

(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
2016年12月8日 (08.12.2016)

WIPO | PCT

(10) 国际公布号
WO 2016/192556 A1

- (51) 国际专利分类号:
G06F 9/54 (2006.01)
- (21) 国际申请号: PCT/CN2016/083359
- (22) 国际申请日: 2016年5月25日 (25.05.2016)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201510289070.8 2015年5月29日 (29.05.2015) CN
- (71) 申请人: 北京金山安全软件有限公司 (BEIJING KINGSOFT INTERNET SECURITY SOFTWARE CO., LTD.) [CN/CN]; 中国北京市海淀区小营西路33号二层东区, Beijing 100085 (CN)。
- (72) 发明人: 刘争辉 (LIU, Zhenghui); 中国北京市朝阳区朝阳北路237号复星国际中心12层, Beijing 100020 (CN)。
- (74) 代理人: 北京清亦华知识产权代理事务所(普通合伙) (TSINGYIHUA INTELLECTUAL PROPERTY LLC); 中国北京市海淀区清华园清华大学照澜院商业楼301室, Beijing 100084 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, 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, 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))。

(54) Title: INTERFACE INVOKING METHOD, DEVICE AND TERMINAL

(54) 发明名称: 接口调用方法、装置及终端

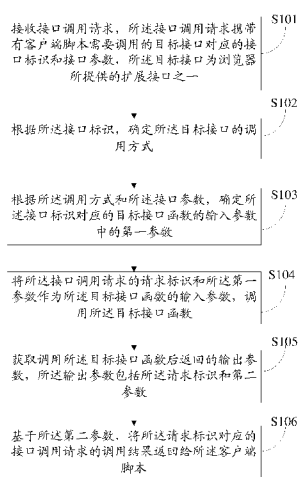


图1

S101 Receive an interface invoking request, the interface invoking request carrying an interface identifier and an interface parameter corresponding to a target interface required to be invoked by a client script, and the target interface being one of expansion interfaces supplied by a browser

S102 Determine, according to the interface identifier, an invoking mode of the target interface

S103 Determine, according to the invoking mode and the interface parameter, a first parameter in input parameters of a target interface function corresponding to the interface identifier

S104 Invoke the target interface function by utilizing a request identifier of the interface invoking request and the first parameter as the input parameters of the target interface function

S105 Acquire output parameters returned after invoking the target interface function, the output parameters comprising the request identifier and a second parameter

S106 Return, based on the second parameter, an invoking result of the interface invoking request corresponding to the request identifier to the client script

(57) Abstract: Disclosed in embodiments of the present invention are an interface invoking method and terminal. The method comprises: receiving an interface invoking request, the interface invoking request carrying an interface identifier and an interface parameter corresponding to a target interface required to be invoked by a client script; determining, according to the interface identifier, an invoking mode of the target interface; determining, according to the invoking mode and the interface parameter, a first parameter in input parameters of a target interface function corresponding to the target interface; invoking the target interface function by utilizing a request identifier of the interface invoking request and the first parameter as the input parameters of the target interface function; acquiring output parameters returned after invoking the target interface function, the output parameters comprising the request identifier and a second parameter; and returning, based on the second parameter, an invoking result of the interface invoking request corresponding to the request identifier to the client script. The embodiments of the present invention improve a development efficiency of an expansion interface.

(57) 摘要: 本发明实施例公开了一种接口调用方法及终端, 包括: 接收接口调用请求, 所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数; 根据所述接口标识, 确定所述目标接口的调用方式; 根据所述调用方式和所述接口参数, 确定所述目标接口对应的目标接口函数的输入参数中的第一参数; 将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数, 调用所述目标接口函数; 获取调用所述目标接口函数后返回的输出参数, 所述输出参数包括所述请求标识和第二参数; 基于所述第二参数, 将所述请求标识对应的

的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例, 可以提高扩展接口的开发效率。

WO 2016/192556 A1

接口调用方法、装置及终端

相关申请的交叉引用

本申请基于申请号为 201510289070.8，申请日为 2015 年 5 月 29 日的中国专利申请提出，并要求该中国专利申请的优先权，该中国专利申请的全部内容在此引入本申请作为参考。

技术领域

本发明涉及计算机技术领域，尤其涉及一种接口调用方法、装置及终端。

10 背景技术

随着互联网技术的发展，浏览器已成为移动终端上必备的应用软件之一，通过移动终端上的浏览器浏览网页时，浏览器内核对网页语法进行解释并渲染。通常，浏览器内核提供一些标准接口供客户端脚本使用或调用，因此，通过在网页中加载客户端脚本，由浏览器解释执行该客户端脚本，可以实现网页的动态展示效果，增强网页的交互性。例如：实现页面元素

