



(12) 发明专利

(10) 授权公告号 CN 113434582 B

(45) 授权公告日 2023.01.03

(21) 申请号 202110708159.9

(22) 申请日 2021.06.24

(65) 同一申请的已公布的文献号  
申请公布号 CN 113434582 A

(43) 申请公布日 2021.09.24

(73) 专利权人 平安国际智慧城市科技股份有限公司

地址 518000 广东省深圳市前海深港合作区妈湾兴海大道3048号前海自贸大厦1-34层

(72) 发明人 姜明全

(74) 专利代理机构 深圳市力道知识产权代理事务所(普通合伙) 44507

专利代理师 张传义

(51) Int.Cl.

G06F 16/25 (2019.01)

G06F 16/28 (2019.01)

G06F 16/248 (2019.01)

(56) 对比文件

CN 109684586 A, 2019.04.26

US 2011202564 A1, 2011.08.18

审查员 石梦洁

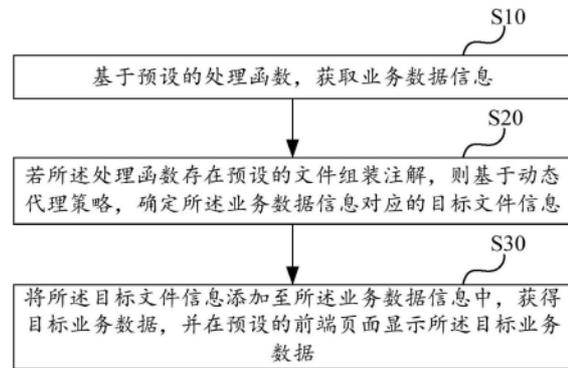
权利要求书2页 说明书9页 附图2页

(54) 发明名称

业务数据处理方法、装置、计算机设备和存储介质

(57) 摘要

本申请涉及数据处理领域,通过在检测到业务数据信息的处理函数存在文件组装注解时,基于动态代理策略实现自动返回目标文件信息并添加至业务数据信息中,避免了多次进行迭代遍历、关系绑定以及文件信息封装等繁琐操作,提高了业务数据信息处理的效率。涉及一种业务数据处理方法、装置、计算机设备和存储介质,该方法包括:基于预设的处理函数,获取业务数据信息;若处理函数存在预设的文件组装注解,则基于动态代理策略,确定业务数据信息对应的目标文件信息;将目标文件信息添加至业务数据信息中,获得目标业务数据,并在预设的前端页面显示目标业务数据。此外,本申请还涉及区块链技术,目标业务数据可存储于区块链中。



1. 一种业务数据处理方法,其特征在于,包括:

基于预设的处理函数,获取业务数据信息,所述业务数据信息存储在预设的业务数据表中;

若所述处理函数存在预设的文件组装注解,则基于动态代理策略,根据所述处理函数与所述业务数据信息确定所述业务数据信息对应的目标文件信息,所述目标文件信息存储在预设的文件信息表中;

将所述目标文件信息添加至所述业务数据信息中,获得目标业务数据,并在预设的前端页面显示所述目标业务数据;

所述基于动态代理策略,根据所述处理函数与所述业务数据信息确定所述业务数据信息对应的目标文件信息,包括:获取所述处理函数中的文件属性注解,以及确定所述业务数据信息对应的业务数据对象;对所述文件属性注解进行解析,获得所述文件属性注解对应的解析信息;根据所述解析信息与所述业务数据对象,确定所述目标文件信息。

2. 根据权利要求1所述的业务数据处理方法,其特征在于,所述确定所述业务数据信息对应的业务数据对象,包括:

基于预设的反射机制,对所述业务数据信息进行实例化,获得所述业务数据对象。

3. 根据权利要求1所述的业务数据处理方法,其特征在于,所述解析信息包括文件类型与业务表名,所述业务数据对象包括业务识别码;

所述根据所述解析信息与所述业务数据对象,确定所述目标文件信息,包括:

根据所述文件类型、所述业务表名以及所述业务识别码,在所述文件信息表进行文件信息查询,获得对应的文件信息查询结果;

根据所述文件信息查询结果,确定所述目标文件信息。

4. 根据权利要求3所述的业务数据处理方法,其特征在于,所述根据所述文件信息查询结果,确定所述目标文件信息,包括:

获取初始文件对象,将所述文件信息查询结果中的字段与每个所述字段的字段值以键值对形式添加至所述初始文件对象中,获得目标文件对象;

根据所述目标文件对象,确定所述目标文件信息。

5. 根据权利要求1所述的业务数据处理方法,其特征在于,所述基于预设的处理函数,获取业务数据信息,包括:

执行所述处理函数,在所述业务数据表进行业务数据查询,获得所述业务数据信息;

所述基于预设的处理函数,获取业务数据信息之前,还包括:

当检测到对所述处理函数的注解添加操作时,根据所述注解添加操作对所述处理函数添加文件组装注解以及文件属性注解。

6. 根据权利要求1-5任一项所述的业务数据处理方法,其特征在于,所述目标文件信息包括文件访问地址信息;

所述在预设的前端页面显示所述目标业务数据之前,还包括:

根据所述文件访问地址信息进行文件查询,获得对应的目标文件;

所述在预设的前端页面显示所述目标业务数据,所述方法还包括:

将所述目标文件加载至所述前端页面。

7. 一种业务数据处理装置,其特征在于,包括:

业务数据确定模块,用于基于预设的处理函数,获取业务数据信息,所述业务数据信息存储在预设的业务数据表中;

文件信息获取模块,用于若所述处理函数存在预设的文件组装注解,则基于动态代理策略,根据所述处理函数与所述业务数据信息确定所述业务数据信息对应的目标文件信息,所述目标文件信息存储在预设的文件信息表中;

文件信息添加模块,用于将所述目标文件信息添加至所述业务数据信息中,获得目标业务数据,并在预设的前端页面显示所述目标业务数据;

所述文件信息获取模块还用于获取所述处理函数中的文件属性注解,以及确定所述业务数据信息对应的业务数据对象;对所述文件属性注解进行解析,获得所述文件属性注解对应的解析信息;根据所述解析信息与所述业务数据对象,确定所述目标文件信息。

8. 一种计算机设备,其特征在于,所述计算机设备包括存储器和处理器;

所述存储器,用于存储计算机程序;

所述处理器,用于执行所述计算机程序并在执行所述计算机程序时实现如权利要求1至6任一项所述的业务数据处理方法。

9. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时使所述处理器实现如权利要求1至6任一项所述的业务数据处理方法。

## 业务数据处理方法、装置、计算机设备和存储介质

### 技术领域

[0001] 本申请涉及数据处理领域,尤其涉及一种业务数据处理方法、装置、计算机设备和存储介质。

### 背景技术

[0002] 在大多数业务数据处理系统中,在将业务数据信息返回给前端页面时,需要同时返回业务数据信息对应的文件信息。在现有技术中,通常将业务数据信息存储在业务数据表中,将文件信息存储在文件信息表中;在返回业务数据信息时,需要根据预设的业务数据信息与文件信息之间的对应关系,从文件信息表中遍历查找对应的文件信息并将文件信息添加至业务数据信息中。然而,当有多个业务数据或多个文件信息需要处理时,这种处理方式需要重复进行迭代遍历、关系绑定以及文件信息封装等繁琐操作,降低了业务数据处理的效率。

[0003] 因此,如何提高处理业务数据的效率成为亟需解决的问题。

### 发明内容

[0004] 本申请提供了一种业务数据处理方法、装置、计算机设备和存储介质,通过在检测到业务数据信息对应的处理函数存在文件组装注解时,基于动态代理策略实现自动返回目标文件信息并添加至业务数据信息中,避免了多次进行迭代遍历、关系绑定以及文件信息封装等繁琐操作,提高了业务数据信息处理的效率。

[0005] 第一方面,本申请提供了一种业务数据处理方法,所述方法包括:

[0006] 基于预设的处理函数,获取业务数据信息;

[0007] 若所述处理函数存在预设的文件组装注解,则基于动态代理策略,确定所述业务数据信息对应的目标文件信息;

[0008] 将所述目标文件信息添加至所述业务数据信息中,获得目标业务数据,并在预设的前端页面显示所述目标业务数据。

[0009] 第二方面,本申请还提供了一种业务数据处理装置,所述装置包括:

[0010] 业务数据确定模块,用于基于预设的处理函数,获取业务数据信息;

[0011] 文件信息获取模块,用于若所述处理函数存在预设的文件组装注解,则基于动态代理策略,确定所述业务数据信息对应的目标文件信息;

[0012] 文件信息添加模块,用于将所述目标文件信息添加至所述业务数据信息中,获得目标业务数据,并在预设的前端页面显示所述目标业务数据。

[0013] 第三方面,本申请还提供了一种计算机设备,所述计算机设备包括存储器和处理器;

[0014] 所述存储器,用于存储计算机程序;

[0015] 所述处理器,用于执行所述计算机程序并在执行所述计算机程序时实现如上述的业务数据处理方法。

[0016] 第四方面,本申请还提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时使所述处理器实现如上述的业务数据处理方法。

[0017] 本申请公开了一种业务数据处理方法、装置、计算机设备和存储介质,通过执行预设的处理函数,可以获取到业务数据信息;通过在判定处理函数存在预设的文件组装注解时,基于动态代理策略确定业务数据信息对应的目标文件信息,可以实现自动返回目标文件信息,避免了多次进行迭代遍历、关系绑定以及文件信息封装等繁琐操作,提高了业务数据信息处理的效率;通过将目标文件信息添加至业务数据信息中,可以获得包含目标文件信息与业务数据信息的目标业务数据;通过在预设的前端页面显示目标业务数据,便于用户直观地查看目标业务数据。

