



(12)发明专利申请

(10)申请公布号 CN 106201610 A

(43)申请公布日 2016. 12. 07

(21)申请号 201610529333.2

(22)申请日 2016.07.06

(71)申请人 青岛海信移动通信技术股份有限公司

地址 266071 山东省青岛市市南区江西路11号

(72)发明人 张百珂

(74)专利代理机构 北京三高永信知识产权代理有限公司 11138

代理人 江崇玉

(51)Int.Cl.

G06F 9/445(2006.01)

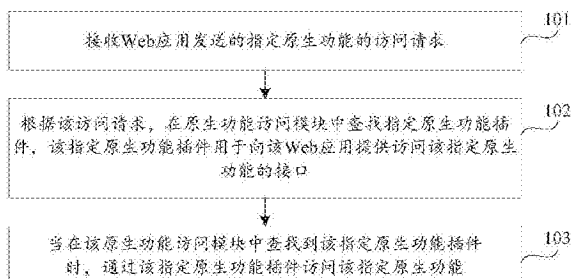
权利要求书2页 说明书14页 附图2页

(54)发明名称

Web应用访问终端原生功能的方法及装置

(57)摘要

本发明公开了一种Web应用访问终端原生功能的方法及装置,属于终端技术领域。所述方法包括:接收Web应用发送的指定原生功能的访问请求;根据该访问请求,在原生功能访问模块中查找指定原生功能插件,该原生功能访问模块用于向该Web应用提供访问终端原生功能的接口,该指定原生功能插件用于向该Web应用提供访问该指定原生功能的接口;当在该原生功能访问模块中查找到该指定原生功能插件时,通过该指定原生功能插件访问该指定原生功能。本发明中,Web应用可以通过原生功能访问模块任意访问终端所有的原生功能,且访问过程与Web应用所在的浏览器无关,不受浏览器功能的限制,从而提高了Web应用的使用性能。



1. 一种Web应用访问终端原生功能的方法,其特征在于,所述方法包括:

接收Web应用发送的指定原生功能的访问请求;

根据所述访问请求,在原生功能访问模块中查找指定原生功能插件,所述原生功能访问模块用于向所述Web应用提供访问终端原生功能的接口,所述指定原生功能插件用于向所述Web应用提供访问所述指定原生功能的接口;

当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

2. 根据权利要求1所述的方法,其特征在于,所述接收所述Web应用发送的指定原生功能的访问请求之前,还包括:

将所述原生功能访问模块的配置文件中存储的多个原生功能插件加载到所述原生功能访问模块的插件管理器中;

相应地,所述当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能,包括:

判断所述插件管理器中加载的多个原生功能插件中是否包括所述指定原生功能插件;

当所述插件管理器中加载的多个原生功能插件中包括所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

3. 根据权利要求1或2所述的方法,其特征在于,所述接收所述Web应用发送的指定原生功能的访问请求之前,还包括:

将所述原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到所述原生功能访问模块的插件管理器中;

相应地,所述当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能,包括:

判断所述插件管理器中加载的多个原生功能插件的标识中是否包括所述指定原生功能插件的标识;

当所述插件管理器中加载的多个原生功能插件的标识中包括所述指定原生功能插件的标识时,判断所述指定原生功能插件是否存储在所述插件管理器中;

当所述指定原生功能插件已存储在所述插件管理器中时,通过所述指定原生功能插件访问所述指定原生功能。

4. 根据权利要求3所述的方法,其特征在于,所述判断所述指定原生功能插件是否存储在所述插件管理器中之后,还包括:

当所述指定原生功能插件未存储在所述插件管理器中时,根据所述指定原生功能插件的标识,将所述指定原生功能插件从服务器中下载至所述插件管理器中;

通过所述指定原生功能插件访问所述指定原生功能。

5. 根据权利要求1所述的方法,其特征在于,所述原生功能访问模块为安装在所述终端中与所述Web应用对应的模块。

6. 一种Web应用访问终端原生功能的装置,其特征在于,所述装置包括:

接收模块,用于接收Web应用发送的指定原生功能的访问请求;

查找模块,用于根据所述访问请求,在原生功能访问模块中查找指定原生功能插件,所述原生功能访问模块用于向所述Web应用提供访问终端原生功能的接口,所述指定原生功

能插件用于向所述Web应用提供访问所述指定原生功能的接口；

访问模块,用于当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

7. 根据权利要求6所述的装置,其特征在于,所述装置还包括:

第一加载模块,用于将所述原生功能访问模块的配置文件中存储的多个原生功能插件加载到所述原生功能访问模块的插件管理器中;

相应地,所述访问模块包括:

判断单元,用于判断所述插件管理器中加载的多个原生功能插件中是否包括所述指定原生功能插件;

第一访问单元,用于当所述插件管理器中加载的多个原生功能插件中包括所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

8. 根据权利要求6或7所述的装置,其特征在于,所述装置还包括:

第二加载模块,用于将所述原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到所述原生功能访问模块的插件管理器中;

相应地,所述访问模块包括:

第二判断单元,用于判断所述插件管理器中加载的多个原生功能插件的标识中是否包括所述指定原生功能插件的标识;

第三判断单元,用于当所述插件管理器中加载的多个原生功能插件的标识中包括所述指定原生功能插件的标识时,判断所述指定原生功能插件是否存储在所述插件管理器中;

第二访问单元,当所述指定原生功能插件已存储在所述插件管理器中时,通过所述指定原生功能插件访问所述指定原生功能。

9. 根据权利要求8所述的装置,其特征在于,所述访问模块还包括:

下载单元,用于当所述指定原生功能插件未存储在所述插件管理器中时,根据所述指定原生功能插件的标识,将所述指定原生功能插件从服务器中下载至所述插件管理器中;

第三访问单元,用于通过所述指定原生功能插件访问所述指定原生功能。

10. 根据权利要求6所述的装置,其特征在于,所述原生功能访问模块为安装在所述终端中与所述Web应用对应的模块。

Web应用访问终端原生功能的方法及装置

技术领域

[0001] 本发明实施例涉及终端技术领域,特别涉及一种Web(网页)应用访问终端原生功能的方法及装置。

背景技术

[0002] Web应用是指采用B/S(Browser/Server,浏览器/服务器)架构,在浏览器上运行且在服务器上进行数据处理的应用,由于Web应用无需下载和更新,且在各个操作系统中均可运行,也即是可以跨平台运行,因而得到了广泛地应用。其中,当用户在终端中使用该Web应用时,可能需要通过该Web应用访问该终端的原生功能,该终端的原生功能是指终端中内置的功能,且只有原生应用才可以访问所有的原生功能,比如终端的原生功能可以为扬声器、摄像头、GPS(Global Positioning System,全球定位系统)、加速传感器等。