15 的特殊显示效果、响应用户动作、验证表单数据等等。

然而，浏览器内核本身提供的标准接口有限，因此，为满足 Web 网页的开发需求，开发人员需根据需要开发出更多的扩展接口，以供 Web 网页的客户端脚本调用。目前，由于开发扩展接口的标准不统一，并没有一种统一来处理客户端脚本调用任意扩展接口的接口调用方法，因此，开发人员在开发扩展接口时，不仅需要针对该扩展接口所提供的功能设计一个接口函数，而且还需要针对该扩展接口设计出客户端脚本调用该接口的处理方法。例如：现有的接口开发方法如图 1a 所示，开发接口 A 和接口 B 时，除开发出接口 A 的接口函数和接口 B 的接口函数外，还需针对接口 A 开发一套脚本调用接口 A 时的调用方法和针对接口 B 开发一套脚本调用接口 B 时的调用方法，导致开发效率低。

20

25 发明内容

本发明实施例提供一种接口调用方法、装置及终端，可以提高扩展接口的开发效率。

第一方面，本发明实施例提供一种接口调用方法，包括：

接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；

30 根据所述接口标识，确定所述目标接口的调用方式；

根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数

中的第一参数；

将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；

5 获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；

基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

在第一方面的第一种可能的实现方式中，所述输出参数的数据类型为预设的第一数据类型；

10 所述根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数之后，所述方法还包括：

将所述第一参数进行格式转换，转换后的第一参数的数据类型为预设的第二数据类型；

所述获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数之后，所述方法还包括：

15 将所述第二参数进行格式转换，转换后的第二参数的数据类型为预设的第三数据类型。

结合第一方面或第一方面的第一种可能的实现方式，在第一方面的第二种可能的实现方式中，所述根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数包括：

20 若所述调用方式为同步调用，将所述接口参数确定为所述目标接口函数的输入参数中的第一参数；

若所述调用方式为异步调用，将所述接口参数中除预设参数外的其它参数确定为所述目标接口函数的输入参数中的第一参数，所述预设参数用于标识回调函数。

25 结合第一方面或第一方面的第一种可能的实现方式，在第一方面的第三种可能的实现方式中，所述基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本包括：

若所述调用方式为同步调用，将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本；

30 若所述调用方式为异步调用，确定所述请求标识对应的目标回调函数，并将所述第二参数作为所述目标回调函数的输入参数，运行所述目标回调函数得到输出结果，进而将所述输出结果作为所述接口调用请求的调用结果返回给所述客户端脚本。

结合第一方面的第三种可能的实现方式，在第一方面的第四种可能的实现方式中，所述

根据所述接口标识，确定所述目标接口的调用方式之后，所述方法还包括：

若所述调用方式为异步调用，将回调信息保存至回调数据库中，所述回调信息包括所述请求标识和所述预设参数所标识的回调函数；

所述确定所述请求标识对应的目标回调函数，包括：

- 5 若所述调用方式为异步调用，根据所述请求标识从所述回调数据库中查询所述请求标识对应的回调信息；

将所述回调信息指定的回调函数确定为目标回调函数。

结合第一方面的第四种可能的实现方式，在第一方面的第五种可能的实现方式中，所述将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本

- 10 之后，所述方法还包括：

删除所述回调数据库中所述请求标识对应的回调信息。

结合第一方面或第一方面的第一种可能的实现方式，在第一方面的第六种可能的实现方式中，所述根据所述接口标识，确定所述目标接口的调用方式包括：

- 15 根据所述接口标识，从预设的接口数据库中查询所述接口标识对应的接口信息，所述接口信息包括接口标识和调用方式；

将所述接口数据库中该接口标识对应的调用方式确定为所述目标接口的调用方式。

第二方面，本发明实施例提供一种接口调用装置，包括：

接收模块，用于接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；

- 20 方式确定模块，用于根据所述接口标识，确定所述目标接口的调用方式；

参数确定模块，用于根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数；

调用模块，用于将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；

- 25 获取模块，用于获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；

返回模块，用于基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

- 30 在第二方面的第一种可能的实现方式中，所述输出参数的数据类型为预设的第一数据类型，所述装置还包括：

第一转换模块，用于将所述第一参数进行格式转换，转换后的第一参数的数据类型为预

设的第二数据类型；

第二转换模块，用于将所述第二参数进行格式转换，转换后的第二参数的数据类型为预设的第三数据类型。

5 结合第二方面或第二方面的第一种可能的实现方式，在第二方面的第二种可能的实现方式中，所述参数确定模块具体用于：

若所述调用方式为同步调用，将所述接口参数确定为所述目标接口函数的输入参数中的第一参数；

若所述调用方式为异步调用，将所述接口参数中除预设参数外的其它参数确定为所述目标接口函数的输入参数中的第一参数，所述预设参数用于标识回调函数。

10 结合第二方面或第二方面的第一种可能的实现方式，在第二方面的第三种可能的实现方式中，所述返回模块包括：

同步返回单元，若所述调用方式为同步调用，将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本；

15 异步返回单元，若所述调用方式为异步调用，确定所述请求标识对应的目标回调函数，并将所述第二参数作为所述目标回调函数的输入参数，运行所述目标回调函数得到输出结果，进而将所述输出结果作为所述接口调用请求的调用结果返回给所述客户端脚本。