### 附图说明

[0018] 为了更清楚地说明本申请实施例技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0019] 图1是本申请实施例提供的一种业务数据处理方法的示意性流程图;

[0020] 图2是本申请实施例提供的一种获取业务数据信息的示意性流程图;

[0021] 图3是本申请实施例提供的一种确定目标文件信息的子步骤的示意性流程图;

[0022] 图4是本申请实施例提供的一种显示目标业务数据的示意性流程图;

[0023] 图5是本申请实施例提供的一种业务数据处理装置的示意性框图;

[0024] 图6是本申请实施例提供的一种计算机设备的结构示意图。

### 具体实施方式

[0025] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0026] 附图中所示的流程图仅是示例说明,不是必须包括所有的内容和操作/步骤,也不是必须按所描述的顺序执行。例如,有的操作/步骤还可以分解、组合或部分合并,因此实际执行的顺序有可能根据实际情况改变。

[0027] 应当理解,在此本申请说明书中所使用的术语仅仅是出于描述特定实施例的目的而并不意在限制本申请。如在本申请说明书和所附权利要求书中所使用的那样,除非上下文清楚地指明其它情况,否则单数形式的“一”、“一个”及“该”意在包括复数形式。

[0028] 还应当理解,在本申请说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合,并且包括这些组合。

[0029] 本申请的实施例提供了一种业务数据处理方法、装置、计算机设备和存储介质。其中,该业务数据处理方法可以应用于服务器或终端中,通过在检测到业务数据信息对应的处理函数存在文件组装注解时,基于动态代理策略实现自动返回目标文件信息并添加至业务数据信息中,避免了多次进行迭代遍历、关系绑定以及文件信息封装等繁琐操作,提高了

业务数据信息处理的效率。

[0030] 其中,服务器可以为独立的服务器,也可以为服务器集群。终端可以是智能手机、平板电脑、笔记本电脑和台式电脑等电子设备。

[0031] 下面结合附图,对本申请的一些实施方式作详细说明。在不冲突的情况下,下述的实施例及实施例中的特征可以相互组合。

[0032] 如图1所示,业务数据处理方法包括步骤S10至步骤S30。

[0033] 步骤S10、基于预设的处理函数,获取业务数据信息。

[0034] 需要说明的是,本申请实施例提供的业务数据处理方法可以应用于服务器或终端上的业务处理系统,用户可以通过业务处理系统发起业务查询请求,业务处理系统根据业务查询请求自动查询并返回目标业务数据。

[0035] 示例性的,业务处理系统在接收到业务处理请求时,可以先执行预设的处理函数,获得待处理的业务数据信息;然后,在确定处理函数存在文件组装注解时,基于动态代理策略,自动查询并返回业务数据信息对应的目标文件信息;最后,将目标文件信息添加至业务数据信息中,获得目标业务数据。

[0036] 可以理解的是,当查询业务数据信息对应的文件信息时,现有技术是通过执行预设的处理函数,多次进行迭代遍历与、关系绑定以及文件信息封装等繁琐操作,最后返回目标业务数据,因此存在处理效率低的问题;并且对应的处理函数需要编写大量代码,导致处理函数过大,降低了处理速度。而本申请实施例提供的业务数据处理方法是通过执行处理函数,返回业务数据信息;然后在确定处理函数存在文件组装注解时,基于动态代理策略,确定业务数据信息对应的目标文件信息;将目标文件信息添加至业务数据信息,得到目标业务数据。从而避免了通过处理函数多次进行迭代遍历、关系绑定以及文件信息封装等繁琐操作,提高了业务数据信息处理的效率。此外,在本申请实施例中,处理函数只需包含少量代码即可实现返回业务数据信息,从而提升了处理速度。

[0037] 在一些实施例中,基于预设的处理函数,获取业务数据信息,可以包括:执行处理函数,在预设的业务数据表进行业务数据查询,获得业务数据信息。

[0038] 需要说明的是,处理函数是预先编译好的函数,用于自动查询业务数据。示例性的,处理函数可以是`public UserInfo getUserInfo() {}`,用于自动查询用户的业务数据信息。

[0039] 示例性的,预设的业务数据表存储有多个业务数据。例如,业务识别号、用户名称以及用户年龄等字段。

[0040] 通过执行处理函数,可以获得待处理的业务数据信息,并且处理函数只需较少代码即可实现返回业务数据信息,提高了处理速度。

[0041] 请参阅2,图2是本申请实施例提供了一种获取业务数据信息的示意性流程图。如图2所示,当检测到用户发起的业务查询请求时,可以根据业务查询请求调用并执行处理函数,在预设的业务数据表进行业务数据查询,获得业务数据信息。其中,业务查询请求可以包括业务识别号、用户名称以及文件信息等字段,用户可以通过业务识别号或用户名称,查询对应的业务数据信息与文件信息。

