

(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
2023 年 2 月 2 日 (02.02.2023)



(10) 国际公布号
WO 2023/005163 A1

(51) 国际专利分类号:
G06F 16/958 (2019.01) **G06F 9/445** (2018.01)

(21) 国际申请号: PCT/CN2022/071877

(22) 国际申请日: 2022 年 1 月 13 日 (13.01.2022)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202110857091.0 2021 年 7 月 28 日 (28.07.2021) 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 518000 (CN)。

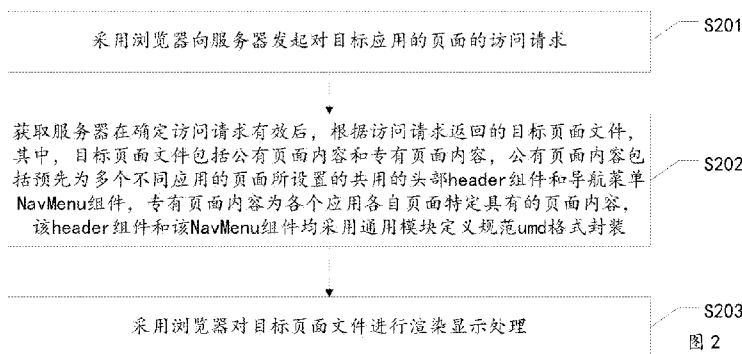
(72) 发明人: 卓根生(ZHUO, Gensheng); 中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

(74) 代理人: 深圳市世联合知识产权代理有限公司(SL INTELLECTUAL PROPERTY CO., LTD.);
中国广东省深圳市福田区沙头街道天安社区泰然四路 29 号天安创新科技广场一期 A 座 805E, Guangdong 518000 (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, IT, 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,

(54) Title: METHOD FOR LOADING APPLICATION PAGE, STORAGE MEDIUM AND RELATED DEVICE THEREOF

(54) 发明名称: 应用页面的加载方法、存储介质、及其相关设备



- S201 Use a browser to initiate an access request to a server for a page of a target application
- S202 Acquire a target page file returned by the server according to the access request after determining that the access request is valid, the target page file comprising common page content and specific page content, the common page content comprising a common header component and a navigation menu (NavMenu) component configured in advance for pages of a plurality of different applications, the specific page content being page content specific to each page of each application, and both the header component and the NavMenu component being packaged using a Universal Module Definition (umd) format
- S203 Use the browser to render and display the target page file

(57) Abstract: A method for loading an application page, a system, a terminal device, a server, a computer device, and a storage medium. The method comprises: using a browser to initiate an access request to a server for a page of a target application (S201); acquiring a target page file returned by the server according to the access request after determining that the access request is valid, the target page file comprising common page content and specific page content, the common page content comprising a common header component and a navigation menu (NavMenu) component configured in advance for pages of a plurality of different applications, and

MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL,
PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL,
ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US,
UZ, VC, VN, WS, 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))。

the specific page content being page content specific to each page of each application, and both the header component and the NavMenu component being packaged using a Universal Module Definition (umd) format (S202); and using the browser to render and display the target page file (S203).

(57) 摘要: 一种应用页面的加载方法、系统、终端设备、服务器、计算机设备及存储介质, 该方法包括: 采用浏览器向服务器发起对目标应用的页面的访问请求 (S201); 获取所述服务器在确定所述访问请求有效后, 根据所述访问请求返回的目标页面文件, 其中, 所述目标页面文件包括公有页面内容和专有页面内容, 所述公有页面内容包括预先为多个不同应用的页面所设置的共用的头部header组件和导航菜单NavMenu组件, 所述专有页面内容为各个应用各自页面特定具有的页面内容, 所述header组件和所述NavMenu组件均采用通用模块定义规范umd格式封装 (S202); 采用所述浏览器对所述目标页面文件进行渲染显示处理 (S203)。

应用页面的加载方法、存储介质、及其相关设备

本申请以 2021 年 7 月 28 日提交的申请号为 202110857091.0, 名称为“应用页面的加载方法、存储介质、及其相关设备”的中国发明专利申请为基础, 并要求其优先权。

技术领域

本申请涉及计算机技术领域, 尤其涉及一种应用页面的加载方法、应用页面的加载系统、终端设备、服务器、计算机设备及存储介质。

背景技术

随着基于各种目的开发的管理应用越来越多, 由于很多项目在初期时项目上线仓促、考虑不够周全、人员变动等多种原因, 以及技术更新换代等因素, 这些管理应用形态各异, 登录形式也各异, 访问不同功能需要来回切换不同的应用, 使用者用起来就很不方便。

这时自然就提出了整合这些管理应用的诉求, 但申请人意识到目前前端开发所通常采用的 Javascript 语言, 由于其不支持“类(class)”和“模块(module)”, 因此导致不同项目之间, 难以实现跨项目的模块代码的引用。这样就导致不同项目之间, 具有相同功能或者可以理论上可以共用的模块, 仍需要重复开发和部署, 不利于提高开发效率。

发明内容

本申请实施例的目的在于提出一种应用页面的加载方法、应用页面的加载系统、终端设备、服务器、计算机设备及存储介质, 其主要目的用于提高现有技术中在开发多个应用的页面时的开发效率。

为了解决上述技术问题, 本申请实施例第一方面提供一种应用页面的加载方法, 该应用页面的加载方法可以包括:

采用浏览器向服务器发起对目标应用的页面的访问请求;

获取该服务器在确定该访问请求有效后, 根据该访问请求返回的目标页面文件, 其中, 该目标页面文件可以包括公有页面内容和专有页面内容, 该公有页面内容可以包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件, 该专有页面内容为各个应用各自页面特定具有的页面内容, 该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

采用该浏览器对该目标页面文件进行渲染显示处理。

为了解决上述技术问题, 本申请实施例第二方面提供另一种应用页面的加载方法, 该应用页面的加载方法可以包括:

获取并存储为多个不同应用的页面所设置的公有页面内容, 该公有页面内容可以包括该多个不同应用的页面所共用的 header 组件和 NavMenu 组件, 该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

获取浏览器发送的对目标应用的页面的访问请求, 并判断该访问请求是否有效;

在确定该访问请求有效时, 根据该访问请求确定目标页面文件, 并向该浏览器返回该目标页面文件, 以使该浏览器对该目标页面文件进行渲染显示处理, 其中, 该目标页面文件可以包括该公有页面内容和专有页面内容, 该专有页面内容为各个应用各自页面特定具有的页面内容。

为了解决上述技术问题, 本申请实施例第三方面提供另一种应用页面的加载方法, 该应用页面的加载方法可以包括:

服务器获取并存储为多个不同应用的页面所设置的公有页面内容, 该公有页面内容可以包括该多个不同应用的页面所共用的 header 组件和 NavMenu 组件, 该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

终端设备采用浏览器向该服务器发起对目标应用的页面的访问请求；

该服务器获取浏览器发送的对目标应用的页面的访问请求，并判断该访问请求是否有效；

该服务器在确定该访问请求有效时，根据该访问请求确定目标页面文件，并向该浏览器返回该目标页面文件，其中，该目标页面文件可以包括该公有页面内容和专有页面内容，该专有页面内容为各个应用各自页面特定具有的页面内容；

该终端设备采用该浏览器对该目标页面文件进行渲染显示处理。

为了解决上述技术问题，本申请实施例第四方面提供一种终端设备，该终端设备可以包括：

发送模块，用于采用浏览器向服务器发起对目标应用的页面的访问请求；

接收模块，用于获取该服务器在确定该访问请求有效后，根据该访问请求返回的目标页面文件，其中，该目标页面文件可以包括公有页面内容和专有页面内容，该公有页面内容可以包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

显示模块，用于采用该浏览器对该目标页面文件进行渲染显示处理。

为了解决上述技术问题，本申请实施例第五方面提供一种服务器，该服务器可以包括：

存储模块，用于获取并存储为多个不同应用的页面所设置的公有页面内容，该公有页面内容可以包括该多个不同应用的页面所共用的 header 组件和 NavMenu 组件，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

判断模块，用于获取浏览器发送的对目标应用的页面的访问请求，并判断该访问请求是否有效；

发送模块，用于在确定该访问请求有效时，根据该访问请求确定目标页面文件，并向该浏览器返回该目标页面文件，以使该浏览器对该目标页面文件进行渲染显示处理，其中，该目标页面文件可以包括该公有页面内容和专有页面内容，该专有页面内容为各个应用各自页面特定具有的页面内容。

为了解决上述技术问题，本申请实施例第六方面提供一种应用页面的加载系统，该应用页面的加载系统包括：

如上述第四方面中所述的终端设备和如上述第五方面中所述的服务器。

为了解决上述技术问题，本申请实施例第七方面还提供一种计算机设备，采用了如下所述的技术方案：

一种计算机设备，包括存储器和处理器，所述存储器中存储有计算机可读指令，所述处理器执行所述计算机可读指令时实现如下所述的应用页面的加载方法的步骤：

采用浏览器向服务器发起对目标应用的页面的访问请求；

获取所述服务器在确定所述访问请求有效后，根据所述访问请求返回的目标页面文件，其中，所述目标页面文件包括公有页面内容和专有页面内容，所述公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，所述专有页面内容为各个应用各自页面特定具有的页面内容，所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