结合第二方面的第三种可能的实现方式，在第二方面的第四种可能的实现方式中，所述装置还包括：

20 保存模块，用于若所述调用方式为异步调用，将回调信息保存至回调数据库中，所述回调信息包括所述请求标识和所述预设参数所标识的回调函数；

所述同步返回单元具体用于：

若所述调用方式为异步调用，根据所述请求标识从所述回调数据库中查询所述请求标识对应的回调信息；

将所述回调信息指定的回调函数确定为目标回调函数。

25 结合第二方面的第四种可能的实现方式，在第二方面的第五种可能的实现方式中，所述装置还包括：

删除模块，用于删除所述回调数据库中所述请求标识对应的回调信息。

结合第二方面或第二方面的第五种可能的实现方式，在第二方面的第六种可能的实现方式中，所述方式确定模块具体用于：

30 根据所述接口标识，从预设的接口数据库中查询所述接口标识对应的接口信息，所述接口信息包括接口标识和调用方式；

将所述接口数据库中该接口标识对应的调用方式确定为所述目标接口的调用方式。

第三方面，本发明实施例提供一种接口调用装置，包括至少一个处理器和存储至少一个处理器可执行的指令的存储器。至少一个处理器用于：接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定所述目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

第四方面，本发明实施例提供一种接口调用装置，包括：接口注册器，用于记录接口的注册信息，所述接口的注册信息包括接口的接口标识和调用方式；接口数据库，用于存储各个接口的接口信息；调用请求数据库，用于存储调用的请求标识和调用方式；参数转换器，用于接收客户端脚本的接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定所述目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数；接口代理，用于获取所述接口调用请求所需的调用信息，所述调用信息包括请求标识、接口标识、第一参数和调用方式；并且根据获取到的所述调用信息重构接口调用请求；接口转发器，接收所述接口代理发送的重构的所述接口调用请求；将所述接口调用请求中的请求标识和调用方式添加至所述调用请求数据库；根据所述接口标识查询所述接口注册器以获取所述接口标识对应的目标接口函数，获取所述目标接口对应的目标接口函数的位置，将所述请求标识和第一参数作为查询到的目标接口函数的输入参数，调用所述目标接口函数；接口通知器，用于获取目标接口函数运行完毕后的输出参数，所述输出参数包括所述请求标识和第二参数，所述第二参数为所述请求标识对应的接口调用请求的调用结果；根据所述请求标识在所述调用请求数据库中查找对应的调用请求和调用方式；以及若所述调用方式为同步调用，将所述输出参数发送给所述接口代理

第五方面，本发明实施例提供一种终端，其特征在于，包括如第二方面、第三方面或第四方面所述的接口调用装置。

第六方面，本发明实施例提供一种计算机可读存储介质，具有存储于其中的指令，当终端的处理器执行该指令时，终端执行如上所述的接口调用方法。

第七方面，本发明实施例提供一种计算机程序，当其在处理器上运行时，执行如上所述

的接口调用方法。

本发明实施例中，接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定所述目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，只需开发该扩展接口对应的接口函数，可以提高扩展接口的开发效率。

附图说明

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。

图 1a 是现有方法提供的一种接口调用的示意图；

图 1 是本发明实施例提供的一种接口调用方法的流程示意图；

图 1b 是图 1 提供的一种接口调用的示意图；

图 2 是本发明实施例提供的一种接口调用方法的另一流程示意图；

图 3 是本发明实施例提供的一种接口调用装置的结构示意图；

图 4 是本发明实施例提供的一种接口调用装置的另一结构示意图；

图 4a 是图 4 提供的其中一种返回模块的结构示意图；

图 5 是本发明实施例提供的一种接口调用装置的又一结构示意图；

图 6 是本发明实施例提供的一种终端的结构示意图；

图 7 是本发明实施例提供的一种接口调用装置的再一结构示意图。

具体实施方式

下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅是本发明的一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，

都属于本发明保护的范围。

本发明实施例提供了一种接口调用方法、装置及终端，下面将结合附图 1-附图 7 对本发明实施例提供的一种接口调用方法、装置及终端分别进行详细介绍。

请参见图 1，图 1 是本发明实施例提供的一种接口调用方法的流程示意图，在本发明实
5 施例中，该方法可以包括以下步骤。

S101：接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一。

在本发明实施例中所描述的方法可以用于终端上的浏览器或类似于浏览器的产品中，所述终端可以是计算机、IPAD、手机等终端，也可以是其它智能设备，具体的，本发明实施例
10 不作任何限制。

