



# (12) 发明专利申请

(10) 申请公布号 CN 117112940 A

(43) 申请公布日 2023. 11. 24

(21) 申请号 202310684897.3

(22) 申请日 2023.06.09

(71) 申请人 深圳市六度人和科技有限公司

地址 518000 广东省深圳市南山区粤海街道科技南十二路28号康佳研发大厦20A-C

(72) 发明人 吴剑文 张星亮

(74) 专利代理机构 深圳盛德大业知识产权代理

事务所(普通合伙) 44333

专利代理师 左光明

(51) Int. Cl.

G06F 16/957 (2019.01)

G06F 16/958 (2019.01)

G06F 9/451 (2018.01)

G06F 8/65 (2018.01)

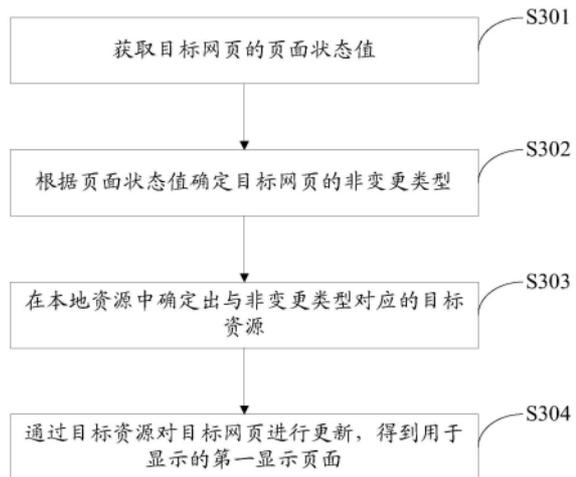
权利要求书2页 说明书14页 附图9页

## (54) 发明名称

一种离线数据显示方法、装置、电子设备及可读存储介质

## (57) 摘要

本发明适用于计算机应用领域,提供了一种离线数据显示方法、装置、电子设备及可读存储介质,方法包括:获取目标网页的页面状态值;根据所述页面状态值确定所述目标网页的非变更类型;在本地资源中确定出与所述非变更类型对应的目标资源;通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。通过目标网页的页面状态值确定目标网页的非变更类型,并在本地资源中确定出与非变更类型对应的目标类型,通过非变更类型对应的目标类型对目标网页进行更新,即得到更新后的显示页面,解决了在弱网或者无网环境下页面显示加载失败的问题。



1. 一种离线数据显示方法,其特征在于,包括以下步骤:  
获取目标网页的页面状态值;  
根据所述页面状态值确定所述目标网页的非变更类型;  
在本地资源中确定出与所述非变更类型对应的目标资源;  
通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。
2. 如权利要求1所述的离线数据显示方法,其特征在于,所述获取目标网页的页面状态值,包括:  
向服务端请求所述目标网页的页面状态值,以使所述服务端返回所述目标网页的页面状态值。
3. 如权利要求2所述的离线数据显示方法,其特征在于,所述根据所述页面状态值确定所述目标网页的非变更类型,包括:  
当所述页面状态值为第一预设值时,确定所述目标网页的非变更类型为UI层;  
当所述页面状态值为第二预设值时,确定所述目标网页的非变更类型为数据层;  
当所述页面状态值为第三预设值时,确定所述目标网页的非变更类型为UI层和数据层。
4. 如权利要求3所述的离线数据显示方法,其特征在于,所述在本地资源中确定出与所述非变更类型对应的目标资源,包括:  
当所述目标网页的非变更类型为UI层时,则在本地资源中确定出所述UI层对应的目标UI资源;  
当所述目标网页的非变更类型为数据层时,则在本地资源中确定出所述数据层对应的目标数据资源;  
当所述目标网页的非变更类型为UI层和数据层时,则在本地资源中确定出与所述UI层对应的目标UI资源,以及确定出与所述数据层对应的目标数据资源。
5. 如权利要求4所述的离线数据显示方法,其特征在于,所述通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面,包括:  
当确定出所述目标UI资源时,向服务端请求数据层对应的数据更新资源,并基于所述目标UI资源和所述数据更新资源对所述目标网页进行更新,得到用于显示的第一显示页面;  
当确定出所述目标数据资源时,向服务端请求UI层对应的UI更新资源,并基于所述目标数据资源和所述UI更新资源对目标网页进行更新,得到用于显示的第一显示页面。
6. 如权利要求1所述的离线数据显示方法,其特征在于,在所述通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面之后,所述方法还包括:  
当所述第一显示页面出现异常,通过原生获取所述目标页面的备用数据;  
基于所述备用数据对所述目标页面进行更新,得到用于显示的第二显示页面。
7. 一种离线数据显示装置,其特征在于,包括:  
第一获取模块,用于获取目标网页的页面状态值;  
第一确定模块,用于根据所述页面状态值确定所述目标网页的非变更类型;  
第二确定模块,用于在本地资源中确定出与所述非变更类型对应的目标资源;  
第一更新模块,用于通过所述目标资源对所述目标网页进行更新,得到用于显示的第

一显示页面。

8. 一种离线数据显示系统,其特征在于,所述系统包括服务端和APP端,所述APP端包括如权利要求7所述的离线数据显示装置。

9. 一种电子设备,包括存储器、处理器以及存储在所述存储器中并能够在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至6任一项所述的一种离线数据显示方法。

10. 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,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至6任一项所述的一种离线数据显示方法。

## 一种离线数据显示方法、装置、电子设备及可读存储介质

### 技术领域

[0001] 本发明属于计算机应用领域,尤其涉及一种离线数据显示方法、装置、电子设备及可读存储介质。

### 背景技术

[0002] 在移动应用程序中,通常需要与服务器进行数据交互,以获取最新的数据和更新应用程序的状态。然而,在某些情况下,如网络故障或服务器维护等情况下,应用程序可能无法访问服务器,从而无法获取最新的数据。在弱网或者无网的环境下webview页面直接打不开,显示加载失败页面,因此需要提前下载静态资源文件,提前下载静态资源文件会造成如果H5页面过多的情况下,下载速度会慢,占比物理内存过高的问题。

### 发明内容

[0003] 本发明实施例提供一种离线数据显示方法,旨在解决现有技术中,在弱网或者无网的环境下webview页面直接打不开,显示加载失败页面的问题。通过目标网页的页面状态值确定目标网页的非变更类型,并在本地资源中确定出与非变更类型对应的目标类型,通过非变更类型对应的目标类型对目标网页进行更新,即得到更新后的显示页面,解决了在弱网或者无网环境下页面显示加载失败的问题。

[0004] 本发明实施例是这样实现的,提供一种离线数据显示方法,包括以下步骤:

[0005] 获取目标网页的页面状态值;

[0006] 根据所述页面状态值确定所述目标网页的非变更类型;

[0007] 在本地资源中确定出与所述非变更类型对应的目标资源;

[0008] 通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。

[0009] 可选的,所述获取目标网页的页面状态值,包括:

[0010] 向服务端请求所述目标网页的页面状态值,以使所述服务端返回所述目标网页的页面状态值。

[0011] 可选的,所述根据所述页面状态值确定所述目标网页的非变更类型,包括:

[0012] 当所述页面状态值为第一预设值时,确定所述目标网页的非变更类型为UI层;

[0013] 当所述页面状态值为第二预设值时,确定所述目标网页的非变更类型为数据层;

[0014] 当所述页面状态值为第三预设值时,确定所述目标网页的非变更类型为UI层和数据层。

[0015] 可选的,所述在本地资源中确定出与所述非变更类型对应的目标资源,包括:

[0016] 当所述目标网页的非变更类型为UI层时,则在本地资源中确定出所述UI层对应的目标UI资源;

[0017] 当所述目标网页的非变更类型为数据层时,则在本地资源中确定出所述数据层对应的目标数据资源;

[0018] 当所述目标网页的非变更类型为UI层和数据层时,则在本地资源中确定出与所述

UI层对应的目标UI资源,以及确定出与所述数据层对应的目标数据资源。

[0019] 可选的,所述通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面,包括:

[0020] 当确定出所述目标UI资源时,向服务端请求数据层对应的数据更新资源,并基于所述目标UI资源和所述数据更新资源对所述目标网页进行更新,得到用于显示的第一显示页面;

[0021] 当确定出所述目标数据资源时,向服务端请求UI层对应的UI更新资源,并基于所述目标数据资源和所述UI更新资源对目标网页进行更新,得到用于显示的第一显示页面。

[0022] 可选的,在所述通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面之后,所述方法还包括:

[0023] 当所述第一显示页面出现异常,通过原生获取所述目标页面的备用数据;

[0024] 基于所述备用数据对所述目标页面进行更新,得到用于显示的第二显示页面。

[0025] 本发明实施例还提供一种离线数据显示装置,包括:

[0026] 第一获取模块,用于获取目标网页的页面状态值;

[0027] 第一确定模块,用于根据所述页面状态值确定所述目标网页的非变更类型;

[0028] 第二确定模块,用于在本地资源中确定出与所述非变更类型对应的目标资源;

[0029] 第一更新模块,用于通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。

[0030] 本发明实施例还提供一种离线数据显示系统,包括服务端和APP端,所述APP端包括本发明实施例中所述的离线数据显示装置。

[0031] 本发明实施例还提供一种电子设备,包括存储器、处理器以及存储在所述存储器中并能够在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现如任一实施例中所述的一种离线数据显示方法。

[0032] 本发明实施例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,所述计算机程序被处理器执行时实现如任一实施例中所述的一种离线数据显示方法。