采用所述浏览器对所述目标页面文件进行渲染显示处理。

为了解决上述技术问题，本申请实施例第八方面还提供一种计算机可读存储介质，采用了如下所述的技术方案：

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可读指令，所述计算机可读指令被处理器执行时实现如下所述的应用页面的加载方法的步骤：

采用浏览器向服务器发起对目标应用的页面的访问请求；

获取所述服务器在确定所述访问请求有效后，根据所述访问请求返回的目标页面文件，

其中，所述目标页面文件包括公有页面内容和专有页面内容，所述公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，所述专有页面内容为各个应用各自页面特定具有的页面内容，所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

采用所述浏览器对所述目标页面文件进行渲染显示处理。

与现有技术相比，本申请实施例主要有以下有益效果：

本申请实施例中，在采用浏览器向服务器发起对目标应用的页面的访问请求后，可以获取该服务器在确定该访问请求有效后，根据该访问请求返回的目标页面文件。之后，再采用该浏览器对该目标页面文件进行渲染显示处理。其中，由于该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于通用模块定义规范（universal module definition, UMD）是一种 Javascript 模块规范，遵循该模块规范对代码进行打包后，可以实现打包的模块代码可以在不同项目之间复用的效果。从而使得，公有页面内容可以实现不同应用之间的复用，避免了这一部分代码的重复开发，从而提高了开发效率。

附图说明

为了更清楚地说明本申请中的方案，下面将对本申请实施例描述中所需要使用的附图作一个简单介绍，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。

图 1 是本申请可以应用于其中的示例性系统架构图；

图 2 根据本申请的应用页面的加载方法的一个实施例的流程图；

图 3 根据本申请的应用页面的加载方法的又一个实施例的流程图；

图 4 根据本申请的应用页面的加载方法的又一个实施例的流程图；

图 5 是根据本申请的终端设备 500 的一个实施例的结构示意图；

图 6 是根据本申请的服务器 600 的一个实施例的结构示意图；

图 7 是根据本申请的应用页面的加载系统 700 的一个实施例的结构示意图；

图 8 是根据本申请的计算机设备 800 的一个实施例的结构示意图。

具体实施方式

除非另有定义，本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同；本文中在申请的说明书中所使用的术语只是为了描述具体的实施例的目的，不是旨在于限制本申请；本申请的说明书和权利要求书及上述附图说明中的术语“包括”和“具有”以及它们的任何变形，意图在于覆盖不排他的包含。本申请的说明书和权利要求书或上述附图中的术语“第一”、“第二”等是用于区别不同对象，而不是用于描述特定顺序。

在本文中提及“实施例”意味着，结合实施例描述的特定特征、结构或特性可以包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例，也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是，本文所描述的实施例可以与其它实施例相结合。

为了使本技术领域的人员更好地理解本申请方案，下面将结合附图，对本申请实施例中的技术方案进行清楚、完整地描述。

如图 1 所示，系统架构 100 可以包括终端设备 101、102、103，网络 104 和服务器 105。网络 104 用以在终端设备 101、102、103 和服务器 105 之间提供通信链路的介质。网络 104 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。

用户可以使用终端设备 101、102、103 通过网络 104 与服务器 105 交互，以接收或发送消息等。终端设备 101、102、103 上可以安装有各种通讯客户端应用，例如网页浏览器应用、购物类应用、搜索类应用、即时通信工具、邮箱客户端、社交平台软件等。

终端设备 101、102、103 可以是具有显示屏并且支持网页浏览的各种电子设备，包括但不限于智能手机、平板电脑、电子书阅读器、MP3 播放器（Moving Picture Experts Group Audio Layer III，动态影像专家压缩标准音频层面 3）、MP4（Moving Picture Experts Group Audio Layer IV，动态影像专家压缩标准音频层面 4）播放器、膝上型便携计算机和台式计算机等等。

服务器 105 可以是提供各种服务的服务器，例如对终端设备 101、102、103 上显示的页面提供支持的后台服务器。

需要说明的是，本申请实施例所提供的应用页面的加载方法一般由服务器/终端设备执行。

应该理解，图 1 中的终端设备、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、网络和服务器的。

继续参考图 2，示出了根据本申请的应用页面的加载方法的一个实施例的流程图所述的应用页面的加载方法，包括以下步骤：

步骤 S201，采用浏览器向服务器发起对目标应用的页面的访问请求。

在本实施例中，应用页面的加载方法可以运行于其上的电子设备（例如图 1 所示的终端设备）上，该电子设备可以通过有线连接方式或者无线连接方式与服务器进行数据交互。其中，电子设备与服务器建立 http 连接后，电子设备可以通过浏览器应用或者其他类型应用中嵌入的浏览器模块，向服务器发起对目标应用的页面的访问请求。

步骤 S202，获取服务器在确定访问请求有效后，根据访问请求返回的目标页面文件，其中，目标页面文件包括公有页面内容和专有页面内容，公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。

在本实施例中，服务器在接收到该访问请求后，可以先依据该访问请求中的链接地址，确定该次访问请求是否有效。在确定有效后，查询确定该链接地址指示的目标页面文件，并向电子设备返回该目标页面文件。其中，为了减少不同应用对应的页面中相同部分内容的重复开发以及统一不同应用之间的样式风格，该目标页面文件，即 HTML 文件可以包括公有页面内容和专有页面内容两部分。公有页面内容包括预先为多个不同应用的页面所设置的共用的 header 组件和用于为页面和功能提供导航的 NavMenu 组件，专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。

具体的，由于目前的网页越来越像桌面流程，嵌入网页的 Javascript 代码越来越庞大，越来越复杂，因此开发者不得不使用软件工程的方法，如模块化编程，管理网页的业务逻辑，提高开发效率。通过模块化编程，开发者只需要实现核心的业务逻辑，其他都可以加载别人已经写好的模块。但 Javascript 不是一种模块化编程语言，其不支持“类(class)”和“模块(module)”，因此 Javascript 编程多为非模块化编程。非模块化编程时，正常某个应用页面中所有的组件文件，如 header 组件和 NavMenu 组件对应的 js 文件，都需要放在该应用的大目录下。这样一来，如果多个应用的页面具有相同的组件，那这些组件也需要重复部署在该多个应用的目录下，且每次修改之后都要重复部署。

基于上述情况，目前定义有多种 Javascript 模块规范，开发人员通过遵循 Javascript 模块规范来进行开发和代码打包，可以实现 Javascript 的“类模块化编程”。其中，UMD 是一种 Javascript 模块规范，模块是实现特定功能的一组方法。

本申请实施例中，为实现公有页面内容在多个应用的网页之间的共用，公有页面内容

中的 header 组件和该 NavMenu 组件均采用 umd 格式封装。封装后的 JS 文件，可以单独部署，无需部署在各个应用的目录中，从而可以实现跨模块和跨应用引用，且部署更为简单。

步骤 S203，采用浏览器对目标页面文件进行渲染显示处理。

在本实施例中，接收到目标页面文件后，可以采用浏览器对目标页面文件的代码进行解析，进而对目标页面文件进行渲染和显示处理。在渲染时，浏览器逐行执行目标页面文件中的代码，依次执行构建 dom 树、构建渲染(render)树、布局 render 树 和绘制 render 树等过程，生成和显示页面视图。具体的，浏览器的渲染引擎首先解析目标页面文件，即 html 文件，按照 body 属性下标签的结构将这些标签转化为内容树中的 dom 节点。之后，进一步解析外部 CSS 文件及 style 标签中的样式信息。这些样式信息以及 html 中的可见性指令将被用来构建另一棵树——render 树。Render 树由一些包含有颜色和大小等属性的矩形组成，它们将被按照正确的顺序显示到屏幕上。Render 树构建好了之后，该渲染引擎会执行布局过程，它将确定每个节点在屏幕上的确切坐标。再下一步就是绘制，即遍历 render 树，并使用 UI 后端层绘制每个节点。需要说明的是，这个过程是逐步完成的，为了更好的用户体验，渲染引擎将会尽可能早地将内容呈现到屏幕上，并不会等到所有的 html 都解析完成之后再去做构建和布局 render 树。它是解析完一部分内容就显示一部分内容，同时，可能还在通过网络下载其余内容。

在本申请实施例一些可能的实现方式中，该采用该浏览器对该目标页面文件进行渲染显示处理的步骤，包括：采用该浏览器对该目标页面文件中的代码进行解析，在解析该公有页面内容对应的代码时，触发向该服务器发起对该 header 组件和该 NavMenu 组件的 XHR 请求；接收该服务器返回的该 header 组件和该 NavMenu 组件，并对该 header 组件、该 NavMenu 组件以及该专有页面内容进行渲染显示处理。