通常，为丰富网页的动态展示效果，增强网页与用户的交互性，浏览器向网页提供了大量接口，供网页中加载的客户端脚本调用，浏览器解释、执行该客户端脚本。例如：浏览器接收到客户端脚本的接口调用请求时，根据该接口调用请求携带的信息运行该信息所指定的目标接口对应的接口函数，从而实现相应的功能。因此，浏览器在加载网页的过程中，若网
15 页中的客户端脚本需要调用浏览器提供的扩展接口，则接收接口调用请求。

S102：根据所述接口标识，确定所述目标接口的调用方式。

接口的调用方式包括同步调用和异步调用，其中，同步调用是一种阻塞式调用，调用方要等待被调用方执行完毕后才返回，属于单向调用；异步调用是一种双向调用，类似于消息或事件的处理机制，调用方仅发生操作请求，不等被调用方执行完毕而立即返回，被调用方
20 在执行完毕后主动调用调用方提供的回调函数(callback)，将执行结果返回给调用方，因此，异步调用可以提升浏览器执行网页的效率。通常，开发人员开发出扩展接口的接口函数后，会根据扩展接口所实现的功能预先设置该接口的调用方式。例如，假设客户端脚本调用接口 A 时，无需接口 A 的返回结果即可执行后续操作，则为提高网页的运行效率，可以将接口 A 的调用方式设置为异步调用，从而，接口 A 的接口参数中存在一个表示回调函数的参数，该
25 参数与接口需要实现的具体功能无关，用于返回接口的运行结果。再例如，假设客户端脚本调用接口 B 时，接口 B 的返回结果为后续操作的前提，则可以将接口 B 的调用方式设置为同步调用。

作为一种示例，可以是预设的接口数据库中记录有扩展接口的接口信息，所述接口信息包括接口标识和调用方式，所述接口标识可以是接口函数的接口名称，也可以是与接口名称
30 一一对应的其它标识符，具体的，本发明实施例不作限制；调用方式包括同步调用和异步调用。为方便说明，假设接口标识为接口名称，所提供的扩展接口有 3 个，则接口数据库的存

储形式可以如表 1 所示，表 1 是本发明实施例提供一种接口数据库的存储示意。

表 1

接口标识	调用方式
recordHistory	异步调用
downloadUrl	异步调用
playAudio	同步调用

从表 1 可知，在浏览器所提供的扩展接口中，接口标识分别为“recordHistory”和“downloadUrl”的接口所对应的调用方式为异步调用，接口标识为“playAudio”的接口所对应的调用方式为同步调用。因此，本发明实施例中，可以根据接口调用请求中所携带的接口标识，从接口数据库中查询出所述接口标识对应的目标接口的调用方式。

作为另一种示例，也可以是接口调用请求所携带的接口参数中包含有标识目标接口的调用方式的参数。例如：可以假设接口参数中的第一个参数用来表示目标接口的调用方式，用“1”表示同步调用，用“0”表示异步调用，当然，表示目标接口的调用方式的参数也可以是接口参数中位于其它位置处的参数，表示调用方式的具体数值也可以是其它字符，本发明实施例不作限制。但基于该假设，可以根据接口参数中第一个参数的具体数值确定目标接口的调用方式，若获取到该参数为“1”，则可以确定接口标识对应的目标接口的调用方式为同步调用；若为“0”，则可以确定接口标识对应的目标接口的调用方式为异步调用。值得说明的是，本发明实施例对标定目标接口的调用方式的参数的具体形式不作限制，预先约定好标定规则即可。本发明实施例中，可以根据接口调用请求中所携带的接口参数，确定所述接口标识对应的目标接口的调用方式。为方便后续步骤的说明，本发明实施例以第一种示例为例。

S103: 根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数。

由于浏览器在分别处理异步调用和同步调用的接口调用请求时存在差异，异步调用需调用方提供一个标识回调函数的参数，被调用方在执行完毕后通过该回调函数返回目标接口函数的执行结果；同步调用是被调用方在执行完毕后直接返回目标接口函数的执行结果。因此，若调用方式为异步调用，则接口调用请求所携带的接口参数中存在一个标识回调函数(callback)的预设参数，该预设参数由调用方提供，被调用方负责调用该预设参数标识的回调函数。

本发明实施例中，规范化客户端脚本调用接口的方式，例如，表示同步调用和异步调用的脚本可以如代码 1 所示。

代码 1

// 同步调用，操作结果通过返回值获得

5 function syncInterface(arg1, arg2,.....);

// 异步调用，操作结果通过回调函数 callback 获得

function asyncInterface(arg1, arg2,....., callback);

// 回调函数规范

function asyncCallback(Object);

10 从代码 1 可以看出，本发明实施例规范化了脚本调用接口的规范，其中，接口参数(arg1、arg2 等)的个数和数据类型根据实际需求设定。在异步调用中，可以将接口参数中的最后一个参数用来表示回调函数的参数，当然，也可以是接口参数中位于其它位置处的参数，具体的，本发明实施例不作限制，预先达成约定即可。异步调用中的回调函数用于将调用结果通知给客户端脚本，其中，回调函数的规范也在代码 1 中列出，它接受一个输入参数，输入
15 参数的类型可以是脚本支持的通用数据类型，例如：Object 对象。