[0042] 示例性的,业务数据信息可以表示为`{"id":1,"name":"小明","age":17,"fileModel":null}`;其中,“id”表示业务识别码,用于识别业务的类型;“fileModel”表示

文件信息。可以理解的是,由于业务数据信息是存储在业务数据表,业务数据信息对应的文件信息存储在文件信息表,因此业务数据信息中的文件信息对应的字段值为空值,需要另外查询文件信息并将文件信息添加至业务数据信息中。

[0043] 在一些实施例中,在基于预设的处理函数,获取业务数据信息之前,还可以包括:当检测到对处理函数的注解添加操作时,根据注解添加操作对处理函数添加文件组装注解以及文件属性注解。

[0044] 需要说明的是,用户可以根据实际需要,对处理函数添加文件组装注解以及文件属性注解;从而可以在处理函数返回待处理的业务数据信息时,基于动态代理策略,确定业务数据信息对应的目标文件信息并将目标文件信息添加至业务数据信息中。

[0045] 示例性的,当检测到用户对处理函数的注解添加操作时,根据注解添加操作对处理函数添加文件组装注解以及文件属性注解。其中,文件组装注解可以是@FileAssembly注解;文件属性注解可以是@FilePropertySign注解。

[0046] 需要说明的是,文件组装注解用于表示哪些处理函数在返回业务数据信息时,需要基于动态代理策略,查询目标文件信息。文件属性注解定义了文件类型与业务表名,用于查询目标文件信息。

[0047] 通过根据注解添加操作对处理函数添加文件组装注解以及文件属性注解,可以在处理函数返回业务数据信息之后,通过处理函数中的文件组装注解与文件属性注解查询并返回目标文件信息。

[0048] 步骤S20、若所述处理函数存在预设的文件组装注解,则基于动态代理策略,确定所述业务数据信息对应的目标文件信息。

[0049] 在本申请实施例中,在执行预设的处理函数,获取业务数据信息之后,需要判断处理函数是否存在预设的文件组装注解。

[0050] 示例性的,可以通过APT工具,对处理函数进行注解扫描,以获取处理函数中的注解。需要说明的是,APT工具(Annotation Processing Tool,注解处理工具器)是一种处理注解的工具,用来在编译时扫描和处理注解。

[0051] 在一些实施例中,当确定处理函数存在预设的文件组装注解时,可以基于动态代理策略,确定业务数据信息对应的目标文件信息。

[0052] 在另一些实施例中,当确定处理函数不存在预设的文件组装注解时,则直接在前端页面显示业务数据信息,或者返回查询失败的提示消息。

[0053] 需要说明的是,动态代理策略可以包括AOP (Aspect Oriented Programming,面向切向编程)技术。AOP技术主要是针对业务处理过程中的切面进行提取,它所面对的是处理过程中的某个步骤或阶段,对业务逻辑的各个部分进行了隔离。从而降低业务逻辑各部分之间的耦合,提高程序的可重用性,提高开发效率。

[0054] 通过在确定处理函数存在预设的文件组装注解时,基于动态代理策略,确定业务数据信息对应的目标文件信息,实现自动返回目标文件信息,避免了通过处理函数多次进行迭代遍历、关系绑定等繁琐操作,从而提高了业务数据信息处理的效率。

[0055] 示例性的,可以基于AOP技术,确定业务数据信息对应的目标文件信息。在本申请实施例中,将对如何确定业务数据信息对应的目标文件信息进行详细说明。请参阅图3,图3是本申请实施例提供的一种确定目标文件信息的子步骤的示意性流程图,具体可以包括以

下步骤S201至步骤S203。

[0056] S201、获取所述处理函数中的文件属性注解,以及确定所述业务数据信息对应的业务数据对象。

[0057] 示例性的,可以通过APT工具,对处理函数进行注解扫描,获得处理函数中的文件属性注解。例如,文件属性注解的内容可以如下所示:

[0058] @FilePropertySign(tablename="t\_user",fileType="user\_info")。其中,tablename表示业务表名;fileType表示文件类型。

[0059] 在一些实施例中,确定业务数据信息对应的业务数据对象,可以包括:基于预设的反射机制,对业务数据信息进行实例化,获得业务数据对象。

[0060] 示例性的,预设的反射机制可以是Java反射机制。需要说明的是,Java反射机制是一种动态获取信息以及动态调用对象的方法的功能。例如,在运行状态中,对于任意一个类,Java反射机制能够获取这个类的所有属性和方法;对于任意一个对象,Java反射机制能够调用它的任意一个方法和属性。实例化是指通过类(class)创建出该类的一个对象(object)的过程。

