。

一种基于Spring实现DAO接口的方法、装置及设备

技术领域

[0001] 本申请涉及数据开发技术领域,特别是涉及一种基于Spring实现DAO接口的方法、装置及设备。

背景技术

[0002] DAO模式是标准的J2EE(Java 2Platform Enterprise Edition)设计模式之一,开发人员使用这个模式把底层的数据访问操作和上层的商务逻辑分开,它的全称是DataAccess Object数据访问对象,典型的DAO的组件分为DAO工厂类、DAO接口类、实现DAO接口的具体类、数据传递对象,具体的DAO类包含了从特定的数据源访问数据的逻辑。

[0003] iBATIS是一个基于SQL(Structured Query Language,结构化查询语言)映射支持Java的持久层框架。支持通过DAO接口与XML(Extensible Markup Language,可扩展标记语言)映射文件的绑定,自动生成接口实现类,提高开发效率。

[0004] 但也会带来以下问题:

[0005] (1) iBATIS只支持SQL结构化查询语言。不支持HQL(Hibernate Query Language)Hibernate查询语言。

[0006] (2) 基于在iBATIS上构建的应用,需要整体迁移时,成本较高。

[0007] (3) 需要编写XML映射文件,会造成大量的XML配置文件,配置文件过多导致管理变得困难。

[0008] (4) 开发人员不得不同时维护代码和配置文件,开发效率变得低下。

发明内容

[0009] 有鉴于此,本申请提供了一种基于Spring实现DAO接口的方法、装置及设备。主要目的在于解决目前的系统框架构建应用所需成本较高,附带产生的配置文件较多,影响开发效率的技术问题。

[0010] 基于上述技术问题,本发明的第一方面提出了一种基于Spring实现DAO接口的方法,所述方法的步骤包括:

[0011] 在Spring.xml配置文件中确定扫描包的预定范围;

[0012] 扫描所述预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean与所述带预定注解的DAO接口类建立映射关系;

[0013] 生成所述带预定注解的DAO接口类的动态代理对象,并将所述动态代理对象存储至缓存库中;

[0014] 接收DAO接口类的命令,根据所述DAO接口类的命令从所述缓存库中查找对应的动态代理对象,若找到与所述DAO接口类的命令对应的动态代理对象,则执行所述DAO接口类的命令,得到结果集,否则,生成所述DAO接口类的命令对应的特征值存储在所述缓存库中,并执行所述DAO接口类的命令,得到结果集;

[0015] 判断所述结果集中是否包含Convert注解,若是,则从Convert注解的参数中获取

所述结果集对应的转换对象的目标类型,将所述结果集转换成所述目标类型,返回转换后的结果集,否则,直接将所述结果集返回。

[0016] 进一步地,所述在Spring.xml配置文件中确定扫描包的预定范围,具体包括:

[0017] 在Spring.xml配置文件中利用BeanDefinitionRegistryPostProcessor确定扫描包的预定范围;

[0018] 为所述扫描包设定对应的注解名称,其中,设定所述扫描包的初始注解名称为AutoDaoImpl;

[0019] 为AutoDaoImpl匹配相应的entityClass参数,利用entityClass参数来对应所述带指定注解的DAO接口类的具体操作的实体类对象。

[0020] 进一步地,所述扫描所述预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean与所述带预定注解的DAO接口类建立映射关系,具体包括:

[0021] 扫描所述预定范围内的扫描包中带预定注解的DAO接口类;

[0022] 定义ClassPathBeanDefinitionScanner生成BeanDefinition,在BeanDefinition中定义FactoryBean与带预定注解的DAO接口类的映射关系,并为BeanDefinition设置对应的注入模式。

[0023] 进一步地,所述生成所述带预定注解的DAO接口类的动态代理对象,并将所述动态代理对象存储至缓存库中,具体包括:

[0024] 将所述带预定注解的DAO接口类设置成自定义DAO接口;

[0025] 利用Proxy对象和所述自定义DAO接口,生成所述自定义DAO接口的动态代理对象,将所述动态代理对象存储至缓存库中。

[0026] 进一步地,所述生成所述DAO接口类的命令对应的特征值存储在所述缓存库中,具体包括:

[0027] 从所述DAO接口类的命令中获取执行名称,对所述执行名称进行解析,得到对应的返回值;

[0028] 根据所述执行名称确定对应的执行参数,其中,所述执行参数包括:常用参数和回调函数参数,所述回调函数参数中记录有回调函数参数的索引位置;

[0029] 根据所述执行名称从数据中查找对应的执行对象;

[0030] 利用Convert将所述执行对象转换成对应的特定对象类型,其中,所述Convert中设定了各种特征值对应转换的对象类型,以及每种对象类型的转换方案;

[0031] 根据所述执行名称的命名规则,对所述执行名称进行拆分,得到对应的特征值,将所述特征值存储在所述缓存库中。

[0032] 进一步地,所述根据所述执行名称的命名规则,对所述执行名称进行拆分,得到对应的特征值,具体包括:

[0033] 根据所述命名规则将所述执行名称拆分为前缀、中部、后缀、关键词,将拆分得到的前缀、中部、后缀、关键词作为特征值;

[0034] 其中,所述前缀包括但不限于下列任一或组合:保存命令、删除命令、修改命令和查询命令;

[0035] 所述中部为自定义执行业务名称;

[0036] 所述后缀包括但不限于下列任一或组合:查询所有数据命令、根据ID查询命令、多

属性关联查询命令、自定义HQL查询命令、自定义SQL查询命令、自定义SQL逻辑查询命令；

[0037] 所述关键词为By,用于将前缀、中部和后缀分开。

[0038] 进一步地,所述执行所述DAO接口类的命令,得到结果集,具体包括:

[0039] 从所述缓存库中查找所述DAO接口类的命令对应的目标特征值,根据所述目标特征值确定对应的执行逻辑;

