



(12) 发明专利申请

(10) 申请公布号 CN 113641700 A

(43) 申请公布日 2021. 11. 12

(21) 申请号 202111003667.3

(22) 申请日 2021.08.30

(71) 申请人 北京沃东天骏信息技术有限公司
地址 100176 北京市北京经济技术开发区
科创十一街18号院2号楼4层A402室

(72) 发明人 尚鹏

(74) 专利代理机构 中原信达知识产权代理有限
责任公司 11219
代理人 杨倩 赵迪

(51) Int. Cl.

G06F 16/242 (2019.01)

G06F 16/25 (2019.01)

G06F 16/28 (2019.01)

G06F 8/36 (2018.01)

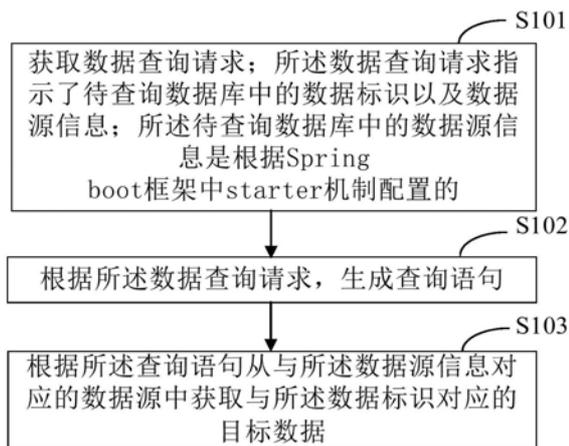
权利要求书2页 说明书11页 附图4页

(54) 发明名称

一种基于Spring boot框架的数据处理方法及装置

(57) 摘要

本发明公开了一种基于Spring boot框架的数据处理方法及装置,涉及计算机技术领域。该方法的一具体实施方式包括:获取数据查询请求;所述数据查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的;根据所述数据查询请求,生成查询语句;根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。该实施方式简化了配置数据源的流程,实现不同数据源的隔离,从而避免了数据源冲突的问题。



1. 一种基于Spring boot框架的数据处理方法,其特征在于,包括:
获取数据查询请求;所述数据查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的;
根据所述数据查询请求,生成查询语句;
根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。
2. 根据权利要求1所述的方法,其特征在于,还包括:
根据数据可视化所需的格式,利用转化注解解释器对所述目标数据进行转化。
3. 根据权利要求2所述的方法,其特征在于,所述根据数据可视化所需的格式,利用转化注解解释器对所述目标数据进行转化,包括:
根据所述转化注解解释器,对所述目标数据进行补充,使得补充后的所述目标数据连续;
基于补充后的目标数据,将所述目标数据组合成Json格式的报表数据。
4. 根据权利要求1所述的方法,其特征在于,还包括:
根据所述待查询数据库中的数据源信息包含的数据源类型和数据源地址信息,生成配置文件;
通过starter加载所述配置文件,生成所述待查询数据库中一个或多个数据源分别对应的链接,将所述链接放入Spring容器中。
5. 根据权利要求1所述的方法,其特征在于,所述根据所述数据查询请求,生成查询语句,包括:
通过调用配置有查询注解的接口,获取所述接口对应的查询注解及参数;所述参数包含了对应于所述数据标识的标准SQL语句;
根据所述待查询数据库中的数据源信息包含的数据源类型,通过查询注解解释器,将所述标准SQL语句转化成所述数据源类型对应的查询语句。
6. 根据权利要求5所述的方法,其特征在于,所述通过调用配置有查询注解的接口,获取所述接口对应的查询注解及参数,包括:
通过调用所述接口,获取Spring容器返回的对应于所述接口的代理类;
通过调用所述代理类的invoke方法,获取所述查询注解以及所述参数。
7. 根据权利要求4所述的方法,其特征在于,所述根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据,包括:
通过调用Spring容器中的连接类,将所述查询语句通过Spring容器中的链接发送给对应的数据源;
根据所述查询语句从所述数据源中获取目标数据。
8. 一种基于Spring boot框架的数据处理装置,其特征在于,包括:请求获取模块,生成模块和数据获取模块;其中,
所述请求获取模块,用于获取数据查询请求;所述数据查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的;
所述生成模块,用于根据所述数据查询请求,生成查询语句;

所述数据获取模块,用于根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。

9.一种电子设备,其特征在于,包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序,

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1-7中任一所述的方法。

10.一种计算机可读介质,其上存储有计算机程序,其特征在于,所述程序被处理器执行时实现如权利要求1-7中任一所述的方法。

一种基于Spring boot框架的数据处理方法及装置

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种基于Spring boot框架的数据处理方法及装置。

背景技术

[0002] 数据可视化的展示对于业务的分析调整有着重要的意义。

[0003] 目前,主要使用手工拼写代码或使用模板生成代码来实现持久化层数据转化成报表数据的过程。

[0004] 在实现本发明过程中,发明人发现现有技术中至少存在如下问题:

[0005] 在数据可视化过程中,使用手工拼写代码或模板生成代码来转化多个数据源的数据时,需要频繁切换数据源,这有可能造成数据不兼容或数据不一致的情况,导致数据的可视化展示出错。

发明内容

[0006] 有鉴于此,本发明实施例提供一种基于Spring boot框架的数据处理方法和装置,能够通过Spring boot的starter机制配置不同类型的数据源,简化了配置数据源的流程,并且实现了不同数据源的隔离,从而避免了数据源冲突的问题。

[0007] 进一步的,通过查询注解生成对应于不同数据源类型的查询语句,并根据不同的查询语句分别从多个数据源中查询数据,可以提高代码的重用率,减少了拼写代码的工作量,并且通过转化注解将查询得到的目标数据快速转化为报表格式的数据,进而提高了从持久化层数据到报表数据的转化效率。

[0008] 为实现上述目的,根据本发明实施例的第一方面,提供了一种基于Spring boot框架的数据处理方法,包括:

[0009] 获取数据查询请求;所述数据查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的;