本申请实施例中，具体在渲染时，在浏览器解析目标页面文件中公有页面内容对应的代码时，可以触发向服务器发起对该 header 组件和该 NavMenu 组件的 XHR (XML Http Request) 请求。之后，在接收到服务器返回的该 header 组件和该 NavMenu 组件，对该 header 组件、该 NavMenu 组件以及该专有页面内容进行渲染显示处理。

在本申请实施例一些可能的实现方式中，在采用浏览器向服务器发起对目标应用的页面的访问请求的步骤之后，加载方法还包括：接收服务器确定访问请求无效后返回的公有页面文件，公有页面文件为预先设置的多个不同应用所共用的页面文件，公有页面文件包括共用的登录页文件或错误处理页文件；采用浏览器对公有页面文件进行渲染显示处理。

具体的，为统一多个应用的网页之间的风格样式，还可以设置多个应用具有共用的公有页面文件，该公有页面文件可以包括共用的登录页文件或错误处理页文件。在通过浏览器向服务器发送访问请求后，若服务器确定该访问请求无效，则将访问请求指向该公有页面文件，并向浏览器返回该公有页面文件。之后，浏览器对该公有页面文件进行渲染显示处理。由于采用了共用的登录页文件或错误处理页文件，可以统一多个应用的网页之间的风格样式，并且也避免为不同应用重复开发这些页面，提高了开发效率。

与现有技术相比，本申请实施例主要有以下有益效果：

本申请实施例中，在采用浏览器向服务器发起对目标应用的页面的访问请求后，可以获取该服务器在确定该访问请求有效后，根据该访问请求返回的目标页面文件。之后，再采用该浏览器对该目标页面文件进行渲染显示处理。其中，由于该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范，遵循该模块规范对代码进行打包后，可以实现打包的模块代码可以在不同项目之间复用的效果。从而使得，公有页面内容可以实现不同应用之间的复用，避免了这一部分代码的重复开发，从而提高了开发效率。

继续参考图 3，示出了根据本申请的应用页面的加载方法的一个实施例的流程图的应

用页面的加载方法，包括以下步骤：

步骤 S301，获取并存储为多个不同应用的页面所设置的公有页面内容，公有页面内容包括多个不同应用的页面所共用的 header 组件和 NavMenu 组件，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。

在本实施例中，应用页面的加载方法可以运行于其上的服务器上，该服务器可以通过有线连接方式或者无线连接方式与电子设备进行数据交互。该服务器通过外部设备进行配置，例如可以获取和存储外部设备为多个不同应用的页面所设置的公有页面内容，公有页面内容包括多个不同应用的页面所共用的 header 组件和 NavMenu 组件。

步骤 S302，获取浏览器发送的对目标应用的页面的访问请求，并判断访问请求是否有效。

本申请实施例中，电子设备与服务器建立 http 连接后，电子设备可以通过浏览器应用或者其他类型应用中嵌入的浏览器模块，向服务器发起对目标应用的页面的访问请求。之后，浏览器判定该访问请求是否有效。

在本申请实施例一些可能的实现方式中，获取浏览器发送的对目标应用的页面的访问请求，并判断访问请求是否有效的步骤，包括：获取浏览器发送的对目标应用的页面的访问请求，提取访问请求中目标应用的应用标识；采用反向代理服务器 nginx 根据应用标识查询是否可以匹配到目标应用的页面文件；若匹配成功，则根据查询访问请求中所携带的链接地址查询确定目标页面文件；判断目标页面文件的令牌 token 是否有效；若 token 有效，则确定访问请求有效；若匹配不成功，或 token 无效，则确定访问请求无效。

具体的，本申请实施例中，首先需要进行如下设置：(1) 对各旧有应用修改打包 webpack 配置，增加唯一应用前缀作为每个应用为唯一标识，如 /A、/B、/C 等；(2) 增加反向代理服务器 nginx，其用于匹配 /A 等应用路径，使访问请求在确定有效后，可以跳转到各应用页面。此时，服务器在接收到访问请求后，可以先提取该访问请求中的链接地址，例如 /A/home 地址，查询 “/A” 应用标识是否预先在 nginx 中有配置，即查询是否可以匹配到 A 应用的页面文件。若可以匹配得到，则进一步查询获取该链接地址指引的目标页面文件。之后，再提取该目标页面文件的 token，判断该 token 是否有效，即是否存在和是否过期，若有效，则确定该次访问请求有效。若前述匹配不成功或者 token 失效，则确定该次访问请求失效。通过在服务器上设置反向代理服务器来代理外部的浏览器发送的访问请求，可以降低网络和服务器的负载，提高访问效率，此外，通过 token 进一步判断目标网页文件是否有效，也可以进一步提高服务器的安全性。

步骤 S303，在确定访问请求有效时，根据访问请求确定目标页面文件，并向浏览器返回目标页面文件，以使浏览器对目标页面文件进行渲染显示处理，其中，目标页面文件包括公有页面内容和专有页面内容，专有页面内容为各个应用各自页面特定具有的页面内容。

本申请实施例中，服务器在通过上述步骤确定访问请求有效后，向浏览器返回根据访问请求中的链接地址查询确定的目标页面文件。从而使得浏览器在接收到目标页面文件后，可以采用浏览器对目标页面文件的代码进行解析，进而对目标页面文件进行渲染和显示处理。在渲染时，浏览器逐行执行目标页面文件中的代码，依次执行构建 dom 树、构建 render 树、布局 render 树和绘制 render 树等过程，生成和显示页面视图。

在本申请实施例一些可能的实现方式中，在确定访问请求有效时，根据访问请求确定目标页面文件，并向浏览器返回目标页面文件，以使浏览器对目标页面文件进行渲染显示处理的步骤，包括：在确定访问请求有效时，根据访问请求确定目标页面文件，并向浏览器返回目标页面文件；获取浏览器在解析目标页面文件中公有页面内容对应的代码时，触发发起的对 header 组件和 NavMenu 组件的 XHR 请求；向浏览器发送 header 组件和 NavMenu 组件，以使浏览器对 header 组件、NavMenu 组件以及专有页面内容进行渲染显示处理。

具体的，本申请实施例中，为实现公有页面内容在多个应用的网页之间的共用，公有页面内容中的 header 组件和该 NavMenu 组件均采用 umd 格式封装。封装后的 JS 文件，可

以在服务器中单独部署，无需部署在各个应用的目录中，从而可以实现跨模块和跨应用引用，且部署更为简单。具体在渲染时，在浏览器解析目标页面文件中公有页面内容对应的代码时，可以触发向服务器发起对该 header 组件和该 NavMenu 组件的 XHR (XML Http Request) 请求。之后，服务器查询确定 header 组件和该 NavMenu 组件对应的 JS 文件，并向浏览器返回，使浏览器对该 header 组件、该 NavMenu 组件以及该专有页面内容进行渲染显示处理。

在本申请实施例一些可能的实现方式中，在获取浏览器发送的对目标应用的页面的访问请求，并判断访问请求是否有效的步骤之前，加载方法还包括：获取并存储公有页面文件，公有页面文件为预先设置的多个不同应用所共用的页面文件，公有页面文件包括共用的登录页文件或错误处理页文件；获取浏览器发送的对目标应用的页面的访问请求，并判断访问请求是否有效的步骤之后，加载方法还包括：在确定访问请求无效时，向浏览器返回公有页面文件，以使浏览器对公有页面文件进行渲染显示处理。

本申请实施例中，首先需要进行如下设置：（1）删除各个应用自身的登录页、错误处理页面，并修改所有跳转到登录页、错误处理页的指向路径为预设的公共页面文件的路径，该公有页面文件包括共用的登录页文件或错误处理页文件，该路径例如/main/login、/main/404、/main/502 等；（2）增加反向代理服务器 nginx，其用于上述公共页面文件的路径，使访问请求在确定无效后，可以跳转到该公共页面文件的路径上。此时，服务器在接收到访问请求后，通过前述的判断方法，若确定该访问请求无效，则可以自动将访问请求的链接地址跳转至公共页面文件的路径。之后，服务器提取公共页面文件，并向浏览器返回该公共页面文件，使浏览器对该公有页面文件进行渲染显示处理。

需要强调的是，为进一步保证上述公有页面文件的私密和安全性，上述公有页面文件还可以存储于一区块链的节点中。

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次网络交易的信息，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。

与现有技术相比，本申请实施例主要有以下有益效果：

本申请实施例中，服务器在接收到浏览器发起的对目标应用的页面的访问请求后，可以判断该访问请求是否有效，若有效，则根据该访问请求向浏览器返回该目标页面文件，以使该浏览器对该目标页面文件进行渲染显示处理。其中，由于该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范，遵循该模块规范对代码进行打包后，可以实现打包的模块代码可以在不同项目之间复用的效果。从而公有页面内容可以实现多个应用的复用，避免了这一部分代码的重复开发，从而提高了开发效率。

继续参考图 4，示出了根据本申请的应用页面的加载方法的一个实施例的流程图所述的应用页面的加载方法，包括以下步骤：

步骤 S401，服务器获取并存储为多个不同应用的页面所设置的公有页面内容，公有页面内容包括多个不同应用的页面所共用的 header 组件和 NavMenu 组件，header 组件和 NavMenu 组件均采用通用模块定义规范 umd 格式封装。