[0040] 根据所述执行逻辑生成相应的HQL和SQL语句并执行,得到所述结果集。

[0041] 本发明的第二方面提出了一种基于Spring实现DAO接口的装置,包括:确定模块、扫描模块、生成模块、查找模块和判断模块依次连接;

[0042] 所述确定模块,用于在Spring.xml配置文件中确定扫描包的预定范围;

[0043] 所述扫描模块,用于扫描所述预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean与所述带预定注解的DAO接口类建立映射关系;

[0044] 所述生成模块,用于生成所述带预定注解的DAO接口类的动态代理对象,并将所述动态代理对象存储至缓存库中;

[0045] 所述查找模块,用于接收DAO接口类的命令,根据所述DAO接口类的命令从所述缓存库中查找对应的动态代理对象,若找到与所述DAO接口类的命令对应的动态代理对象,则执行所述DAO接口类的命令,得到结果集,否则,生成所述DAO接口类的命令对应的特征值存储在所述缓存库中,并执行所述DAO接口类的命令,得到结果集;

[0046] 所述判断模块,用于判断所述结果集中是否包含Convert注解,若是,则从Convert注解的参数中获取所述结果集对应的转换对象的目标类型,将所述结果集转换成所述目标类型,返回转换后的结果集,否则,直接将所述结果集返回。

[0047] 进一步地,所述确定模块具体包括:

[0048] 确定单元,用于在Spring.xml配置文件中利用BeanDefinitionRegistryPostProcessor确定扫描包的预定范围;

[0049] 设定单元,用于为所述扫描包设定对应的注解名称,其中,设定所述扫描包的初始注解名称为AutoDaoImpl;

[0050] 匹配单元,用于为AutoDaoImpl匹配相应的entityClass参数,利用entityClass参数来对应所述带指定注解的DAO接口类的具体操作的实体类对象。

[0051] 进一步地,所述扫描模块具体包括:

[0052] 扫描单元,用于扫描所述预定范围内的扫描包中带预定注解的DAO接口类;

[0053] 定义单元,用于定义ClassPathBeanDefinitionScanner生成BeanDefinition,在BeanDefinition中定义FactoryBean与带预定注解的DAO接口类的映射关系,并为BeanDefinition设置对应的注入模式。

[0054] 进一步地,所述生成模块具体包括:

[0055] 设置单元,用于将所述带预定注解的DAO接口类设置成自定义DAO接口;

[0056] 生成单元,用于利用Proxy对象和所述自定义DAO接口,生成所述自定义DAO接口的动态代理对象,将所述动态代理对象存储至缓存库中。

[0057] 进一步地,所述查找模块具体包括:

[0058] 解析单元,用于从所述DAO接口类的命令中获取执行名称,对所述执行名称进行解析,得到对应的返回值;

[0059] 参数确定单元,用于根据所述执行名称确定对应的执行参数,其中,所述执行参数包括:常用参数和回调函数参数,所述回调函数参数中记录有回调函数参数的索引位置;

[0060] 查找单元,用于根据所述执行名称从数据中查找对应的执行对象;

[0061] 转换单元,用于利用Convert将所述执行对象转换成对应的特定对象类型,其中,所述Convert中设定了各种特征值对应转换的对象类型,以及每种对象类型的转换方案;

[0062] 拆分单元,用于根据所述执行名称的命名规则,对所述执行名称进行拆分,得到对应的特征值,将所述特征值存储在所述缓存库中。

[0063] 进一步地,所述拆分单元具体用于:

[0064] 根据所述命名规则将所述执行名称拆分为前缀、中部、后缀、关键词,将拆分得到的前缀、中部、后缀、关键词作为特征值;

[0065] 其中,所述前缀包括但不限于下列任一或组合:保存命令、删除命令、修改命令和查询命令;

[0066] 所述中部为自定义执行业务名称;

[0067] 所述后缀包括但不限于下列任一或组合:查询所有数据命令、根据ID查询命令、多属性关联查询命令、自定义HQL查询命令、自定义SQL查询命令、自定义SQL逻辑查询命令;

[0068] 所述关键词为By,用于将前缀、中部和后缀分开。

[0069] 进一步地,所述查找单元,还用于从所述缓存库中查找所述DAO接口类的命令对应的目标特征值,根据所述目标特征值确定对应的执行逻辑;

[0070] 所述查找模块具体还包括:

[0071] 语句生成单元,用于根据所述执行逻辑生成相应的HQL和SQL语句并执行,得到所述结果集。

[0072] 依据本申请的第三方面,提供了一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,所述处理器执行所述计算机程序时实现第一方面所述的基于Spring实现DAO接口的方法的步骤。

[0073] 依据本申请的第四方面,提供了一种计算机存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现第一方面所述的基于Spring实现DAO接口的方法的步骤。

[0074] 借由上述技术方案,本申请提供一种基于Spring实现DAO接口的方法、装置及设备,通过Spring框架指定扫描包的预定范围,将带预定注解的DAO接口类和FactoryBean建立对应关系,并生成相应的动态代理对象存储到缓存库中,然后当程序员需要使用时,发送DAO接口类的命令,从缓存库中查找该命令对应的动态代理对象,若找到,则执行该命令,若找不到,则根据该命令生成相应的特征值,存储至缓存库中,再执行该命令,最后若该结果集中包含有Convert注解,需要根据Convert注解将结果集转换成相应的目标类型后返回,若该结果集中没有Convert注解,则直接将该结果集返回。这样,保证得到的DAO接口能够同时支持SQL和HQL,进行应用构建时,无需将数据整体迁移,不会产生大量的XML配置文件,管理简洁方便,开发人员只需对开发代码进行维护,提高开发效率。

[0075] 上述说明仅是本申请技术方案的概述,为了能够更清楚了解本申请的技术手段,而可依照说明书的内容予以实施,并且为了让本申请的上述和其它目的、特征和优点能够更明显易懂,以下特举本申请的具体实施方式。