[0010] 根据所述数据查询请求,生成查询语句;

[0011] 根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。

[0012] 可选地,该方法还包括:

[0013] 根据数据可视化所需的格式,利用转化注解解释器对所述目标数据进行转化。

[0014] 可选地,所述根据数据可视化所需的格式,利用转化注解解释器对所述目标数据进行转化,包括:

[0015] 根据所述转化注解解释器,对所述目标数据进行补充,使得补充后的所述目标数据连续;

[0016] 基于补充后的目标数据,将所述目标数据组合成Json格式的报表数据。

[0017] 可选地,该方法还包括:

[0018] 根据所述待查询数据库中的数据源信息包含的数据源类型和数据源地址信息,生成配置文件;

[0019] 通过starter加载所述配置文件,生成所述待查询数据库中一个或多个数据源分别对应的链接,将所述链接放入Spring容器中。

[0020] 可选地,所述根据所述数据查询请求,生成查询语句,包括:

[0021] 通过调用配置有查询注解的接口,获取所述接口对应的查询注解及参数;所述参数包含了对应于所述数据标识的标准SQL语句;

[0022] 根据所述待查询数据库中的数据源信息包含的数据源类型,通过查询注解解释器将所述标准SQL语句转化成所述数据源类型对应的查询语句。

[0023] 可选地,所述通过调用配置有查询注解的接口,获取所述接口对应的查询注解及参数,包括:

[0024] 通过调用所述接口,获取Spring容器返回的对应用于所述接口的代理类;

[0025] 通过调用所述代理类的invoke方法,获取所述查询注解以及所述参数。

[0026] 可选地,所述根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据,包括:

[0027] 通过调用Spring容器的连接类,将所述查询语句通过Spring容器中的链接发送给对应的数据源;

[0028] 根据所述查询语句从所述数据源中获取目标数据。

[0029] 根据本发明实施例的第二方面,提供了一种基于Spring boot框架的数据处理装置,包括:请求获取模块,生成模块和数据获取模块;其中,

[0030] 所述请求获取模块,用于获取数据查询请求;所述数据查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的;

[0031] 所述生成模块,用于根据所述数据查询请求,生成查询语句;

[0032] 所述数据获取模块,用于根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。

[0033] 根据本发明实施例的第三方面,提供了一种电子设备,包括:

[0034] 一个或多个处理器;

[0035] 存储装置,用于存储一个或多个程序,

[0036] 当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如上述第一方面提供的一种基于Spring boot框架的数据处理方法中任一所述的方法。

[0037] 根据本发明实施例的第四方面,提供了一种计算机可读介质,其上存储有计算机程序,所述程序被处理器执行时实现如上述第一方面提供的一种基于Spring boot框架的数据处理方法中任一所述的方法。

[0038] 上述发明中的一个实施例具有如下优点或有益效果:能够通过Spring boot的starter机制配置不同类型的数据源,简化了配置数据源的流程,并且实现了不同数据源的隔离,从而避免了数据源冲突的问题。

[0039] 进一步的,通过查询注解生成对应于不同数据源类型的查询语句,并根据不同的查询语句分别从多个数据源中查询数据,可以提高代码的重用率,减少了拼写代码的工作量,并且通过转化注解将查询得到的目标数据快速转化为报表格式的数据,进而提高了从持久化层数据到报表数据的转化效率。

[0040] 上述的非惯用的可选方式所具有的进一步效果将在下文中结合具体实施方式加以说明。

附图说明

[0041] 附图用于更好地理解本发明,不构成对本发明的不当限定。其中:

[0042] 图1是本发明一个实施例提供的一种基于Spring boot框架的数据处理方法的流程示意图;

[0043] 图2是本发明一个实施例提供的数据库查询和转化的流程示意图;

[0044] 图3是本发明一个实施例提供的基于Spring boot的自定义starter的配置流程示意图;

[0045] 图4是本发明一个实施例提供的一种通过查询注解进行数据库查询的流程示意图;

[0046] 图5是本发明一个实施例提供的一种基于Spring boot框架的数据处理装置的结构示意图;

[0047] 图6是本发明实施例可以应用于其中的示例性系统架构图;

[0048] 图7是适于用来实现本发明实施例的终端设备或服务器的计算机系统的结构示意图。

具体实施方式

[0049] 以下结合附图对本发明的示范性实施例做出说明,其中包括本发明实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本发明的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0050] 需要指出的是,在不冲突的情况下,本发明的实施例以及实施例中的技术特征可以相互结合。

[0051] 如图1所示,本发明实施例提供了一种基于Spring boot框架的数据处理方法,该方法可以包括以下步骤S101至S103:

[0052] 步骤S101:获取数据库查询请求;所述数据库查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的。

[0053] 在数据可视化的过程中,需要先将持久化层的数据查询出来再做转化,持久化层的数据通常存储于不同的数据源中。从数据库查询请求中,可以确定需要查询的数据标识以及数据源信息。

[0054] 可以理解的是,数据库查询请求中指示的数据源信息提供了关于数据源的具体访问信息,也即通过数据源信息可以连接并访问对应的数据库中的数据源并进行数据库查询。在数据可视化的过程中,可以在代码中直接写入数据库的查询语句以及数据源的访问地址、数

据源类型、数据源的访问用户及访问密码等信息。这种简单集成数据源的方式对于一个或少数几个数据源来说比较高效。然而在数据源较多情况下,在代码中集成大量数据源的信息比较容易出错,而且重复性的代码会较多,对系统具有一定的侵入性。

[0055] 因此在本发明的一个实施例中,优选的采取基于Spring boot的框架中的starter机制来配置数据库中的数据源,具体方式如下:根据所述待查询数据库中的数据源信息包含的数据源类型和数据源地址信息,生成配置文件;通过starter加载所述配置文件,生成所述待查询数据库中一个或多个数据源分别对应的链接,将所述链接放入Spring容器中。

