(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国 际 局

(43) 国际公布日
2020 年 10 月 1 日 (01.10.2020)

WIPO | PCT

(10) 国际公布号
WO 2020/192134 A1

(51) 国际专利分类号:
G06F 9/448 (2018.01)

(21) 国际申请号: PCT/CN2019/117506

(22) 国际申请日: 2019 年 11 月 12 日 (12.11.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201910243158.4 2019年3月28日 (28.03.2019) CN

(71) 申请人: 平安科技（深圳）有限公司 (PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) [CN/CN]; 中国广东省深圳市福田区福田街道福安社区益田路5033号平安金融中心23楼, Guangdong 518033 (CN)。

(72) 发明人: 许永夫 (XU, Yongfu); 中国广东省深圳市福田区福田街道福安社区益田路5033号平安金融中心23楼, Guangdong 518033 (CN)。

(74) 代理人: 广州华进联合专利商标代理有限公司 (ADVANCE CHINA IP LAW OFFICE); 中国广东省广州市天河区珠江东路6号4501房（部位：自编01-03和08-12单元）（仅限办公用途）, Guangdong 510623 (CN)。

(81) 指定国(除另有指明，要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL,

(54) **Title:** EXCEPTION HANDLING METHOD AND APPARATUS BASED ON SPRING MVC, AND COMPUTER DEVICE AND STORAGE MEDIUM

(54) **发明名称**: 基于Spring MVC的异常处理方法、装置、计算机设备和存储介质

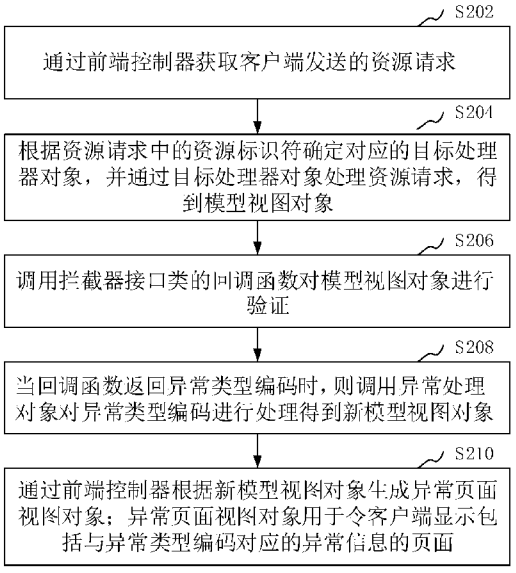


图 2

(57) **Abstract:** An exception handling method based on Spring MVC, the method comprising: acquiring, by means of a front-end controller, a resource request sent by a client; determining a corresponding target handler object according to a resource identifier in the resource request, and handling the resource request by means of the target handler object to obtain a model view object; calling a callback function of an interceptor interface class to verify the model view object; when the callback function returns an exception type code, calling an exception handling object to handle the exception type code to obtain a new model view object; and generating an exception page view object according to the new model view object by means of the front-end controller, wherein the exception page view object is used for enabling the client to display a page comprising exception information corresponding to the exception type code.

WO 2020/192134 A1

[见续页]

PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(84) 指定国(除另有指明，要求每一种可提供的地区保护)：ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布：

— 包括国际检索报告(条约第21条(3))。

(57) 摘要：一种基于Spring MVC的异常处理方法，包括：通过前端控制器获取客户端发送的资源请求；根据资源请求中的资源标识符确定对应的目标处理器对象，并通过目标处理器对象处理资源请求，得到模型视图对象；调用拦截器接口类的回调函数对模型视图对象进行验证；当回调函数返回异常类型编码时，则调用异常处理对象对异常类型编码进行处理得到新模型视图对象；通过前端控制器根据新模型视图对象生成异常页面视图对象；异常页面视图对象用于令客户端显示包括与异常类型编码对应的异常信息的页面。

基于 Spring MVC 的异常处理方法、装置、计算机设备和存储介质

本申请要求于 2019 年 03 月 28 日提交中国专利局，申请号为 201910243158.4 ，申请名称为“基于 Spring MVC 的异常处理方法、装置、计算机设备和存储介质”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及一种基于 Spring MVC 的异常处理方法、装置、计算机设备和存储介质。

背景技术

Spring MVC 框架是围绕一个 DispatcherServlet 接口类来设计的，这个接口类实现前端控制器的基本功能，会把接收到的数据请求分发给各个处理器对象（Handler），并支持可配置的处理器映射、视图渲染、本地化。DispatcherServlet 接口类是通过各种配置文件实现的，在这些配置文件中定义各种实现类，就可以实现前端控制器的基本功能。

然而，发明人意识到，现有的对 DispatcherServlet 接口类配置较为混乱，导致前置控制器只能处理简单的异常信息。如何利用 Spring MVC 框架实现对不同业务场景下返回的各种复杂的异常信息进行统一处理，是一个需要解决的问题。

发明内容

根据本申请公开的各种实施例，提供一种基于 Spring MVC 的异常处理方法、装置、计算机设备和存储介质。

一种基于 Spring MVC 的异常处理方法包括：

通过前端控制器获取客户端发送的资源请求；

根据所述资源请求中的资源标识符确定对应的目标处理器对象；并通过所述目标处理器对象处理所述资源请求，得到模型视图对象；

调用拦截器接口类的回调函数对所述模型视图对象进行验证；

当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行

处理得到新模型视图对象；及

通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

一种基于 Spring MVC 的异常处理装置包括：

资源请求获取模块，用于通过前端控制器获取客户端发送的资源请求；

目标处理器对象获取模块，用于根据所述资源请求中的资源标识符确定对应的目标处理器对象，并通过所述目标处理器对象处理所述资源请求，得到模型视图对象；

回调处理模块，用于调用拦截器接口类的回调函数对所述模型视图对象进行验证；及

异常处理模块，用于当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象；通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

一种计算机设备，包括存储器和一个或多个处理器，所述存储器中储存有计算机可读指令，所述计算机可读指令被所述处理器执行时，使得所述一个或多个处理器执行以下步骤：

通过前端控制器获取客户端发送的资源请求；

根据所述资源请求中的资源标识符确定对应的目标处理器对象，并通过所述目标处理器对象处理所述资源请求，得到模型视图对象；

调用拦截器接口类的回调函数对所述模型视图对象进行验证；

当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象；及

通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

一个或多个存储有计算机可读指令的非易失性计算机可读存储介质，计算机可读指令被一个或多个处理器执行时，使得一个或多个处理器执行以下步骤：

通过前端控制器获取客户端发送的资源请求；

根据所述资源请求中的资源标识符确定对应的目标处理器对象，并通过所述目标处理

器对象处理所述资源请求，得到模型视图对象；

调用拦截器接口类的回调函数对所述模型视图对象进行验证；

当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象；及

通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

本申请的一个或多个实施例的细节在下面的附图和描述中提出。本申请的其它特征和优点将从说明书、附图以及权利要求书变得明显。

附图说明

为了更清楚地说明本申请实施例中的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅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它的附图。

