



(12) 发明专利

(10) 授权公告号 CN 114020249 B

(45) 授权公告日 2024. 07. 19

(21) 申请号 202111154562.8

(22) 申请日 2021.09.29

(65) 同一申请的已公布的文献号

申请公布号 CN 114020249 A

(43) 申请公布日 2022.02.08

(73) 专利权人 北京城市网邻信息技术有限公司

地址 100015 北京市朝阳区酒仙桥北路甲

10号院101号楼1-7层内1层103室

(72) 发明人 王亮 武明礼

(74) 专利代理机构 北京润泽恒知识产权代理有

限公司 11319

专利代理师 吕俊秀

(51) Int. Cl.

G06F 8/30 (2018.01)

G06F 9/455 (2006.01)

(56) 对比文件

CN 105830025 A, 2016.08.03

CN 113420531 A, 2021.09.21

审查员 马晓彤

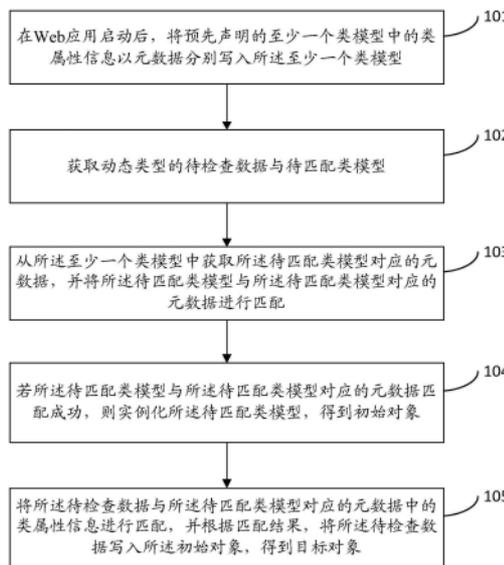
权利要求书3页 说明书10页 附图3页

(54) 发明名称

数据处理方法、装置、电子设备和存储介质

(57) 摘要

本发明提供了一种数据处理方法、装置、电子设备和存储介质,该方法包括:在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;获取动态类型的待检查数据与待匹配类模型;从至少一个类模型中获取待匹配类模型对应的元数据,并将待匹配类模型与所述待匹配类模型对应的元数据进行匹配;若所述待匹配类模型与所述待匹配类模型对应的元数据匹配成功,则实例化所述待匹配类模型,得到初始对象;将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。本发明实现了在运行时对数据的处理,可以避免运行时的数据异常。



1. 一种数据处理方法,其特征在于,包括:

在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

获取动态类型的待检查数据与待匹配类模型;

从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

若所述待匹配类模型与所述待匹配类模型对应的元数据匹配成功,则实例化所述待匹配类模型,得到初始对象;

将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

2. 根据权利要求1所述的方法,其特征在于,所述根据匹配结果,则将所述待检查数据写入所述初始对象,得到目标对象,包括:

若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配成功,则将所述待检查数据写入所述初始对象,得到目标对象;

若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配失败,则调用所述待匹配类模型对应的元数据中的转换器对所述待检查数据进行转换,并将转换后的待检查数据写入所述初始对象,得到目标对象。

3. 根据权利要求2所述的方法,其特征在于,还包括:

若所述元数据中不存在所述待匹配类模型对应的转换器,或者所述转换器执行错误,则输出第一异常提示信息。

4. 根据权利要求1所述的方法,其特征在于,在将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配之前,还包括:

创建对象代理;

将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象,包括:

通过所述对象代理将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

5. 根据权利要求1所述的方法,其特征在于,还包括:

若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出第二异常提示信息。

6. 一种数据处理方法,其特征在于,包括:

在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

获取动态类型的待检查数据与待匹配类模型;

从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出异常提示信息。

7. 一种数据处理装置,其特征在于,包括:

类模型注册模块,用于在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

数据获取模块,用于获取动态类型的待检查数据与待匹配类模型;

模型匹配模块,用于从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

实例化模块,用于若所述待匹配类模型与所述待匹配类模型对应的元数据匹配成功,则实例化所述待匹配类模型,得到初始对象;

属性匹配模块,用于将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

8. 根据权利要求7所述的装置,其特征在于,所述属性匹配模块包括:

第一数据写入单元,用于若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配成功,则将所述待检查数据写入所述初始对象,得到目标对象;

第二数据写入单元,用于若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配失败,则调用所述待匹配类模型对应的元数据中的转换器对所述待检查数据进行转换,并将转换后的待检查数据写入所述初始对象,得到目标对象。

9. 根据权利要求8所述的装置,其特征在于,所述装置还包括:

第一异常提示输出模块,用于若所述元数据中不存在所述待匹配类模型对应的转换器,或者所述转换器执行错误,则输出第一异常提示信息。

10. 根据权利要求7所述的装置,其特征在于,所述装置还包括:

创建对象代理;

所述属性匹配模块具体用于:

通过所述对象代理将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

11. 根据权利要求7所述的装置,其特征在于,所述装置还包括:

第二异常提示输出模块,用于若从所述类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出第二异常提示信息。

12. 一种数据处理装置,其特征在于,包括:

类模型注册模块,用于在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

数据获取模块,用于获取动态类型的待检查数据与待匹配类模型;

模型匹配模块,用于从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

异常处理模块,用于若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出异常提示信息。

13. 一种电子设备,其特征在于,包括:处理器、存储器及存储在所述存储器上并可在所述处理器上运行的计算机程序,所述计算机程序被所述处理器执行时实现如权利要求1至5中任一项所述的数据处理方法的步骤或者实现如权利要求6所述的数据处理方法的步骤。

14. 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如权利要求1至5中任一项所述的数据处理方法的步骤或者实现如权利要求6所述的数据处理方法的步骤。

## 数据处理方法、装置、电子设备和存储介质

### 技术领域

[0001] 本发明涉及计算机技术领域,特别是涉及一种数据处理方法、装置、电子设备和存储介质。

### 背景技术

[0002] JavaScript是一种弱类型、动态类型的语言,因此在JavaScript Web应用开发中,最常见的错误类型一般为TypeError(值类型非预期)和ReferenceError(一个不存在的变量被引用)。TypeScript为JavaScript添加了静态类型定义,使得开发者能够在运行代码之前就找到错误。

[0003] 但是,静态类型检查无法应对运行时的数据异常,而Web应用往往存在运行时的数据异常。

### 发明内容

[0004] 鉴于上述问题,提出了本发明实施例以便提供一种克服上述问题或者至少部分地解决上述问题的一种数据处理方法、装置、电子设备和存储介质。

[0005] 依据本发明的第一方面,提供了一种数据处理方法,包括:

[0006] 在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

[0007] 获取动态类型的待检查数据与待匹配类模型;

[0008] 从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

[0009] 若所述待匹配类模型与所述待匹配类模型对应的元数据匹配成功,则实例化所述待匹配类模型,得到初始对象;

[0010] 将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

[0011] 依据本发明的第二方面,提供了一种数据处理方法,包括:

[0012] 在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

[0013] 获取动态类型的待检查数据与待匹配类模型;

[0014] 从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

[0015] 若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出异常提示信息。

[0016] 依据本发明的第三方面,提供了一种数据处理装置,包括:

[0017] 类模型注册模块,用于在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

- [0018] 数据获取模块,用于获取动态类型的待检查数据与待匹配类模型;
- [0019] 模型匹配模块,用于从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;
- [0020] 实例化模块,用于若所述待匹配类模型与所述待匹配类模型对应的元数据匹配成功,则实例化所述待匹配类模型,得到初始对象;
- [0021] 属性匹配模块,用于将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。
- [0022] 依据本发明的第四方面,提供了一种数据处理装置,包括:
- [0023] 类模型注册模块,用于在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;
- [0024] 数据获取模块,用于获取动态类型的待检查数据与待匹配类模型;
- [0025] 模型匹配模块,用于从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;
- [0026] 异常处理模块,用于若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出异常提示信息。
- [0027] 依据本发明的第五方面,还提供了一种电子设备,包括:处理器、存储器及存储在所述存储器上并可在所述处理器上运行的计算机程序,所述计算机程序被所述处理器执行时实现如第一方面或第二方面所述的数据处理方法的步骤。
- [0028] 依据本发明的第六方面,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如第一方面或第二方面所述的数据处理方法的步骤。
- [0029] 本发明提供的数据处理方法、装置、电子设备和存储介质,通过在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入至少一个类模型,获取到动态类型的待检查数据与待匹配类模型后,从至少一个类模型中获取待匹配类模型对应的元数据,并将待匹配类模型与待匹配类模型对应的元数据进行匹配,若匹配成功则实例化待匹配类模型,得到初始对象,将待检查数据与待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将待检查数据写入初始对象,得到目标对象,由于在运行时将编码时声明的类属性信息和类以元数据写入对应的类模型,并基于该类模型对待检查数据对应的待匹配类模型和待检查数据进行检查,实现了在运行时对数据的处理,可以避免运行时的数据异常。
- [0030] 上述说明仅是本发明技术方案的概述,为了能够更清楚了解本发明的技术手段,而可依照说明书的内容予以实施,并且为了让本发明的上述和其它目的、特征和优点能够更明显易懂,以下特举本发明的具体实施方式。

## 附图说明

- [0031] 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,而并不认为是对本发明

的限制。

[0032] 图1是本发明实施例提供的一种数据处理方法的步骤流程图；

[0033] 图2是本发明实施例提供的一种数据处理方法的步骤流程图；

[0034] 图3是本发明实施例提供的一种数据处理装置的结构框图；

[0035] 图4是本发明实施例提供的一种数据处理装置的结构框图。

### 具体实施方式

[0036] 下面将参照附图更详细地描述本发明的示例性实施例。虽然附图中显示了本发明的示例性实施例，然而应当理解，可以以各种形式实现本发明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发明，并且能够将本发明的范围完整的传达给本领域的技术人员。