[0061] 可以理解的是,在本申请实施例中,业务数据信息可以当作一个类;可以对业务数据信息进行实例化,得到对应的业务数据对象。其中,具体的实例化过程,在此不作限定。

[0062] 示例性的,业务数据信息对应的业务数据对象包括业务识别码。可以理解的是,由于业务数据信息包括业务识别码,因此实例化得到的业务数据对象也包括业务识别码。

[0063] S202、对所述文件属性注解进行解析,获得所述文件属性注解对应的解析信息。

[0064] 示例性的,可以对文件属性注解@FilePropertySign进行解析,得到文件属性注解@FilePropertySign对应的解析信息。其中,解析是指读取文件属性注解中的相关信息。

[0065] 示例性的,解析信息至少包括文件类型与业务表名等字段。

[0066] S203、根据所述解析信息与所述业务数据对象,确定所述目标文件信息。

[0067] 需要说明的是,在本申请实施例中,在确定业务数据信息对应的业务数据对象以及获得文件属性注解对应的解析信息之后,可以根据解析信息与业务数据对象,确定目标文件信息。

[0068] 在一些实施例中,根据解析信息与业务数据对象,确定目标文件信息,可以包括:根据文件类型、业务表名以及业务识别码,在预设的文件信息表进行文件信息查询,获得对应的文件信息查询结果;根据文件信息查询结果,确定目标文件信息。

[0069] 示例性的,预设的文件信息表可以包括不同业务类型、不同业务表名对应的文件信息。可以理解的是,由于同一类型业务可能对应有多个业务表名,因此需要同时根据业务类型与业务表名,才可以准确地确定某个业务表;进而根据文件类型在业务表中,确定某个文件的文件信息。

[0070] 示例性的,可以根据文件类型、业务表名以及业务识别码,在预设的文件信息表进行文件信息查询,获得对应的文件信息查询结果。其中,文件信息查询结果可以包括文件名称、文件类型以及文件访问地址等字段与字段值。

[0071] 通过根据文件类型、业务表名以及业务识别码,可以准确地根据预设的文件信息表查询获得对应的文件信息查询结果。

[0072] 在一些实施方式中,根据文件信息查询结果,确定目标文件信息,可以包括:获取

初始文件对象,将文件信息查询结果中的字段与每个字段的字段值以键值对形式添加至初始文件对象中,获得目标文件对象;根据目标文件对象,确定目标文件信息。

[0073] 示例性的,可以预先创建一个初始文件对象;然后,基于反射机制,将文件信息查询结果中的字段与每个字段的字段值以键值对形式添加至初始文件对象中,获得目标文件对象。

[0074] 例如,目标文件对象可以表示为{"fileModel":[{"FileKey":"123hhh12765", "FileName":"PC-1278.jpg", "FileTYpe": ".jpg", "businessName":"用户照片", "businessType":"user\_info", "fileUrl":"http://anssbnh\*\*\*ssscscs\*\*"}]。其中, "FileKey"表示文件键值; "FileName"表示文件名称; "FileTYpe"表示文件类型; "businessName"表示业务表名; "businessType"表示业务类型; "fileUrl"表示文件访问地址信息。

[0075] 示例性的,可以将目标文件对象,确定为目标文件信息。

[0076] 步骤S30、将所述目标文件信息添加至所述业务数据信息中,获得目标业务数据,并在预设的前端页面显示所述目标业务数据。

[0077] 请参阅图4,图4是本申请实施例提供的一种显示目标业务数据的示意性流程图。如图4所示,在获得目标文件信息之后,可以将目标文件信息添加至业务数据信息中,获得目标业务数据;然后在预设的前端页面显示目标业务数据。其中,预设的前端页面可以是业务处理系统中的页面。例如,将目标业务数据渲染至前端页面中。

[0078] 示例性的,目标业务数据可以包括:{"id":1, "name":"小明", "age":17, "fileModel":[{"FileKey":"123hhh12765", "FileName":"PC-1278.jpg", "FileTYpe": ".jpg", "businessName":"用户照片", "businessType":"user\_info", "fileUrl":"http://anssbnh\*\*\*ssscscs\*\*"}]。}

[0079] 为进一步保证上述目标业务数据的私密和安全性,上述目标业务数据可以存储于一区块链的节点中。

[0080] 通过将目标文件信息添加至业务数据信息中,可以自动获得目标业务数据,避免了通过处理函数多次进行文件信息封装的繁琐操作,提高了业务数据信息处理的效率。通过在预设的前端页面显示目标业务数据,便于用户直观地查看目标业务数据。