图 1 为根据一个或多个实施例中基于 Spring MVC 的异常处理方法的应用场景图。

图 2 为根据一个或多个实施例中基于 Spring MVC 的异常处理方法的流程示意图。

图 3 为根据一个或多个实施例中基于 Spring MVC 的异常处理方法的框架示意图。

图 4 为根据一个或多个实施例中基于 Spring MVC 的异常处理装置的框图。

图 5 为根据一个或多个实施例中计算机设备的框图。

具体实施方式

为了使本申请的技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处描述的具体实施例仅仅用以解释本申请，并不用于限定本申请。

本申请提供的基于 Spring MVC 的异常处理方法，可以应用于如图 1 所示的应用环境中。终端 102 与网页服务器 104 通过网络进行通信。终端 102 获取用户在浏览器中触发的资源请求，并通过网络将该资源请求发送至网页服务器 104，网页服务器 104 通过前端控制器获取客户端发送的资源请求，并根据该资源请求中的资源标识符确定对应的目标处理

器对象，通过获取的目标处理器对象处理该资源请求，得到模型视图对象；然后调用拦截器接口类的回调函数对模型视图对象进行验证；当回调函数返回异常类型编码时，则前端控制器可以调用异常处理对象对异常类型编码进行处理得到新模型视图对象；前端控制器根据新模型视图对象生成异常页面视图对象；当客户端接收到该异常页面视图对象，就可以渲染显示包括与异常类型编码对应的异常信息的页面。终端 102 可以但不限于是各种个人计算机、笔记本电脑、智能手机、平板电脑和便携式可穿戴设备，网页服务器 104 可以用独立的服务器或者是多个服务器组成的服务器集群来实现。

在其中一个实施例中，如图 2 所示，提供了一种 Spring MVC 的异常处理方法，以该方法应用于图 1 中的网页服务器 104 为例进行说明，包括以下步骤：

步骤 S202，通过前端控制器获取客户端发送的资源请求。

资源请求是客户端根据当前获取的资源标识符（URL）生成的 HTTP 报文。资源标识符用于唯一标识要请求的资源，当用户访问一个网站时，客户端根据访问的网站对应的 URL 生成一个资源请求，并通过网络将该资源请求发送至网页服务器。在网页服务器接收到该资源请求后，对该资源请求中的资源标识符进行检查，如果发现是静态资源的请求，就将该请求转由网页服务器默认的服务端程序（Default-servlet-handler）处理，得到与该资源请求对应的目标数据，根据目标数据生成与资源请求对应的目标资源载体对象并返回至客户端。如果不是静态资源的请求，则根据配置文件匹配与该资源标识符对应的服务端程序，将该资源请求转发给该服务端程序的前端控制器，由前端控制器接收该资源请求后，调用该服务端程序所包括的其它组件处理该资源请求。

例如，如果客户端请求的是静态资源（*.htm，后缀名为*.htm 的文件），则将请求直接转交给网页服务器，之后网页服务器从文件系统中取出内容，发送回客户端浏览器进行解析执行。如果客户端请求的是动态资源（*.jsp、*.asp、*.aspx、*.php），则先将请求转交给服务端程序的前端控制器，经业务层处理后从数据库中取出数据后动态拼凑页面的展示内容，再将所有的展示内容返回给网页服务器，通过网页服务器将展示内容返回至客户端浏览器进行解析、渲染并显示。

前端控制器是预先自定义的继承自 Spring MVC 框架中的 SpringDispatcherServlet 接口类的实现类，前端控制器比如可以是 WebSupportDispatcherServlet，它是整个流程

控制的中心，控制其它接口类的调度和执行，它把用户的资源请求中的URL对应的数据经过业务处理层定义的处理器处理之后封装成MVC架构中的模型（Model），然后再将该模型（Model）返回给对应的界面（View）进行展示。实现类是接口类的具体实现方法，比如某个接口类有3个实现类A，B，C，分别为不同类型的实现类，通过确定被接口类引用的对象是类型A的实现类，就可以在调用接口类中的方法时调用类型A的实现类中方法进行处理。

步骤 S204，根据资源请求中的资源标识符确定对应的目标处理器对象，并通过目标处理器对象处理资源请求，得到模型视图对象。

处理器对象是业务处理层定义的用于资源请求进行处理的对象，包括多个处理函数，也叫处理方法，通过该处理器对象中的处理函数，对资源请求进行处理，返回处理结果。前端控制器可以根据该资源请求中的URL查找与该资源请求对应的处理器映射器，处理器映射器就可以根据配置文件或注解把请求映射成目标处理器对象。

模型视图对象（ModelAndView）用来存储处理完后的结果数据以及用于显示该结果数据的视图信息。Model代表模型，View代表视图，目标处理器对象处理完资源请求后，将结果数据存储在该模型视图对象的模型属性中，将要返回的视图信息存储在该模型视图对象的视图属性中，然后将模型视图对象返回至前端控制器，前端控制器可以调用配置文件中定义的视图解析器，对该模型视图对象进行解析，将解析得到的结果数据显示在视图信息所指定的页面上。

在其中一个实施例中，根据资源请求中的资源标识符确定对应的目标处理器对象包括：从资源请求中提取资源标识符；通过前端控制器根据资源标识符调用相应的映射器对象；通过映射器对象获取与资源请求对应的处理器对象；通过前端控制器查找与处理器对象对应的处理器适配器；通过处理器适配器从处理器对象中确定目标处理器对象。

具体地，可以预先自定义继承自 Spring MVC 框架中的默认注解控制器映射处理器（DefaultAnnotationHandlerMapping）接口类的实现类，也就是自定义的映射器对象（WebSupportHandlerMapping），前端控制器可以调用自定义的映射器对象中的处理器获取函数（getHandler），处理器获取函数可以通过资源请求中的 URL 查找处理器对象（Handler）和多个拦截器对象（HandlerInterceptor），映射器对象将查找到的处理器对

象和多个拦截器对象封装成一个处理器执行链(HandlerExecuteChain)并返回给前端控制器，前端控制器遍历配置的所有处理器适配器，从所有的适配器对象中查找与处理器执行链中的处理器对象（Handler）对应的处理器适配器（HandlerAdapter），由处理器适配器从处理器对象中调用具体的目标处理器对象来处理用户的资源请求。

在其中一个实施例中，在通过目标处理器对象处理资源请求，得到模型视图对象之前，前端控制器还调用执行链中各个拦截器对象中的预处理函数（preHandler），对资源请求进行预处理，当它返回为 false 时，表示请求的整个过程结束。当返回值为 true 时，就会继续调用下一个拦截器对象的预处理函数，直至最后一个拦截器对象也返回 true 时，就会执行步骤 S206，通过目标处理器对象中的函数处理资源请求。

在其中一个实施例中，目标处理器对象继承自同一处理器接口类；通过目标处理器对象处理资源请求，得到模型视图对象包括：获取处理器接口类定义的参数处理函数；将资源请求中的请求参数作为函数参数传递给参数处理函数，获取参数处理函数返回的与资源请求对应的模型视图对象。