[0037] 图1是本发明实施例提供的一种数据处理方法的步骤流程图，可以由服务器执行，如图1所示，该方法可以包括：

[0038] 步骤101，在Web应用启动后，将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型。

[0039] 其中，元数据(Metadata)是一种用于描述数据的数据。可以用于描述数据的属性、组织、关系等信息。如HTML的<meta>标签就是典型的元数据。类模型是通过装饰器声明的类，可以包括类模型标识(可以是类的名称)和具体的类模型数据，类模型数据可以包括类和类属性信息，类属性信息也是通过装饰器声明的。

[0040] 需要对Web应用中的动态类型数据进行检查和转换等处理时，可以预先在Web应用对应的代码中对要检查的数据所对应的类和类属性信息进行声明，实现对类模型的声明，可以通过声明装饰器来快速声明类(Class)模型，声明类属性信息以及转换器，例如，可以通过装饰器@Model声明类，并通过装饰器@Member声明类属性信息和转换器。其中，类属性信息可以是类属性类型。

[0041] 在Web应用启动后，可以将预先声明的至少一个类模型中的类属性信息以元数据分别写入至少一个类模型中，即以元数据的形式来表示每个类模型，实现对所述至少一个类模型的注册。其中，Web应用启动对于网页来说是在打开页面时启动Web应用，而对于Web服务来说是对Web服务进行初始化时启动Web应用。

[0042] 步骤102，获取动态类型的待检查数据与待匹配类模型。

[0043] 其中，动态类型是在代码运行时才会确认数据类型，而不是在编译时。在JavaScript中可以在运行时改变对象属性的类型，或者为对象增加一个新属性。待检查数据一般为json数据。待匹配类模型为与待检查数据对应的类模型，可能在所述至少一个类模型中存在与待匹配类模型相同类模型标识的类模型，也可能不存在，具体是否存在可以通过在至少一个类模型中查找是否包括待匹配类模型的类模型标识来确定。

[0044] 可以根据指定的数据接口获取动态类型的待检查数据和对应的待匹配类模型，或者，也可以人为指定动态类型的待检查数据并给出对应的待匹配类模型。所述待匹配类模型以元数据的形式表示。

[0045] 步骤103，从所述至少一个类模型中获取所述待匹配类模型对应的元数据，并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配。

[0046] 其中,待匹配类模型对应的元数据是在所述至少一个类模型中与待匹配类模型的类模型标识相同的类模型所具有的元数据。

[0047] 在预先注册的至少一个类模型中查找是否存在与待匹配类模型的类模型标识相同的类模型,如果存在则从至少一个类模型中获取与待匹配类模型的类模型标识相同的类模型的元数据,将获取到的元数据确定为待匹配类模型对应的元数据,即得到待匹配类模型对应的注册的元数据,将待匹配类模型本身所具有的元数据与待匹配类模型对应的注册的元数据进行匹配,以确定待匹配类模型与待匹配类模型对应的元数据是否完全一致,在匹配过程中,如果待匹配类模型中的一个类属性为其他类,则按照类和类属性逐级进行匹配。例如,待匹配类模型是由两个类模型递归所组成的模型,这两个类模型分别为Class A和Class B,而Class A的类属性信息为Class B,Class B的类属性信息为具体的值,在将待匹配类模型与元数据进行匹配时,首先将待匹配类模型中的Class A与元数据中的Class A进行匹配,若两者一致,则匹配待匹配类模型中Class A的类属性信息Class B与元数据中Class A的类属性信息,如果两者一致,则将待匹配类模型中Class B的类属性信息与元数据中Class B的类属性信息进行匹配,得到最终的匹配结果,在任意次匹配中如果匹配不一致,则确定匹配失败,只有所有数据均匹配一致时才确定匹配成功。上述仅为举例,待匹配类模型不限于由两个类模型递归组成,还可以包括更多级的递归,匹配时只需逐级进行匹配。

[0048] 步骤104,若所述待匹配类模型与所述待匹配类模型对应的元数据匹配成功,则实例化所述待匹配类模型,得到初始对象。

[0049] 如果待匹配类模型与待匹配类模型对应的元数据完全一致,则确定匹配成功,则可以对待匹配类模型进行实例化,得到的实例实体为初始对象,该初始对象为一个空的对象,可以将后续的待检查数据写入初始对象,以得到目标对象。

[0050] 步骤105,将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

[0051] 由于待检查数据的数据类型不一定符合要求,所以需要对待检查数据进行检查。待检查数据可能会有多条数据,这时可以遍历每条待检查数据,并将当前遍历到的待检查数据与元数据中该待检查数据所对应的类属性信息进行匹配,根据匹配结果,将待检查数据写入初始对象,如果匹配结果为匹配成功则可以直接将待检查数据写入初始对象,如果匹配结果为匹配失败则可以对待检查数据进行转换后写入初始对象,将所有的待检查数据写入初始对象后,得到目标对象。在得到目标对象后,可以将目标对象返回至调用方,即将目标对象返回至Web应用。例如,在待检查数据为用英语表示的数字,而元数据中待检查数据对应的类属性信息为阿拉伯数字时,则待检查数据与类属性信息两者不一致,即匹配结果为匹配失败,这时可以将待检查数据的英语数字转换为阿拉伯数字,并将转换后的待检查数据写入初始对象,得到目标对象。