步骤 S402，终端设备采用浏览器向服务器发起对目标应用的页面的访问请求。

步骤 S403，服务器获取浏览器发送的对目标应用的页面的访问请求，并判断访问请求是否有效。

步骤 S404，服务器在确定访问请求有效时，根据访问请求确定目标页面文件，并向浏

览器返回目标页面文件，其中，目标页面文件包括公有页面内容和专有页面内容，专有页面内容为各个应用各自页面特定具有的页面内容。

步骤 S405，终端设备采用浏览器对目标页面文件进行渲染显示处理。

需要说明的是，上述步骤 S401、步骤 S403 和步骤 S404 分别与前述图 3 所示的实施例中步骤 S301、步骤 S302 和步骤 S303 类似，而上述步骤 S402 和步骤 S405 分别与前述图 2 所示的实施例中步骤 S201 和步骤 S203 类似，具体此处不再过多赘述。

与现有技术相比，本申请实施例主要有以下有益效果：

本申请实施例中，服务器在接收到终端设备通过浏览器发起的对目标应用的页面的访问请求后，可以判断该访问请求是否有效，若有效，则根据该访问请求向浏览器返回该目标页面文件，以使该浏览器对该目标页面文件进行渲染显示处理。其中，由于该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范，遵循该模块规范对代码进行打包后，可以实现打包的模块代码可以在不同项目之间复用的效果。从而公有页面内容可以实现多个应用的复用，避免了这一部分代码的重复开发，从而提高了开发效率。

本申请可用于众多通用或专用的计算机系统环境或配置中。例如：个人计算机、服务器计算机、手持设备或便携式设备、平板型设备、多处理器系统、基于微处理器的系统、置顶盒、可编程的消费电子设备、网络 PC、小型计算机、大型计算机、包括以上任何系统或设备的分布式计算环境等等。本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如流程模块。一般地，流程模块包括执行特定任务或实现特定抽象数据类型的例程、流程、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，流程模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

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，该计算机可读指令可存储于一计算机可读存储介质中，该流程在执行时，可包括如上述各方法的实施例的流程。其中，前述的存储介质可为磁碟、光盘、只读存储记忆体（Read-Only Memory，ROM）等非易失性存储介质，或随机存储记忆体（Random Access Memory，RAM）等。

应该理解的是，虽然附图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，其可以以其他的顺序执行。而且，附图的流程图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，其执行顺序也不必然是依次进行，而是可以与其他步骤或者其他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。

进一步参考图 5，作为对上述图 2 所示方法的实现，本申请提供了一种终端设备的一个实施例，如图 5 所示，本实施例所述的终端设备 500 包括：

发送模块 501，用于采用浏览器向服务器发起对目标应用的页面的访问请求；

接收模块 502，用于获取该服务器在确定该访问请求有效后，根据该访问请求返回的目标页面文件，其中，该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

显示模块 503，用于采用该浏览器对该目标页面文件进行渲染显示处理。

在本申请实施例一些可能的实现方式中，显示模块 503，包括：

解析子模块，用于采用该浏览器对该目标页面文件中的代码进行解析，在解析该公有

页面内容对应的代码时，触发向该服务器发起对该 header 组件和该 NavMenu 组件的 XHR 请求；

渲染子模块，用于接收该服务器返回的该 header 组件和该 NavMenu 组件，并对该 header 组件、该 NavMenu 组件以及该专有页面内容进行渲染显示处理。

在本申请实施例一些可能的实现方式中，显示模块 503，还用于接收该服务器确定该访问请求无效后返回的公有页面文件，该公有页面文件为预先设置的多个不同应用所共用的页面文件，该公有页面文件包括共用的登录页文件或错误处理页文件；

显示模块 503，还用于采用该浏览器对该公有页面文件进行渲染显示处理。

与现有技术相比，本申请实施例主要有以下有益效果：

本申请实施例中，终端设备在采用浏览器向服务器发起对目标应用的页面的访问请求后，可以获取该服务器在确定该访问请求有效后，根据该访问请求返回的目标页面文件。之后，再采用该浏览器对该目标页面文件进行渲染显示处理。其中，由于该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范，遵循该模块规范对代码进行打包后，可以实现打包的模块代码可以在不同项目之间复用的效果。从而使得，公有页面内容可以实现多个应用的复用，避免了这一部分代码的重复开发，从而提高了开发效率。

进一步参考图 6，作为对上述图 3 所示方法的实现，本申请提供了一种服务器的一个实施例，如图 6 所示，本实施例所述的服务器 600 包括：

存储模块 601，用于获取并存储为多个不同应用的页面所设置的公有页面内容，该公有页面内容包括该多个不同应用的页面所共用的 header 组件和 NavMenu 组件，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

判断模块 602，用于获取浏览器发送的对目标应用的页面的访问请求，并判断该访问请求是否有效；

发送模块 603，用于在确定该访问请求有效时，根据该访问请求确定目标页面文件，并向该浏览器返回该目标页面文件，以使该浏览器对该目标页面文件进行渲染显示处理，其中，该目标页面文件包括该公有页面内容和专有页面内容，该专有页面内容为各个应用各自页面特定具有的页面内容。

在本申请实施例一些可能的实现方式中，发送模块 603，包括：

第一发送子模块，用于在确定该访问请求有效时，根据该访问请求确定目标页面文件，并向该浏览器返回该目标页面文件；

接收子模块，用于获取浏览器在解析该目标页面文件中该公有页面内容对应的代码时，触发发起的对该 header 组件和该 NavMenu 组件的 XHR 请求；

第二发送子模块，用于向该浏览器发送该 header 组件和该 NavMenu 组件，以使该浏览器对该 header 组件、该 NavMenu 组件以及该专有页面内容进行渲染显示处理。

在本申请实施例一些可能的实现方式中，

存储模块 601，还用于获取并存储公有页面文件，该公有页面文件为预先设置的多个不同应用所共用的页面文件，该公有页面文件包括共用的登录页文件或错误处理页文件；

发送模块 603，还用于在确定该访问请求无效时，向该浏览器返回该公有页面文件，以使该浏览器对该公有页面文件进行渲染显示处理。

在本申请实施例一些可能的实现方式中，判断模块 602，包括

提取子模块，用于获取浏览器发送的对目标应用的页面的访问请求，提取该访问请求中该目标应用的应用标识；

反向代理子模块，用于采用反向代理服务器 nginx 根据该应用标识查询是否可以匹配到该目标应用的页面文件；

查询子模块，用于若匹配成功，则根据查询该访问请求中所携带的链接地址查询确定目标页面文件；

判断子模块，用于判断该目标页面文件的令牌 token 是否有效；若该 token 有效，则确定该访问请求有效；若匹配不成功或该 token 无效，则确定该访问请求无效。

与现有技术相比，本申请实施例主要有以下有益效果：

本申请实施例中，服务器在接收到浏览器发起的对目标应用的页面的访问请求后，可以判断该访问请求是否有效，若有效，则根据该访问请求向浏览器返回该目标页面文件，以使该浏览器对该目标页面文件进行渲染显示处理。其中，由于该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范，遵循该模块规范对代码进行打包后，可以实现打包的模块代码可以在不同项目之间复用的效果。从而公有页面内容可以实现多个应用的复用，避免了这一部分代码的重复开发，从而提高了开发效率。

进一步参考图 7，作为对上述图 4 所示方法的实现，本申请提供了一种应用页面的加载系统的一个实施例，如图 7 所示，本实施例所述的应用页面的加载系统 700 包括：

如上述图 5 所示实施例中所述的终端设备 500 和如上述图 6 所示实施例中所述的服务器 600。

与现有技术相比，本申请实施例主要有以下有益效果：

本申请实施例中，服务器在接收到终端设备通过浏览器发起的对目标应用的页面的访问请求后，可以判断该访问请求是否有效，若有效，则根据该访问请求向浏览器返回该目标页面文件，以使该浏览器对该目标页面文件进行渲染显示处理。其中，由于该目标页面文件包括公有页面内容和专有页面内容，该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，该专有页面内容为各个应用各自页面特定具有的页面内容，该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范，遵循该模块规范对代码进行打包后，可以实现打包的模块代码可以在不同项目之间复用的效果。从而公有页面内容可以实现多个应用的复用，避免了这一部分代码的重复开发，从而提高了开发效率。

为解决上述技术问题，本申请实施例还提供计算机设备。具体请参阅图 8，图 8 为本实施例计算机设备基本结构框图。

所述计算机设备 800 包括通过系统总线相互通信连接存储器 801、处理器 802、网络接口 803。需要指出的是，图中仅示出了具有组件 801-803 的计算机设备 800，但是应理解的是，并不要求实施所有示出的组件，可以替代地实施更多或者更少的组件。其中，本技术领域技术人员可以理解，这里的计算机设备是一种能够按照事先设定或存储的指令，自动进行数值计算和/或信息处理的设备，其硬件包括但不限于微处理器、专用集成电路（Application Specific Integrated Circuit, ASIC）、可编程门阵列（Field-Programmable Gate Array, FPGA）、数字处理器（Digital Signal Processor, DSP）、嵌入式设备等。