[0033] 本发明所达到的有益效果,本申请通过获取目标网页的页面状态值;根据所述页面状态值确定所述目标网页的非变更类型;在本地资源中确定出与所述非变更类型对应的目标资源;通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。通过目标网页的页面状态值确定目标网页的非变更类型,并在本地资源中确定出与非变更类型对应的目标类型,通过非变更类型对应的目标类型对目标网页进行更新,即得到更新后的显示页面,解决了在弱网或者无网环境下页面显示加载失败的问题。

## 附图说明

[0034] 图1为本申请实施例提供的一种离线数据显示系统的架构图

[0035] 图2为本申请实施例提供的一种离线数据显示方法的流程图;

[0036] 图3为本申请实施例提供的另一种离线数据显示方法的流程图;

[0037] 图4为本申请实施例提供的另一种离线数据显示方法的流程图;

[0038] 图5为本申请实施例提供的图3中步骤S302的流程图;

- [0039] 图6为本申请实施例提供的图3中步骤S303的流程图；
- [0040] 图7为本申请实施例提供的图3中步骤S304的流程图；
- [0041] 图8为本申请实施例提供的web页面更新的流程图；
- [0042] 图9为本申请实施例提供的一种数据存储的结构示意图；
- [0043] 图10为本申请实施例提供的另一种离线数据显示方法的流程图；
- [0044] 图11为本申请实施例提供的一种离线数据显示装置的结构示意图；
- [0045] 图12为本申请实施例提供的图6中第一确定模块的结构示意图；
- [0046] 图13为本申请实施例提供的图6中第二确定模块的结构示意图；
- [0047] 图14为本申请实施例提供的图6中第一更新模块的结构示意图；
- [0048] 图15为本申请实施例提供另一种离线数据显示装置的结构示意图；
- [0049] 图16为本申请实施例提供的一种电子设备的结构示意图。

### 具体实施方式

[0050] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

#### [0051] 实施例一

[0052] 图1示出了本申请实施例提供的一种离线数据显示系统的架构图。离线数据显示系统包括服务端和APP端,当APP启动时初始化websocket长连接,该长连接主要获取页面状态值和监听页面状态值的变更;页面状态值总共有四种:①state:0,表示该页面没有本地缓存,需要全量加载该页面,也就是需要联网以正常的方式打开该页面;②state:1,表示数据层需要更新,UI层不变,UI层使用本地;③state:2,表示UI层需要更新,数据层数据不变,数据层使用本地;④state:3,表示数据和UI都不需要更新,全部直接使用本地资源文件。

[0053] 利用webview发送请求向服务端获取页面所有状态,并且将页面状态存储在本地;存储方式以kv方式存储,k为key,k可以是是页面url地址,v为value,v为状态值。

[0054] 上述WebSocket是一种在单个TCP连接上提供全双工通信的协议。它允许在浏览器和服务器之间建立一个持久化的连接,从而使得服务器可以主动地将数据推送到浏览器中。在建立WebSocket连接之后,服务器可以直接向客户端发送数据,而不需要等待客户端的请求。

[0055] 上述kv代表键值存储,是指一种数据存储方式,其中每个数据项都由一个唯一的键和一个相关的值组成。这种数据存储方式通常用于需要快速读取和写入数据的应用程序,例如缓存、配置文件和日志记录。数据的存储方式是键值存储,即每个数据项都由一个唯一的键和一个相关的值组成。上述key代表的是键,键是页面URL地址;值是状态,值指的是该页面的状态。

[0056] 上述页面URL地址指的是统一资源定位符(URL),它是一个用于标识互联网上特定资源位置的字符串。URL通常由协议、主机名和资源路径组成。例如,以下是一个URL地址:https://workec.com/getinfo.xml;在这个URL中,https是协议,workec.com是主机名,getinfo.xml是资源路径。通常,URL地址用于在Web浏览器中打开Web页面,但它们也可以用于其他目的,例如在Web应用程序中获取数据或执行其他操作。

[0057] 当打开webview页面时,根据页面url地址获取状态,如果状态值(state)为:

[0058] state:0,表示本地没有缓存,需要全量打开该页面,也就是需要以联网的正常方式打开该网页;

[0059] state:1时,表示数据层需要更新,UI不变,这时页面会加载本地资源,然后发送网络请求获取最新数据,数据返回时再刷新UI,从用户角度上页面会瞬间打开然后等待最新数据返回时再刷新UI,这时状态值会更新为state:3;

[0060] state:2时,表示数据不变,只是UI做了改动,这时页面会先加载本地资源,然后向服务端回去最新的UI资源文件,UI资源文件获取成功之后页面再做替换,从用户角度上页面会瞬间打开,等待最新的UI资源文件就在成功之后显示最新的UI样式,这时状态值会更新为stat:3;

[0061] state:3时,表示数据层和UI都不需要改变,直接使用本地资源文件。

[0062] 在本发明实施例中,当前端页面做了修改时,会通过websocket通知APP端做状态变更,如果数据层变更了则会将url地址和state:1通知APP端,如果是UI层变更了则会将url地址和state:2通知APP端,APP端收到状态变更通知本地状态存储的url对应的state做更改,当打开websocket页面时会从本地状态中获取最新的state做逻辑处理。逻辑处理就是根据state:n时做的相应的处理。

[0063] 实施例二

[0064] 在本实施例中,基于上述实施例一,如图2所示,图2为本申请实施例提供的一种离线数据显示方法的流程图。具体的,当APP启动时,打开websocket,APP启动时初始化websocket长连接;利用websocket发送请求向服务端获取页面所有状态,并且将页面状态存储在本地;当前端页面做了修改时,会通过websocket通知APP端做状态变更。

[0065] 上述长连接主要获取页面状态值和监听页面状态值的变更。页面状态值指的是在应用程序对于每个页面都有一个状态值。根据该状态值,应用程序可以采取不同的策略来处理页面的加载和更新。

[0066] 上述WebSocket是一种在单个TCP连接上提供全双工通信的协议。它允许在浏览器和服务器之间建立一个持久化的连接,从而使得服务器可以主动地将数据推送到浏览器中。在建立WebSocket连接之后,服务器可以直接向客户端发送数据,而不需要等待客户端的请求。

[0067] 在本发明实施例中,页面状态值总共有四种:state:0,该页面没有本地缓存需要全量加载该页面;state:1,数据层需要更新,UI层使用本地;state:2,UI层需要更新,数据层使用本地;state:3,数据层UI层都不需要更新,全部使用本地资源。

[0068] 上述存储方式以kv方式存储,k为key,k可以是是页面url地址,v为value,v为状态值。上述kv代表键值存储,是指一种数据存储方式,其中每个数据项都由一个唯一的键和一个相关的值组成。这种数据存储方式通常用于需要快速读取和写入数据的应用程序,例如缓存、配置文件和日志记录。数据的存储方式是键值存储,即每个数据项都由一个唯一的键和一个相关的值组成。上述key代表的是键,键是页面URL地址;值是状态,值指的是该页面的状态。

[0069] 上述页面URL地址指的是统一资源定位符(URL),它是一个用于标识互联网上特定资源位置的字符串。URL通常由协议、主机名和资源路径组成。例如,以下是一个URL地址:

https://workec.com/getinfo.xml;在这个URL中,https是协议,workec.com是主机名,getinfo.xml是资源路径。通常,URL地址用于在Web浏览器中打开Web页面,但它们也可以用于其他目的,例如在Web应用程序中获取数据或执行其他操作。

[0070] 当打开webview页面时,根据页面url地址获取状态,如果状态值为:

[0071] state:0,表示本地没有缓存,需要全量打开该页面,也就是需要以联网的正常方式打开该网页;

[0072] state:1,表示数据层需要更新,UI不变,这时页面会加载本地资源,然后发送网络请求获取最新数据,数据返回时再刷新UI,从用户角度上页面会瞬间打开然后等待最新数据返回时再刷新UI,这时状态值会更新为state:3;

[0073] state:2,表示数据不变,只是UI做了改动,这时页面会先加载本地资源,然后向服务端回去最新的UI资源文件,UI资源文件获取成功之后页面再做替换,从用户角度上页面会瞬间打开,等待最新的UI资源文件就在成功之后显示最新的UI样式,这时状态值会更新为stat:3;

[0074] state:3,表示数据层和UI都不需要改变,直接使用本地资源文件。

[0075] 如果是数据层变更了则会将url地址和state:1通知APP端,APP端收到状态变更将本地状态存储的url对应的state:1做更改,当打开webview页面时会从本地状态存储中获取最新的数据资源,数据返回时再刷新UI,这时状态会更新为state:3。

[0076] 如果是UI层变更了则会将URL地址和state:2通知APP端;APP端收到状态变更将本地状态存储的url对应的state:2做更改,当打开webview页面时会从本地状态存储中获取最新的UI资源文件,UI资源文件获取成功之后在做替换,等待最新的UI资源文件加载成功之后显示最新的UI样式,这时状态会更新为state:3。

[0077] 在本发明实施例中,当前端页面做了修改时,会通过websocket通知APP端做状态变更,APP端收到状态变更通知则将本地状态存储的url对应的state做更改,当打开webview页面时会从本地状态存储中获取最新的state做相应的逻辑处理。逻辑处理就是根据state:n时做的相应的处理。

[0078] 实施例二

[0079] 在本实施例中,图3示出了本申请实施例提供的另一种离线数据显示方法的流程图。如图3所示,该方法包括步骤S301至S304。各个步骤的具体实现原理如下:

[0080] S301、获取目标网页的页面状态值。

[0081] 在本发明实施例中,上述网页可以是不同类型的网页,可以是静态网页、动态网页以及单页应用程序等网页。静态网页通常由HTML、CSS和JavaScript等静态文件组成。静态网页的内容在访问时不会发生变化,因为它们没有与服务器进行交互。通常,静态网页用于显示静态内容,如公司信息、新闻文章等。动态网页的内容在访问时会发生变化。这是因为动态网页的内容是从服务器上动态生成的,通常使用服务器端脚本语言(如PHP、ASP.NET等)来实现。动态网页通常用于显示个性化内容,如社交媒体、电子商务网站等。单页应用程序是一种使用JavaScript等技术来实现的Web应用程序,可以在不刷新页面的情况下实现内容的动态更新。单页面应用程序通常使用AJAX技术来从服务器检索数据,并使用客户端路由器来实现页面导航和内容更新。

[0082] 上述目标网页可以是应用程序中用户浏览的某一个网页,具体可以是用户浏览想

要访问的特定网页。

[0083] 在本发明实施例中,本发明网页具体可以是webview页面。本发明具体以webview页面举例说明。Webview是一种可以在应用程序中嵌入浏览器的技术。它可以使应用程序显示网页,同时也可以实现与网页交互。

[0084] 需要说明的是,网页页面包括数据层和UI层。数据层是指应用程序中与数据相关的部分,包括数据的存储、处理和管理等;数据层的主要职责是管理应用程序中的数据,并提供数据的访问接口;数据层通常由数据库和数据访问层组成,其中数据库负责数据的存储,数据访问层负责数据的管理和访问;在应用程序中,数据层提供了一种可靠地数据存储和访问方式,以确保应用程序的数据正确性和一致性。UI层是指应用程序中与用户界面相关的部分,包括用户界面的设计、布局和显示等;UI层的主要职责是处理用户的输入和输出,并将数据展示给用户;在应用程序中,UI层通常由界面设计、布局、控件和事件处理等组成,以提供良好的用户体验和交互性。

[0085] 上述页面状态值指的是在应用程序对于每个页面都有一个状态值。根据该状态值,应用程序可以采取不同的策略来处理页面的加载和更新。

[0086] 在本发明实施例中,页面状态值总共有四种:①state:0时,表示本地没有保存,需要全量加载页面;②state:1时,表示页面状态值可以是数据层需要更新,UI层使用本地;③state:2时,表示UI层需要更新,数据层使用本地;④state:3时,表示数据层和UI层都不需要更新,全部使用本地资源。

[0087] 可选的,获取目标网页的页面状态值的步骤包括:

[0088] 向服务端请求目标网页的页面状态值,以使服务端返回目标网页的页面状态值。

[0089] 在本发明实施例中,上述服务端是指在web应用程序中,运行在服务器上的程序,用于处理客户端的请求,生成响应并返回给客户端。具体来说,服务端会根据应用程序的请求,查找页面状态值,然后将状态值作为响应返回给应用程序。

[0090] 在本发明实施例中,获取目标网页的页面状态值是为了确定当前正在使用的页面状态,以便于应用程序可以采取不同的策略来处理页面的加载和更新,以确保页面的正确性和最新性。

[0091] S302、根据页面状态值确定目标网页的非变更类型。

[0092] 在本发明实施例中,上述非变更类型是指页面的内容和状态不会发生任何变化的类型。非变更类型可以是UI层,也可以是数据层。

[0093] 上述页面状态值总共有四种,①state:0时,表示本地没有保存,需要全量加载页面;②state:1时,表示页面状态值可以是数据层需要更新,UI层使用本地;③state:2时,表示UI层需要更新,数据层使用本地;④state:3时,表示数据层和UI层都不需要更新,全部使用本地资源。

[0094] 在一种可能的实施例中,当state:1时,表示该页面数据层需要更新,UI层使用本地。根据页面状态值确定目标网页的非变更类型为UI层。

[0095] 在另一种可能的实施例中,当state:2时,表示UI层需要更新,数据层使用本地。根据页面状态值确定目标网页的非变更类型为数据层。

[0096] 在另一种可能的实施例中,当state:3时,表示数据层和UI层都不需要更新,全部使用本地资源。根据页面状态值确定目标网页的非变更类型为数据层和UI层。

[0097] 在本发明实施例中,用户可以通过网页的状态值确定目标网页是否为非变更类型,这种类型的页面和状态已经存储到本地资源,应用程序可以直接使用本地资源页面。

[0098] S303、在本地资源中确定出与非变更类型对应的目标资源。

[0099] 在本发明实施例中,上述本地资源可以是应用程序在本地设备上存储的资源。在应用程序中,为了提高数据访问的效率和响应速度,通常会将一些常用的数据存储在本地设备上,以便快速访问。存储数据可以是文件、图片、音频、视频等各种类型的资源。使用本地存储数据可以减少应用程序对网络的依赖,提高应用程序的性能和用户体验。

[0100] 上述目标资源是应用程序中与页面状态值的页面所对应的本地资源。

[0101] 在一种可能的实施例中,当state:1时,目标资源可以是应用程序中与state:1的页面对应的本地UI资源,则数据层发生改变需要更新。

[0102] 在另一种可能的实施例中,当state:2时,上述目标资源可以是应用程序中与state:2的页面对应的本地数据资源,则UI层发生改变需要更新。

[0103] 在另一种可能的实施例中,当state:3时,上述目标资源可以是应用程序中与state:3的页面对应的本地数据资源和UI资源,则数据层和UI层不会发生任何改变。

[0104] 需要说明的是,UI资源是用于构建UI层的所有资源,例如htkl、css和jst文件等。数据资源是用于存储数据的所有资源,例如数据库、文件和js接口等。

[0105] 具体的,在本地资源中确定出UI层对应的目标UI层资源。在本地资源中确定出数据层对应的目标数据层资源。在本地资源中确定出数据层和UI层对应的目标数据层资源和UI层资源。

[0106] 在本发明实施例中,需要在本地资源中确定出非变更类型对应的目标资源。根据页面状态值的不同,应用程序可以采取不同的策略来处理页面的加载和更新,以确保页面的正确性和最新性。

[0107] S304、通过目标资源对目标网页进行更新,得到用于显示的第一显示页面。

[0108] 在本发明实施例中,上述目标资源是应用程序与页面状态值的页面所对应的本地资源。上述目标网页可以是应用程序中用户浏览的某一个网页,具体可以是用户浏览想要访问的特定网页。

[0109] 在一种可能的实施例中,当state:1时,表示该页面数据层需要更新,UI层使用本地。根据页面状态值确定目标网页的非变更类型为UI层。目标资源可以是应用程序中与state:1的页面对应的本地UI层资源,则数据层发生改变需要更新。

[0110] 在另一种可能的实施例中,当state:2时,表示UI层需要更新,数据层使用本地。根据页面状态值确定目标网页的非变更类型为数据层。目标资源可以是应用程序中与state:2的页面对应的本地数据层资源,则UI层发生改变需要更新。

[0111] 在本发明实施例,获取目标网页的页面状态值;根据所述页面状态值确定所述目标网页的非变更类型;在本地资源中确定出与所述非变更类型对应的目标资源;通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。通过目标网页的页面状态值确定目标网页的非变更类型,并在本地资源中确定出与非变更类型对应的目标类型,通过非变更类型对应的目标类型对目标网页进行更新,即得到更新后的显示页面,解决了在弱网或者无网环境下页面显示加载失败的问题。

[0112] 实施例四

[0113] 在本实施例中,请参见图4,图4为本申请实施例提供的另一种离线数据显示方法的流程图。具体的,APP打开后,启动webview时,如果有本地资源首先会加载本地资源;然后会启动js接收数据接口;判断数据层是否有变更,当数据有变更时,会请求最新的数据,得到数据返回时将数据未给js接口,js接口收到数据时做页面数据更新;判断UI是否有变更,当UI需要变更时,会请求最新的最新页面资源文件,请求成功之后刷新页面;UI变更时只是加载UI资源,由于数据层没有变化,所以数据层使用本地,这相比静态资源(UI和数据)速度要快很多。

[0114] 上述js接收数据接口主要是有两个任务:数据更新和UI更新;js接收数据接口主要有两个参数:参数1(url地址或者命令字),参数2(json数据集);这按照接口模块化进行,页面模块多的情况下没必要真个页面的数据都获取一次,哪一个模块有更新就只需要更新哪一个模块。

[0115] 在本发明实施例中,上述js接收数据接口是指在JavaScript代码中使用的一种机制,该机制允许JavaScript代码从Web服务器中获取数据。通过使用json数据格式,服务器可以将数据以一种易于阅读和理解的方式传输到JavaScript代码中。JavaScript接收数据接口使得JavaScript代码可以从服务器中获取数据,并将其用于Web应用程序中。上述json数据集通常用于在JavaScript代码和服务器之间传输数据。例如,当用户通过app端请求从服务器获取数据时,服务器会将数据作为json数据集返回给用户,然后用户可以在JavaScript代码中使用这些数据。

[0116] 实施例五

[0117] 在本实施例中,基于上述实施例三,如图5所示,步骤S302包括步骤S3021至S3023。各个步骤的具体实现原理如下:

[0118] S3021、当页面状态值为第一预设值时,确定目标网页的非变更类型为UI层。

[0119] 在本发明实施例中,APP端中每个页面对应一个页面状态值。上述页面状态值总共有四种:①state:0时,表示本地没有保存,需要全量加载页面;②state:1时,表示页面状态值可以是数据层需要更新,UI层使用本地;③state:2时,表示UI层需要更新,数据层使用本地;④state:3时,表示数据层和UI层都不需要更新,全部使用本地资源。