[0003] 目前,浏览器中可以提供一些应用接口API(Application Programming Interface,应用程序编程接口),如提供访问摄像头接口-getUserMedia等。在Web应用运行的过程中,当该Web应用需要访问终端的指定原生功能时,该终端首先需要查找该浏览器是否含有访问该指定原生功能的接口,若含有,则进一步判断所使用的浏览器是否支持该接口,一旦判断出该浏览器支持该接口,该Web应用便可以调用该接口,以通过该接口直接访问该指定设备原生功能;若所使用的浏览器没有该接口或者是不支持该接口,则该Web应用将无法访问该指定原生功能。

[0004] 在实现本发明的过程中,发明人发现现有技术至少存在以下问题:

[0005] 由于浏览器能够提供的访问原生功能的接口不全,也即是所支持的原生功能不全,因此Web应用只能通过浏览器访问终端的部分原生功能,而无法访问设备的所有原生功能。而且,不同浏览器支持原生功能的程度不同,比如在不同浏览器中使用同一Web应用访问同一原生功能时,有的浏览器可以使用该原生功能,有的则不可以,从而限制了Web应用的使用。

发明内容

[0006] 为了解决现有技术的问题,本发明实施例提供了一种Web应用访问终端原生功能的方法及装置。所述技术方案如下:

[0007] 第一方面,提供了一种Web应用访问终端原生功能的方法,所述方法包括:

[0008] 接收Web应用发送的指定原生功能的访问请求;

[0009] 根据所述访问请求,在原生功能访问模块中查找指定原生功能插件,所述原生功能访问模块用于向所述Web应用提供访问终端原生功能的接口,所述指定原生功能插件用于向所述Web应用提供访问所述指定原生功能的接口;

[0010] 当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

[0011] 可选地,所述接收所述Web应用发送的指定原生功能的访问请求之前,还包括:

- [0012] 将所述原生功能访问模块的配置文件中存储的多个原生功能插件加载到所述原生功能访问模块的插件管理器中；
- [0013] 相应地,所述当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能,包括:
- [0014] 判断所述插件管理器中加载的多个原生功能插件中是否包括所述指定原生功能插件;
- [0015] 当所述插件管理器中加载的多个原生功能插件中包括所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。
- [0016] 可选地,所述接收所述Web应用发送的指定原生功能的访问请求之前,还包括:
- [0017] 将所述原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到所述原生功能访问模块的插件管理器中;
- [0018] 相应地,所述当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能,包括:
- [0019] 判断所述插件管理器中加载的多个原生功能插件的标识中是否包括所述指定原生功能插件的标识;
- [0020] 当所述插件管理器中加载的多个原生功能插件的标识中包括所述指定原生功能插件的标识时,判断所述指定原生功能插件是否存储在所述插件管理器中;
- [0021] 当所述指定原生功能插件已存储在所述插件管理器中时,通过所述指定原生功能插件访问所述指定原生功能。
- [0022] 可选地,所述判断所述指定原生功能插件是否存储在所述插件管理器中之后,还包括:
- [0023] 当所述指定原生功能插件未存储在所述插件管理器中时,根据所述指定原生功能插件的标识,将所述指定原生功能插件从服务器中下载至所述插件管理器中;
- [0024] 通过所述指定原生功能插件访问所述指定原生功能。
- [0025] 可选地,所述原生功能访问模块为安装在所述终端中与所述Web应用对应的模块。
- [0026] 第二方面,提供了一种Web应用访问终端原生功能的装置,所述装置包括:
- [0027] 接收模块,用于接收Web应用发送的指定原生功能的访问请求;
- [0028] 查找模块,用于根据所述访问请求,在原生功能访问模块中查找指定原生功能插件,所述原生功能访问模块用于向所述Web应用提供访问终端原生功能的接口,所述指定原生功能插件用于向所述Web应用提供访问所述指定原生功能的接口;
- [0029] 访问模块,用于当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。
- [0030] 可选地,所述装置还包括:
- [0031] 第一加载模块,用于将所述原生功能访问模块的配置文件中存储的多个原生功能插件加载到所述原生功能访问模块的插件管理器中;
- [0032] 相应地,所述访问模块包括:
- [0033] 判断单元,用于判断所述插件管理器中加载的多个原生功能插件中是否包括所述指定原生功能插件;
- [0034] 第一访问单元,用于当所述插件管理器中加载的多个原生功能插件中包括所述指

定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

[0035] 可选地,所述装置还包括:

[0036] 第二加载模块,用于将所述原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到所述原生功能访问模块的插件管理器中;

[0037] 相应地,所述访问模块包括:

[0038] 第二判断单元,用于判断所述插件管理器中加载的多个原生功能插件的标识中是否包括所述指定原生功能插件的标识;

[0039] 第三判断单元,用于当所述插件管理器中加载的多个原生功能插件的标识中包括所述指定原生功能插件的标识时,判断所述指定原生功能插件是否存储在所述插件管理器中;

[0040] 第二访问单元,当所述指定原生功能插件已存储在所述插件管理器中时,通过所述指定原生功能插件访问所述指定原生功能。

[0041] 可选地,所述访问模块还包括:

[0042] 下载单元,用于当所述指定原生功能插件未存储在所述插件管理器中时,根据所述指定原生功能插件的标识,将所述指定原生功能插件从服务器中下载至所述插件管理器中;

[0043] 第三访问单元,用于通过所述指定原生功能插件访问所述指定原生功能。

[0044] 可选地,所述原生功能访问模块为安装在所述终端中与所述Web应用对应的模块。

[0045] 本发明实施例提供的技术方案带来的有益效果是:

[0046] 本发明实施例中,终端可以接收Web应用发送的指定原生功能的访问请求,并根据该访问请求,在原生功能访问模块中查找指定原生功能插件,当在该原生功能访问模块中查找到该指定原生功能插件时,即可通过该指定原生功能插件访问该指定原生功能,其中,该原生功能访问模块用于向该Web应用提供访问终端原生功能的接口,因此通过本发明实施例提供的方法,该Web应用可以通过该原生功能访问模块任意访问终端所有的原生功能,且访问过程与Web应用所在的浏览器无关,不受浏览器功能的限制,从而提高了Web应用的使用性能。