附图说明

[0076] 通过阅读下文优选实施方式的详细描述,各种其他的优点和益处对于本领域普通技术人员将变得清楚明了。附图仅用于示出优选实施方式的目的,而并不认为是对本申请的限制。而且在整个附图中,用相同的参考符号表示相同的部件。在附图中:

[0077] 图1为本申请的一个实施例中基于Spring实现DAO接口的方法的流程图;

[0078] 图2为本申请的一个实施例的执行时序图;

[0079] 图3为本申请的另一个实施例中基于Spring实现DAO接口的方法的流程图;

[0080] 图4为本申请的再一个实施例中基于Spring实现DAO接口的装置的系统框图;

[0081] 图5为本申请的计算机设备的结构示意图。

具体实施方式

[0082] 下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例,然而应当理解,可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反,提供这些实施例是为了能够更透彻地理解本公开,并且能够将本公开的范围完整的传达给本领域的技术人员。

[0083] 如图1所示,本申请的第一实施例提供了一种基于Spring实现DAO接口的方法,包括如下步骤:

[0084] 步骤101,在Spring.xml配置文件中确定扫描包的预定范围。

[0085] 在该步骤中, Spring.xml配置文件为在Spring系统框架下xml配置文件。根据需要调取Spring系统框架下的xml配置文件,从所有xml配置文件中指定对应需要的扫描包的预定范围,以供根据该预定范围对扫描包进行扫描。

[0086] 步骤102,扫描预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean(工厂bean)与带预定注解的DAO接口类建立映射关系。

[0087] 在该步骤中,由于Spring提供了一个org.springframework.bean.factory.FactoryBean的工厂类接口,可以通过实现该接口定制实例化bean的逻辑,因此,基于该逻辑将FactoryBean与带预定注解的DAO接口类建立映射关系。

[0088] 步骤103,生成带预定注解的DAO接口类的动态代理对象,并将动态代理对象存储至缓存库中。

[0089] 在该步骤中,对得到的带预定注解的DAO接口类进行动态解析,生成相应的动态代理对象,每个动态代理对应与带预定注解的DAO接口类一一对应,然后将这些动态代理对象在缓存库中进行存储,以供后续进行查找。

[0090] 步骤104,接收DAO接口类的命令,根据DAO接口类的命令从缓存库中查找对应的动态代理对象,若找到与DAO接口类的命令对应的动态代理对象,则执行DAO接口类的命令,得到结果集,否则,生成DAO接口类的命令对应的特征值存储在缓存库中,并执行DAO接口类的命令,得到结果集。

[0091] 在该步骤中,开发人员想要使用时,发出DAO接口类的命令,并按照该步骤的上述方案得到执行该命令后的结果集。这样,在执行过程中,不会产生大量的XML配置文件,使得开发人员管理简洁方便。

[0092] 步骤105,判断结果集中是否包含Convert注解,若是,则从Convert注解的参数中

获取结果集对应的转换对象的目标类型,将结果集转换成目标类型,返回转换后的结果集,否则,直接将结果集返回。

[0093] 在该步骤中,结果集中有些需要将其转换成相应对象类型,对象类型即为结果集各个数据的属性,例如,人就是一种类型,每个活生生的人的个体,就是具体的对象。

[0094] 对结果集进行类型转换,这样有助于开发人员根据结果集进行理解,进而有效提高开发的效率。

[0095] 通过上述方案,通过Spring框架指定扫描包的预定范围,将带预定注解的DAO接口类和FactoryBean建立对应关系,并生成相应的动态代理对象存储到缓存库中,然后当程序员需要使用时,发送DAO接口类的命令,从缓存库中查找该命令对应的动态代理对象,若找到,则执行该命令,若找不到,则根据该命令生成相应的特征值,存储至缓存库中,再执行该命令,最后若该结果集中包含有Convert注解,需要根据Convert注解将结果集转换成相应的目标类型后返回,若该结果集中没有Convert注解,则直接将该结果集返回。这样,得到的DAO接口能够同时支持SQL和HQL,利用上述得到的DAO接口进行应用构建时,无需将数据整体迁移,有效降低成本,且在进行应用构建时不会产生大量的XML配置文件,管理简洁方便,开发人员在后期维护时,只需对开发代码进行维护,使得开发效率得到有效的提高。

[0096] 在具体实施例中,步骤101具体包括:

[0097] 步骤1011,在Spring.xml配置文件中利用BeanDefinitionRegistryPostProcessor确定扫描包的预定范围。

[0098] 在该步骤中,BeanDefinitionRegistryPostProcessor继承自BeanFactoryPostProcessor,是一种比较特殊的BeanFactoryPostProcessor.BeanDefinitionRegistryPostProcessor中定义的postProcessBeanDefinitionRegistry (BeanDefinitionRegistry registry)方法可以实现自定义的注册bean定义的逻辑,因此,利用BeanDefinitionRegistryPostProcessor定义Spring框架中xml配置文件的扫描范围。

[0099] 步骤1012,为扫描包设定对应的注解名称,其中,设定扫描包的初始注解名称为AutoDaoImpl。

[0100] 在该步骤中,名称AutoDaoImpl可根据开发人员的实际需要自行设定。

[0101] 步骤1013,为AutoDaoImpl匹配相应的entityClass参数,利用entityClass参数来对应带指定注解的DAO接口类的具体操作的实体类对象。

[0102] 在该步骤中,entityClass参数是指实体分类参数,根据带指定注解的DAO接口类对应的具体执行操作,确定相应的实体类对象,进而保证根据该带指定注解的DAO接口类进行动态解析后能够生成准确的动态代理对象。

[0103] 在具体实施例中,步骤102具体包括:

[0104] 步骤1021,扫描预定范围内的扫描包中带预定注解的DAO接口类。

[0105] 步骤1022,定义ClassPathBeanDefinitionScanner生成BeanDefinition,在BeanDefinition中定义FactoryBean与带预定注解的DAO接口类的映射关系,并为BeanDefinition设置对应的注入模式。