[0056] 其中,待查询数据库中的数据源可以是很多个,这些数据源通常为关系型数据库,但是数据库具体的类型可以不同,例如可以是SQLServer、Oracle、DB2、Mysql等。并且每一个数据源都有其地址信息,也即,数据源的访问信息,包括数据源的url地址、数据源访问时的用户以及密码信息等。

[0057] 在本发明的一个可实施方式中,可以根据Spring boot框架的原理自定义一个starter,根据starter的配置规范,将数据源的类型以及数据源的地址信息生成为starter的配置文件,通过starter加载配置文件后,为不同的数据源生成不同的数据源链接,将生成的链接放入到Spring容器中。在本发明的一个实施例中,通过自定义的starter自动配置数据源的流程可以如图3中步骤S301至304所示。

[0058] 可理解的是,在查询数据的时候,通过starter配置的数据源链接,可以方便地将查询语句通过这些链接发送给不同的数据源以获取目标数据,这样使得不同的数据源相互独立,从而可以起到隔绝数据源的效果。

[0059] 步骤S102:根据所述数据查询请求,生成查询语句。

[0060] 可理解的是,当获取到数据查询请求后,需要根据不同数据源类型确定不同的查询语句,以便使用这些查询语句在对应的数据源中查询数据。

[0061] 对于不同的数据库类型,其SQL查询语句不完全相同,例如,Mysql和SQLServer的数据库查询语句相似,但也存在比较多的差异。因此为了屏蔽底层查询语句的具体细节,提取出通用的查询流程,减少代码的重复量,在本发明的一个实施例中,优选采用注解的机制来实现,其方式如下:通过调用配置有查询注解的接口,获取所述接口对应的查询注解及参数;所述参数包含了对应于所述数据标识的标准SQL语句;根据所述待查询数据库中的数据源信息包含的数据源类型,通过查询注解解释器,将所述标准SQL语句转化成所述数据源类型对应的查询语句。

[0062] 在查询之前,可以预先定义一个查询注解,再新建一个接口,在该接口上配置该查询注解,也可以在系统已有的接口上配置该查询注解。在接口上配置查询注解时,可以在查询注解的参数中加入标准SQL语句,该SQL语句是针对要查询的数据标识而组织的。

[0063] 例如,当前数据查询请求指示了需要从一张数据表中查询出message_type为messageA的所有数据,其中,数据标识为messageA,数据表为stripe_log_table,标准的SQL语句可以是:Select*From Stripe_log_table where message_type='messageA'。假如新建的接口为Test(),查询注解为@SelectSql(),那么可以将此标准SQL语句作为查询注解@SelectSql的参数,并将添加了参数的查询注解添加到接口之前,如下所示:

[0064] @SelectSql("Select*From Stripe_log_table where message_type='messageA'")Test()。

[0065] 当然,查询注解中还可以包含其他参数,例如在标准SQL查询语句中通过传递参数的方式来确定要查询的数据标识,例如message_type的值可以由变量\$messageType来确定,这个变量可以通过调用Test()接口传递给查询注解。例如可以写成如下方式:

[0066] @SelectSql(“Select*From Stripe_log_table where message_type=’ \$messageType’”)Test (@SqlParameter(“messageType”)String messageType)。其中,@SqlParameter(“messageType”)即为向@SelectSql传递参数的注解。

[0067] 在调用接口之前,还可以在starter的配置中配置查询注解所在的Java包,以便在starter启动时,自动在Java包中扫描接口,为配置有查询注解的接口生成动态代理类。该配置过程可以如图3中步骤S305至S308中所示。

[0068] 在调用接口时,本发明的一个可实施方式是:通过调用所述接口,获取Spring容器返回的对应于所述接口的代理类;通过调用所述代理类的invoke方法,获取所述查询注解以及所述参数。

[0069] 例如,在调用Test()接口时,Spring容器会返回基于starter的配置生成的对应于Test()接口的动态代理类P,通过调用动态代理类P中的invoke()方法,获取到Test()接口上配置的查询注解@SelectSql,以及查询注解中的参数,也即标准的SQL语句。

[0070] 在获取到标准的SQL语句之后,本发明的一个可实施方式为:根据所述根据所述待查询数据库中的数据源信息包含的数据源类型,通过查询注解解释器,将所述标准SQL语句转化成所述数据源类型对应的查询语句。

[0071] 其中,查询注解解释器可以根据starter中配置的数据源类型的不同,将查询注解参数中包含的标准SQL语句转化为与数据源类型对应的原生查询语句。例如,对于Mysql类型的数据源,则转化为符合Mysql查询语法的查询语句,对于SQLServer类型的数据源,则转化为符合SQLServer查询语法的查询语句。

[0072] 可理解的是,将通用的查询流程提取到查询注解中,通过查询注解解释器对于不同的数据源类型生成对应的查询语句,可以减少重复性的代码量。而且查询注解解释器可以做到兼容常用的不同类型的数据源,在每次新增数据查询请求时,查询注解解释器可以做较少改动即能适应于不同的数据源类型,节省了因为数据源不同而修改代码的时间。

[0073] 步骤S103:根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。

[0074] 通过查询注解解释器转化生成了与数据源类型对应的查询语言后,则可以将查询语句发送给数据源进行查询,以得到目标数据。在本发明的一个实施例中,可以采用如下方式:通过调用Spring容器中的连接类,将所述查询语句通过Spring容器中的链接发送给对应的数据源;根据所述查询语句从所述数据源中获取目标数据。

[0075] 其中Spring容器中的连接类是在通过starter配置数据源时被装载进Spring容器中的。通过调用此连接类,可以将生成的不同类型数据源对应的查询语句发送给不同类型的数据源,例如,将Mysql的查询语句发送给数据库中类型为Mysql的数据源,将SQLServer的查询语句发送给数据库中类型为SQLServer的数据源。从而可以通过这些查询语句从不同的数据源中查询出目标数据。