附图说明

[0047] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,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。

[0048] 图1是本发明提供的一种Web应用访问终端原生功能的方法流程图;

[0049] 图2是本发明提供的一种Web应用访问终端原生功能的装置框图;

[0050] 图3是本发明实施例提供的一种终端300的结构示意图。

具体实施方式

[0051] 为使本发明的目的、技术方案和优点更加清楚,下面将结合附图对本发明实施方式作进一步地详细描述。

[0052] 在对本发明实施例进行详细的解释说明之前,先对本发明实施例的应用场景予以介绍。本发明实施例提供的方法应用于终端中,该终端可以为计算机、智能手机、平板电脑、笔记本电脑、超级移动个人计算机(英文:Ultra-mobile Personal Computer,简称:UMPC)、上网本、个人数字助理(英文:Personal Digital Assistant,简称:PDA)等,本发明实施例对此不做限定。进一步地,该终端可以安装有浏览器,浏览器中可以运行诸如社交应用、网购应用等的Web应用。另外,在Web应用运行的过程中,根据用户的操作,Web应用还可以访问终端的原生功能。

[0053] 图1是本发明实施例提供的一种Web应用访问终端原生功能的方法流程图,参见图1,该方法可以包括如下几个步骤:

[0054] 步骤101:接收Web应用发送的指定原生功能的访问请求。

[0055] 本发明实施例中,终端可以在Web应用运行时,接收Web应用发送的指定原生功能的访问请求。

[0056] 其中,该访问请求可以基于该Web应用的显示界面触发,可以由用户触发,用户可以通过指定操作触发,该指定操作可以为点击操作、滑动操作、语音操作等,本发明实施例对此不做限定。

[0057] 其中,该指定原生功能的访问请求可以携带该指定原生功能的标识或者用于指示访问该指定原生功能的代码等,以表示该访问请求用于访问该指定原生功能,本发明实施例对此不做限定。其中,该指定原生功能可以为终端内置的任一原生功能,如扬声器、麦克风、摄像头、GPS、加速传感器、温度传感器、指南针、消息提醒、重力感应等,本发明实施例对也不做限定。

[0058] 例如,用户可以通过点击Web应用页面中的拍照图标触发摄像头的访问请求,该访问请求可以携带摄像头标识或者调用摄像头的代码等。

[0059] 另外,在步骤101之前,还需在浏览器中启动该Web应用。具体地,该终端可以在基于浏览器接收到Web应用的启动指令时,启动该Web应用。而在启动Web应用时,可以通过调用start_url()接口启动该Web应用,当然也可以通过其他方法启动该Web应用,本发明实施例对此不做限定。

[0060] 其中,该启动指令可以由用户触发,用户可以通过指定操作触发。例如,用户可以通过点击浏览器中显示的Web应用图标触发该Web应用的启动指令等。

[0061] 需要说明的一点是,本发明实施例所示的Web应用为基于开发的Web应用运行平台生成的且可以在浏览器中运行的Web应用。本发明实施例中,可以将开发者用html5编写的纯Web应用打包成可以在不同操作平台上运行的Web应用,如在Android、IOS、Blackberry、Windows Phone、和Firefox OS等操作平台上运行的应用等,并可以将打包后的Web应用发布到所支持的操作平台中。

[0062] 需要说明的另一点是,本发明实施例中,为了便于Web应用自由访问终端的原生功能,可以在终端中安装原生功能访问模块,该原生功能访问模块用于向Web应用提供访问终端原生功能的接口,从而使得终端可以在接收到Web应用发送的原生功能的访问请求时,通过该原生功能访问模块调用访问原生功能的接口,从而实现对原生功能的访问。

[0063] 其中,该原生功能访问模块可以为cordova,当然随着技术的发展,该原生功能访问模块也可以为其他可以实现上述功能的模块,本发明实施例对此不做现代。

[0064] 其中,该原生功能访问模块的安装方式可以包括以下方式中的任一种:

[0065] 在第一种方式中,该原生功能访问模块可以为安装在终端中与各个Web应用分别对应的模块,也即是,该终端可以安装多个原生功能访问模块,每个Web应用对应于一个原生功能访问模块,每个Web应用仅通过对应的原生功能访问模块访问终端的原生功能。

[0066] 进一步地,可以将Web应用与对应的原生功能访问模块一同打包,例如可以将Web应用与对应的原生功能访问模块打包成一个应用程序,以便该Web应用运行时,可以通过对应的原生功能访问模块提供的接口访问终端的原生功能。

[0067] 在第二种方式中,该原生功能访问模块也可以为安装在终端中与所有Web应用均对应的模块,也即是,该终端可以仅安装一个原生功能访问模块,该原生功能访问模块与所有Web应用对应,每个Web应用都是通过这一个原生功能访问模块访问终端的原生功能。

[0068] 本发明实施例中,在利用该原生功能访问模块访问终端的原生功能之前,还需启动该原生功能访问模块。其中,该原生功能访问模块的启动方式可以包括以下方式中的任一种:

[0069] 在第一种方式中,终端可以在启动Web应用之后,立即启动原生功能访问模块,以便Web应用可以通过原生功能访问模块随时调用终端的原生功能。

[0070] 在第二种方式中,为了节约资源,终端还可以在启动Web应用之后不立即启动原生功能访问模块,只有在接收到Web应用发送的原生功能的访问请求时,才启动原生功能访问模块。

[0071] 其中,启动原生功能访问模块时,可以采用LoadUrl()方法进行启动,也可以采用其他方法进行启动,本发明实施例对此不做限定。其中,该终端可以在该原生功能访问模块所生成的工程应用Activity(活动类)中启动LoadUrl()方法,也可以在通过其他方式启动LoadUrl()方法,本发明实施例对此不做限定。

[0072] 另外,在原生功能访问模块启动之后,为了使得该原生功能访问模块可以向Web应用提供访问终端原生功能的所有接口,还可以对该原生功能访问模块进行初始化。其中,对原生功能访问模块进行初始化的过程可以包括以下方式中的任一种:

[0073] 在第一种方式中,将该原生功能访问模块的配置文件中存储的多个原生功能插件加载到该原生功能访问模块的插件管理器中。

[0074] 其中,该多个原生功能插件中的每个原生功能插件对应于终端中的一个原生功能,用于向Web应用提供访问对应原生功能的接口。该插件管理器用于对加载的原生功能插件进行管理,以便后续过程中,根据接收到的指定原生功能的访问请求从该插件管理器中调用指定原生功能插件。