所述计算机设备可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述计算机设备可以与用户通过键盘、鼠标、遥控器、触摸板或声控设备等方式进行人机交互。

所述存储器 801 至少包括一种类型的可读存储介质，所述可读存储介质包括闪存、硬盘、多媒体卡、卡型存储器（例如，SD 或 DX 存储器等）、随机访问存储器（RAM）、静态随机访问存储器（SRAM）、只读存储器（ROM）、电可擦除可编程只读存储器（EEPROM）、可编程只读存储器（PROM）、磁性存储器、磁盘、光盘等，所述计算机可读存储介质可以是非易失性，也可以是易失性。在一些实施例中，所述存储器 801 可以是所述计算机设备

800 的内部存储单元,例如该计算机设备 800 的硬盘或内存。在另一些实施例中,所述存储器 801 也可以是所述计算机设备 800 的外部存储设备,例如该计算机设备 800 上配备的插接式硬盘,智能存储卡(Smart Media Card, SMC),安全数字(Secure Digital, SD)卡,闪存卡(Flash Card)等。当然,所述存储器 801 还可以既包括所述计算机设备 800 的内部存储单元也包括其外部存储设备。本实施例中,所述存储器 801 通常用于存储安装于所述计算机设备 800 的操作系统和各类应用软件,例如前述图 2、图 3 或图 4 中所述的应用页面的加载方法的计算机可读指令等。此外,所述存储器 801 还可以用于暂时地存储已经输出或者将要输出的各类数据。

所述处理器 802 在一些实施例中可以是中央处理器(Central Processing Unit, CPU)、控制器、微控制器、微处理器、或其他数据处理芯片。该处理器 802 通常用于控制所述计算机设备 800 的总体操作。本实施例中,所述处理器 802 用于运行所述存储器 801 中存储的计算机可读指令或者处理数据,例如运行所述应用页面的加载方法的计算机可读指令。

所述网络接口 803 可包括无线网络接口或有线网络接口,该网络接口 803 通常用于在所述计算机设备 800 与其他电子设备之间建立通信连接。

与现有技术相比,本申请实施例主要有以下有益效果:

本申请实施例中,计算机在执行该计算机可读指令时,在接收到终端设备通过浏览器发起的对目标应用的页面的访问请求后,可以判断该访问请求是否有效,若有效,则根据该访问请求向浏览器返回该目标页面文件,以使该浏览器对该目标页面文件进行渲染显示处理。其中,由于该目标页面文件包括公有页面内容和专有页面内容,该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件,该专有页面内容为各个应用各自页面特定具有的页面内容,该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范,遵循该模块规范对代码进行打包后,可以实现打包的模块代码可以在不同项目之间复用的效果。从而公有页面内容可以实现多个应用的复用,避免了这一部分代码的重复开发,从而提高了开发效率。

本申请还提供了另一种实施方式,即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可读指令,所述计算机可读指令可被至少一个处理器执行,以使所述至少一个处理器执行如上述图 2、图 3 或图 4 所述的应用页面的加载方法的步骤。

与现有技术相比,本申请实施例主要有以下有益效果:

本申请实施例中,计算机可读指令可被至少一个处理器执行,在接收到终端设备通过浏览器发起的对目标应用的页面的访问请求后,可以判断该访问请求是否有效,若有效,则根据该访问请求向浏览器返回该目标页面文件,以使该浏览器对该目标页面文件进行渲染显示处理。其中,由于该目标页面文件包括公有页面内容和专有页面内容,该公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件,该专有页面内容为各个应用各自页面特定具有的页面内容,该 header 组件和该 NavMenu 组件均采用通用模块定义规范 umd 格式封装。由于 UMD 是一种 Javascript 模块规范,遵循该模块规范对代码进行打包后,可以实现打包的模块代码可以在不同项目之间复用的效果。从而公有页面内容可以实现多个应用的复用,避免了这一部分代码的重复开发,从而提高了开发效率。

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,本申请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设备(可以是手机,计算机,服务器,空调器,或者网络设备等)执行本申请各个实施例所述的方法。

显然,以上所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例,附图

中给出了本申请的较佳实施例，但并不限制本申请的专利范围。本申请可以以许多不同的形式来实现，相反地，提供这些实施例的目的是使对本申请的公开内容的理解更加透彻全面。尽管参照前述实施例对本申请进行了详细的说明，对于本领域的技术人员来而言，其依然可以对前述各具体实施方式所记载的技术方案进行修改，或者对其中部分技术特征进行等效替换。凡是利用本申请说明书及附图内容所做的等效结构，直接或间接运用在其他相关的技术领域，均同理在本申请专利保护范围之内。

权 利 要 求 书

1. 一种应用页面的加载方法，其中，包括：

采用浏览器向服务器发起对目标应用的页面的访问请求；

获取所述服务器在确定所述访问请求有效后，根据所述访问请求返回的目标页面文件，

5 其中，所述目标页面文件包括公有页面内容和专有页面内容，所述公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件，所述专有页面内容为各个应用各自页面特定具有的页面内容，所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

采用所述浏览器对所述目标页面文件进行渲染显示处理。

10 2. 根据权利要求 1 所述的应用页面的加载方法，其中，所述采用所述浏览器对所述目标页面文件进行渲染显示处理的步骤，包括：

采用所述浏览器对所述目标页面文件中的代码进行解析，在解析所述公有页面内容对应的代码时，触发向所述服务器发起对所述 header 组件和所述 NavMenu 组件的 XHR 请求；

15 接收所述服务器返回的所述 header 组件和所述 NavMenu 组件，并对所述 header 组件、所述 NavMenu 组件以及所述专有页面内容进行渲染显示处理。

3. 根据权利要求 1 或 2 所述的应用页面的加载方法，其中，在所述采用浏览器向服务器发起对目标应用的页面的访问请求的步骤之后，所述加载方法还包括：

20 接收所述服务器确定所述访问请求无效后返回的公有页面文件，所述公有页面文件为预先设置的多个不同应用所共用的页面文件，所述公有页面文件包括共用的登录页文件或错误处理页文件；

采用所述浏览器对所述公有页面文件进行渲染显示处理。

4. 一种应用页面的加载方法，其中，包括：

25 获取并存储为多个不同应用的页面所设置的公有页面内容，所述公有页面内容包括所述多个不同应用的页面所共用的 header 组件和 NavMenu 组件，所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

获取浏览器发送的对目标应用的页面的访问请求，并判断所述访问请求是否有效；

30 在确定所述访问请求有效时，根据所述访问请求确定目标页面文件，并向所述浏览器返回所述目标页面文件，以使所述浏览器对所述目标页面文件进行渲染显示处理，其中，所述目标页面文件包括所述公有页面内容和专有页面内容，所述专有页面内容为各个应用各自页面特定具有的页面内容。

5. 根据权利要求 4 所述的应用页面的加载方法，其中，所述在确定所述访问请求有效时，根据所述访问请求确定目标页面文件，并向所述浏览器返回所述目标页面文件，以使所述浏览器对所述目标页面文件进行渲染显示处理的步骤，包括：

35 在确定所述访问请求有效时，根据所述访问请求确定目标页面文件，并向所述浏览器返回所述目标页面文件；

获取浏览器在解析所述目标页面文件中所述公有页面内容对应的代码时，触发发起的对所述 header 组件和所述 NavMenu 组件的 XHR 请求；

向所述浏览器发送所述 header 组件和所述 NavMenu 组件，以使所述浏览器对所述 header 组件、所述 NavMenu 组件以及所述专有页面内容进行渲染显示处理。

40 6. 根据权利要求 4 或 5 所述的应用页面的加载方法，其中，在所述获取浏览器发送的对目标应用的页面的访问请求，并判断所述访问请求是否有效的步骤之前，所述加载方法还包括：

获取并存储公有页面文件，所述公有页面文件为预先设置的多个不同应用所共用的页面文件，所述公有页面文件包括共用的登录页文件或错误处理页文件；

45 所述获取浏览器发送的对目标应用的页面的访问请求，并判断所述访问请求是否有效的步骤之后，所述加载方法还包括：

在确定所述访问请求无效时,向所述浏览器返回所述公有页面文件,以使所述浏览器对所述公有页面文件进行渲染显示处理。

7. 根据权利要求4或5所述的应用页面的加载方法,其中,所述获取浏览器发送的对目标应用的页面的访问请求,并判断所述访问请求是否有效的步骤,包括:

5 获取浏览器发送的对目标应用的页面的访问请求,提取所述访问请求中所述目标应用的应用标识;

采用反向代理服务器 nginx 根据所述应用标识查询是否可以匹配到所述目标应用的页面文件;

若匹配成功,则根据查询所述访问请求中所携带的链接地址查询确定目标页面文件;

10 判断所述目标页面文件的令牌 token 是否有效;

若所述 token 有效,则确定所述访问请求有效;