具体地，可以根据业务逻辑自定义处理器对象中的参数处理函数，与资源请求对应的处理器对象有多种类型，不同的资源请求对应的处理器对象可以是不同的，不同的处理器对象可以继承自同一处理器接口类或实现了同一个处理器接口。参数处理函数可以预先定义在该处理器接口类中，可以根据不同的业务逻辑创建继承自该处理器接口类的处理器对象。在对该资源请求进行处理时，可将资源请求中的请求参数作为函数参数传递给处理器接口类中的参数处理函数，经过该参数处理函数的处理后，返回与资源请求对应的模型视图对象。

在其中一个实施例中，通过前端控制器确定与处理器对象适配的目标处理器对象的步骤具体包括：前端控制器遍历所有的处理器适配器，查找与处理器对象对应的处理器适配器，通过该处理器适配器调用与处理器对象适配的具体的目标处理器，通过目标处理器处理用户的请求。

在其中一个实施例中，基于 Spring MVC 框架的注解功能，在处理器对象的注解 @Controller 之后利用预先确定的注解类型@ControllerService 写入处理器对象类的属性，可以在前端控制器接收到资源请求时，自动根据上下文获取到处理器对象，在该处理

器对象中通过自定义的注解类型引入对应的参数处理函数，以便处理器对象直接引用该参数处理函数对资源请求进行处理。

步骤 S206，调用拦截器接口类的回调函数对模型视图对象进行验证。

具体地，在处理资源请求之后，前端控制器调用通过映射器对象确定的拦截器对象中的回调函数（即在处理了资源请求之后执行的函数），对得到的模型视图对象进行验证，实现处理器的后处理，可以通过模型视图对象中的结果数据或视图信息进行处理，以对模型视图对象包括的结果数据进行检查。

步骤 S208，当回调函数返回异常类型编码时，则调用异常处理对象对异常类型编码进行处理得到新模型视图对象。

异常类型编码是对各自异常信息进行分类后各自对应的统一编码。可以预先自定义继承自 Spring MVC 框架中异常处理接口类的实现类，作为异常处理对象。比如，异常处理对象 WebSupportDispatcherServlet 继承自 SimpleMappingExceptionResolver，SimpleMappingExceptionResolver 是 Spring MVC 框架中异常处理接口类 HandlerExceptionResolver 的实现类。该异常处理对象就具有了该异常处理接口类中所包括的处理方法，无需重复编码。具体地，当拦截器接口类中的回调函数对模型视图对象进行验证并返回异常编码类型时，则可以调用异常处理对象所包括的处理方法对异常类型编码进行处理，得到新模型视图对象。新模型视图对象包括显示该异常类型编码所对应的异常信息所需要的异常信息数据和视图信息。本实施例中，对于任意业务中出现的异常，都会被拦截器的回调函数确定相应的异常类型，并通过返回新模型视图对象来显示相应的异常信息，能够实现对各种复杂异常信息的统一处理。

在其中一个实施例中，上述方法还包括：获取异常信息，对各异常信息进行分类；按照各异常信息的类型进行编码，得到异常类型编码。

具体地，可以根据实际的业务需要将不同的异常信息进行分类并进行统一编码，还可以指定一个处理方式，以提示用户。以下为拦截器中的回调函数返回的异常类型编码与前端控制器要返回的各种异常信息的对应关系的示例：

000000——验证通过；

100002——没有找到对应接口；

100003——保存失败；

100004——编辑保存失败。

在其中一个实施例中，异常处理对象继承自同一异常处理接口类；调用异常处理对象对异常类型编码进行处理得到新模型视图对象包括：获取异常处理接口类定义的异常处理函数；将异常类型编码作为函数参数传递给异常处理函数，获取异常处理函数返回的新模型视图对象。

具体地，可以根据业务逻辑自定义异常处理对象中的异常处理函数，在对异常编码类型进行处理时，可以均采用统一的异常处理对象中的异常处理函数对异常编码类型进行处理，获取返回的包括异常信息数据的新模型视图对象。

在其中一个实施例中，模型视图对象包括逻辑视图名和模型数据，方法还包括：当回调函数未返回异常类型编码时，则通过前端控制器获取视图解析对象；通过视图解析对象对逻辑视图名进行解析，得到视图对象；通过视图对象渲染模型数据，得到响应数据；通过前端控制器将响应数据返回至客户端。

具体地，当拦截器接口类的回调函数对模型视图对象进行验证未出现异常时，则直接通过前端控制器请求视图解析器对模型视图对象进行解析，得到视图对象，并返回视图对象至前端控制器，再由前端控制器通过视图对象渲染模型数据，得到响应数据，并将响应数据返回至客户端，客户端就可以根据响应数据显示网页页面。

步骤 210，通过前端控制器根据新模型视图对象生成异常页面视图对象；异常页面视图对象用于令客户端显示包括与异常类型编码对应的异常信息的页面。

具体地，对于得到的新模型视图对象，前端控制器可以调用视图解析对象进行解析后得到异常页面视图对象，并根据异常页面视图对象生成响应数据后反馈至客户端，客户端可以根据响应数据中所包括的异常页面视图信息显示包括异常信息的页面。

在其中一个实施例中，通过前端控制器根据新模型视图对象生成异常页面视图对象包括：通过前端控制器调用视图解析对象；通过视图解析对象对新模型视图对象中的逻辑视图名进行解析，得到页面地址；根据新模型视图对象中的模型数据和页面地址，生成异常页面视图对象；将异常页面视图对象返回至客户端，以使客户端根据异常页面视图对象渲染显示包括与异常类型编码对应的异常信息的页面。

具体地，前端控制器调用视图解析对象，请求视图解析对象对新模型视图对象进行解析，视图解析对象解析后得到页面地址，该页面地址中的页面是用于显示与该异常类型编码对应的错误信息，前端控制器可以根据返回的页面地址和新模型视图对象中的模型数据，生成异常页面对象，这样，客户端就可以根据异常页面对象渲染显示包括与出现的异常类型编码对应的异常信息的页面，用户可以根据异常信息明确异常原因。

上述基于 Spring MVC 的异常处理方法，在通过前端控制器找到目标处理器对象处理客户端发送的资源请求得到模型视图对象后，在对模型视图对象进行解析之前，通过前端控制器调用拦截器接口类的回调函数对得到的模型视图对象进行验证，当模型视图对象存在异常时不会通过验证，回调函数会返回与异常对应的异常类型编码，不同的异常会返回不同的异常类型编码，实现对不同异常的统一处理，再进一步调用异常处理对象对返回的异常类型编码进行处理得到新模型视图对象，而新的模型视图对象可以用于生成异常页面视图对象，当客户端接收到前端控制器返回的新模型视图对象时，可以根据新模型视图对象渲染并显示包括异常信息的页面，该异常信息与该异常类型编码对应，实现了对各种异常信息的统一处理。