[0076] 在获取到目标数据之后,可以将目标数据存储到内存中,以便后续提供给转化注解使用。

[0077] 可以理解的是,通过查询注解以及查询注解解释器,使得在新增数据查询请求时,可以将已有的查询注解中的标准SQL语句修改成新增数据标识的标准SQL查询语句,或者新增一个查询注解并在参数中写入对应于新增数据标识的标准SQL语句,即能完成新增的查询目标数据的目的,这样可以大大节省重写代码的时间,提高数据可视化的效率。

[0078] 在从多个数据源获取到目标数据之后,需要将这些目标数据进行转化得到数据可视化所需的格式,也即转化为报表数据,以便可以根据报表数据生成具体的报表图形,例如生成折线图、柱状图、饼图等。

[0079] 在对目标数据进行转化时,本发明的一个实施例优选采用了注解机制来做数据转化,其方式可以如下:根据数据可视化所需的格式,利用转化注解解释器对所述目标数据进行转化。

[0080] 其中,转化注解解释器可以将转化注解获取的数据进行转化处理从而得到报表数据。转化注解解释器对于数据的处理流程是通用的,可以兼容多种报表数据格式,适用于大部分常用数据源的数据转化。可以理解的是,在新增数据可视化的请求时,转化注解解释器可以不做太多改动即可以将数据转化为需要的报表数据格式,节省了大量手工拼写代码的时间,达到快速响应并转化报表数据的目的,从而提高了报表数据的转化效率。

[0081] 使用转化注解解释器对目标数据进行转化的具体方式,可以如本发明的一个实施例中提供的,包括:根据所述转化注解解释器,对所述目标数据进行补充,使得补充后的所述目标数据连续;基于补充后的目标数据,将所述目标数据组合成Json格式的报表数据。

[0082] 在目标数据存储在内存之后,通过转化注解利用反射机制从内存里存储的目标数据中读取数据,再通过转化注解解释器将这些数据放入对应的集合中以方便转化。例如,将转化注解@X读取的数据放入集合x中,将转化注解@Y读取的数据放入集合y中,将转化注解@Data读取的数据放入在集合z中。这样可以按照生成折线图的坐标轴来对应数据集合,例如,X坐标轴的数据在x集合中,Y坐标轴的数据在y集合中,而X-Y坐标轴对应的点具体数据在集合z中。

[0083] 其中在集合z中,数据的key为 $X' - Y'$,key中的 X' 和 Y' 的值可能已经都在集合x和集合y中,也可能其中一个在已有的集合x或y中,还有可能均不在集合x与集合y中。也就是说结合集合z中的数据,可以有以下四种情况:对于集合z中的数据Z1,其对应的key为 $(X' 1 - Y' 1)$,key中的 $X' 1$ 在集合x中, $Y' 1$ 在集合y中;对于数据Z2,其对应的key为 $(X' 2 - Y' 2)$,key中的 $X' 2$ 在集合x, $Y' 2$ 不在集合y中;对于数据Z3,其对应的key为 $(X' 3 - Y' 3)$,key中的 $X' 3$ 不在集合x中, $Y' 3$ 在集合y中;对于数据Z4,其对应的key为 $(X' 4 - Y' 4)$,key中的 $X' 4$ 不在集合x, $Y' 4$ 也不在集合y中。

[0084] 那么为了让X、Y坐标轴的数据与z集合中的数据一一对应,可以将z集合数据的key里不在集合x中的 X' 的值,补充到对应的x集合中;将z集合数据的key里不在集合y中的 Y' 的值,补充到对应的y集合中。

[0085] 对于连续型的坐标轴数据,还可以补充坐标集合中缺少的连续值,以使坐标轴数据连续。例如,Y轴为时间轴,当前集合y中的数据为7月1日、7月3日、7月5日,此时可以补充集合中没有的数据,例如,7月2日、7月4日这两个数据,以使得坐标轴Y的时间数据连续。

[0086] 根据补充后的数据集合x、y、z,将集合中的数据组合成Json格式的报表数据,以便为生成可视化的报表图形时所使用。

[0087] 根据本发明实施例提供一种基于Spring boot框架的数据处理方法,能够通过Spring boot的starter机制配置不同类型的数据源,简化了配置数据源的流程,并且实现了不同数据源的隔离,从而避免了数据源冲突的问题。

[0088] 进一步的,通过查询注解生成对应于不同数据源类型的查询语句,并根据不同的查询语句分别从多个数据源中查询数据,可以提高代码的重用率,减少了拼写代码的工作量,并且通过转化注解将查询得到的目标数据快速转化为报表格式的数据,进而提高了从持久化层数据到报表数据的转化效率。

[0089] 图2是本发明一个实施例提供的一种数据查询和转化的流程示意图。其具体步骤如下:

[0090] 步骤S201:获取查询注解。

[0091] 通过调用配置有查询注解的接口,获取查询注解以及参数。

[0092] 步骤S202:通过查询注解解释器,生成查询语句。

[0093] 通过查询注解解释器,将查询注解中包含的标准SQL语句转化生成为与数据源类型相对应的查询语句。

[0094] 步骤S203:将查询语句发送给数据源。

[0095] 通过starter配置的Spring容器中的链接,将查询语句发送给不同类型的数据源。

[0096] 步骤S204:从数据源中通过查询语句获取目标数据。

[0097] 获取到目标数据后,将目标数据存储在内存在中。

[0098] 步骤S205:通过转化注解读取目标数据。

[0099] 利用反射机制通过转化注解从内存里的目标数据中获取转化注解对应的数据。

[0100] 步骤S206:通过转化注解解释器,生成报表数据。

[0101] 通过转化注解解释器,将转化注解获取的数据进行补充,并将数据组合成Json格式的报表数据。后续可以将生成的报表数据发送给生成报表图形的接口以便生成图表。

[0102] 如图3所示,本发明实施例提供了一种基于Spring boot框架的数据处理方法中自定义starter配置数据源以及配置接口的动态代理的流程示意图。

[0103] 其中,配置数据源的具体步骤可以为S301至S304:

[0104] 步骤S301:将数据源信息写入自定义bean对应的配置文件。

[0105] 其中数据源信息包括数据源的类型以及数据源的url地址、数据源的访问用户及密码信息。。

[0106] 步骤S302:确定是否满足创建bean的条件。

[0107] 步骤S303:加载自定义bean。

[0108] 步骤S304:加载对应的配置文件,根据数据源类型,生成不同的链接,放入Spring容器中。

[0109] 加载bean对应的数据源的配置文件,根据数据源类型的不同,生成不同的数据源的链接,将这些链接放入Spring容器中。

[0110] 配置接口的动态代理的具体步骤可以为S305至S308:

[0111] 步骤S305:配置查询注解所在Java包。

[0112] 其中查询注解可以为多个,将查询注解所在包的路径写入starter的配置文件中。

[0113] 步骤S306:导入查询注解扫描器。

- [0114] 步骤S307:根据注解所在Java包,扫描接口。
- [0115] 使用步骤S306中的扫描器,对Java包中的接口进行扫描,获取配置有查询注解的接口。
- [0116] 步骤S308:生成接口对应的动态代理类,将动态代理类放入beanDefinition中。
- [0117] 在配置完查询注解的接口动态代理类后,在Spring的后续流程则可以根据beanDefinition对bean进行实例化,然后在接口的调用过程中,通过Spring容器中实例化的bean返回对应的动态代理类,利用动态代理类中的方法获取查询注解参数中的标准SQL语句,并通过查询注解解释器生成对应于不同数据源类型的查询语句,从不同数据源中查询数据。其具体流程可以如图4所示。
- [0118] 图4是本发明一个实施例提供的一种基于Spring boot框架的数据处理方法中数据查询的流程示意图,具体步骤可以如下:
- [0119] 步骤S401:运行时,调用配置有查询注解的接口。
- [0120] 步骤S402:Spring容器返回动态代理类。
- [0121] Spring容器返回通过starter配置的接口对应的动态代理类。
- [0122] 步骤S403:调用动态代理类中的invoke方法。
- [0123] 步骤S404:获取查询注解及参数。
- [0124] 通过invoke方法以及反射机制,可以获取接口上的查询注解及参数。
- [0125] 步骤S405:根据查询注解解释器生成不同的查询语句。
- [0126] 将查询注解参数中包含的标准SQL语句转化为对应于不同数据源类型的原生查询语句。
- [0127] 步骤S406:调用Spring容器中的连接类。
- [0128] Spring容器中的连接类是在starter配置数据源时装载到Spring容器中的。通过连接类,可以将这些原生的查询语句通过Spring容器中的链接发送给不同的数据源,从而从不同的数据源中查询出目标数据。
- [0129] 步骤S407:获取查询结果并返回。
- [0130] 获取从不同的数据源中查询到的目标数据,将结果返回给代理类的invoke方法,从而将目标数据返回并存储到内存里。
- [0131] 如图5所示,本发明实施例提供了一种基于Spring boot框架的数据处理装置500,包括:请求获取模块501,生成模块502和数据获取模块503;其中,
- [0132] 所述请求获取模块501,用于获取数据查询请求;所述数据查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的;
- [0133] 所述生成模块502,用于根据所述数据查询请求,生成查询语句;
- [0134] 所述数据获取模块503,用于根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。
- [0135] 本发明一个实施例中,所述请求获取模块501,用于根据所述待查询数据库中的数据源信息包含的数据源类型和数据源地址信息,生成配置文件;通过starter加载所述配置文件,生成所述待查询数据库中一个或多个数据源分别对应的链接,将所述链接放入Spring容器中。

[0136] 本发明一个实施例中,所述生成模块502,用于通过调用配置有查询注解的接口,获取所述接口对应的查询注解及参数;所述参数包含了对应于所述数据标识的标准SQL语句;根据所述待查询数据库中的数据源信息包含的数据源类型,通过查询注解解释器,将所述标准SQL语句转化成所述数据源类型对应的查询语句。

[0137] 本发明一个实施例中,所述生成模块502,用于通过调用所述接口,获取Spring容器返回的对应用于所述接口的代理类;通过调用所述代理类的invoke方法,获取所述查询注解以及所述参数。

[0138] 本发明一个实施例中,所述数据获取模块503,用于通过调用Spring容器中的连接类,将所述查询语句通过Spring容器中的链接发送给对应的数据源;根据所述查询语句从所述数据源中获取目标数据。

[0139] 本发明一个实施例中,所述数据获取模块503,用于根据数据可视化所需的格式,利用转化注解解释器对所述目标数据进行转化。

[0140] 本发明一个实施例中,所述数据获取模块503,用于根据所述转化注解解释器,对所述目标数据进行补充,使得补充后的所述目标数据连续;基于补充后的目标数据,将所述目标数据组合成Json格式的报表数据。

[0141] 根据本发明实施例提供一种基于Spring boot框架的数据处理装置,能够通过Spring boot的starter机制配置不同类型的数据源,简化了配置数据源的流程,并且实现了不同数据源的隔离,从而避免了数据源冲突的问题。

[0142] 进一步的,通过查询注解生成对应于不同数据源类型的查询语句,并根据不同的查询语句分别从多个数据源中查询数据,可以提高代码的重用率,减少了拼写代码的工作量,并且通过转化注解将查询得到的目标数据快速转化为报表格式的数据,进而提高了从持久化层数据到报表数据的转化效率。

[0143] 图6出了可以应用本发明实施例的一种基于Spring boot框架的数据处理方法及装置的示例性系统架构600

[0144] 如图6示,系统架构600可以包括终端设备601、602、603,网络604和服务器605。网络604用以在终端设备601、602、603和服务器605之间提供通信链路的介质。网络604可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0145] 用户可以使用终端设备601、602、603通过网络604与服务器605交互,以接收或发送消息等。