[0075] 在该种方式中,配置文件中可以存储对应于终端所有原生功能的具体原生功能插件,在原生功能访问模块启动后,即可将该配置文件中存储的所有具体原生功能插件加载到插件管理器中。

[0076] 在第二种方式中,将该原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到该原生功能访问模块的插件管理器中。

[0077] 其中,原生功能插件的标识可以为原生功能插件的名称、ID(identification,身份标识号),也可以为该原生功能插件的类名称,如class名称等,本发明实施例对此不做限定。

[0078] 在该种方式中,配置文件中存储有用于访问终端所有原生功能的原生功能插件的标识,在对原生功能访问模块进行初始化的过程中,为了节约初始化时间,提高加载效率,可以无需立即加载具体的原生功能插件,而仅是将配置文件中存储的多个原生功能插件的标识加载到插件管理器中,从而将原生功能插件的标识作为检索标识,以便根据该原生功能插件的标识调用对应的原生功能插件。

[0079] 加载完成之后,该插件管理器可以将该多个原生功能插件的标识保存在插件列表中,其中,该插件列表可以为hashmap(哈希表)、数据库列表等,本发明实施例对此不做限定。

[0080] 另外,将该多个原生功能插件的标识保存在插件列表中后,该原生功能访问模块还可以用service名称作为对应原生功能插件的key值,该service名称可以为对应原生功能插件的序号等,后续过程中,该原生功能访问模块进行根据对应原始功能插件的key值,即可从该插件管理器中查找对应的原生功能插件,本发明实施例对此不做限定。

[0081] 例如,以该原生功能访问模块为cordava为例,终端可以在Web应用启动时,在所生成的工程应用Activity中启动LoadUrl()方法,从而通过LoadUrl()方法启动cordava,并开始对cordava进行一系列的初始化。其中,在对cordava进行初始化时,插件管理器pluginManager可以将配置文件中加载的所有插件plugin的class名称保存到一个hashmap中,并用service名称作为对应插件plugin的key值,该service名称可以为对应插件plugin的序号等,本发明实施例对此不做限定。完成cordava初始化之后,该终端即可随时接收Web应用发送的指定原生功能的访问请求,并通过cordava响应该访问请求。

[0082] 需要说明的是,本发明实施例仅是该原生功能访问模块包括配置文件和插件管理器为例进行说明,而实际应用中,当该原生功能访问模块不包括配置文件而只包括插件管理器时,该插件管理器中也可以直接存储有原生功能插件或原生功能插件的标识,从而无需从配置文件中加载,也即无需对该原生功能访问模块进行初始化,本发明实施例对此不做限定。

[0083] 步骤102:根据该访问请求,在原生功能访问模块中查找指定原生功能插件,该指定原生功能插件用于向该Web应用提供访问该指定原生功能的接口。

[0084] 其中,根据原生功能访问模块初始化方式的不同,在原生功能访问模块中查找指定原生功能插件的过程也不同。具体地,在原生功能访问模块中查找指定原生功能插件的过程可以包括以下方式中的任一种:

[0085] 在第一种方式中,当基于步骤101中的第一种方式进行初始化时,也即是,当将该原生功能访问模块的配置文件中存储的多个原生功能插件加载到该原生功能访问模块的插件管理器中时,该终端即可根据该访问请求,在该插件管理器中查找该指定原生功能插件。

[0086] 在第二种方式中,当基于步骤102中的第二种方式进行初始化时,也即是,当将该原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到该原生功能访问模块的插件管理器中时,该终端可以先根据该访问请求,在该插件管理器中查找指定原生功能插件的标识,当查找到该指定原生功能插件的标识时,再根据该指定原生功能插件的标识查找对应的指定原生功能插件。

[0087] 步骤103:当在该原生功能访问模块中查找到该指定原生功能插件时,通过该指定

原生功能插件访问该指定原生功能。

[0088] 基于步骤102的第一种方式,当在该原生功能访问模块中查找到该指定原生功能插件时,通过该指定原生功能插件访问该指定原生功能的过程可以包括:

[0089] 判断该插件管理器中加载的多个原生功能插件中是否包括该指定原生功能插件;当该插件管理器中加载的多个原生功能插件中包括该指定原生功能插件时,通过该指定原生功能插件访问该指定原生功能。

[0090] 该种方式中,由于初始化过程中在插件管理器中加载的是具体的原生功能插件,因此,当接收到指定原生功能的访问请求时,即可在该插件管理器中直接查找该指定原生功能插件,当该插件管理器中加载的多个原生功能插件中包括该指定原生功能插件时,也就表示能够查找到该指定原生功能插件时,此时即可通过该指定原生功能插件访问该指定原生功能。

[0091] 基于步骤102的第二种方式,当在该原生功能访问模块中查找到该指定原生功能插件时,通过该指定原生功能插件访问该指定原生功能的过程可以包括:

[0092] 判断该插件管理器中加载的多个原生功能插件的标识中是否包括该指定原生功能插件的标识;当该插件管理器中加载的多个原生功能插件的标识中包括该指定原生功能插件的标识时,判断该指定原生功能插件是否存储在该插件管理器中;当该指定原生功能插件已存储在该插件管理器中时,通过该指定原生功能插件访问该指定原生功能。

[0093] 该种方式中,由于初始化过程中在插件管理器中加载的并不是具体的原生功能插件,而是原生功能插件的标识,因此,当接收到指定原生功能的访问请求时,需要先在插件管理器中查找该指定原生功能插件的标识,当查找到该指定原生功能插件的标识时,还需进一步确定与该指定原生功能插件的标识对应的指定原生功能插件是否已加载,即是否已实例化,当该指定原生功能插件存储在该插件管理器中,也就表示该指定原生功能插件已加载,也即是已实例化,从而即可直接通过该指定原生功能插件访问该指定原生功能。

[0094] 另外,在上述第二种方式中,当在插件管理器中查找到该指定原生功能插件的标识,但该指定原生功能插件未存储在该插件管理器中时,还可以先根据该指定原生功能插件的标识,将该指定原生功能插件从服务器中下载至该插件管理器中,再通过该指定原生功能插件访问该指定原生功能。