[0106] 在上述方案中,ClassPathBeanDefinitionScanner是Spring的类扫描器分析ClassPathBeanDefinitionScanner作用就是将指定包下的类。

[0107] 利用过滤器从指定包下面查找到的带预定注解的DAO接口类进行过滤,将

FactoryBean与带预定注解的DAO接口类的映射关系,包装成BeanDefinition的形式注册到IOC容器中。

[0108] 在具体实施例中,步骤103具体包括:

[0109] 步骤1031,将带预定注解的DAO接口类设置成自定义DAO接口。

[0110] 步骤1032,利用Proxy对象和自定义DAO接口,生成自定义DAO接口的动态代理对象,将动态代理对象存储至缓存库中。

[0111] 在上述方案中,Proxy对象是通过Proxy创建1个代理对象,然后通过操作代理对象允许对指定的对象的一些行为进行自定义处理。因此利用Proxy对象对自定义DAO接口进行动态自定义处理,生成相应的动态代理对象,将这些动态代理对象在缓存库中进行存储,以供后续进行查找。

[0112] 具体为:Proxy构造函数接收2个对象,第1个参数就是要处理的对象(即,自定义DAO接口),第2个参数就是自定义处理后方法的合集(即,动态代理对象)。

[0113] 在具体实施例中,步骤104还包括:

[0114] 步骤1041,从DAO接口类的命令中获取执行名称,对执行名称进行解析,得到对应的返回值。

[0115] 在该步骤中,每个DAO接口类的命令中均携带有相应的执行任务的执行名称,根据该执行名称对应的执行任务进行解析,得到与该执行任务相匹配的返回值。返回值中记录了返回值的类型,如基本类型,对象类型。

[0116] 步骤1042,根据执行名称确定对应的执行参数,其中,执行参数包括:常用参数和回调函数参数,回调函数参数中记录有回调函数参数的索引位置。

[0117] 在该步骤中,根据该返回值确定执行名称对应的执行参数。其中的常用参数是指执行名称是实体类的属性名称,如实体类是人类,则属性名称可以是name、age等。

[0118] 回调函数参数记录索引位置的好处就是可以在回调函数参数的不同位置灵活传入匿名内部类,而不局限于指定具体的传入位置。利用匿名内部类就可以实现用户的自定义业务逻辑。

[0119] 步骤1043,根据执行名称从数据中查找对应的执行对象。

[0120] 在该步骤中,每个执行名称都对应有多个执行对象。

[0121] 步骤1044,利用Convert将执行对象转换成对应的特定对象类型,其中,Convert中设定了各种特征值对应转换的对象类型,以及每种对象类型的转换方案。

[0122] 在该步骤中,Convert是文件系统修改命令,可以对执行对象进行修改转换。

[0123] 步骤1045,根据执行名称的命名规则,对执行名称进行拆分,得到对应的特征值,将特征值存储在缓存库中。

[0124] 在该步骤中,每个执行名称中都有对应的命名规则,例如设定前端为何值,后端为何值。根据各类执行名称的命名规则,将执行名称拆分成几部分,将拆分后的结果作为特征值,将该特征值与DAO接口类的命令中的DAO接口类进行关联后存储在缓存库中,以供后续进行查找。

[0125] 在具体实施例中,步骤1045具体包括:根据命名规则将执行名称拆分为前缀、中部、后缀、关键词,将拆分得到的前缀、中部、后缀、关键词作为特征值。

[0126] 其中,前缀包括但不限于下列任一或组合:保存命令、删除命令、修改命令和查询

命令。

[0127] 中部为自定义执行业务名称。

[0128] 后缀包括但不限于下列任一或组合：查询所有数据命令、根据ID查询命令、多属性关联查询命令、自定义HQL查询命令、自定义SQL查询命令、自定义SQL逻辑查询命令。

[0129] 关键词为By,用于将前缀、中部和后缀分开。

[0130] 例如：前缀标识中包括但不限于：增加：`void saveXXXX(T obj)`、更新：`void updateXXXX(T obj)`、增加或更新：`void saveOrUpdateXXXX(T obj)`、删除：`void deleteXXXXById(Integer id)`；

[0131] 后缀标识中包括但不限于：根据ID查询：`T getXXXXById(Integer id)`、多属性关联查询：`T getXXXXByProperty1AndProperty2(String Property1,int Property2)`、自定义HQL查询：`T getUniqueXXXXByHql(String hql,Object[] args)`、自定义SQL查询：`T getXXXXBySqlCommand(ISqlCommand<T> sqlCommand)`、根据ID查找并返回转换后的对象：`@Convert(clazz=XXX.class) XXX getXXXXById(Integer id)`。另外。后缀中多属性关联查询命令包括：实体类对象中的多属性之间的与、或、非逻辑关系。

[0132] 通过上述方案，将解析后的所有特征值存入缓存中，避免做二次解析，有效加快处理效率。

[0133] 在具体实施例中，步骤104还包括：

[0134] 步骤1046，从缓存库中查找DAO接口类的命令对应的目标特征值，根据目标特征值确定对应的执行逻辑。

[0135] 步骤1047，根据执行逻辑生成相应的HQL和SQL语句并执行，得到结果集。

[0136] 在上述方案中，缓存库中存储有各个DAO接口类对应的特征值，根据DAO接口类的命令从缓存库中查找，将找到的特征值作为目标特征值，该目标特征值中包含有多个数据值，将这些数据值按照执行逻辑生成生成相应的HQL和SQL语句。然后按照HQL和SQL语句进行执行处理，每个语句对应得到相应的处理结果，将这些结果进行整合后形成结果集。