[0120] 上述非变更类型是指页面的内容和状态不会发生任何变化的数据类型,进一步可以理解为不需要联网下载的数据类型。非变更类型可以是UI层,也可以是数据层。

[0121] 上述预设值可以是系统或程序中预先定义好的默认值。

[0122] 上述第一预设值具体可以是state:1,state:1表示数据层需要更新,UI层不变。

[0123] 具体的,当页面状态值为state:1时,根据页面状态值确定目标网页的非变更类型为UI层。

[0124] S3022、当页面状态值为第二预设值时,确定目标网页的非变更类型为数据层。

[0125] 在本发明实施例中,上述第二预设值具体可以是state:2,state:2表示UI层需要更新,数据层不变。

[0126] 具体的,当页面状态值为state:2时,根据页面状态值确定目标网页的非变更类型为数据层。

[0127] S3023、当页面状态值为第三预设值时,确定目标网页的非变更类型为UI层和数据层。

[0128] 在本发明实施例中,上述第三预设值具体可以是state:3,state:3表示数据层和UI层都不需要更新,全部使用本地资源。

[0129] 具体的,当页面状态值为state:3,根据页面状态值确定目标网页的非变更类型为UI层和数据层。

[0130] 实施例六

[0131] 在本实施例中,基于上述实施例二,如图6所示,步骤S303包括步骤S3031至S3033。各个步骤的具体实现原理如下:

[0132] S3031、当目标网页的非变更类型为UI层时,则在本地资源中确定出UI层对应的目标UI资源。

[0133] 在本发明实施例中,上述本地资源指在用户设备上存储的资源,可以是用户的计算机、手机或平板电脑等。上述目标网页可以是用户浏览的一个网页,具体可以是用户浏览想要访问的特定网页。上述非变更类型是指页面的内容和状态不会发生任何变化的类型。上述UI层是指应用程序中与用户界面相关的部分,包括用户界面的设计、布局和显示等。

[0134] 当目标网页的非变更类型为UI层时,则目标网页对应的页面状态值state:1。上述目标UI资源是指应用程序中与state:1的UI层对应的本地UI层资源。UI层内容和状态不会发生改变,应用程序可以直接使用本地资源中与目标网页对应的UI层资源。

[0135] 在本发明实施例中,当目标网页的非变更类型为UI层,则在本地资源中确定出UI层对应的目标UI资源。以便应用程序可以快速访问,提高应用程序的性能。

[0136] S3032、当目标网页的非变更类型为数据层时,则在本地资源中确定出数据层对应的目标数据资源。

[0137] 在本发明实施例中,上述数据层指应用程序中与数据相关的部分,包括数据的存储、处理和管理等。

[0138] 当目标网页的非变更类型为数据层时,则目标网页对应的页面状态值state:2。上述目标数据资源是指应用程序中与state:2的数据层对应的本地数据层资源。数据层内容和状态不会发生改变,应用程序可以直接使用本地资源中与目标页面对应的数据层资源。

[0139] 在本发明实施例中,当目标网页的非变更类型为数据层时,则在本地资源中确定出数据层对应的目标数据层资源。以便应用程序可以快速访问内容,提高应用程序的性能。

[0140] S3033、当目标网页的非变更类型为UI层和数据层时,则在本地资源中确定出与UI层对应的目标UI资源,以及确定出与数据层对应的目标数据资源。

[0141] 在本发明实施例中,当目标网页的非变更类型为UI层和数据层时,则目标网页对应的页面状态值为state:3。上述目标UI资源是指应用程序中与state:3的UI层对应的本地UI资源;上述目标数据资源是指应用程序中与页面状态值=3的数据层对应的本地数据层资源。

[0142] 在本发明实施例中,当目标网页的非变更类型为UI层和数据层时,则在本地资源中确定出UI层对应的目标UI资源,以及确定出数据层对应的目标数据资源。当UI层和数据层不需要变更时,可以直接使用本地资源,而不需要从服务器请求数据,这样可以提高页面加载速度。

[0143] 实施例七

[0144] 在本实施例中,基于上述实施例二,如图7所示,步骤S304包括步骤S3041至S3042。

各个步骤的具体实现原理如下：

[0145] S3041、当确定出目标UI资源时，向服务端请求数据层对应的数据更新资源，并基于目标UI资源和数据更新资源对目标网页进行更新，得到用于显示的第一显示页面。

[0146] 在本发明实施例中，上述目标UI资源是指应用程序中与state:1的UI层对应的本地UI层资源。UI层内容和状态不会发生改变，应用程序可以直接使用本地资源的UI层资源。

[0147] 具体的，当确定出目标UI资源时，则目标网页的非变更类型为UI层，目标网页对应的页面状态值state:1。state:1表示数据层需要更新，UI层不变。

[0148] 具体的，当确定出目标UI资源时，这时页面会先加载本地资源，然后向服务端请求数据层对应的数据更新资源，在数据返回之前，页面会等待，不会刷新UI层，直到数据返回后，应用程序会使用最新的数据更新页面的数据层，然后刷新UI层，这样就可以得到更新数据的显示页面。

[0149] S3042、当确定出目标数据资源时，向服务端请求UI层对应的UI更新资源，并基于目标数据资源和UI更新资源对目标网页进行更新，得到用于显示的第一显示页面。

[0150] 在本发明实施例中，上述目标数据资源是指应用程序中与state:2的数据层对应的本地数据层资源。数据层内容和状态不会发生改变，应用程序可以直接使用本地资源的数据层资源。

[0151] 当确定出目标数据资源时，则目标网页的非变更类型为数据层，目标网页对应的页面状态值state:2。state:2表示UI层需要更新，数据层不变。

[0152] 具体的，当确定出目标数据资源时，这时页面会先加载本地资源，然后服务端请求UI层对应的UI更新资源，UI资源文件获取成功之后页面再做替换，并对网页更新，更新之后得到最新的UI样式的显示页面。

[0153] 在一中可能的实施例中，上述目标UI资源可以是指应用程序中与state:3的UI层对应的本地UI资源；上述目标数据资源可以是指应用程序中与页面状态值=3的数据层对应的本地数据层资源。当确定出目标UI资源和目标数据资源时，这时页面加载本地资源，然后直接全部使用本地资源文件，得到本地资源显示页面。

[0154] 可选的，在步骤S304之后，还包括：

[0155] 当第一显示页面出现异常，通过原生获取目标页面的备用数据。

[0156] 基于备用数据对目标页面进行更新，得到用于显示的第二显示页面。

[0157] 在本发明实施例中，上述原生可以理解为应用程序的本地代码，即使用原生语言（如Java、Objective-C等）编写的代码。

[0158] 上述备用数据可以是在主要数据不可用时使用的替代数据。通常，备用数据应该是相似的，但不一定是完全相同的，以主要数据为基础。备用数据通常用于确保在主要数据不可用时，您的网站或应用程序仍然可以正常运行。

[0159] 在本发明实施例中，当第一显示页面异常的情况下，可以通过使用备用数据对目标页面进行更新，以得到用于显示的第二显示页面。第二显示页面是基于备用数据而创建的，并且应该足够相似，以使用户可以继续使用网站或应用程序。

[0160] 实施例八

[0161] 在本实施例中，请参见图8，图8为本申请实施例提供的web页面更新的结构示意图。具体的，一个web页面有多个模块，每个模块通过一个接口获取json数据集合，这个称为

json数据模块化;在本地存储以xml方式进行物理存储,key为请求接口url或者命令字,value为返回的json数据;一个页面以一个xml文件,xml的命令方式为url页面地址,例如:  
https://workec.com/getinfo.xml,

[0162] https://workec.com/getinfo这个就是个人资料页面的web地址,xml为xml文件后缀;webview定义调用js函数,setJsonData(String url,String strJson),主要负责把本地数据传递到web页中,url表示数据模块接口,strJson表示数据模块接口对应的数据集;通过setJsonData接口将数据传递给web页面进行页面数据更新;定义web页面回调接口getJsonData,该主要负责当web页面数据出现异常需要向原生获取数据的场景下使用,接口参数与setJsonData一致,获取完数据再调用getJsonData接口;可以认为getJsonData是原生主动调用,而setJsonData响应时被动调用getJsonData接口。

[0163] 实施例九

[0164] 在本实施例中,请参见图9,图9为本申请实施例提供的一种数据存储的结构示意图。具体的,本地资源中,页面资源存储和数据存储都是放在app私有目录中,从安全性考虑,外部app是无法访问的;以web地址url作为文件夹名称,这个文件夹下全是这个页面所需要用到的页面资源(UI)和数据资源;数据资源以xml文件存储,一个页面以一个xml文件,xml的命令方式为url页面地址,例如:https://workec.com/getinfo.xml,

[0165] https://workec.com/getinfo这个就是个人资料页面的web地址,xml为xml文件后缀;资源文件包括html文件、css、js文件;图片、媒体等资源文件夹;如果数据或页面资源有更新,就在该文件夹下进行资源和数据更新。

[0166] 实施例十

[0167] 在本实施例中,请参见图10,图10为本申请实施例提供的另一种离线数据显示方法的结构示意图。具体的,上述离线数据为web页面离线数据,web页面离线数据显示方法主要包括页面UI资源的离线、页面数据离线、页面全量离线三种方案;本地对每个web页面做状态state管理,state变更由websocket通知;离线需要是将数据、UI资源进行分类存储;并且提供js方法回调函数,该方法主要是web页面数据一次需要主动向native获取数据;原生主要到web页面提供数据;如果页面数据没有更新,就可以将原来的web获取到的数据传递给web页面;或者数据有变更,这时原生会请求到最新的数据传递给页面进行数据刷新;页面变更请求,当页面UI有变更,这时原生会请求最新的UI资源对页面进行更新;页面刷新机制,无论是数据有变更还是UI有变更都需要刷新页面,这个刷新只是局部的包括数据、UI;而不是向全量加载那样有高耗时现象。