[0095] 其中,该服务器可以是为终端提供原生功能访问服务的服务器,该服务器中可以存储有多个原生功能插件,并可以对原生功能插件进行更新。

[0096] 本发明实施例中,当该指定原生功能插件未实例化时,需要先实例化该指定原生功能插件,实例化完成后,才能通过该指定原生功能插件访问该指定原生功能。

[0097] 具体地,该终端可以基于原生功能访问模块,利用java反射机制实例化该指定原生功能插件,当然也可以采用其他方法实例化该指定原生功能插件,本发明实施例对此不做限定。

[0098] 另外,若该配置文件中不仅存储有多个原生功能插件的标识,还存储有多个原生功能插件,则当在插件管理器中查找到该指定原生功能插件的标识,但该指定原生功能插件未存储在该插件管理器中时,还可以从配置文件中加载该指定原生功能插件,本发明实施例对此不做限定。

[0099] 该种方式中,当该Web应用第一次发送指定原生功能的访问请求时,原生功能访问

模块的插件管理器中仅存储有指定原生功能插件的标识,并未加载与该指定原生功能对应的指定原生功能插件,也即是,该指定原生功能插件还未实例化,因此此时,需要先根据该指定原生功能插件的标识从服务器中下载该指定原生功能插件,再通过下载的指定原生功能插件访问该指定原生功能;之后,当该Web应用再次发送该指定原生功能的访问请求时,由于该指定原生功能插件已被加载至该插件管理器中,因此无需从服务器中下载或从配置文件中加载,即可直接通过已存储的指定原生功能插件访问该指定原生功能。

[0100] 本发明实施例中,通过在对原生功能访问模块进行初始化时,仅将配置文件中的原生功能插件的标识加载至插件管理器中,后续过程中当需要访问指定原生功能时,再下载指定原生功能插件,节省了初始化的时间,提高了初始化的效率,提高了用户体验。

[0101] 现有技术中,Web应用访问终端的原生功能时,需要先判断所使用的浏览器是否支持访问原生功能的接口,若支持则可以访问该原生功能,否则将无法使用该原生功能,使得Web应用对原生功能的访问受浏览器的限制较大。而本实施例中,Web应用访问原生功能时,浏览器只是一个显示平台,Web应用可以通过原生功能访问模块提供的接口像原生应用一样任意访问终端的所有原生功能,而不会受到浏览器自身功能的影响,使得Web应用可以更加灵活、方便地访问终端的原生功能,提高了用户体验。

[0102] 进一步地,当该指定原生功能插件已存储在该插件管理器中时,通过该指定原生功能插件访问该指定原生功能的过程还可以包括:当该指定原生功能插件已存储在该插件管理器中时,请求该指定原生功能插件,并向该Web应用发送该访问请求的确认响应,由该Web应用根据该确认响应调用该原生功能。

[0103] 其中,该原生功能访问模块可以通过指定函数或代码请求该指定原生功能插件,当通过指定函数请求该指定原生功能插件,该指定函数可以为exec()函数等,本发明实施例对此不做限定。

[0104] 进一步地,该访问请求的确认响应还可以包括该访问请求的请求消息ID和确认信息,相应地,向该Web应用发送该访问请求的确认响应的过程还可以包括:根据该请求消息ID,查找该访问请求,并将该确认信息返回到该访问请求中。其中,该确认信息用于指示该Web应用可以访问该指定原生功能。

[0105] 具体地,该终端可以根据该请求消息ID,利用指定查找函数查找该访问请求,该指定查找函数可以为callbackID函数等,当然也可以其他查找函数,本发明实施例对此不做限定。

[0106] 其中,在将该确认信息返回到该访问请求的过程中,可以将该确认信息返回到该访问请求的响应函数中,该响应函数用于指示该访问请求的响应结果。则后续过程中,该Web应用即可调用该响应函数,并根据该响应函数中的访问结果,通过该指定原生功能插件提供的接口调用该终端的指定原生功能。其中,该响应函数可以为success函数等,本发明实施例对此不做限定。

[0107] 另外,当未查找到该指定原生功能插件,或者该指定原生功能插件无法调用时,还可以向该Web应用发送该访问请求的失败响应,用于指示该Web应用不能访问该指定原生功能,本发明实施例对此不做限定。

[0108] 进一步地,由于Web应用可能会向原生功能访问模块发送不同原生功能的访问请求,而该原生功能访问模块则会对该不同原生功能的访问请求分别进行响应,从而向该Web

应用返回多个响应结果,因此在一种可能的实现方式中,可以先将多个响应结果按响应的顺序存放在一个消息队列中,该消息队列用于管理返回给Web应用的响应结果。之后再按存放顺序从该消息队列中读取响应结果,并将读取的响应结果发送至该Web应用的对应访问请求中,以便该访问请求根据对应的响应结果访问对应的原生功能。

[0109] 例如,以该原生功能访问模块为cordava为例,在一种可能的实现方式中,cordova中可以包括核心库文件,该核心库文件用于存储cordova常用的核心运行函数,该核心库文件可以包括如下内容:

[0110] CordovaActivity:Cordova活动类入口,也即是通过Cordova调用原生功能插件的入口,在完成Cordova的初始化工作之后,可以通过CordovaActivity实现具体的原生功能访问。

[0111] PluginManager:插件管理器,用于对Cordova加载的原生功能插件进行存储和管理。

[0112] ExposedJsApi:网页调用接口,用于javascript调用Native,也即是用于Web应用的网页端调用终端本地的原生功能,具体用于根据指定原生功能访问请求在插件管理器PluginManager中查找并调用指定原生功能插件。

[0113] NativeToJsMessageQueue:消息队列,用于Native调用javascript,也即是用于管理终端返回的对Web应用发送的原生功能访问请求的响应。

[0114] 相应地,基于上述cordova的核心库文件,在cordova启动后即可对cordova进行初始化,具体地,在对cordova进行初始化的过程中,可以先对cordova的关联对象CordovaWebView进行初始化,CordovaWebView初始化完成之后,再对插件管理器进行初始化,从而完成cordova的整个初始化工作,以便为后续与网页内容进行交互做准备。

[0115] 其中,CordovaWebView为cordova的关联对象,包含了PluginManager,NativeToJsMessageQueue,ExposedJsApi等成员变量,而对CordovaWebView进行初始化也即是对上述变量中的内容进行初始化。

[0116] 具体地,CordovaWebView初始化的过程中,CordovaWebView对象可以创建一个属于当前CordovaWebView对象的插件管理器PluginManager对象、一个消息队列NativeToJsMessageQueue对象、以及一个网页接口JavascriptInterface对象中的网页调用接口ExposedJsApi对象,并可以将ExposedJsApi对象添加到CordovaWebView对象中,且JavascriptInterface名称可以为:cordovaNative。