[0081] 在一些实施例中,在预设的前端页面显示目标业务数据之前,还可以包括:根据文件访问地址信息进行文件查询,获得对应的目标文件。

[0082] 需要说明的是,在前端页面显示目标业务数据时,由于目标业务数据中包括文件名称与文件名称对应的文件,因此需要同时加载对应的文件。

[0083] 示例性的,可以根据文件访问地址信息,在本地磁盘或本地数据库进行文件查询,获得对应的目标文件。

[0084] 示例性的,也可以根据文件访问地址信息,在云端服务器进行文件查询,并下载对应的目标文件。

[0085] 示例性的,还可以根据文件访问地址信息,在区块链节点进行文件查询,获得对应的目标文件。

[0086] 在一些实施例中,在预设的前端页面显示目标业务数据时,还可以包括:将目标文件加载至前端页面。

[0087] 示例性的,在根据文件访问地址信息进行文件查询,获得对应的目标文件之后,可以将目标文件加载至前端页面。

[0088] 例如,若目标文件是用户照片,则可以将用户照片加载至前端页面。例如,若目标文件是产品图片,则可以将产品图片加载至前端页面。

[0089] 通过根据文件访问地址信息加载目标文件并在前端页面显示,可以实现显示目标业务数据的同时,显示对应的目标文件,便于用户直观地查看目标文件。

[0090] 上述实施例提供的业务数据处理方法,通过执行处理函数,可以获得待处理的业务数据信息,并且处理函数只需较少代码即可实现返回业务数据信息,提高了处理速度;通过根据注解添加操作对处理函数添加文件组装注解以及文件属性注解,可以在处理函数返回业务数据信息之后,通过处理函数中的文件组装注解与文件属性注解查询并返回目标文件信息;通过在确定处理函数存在预设的文件组装注解时,基于动态代理策略,确定业务数据信息对应的目标文件信息,实现自动返回目标文件信息,避免了通过处理函数多次进行迭代遍历、关系绑定等繁琐操作,从而提高了业务数据信息处理的效率;通过根据文件类型、业务表名以及业务识别码,可以准确地在预设的文件信息表查询获得对应的文件信息查询结果;通过将目标文件信息添加至业务数据信息中,可以自动获得目标业务数据,避免了通过处理函数多次进行文件信息封装的繁琐操作,提高了业务数据信息处理的效率;通过在预设的前端页面显示目标业务数据,便于用户直观地查看目标业务数据;通过根据文件访问地址信息加载目标文件并在前端页面显示,可以实现显示目标业务数据的同时,显示对应的目标文件,便于用户直观地查看目标文件。

[0091] 请参阅图5,图5是本申请的实施例还提供一种业务数据处理装置1000的示意性框图,该业务数据处理装置用于执行前述的业务数据处理方法。其中,该业务数据处理装置可以配置于服务器或终端中。

[0092] 如图5所示,该业务数据处理装置1000,包括:业务数据确定模块1001、文件信息获取模块1002和文件信息添加模块1003。

[0093] 业务数据确定模块1001,用于基于预设的处理函数,获取业务数据信息。

[0094] 文件信息获取模块1002,用于若所述处理函数存在预设的文件组装注解,则基于动态代理策略,确定所述业务数据信息对应的目标文件信息。

[0095] 文件信息添加模块1003,用于将所述目标文件信息添加至所述业务数据信息中,获得目标业务数据,并在预设的前端页面显示所述目标业务数据。

[0096] 需要说明的是,所属领域的技术人员可以清楚地了解到,为了描述的方便和简洁,上述描述的装置和各模块的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0097] 上述的装置可以实现为一种计算机程序的形式,该计算机程序可以在如图6所示的计算机设备上运行。

[0098] 请参阅图6,图6是本申请实施例提供的一种计算机设备的结构示意图。

[0099] 请参阅图6,该计算机设备包括通过系统总线连接的处理器和存储器,其中,存储器可以包括非易失性存储介质和内存储器。

[0100] 处理器用于提供计算和控制能力,支撑整个计算机设备的运行。

[0101] 内存储器为非易失性存储介质中的计算机程序的运行提供环境,该计算机程序被

处理器执行时,可使得处理器执行任意一种业务数据处理方法。

[0102] 应当理解的是,处理器可以是中央处理单元(Central Processing Unit,CPU),该处理器还可以是其他通用处理器、数字信号处理器(Digital Signal Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现场可编程门阵列(Field-Programmable Gate Array,FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。其中,通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0103] 其中,在一个实施例中,所述处理器用于运行存储在存储器中的计算机程序,以实现如下步骤:

[0104] 基于预设的处理函数,获取业务数据信息;若所述处理函数存在预设的文件组装注解,则基于动态代理策略,确定所述业务数据信息对应的目标文件信息;将所述目标文件信息添加至所述业务数据信息中,获得目标业务数据,并在预设的前端页面显示所述目标业务数据。

[0105] 在一个实施例中,所述处理器在实现基于动态代理策略,确定所述业务数据信息对应的目标文件信息时,用于实现:

[0106] 获取所述处理函数中的文件属性注解,以及确定所述业务数据信息对应的业务数据对象;对所述文件属性注解进行解析,获得所述文件属性注解对应的解析信息;根据所述解析信息与所述业务数据对象,确定所述目标文件信息。

[0107] 在一个实施例中,所述处理器在实现确定所述业务数据信息对应的业务数据对象时,用于实现:

[0108] 基于预设的反射机制,对所述业务数据信息进行实例化,获得所述业务数据对象。

[0109] 在一个实施例中,所述解析信息包括文件类型与业务表名,所述业务数据对象包括业务识别码;所述处理器在实现根据所述解析信息与所述业务数据对象,确定所述目标文件信息时,用于实现:

[0110] 根据所述文件类型、所述业务表名以及所述业务识别码,在预设的文件信息表进行文件信息查询,获得对应的文件信息查询结果;根据所述文件信息查询结果,确定所述目标文件信息。

[0111] 在一个实施例中,所述处理器在实现根据所述文件信息查询结果,确定所述目标文件信息时,用于实现:

[0112] 获取初始文件对象,将所述文件信息查询结果中的字段与每个所述字段的字段值以键值对形式添加至所述初始文件对象中,获得目标文件对象;根据所述目标文件对象,确定所述目标文件信息。

[0113] 在一个实施例中,所述处理器在实现基于预设的处理函数,获取业务数据信息时,用于实现:

[0114] 执行所述处理函数,在预设的业务数据表进行业务数据查询,获得所述业务数据信息。

[0115] 在一个实施例中,所述处理器在实现基于预设的处理函数,获取业务数据信息之前,还用于实现:

[0116] 当检测到对所述处理函数的注解添加操作时,根据所述注解添加操作对所述处理

函数添加文件组装注解以及文件属性注解。

[0117] 在一个实施例中,所述目标文件信息包括文件访问地址信息;所述处理器在实现在预设的前端页面显示所述目标业务数据之前,还用于实现:

[0118] 根据所述文件访问地址信息进行文件查询,获得对应的目标文件。

[0119] 在一个实施例中,所述处理器在实现在预设的前端页面显示所述目标业务数据时,还用于实现:

[0120] 将所述目标文件加载至所述前端页面。

[0121] 本申请的实施例中还提供一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序中包括程序指令,所述处理器执行所述程序指令,实现本申请实施例提供的任一项业务数据处理方法。

[0122] 其中,所述计算机可读存储介质可以是前述实施例所述的计算机设备的内部存储单元,例如所述计算机设备的硬盘或内存。所述计算机可读存储介质也可以是所述计算机设备的外部存储设备,例如所述计算机设备上配备的插接式硬盘,智能存储卡(Smart Media Card,SMC),安全数字卡(Secure Digital Card,SD Card),闪存卡(Flash Card)等。

[0123] 进一步地,所述计算机可读存储介质可主要包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序等;存储数据区可存储根据区块链节点的使用所创建的数据等。

[0124] 本申请所指区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的新型应用模式。区块链(Blockchain),本质上是一个去中心化的数据库,是一串使用密码学方法相关联产生的数据块,每一个数据块中包含了一批网络交易的信息,用于验证其信息的有效性(防伪)和生成下一个区块。区块链可以包括区块链底层平台、平台产品服务层以及应用服务层等。

[0125] 以上所述,仅为本申请的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到各种等效的修改或替换,这些修改或替换都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应以权利要求的保护范围为准。