[0168] 实施例十一

[0169] 对应于上述图3所示的一种离线数据显示方法,图11示出的是本申请实施例提供的一种离线数据显示装置的示意图,装置M110包括:

[0170] 第一获取模块M1101,用于获取目标网页的页面状态值;

[0171] 第一确定模块M1102,用于根据所述页面状态值确定所述目标网页的非变更类型;

[0172] 第二确定模块M1103,用于在本地资源中确定出与所述非变更类型对应的目标资源;

[0173] 第一更新模块M1104,用于通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。

[0174] 可选的,第一获取模块M1101还用于向服务端请求所述目标网页的页面状态值,以使所述服务端返回所述目标网页的页面状态值。

[0175] 可选的,参考图12所示,图12为本申请实施例提供的第一确定模块的结构示意图。第一确定模块M1102包括:

[0176] 第一确定子模块M11021,用于当所述页面状态值为第一预设值时,确定所述目标网页的非变更类型为UI层;

[0177] 第二确定子模块M11022,用于当所述页面状态值为第二预设值时,确定所述目标网页的非变更类型为数据层;

[0178] 第三确定子模块M11023,用于当所述页面状态值为第三预设值时,确定所述目标网页的非变更类型为UI层和数据层。

[0179] 可选的,参考图13所示,图13为本申请实施例提供的第二确定模块的结构示意图。第二确定模块M1103包括:

[0180] 第四确定子模块M11031,用于当所述目标网页的非变更类型为UI层时,则在本地资源中确定出所述UI层对应的目标UI资源;

[0181] 第五确定子模块M11032,用于当所述目标网页的非变更类型为数据层时,则在本地资源中确定出所述数据层对应的目标数据资源;

[0182] 第六确定子模块M11033,用于当所述目标网页的非变更类型为UI层和数据层时,则在本地资源中确定出与所述UI层对应的目标UI资源,以及确定出与所述数据层对应的目标数据资源。

[0183] 可选的,参考图14所示,图14为本申请实施例提供的第一更新模块的结构示意图。第一更新模块M604包括:

[0184] 第一更新子模块M11041,用于当确定出所述目标UI资源时,向服务端请求数据层对应的数据更新资源,并基于所述目标UI资源和所述数据更新资源对所述目标网页进行更新,得到用于显示的第一显示页面;

[0185] 第二更新子模块M11042,用于当确定出所述目标数据资源时,向服务端请求UI层对应的UI更新资源,并基于所述目标数据资源和所述UI更新资源对目标网页进行更新,得到用于显示的第一显示页面。

[0186] 可选的,参考图15所示,本申请实施例提供的另一种离线数据显示装置的示意图,装置M110还包括:

[0187] 第二获取模块M1105,用于当所述第一显示页面出现异常,通过原生获取所述目标页面的备用数据;

[0188] 第二更新模块M1106,用于基于所述备用数据对所述目标页面进行更新,得到用于显示的第二显示页面。

[0189] 实施例十二

[0190] 图16为本申请实施例提供的一种电子设备的结构示意图。如图16所示,该实施例的电子设备D16包括:至少一个处理器D160(图16中仅示出一个)、存储器D161、网络接口D162,以及存储在存储器D161中并可在至少一个处理器D160上运行的计算机程序。其中,处理器D160用于调用存储器D161存储的计算机程序,并执行如下步骤:

[0191] 获取目标网页的页面状态值;

- [0192] 根据所述页面状态值确定所述目标网页的非变更类型；
- [0193] 在本地资源中确定出与所述非变更类型对应的目标资源；
- [0194] 通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面。
- [0195] 可选的,处理器D160执行的所述获取目标网页的页面状态值,包括:
- [0196] 向服务端请求所述目标网页的页面状态值,以使所述服务端返回所述目标网页的页面状态值。
- [0197] 可选的,处理器D160执行的所述根据所述页面状态值确定所述目标网页的非变更类型,包括:
- [0198] 当所述页面状态值为第一预设值时,确定所述目标网页的非变更类型为UI层;
- [0199] 当所述页面状态值为第二预设值时,确定所述目标网页的非变更类型为数据层;
- [0200] 当所述页面状态值为第三预设值时,确定所述目标网页的非变更类型为UI层和数据层。
- [0201] 可选的,处理器D160执行的所述在本地资源中确定出与所述非变更类型对应的目标资源,包括:
- [0202] 当所述目标网页的非变更类型为UI层时,则在本地资源中确定出所述UI层对应的目标UI资源;
- [0203] 当所述目标网页的非变更类型为数据层时,则在本地资源中确定出所述数据层对应的目标数据资源;
- [0204] 当所述目标网页的非变更类型为UI层和数据层时,则在本地资源中确定出与所述UI层对应的目标UI资源,以及确定出与所述数据层对应的目标数据资源。
- [0205] 可选的,处理器D160执行的所述通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面,包括:
- [0206] 当确定出所述目标UI资源时,向服务端请求数据层对应的数据更新资源,并基于所述目标UI资源和所述数据更新资源对所述目标网页进行更新,得到用于显示的第一显示页面;
- [0207] 当确定出所述目标数据资源时,向服务端请求UI层对应的UI更新资源,并基于所述目标数据资源和所述UI更新资源对目标网页进行更新,得到用于显示的第一显示页面。
- [0208] 可选的,在所述通过所述目标资源对所述目标网页进行更新,得到用于显示的第一显示页面之后,处理器D160执行的所述方法还包括:
- [0209] 当所述第一显示页面出现异常,通过原生获取所述目标页面的备用数据;
- [0210] 基于所述备用数据对所述目标页面进行更新,得到用于显示的第二显示页面。
- [0211] 本发明实施例提供的电子设备D16能够实现一种离线数据显示方法实施例中的各个实施方式,以及相应有益效果,为避免重复,这里不再赘述。
- [0212] 需要指出的是,图中仅示出了具有组件的D160-D162,但是应理解的是,并不要求实施所有示出的组件,可以替代的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,这里的电子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。
- [0213] 所述处理器D160在一些实施例中可以是中央处理单元(Central Processing Unit,CPU),该处理器D160还可以是其它通用处理器、数字信号处理器(Digital Signal

Processor,DSP)、专用集成电路(Application Specific Integrated Circuit,ASIC)、现成可编程门阵列(Field-Programmable Gate Array,FPGA)或者其它可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。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。

[0214] 所述存储器D161在一些实施例中可以是所述电子设备D16的内部存储单元,例如电子设备D16的硬盘或内存。所述存储器D161在另一些实施例中也可以是所述电子设备D16的外部存储设备,例如所述电子设备D16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。进一步地,所述存储器D161还可以既包括所述电子设备D16的内部存储单元也包括外部存储设备。所述存储器D161用于存储操作系统、应用程序、引导装载程序(BootLoader)、数据以及其它程序等,例如所述计算机程序的程序代码等。所述存储器D161还可以用于暂时地存储已经输出或者将要输出的数据。

[0215] 所述网络接口D162可包括无线网络接口或有线网络接口,该网络接口D162通常用于在电子设备D16与其他电子设备之间建立通信连接。

[0216] 本申请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,该计算机程序被处理器D160执行时实现本发明实施例提供的一种离线数据显示方法实施例的各个程,且能达到相同的技术效果,为避免重复,这里不再赘述。

[0217] 所述集成的单元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,可以存储在一个计算机可读存储介质中。基于这样的理解,本申请实现上述实施例方法中的全部或部分流程,可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机可读存储介质中,该计算机程序在被处理器执行时,可实现上述各个方法实施例的步骤。其中,所述计算机程序包括计算机程序代码,所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质至少可以包括:能够将计算机程序代码携带到拍照装置/终端设备的任何实体或装置、记录介质、计算机存储器、只读存储器(Read-Only Memory,ROM)、随机存取存储器(Random Access Memory,RAM)、电载波信号、电信信号以及软件分发介质。例如U盘、移动硬盘、磁碟或者光盘等。