本发明实施例中，若调用方式为同步调用，则根据接口调用请求携带的接口参数，可以直接确定该接口参数为目标接口对应的目标接口函数的输入参数中的第一参数。若调用方式为异步调用，则根据接口调用请求携带的接口参数，确定该接口参数中除预设参数外的其它参数为目标接口对应的目标接口函数的输入参数中的第一参数，其中，预设参数用于标识回
20 调函数，每个预设参数对应一个回调函数。

S104: 将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数。

具体实现中，在将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数之前，可以确定所述接口调用请求的请求标识。进一步
25 的，考虑到浏览器在加载网页的过程中，会存在同时处理多项接口调用请求的情况，因此，采用请求标识来标定客户端脚本调用接口的每项接口调用请求，该请求标识与其对应的接口调用请求的所有信息关联，所述信息包括该接口调用请求的接口标识、接口参数、调用结果等，从而，通过请求标识来跟踪每项接口调用请求及调用结果。本发明实施例中，接收到接口调用请求后，确定该接口调用请求的请求标识，其中，请求标识可以是浏览器内核接收到
30 接口调用请求时自动分配的。可知，一个请求标识唯一标定一项接口调用请求，且请求标识与该项接口调用请求所携带的接口标识和接口参数关联。

本发明实施例中，将接口函数的设计方式进行规范化，即每个接口函数接收一个请求标识作为输入参数的其中一个参数，且将该请求标识也作为输出参数的其中一个参数。例如：将请求标识作为接口函数的输入参数的其中一个参数，即输入参数包括请求标识和第一参数；将该请求标识作为接口函数的输出参数的其中一个参数，即输出参数包括请求标识和

5 第二参数，请求标识和第二参数关联。从而，接口函数执行完毕后，获取该接口函数的输出参数，输出参数包括请求标识和第二参数，通过请求标识区分该请求标识关联的第二参数为哪项接口调用请求的调用结果。

具体实现中，将请求标识和第一参数作为目标接口对应的目标接口函数的输入参数，调用该目标接口函数，即运行该目标接口函数。

10 S105：获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数。

本发明实施例中，接口函数的输出参数包括请求标识和第二参数，其中，第二参数为该接口函数实现接口功能后用来返回给客户端脚本的参数，请求标识用来标识该第二参数为客户端脚本的哪项接口调用请求的调用结果。

15 具体实现中，目标接口函数运行完毕后，获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数。

S106：基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

作为一种示例，若调用方式为同步调用，由于调用方需等待被调用方返回调用结果后才

20 执行后续操作，因此，将第二参数作为请求标识对应的接口调用请求的调用结果直接返回给调用方，其中，请求标识与该第二参数关联。

作为另一种示例，若调用方式为异步调用，由于调用方无需等待被调用方返回调用结果后即可继续执行后续操作，调用结果通过回调函数返回给调用方。因此，被调用方将第二参数作为回调函数的输入参数，通过调用该回调函数的方式返回请求标识对应的接口调用请求

25 的调用结果给调用方，其中，请求标识与该第二参数关联，回调函数为请求标识对应的接口调用请求所携带的接口参数中标定的预设参数。可以理解的，调用方为客户端脚本，被调用方为浏览器。从而，将接口调用请求的调用结果返回给客户端脚本，完成接口的调用。

为方便说明，可以将本发明方法理解为接口中间层，如图 1b 所示，图 1b 是图 1 提供的一种接口调用的示意图。浏览器在执行网页的过程中，若网页中的客户端脚本需要调用浏览器提供的扩展接口，则接口中间层接收接口调用请求，该接口调用请求中携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，其中，目标接口为浏览器所提供的扩展接

30

口中的任一扩展接口；采用上述方法调用目标接口对应的目标接口函数，并将该目标接口函数的运行结果作为该接口调用请求的调用结果返回给客户端脚本。从而，通过该接口中间层统一处理接口的调用流程，开发人员在开发扩展接口时，只需针对扩展接口开发出对应的接口函数即可，无需针对每个扩展接口都设计一套该接口的调用方法，降低工作量，提高开发效率。

在本发明实施例描述的接口调用方法中，接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，只需开发该扩展接口对应的接口函数，提高开发效率。

请参见图 2，图 2 是本发明实施例提供的一种接口调用方法的另一流程示意图，在本发明实施例中，该方法可以包括以下步骤。

S201：接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数。

S202：根据所述接口标识，确定所述目标接口的调用方式。

具体实现中，根据接口调用请求所要调用的目标接口，确定该目标接口的调用方式，所述调用方式包括同步调用和异步调用。通常，各接口的接口信息预先存储在接口数据库中，所述接口信息包括接口标识和调用方式。当浏览器接收到该接口调用请求时，根据该接口调用请求中携带的接口标识，从接口数据库中查询该接口标识对应的接口信息，接口信息包括接口标识和调用方式；从而，将接口数据库中该接口标识对应的调用方式确定为目标接口的调用方式。当然，目标接口的调用方式也可以是客户端脚本调用该目标接口时，在接口调用请求的接口参数中标定，具体的，本发明实施例不作限制。