[0137] 最后按照步骤105，将结果集能够进行类型转换的转换对象，均转换成相应的目标类型，将得到的结果集返回给开发人员进行处理。

[0138] 通过本实施例的上述方案，得到的DAO接口能够同时支持SQL和HQL，利用上述得到的DAO接口进行应用构建时，无需将数据整体迁移，有效降低成本，且在进行应用构建时不会产生大量的XML配置文件，管理简洁方便，开发人员在后期维护时，只需对开发代码进行维护，使得开发效率得到有效的提高。

[0139] 本申请的另一个实施例提出了一种基于Spring实现DAO接口的方法，执行时序图如图2所示。

[0140] 本实施例的执行过程包括：

[0141] 扫描指定包下带指定注解的DAO接口类，生成FactoryBean与DAO接口类的对应关系。

[0142] 生成DAO接口类对应的DAO动态代理类。

[0143] 接收DAO接口类的方法（即，DAO接口类的命令），判断该方法在对应的DAO代理类中是否存在，若是执行找到的DAO代理类的方法，返回执行结果（即，结果集），否则在DAO代理类中生成DAO接口类的方法的特征值并执行该方法，返回执行结果。

[0144] 执行返回结果时,如果返回结果中指明了Convert注解,Convert参数中指明了转换的类(即,目标类型),将返回结果中结果集自动转换为指定的类。

[0145] 首先扫描指定包下带指定注解的DAO接口类,在Spring.xml配置文件中,利用继承了BeanDefinitionRegistryPostProcessor的自定义实现类指定扫描包的范围,并指明注解名称,默认实现的注解为AutoDaoImpl(即,初始注解名称),带有此注解的DAO接口类,将会成为被自动实现的目标。该注解带有entityClass参数值,该参数用来指明DAO接口类操作的对应实体类对象。

[0146] 生成FactoryBean与DAO接口类的对应关系是指自定义实现ClassPathBeanDefinitionScanner来生成BeanDefinition,BeanDefinition是Spring中描述了一个bean的实例,包括属性值,构造方法参数值和继承自它的类的更多信息。在BeanDefinition中定义了FactoryBean与DAO接口,并设置了注入模式。

[0147] FactoryBean是工厂Bean,其返回的对象不是指定类的一个实例,而是该FactoryBean的getObject方法所返回的对象。之后在getObject方法中,利用Proxy对象和自定义DAO接口来生成该接口的动态代理对象。

[0148] 在调用DAO接口的方法时,会判断该方法在对应的DAO代理类中是否存在,调用的方法名称去方法缓存中查找,判断是否存在该方法定义,如果不存在,将解析此方法并存入缓存中。

[0149] 解析方法特征值包括根据调用的方法名解析方法的返回值、根据方法名获取方法参数、获取方法名上的注解、根据方法命名规则拆分方法名。

[0150] 方法的返回值,分析了记录返回值的类型,如基本类型,对象类型。

[0151] 获取方法参数,参数分为一般参数和回调函数参数,回调函数参数,记录了回调函数参数的索引位置。记录索引位置的好处就是可以在回调函数参数的不同位置灵活传入匿名内部类,而不局限于指定具体的传入位置。利用匿名内部类就可以实现用户的自定义业务逻辑。

[0152] 获取方法名上是否存在Convert注解。Convert注解提供了类转换能力,可将查询的结果集转换为对应的类对象。Convert中的clazz参数指定转换后的类对象。

[0153] 将方法名根据规则拆分为前缀、中部、后缀、关键词四部分,前缀标识有保存、删除、修改、查询。中部为自定义业务名称。后缀有查询所有数据、根据ID查询、多属性关联查询、自定义HQL查询、自定义SQL查询、自定义SQL逻辑查询。关键词为By,用于区分后缀部分。

[0154] 例如:

[0155] 增加:void saveXXXX(T obj);

[0156] 更新:void updateXXXX(T obj);

[0157] 增加或更新:void saveOrUpdateXXXX(T obj);

[0158] 删除:void deleteXXXXById(Integer id);

[0159] 根据ID查询:T getXXXXById(Integer id);

[0160] 多属性关联查询:T getXXXXByProperty1AndProperty2(String Property1,int Property2);

[0161] 自定义HQL查询:T getUniqueXXXXByHql(String hql,Object[] args);

[0162] 自定义SQL查询:T getXXXXBySqlCommand(ISqlCommand<T> sqlCommand);

- [0163] 根据ID查找并返回转换后的对象；
- [0164] @Convert(clazz=XXX.class)
- [0165] XXX getXXXXById(Integerid)。
- [0166] 将解析后的方法特征值存入缓存中,避免做二次解析,加快处理效率。
- [0167] 每次将根据已经在缓存中处理好的特征值,找到对应的执行逻辑,生成相应的HQL和SQL语句利用并执行,返回结果集。
- [0168] 如图3所示,为本实施例的执行过程,具体为:
- [0169] 代理调用接口方法,利用Hash进行寻址;
- [0170] 从方法缓存区中进行查找,判断是否存在该接口方法;
- [0171] 若未找到,将该接口方法输入方法特征值解析器,通过返回值类型记录器、毁掉函数位置记录器、关键词解析器、转换类记录器、转换类属性缓存区、驼峰命名解析器、SQL拼接器和HQL拼接器,处理后成功对接口方法进行解析,并将解析结果存入缓存区;
- [0172] 若找到,则将接口方法输入方法执行区,通过特征值解析器、方法参数解析器、SQL解析器、HQL解析器和Query查询器处理后得到结果集;
- [0173] 将得到的结果集输入结果集转换器进行转换处理后输出目标结果集。
- [0174] 通过本实施例的上述方案,无需iBATIS的构架下就可通过注解的方式自动实现DAO接口,实现零配置、无迁移问题、同时支持SQL和HQL的效果。
- [0175] 本申请的再一个实施例提出了一种基于Spring实现DAO接口的装置,如图4所示,包括:确定模块、扫描模块、生成模块、查找模块和判断模块依次连接;
- [0176] 确定模块,用于在Spring.xml配置文件中确定扫描包的预定范围;
- [0177] 扫描模块,用于扫描预定范围内的扫描包中带预定注解的DAO接口类,将FactoryBean与带预定注解的DAO接口类建立映射关系;
- [0178] 生成模块,用于生成带预定注解的DAO接口类的动态代理对象,并将动态代理对象存储至缓存库中;
- [0179] 查找模块,用于接收DAO接口类的命令,根据DAO接口类的命令从缓存库中查找对应的动态代理对象,若找到与DAO接口类的命令对应的动态代理对象,则执行DAO接口类的命令,得到结果集,否则,生成DAO接口类的命令对应的特征值存储在缓存库中,并执行DAO接口类的命令,得到结果集;
- [0180] 判断模块,用于判断结果集中是否包含Convert注解,若是,则从Convert注解的参数中获取结果集对应的转换对象的目标类型,将结果集转换成目标类型,返回转换后的结果集,否则,直接将结果集返回。
- [0181] 在具体实施例中,确定模块具体包括:
- [0182] 确定单元,用于在Spring.xml配置文件中利用BeanDefinitionRegistryPostProcessor确定扫描包的预定范围;
- [0183] 设定单元,用于为扫描包设定对应的注解名称,其中,设定扫描包的初始注解名称为AutoDaoImpl;
- [0184] 匹配单元,用于为AutoDaoImpl匹配相应的entityClass参数,利用entityClass参数来对应带指定注解的DAO接口类的具体操作的实体类对象。
- [0185] 在具体实施例中,扫描模块具体包括:

- [0186] 扫描单元,用于扫描预定范围内的扫描包中带预定注解的DAO接口类;
- [0187] 定义单元,用于定义ClassPathBeanDefinitionScanner生成BeanDefinition,在BeanDefinition中定义FactoryBean与带预定注解的DAO接口类的映射关系,并为BeanDefinition设置对应的注入模式。
- [0188] 在具体实施例中,生成模块具体包括:
- [0189] 设置单元,用于将带预定注解的DAO接口类设置成自定义DAO接口;
- [0190] 生成单元,用于利用Proxy对象和自定义DAO接口,生成自定义DAO接口的动态代理对象,将动态代理对象存储至缓存库中。
- [0191] 在具体实施例中,查找模块具体包括:
- [0192] 解析单元,用于从DAO接口类的命令中获取执行名称,对执行名称进行解析,得到对应的返回值;
- [0193] 参数确定单元,用于根据执行名称确定对应的执行参数,其中,执行参数包括:常用参数和回调函数参数,回调函数参数中记录有回调函数参数的索引位置;
- [0194] 查找单元,用于根据执行名称从数据中查找对应的执行对象;
- [0195] 转换单元,用于利用Convert将执行对象转换成对应的特定对象类型,其中,Convert中设定了各种特征值对应转换的对象类型,以及每种对象类型的转换方案;
- [0196] 拆分单元,用于根据执行名称的命名规则,对执行名称进行拆分,得到对应的特征值,将特征值存储在缓存库中。
- [0197] 在具体实施例中,拆分单元具体用于:
- [0198] 根据命名规则将执行名称拆分为前缀、中部、后缀、关键词,将拆分得到的前缀、中部、后缀、关键词作为特征值;
- [0199] 其中,前缀包括但不限于下列任一或组合:保存命令、删除命令、修改命令和查询命令;
- [0200] 中部为自定义执行业务名称;
- [0201] 后缀包括但不限于下列任一或组合:查询所有数据命令、根据ID查询命令、多属性关联查询命令、自定义HQL查询命令、自定义SQL查询命令、自定义SQL逻辑查询命令;
- [0202] 关键词为By,用于将前缀、中部和后缀分开。
- [0203] 在具体实施例中,查找单元,还用于从缓存库中查找DAO接口类的命令对应的目标特征值,根据目标特征值确定对应的执行逻辑;
- [0204] 查找模块具体还包括:
- [0205] 语句生成单元,用于根据执行逻辑生成相应的HQL和SQL语句并执行,得到结果集。
- [0206] 基于上述基于Spring实现DAO接口方法和装置的实施例,为了实现上述目的,本申请实施例还提供了一种计算机设备,如图5所示,包括存储器和处理器,其中存储器和处理器均设置在总线上存储器存储有计算机程序,处理器执行计算机程序时实现图1所示的基于Spring实现DAO接口的方法。
- [0207] 基于这样的理解,本申请的技术方案可以以软件产品的形式体现出来,该软件产品可以存储在一个非易失性存储器(可以是CD-ROM,U盘,移动硬盘等)中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备)执行本申请各个实施场景所述的方法。

[0208] 可选地,该设备还可以连接用户接口、网络接口、摄像头、射频(Radio Frequency, RF)电路,传感器、音频电路、WI-FI模块等等。用户接口可以包括显示屏(Display)、输入单元比如键盘(Keyboard)等,可选用户接口还可以包括USB接口、读卡器接口等。网络接口可选的可以包括标准的有线接口、无线接口(如蓝牙接口、WI-FI接口)等。

[0209] 本领域技术人员可以理解,本实施例提供的一种计算机设备的结构并不构成对该实体设备的限定,可以包括更多或更少的部件,或者组合某些部件,或者不同的部件布置。

[0210] 基于上述如图1所示方法和图4所示装置的实施例,相应的,本申请实施例还提供了一种存储介质,其上存储有计算机程序,该程序被处理器执行时实现上述如图1所示的基于Spring实现DAO接口的方法。

[0211] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到本申请可以借助软件加必要的通用硬件平台的方式来实现,也可以通过硬件实现。