[0052] 在本发明的一个实施例中,所述根据匹配结果,则将所述待检查数据写入所述初始对象,得到目标对象,包括:

[0053] 若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配成功,则将所述待检查数据写入所述初始对象,得到目标对象;

[0054] 若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配失败,

则调用所述待匹配类模型对应的元数据中的转换器对所述待检查数据进行转换,并将转换后的待检查数据写入所述初始对象,得到目标对象。

[0055] 在将预先声明的至少一个类模型中的类属性信息以元数据分别写入至少一个类模型时,还可以将类模型中的可选转换器以元数据写入类模型。所以待匹配类模型对应的元数据中可能会包括转换器,转换器可以将匹配失败的待检查数据转换为符合要求的数据。

[0056] 在待检查数据与待匹配类模型对应的元数据中的类属性信息匹配成功时,表明待检查数据的格式或类型与预先设置的格式或类型相同,这时可以直接将待检查数据写入初始对象,得到目标对象。在待检查数据与待匹配类模型对应的元数据中的类属性信息匹配失败时,表明待检查数据的格式或类型与预先设置的格式或类型不同,这时可以调用待匹配类模型对应的元数据中的转换器来对待检查数据进行转换,以将待检查数据的属性信息转换为与元数据中的类属性信息相同的数据,并将转换后的待检查数据写入初始对象,得到目标对象。

[0057] 在有多条待检查数据时,分别对每条待检查数据进行类属性信息的匹配,在所有的待检查数据直接写入初始对象或者转换后写入初始对象后,得到目标对象。

[0058] 通过在待检查数据与待匹配类模型对应的元数据中的类属性信息匹配成功时,将待检查数据写入初始对象,在待检查数据与待匹配类模型对应的元数据中的类属性信息匹配不成功时,调用转换器将待检查数据转换为所述类属性信息的数据后写入初始对象,实现了在运行时对数据的检查和转换,可以满足在复杂业务场景下对数据格式的处理,提升Web应用的健壮性。

[0059] 在本发明的一个实施例中,所述方法还包括:若所述元数据中不存在所述待匹配类模型对应的转换器,或者所述转换器执行错误,则输出第一异常提示信息。

[0060] 在元数据中不存在待匹配类模型对应的转换器或者转换器在对待检查数据进行转换的过程中执行错误时,生成对应的第一异常提示信息,并输出第一异常提示信息,如可以将第一异常提示信息输出至Web应用,这样可以使得用户了解到数据的异常信息。

[0061] 本实施例提供的数据处理方法,通过在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入至少一个类模型,获取到动态类型的待检查数据与待匹配类模型后,从至少一个类模型中获取待匹配类模型对应的元数据,并将待匹配类模型与待匹配类模型对应的元数据进行匹配,若匹配成功则实例化待匹配类模型,得到初始对象,将待检查数据与待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将待检查数据写入初始对象,得到目标对象,由于在运行时将编码时声明的类属性信息和类以元数据写入对应的类模型,并基于该类模型对待检查数据对应的待匹配类模型和待检查数据进行检查,实现了在运行时对数据的处理,可以避免运行时的数据异常。

[0062] 在上述技术方案的基础上,在将所述待检查数据与所述待匹配模型对应的元数据中的类属性信息进行匹配之前,还包括:创建对象代理;

[0063] 将所述待检查数据与所述待匹配模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象,包括:通过所述对象代理将所述待检查数据与所述待匹配模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

[0064] 其中,对象代理(Object Proxy)用于拦截对于对象的操作,如属性赋值、属性读取等,在JavaScript中还可以进行自定义操作。

[0065] 在对待匹配类模型进行实例化得到初始对象后,将待检查数据与待匹配类模型对应的元数据中的类属性信息进行匹配之前,创建对象代理,之后可以通过对象代理来对待检查数据进行检查和转换,以及对初始对象的读写,即通过对象代理拦截将待检查数据直接写入初始对象的操作,并在拦截后,通过对象代理将待检查数据与待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果将待检查数据写入初始对象,得到目标对象。通过使用对象代理来代理对待检查数据的检查和转换,实现了在运行时对数据的检查和转换。

[0066] 在上述技术方案的基础上,所述方法还包括:若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出第二异常提示信息。

[0067] 在至少一个类模型中无法获取到待匹配类模型对应的元数据,也就是在至少一个类模型中没有与待匹配类模型的类模型标识相同的类模型,或者待匹配类模型与待匹配类模型对应的元数据匹配失败时,生成对应的第二异常提示信息,并输出第二异常提示信息,如可以将第二异常提示信息输出至Web应用,这样可以使得用户了解到数据的异常信息。其中,待匹配类模型与待匹配类模型对应的元数据匹配失败即待匹配类模型与待匹配类模型对应的元数据匹配不一致。