但作为一种优选的实施方式，确定目标接口的调用方式可以是根据接口标识，从预设的接口数据库中查询该接口标识对应的接口信息，所述接口信息包括接口标识和调用方式；将从接口数据库中所查询到的该接口标识对应的调用方式确定为所述目标接口的调用方式。

S203: 根据所述调用方式和所述接口参数, 确定所述接口标识对应的目标接口函数的输入参数中的第一参数。

本发明实施例中, 异步调用需客户端脚本在接口参数中携带一个标识回调函数的参数, 为方便描述, 将标识回调函数的参数称为预设参数。例如, 可以将异步调用的接口参数中的最后一个参数用来存放预设参数, 当然, 也可以根据预先约定设置预设参数在接口参数中的其它位置, 具体的, 本发明实施例不作限制。在一种实现方式中, 若确定出调用方式为异步调用, 获取接口参数中标识回调函数的预设参数, 将回调信息保存至预设的回调数据库中, 所述回调信息包括所述请求标识和所述预设参数所标识的回调函数, 当然, 回调信息还可以包括运行该回调函数时所需的环境变量等其它信息, 具体的, 本发明实施例不作限制。从而, 方便后续步骤(例如, 步骤 S208)返回异步调用的调用结果时, 根据调用目标接口函数后返回的输出参数中的请求标识, 从该回调数据库中查询该请求标识对应的目标回调函数。

作为一种示例, 表示同步调用和异步调用的脚本可以如上述实施例中的代码 1 所示。可知, 同步调用的接口参数均为目标接口对应的目标接口函数所需要的参数, 异步调用的接口参数中用来表示回调函数的预设参数(callback)在目标接口对应的目标接口函数中不需要, 但在目标接口函数运行完毕后, 需要调用该预设参数所表示的回调函数, 返回调用结果。本发明实施例中, 根据目标接口的调用方式和接口参数来确定目标接口对应的目标接口函数的输入参数中的第一参数。具体实现中, 若调用方式为同步调用, 将接口调用请求中携带的接口参数确定为目标接口对应的目标接口函数的输入参数中的第一参数; 若调用方式为异步调用, 将接口调用请求携带的接口参数中除预设参数外的其它参数确定为目标接口函数的输入参数中的第一参数, 其中, 预设参数用于标识回调函数, 每个预设参数对应一个回调函数。

S204: 将所述第一参数进行格式转换, 转换后的第一参数的数据类型为预设的第二数据类型。

由于目前的浏览器大多为多核浏览器, 即浏览器中包括两个或两个以上的浏览器内核, 现有方法在针对多核浏览器开发扩展接口时, 需针对每一种浏览器内核都设计一个与该扩展接口所要实现的功能相同的接口函数, 不仅开发成本高, 而且由于不同的浏览器内核对接口函数的设计规范不一样, 不同的浏览器内核针对相同的接口标识所调用的接口函数不同, 导致接口调用流程复杂, 从而, 调用接口的效率低。

本发明实施例中, 将接口函数的设计规范化, 不管针对什么样的浏览器内核, 均采用相同的规范来设计接口函数。例如, 接口函数的设计规范可以如代码 2 所示, 在添加接口函数时, 开发人员只需根据该规范来实现相应的功能即可。值得说明的是, 代码 2 仅为示例, 本发明并不限于该示例。

// 接口函数规范, id 表示与接口调用请求对应的请求标识, kv_map 保存脚本传入的参数

```
Value InterfaceFunction(int id, KeyValueMap kv_map);
```

在该规范中, id 为请求标识, 用于跟踪该接口调用请求及调用结果; 接口函数的第一
5 参数 kv_map 的数据类型为预设的第二数据类型, 所述第二数据类型可以是预先定义的用于
接纳任何类型数据的数据结构类型, 也可以是编程语言自带的数据库中的通用数据类型, 该
数据类型可以接收任何数据类型的数据。例如: KeyValueMap 类型或 Value 类型, 具体的,
本发明实施例不作限制。为方便说明, 本发明实施例以第二数据类型为 KeyValueMap 类型为例,
10 但应当说明的是, 并不限制第二数据类型为 KeyValueMap 类型。其中, KeyValueMap 或
Value 是一种数据类型, 该类型像储物箱一样, 可以用来保存多种不同类型的数据, 例如:
整数、浮点数、字符串等等。具体的, KeyValueMap 是一种键-值 (Key-Value) 映射表, 该
映射表中保存了键 (Key) 和值 (Value) 的一一对应关系, 键 (Key) 可以是数字 (参数的
顺序) 或字符串 (参数的名称), 值 (Value) 是参数中对应某个键的具体值。