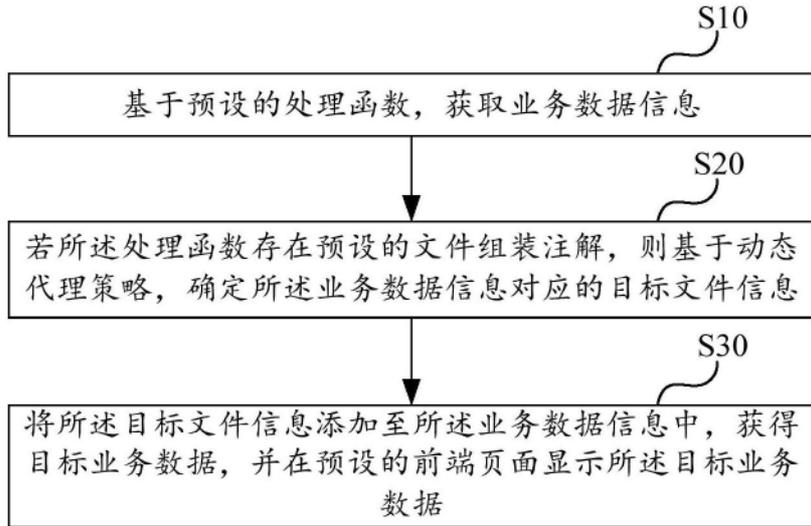


图1

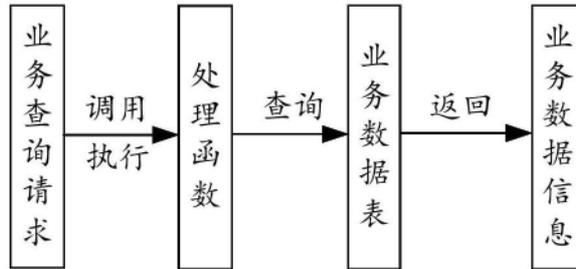


图2

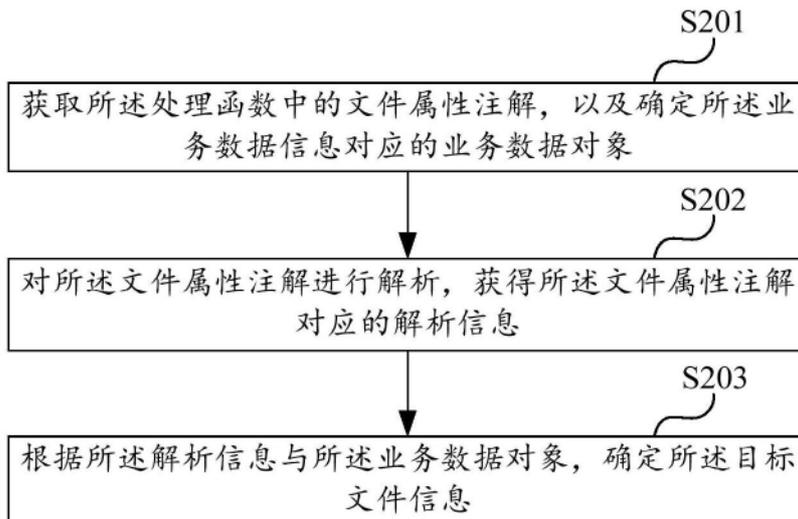


图3

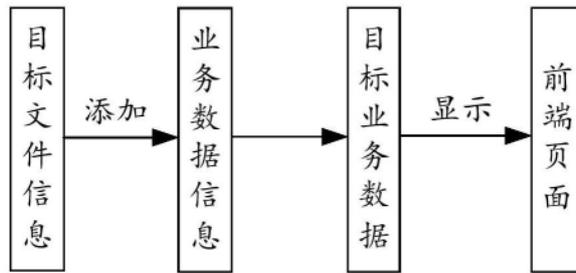


图4

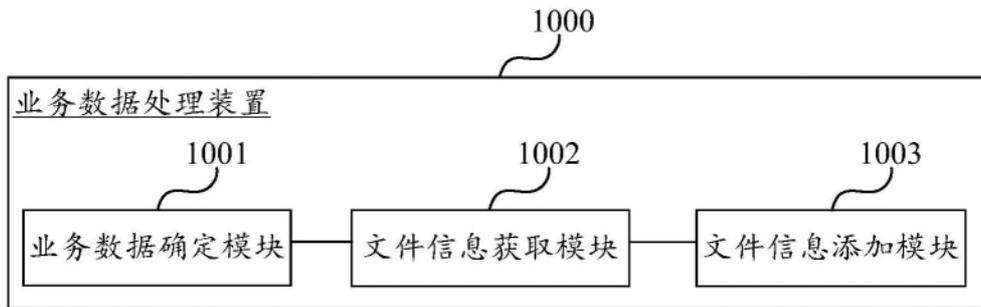


图5

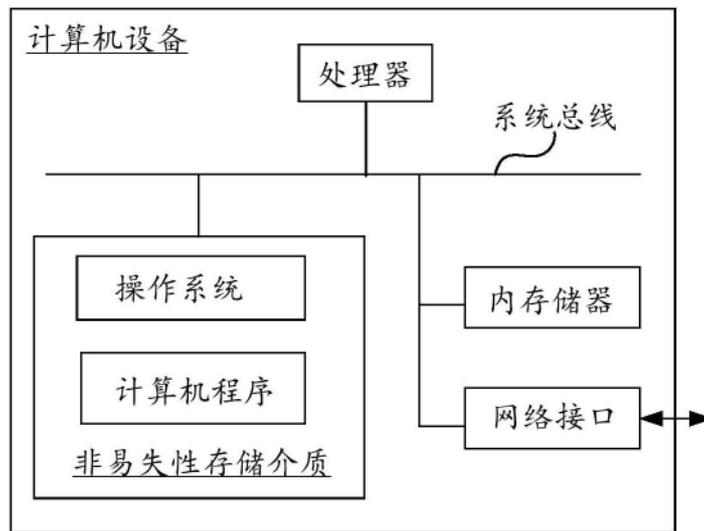


图6