如图 3 所示，为一个具体的实施例中基于 Spring MVC 的异常处理方法的框架示意图。参照图 3，该方法具体包括以下步骤：

1、通过前端控制器（WebSupportDispatcherServlet）接收资源请求，并根据该资源请求中的资源标识符查找与该资源请求对应的处理器映射器（WebSupportHandlerMapping）；

2、若未查找到处理器映射器，则根据资源标识符获取相应的静态资源，若找不到相应的静态资源，则报 404 错误；

3、若查找到处理器映射器，则映射处理器根据自己的配置或注解（WebSupportWeb2AppBeanPostProcessor）把请求映射为执行链对象（HandlerExecutionChain），并将执行链对象返回给前端控制器，执行链对象包括一个处理器对象（Handler）和多个拦截器对象（HandlerInterceotor）；

4、前端控制器遍历配置的所有适配器，查找与处理器对象对应的处理器适配器 HandlerAdapter；

5、调用拦截器对象的预处理（preHandle）方法，该方法将在用户的资源请求被处理之前被调用，当该方法被调用后并返回 true 时，才会执行步骤 6；

6、处理器适配器调用与处理器对象适配的具体的目标处理器，通过目标处理器处理资源请求，目标处理器处理完毕后返回模型视图对象（ModelAndView）；

7、在处理资源请求之后，调用拦截器对象的回调函数（postHandler），对得到的模型视图对象进行操作；

8、若存在异常，并返回了异常类型编码，则调用继承自异常处理接口（HandlerExceptionResolver）的异常处理类（WebMappingExceptionResolver）对异常类型编码进行处理，返回发生新模型视图对象给前端控制器；

9、前端控制器将新模型视图对象传递给视图解析器 ViewResolver，视图解析器根据配置将新模型视图对象中的逻辑视图名解析为具体的视图对象（View），并返回给前端控制器；前端控制器根据视图对象对传进来的模型数据进行渲染得到响应数据。

应该理解的是，虽然图 2、图 3 的流程图中的各个步骤按照箭头的指示依次显示，但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明，这些步骤的执行并没有严格的顺序限制，这些步骤可以以其它的顺序执行。而且，图 2、图 3 中的至少一部分步骤可以包括多个子步骤或者多个阶段，这些子步骤或者阶段并不必然是在同一时刻执行完成，而是可以在不同的时刻执行，这些子步骤或者阶段的执行顺序也不必然是依次进行，而是可以与其它步骤或者其它步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

在其中一个实施例中，如图 4 所示，提供了一种基于 Spring MVC 的异常处理装置 400，包括：资源请求获取模块 402、目标处理器对象获取模块 404、模型视图对象获取模块 406、回调处理模块 408 和异常处理模块 410，其中：

资源请求获取模块 402，用于通过前端控制器获取客户端发送的资源请求；

模型视图对象获取模块 404，用于根据资源请求中的资源标识符确定对应的目标处理器对象，并通过目标处理器对象处理资源请求，得到模型视图对象；

回调处理模块 406，用于调用拦截器接口类的回调函数对模型视图对象进行验证；

异常处理模块 408，用于当回调函数返回异常类型编码时，则调用异常处理对象对异常类型编码进行处理得到新模型视图对象；通过前端控制器根据新模型视图对象生成异常页面视图对象；异常页面视图对象用于令客户端显示包括与异常类型编码对应的异常信息的页面。

在其中一个实施例中，模型视图对象获取模块 404 还用于从资源请求中提取资源标识符；通过前端控制器根据资源标识符调用相应的映射器对象；通过映射器对象获取与资源请求对应的处理器对象；通过前端控制器查找与处理器对象对应的处理器适配器；通过处理器适配器从处理器对象中确定目标处理器对象。

在其中一个实施例中，目标处理器对象继承自同一处理器接口类；模型视图对象获取模块 404 还用于获取处理器接口类定义的参数处理函数；将资源请求中的请求参数作为函数参数传递给参数处理函数，获取参数处理函数返回的与资源请求对应的模型视图对象。

在其中一个实施例中，异常处理对象继承自同一异常处理接口类；异常处理模块 408 还用于获取异常处理接口类定义的异常处理函数；将异常类型编码作为函数参数传递给异常处理函数，获取异常处理函数返回的新模型视图对象。

在其中一个实施例中，异常处理模块 408 还用于通过前端控制器调用视图解析对象；通过视图解析对象对新模型视图对象中的逻辑视图名进行解析，得到页面地址；根据新模型视图对象中的模型数据和页面地址，生成异常页面视图对象； 将异常页面视图对象返回至客户端，以使客户端根据异常页面视图对象渲染显示包括与异常类型编码对应的异常信息的页面。

在其中一个实施例中，模型视图对象包括逻辑视图名和模型数据，装置还包括视图解析模块，视图解析模块用于当回调函数未返回异常类型编码时，则通过前端控制器获取视图解析对象；通过视图解析对象对逻辑视图名进行解析，得到视图对象；通过视图对象渲染模型数据，得到响应数据；通过前端控制器将响应数据返回至客户端。

在其中一个实施例中，装置还包括异常信息编码模块，异常信息编码模块用于获取异常信息，对各异常信息进行分类；按照各异常信息的类型进行编码，得到异常类型编码。

上述基于 Spring MVC 的异常处理装置，在通过前端控制器找到目标处理器对象处理客户端发送的资源请求得到模型视图对象后，在对模型视图对象进行解析之前，通过前端

控制器调用拦截器接口类的回调函数对得到的模型视图对象进行验证，当模型视图对象存在异常时不会通过验证，回调函数会返回与异常对应的异常类型编码，不同的异常会返回不同的异常类型编码，实现对不同异常的统一处理，再进一步调用异常处理对象对返回的异常类型编码进行处理得到新模型视图对象，而新的模型视图对象可以用于生成异常页面视图对象，当客户端接收到前端控制器返回的新模型视图对象时，可以根据新模型视图对象渲染并显示包括异常信息的页面，该异常信息与该异常类型编码对应，实现了对各种异常信息的统一处理。

关于基于Spring MVC的异常处理装置的具体限定可以参见上文中对于基于Spring MVC的异常处理方法的限定，在此不再赘述。上述基于Spring MVC的异常处理装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中，也可以以软件形式存储于计算机设备中的存储器中，以便于处理器调用执行以上各个模块对应的操作。

在其中一个实施例中，提供了一种计算机设备，该计算机设备可以是服务器，其内部结构图可以如图 5 所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机可读指令和数据库。该内存储器为非易失性存储介质中的操作系统和计算机可读指令的运行提供环境。该计算机设备的数据库用于存储网页资源数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机可读指令被处理器执行时以实现一种基于 Spring MVC 的异常处理方法。

本领域技术人员可以理解，图 5 中示出的结构，仅仅是与本申请方案相关的部分结构的框图，并不构成对本申请方案所应用于其上的计算机设备的限定，具体的计算机设备可以包括比图中所示更多或更少的部件，或者组合某些部件，或者具有不同的部件布置。