本发明实施例中, 将所确定的第一参数进行格式转换, 转换后的第一参数的数据类型为
15 预设的第二数据类型。即在调用目标接口对应的目标接口函数前需要把脚本传递进来的第一
参数进行转换, 如将第一参数转换成一个 KeyValueMap 对象等。

为方便说明, 假设客户端脚本欲调用一个添加网页历史记录接口, 且该接口的调用方式
为异步调用, 则客户端脚本提供的接口规范可以为: function recordHistory(url, date,
callback), 第一个参数表示需要传入网页的网址 (url, 假定为 http:\\www.liebao.cn),
20 第二个参数表示需要传入访问的时间 (time, 假定为 2014-07-20 20:00:00), 最后一个参数
为脚本函数对象, 该参数不作为调用接口函数的参数, 而用于在接口函数操作成功后通知脚
本, 因此, 步骤 S103 根据接口参数所确定的第一参数不包括将该参数。将所确定的第一参
数进行格式转换后, 得到转换后的 KeyValueMap 类型的第一参数可以如表 2 所示, 表 2 是本
发明实施例提供的一种第二数据类型的示意。

25 表 2

键 (Key)	值 (Value)
"1"	"http:\\www.liebao.cn"
"url"	
"2"	"2014-07-20 20:00:00"
"time"	

从而, 保证不同的浏览器内核接收到客户端脚本的接口调用请求后, 均可以通过本步骤

将第一参数转换为数据类型相同的参数，对接口函数的输入参数的数据类型实现了统一，因此，开发人员针对多核浏览器开发一套接口时只需设计一个接口函数。

S205: 将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数。

5 具体实现中，在将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数之前，可以确定所述接口调用请求的请求标识。考虑到浏览器在加载网页的过程中，会存在同时处理多项接口调用请求的情况，因此，采用请求标识来标定客户端脚本调用接口的每项接口调用请求，该请求标识与其对应的接口调用请求的所有信息关联，所述信息包括该接口调用请求的接口标识、接口参数、调用结果等，从而，
10 通过请求标识来跟踪每项接口调用请求及调用结果。本发明实施例中，接收到接口调用请求后，确定该接口调用请求的请求标识，其中，请求标识可以是浏览器内核接收到接口调用请求时自动分配的。可知，一个请求标识唯一标定一项接口调用请求，且请求标识与该项接口调用请求所携带的接口标识和接口参数关联。

通常，开发人员开发出一套扩展接口后，将该接口的相关信息保存在预设的接口数据库中，其中，接口的相关信息包括接口标识和接口函数，所述接口标识可以是接口名称。若开发人员已提供了接口调用请求所指定的目标接口，则由接口调用请求中携带的接口标识可以在预设的信息库中查找到一个唯一的与该接口标识匹配的接口函数。本发明实施例中，根据接口调用请求中携带的接口标识，查询接口数据库中是否存在与所述接口标识匹配的接口函数。若存在，说明浏览器提供该接口标识所指定的目标接口，根据接口调用请求对应的请求
20 标识和转换后的第一参数运行所查找到的目标接口函数；若不存在，说明浏览器不提供该目标接口，则返回预设的错误信息。

若查询到接口数据库中存在与接口标识匹配的目标接口函数，则将请求标识和步骤 S204 中转换后的第一参数作为所查找到的目标接口函数的输入参数，调用该目标接口函数。换句话说，将请求标识和转换后的第一参数代入所查找到的目标接口函数中，运行或执行所查找
25 到的目标接口函数。从而，针对不同的浏览器内核，采用本发明实施例后，可以采用统一的规范调用接口函数，不同的浏览器内核调用接口的方式相同，调用接口的效率高，网页的执行效率也得到提高。

S206: 获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数，所述输出参数的数据类型为预设的第一数据类型。

30 调用目标接口对应的目标接口函数后，获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数，所述输出参数的数据类型为预设的第一数据类

型，其中，第二参数用来表示目标接口函数的运行结果。具体实现中，所述第一数据类型可以是预先定义的用于接纳任何类型数据的数据类型，也可以是编程语言自带的数据库中的通用数据类型，该数据类型可以接收任何数据类型的数据。例如：Key-ValueMap 类型或 Value 类型，具体的，本发明实施例不作限制。为方便说明，代码 2 或本发明实施例中以第一数据类型为 Value 类型为例，但本发明实施例并不限制第一数据类型为 Value 类型。从而，开发人员针对不同的浏览器内核开发扩展接口时，只需根据规范设计一个接口函数，无需针对不同的浏览器内核分别设计一个接口函数，且无需设计该接口函数对应的接口的调用方法，不仅可以减少开发成本，而且提高开发效率。

值得说明的是，在具体实现中，也可以是第二参数的数据类型为预设的第一数据类型，接口函数的输出参数的数据类型为结构体，该结构体包括请求标识和第二参数。例如，以第一数据类型为 Value 类型为例，结构体的规范可以如代码 3 所示。