[0068] 本发明实施例在编码时通过声明类(Class)时为类属性(Property)添加类型(Type)描述的元数据实现类属性类型的注册。在运行时创建编码时所声明的类的对象实体,通过Object Proxy代理该对象属性的读写操作,进而获取相应的类型声明与转换方案,从而在运行时实现类型检查与转换。通过声明即可实现对数据的检查与转换,无需手动检查数据;实现了声明复用,声明的类模型可以实现TypeScript中interface、type的功能;该方案是一个轻量级的解决方案,没有繁杂的API,编写和测试代码都变得轻松,代码也更容易阅读、易于维护;分离了业务逻辑:分离了数据校验、数据转换逻辑与Web应用的业务逻辑,功能更聚焦。

[0069] 需要说明的是,对于方法实施例,为了简单描述,故将其都表述为一系列的动作组合,但是本领域技术人员应该知悉,本发明实施例并不受所描述的动作顺序的限制,因为依据本发明实施例,某些步骤可以采用其他顺序或者同时进行。其次,本领域技术人员也应该知悉,说明书中所描述的实施例均属于优选实施例,所涉及的动作并不一定是本发明实施例所必须的。

[0070] 图2是本发明实施例提供的一种数据处理方法的步骤流程图,可以由服务器执行,如图2所示,该方法可以包括:

[0071] 步骤201,在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型。

[0072] 步骤202,获取动态类型的待检查数据与待匹配类模型。

[0073] 步骤203,从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配。

[0074] 其中,步骤201-步骤203与上述实施例中的步骤101-步骤103的具体内容相同,这

里不再赘述。

[0075] 步骤204,若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出异常提示信息。

[0076] 在至少一个类模型中无法获取到待匹配类模型对应的元数据,也就是在至少一个类模型中没有与待匹配类模型的类模型标识相同的类模型,或者待匹配类模型与待匹配类模型对应的元数据匹配失败时,生成对应的异常提示信息,并输出该异常提示信息,如可以将异常提示信息输出至Web应用,这样可以使得用户了解到数据的异常信息。其中,待匹配类模型与待匹配类模型对应的元数据匹配失败即待匹配类模型与待匹配类模型对应的元数据匹配不一致。

[0077] 本实施例提供的数据处理方法,通过在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入至少一个类模型,获取到动态类型的待检查数据与待匹配类模型后,从至少一个类模型中获取待匹配类模型对应的元数据,并将待匹配类模型与待匹配类模型对应的元数据进行匹配,若从至少一个类模型中无法获取待匹配类模型对应的元数据,或者待匹配类模型与待匹配类模型对应的元数据匹配失败,则输出异常提示信息,由于在运行时将编码时声明的类属性信息和类以元数据写入对应的类模型,并基于该类模型对待检查数据对应的待匹配类模型进行检查,实现了在运行时对数据的处理,而且在发现异常时可以及时提示。

[0078] 图3是本发明实施例提供的一种数据处理装置的结构框图,如图3所示,该数据处理装置可以包括:

[0079] 类模型注册模块301,用于在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

[0080] 数据获取模块302,用于获取动态类型的待检查数据与待匹配类模型;

[0081] 模型匹配模块303,用于从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

[0082] 实例化模块304,用于若所述待匹配类模型与所述待匹配类模型对应的元数据匹配成功,则实例化所述待匹配类模型,得到初始对象;

[0083] 属性匹配模块305,用于将所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

[0084] 可选的,所述属性匹配模块包括:

[0085] 第一数据写入单元,用于若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配成功,则将所述待检查数据写入所述初始对象,得到目标对象;

[0086] 第二数据写入单元,用于若所述待检查数据与所述待匹配类模型对应的元数据中的类属性信息匹配失败,则调用所述待匹配类模型对应的元数据中的转换器对所述待检查数据进行转换,并将转换后的待检查数据写入所述初始对象,得到目标对象。

[0087] 可选的,所述装置还包括:

[0088] 第一异常提示输出模块,用于若所述元数据中不存在所述待匹配类模型对应的转换器,或者所述转换器执行错误,则输出第一异常提示信息。

[0089] 可选的,所述装置还包括:

[0090] 创建对象代理;

[0091] 所述属性匹配模块具体用于:

[0092] 通过所述对象代理将所述待检查数据与所述待匹配模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将所述待检查数据写入所述初始对象,得到目标对象。

[0093] 可选的,所述装置还包括:

[0094] 第二异常提示输出模块,用于若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出第二异常提示信息。

[0095] 本实施例提供的数据处理装置,通过在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入至少一个类模型,获取到动态类型的待检查数据与待匹配类模型后,从至少一个类模型中获取待匹配类模型对应的元数据,并将待匹配类模型与待匹配类模型对应的元数据进行匹配,若匹配成功则实例化待匹配类模型,得到初始对象,将待检查数据与待匹配类模型对应的元数据中的类属性信息进行匹配,并根据匹配结果,将待检查数据写入初始对象,得到目标对象,由于在运行时将编码时声明的类属性信息和类以元数据写入对应的类模型,并基于该类模型对待检查数据对应的待匹配类模型和待检查数据进行检查,实现了在运行时对数据的处理,可以避免运行时的数据异常。