在其中一个实施例中，本申请提供的基于 Spring MVC 的异常处理装置可以实现为一种计算机可读指令的形式，计算机可读指令可在如图 5 所示的计算机设备上运行。计算机设备的存储器中可存储组成该基于 Spring MVC 的异常处理装置的各个程序模块，比如，

图 4 所示的资源请求获取模块 402、模型视图对象获取模块 404、回调处理模块 406 和异常处理模块 408。各个程序模块构成的计算机可读指令使得处理器执行本说明书中描述的本申请各个实施例的基于 Spring MVC 的异常处理方法中的步骤。

例如，图 5 所示的计算机设备可以通过如图 Z 所示的基于 Spring MVC 的异常处理装置中的资源请求获取模块 402 执行步骤 S202。计算机设备可通过模型视图对象获取模块 404 执行步骤 S204。计算机设备可通过回调处理模块 406 执行步骤 S208。计算机设备可通过异常处理模块 408 执行步骤 S210。

一种计算机设备，包括存储器和一个或多个处理器，存储器中储存有计算机可读指令，计算机可读指令被处理器执行时，使得一个或多个处理器执行本申请各个实施例的基于 Spring MVC 的异常处理方法中的步骤。

一个或多个存储有计算机可读指令的非易失性计算机可读存储介质，计算机可读指令被一个或多个处理器执行时，使得一个或多个处理器执行本申请各个实施例的基于 Spring MVC 的异常处理方法中的步骤。

本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机可读指令来指令相关的硬件来完成，所述的计算机可读指令可存储于一非易失性计算机可读取存储介质中，该计算机可读指令在执行时，可包括如上述各方法的实施例的流程。本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器（ROM）、可编程 ROM（PROM）、电可编程 ROM（EPROM）、电可擦除可编程 ROM（EEPROM）或闪存。易失性存储器可包括随机存取存储器（RAM）或者外部高速缓冲存储器。作为说明而非局限，RAM 以多种形式可得，诸如静态 RAM（SRAM）、动态 RAM（DRAM）、同步 DRAM（SDRAM）、双数据率 SDRAM（DDRSDRAM）、增强型 SDRAM（ESDRAM）、同步链路（Synchlink） DRAM（SLDRAM）、存储器总线（Rambus）直接 RAM（RDRAM）、直接存储器总线动态 RAM（DRDRAM）、以及存储器总线动态 RAM（RDRAM）等。

以上实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

以上所述实施例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对发明专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请专利的保护范围应以所附权利要求为准。

权利要求书

1、一种基于 Spring MVC 的异常处理方法，包括：

通过前端控制器获取客户端发送的资源请求；

根据所述资源请求中的资源标识符确定对应的目标处理器对象，并通过所述目标处理器对象处理所述资源请求，得到模型视图对象；

调用拦截器接口类的回调函数对所述模型视图对象进行验证；

当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象；及

通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

2、根据权利要求 1 所述的方法，其特征在于，所述根据所述资源请求中的资源标识符确定对应的目标处理器对象，包括：

从所述资源请求中提取资源标识符；

通过所述前端控制器根据所述资源标识符调用相应的映射器对象；

通过所述映射器对象获取与所述资源请求对应的处理器对象；

通过所述前端控制器查找与所述处理器对象对应的处理器适配器；及

通过所述处理器适配器从所述处理器对象中确定目标处理器对象。

3、根据权利要求 1 所述的方法，其特征在于，所述目标处理器对象继承自同一处理器接口类；所述通过所述目标处理器对象处理所述资源请求，得到模型视图对象，包括：

获取所述处理器接口类定义的参数处理函数；及

将所述资源请求中的请求参数作为函数参数传递给所述参数处理函数，获取所述参数处理函数返回的与所述资源请求对应的模型视图对象。

4、根据权利要求 1 所述的方法，其特征在于，所述异常处理对象继承自同一异常处理接口类；所述调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象，包括：

获取所述异常处理接口类定义的异常处理函数；及

将所述异常类型编码作为函数参数传递给所述异常处理函数，获取所述异常处理函数

返回的新模型视图对象。

5、根据权利要求 1 所述的方法，其特征在于，所述通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象，包括：

通过所述前端控制器调用视图解析对象；

通过所述视图解析对象对所述新模型视图对象中的逻辑视图名进行解析，得到页面地址；

根据所述新模型视图对象中的模型数据和所述页面地址，生成异常页面视图对象；及

将所述异常页面视图对象返回至所述客户端，令所述客户端根据所述异常页面视图对象渲染显示包括与所述异常类型编码对应的异常信息的页面。

6、根据权利要求 1 至 5 所述的方法，其特征在于，所述模型视图对象包括逻辑视图名和模型数据，所述方法还包括：

当所述回调函数未返回异常类型编码时，则

通过所述前端控制器获取视图解析对象；

通过所述视图解析对象对所述逻辑视图名进行解析，得到视图对象；

通过所述视图对象渲染所述模型数据，得到响应数据；及

通过所述前端控制器将所述响应数据返回至客户端。

7、根据权利要求 1 至 5 中任一项所述的方法，其特征在于，所述方法还包括：

获取异常信息，对各所述异常信息进行分类；及

按照各所述异常信息的类型进行编码，得到异常类型编码。

8、一种基于 Spring MVC 的异常处理装置，包括：

资源请求获取模块，用于通过前端控制器获取客户端发送的资源请求；

模型视图对象获取模块，用于根据所述资源请求中的资源标识符确定对应的目标处理器对象；并通过所述目标处理器对象处理所述资源请求，得到模型视图对象；

回调处理模块，用于调用拦截器接口类的回调函数对所述模型视图对象进行验证；及

异常处理模块，用于当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象；通过所述前端控制器根据所述新模型视图

对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

9、根据权利要求 8 所述的装置，其特征在于，所述模型视图对象获取模块，还用于从所述资源请求中提取资源标识符；通过所述前端控制器根据所述资源标识符调用相应的映射器对象；通过所述映射器对象获取与所述资源请求对应的处理器对象；通过所述前端控制器查找与所述处理器对象对应的处理器适配器；及通过所述处理器适配器从所述处理器对象中确定目标处理器对象。

10、根据权利要求 8 所述的装置，其特征在于，所述目标处理器对象继承自同一处理器接口类；所述模型视图对象获取模块，还用于获取所述处理器接口类定义的参数处理函数；及将所述资源请求中的请求参数作为函数参数传递给所述参数处理函数，获取所述参数处理函数返回的与所述资源请求对应的模型视图对象。

11、根据权利要求 8 所述的装置，其特征在于，所述异常处理对象继承自同一异常处理接口类；所述异常处理模块，还用于获取所述异常处理接口类定义的异常处理函数；及将所述异常类型编码作为函数参数传递给所述异常处理函数，获取所述异常处理函数返回的新模型视图对象。