[0117] 其中,在创建ExposedJsApi时需要CordovaWebView的PluginManager对象和NativeToJsMessageQueue对象,且所有的Web应用与终端自身的原生功能的代码交互均可通过ExposedJsApi对象的exec方法实现。例如,Web应用可以通过exec()函数向Cordova发送指定原生功能的访问请求,Cordova可以通过exec()函数请求指定原生功能插件,并向该指定原生功能的访问请求返回响应结果等。

[0118] CordovaWebView初始化完成之后,可以调用CordovaWebView.loadUrl()开始对插件管理器的初始化,也即是开始向PluginManager中加载plugin。在一种可能的实现方式中,PluginManager在加载plugin的时候可以立即实例化plugin对象,而只是将plugin的Class名字保存到一个hashmap中,用service名字作为key值。当Web应用通过JavascriptInterface接口的ExposedJsApi对象请求终端的原生功能时,也即是通过

Cordova向终端发送原生功能的访问请求时,Cordova中的PluginManager会从hashmap中查找对应的plugin,并判断该plugin是否已实例化,若还未实例化,则利用java反射机制实例化该plugin,并通过exec()函数请求该原生功能插件。

[0119] 另外,在接收到Web应用发送的原生功能的访问请求后,Cordova会根据插件管理器的查找结果确定访问请求的响应结果,该响应结果可以为确认响应或失败响应。由于Web应用发送的原生功能的访问请求可能有多个,相应的响应结果也有多个,因此Cordova还可以先将响应结果存储在NativeToJsMessageQueue中。后续过程中,Cordova可以从队列头读取响应结果,并将响应结果发送至Web应用,例如Cordova可以从NativeToJsMessageQueue队列头pop头数据,再利用callbackID函数反向查找对应的访问请求,并将数据返回给该访问请求的success函数中,以便该Web应用根据该success函数调用终端的原生功能插件,从而在Web应用的网页中实现通过该原生功能插件提供的接口调用该原生功能。

[0120] 通过上述方法,原生应用访问模块可以提供一组API使Web应用具备在JavaScript网页中访问终端原生功能的能力,因此,在使用原生应用访问模块提供的API访问终端的原始功能时,Web应用与终端无需进行代码的交互,如进行Java或C的交互等,即可在网页中访问终端的原生功能。而且,本发明实施例中,可以将Web应用打包成系统原生应用格式并发布到所支持的操作平台中,且原生应用访问模块提供的API在多个操作平台基本一致,因此,本发明实施例提供的方法可以适用于所有操作平台,适用范围广,灵活性高。

[0121] 综上所述,本发明实施例中,终端可以接收Web应用发送的指定原生功能的访问请求,并根据该访问请求,在原生功能访问模块中查找指定原生功能插件,当在该原生功能访问模块中查找到该指定原生功能插件时,即可通过该指定原生功能插件访问该指定原生功能,其中,该原生功能访问模块用于向该Web应用提供访问终端原生功能的接口,因此通过本发明实施例提供的方法,该Web应用可以通过该原生功能访问模块任意访问终端所有的原生功能,且访问过程与Web应用所在的浏览器无关,不受浏览器功能的限制,从而提高了Web应用的使用性能。

[0122] 图2是本发明实施例提供的一种Web应用访问终端原生功能的装置框图,参见图2,该装置可以包括:

[0123] 接收模块201,用于接收Web应用发送的指定原生功能的访问请求;

[0124] 查找模块202,用于根据该访问请求,在原生功能访问模块中查找指定原生功能插件,该原生功能访问模块用于向该Web应用提供访问终端原生功能的接口,该指定原生功能插件用于向该Web应用提供访问该指定原生功能的接口;

[0125] 访问模块203,用于当在该原生功能访问模块中查找到该指定原生功能插件时,通过该指定原生功能插件访问该指定原生功能。

[0126] 可选地,该装置还包括:

[0127] 第一加载模块,用于将该原生功能访问模块的配置文件中存储的多个原生功能插件加载到该原生功能访问模块的插件管理器中;

[0128] 相应地,该访问模块包括:

[0129] 判断单元,用于判断该插件管理器中加载的多个原生功能插件中是否包括该指定原生功能插件;

[0130] 第一访问单元,用于当该插件管理器中加载的多个原生功能插件中包括该指定原

生功能插件时,通过该指定原生功能插件访问该指定原生功能。

[0131] 可选地,该装置还包括:

[0132] 第二加载模块,用于将该原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到该原生功能访问模块的插件管理器中;

[0133] 相应地,该访问模块包括:

[0134] 第二判断单元,用于判断该插件管理器中加载的多个原生功能插件的标识中是否包括该指定原生功能插件的标识;

[0135] 第三判断单元,用于当该插件管理器中加载的多个原生功能插件的标识中包括该指定原生功能插件的标识时,判断该指定原生功能插件是否存储在该插件管理器中;

[0136] 第二访问单元,当该指定原生功能插件已存储在该插件管理器中时,通过该指定原生功能插件访问该指定原生功能。

[0137] 可选地,该访问模块还包括:

[0138] 下载单元,用于当该指定原生功能插件未存储在该插件管理器中时,根据该指定原生功能插件的标识,将该指定原生功能插件从服务器中下载至该插件管理器中;

[0139] 第三访问单元,用于通过该指定原生功能插件访问该指定原生功能。

[0140] 可选地,该原生功能访问模块为安装在该终端中与该Web应用对应的模块。

[0141] 本发明实施例中,终端可以接收Web应用发送的指定原生功能的访问请求,并根据该访问请求,在原生功能访问模块中查找指定原生功能插件,当在该原生功能访问模块中查找到该指定原生功能插件时,即可通过该指定原生功能插件访问该指定原生功能,其中,该原生功能访问模块用于向该Web应用提供访问终端原生功能的接口,因此通过本发明实施例提供的方法,该Web应用可以通过该原生功能访问模块任意访问终端所有的原生功能,且访问过程与Web应用所在的浏览器无关,不受浏览器功能的限制,从而提高了Web应用的使用性能。

[0142] 需要说明的是:上述实施例提供的Web应用访问终端原生功能的装置在访问终端的原生功能时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将设备的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的Web应用访问终端原生功能的装置与Web应用访问终端原生功能的方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0143] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0144] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来指令相关的硬件完成,所述的程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0145] 以上所述仅为本发明的较佳实施例,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、改进等,均应包含在本发明的保护范围之内。