[0096] 图4是本发明实施例提供的一种数据处理装置的结构框图,如图4所示,该数据处理装置可以包括:

[0097] 类模型注册模块401,用于在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入所述至少一个类模型;

[0098] 数据获取模块402,用于获取动态类型的待检查数据与待匹配类模型;

[0099] 模型匹配模块403,用于从所述至少一个类模型中获取所述待匹配类模型对应的元数据,并将所述待匹配类模型与所述待匹配类模型对应的元数据进行匹配;

[0100] 异常处理模块404,用于若从所述至少一个类模型中无法获取所述待匹配类模型对应的元数据,或者所述待匹配类模型与所述待匹配类模型对应的元数据匹配失败,则输出异常提示信息。

[0101] 本实施例提供的数据处理装置,通过在Web应用启动后,将预先声明的至少一个类模型中的类属性信息以元数据分别写入至少一个类模型,获取到动态类型的待检查数据与待匹配类模型后,从至少一个类模型中获取待匹配类模型对应的元数据,并将待匹配类模型与待匹配类模型对应的元数据进行匹配,若从至少一个类模型中无法获取待匹配类模型对应的元数据,或者待匹配类模型与待匹配类模型对应的元数据匹配失败,则输出异常提示信息,由于在运行时将编码时声明的类属性信息和类以元数据写入对应的类模型,并基于该类模型对待检查数据对应的待匹配类模型进行检查,实现了在运行时对数据的处理,而且在发现异常时可以及时提示。

[0102] 优选的,本发明实施例还提供了一种电子设备,包括:处理器,存储器,存储在存储器上并可在处理器上运行的计算机程序,该计算机程序被处理器执行时实现上述数据处理方法实施例的各个过程,且能达到相同的技术效果,为避免重复,这里不再赘述。

[0103] 本发明实施例还提供了一种计算机可读存储介质,计算机可读存储介质上存储有

计算机程序,计算机程序被处理器执行时实现上述数据处理方法实施例的各个过程,且能达到相同的技术效果,为避免重复,这里不再赘述。其中,所述的计算机可读存储介质,如只读存储器(Read-Only Memory,简称ROM)、随机存取存储器(Random Access Memory,简称RAM)、磁碟或者光盘等。

[0104] 本说明书中的各个实施例均采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似的部分互相参见即可。

[0105] 需要说明的是,在本文中,术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

[0106] 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上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端(可以是手机,计算机,服务器,空调器,或者网络设备等)执行本发明各个实施例所述的方法。

[0107] 上面结合附图对本发明的实施例进行了描述,但是本发明并不局限于上述的具体实施方式,上述的具体实施方式仅仅是示意性的,而不是限制性的,本领域的普通技术人员在本发明的启示下,在不脱离本发明宗旨和权利要求所保护的范围情况下,还可做出很多形式,均属于本发明的保护之内。

[0108] 本领域普通技术人员可以意识到,结合本发明实施例中公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本发明的范围。

[0109] 所属领域的技术人员可以清楚地了解到,为描述的方便和简洁,上述描述的系统、装置和单元的具体工作过程,可以参考前述方法实施例中的对应过程,在此不再赘述。

[0110] 在本申请所提供的实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0111] 所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0112] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以

是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。

[0113] 所述功能如果以软件功能单元的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,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,或者网络设备等)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、ROM、RAM、磁碟或者光盘等各种可以存储程序代码的介质。

[0114] 以上所述,仅为本发明的具体实施方式,但本发明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发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求的保护范围为准。