[0212] 通过应用本申请的技术方案,利用Spring框架指定扫描包的预定范围,将带预定注解的DAO接口类和FactoryBean建立对应关系,并生成相应的动态代理对象存储到缓存库中,然后当程序员需要使用时,发送DAO接口类的命令,从缓存库中查找该命令对应的动态代理对象,若找到,则执行该命令,若找不到,则根据该命令生成相应的特征值,存储至缓存库中,再执行该命令,最后若该结果集中包含有Convert注解,需要根据Convert注解将结果集转换成相应的目标类型后返回,若该结果集中没有Convert注解,则直接将该结果集返回。这样,保证得到的DAO接口能够同时支持SQL和HQL,进行应用构建时,无需将数据整体迁移,不会产生大量的XML配置文件,管理简洁方便,开发人员只需对开发代码进行维护,提高开发效率。

[0213] 本领域技术人员可以理解附图只是一个优选实施场景的示意图,附图中的模块或流程并不一定是实施本申请所必须的。本领域技术人员可以理解实施场景中的装置中的模块可以按照实施场景描述进行分布于实施场景的装置中,也可以进行相应变化位于不同于本实施场景的一个或多个装置中。上述实施场景的模块可以合并为一个模块,也可以进一步拆分成多个子模块。

[0214] 上述本申请序号仅仅为了描述,不代表实施场景的优劣。以上公开的仅为本申请的几个具体实施场景,但是,本申请并非局限于此,任何本领域的技术人员能思之的变化都应落入本申请的保护范围。