若匹配不成功或所述 token 无效,则确定所述访问请求无效。

8. 一种终端设备,其中,包括:

发送模块,用于采用浏览器向服务器发起对目标应用的页面的访问请求;

15 接收模块,用于获取所述服务器在确定所述访问请求有效后,根据所述访问请求返回的目标页面文件,其中,所述目标页面文件包括公有页面内容和专有页面内容,所述公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件,所述专有页面内容为各个应用各自页面特定具有的页面内容,所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

20 显示模块,用于采用所述浏览器对所述目标页面文件进行渲染显示处理。

9. 一种服务器,其中,包括:

存储模块,用于获取并存储为多个不同应用的页面所设置的公有页面内容,所述公有页面内容包括所述多个不同应用的页面所共用的 header 组件和 NavMenu 组件,所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

25 判断模块,用于获取浏览器发送的对目标应用的页面的访问请求,并判断所述访问请求是否有效;

发送模块,用于在确定所述访问请求有效时,根据所述访问请求确定目标页面文件,并向所述浏览器返回所述目标页面文件,以使所述浏览器对所述目标页面文件进行渲染显示处理,其中,所述目标页面文件包括所述公有页面内容和专有页面内容,所述专有页面内容为各个应用各自页面特定具有的页面内容。

30 10. 一种计算机设备,包括存储器和处理器,所述存储器中存储有计算机可读指令,所述处理器执行所述计算机可读指令时实现如下所述的应用页面的加载方法的步骤:

采用浏览器向服务器发起对目标应用的页面的访问请求;

35 获取所述服务器在确定所述访问请求有效后,根据所述访问请求返回的目标页面文件,其中,所述目标页面文件包括公有页面内容和专有页面内容,所述公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件,所述专有页面内容为各个应用各自页面特定具有的页面内容,所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

采用所述浏览器对所述目标页面文件进行渲染显示处理。

40 11. 根据权利要求10所述的计算机设备,其中,所述采用所述浏览器对所述目标页面文件进行渲染显示处理的步骤,包括:

采用所述浏览器对所述目标页面文件中的代码进行解析,在解析所述公有页面内容对应的代码时,触发向所述服务器发起对所述 header 组件和所述 NavMenu 组件的 XHR 请求;

45 接收所述服务器返回的所述 header 组件和所述 NavMenu 组件,并对所述 header 组件、所述 NavMenu 组件以及所述专有页面内容进行渲染显示处理。

12. 根据权利要求10或11所述的计算机设备,其中,在所述采用浏览器向服务器发

起对目标应用的页面的访问请求的步骤之后, 所述加载方法还包括:

接收所述服务器确定所述访问请求无效后返回的公有页面文件, 所述公有页面文件为预先设置的多个不同应用所共用的页面文件, 所述公有页面文件包括共用的登录页文件或错误处理页文件;

5 采用所述浏览器对所述公有页面文件进行渲染显示处理。

13. 一种计算机设备, 包括存储器和处理器, 所述存储器中存储有计算机可读指令, 所述处理器执行所述计算机可读指令时实现如下所述的应用页面的加载方法的步骤:

获取并存储为多个不同应用的页面所设置的公有页面内容, 所述公有页面内容包括所述多个不同应用的页面所共用的 header 组件和 NavMenu 组件, 所述 header 组件和所述
10 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

获取浏览器发送的对目标应用的页面的访问请求, 并判断所述访问请求是否有效;

在确定所述访问请求有效时, 根据所述访问请求确定目标页面文件, 并向所述浏览器返回所述目标页面文件, 以使所述浏览器对所述目标页面文件进行渲染显示处理, 其中, 所述目标页面文件包括所述公有页面内容和专有页面内容, 所述专有页面内容为各个应用
15 各自页面特定具有的页面内容。

14. 根据权利要求 13 所述的计算机设备, 其中, 所述在确定所述访问请求有效时, 根据所述访问请求确定目标页面文件, 并向所述浏览器返回所述目标页面文件, 以使所述浏览器对所述目标页面文件进行渲染显示处理的步骤, 包括:

在确定所述访问请求有效时, 根据所述访问请求确定目标页面文件, 并向所述浏览器
20 返回所述目标页面文件;

获取浏览器在解析所述目标页面文件中所述公有页面内容对应的代码时, 触发发起的对所述 header 组件和所述 NavMenu 组件的 XHR 请求;

向所述浏览器发送所述 header 组件和所述 NavMenu 组件, 以使所述浏览器对所述 header 组件、所述 NavMenu 组件以及所述专有页面内容进行渲染显示处理。

15. 一种计算机可读存储介质, 其中, 所述计算机可读存储介质上存储有计算机可读指令, 所述计算机可读指令被处理器执行时实现如下所述的应用页面的加载方法的步骤:

采用浏览器向服务器发起对目标应用的页面的访问请求;

获取所述服务器在确定所述访问请求有效后, 根据所述访问请求返回的目标页面文件, 其中, 所述目标页面文件包括公有页面内容和专有页面内容, 所述公有页面内容包括预先为多个不同应用的页面所设置的共用的头部 header 组件和导航菜单 NavMenu 组件, 所述
30 专有页面内容为各个应用各自页面特定具有的页面内容, 所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装;

采用所述浏览器对所述目标页面文件进行渲染显示处理。

16. 根据权利要求 15 所述的计算机可读存储介质, 其中, 所述采用所述浏览器对所述
35 目标页面文件进行渲染显示处理的步骤, 包括:

采用所述浏览器对所述目标页面文件中的代码进行解析, 在解析所述公有页面内容对应的代码时, 触发向所述服务器发起对所述 header 组件和所述 NavMenu 组件的 XHR 请求;

接收所述服务器返回的所述 header 组件和所述 NavMenu 组件, 并对所述 header 组件、所述 NavMenu 组件以及所述专有页面内容进行渲染显示处理。

17. 根据权利要求 15 或 16 所述的计算机可读存储介质, 其中, 在所述采用浏览器向
40 服务器发起对目标应用的页面的访问请求的步骤之后, 所述加载方法还包括:

接收所述服务器确定所述访问请求无效后返回的公有页面文件, 所述公有页面文件为预先设置的多个不同应用所共用的页面文件, 所述公有页面文件包括共用的登录页文件或错误处理页文件;

45 采用所述浏览器对所述公有页面文件进行渲染显示处理。

18. 一种计算机可读存储介质, 其中, 所述计算机可读存储介质上存储有计算机可读

指令，所述计算机可读指令被处理器执行时实现如下所述的应用页面的加载方法的步骤：

获取并存储为多个不同应用的页面所设置的公有页面内容，所述公有页面内容包括所述多个不同应用的页面所共用的 header 组件和 NavMenu 组件，所述 header 组件和所述 NavMenu 组件均采用通用模块定义规范 umd 格式封装；

5 获取浏览器发送的对目标应用的页面的访问请求，并判断所述访问请求是否有效；

在确定所述访问请求有效时，根据所述访问请求确定目标页面文件，并向所述浏览器返回所述目标页面文件，以使所述浏览器对所述目标页面文件进行渲染显示处理，其中，所述目标页面文件包括所述公有页面内容和专有页面内容，所述专有页面内容为各个应用各自页面特定具有的页面内容。

10 19. 根据权利要求 18 所述的计算机可读存储介质，其中，所述在确定所述访问请求有效时，根据所述访问请求确定目标页面文件，并向所述浏览器返回所述目标页面文件，以使所述浏览器对所述目标页面文件进行渲染显示处理的步骤，包括：

在确定所述访问请求有效时，根据所述访问请求确定目标页面文件，并向所述浏览器返回所述目标页面文件；

15 获取浏览器在解析所述目标页面文件中所述公有页面内容对应的代码时，触发发起的对所述 header 组件和所述 NavMenu 组件的 XHR 请求；

向所述浏览器发送所述 header 组件和所述 NavMenu 组件，以使所述浏览器对所述 header 组件、所述 NavMenu 组件以及所述专有页面内容进行渲染显示处理。

20 20. 根据权利要求 18 或 19 所述的计算机可读存储介质，其中，在所述获取浏览器发送的对目标应用的页面的访问请求，并判断所述访问请求是否有效的步骤之前，所述加载方法还包括：