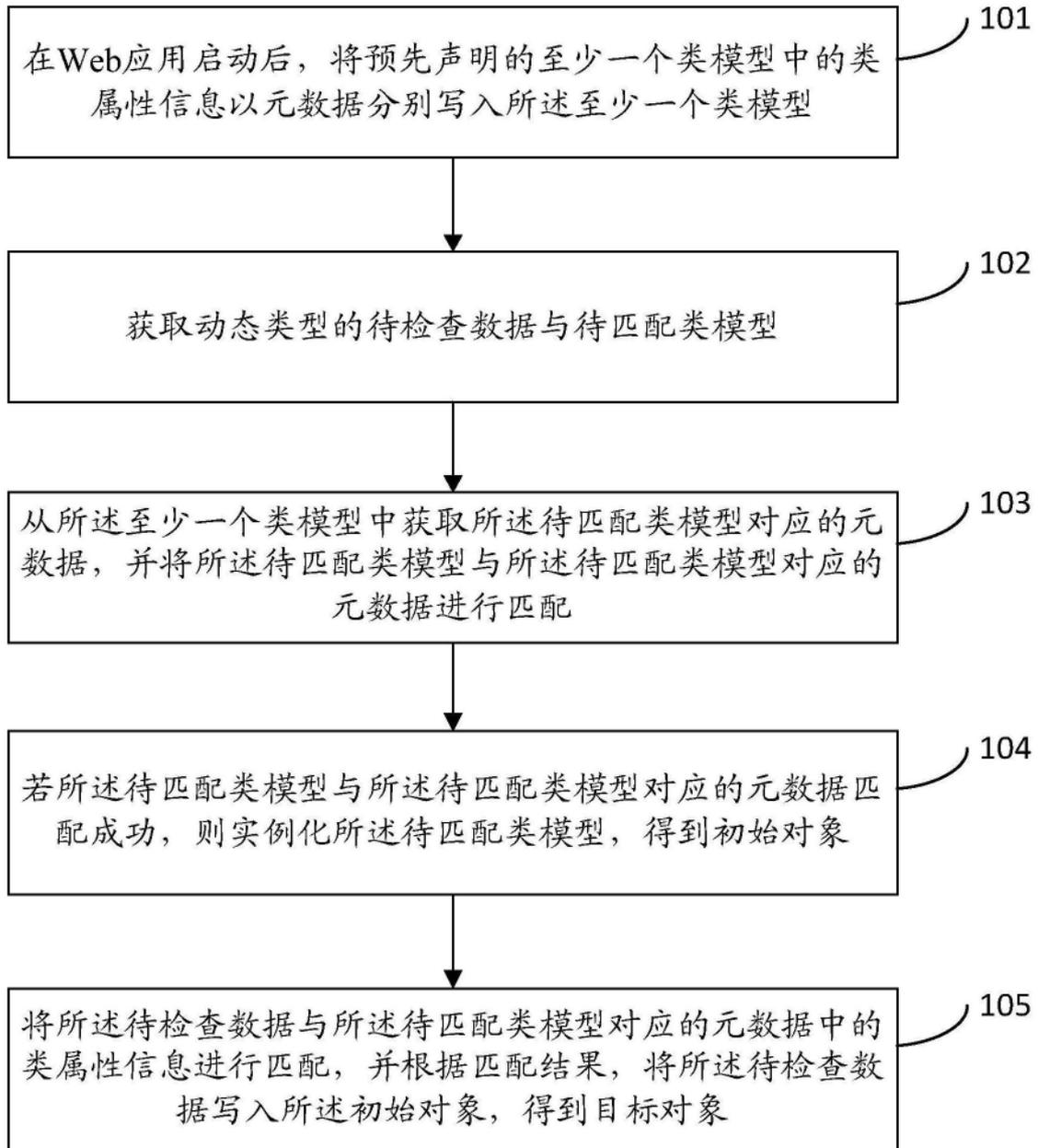


图1

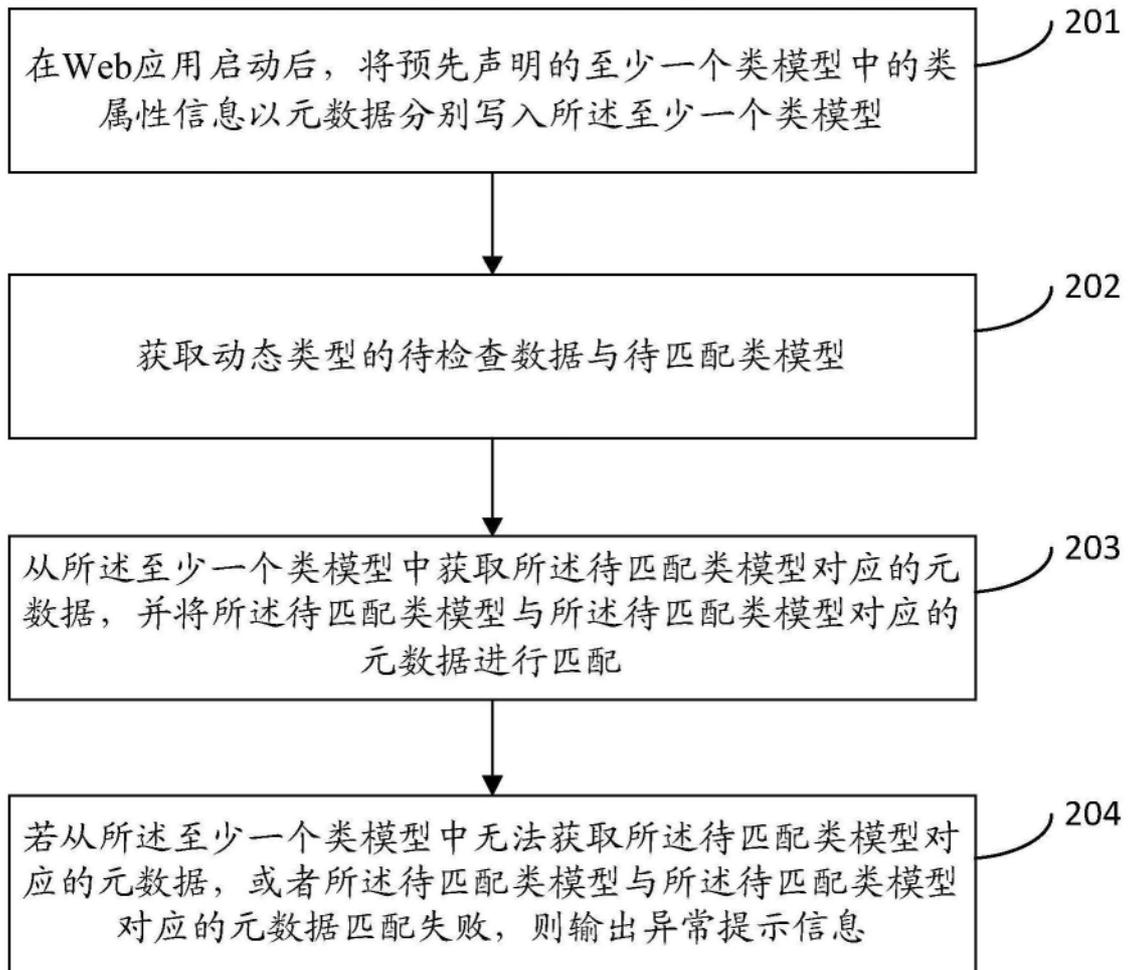


图2