12、根据权利要求 8 所述的装置，其特征在于，所述异常处理模块，还用于通过所述前端控制器调用视图解析对象；通过所述视图解析对象对所述新模型视图对象中的逻辑视图名进行解析，得到页面地址；根据所述新模型视图对象中的模型数据和所述页面地址，生成异常页面视图对象；及将所述异常页面视图对象返回至所述客户端，令所述客户端根据所述异常页面视图对象渲染显示包括与所述异常类型编码对应的异常信息的页面。

13、根据权利要求 8 至 11 任一项所述的装置，其特征在于，所述模型视图对象包括逻辑视图名和模型数据，所述装置还包括视图解析模块，所述视图解析模块用于当所述回调函数未返回异常类型编码时，则通过所述前端控制器获取视图解析对象；通过所述视图解析对象对所述逻辑视图名进行解析，得到视图对象；通过所述视图对象渲染所述模型数据，得到响应数据；及通过所述前端控制器将所述响应数据返回至客户端。

14、根据权利要求 8 至 11 任一项所述的装置，其特征在于，所述装置还包括异常信息编码模块，所述异常信息编码模块用于获取异常信息，对各所述异常信息进行分类；及

按照各所述异常信息的类型进行编码，得到异常类型编码。

15、一种计算机设备，包括存储器及一个或多个处理器，所述存储器中储存有计算机可读指令，所述计算机可读指令被所述一个或多个处理器执行时，使得所述一个或多个处理器执行以下步骤：