[0218] 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。

[0219] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等,均应包含在本发明的保护范围之内。

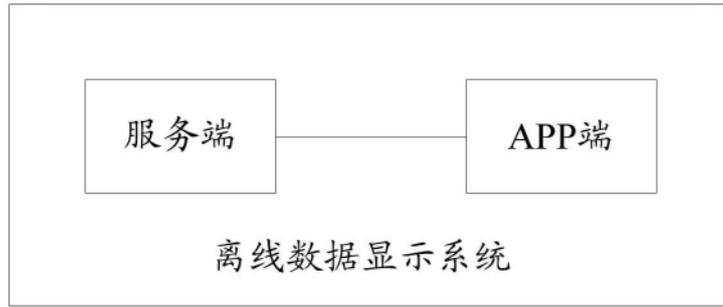


图1

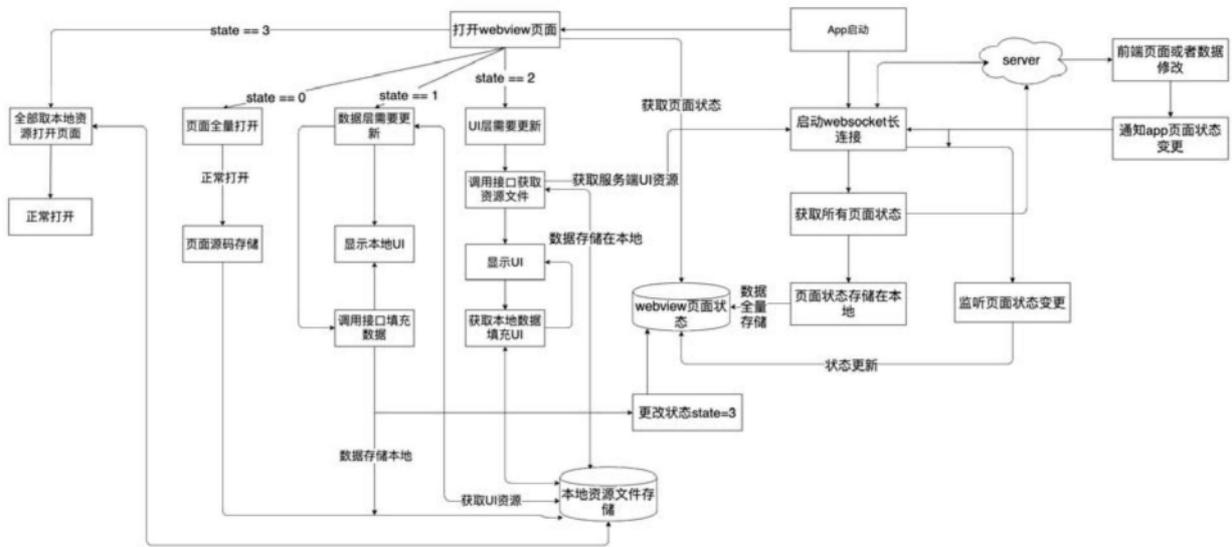


图2

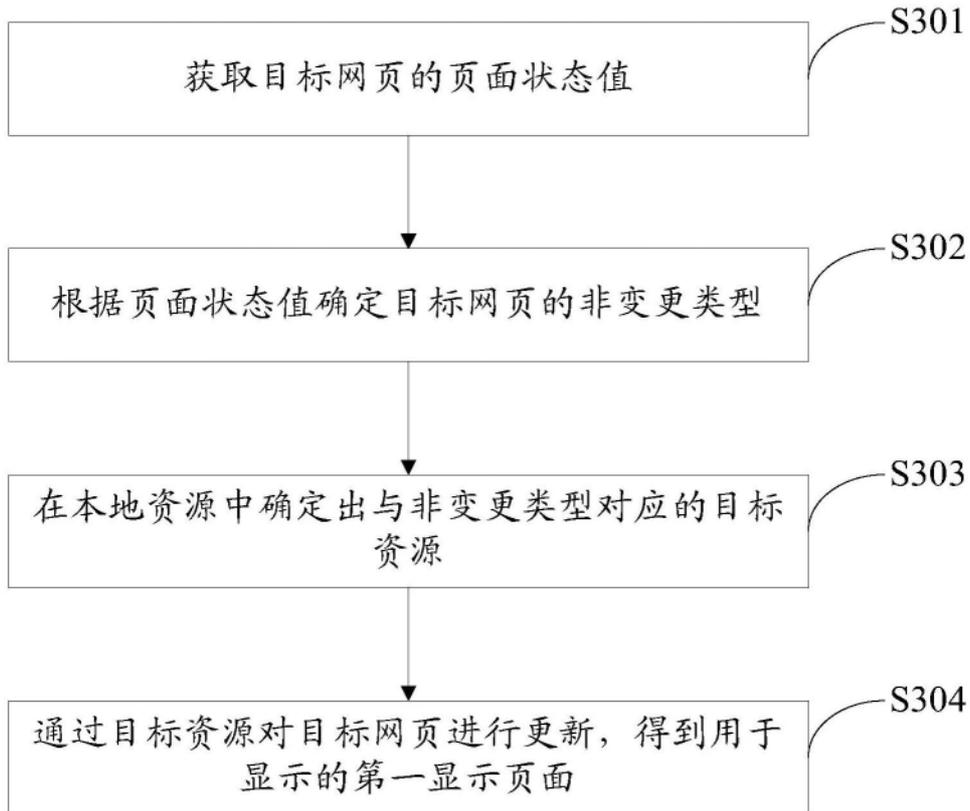


图3