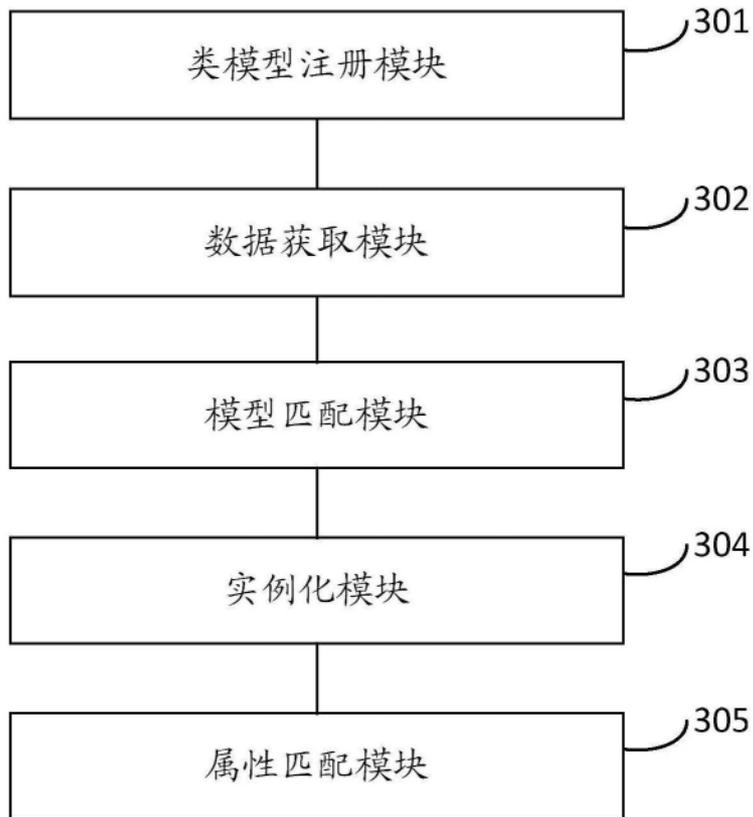


图3

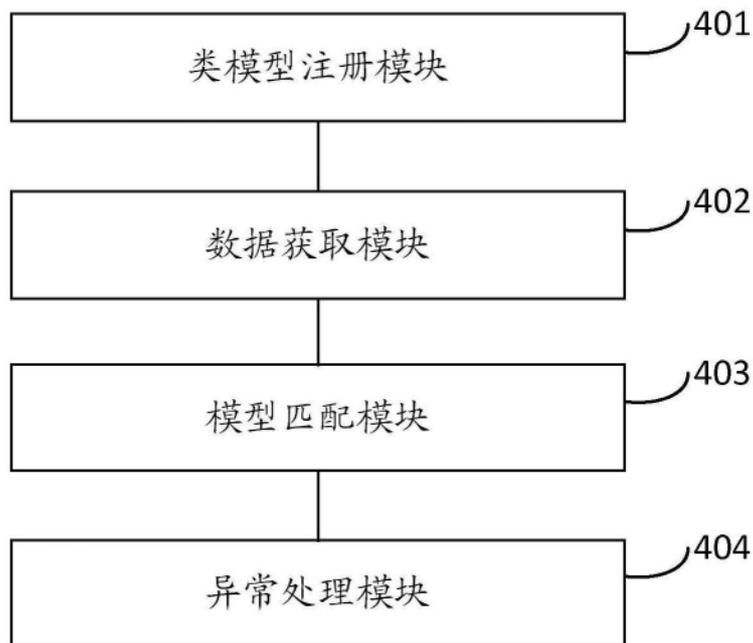


图4