[0146] 图3是本发明实施例提供的一种终端300的结构示意图。例如,终端300可以是移动电话,计算机或平板电脑等。

[0147] 参照图3,终端300可以包括以下一个或多个组件:处理组件302,存储器304,电源组件306,多媒体组件308,音频组件310,输入/输出(I/O)的接口312,传感器组件314,以及

通信组件316。

[0148] 处理组件302通常控制终端300的整体操作,诸如与显示,电话呼叫,数据通信,相机操作和记录操作相关联的操作。处理组件302可以包括一个或多个处理器320来执行指令,以完成上述的方法的全部或部分步骤。此外,处理组件302可以包括一个或多个模块,便于处理组件302和其他组件之间的交互。例如,处理组件302可以包括多媒体模块,以方便多媒体组件308和处理组件302之间的交互。

[0149] 存储器304被配置为存储各种类型的数据以支持在终端300的操作。这些数据的示例包括用于在终端300上操作的任何应用程序或方法的指令,联系人数据,电话簿数据,消息,图片,视频等。存储器304可以由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。

[0150] 电源组件306为终端300的各种组件提供电源。电源组件306可以包括电源管理系统,一个或多个电源,及其他与为终端300生成、管理和分配电源相关联的组件。

[0151] 多媒体组件308包括在所述终端300和用户之间的提供一个输出接口的屏幕。在一些实施例中,屏幕可以包括液晶显示器(LCD)和触摸面板(TP)。如果屏幕包括触摸面板,屏幕可以被实现为触摸屏,以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。所述触摸传感器可以不仅感测触摸或滑动动作的边界,而且还检测与所述触摸或滑动操作相关的持续时间和压力。在一些实施例中,多媒体组件308包括一个前置摄像头和/或后置摄像头。当终端300处于操作模式,如拍摄模式或视频模式时,前置摄像头和/或后置摄像头可以接收外部的多媒体数据。每个前置摄像头和后置摄像头可以是一个固定的光学透镜系统或具有焦距和光学变焦能力。

[0152] 音频组件310被配置为输出和/或输入音频信号。例如,音频组件310包括一个麦克风(MIC),当终端300处于操作模式,如呼叫模式、记录模式和语音识别模式时,麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器304或经由通信组件316发送。在一些实施例中,音频组件310还包括一个扬声器,用于输出音频信号。

[0153] I/O接口312为处理组件302和外围接口模块之间提供接口,上述外围接口模块可以是键盘,点击轮,按钮等。这些按钮可包括但不限于:主页按钮、音量按钮、启动按钮和锁定按钮。

[0154] 传感器组件314包括一个或多个传感器,用于为终端300提供各个方面的状态评估。例如,传感器组件314可以检测到终端300的打开/关闭状态,组件的相对定位,例如所述组件为终端300的显示器和小键盘,传感器组件314还可以检测终端300或终端300一个组件的位置改变,用户与终端300接触的存在或不存在,终端300方位或加速/减速和终端300的温度变化。传感器组件314可以包括接近传感器,被配置用来在没有任何的物理接触时检测附近物体的存在。传感器组件314还可以包括光传感器,如CMOS或CCD图像传感器,用于在成像应用中使用。在一些实施例中,该传感器组件314还可以包括加速度传感器,陀螺仪传感器,磁传感器,压力传感器或温度传感器。

[0155] 通信组件316被配置为便于终端300和其他设备之间有线或无线方式的通信。终端300可以接入基于通信标准的无线网络,如WiFi,2G或3G,或它们的组合。在一个示例性实施

例中,通信组件316经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中,所述通信组件316还包括近场通信(NFC)模块,以促进短程通信。例如,在NFC模块可基于射频识别(RFID)技术,红外数据协会(IrDA)技术,超宽带(UWB)技术,蓝牙(BT)技术和其他技术来实现。

[0156] 在示例性实施例中,终端300可以被一个或多个应用专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑器件(PLD)、现场可编程门阵列(FPGA)、控制器、微控制器、微处理器或其他电子元件实现,用于执行上述方法。

[0157] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器304,上述指令可由终端300的处理器320执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0158] 一种非临时性计算机可读存储介质,当所述存储介质中的指令由移动终端的处理器执行时,使得移动终端能够执行一种Web应用访问终端原生功能的方法,所述方法包括:

[0159] 接收Web应用发送的指定原生功能的访问请求;

[0160] 根据所述访问请求,在原生功能访问模块中查找指定原生功能插件,所述原生功能访问模块用于向所述Web应用提供访问终端原生功能的接口,所述指定原生功能插件用于向所述Web应用提供访问所述指定原生功能的接口;

[0161] 当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

[0162] 可选地,所述接收所述Web应用发送的指定原生功能的访问请求之前,还包括:

[0163] 将所述原生功能访问模块的配置文件中存储的多个原生功能插件加载到所述原生功能访问模块的插件管理器中;

[0164] 相应地,所述当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能,包括:

[0165] 判断所述插件管理器中加载的多个原生功能插件中是否包括所述指定原生功能插件;

[0166] 当所述插件管理器中加载的多个原生功能插件中包括所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能。

[0167] 可选地,所述接收所述Web应用发送的指定原生功能的访问请求之前,还包括:

[0168] 将所述原生功能访问模块的配置文件中存储的多个原生功能插件的标识加载到所述原生功能访问模块的插件管理器中;

[0169] 相应地,所述当在所述原生功能访问模块中查找到所述指定原生功能插件时,通过所述指定原生功能插件访问所述指定原生功能,包括:

[0170] 判断所述插件管理器中加载的多个原生功能插件的标识中是否包括所述指定原生功能插件的标识;

[0171] 当所述插件管理器中加载的多个原生功能插件的标识中包括所述指定原生功能插件的标识时,判断所述指定原生功能插件是否存储在所述插件管理器中;

[0172] 当所述指定原生功能插件已存储在所述插件管理器中时,通过所述指定原生功能插件访问所述指定原生功能。

[0173] 可选地,所述判断所述指定原生功能插件是否存储在所述插件管理器中之后,还包括:

[0174] 当所述指定原生功能插件未存储在所述插件管理器中时,根据所述指定原生功能插件的标识,将所述指定原生功能插件从服务器中下载至所述插件管理器中;

[0175] 通过所述指定原生功能插件访问所述指定原生功能。

[0176] 可选地,所述原生功能访问模块为安装在所述终端中与所述Web应用对应的模块。

[0177] 最后应说明的是:以上各实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述各实施例对本发明进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本发明各实施例技术方案的范围。