获取并存储公有页面文件，所述公有页面文件为预先设置的多个不同应用所共用的页面文件，所述公有页面文件包括共用的登录页文件或错误处理页文件；

25 所述获取浏览器发送的对目标应用的页面的访问请求，并判断所述访问请求是否有效的步骤之后，所述加载方法还包括：

在确定所述访问请求无效时，向所述浏览器返回所述公有页面文件，以使所述浏览器对所述公有页面文件进行渲染显示处理。

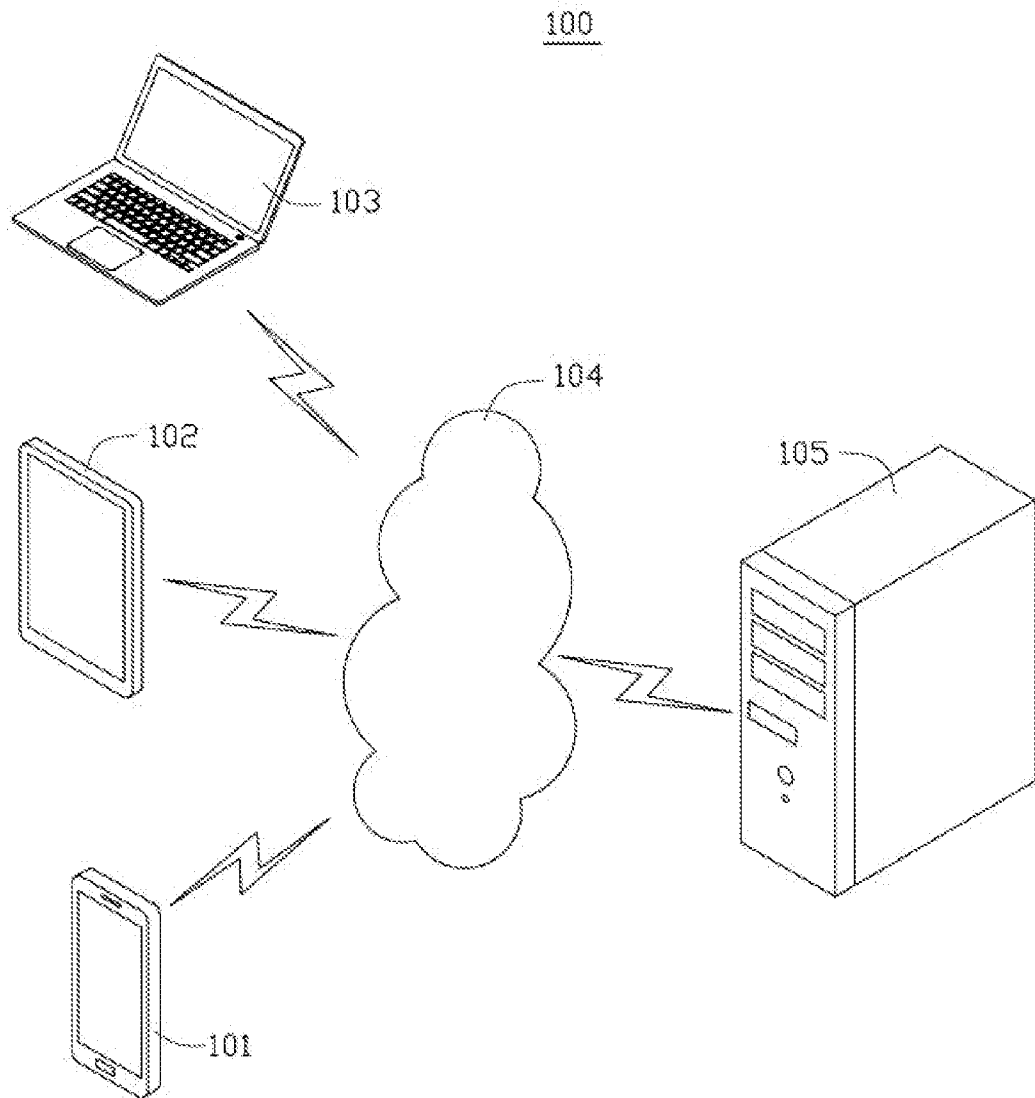


图 1

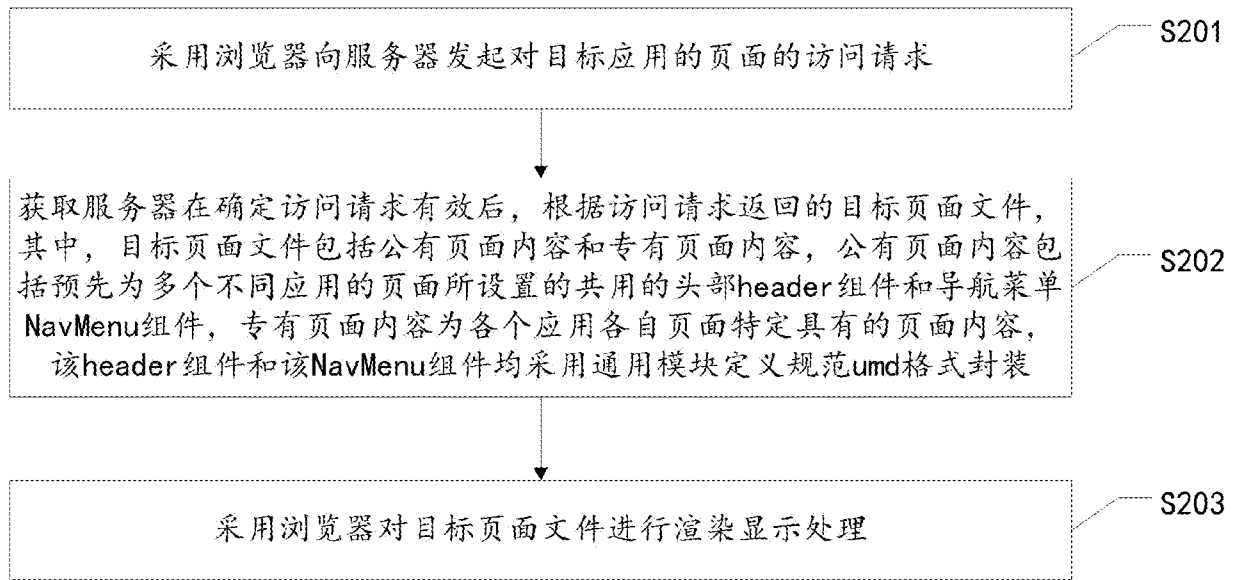


图 2

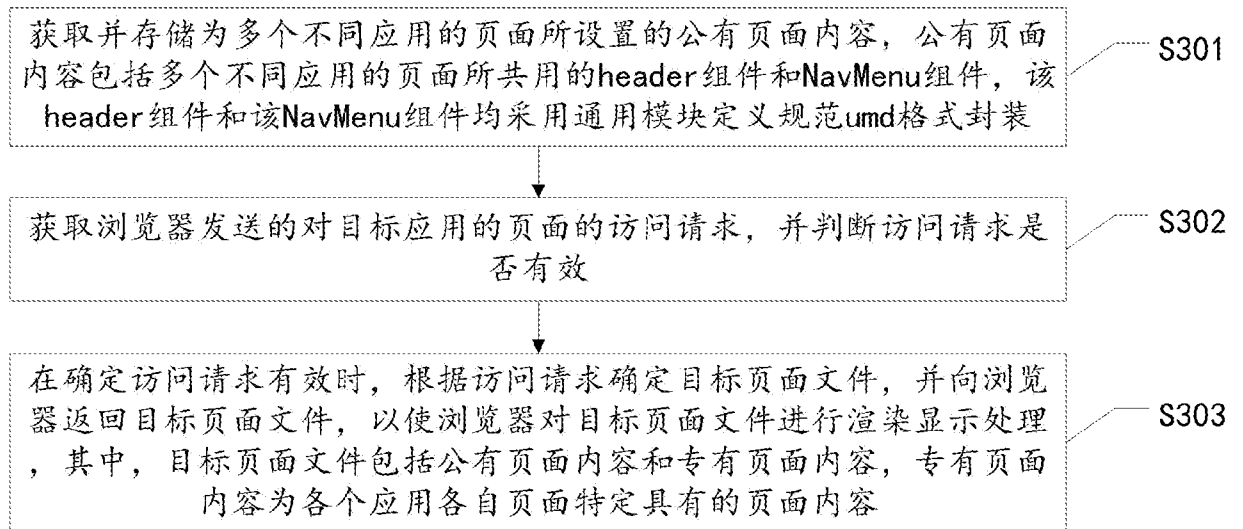


图 3

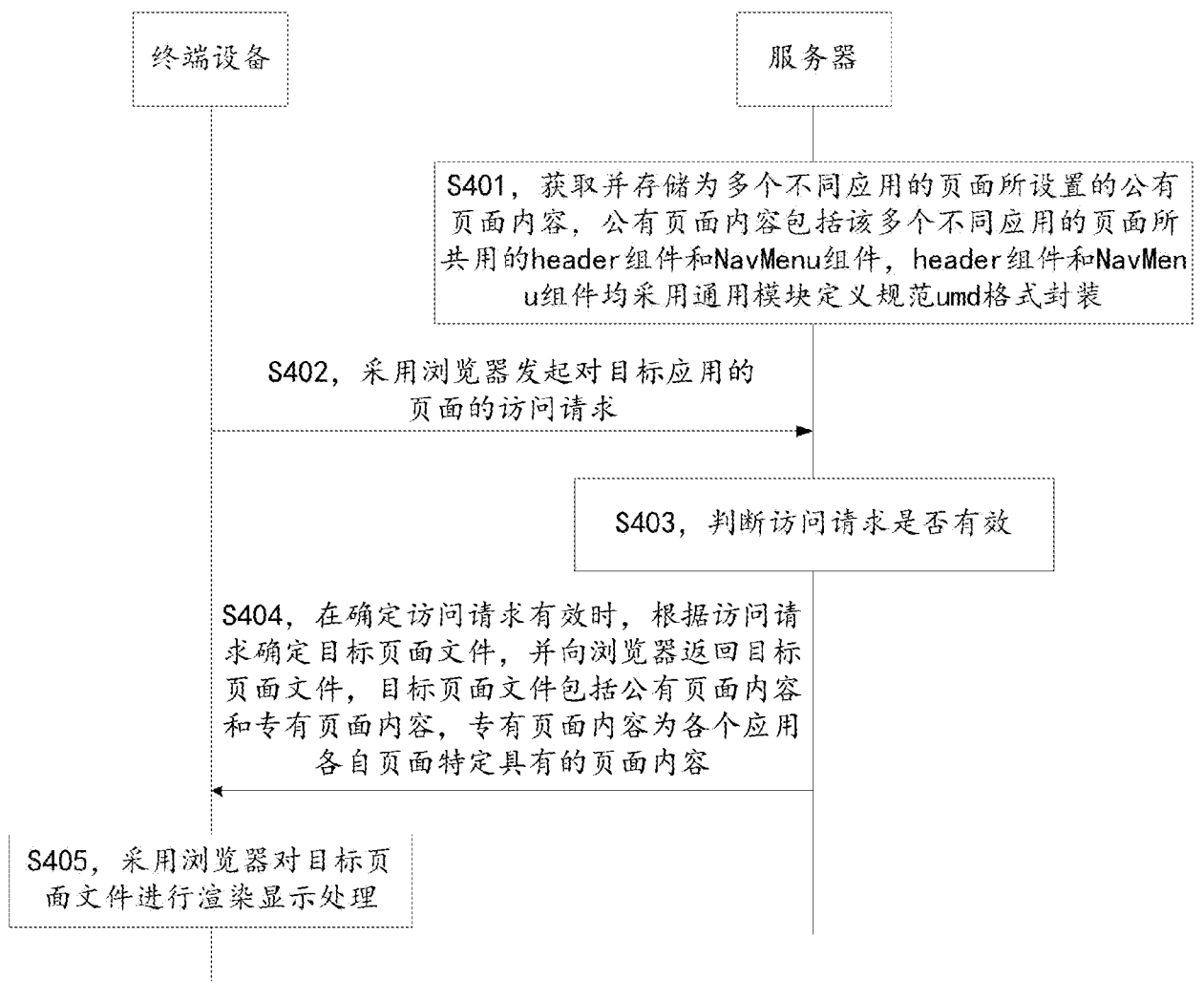


图 4



图 5



图 6

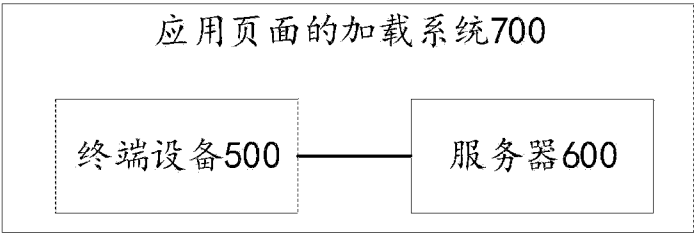


图 7

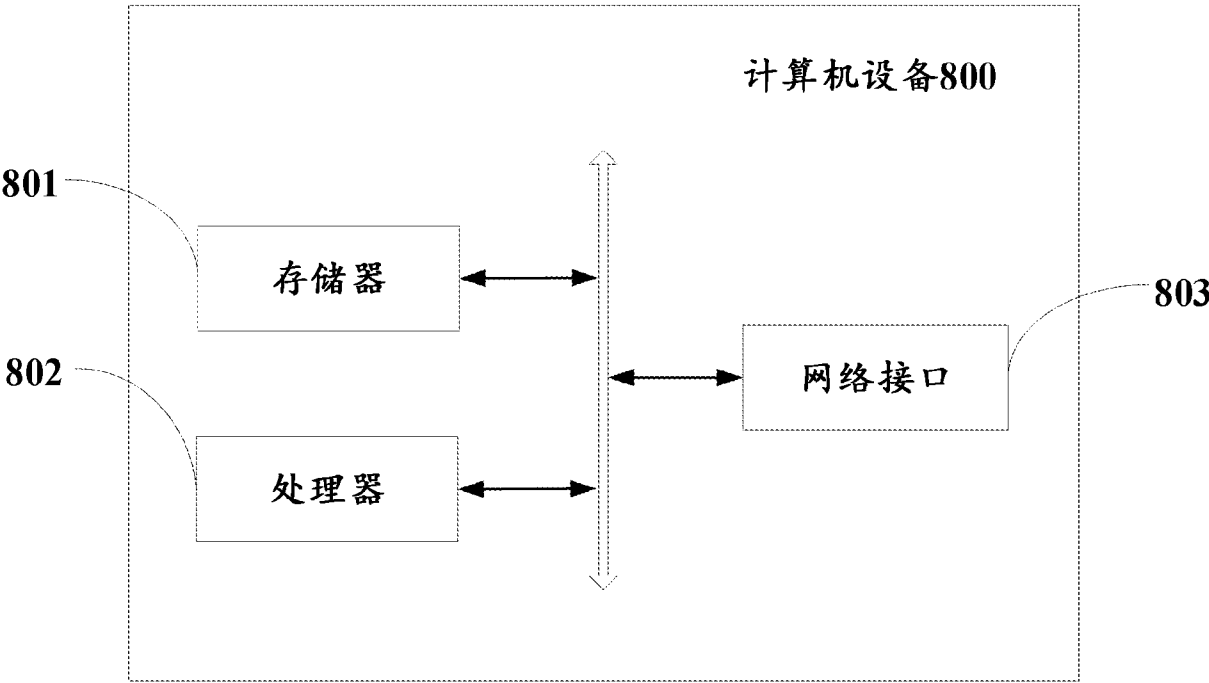


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/071877

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/958(2019.01)i; G06F 9/445(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

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)

CNKI, CNPAT, WPI, EPODOC, IEEE: 公有, 页面, 组件, 公共, 导航, 头部, 通用模块, common, page, component, navigation, head, universal module definition, umd

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 113536185 A (PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) 22 October 2021 (2021-10-22) claims 1-11, and description, paragraphs [0053]-[0141]	1-20
Y	CN 112817591 A (SHANDONG INSPUR GENESOFT INFORMATION TECHNOLOGY CO., LTD.) 18 May 2021 (2021-05-18) description, paragraphs [0045]-[0114], and figure 2	1-20
Y	CN 109491653 A (TAIKANG INSURANCE GROUP CO., LTD.) 19 March 2019 (2019-03-19) description, paragraph [0021]	1-20
Y	CN 111209028 A (JD DIGITS HOLDING CO., LTD.) 29 May 2020 (2020-05-29) description, paragraph [0172]	1-20
A	US 2020195705 A1 (ALIBABA GROUP HOLDING LIMITED) 18 June 2020 (2020-06-18) 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

12 April 2022

Date of mailing of the international search report

26 April 2022