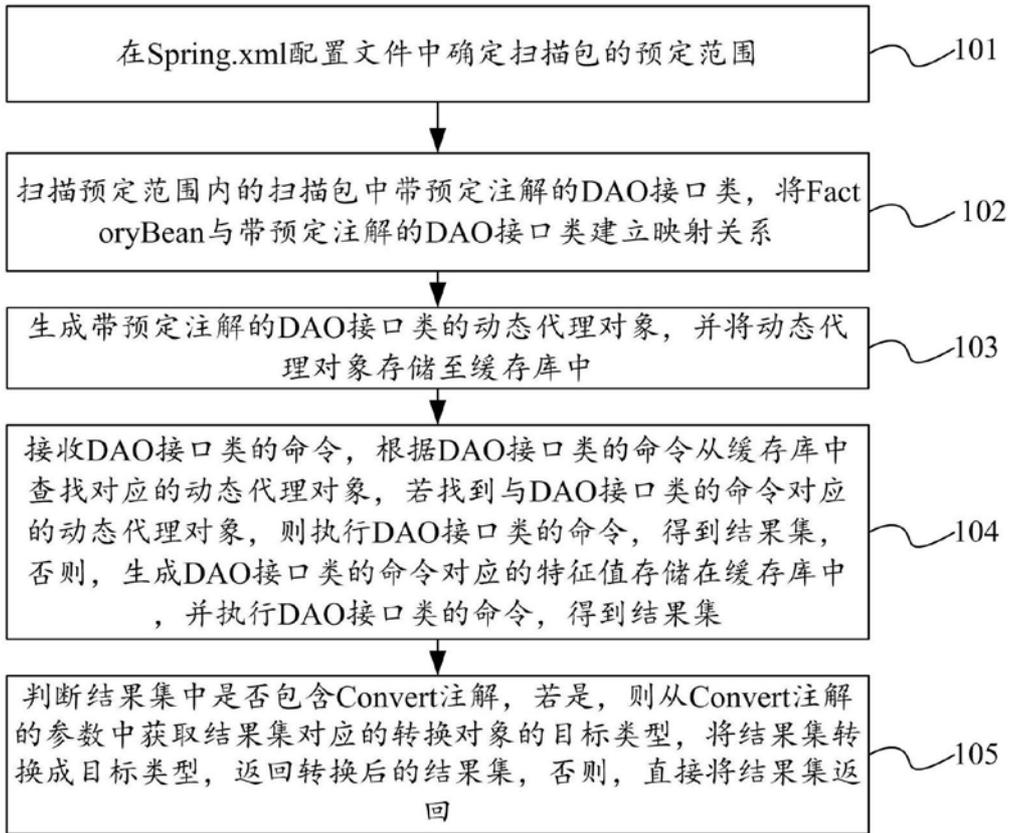


图1

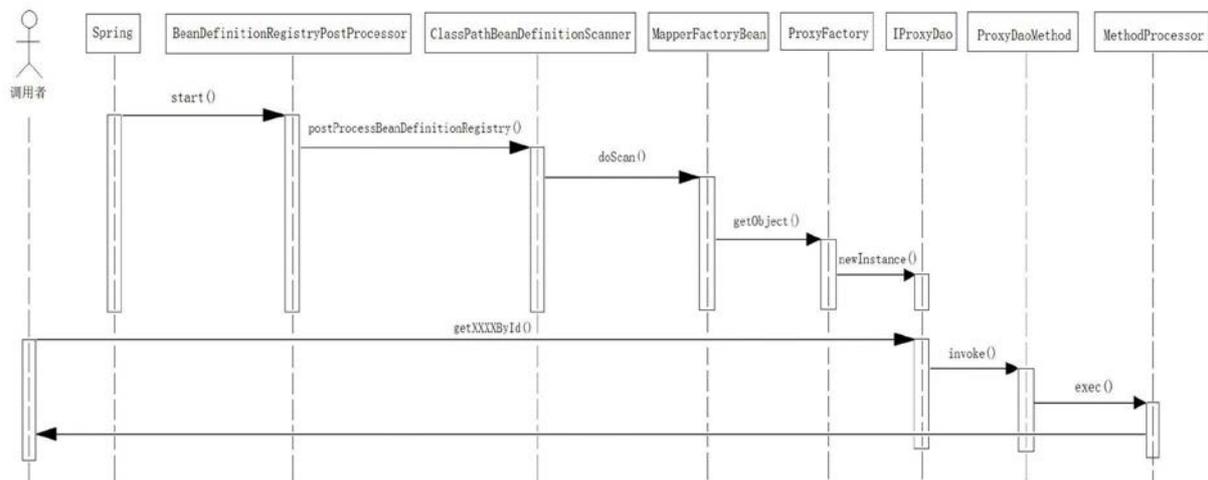


图2

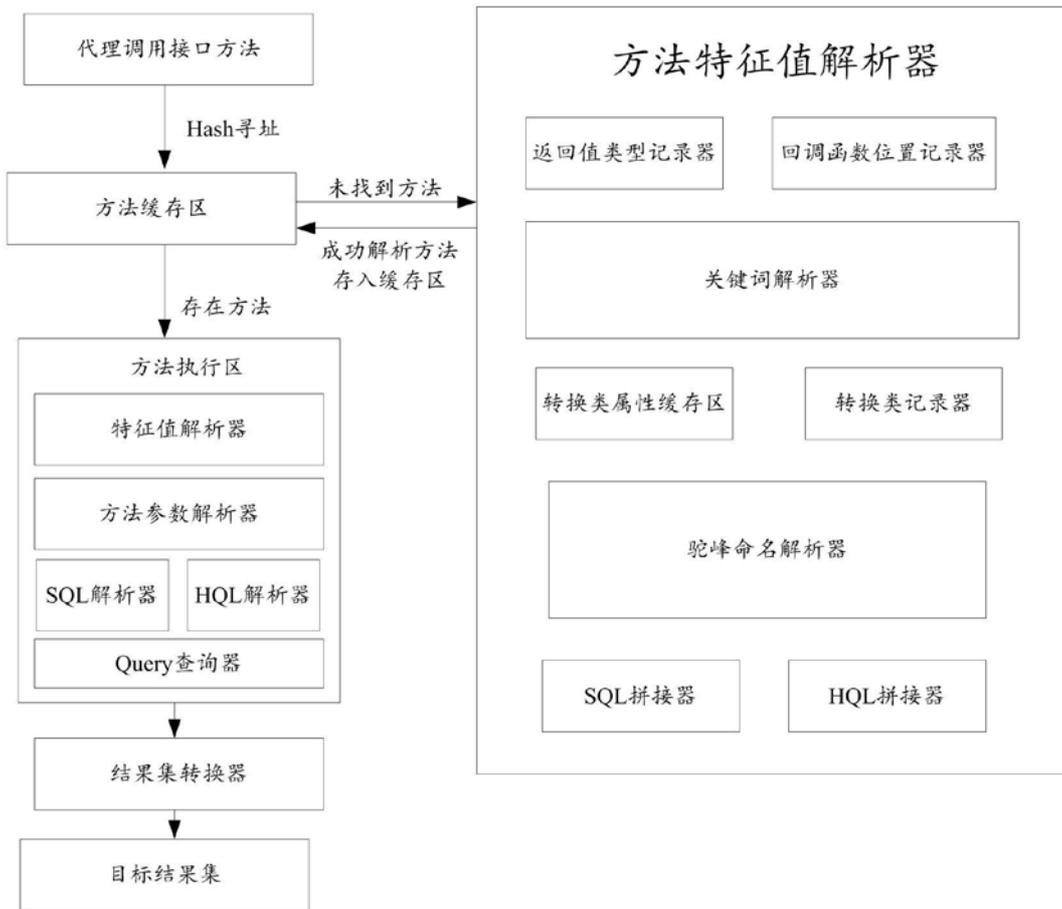


图3

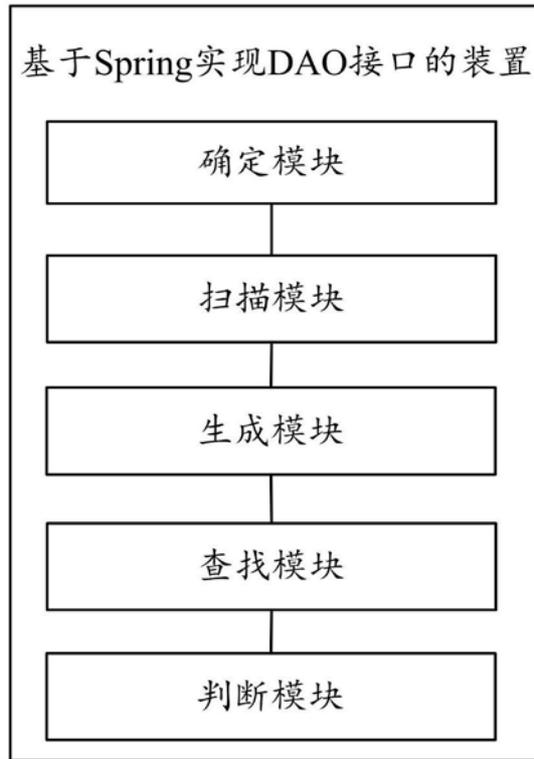


图4

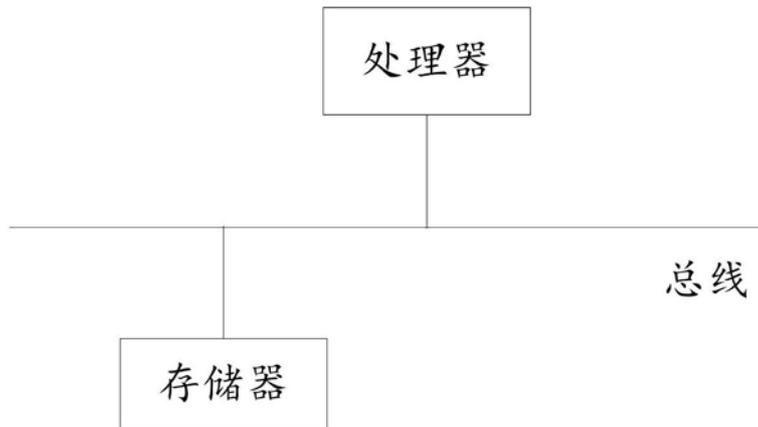


图5