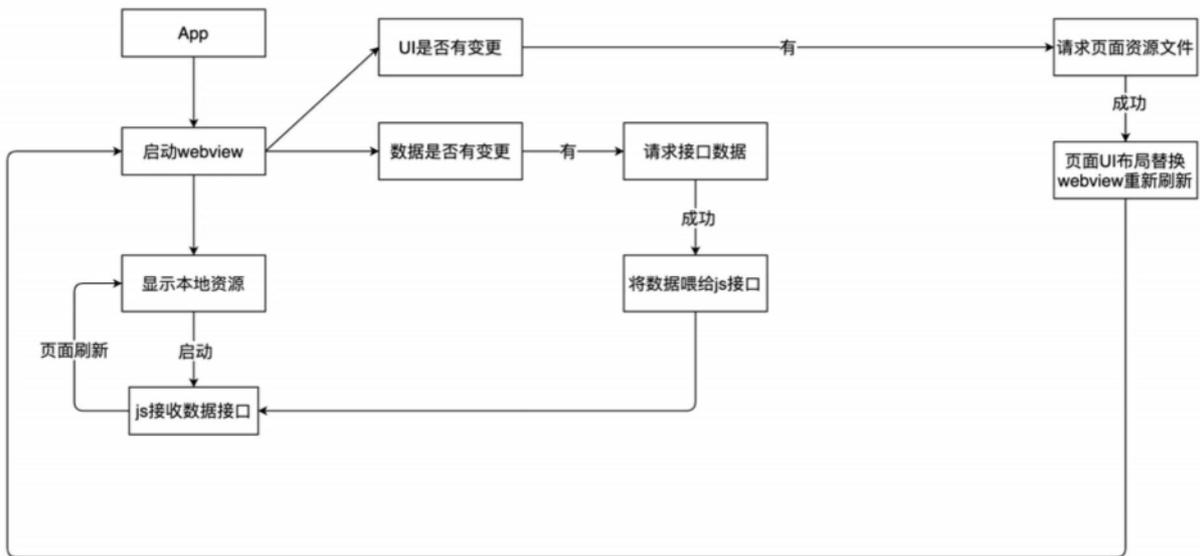


图4

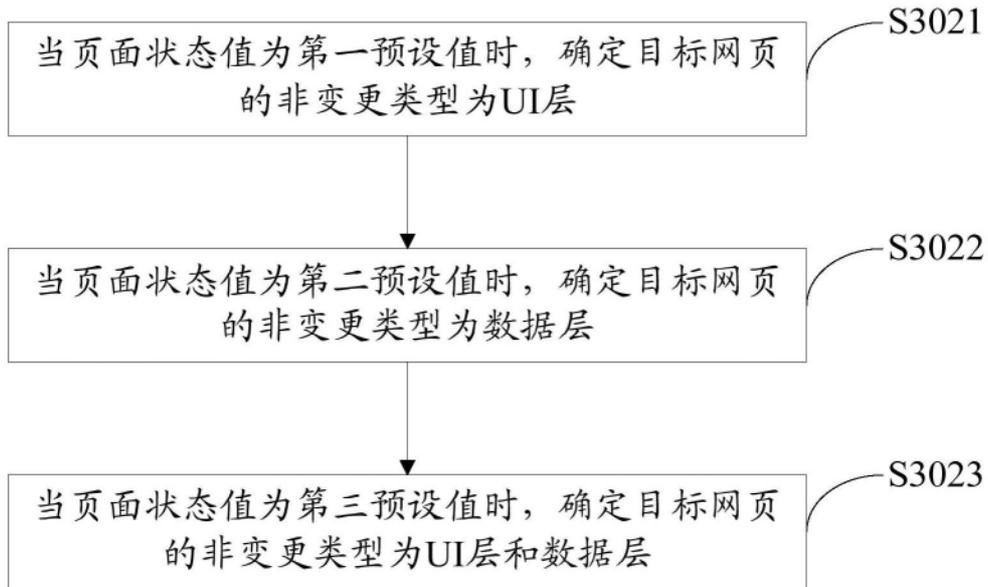


图5

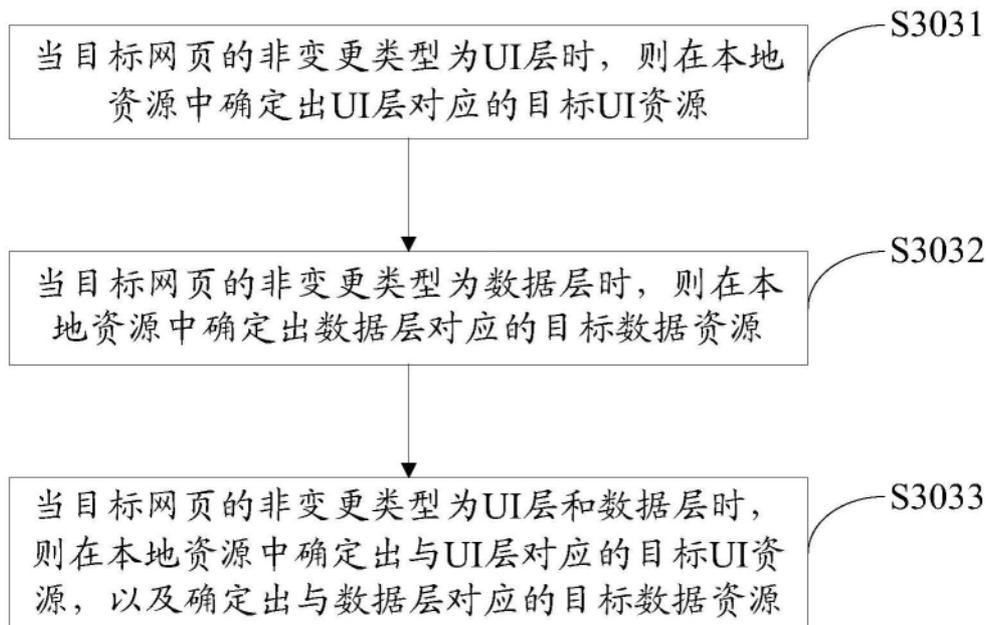


图6

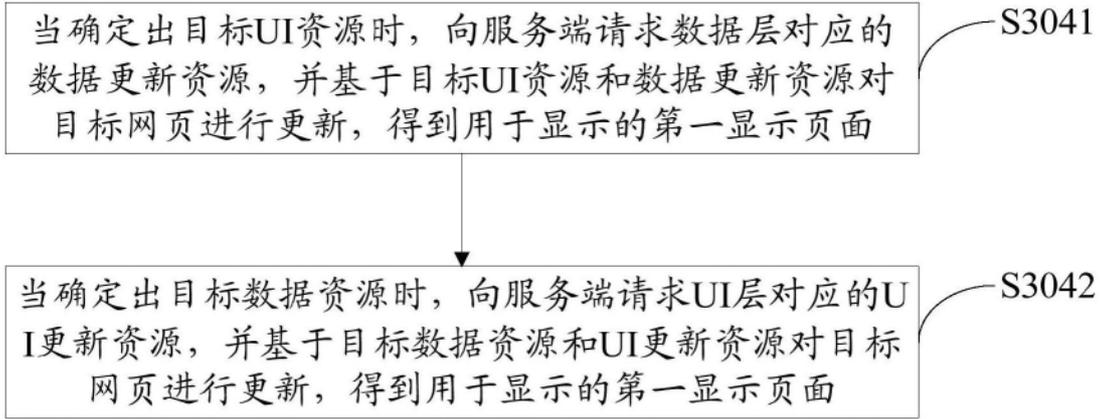


图7

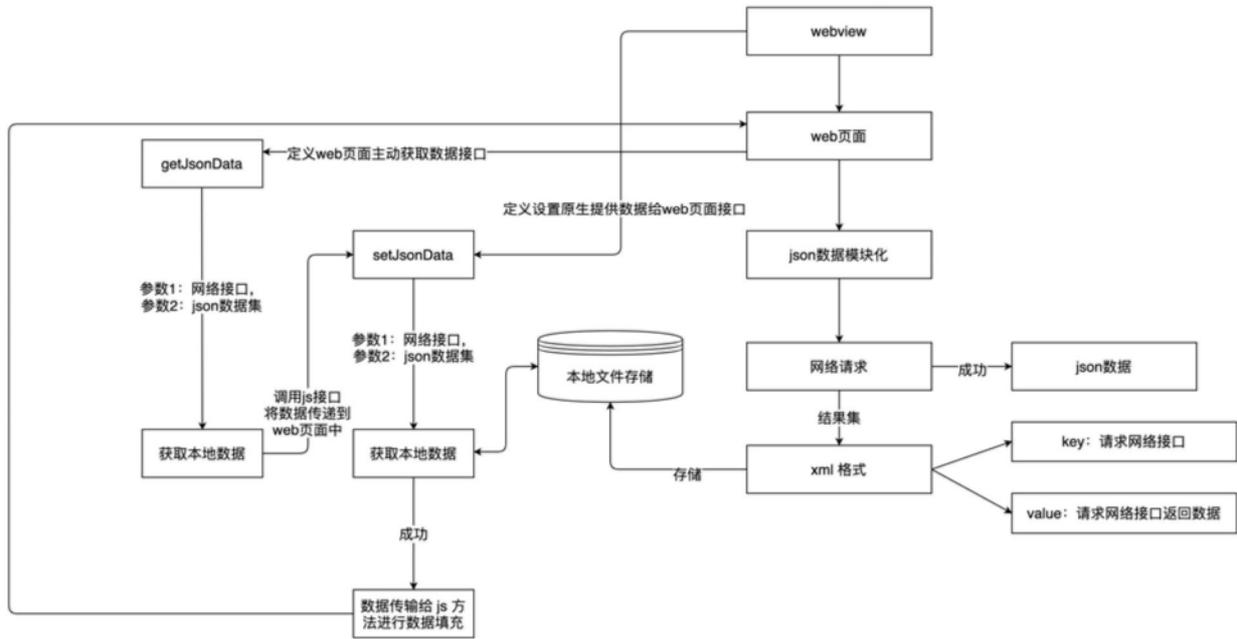


图8

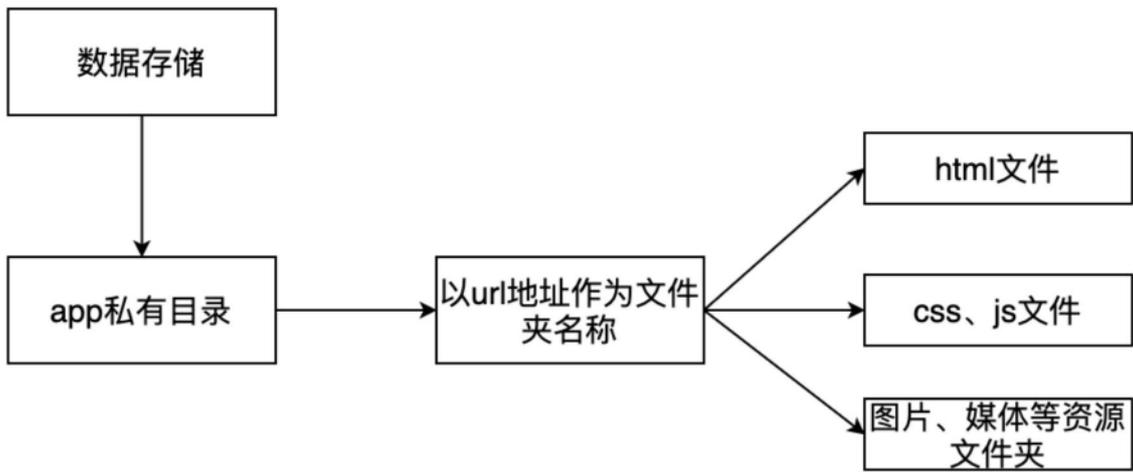


图9

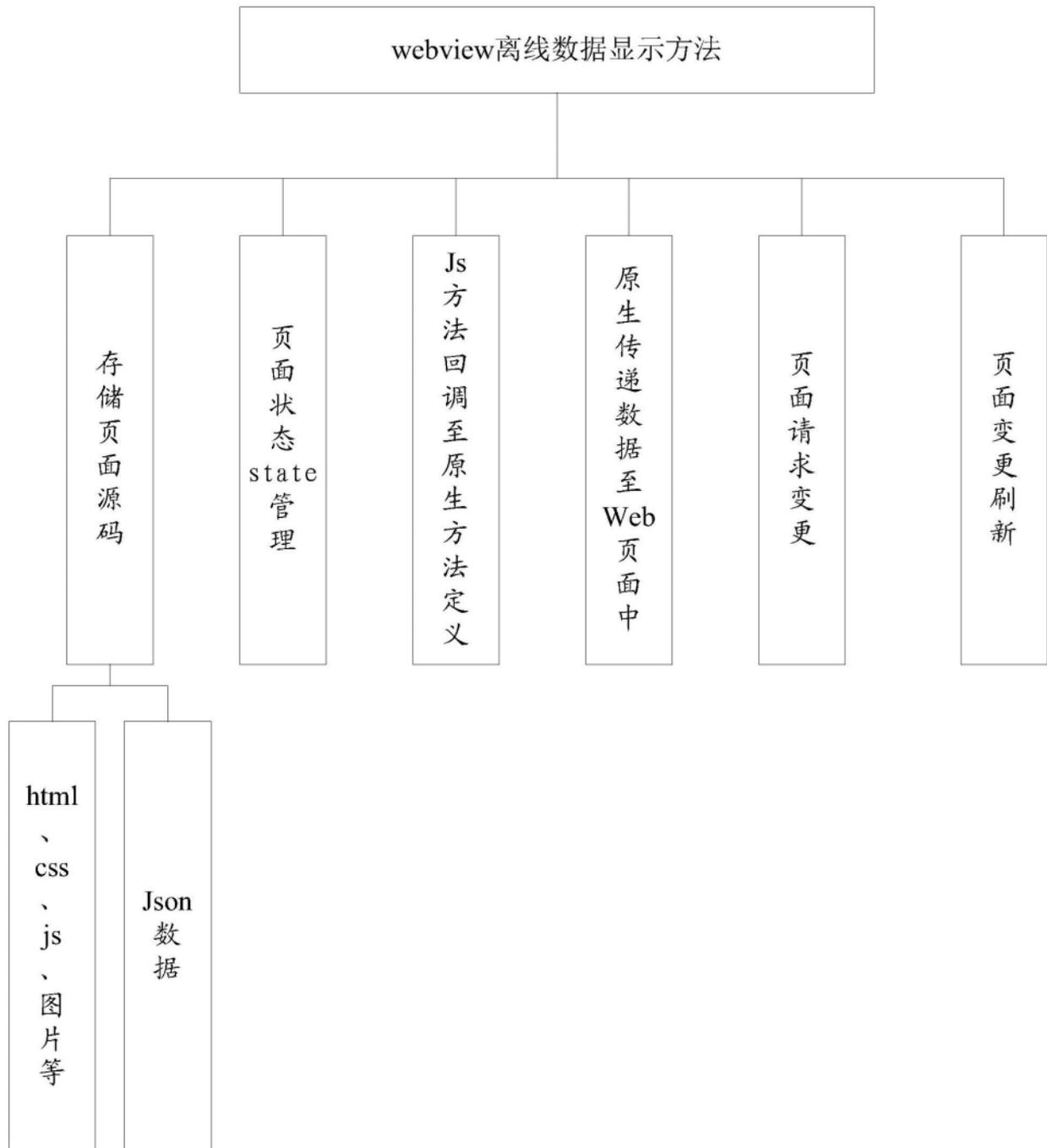