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao, Haidian District, Beijing
100088, China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2022/071877

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	113536185	A	22 October 2021	None			
CN	112817591	A	18 May 2021	None			
CN	109491653	A	19 March 2019	None			
CN	111209028	A	29 May 2020	None			
US	2020195705	A1	18 June 2020	CN	111339455	A	26 June 2020
				WO	2020131756	A1	25 June 2020

国际检索报告

国际申请号

PCT/CN2022/071877

A. 主题的分类

G06F 16/958(2019.01)i; G06F 9/445(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

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))

CNKI, CNPAT, WPI, EP0DOC, IEEE: 公有, 页面, 组件, 公共, 导航, 头部, 通用模块, common, page, component, navigation, head, universal module definition, umd

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 113536185 A (平安科技深圳有限公司) 2021年10月22日 (2021 - 10 - 22) 权利要求1-11, 说明书第[0053]-[0141]段	1-20
Y	CN 112817591 A (山东浪潮通软信息科技有限公司) 2021年5月18日 (2021 - 05 - 18) 说明书第[0045]-[0114]段、图2	1-20
Y	CN 109491653 A (泰康保险集团股份有限公司) 2019年3月19日 (2019 - 03 - 19) 说明书第[0021]段	1-20
Y	CN 111209028 A (京东数字科技控股有限公司) 2020年5月29日 (2020 - 05 - 29) 说明书第[0172]段	1-20
A	US 2020195705 A1 (ALIBABA GROUP HOLDING LIMITED) 2020年6月18日 (2020 - 06 - 18) 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

2022年4月12日

国际检索报告邮寄日期

2022年4月26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

授权官员

刘长勇

电话号码 86-(10)-53961368

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2022/071877

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	113536185	A	2021年10月22日	无	
CN	112817591	A	2021年5月18日	无	
CN	109491653	A	2019年3月19日	无	
CN	111209028	A	2020年5月29日	无	
US	2020195705	A1	2020年6月18日	CN 111339455 A	2020年6月26日
				WO 2020131756 A1	2020年6月25日