通过前端控制器获取客户端发送的资源请求；

根据所述资源请求中的资源标识符确定对应的目标处理器对象，并通过所述目标处理器对象处理所述资源请求，得到模型视图对象；

调用拦截器接口类的回调函数对所述模型视图对象进行验证；

当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象；及

通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

16、根据权利要求 15 所述的计算机设备，其特征在于，所述处理器执行所述计算机可读指令时还执行以下步骤：

从所述资源请求中提取资源标识符；

通过所述前端控制器根据所述资源标识符调用相应的映射器对象；

通过所述映射器对象获取与所述资源请求对应的处理器对象；

通过所述前端控制器查找与所述处理器对象对应的处理器适配器；及

通过所述处理器适配器从所述处理器对象中确定目标处理器对象。

17、根据权利要求 15 所述的计算机设备，其特征在于，所述目标处理器对象继承自同一处理器接口类；所述处理器执行所述计算机可读指令时还执行以下步骤：

获取所述处理器接口类定义的参数处理函数；及

将所述资源请求中的请求参数作为函数参数传递给所述参数处理函数，获取所述参数处理函数返回的与所述资源请求对应的模型视图对象。

18、一个或多个存储有计算机可读指令的非易失性计算机可读存储介质 ，所述计算

机可读指令被一个或多个处理器执行时，使得所述一个或多个处理器执行以下步骤：

通过前端控制器获取客户端发送的资源请求；

根据所述资源请求中的资源标识符确定对应的目标处理器对象，并通过所述目标处理器对象处理所述资源请求，得到模型视图对象；

调用拦截器接口类的回调函数对所述模型视图对象进行验证；

当所述回调函数返回异常类型编码时，则调用异常处理对象对所述异常类型编码进行处理得到新模型视图对象；及

通过所述前端控制器根据所述新模型视图对象生成异常页面视图对象；所述异常页面视图对象用于令所述客户端显示包括与所述异常类型编码对应的异常信息的页面。

19、根据权利要求 18 所述的存储介质，其特征在于，所述计算机可读指令被所述处理器执行时还执行以下步骤：

从所述资源请求中提取资源标识符；

通过所述前端控制器根据所述资源标识符调用相应的映射器对象；

通过所述映射器对象获取与所述资源请求对应的处理器对象；

通过所述前端控制器查找与所述处理器对象对应的处理器适配器；及

通过所述处理器适配器从所述处理器对象中确定目标处理器对象。

20、根据权利要求 18 所述的存储介质，其特征在于，所述目标处理器对象继承自同一处理器接口类；所述计算机可读指令被所述处理器执行时还执行以下步骤：

获取所述处理器接口类定义的参数处理函数；及

将所述资源请求中的请求参数作为函数参数传递给所述参数处理函数，获取所述参数处理函数返回的与所述资源请求对应的模型视图对象。

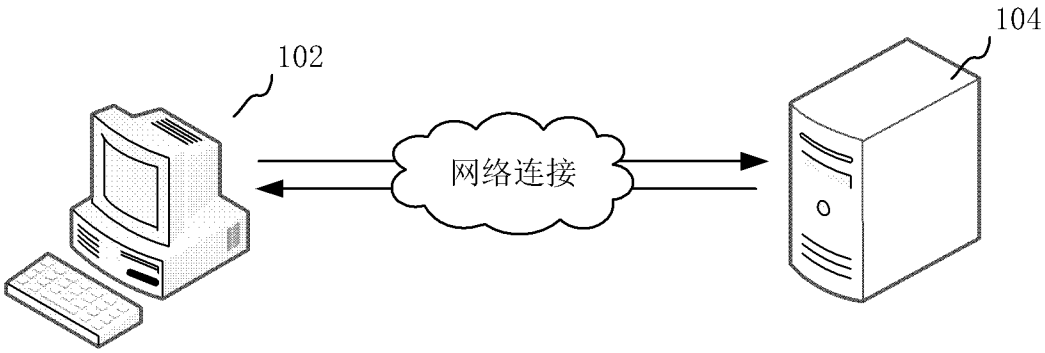

102
104
网络连接

图 1

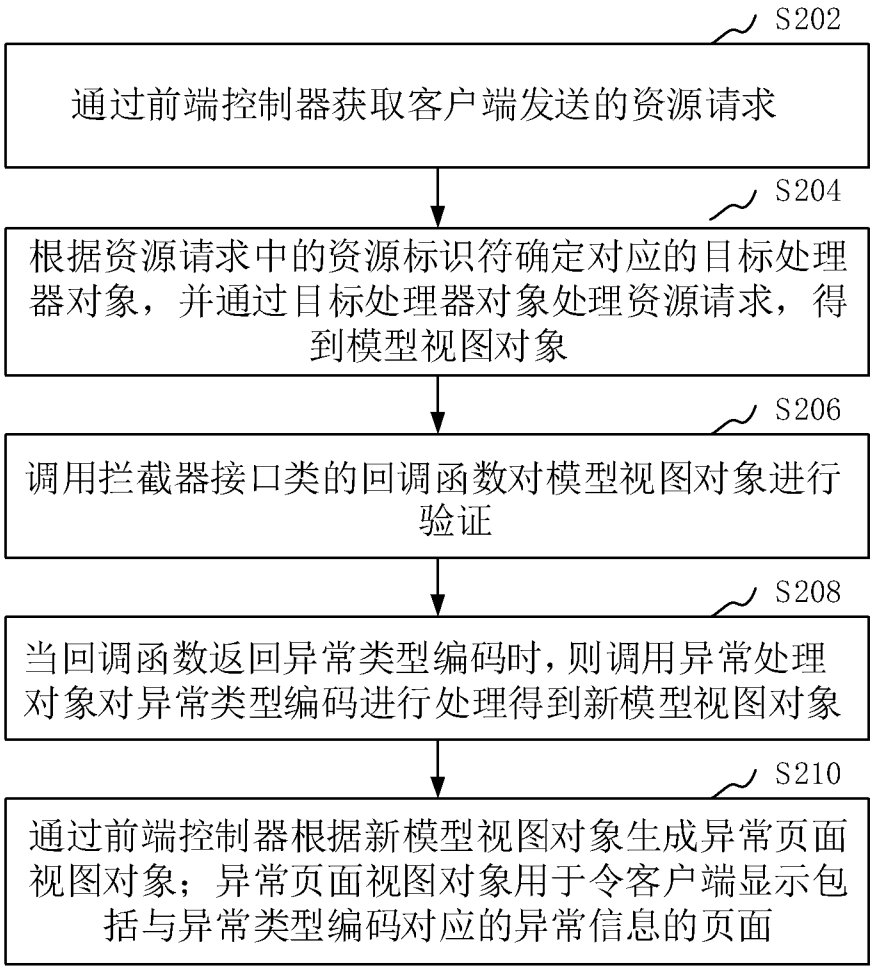

S202
通过前端控制器获取客户端发送的资源请求
S204
根据资源请求中的资源标识符确定对应的目标处理器对象，并通过目标处理器对象处理资源请求，得到模型视图对象
S206
调用拦截器接口类的回调函数对模型视图对象进行验证
S208
当回调函数返回异常类型编码时，则调用异常处理对象对异常类型编码进行处理得到新模型视图对象
S210
通过前端控制器根据新模型视图对象生成异常页面视图对象；异常页面视图对象用于令客户端显示包括与异常类型编码对应的异常信息的页面

图 2

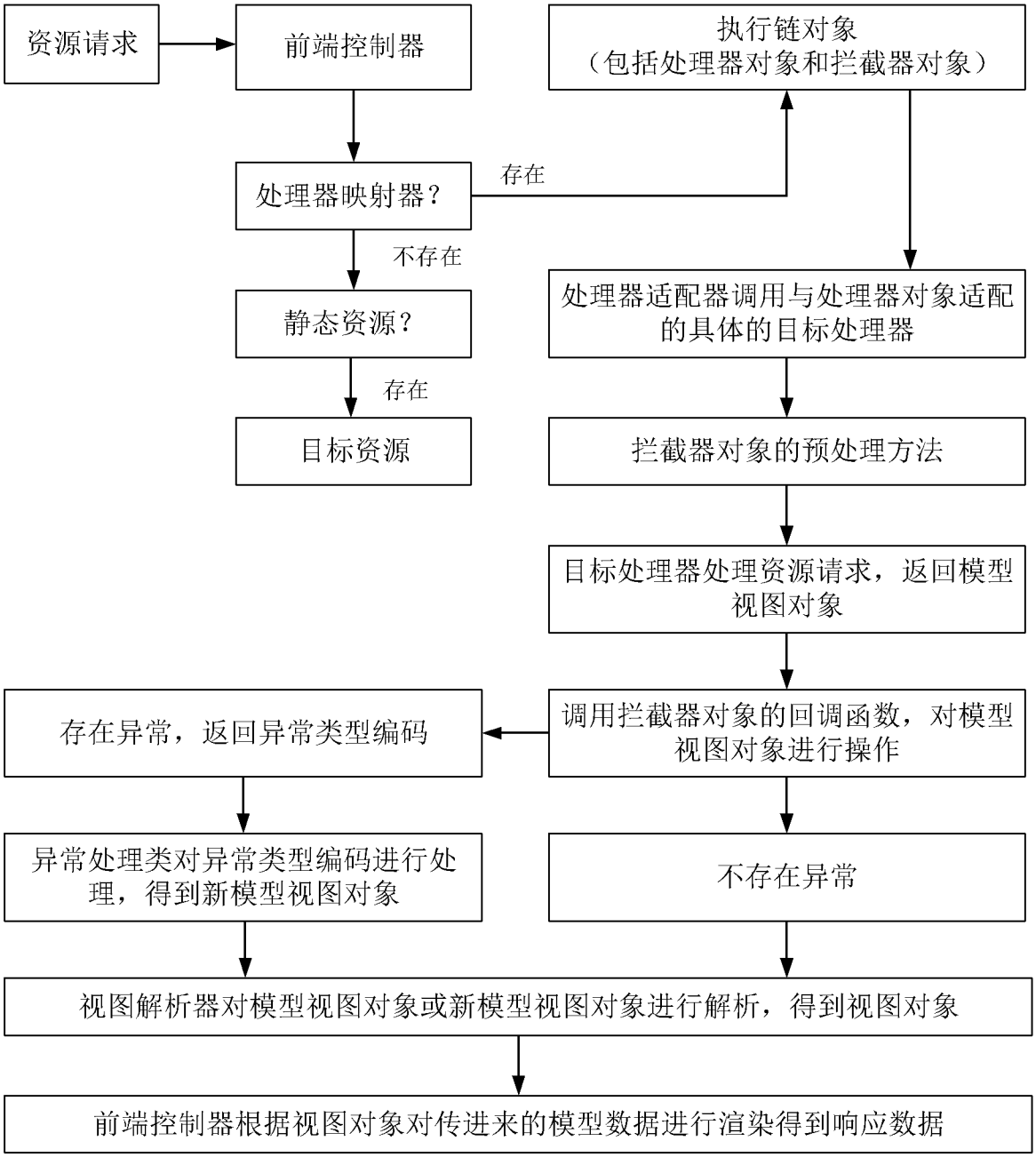

资源请求
前端控制器
执行链对象
（包括处理器对象和拦截器对象）
处理器映射器？
存在
不存在
静态资源？
处理器适配器调用与处理器对象适配的具体的目标处理器
存在
目标资源
拦截器对象的预处理方法
目标处理器处理资源请求，返回模型视图对象
存在异常，返回异常类型编码
调用拦截器对象的回调函数，对模型视图对象进行操作
异常处理类对异常类型编码进行处理，得到新模型视图对象
不存在异常
视图解析器对模型视图对象或新模型视图对象进行解析，得到视图对象
前端控制器根据视图对象对传进来的模型数据进行渲染得到响应数据