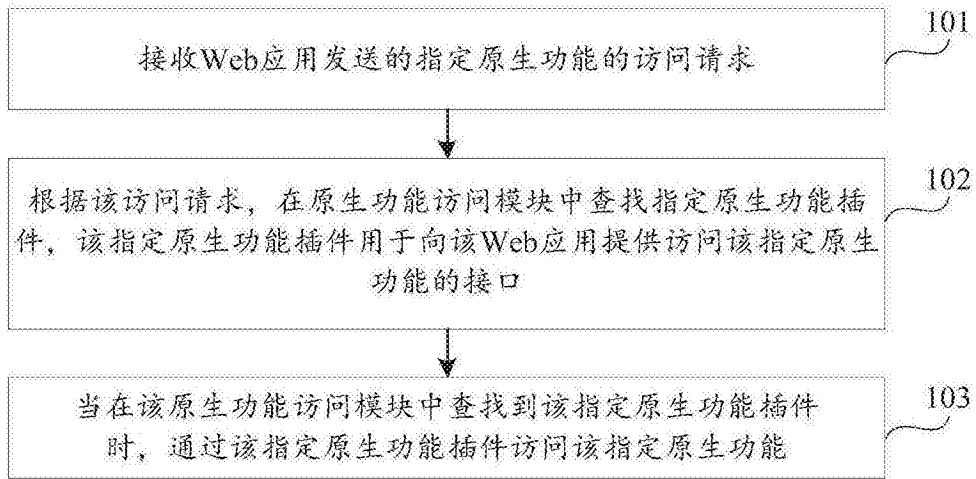


图1

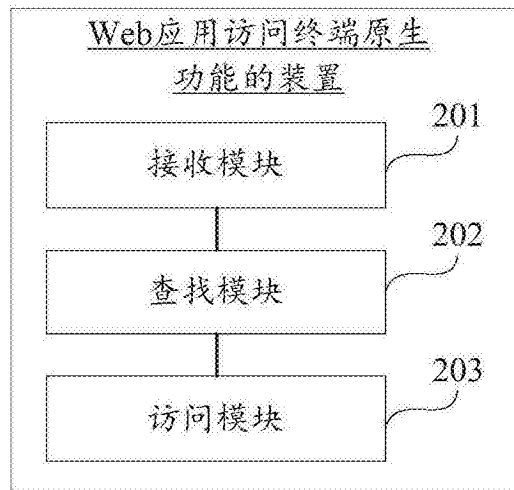


图2

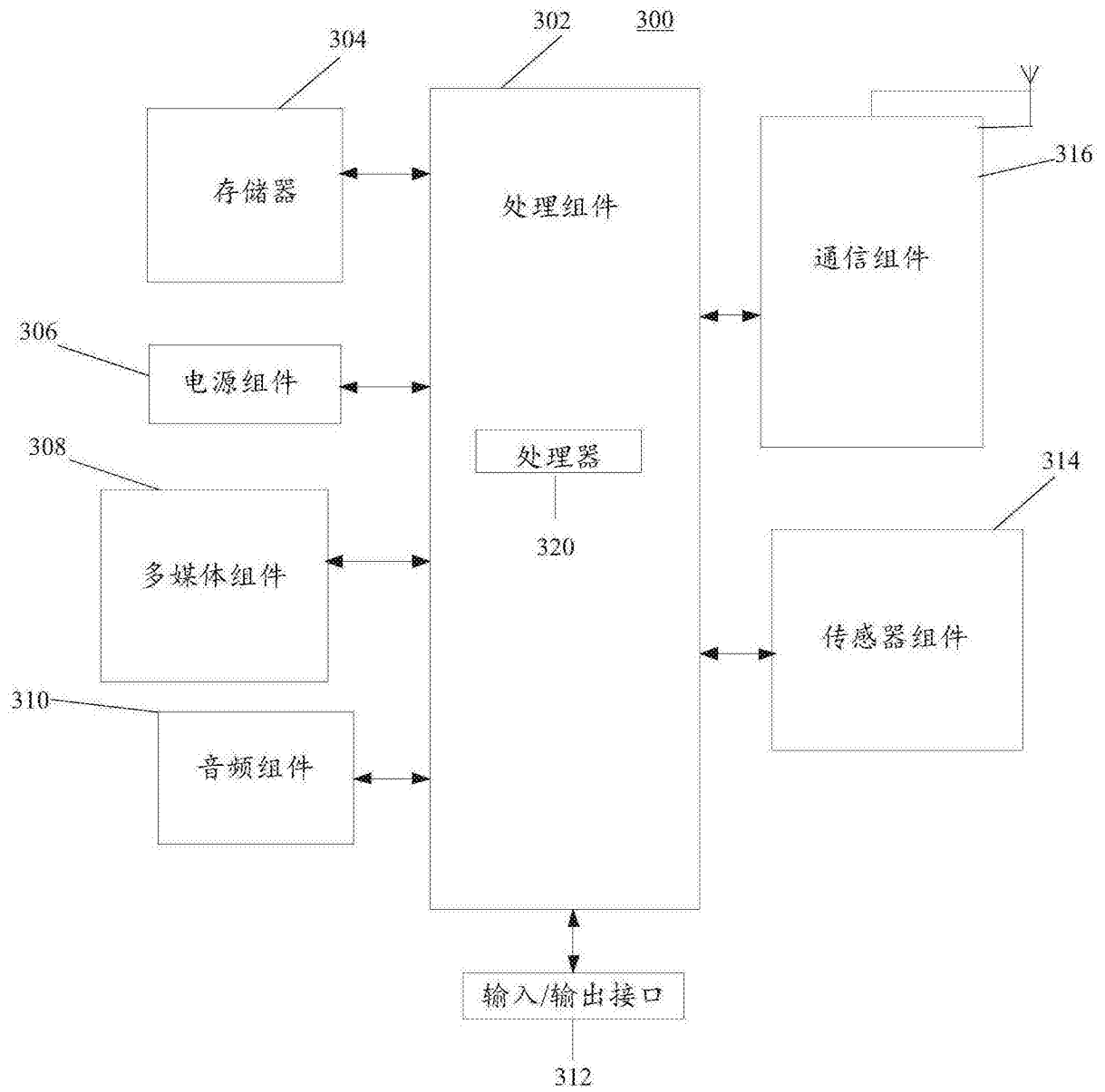


图3