[0146] 终端设备601、602、603可以是具有显示屏并且支持信息浏览的各种电子设备,包括但不限于智能手机、平板电脑、膝上型便携计算机和台式计算机等等。

[0147] 服务器605可以是提供各种服务的服务器,例如对用户利用终端设备601、602、603提出的数据查询请求提供支持的后台管理服务器。后台管理服务器可以对接收到的数据查询请求等数据进行分析等处理,并将处理结果反馈给终端设备。

[0148] 需要说明的是,本发明实施例所提供的基于Spring boot框架的数据处理方法一般由服务器605执行,相应地,基于Spring boot框架的数据处理装置一般设置于服务器605中。

[0149] 应该理解,图6的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端设备、网络和服务器。

[0150] 下面参考图7,其示出了适于用来实现本发明实施例的终端设备的计算机系统700的结构示意图。图7出的终端设备仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0151] 如图7示,计算机系统700包括中央处理单元(CPU)701,其可以根据存储在只读存储器(ROM)702中的程序或者从存储部分708加载到随机访问存储器(RAM)703中的程序而执行各种适当的动作和处理。在RAM 703中,还存储有系统700操作所需的各种程序和数据。CPU 701、ROM 702以及RAM 703通过总线704彼此相连。输入/输出(I/O)接口705也连接至总线704。

[0152] 以下部件连接至I/O接口705:包括键盘、鼠标等的输入部分706;包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分707;包括硬盘等的存储部分708;以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分709。通信部分709经由诸如因特网的网络执行通信处理。驱动器Z10也根据需要连接至I/O接口705。可拆卸介质711,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器710上,以便于从其上读出的计算机程序根据需要被安装入存储部分708。

[0153] 特别地,根据本发明公开的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本发明公开的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信部分709从网络上被下载和安装,和/或从可拆卸介质711被安装。在该计算机程序被中央处理单元(CPU)701执行时,执行本发明的系统中限定的上述功能。

[0154] 需要说明的是,本发明所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本发明中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本发明中,计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0155] 附图中的流程图和框图,图示了按照本发明各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所

标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0156] 描述于本发明实施例中所涉及到的模块可以通过软件的方式实现,也可以通过硬件的方式来实现。所描述的模块也可以设置在处理器中,例如,可以描述为:一种处理器包括获取模块,生成模块,确定模块。其中,这些模块的名称在某种情况下并不构成对该模块本身的限定,例如,获取模块还可以被描述为“获取请求的模块”。

[0157] 作为另一方面,本发明还提供了一种计算机可读介质,该计算机可读介质可以是上述实施例中描述的设备中所包含的;也可以是单独存在,而未装配入该设备中。上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被一个该设备执行时,使得该设备包括:获取数据查询请求;所述数据查询请求指示了待查询数据库中的数据标识以及数据源信息;所述待查询数据库中的数据源信息是根据Spring boot框架中starter机制配置的;根据所述数据查询请求,生成查询语句;根据所述查询语句从与所述数据源信息对应的数据源中获取与所述数据标识对应的目标数据。

[0158] 根据本发明实施例的技术方案,能够通过Spring boot的starter机制配置不同类型的数据源,简化了配置数据源的流程,并且实现了不同数据源的隔离,从而避免了数据源冲突的问题。

[0159] 进一步的,通过查询注解生成对应于不同数据源类型的查询语句,并根据不同的查询语句分别从多个数据源中查询数据,可以提高代码的重用率,减少了拼写代码的工作量,并且通过转化注解将查询得到的目标数据快速转化为报表格式的数据,进而提高了从持久化层数据到报表数据的转化效率。

[0160] 上述具体实施方式,并不构成对本发明保护范围的限制。本领域技术人员应该明白的是,取决于设计要求和因素,可以发生各种各样的修改、组合、子组合和替代。任何在本发明的精神和原则之内所作的修改、等同替换和改进等,均应包含在本发明保护范围之内。