图 3

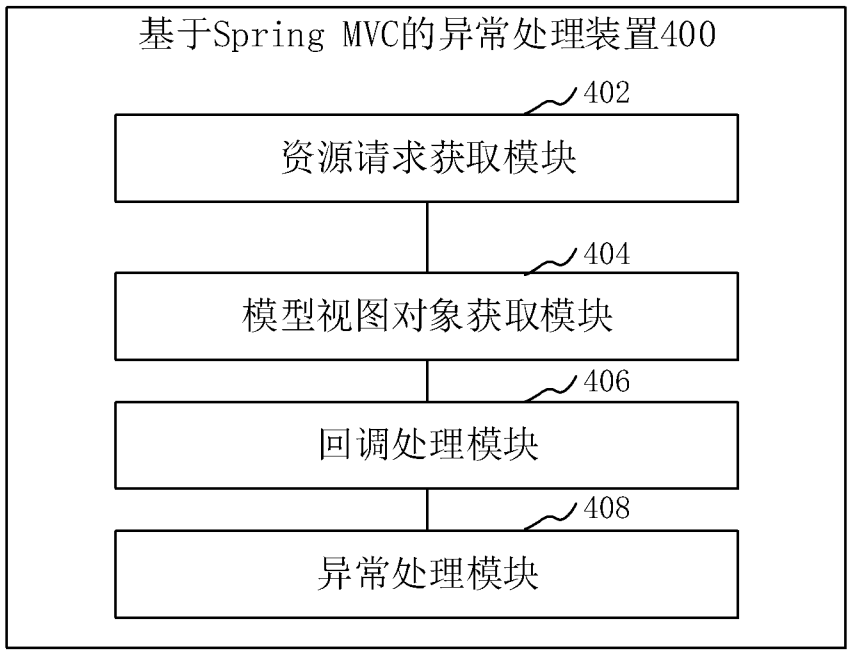

基于Spring MVC的异常处理装置400
402
资源请求获取模块
404
模型视图对象获取模块
406
回调处理模块
408
异常处理模块

图 4

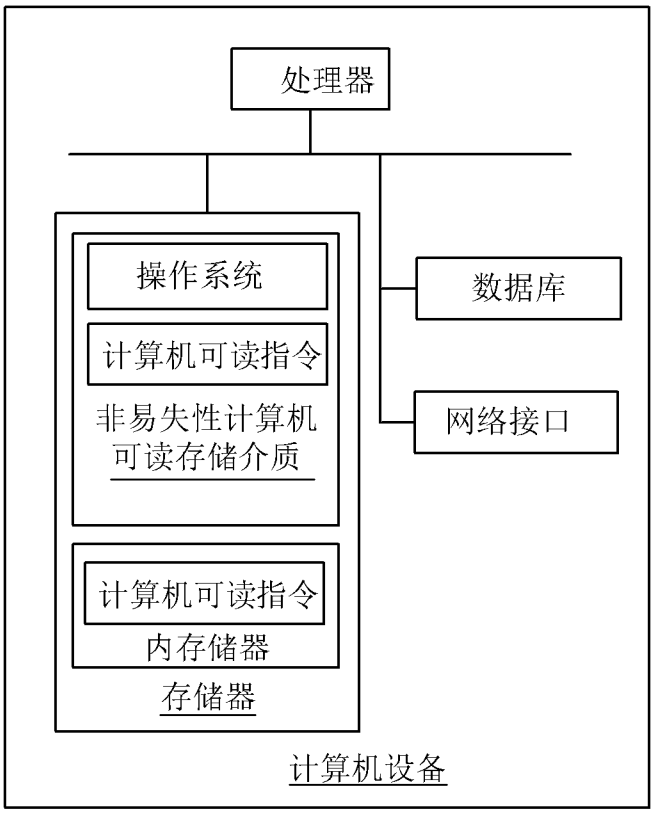

处理器
操作系统
计算机可读指令
非易失性计算机可读存储介质
计算机可读指令
内存储器
存储器
数据库
网络接口
计算机设备

图 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2019/117506

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/448(2018.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI, EPODOC, CNPAT, CNKI: 异常, 视图, 模型, 回调, 编码, 码, 统一, 页面, 模型视图对象, abnormal, view, model, call back, code, unify, unite, page, model and view

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 110069297 A (PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) 30 July 2019 (2019-07-30) description, paragraphs [0005]-[0054]	1-20
A	CN 104317591 A (XI'AN FUTURE INTERNATIONAL INFORMATION CO., LTD. et al.) 28 January 2015 (2015-01-28) description, paragraphs [0045]-[0092], and figures 1-2	1-20
A	CN 103955510 A (ELECTRIC POWER RESEARCH INSTITUTE OF GUANGXI POWER GRID CORPORATION et al.) 30 July 2014 (2014-07-30) description, paragraphs [0023]-[0027]	1-20
A	CN 106484509 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 08 March 2017 (2017-03-08) entire document	1-20
A	US 2013007325 A1 (SAHITA, Ravi L. et al.) 03 January 2013 (2013-01-03) entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 December 2019	**01 February 2020**

Name and mailing address of the ISA/CN	Authorized officer
China National Intellectual Property Administration (ISA/CN) **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088** **China**	
Facsimile No. **(86-10)62019451**	Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2019/117506

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 110069297 A	30 July 2019	None	
CN 104317591 A	28 January 2015	None	
CN 103955510 A	30 July 2014	None	
CN 106484509 A	08 March 2017	None	
US 2013007325 A1	03 January 2013	CN 103620614 A	05 March 2014
		WO 2013006500 A1	10 January 2013
		EP 2727043 A1	07 May 2014

国际检索报告

国际申请号
PCT/CN2019/117506

A. 主题的分类

G06F 9/448(2018.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F H04L

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称，和使用的检索词（如使用）)

WPI, EPODOC, CNPAT, CNKI：异常，视图，模型，回调，编码，码，统一，页面，模型视图对象，abnormal，view，model，call back，code，unify，unite，page，model and view

C. 相关文件

类 型*	引用文件，必要时，指明相关段落	相关的权利要求
PX	CN 110069297 A (平安科技深圳有限公司) 2019年 7月 30日 (2019-07-30) 说明书第[0005]-[0054]段	1-20
A	CN 104317591 A (西安未来国际信息股份有限公司等) 2015年 1月 28日 (2015-01-28) 说明书第[0045]-[0092]段、图1-2	1-20
A	CN 103955510 A (广西电网公司电力科学研究院等) 2014年 7月 30日 (2014-07-30) 说明书第[0023]-[0027]段	1-20
A	CN 106484509 A (腾讯科技深圳有限公司) 2017年 3月 8日 (2017-03-08) 全文	1-20
A	US 2013007325 A1 (SAHITA, Ravi L.等) 2013年 1月 3日 (2013-01-03) 全文	1-20

☐ 其余文件在C栏的续页中列出。

☑ 见同族专利附件。

* 引用文件的具体类型：

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件，或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件（如具体说明的）

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布，与申请不相抵触，但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件，单独考虑该文件，认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件，当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时，要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期	国际检索报告邮寄日期
2019年 12月 30日	2020年 2月 1日
ISA/CN的名称和邮寄地址 中国国家知识产权局(ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088 传真号 (86-10)62019451	受权官员 王欣 电话号码 86-(10)-53961617

国际检索报告
关于同族专利的信息

国际申请号
PCT/CN2019/117506

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	110069297	A	2019年 7月 30日	无			
CN	104317591	A	2015年 1月 28日	无			
CN	103955510	A	2014年 7月 30日	无			
CN	106484509	A	2017年 3月 8日	无			
US	2013007325	A1	2013年 1月 3日	CN	103620614	A	2014年 3月 5日
				WO	2013006500	A1	2013年 1月 10日
				EP	2727043	A1	2014年 5月 7日