图10

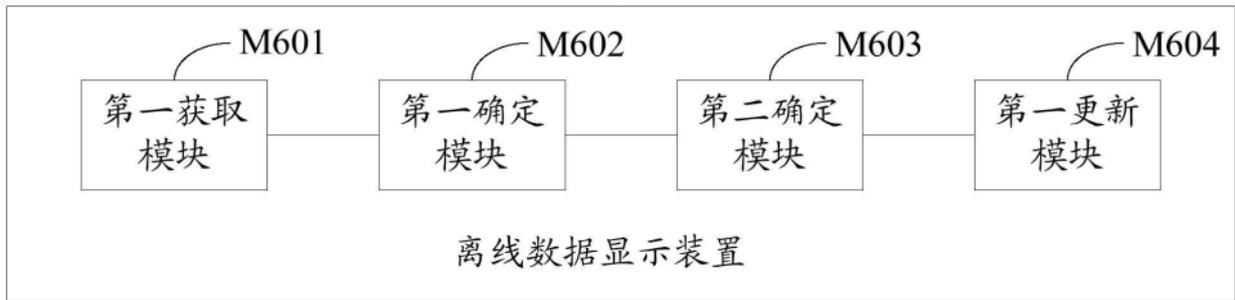


图11

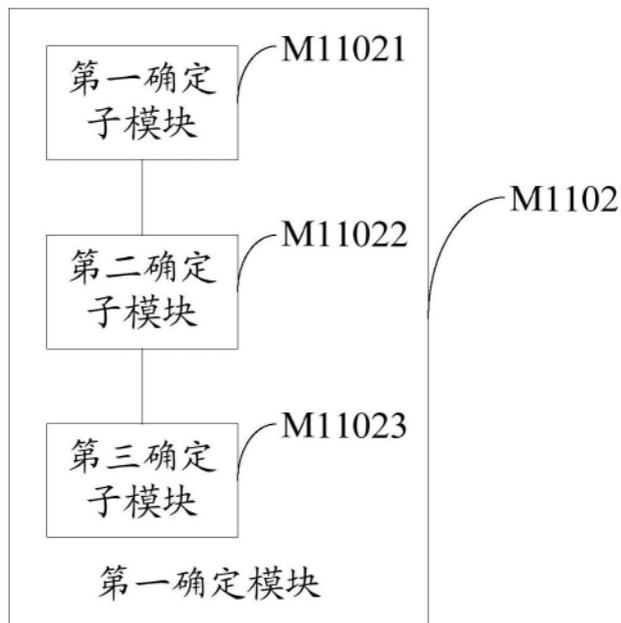


图12

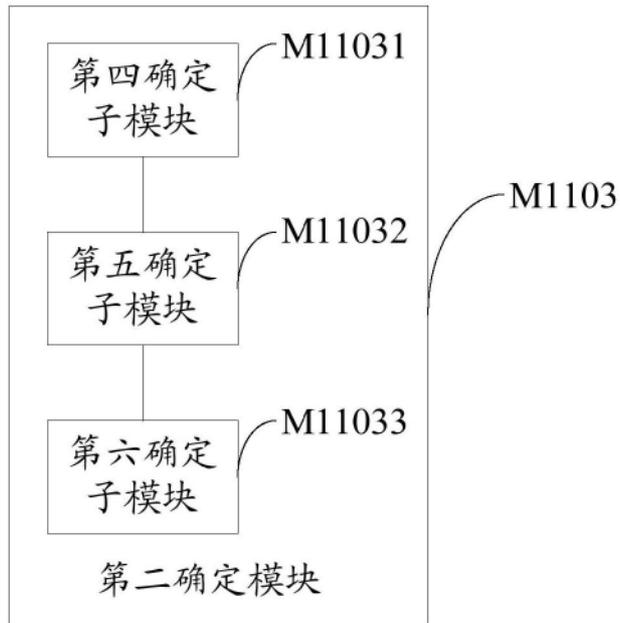


图13

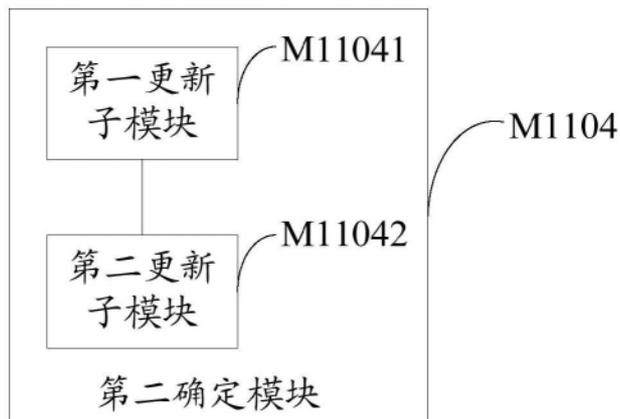


图14

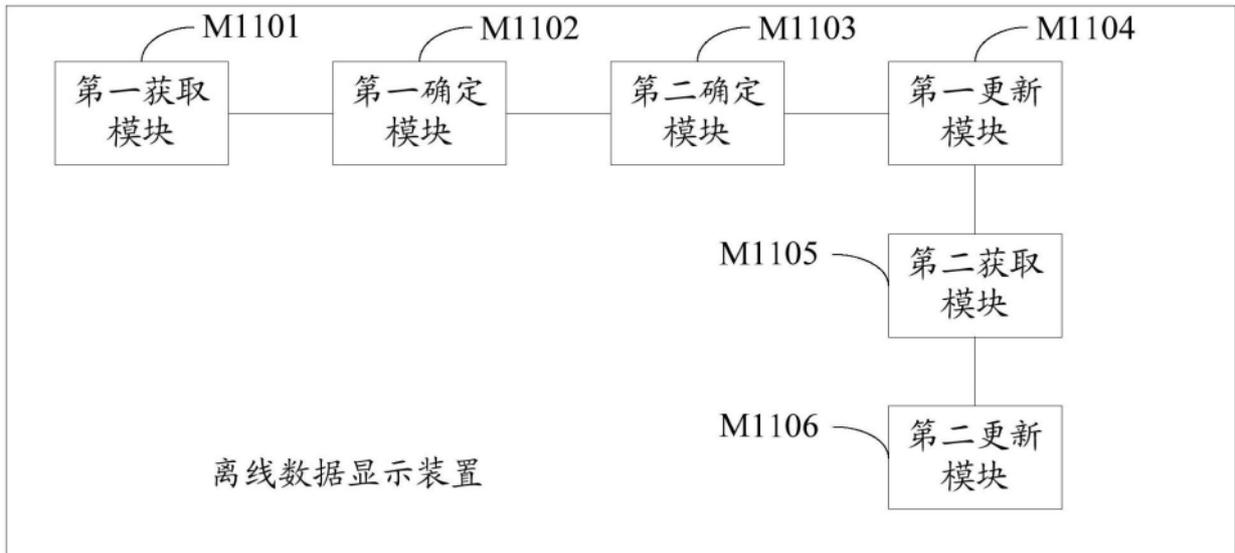


图15

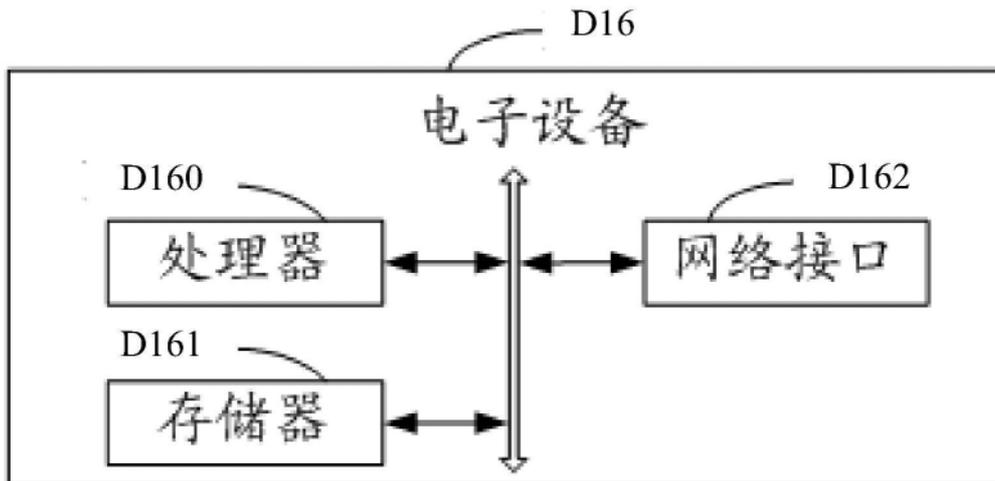


图16