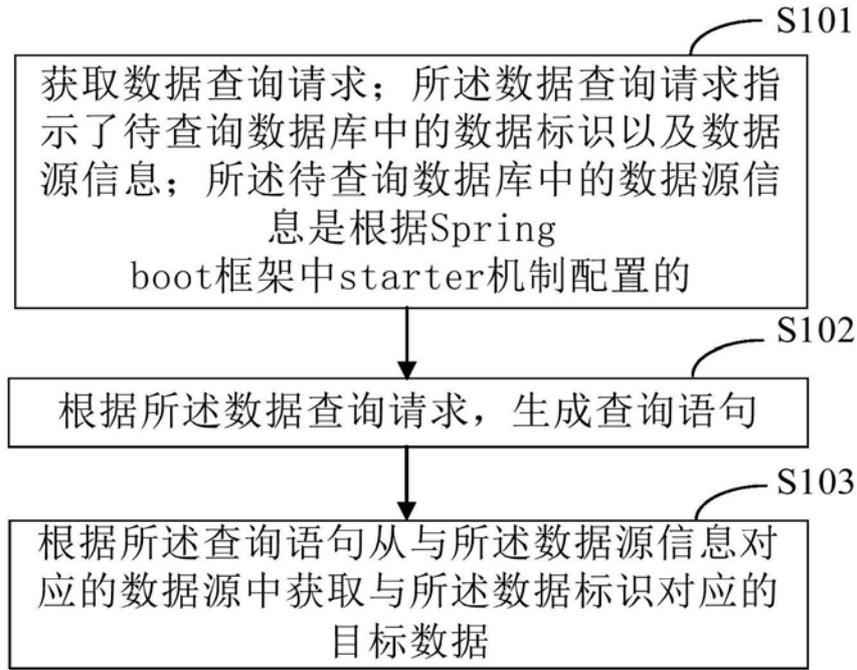


图1

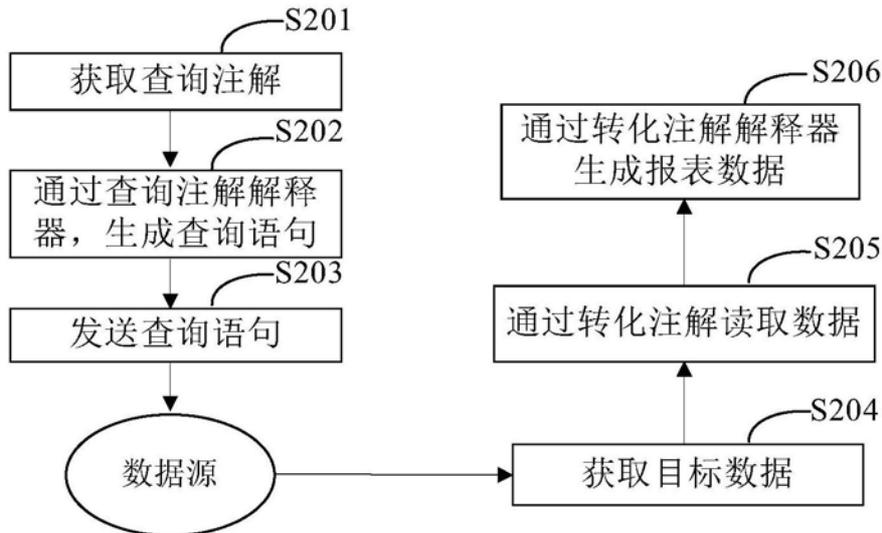


图2

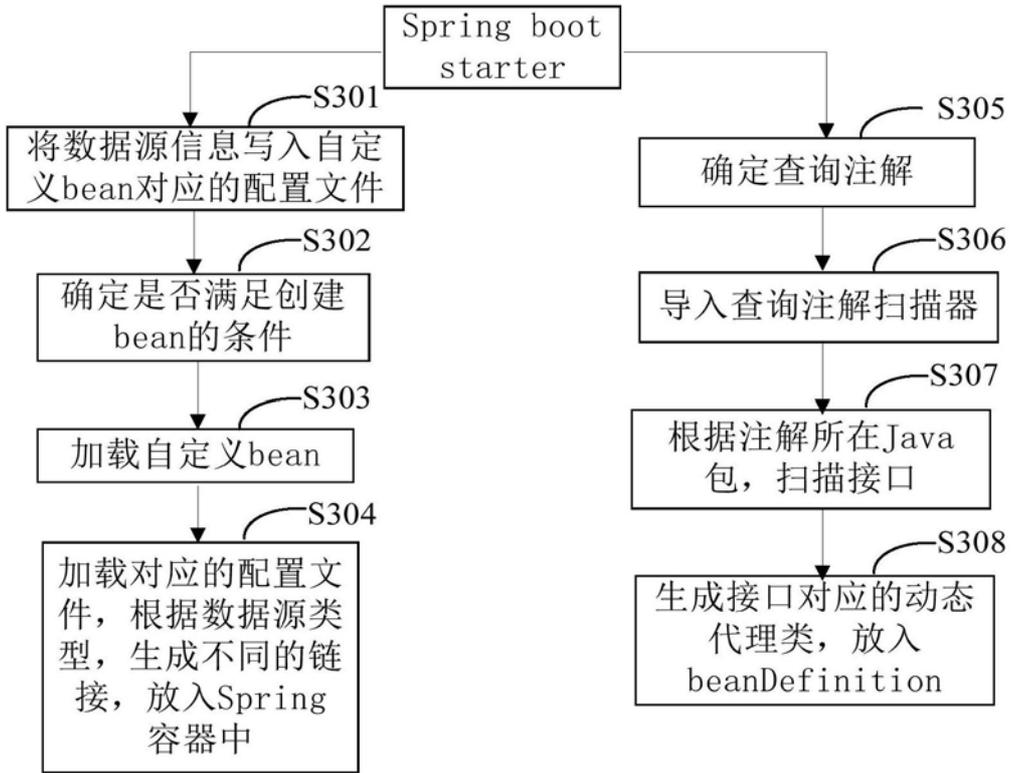


图3

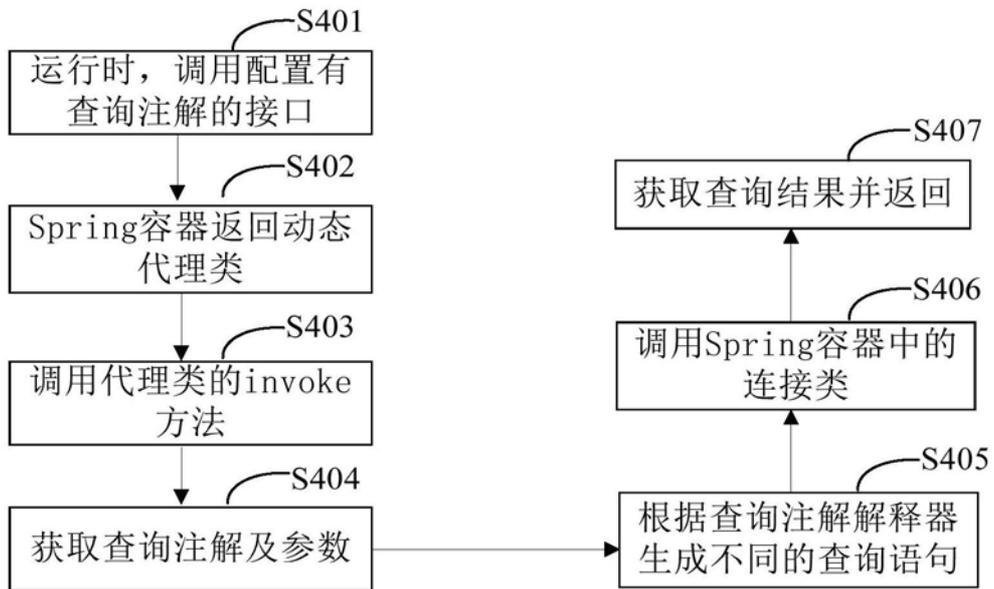


图4

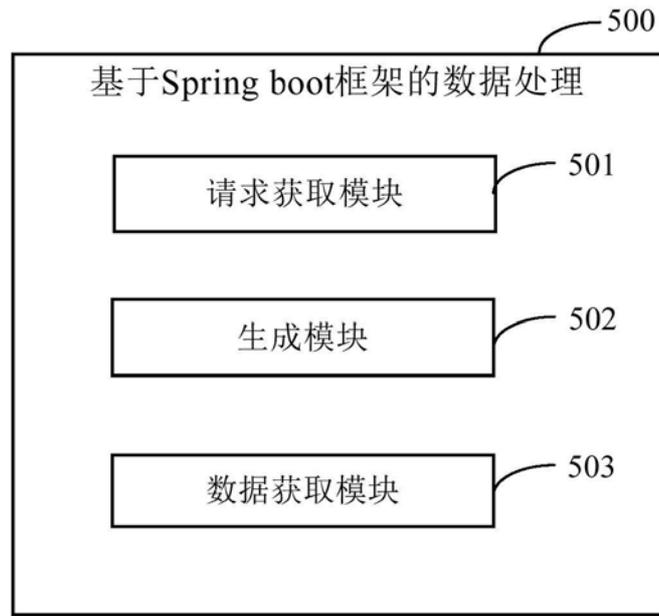


图5

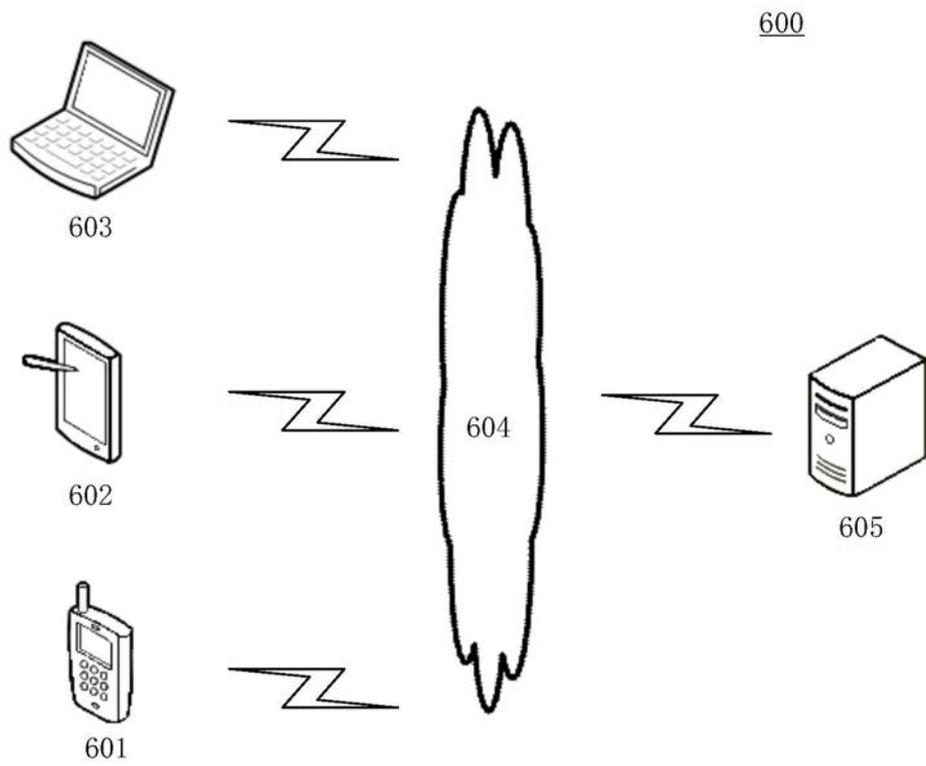


图6

700

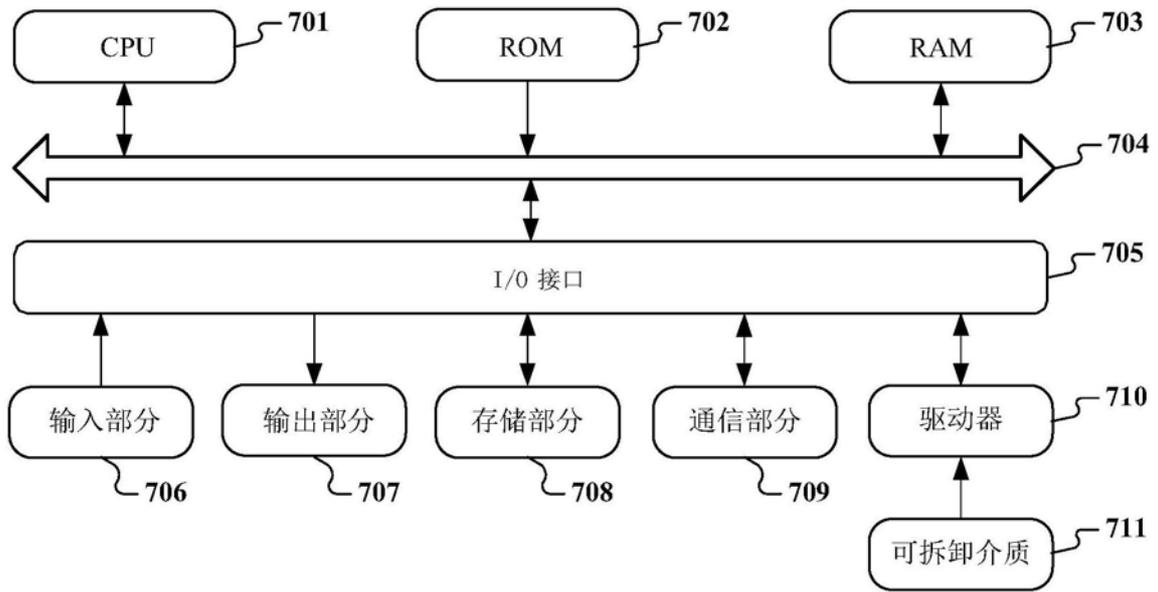


图7