代码 3

```

struct InterfaceResult
{
15     Value val;
    int id;
};

```

S207：将所述第二参数进行格式转换，转换后的第二参数的数据类型为预设的第三数据类型。

作为一种优选的实施方式，将所获取到的调用目标接口函数后返回的输出参数中的第二参数进行格式转换，转换后的第二参数的数据类型为预设的第三数据类型。其中，第三数据类型可以是预先定义的用于接纳任何类型数据的数据结构类型，也可以是客户端脚本语言中的通用数据类型，例如，Object 对象，具体的，本发明实施例不作限制。从而，对返回给客户端脚本的调用结果进行规范化处理，提高脚本的执行效率。值得说明的是，也可以不针对第二参数进行格式转换，跳过本步骤而执行步骤 S208。

S208：基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

根据目标接口的调用方式，将第二参数作为该输出参数中的请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。具体实现中，若目标接口的调用方式为同步调用，将第二参数或经格式转换后的第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本；若所述调用方式为异步调用，根据所述请求标识，确定该请求标识对

应的目标回调函数，将第二参数或经格式转换后的第二参数作为所述目标回调函数的输入参数，运行所述目标回调函数得到输出结果，进而将所述输出结果作为所述接口调用请求的调用结果返回给客户端脚本。

5 在一种实施例中，若调用方式为异步调用，则确定所述请求标识对应的目标回调函数的方式可以是：根据所述请求标识，从所述回调数据库中查询所述请求标识对应的回调信息；将所述回调信息指定的回调函数确定为目标回调函数。同时，若获取到该回调信息中存储有运行该目标回调函数的其它信息，则根据这些信息设置当前的环境变量，保证正常运行该目标回调函数，将调用结果通知给客户端脚本。

10 作为一种可行的实施方式，将调用结果返回给客户端脚本后，删除所述回调数据库中所述请求标识对应的回调信息。从而，降低数据量，提高目标回调函数的查询效率。

在本发明实施例描述的接口调用方法中，接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，
15
20 只需开发该扩展接口对应的接口函数，提高开发效率。

下面为本发明装置实施例，本发明装置实施例与本发明方法实施例属于同一构思，用于执行本发明方法实施例中描述的方法。为了便于说明，本发明装置实施例仅示出与本发明装置实施例相关的部分，具体技术细节未揭示的，请参照本发明方法实施例的描述，此处不再一一赘述。
25

请参见图 3，图 3 是本发明实施例提供的一种接口调用装置的结构示意图，在本发明实施例中，该装置可以包括：接收模块 101、方式确定模块 102、参数确定模块 103、调用模块 104、获取模块 105 和返回模块 106。

接收模块 101，用于接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口中的任一扩展接口。
30

方式确定模块 102，用于根据所述接口标识，确定所述目标接口的调用方式。

参数确定模块 103，用于根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数。

调用模块 104，用于将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数。

获取模块 105，用于获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数。

返回模块 106，用于基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

10 在本发明实施例描述的接口调用装置中，接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，只需开发该扩展接口对应的接口函数，提高开发效率。

20

请参见图 4，图 4 是本发明实施例提供的一种接口调用装置的另一结构示意图，在本发明实施例中，该装置可以包括：接收模块 201、方式确定模块 202、参数确定模块 203、调用模块 204、获取模块 205、返回模块 206、第一转换模块 207、第二转换模块 208、保存模块 301 和删除模块 302。

25 接收模块 201，用于接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一。

方式确定模块 202，用于根据所述接口标识，确定所述目标接口的调用方式。

参数确定模块 203，用于根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数。

30 调用模块 204，用于将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数。

获取模块 205，用于获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数。

返回模块 206，用于基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

5 在一种实施例中，所述输出参数的数据类型为预设的第一数据类型。

第一转换模块 207，用于将所述第一参数进行格式转换，转换后的第一参数的数据类型为预设的第二数据类型。

第二转换模块 208，用于将所述第二参数进行格式转换，转换后的第二参数的数据类型为预设的第三数据类型。

10 保存模块 301，用于若所述调用方式为异步调用，将回调信息保存至回调数据库中，所述回调信息包括所述请求标识和所述预设参数所标识的回调函数。

删除模块 302，用于删除所述回调数据库中所述请求标识对应的回调信息。

在另一种实施例中，参数确定模块 203 具体用于：

15 若所述调用方式为同步调用，将所述接口参数确定为所述目标接口函数的输入参数中的第一参数；

若所述调用方式为异步调用，将所述接口参数中除预设参数外的其它参数确定为所述目标接口函数的输入参数中的第一参数，所述预设参数用于标识回调函数。

在另一种实施例中，返回模块 206 包括：同步返回单元 2061 和异步返回单元 2062，如图 4a 所示，图 4a 是图 4 提供的其中一种返回模块的结构示意图。

20 同步返回单元 2061，若所述调用方式为同步调用，将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本；

异步返回单元 2062，若所述调用方式为异步调用，确定所述请求标识对应的目标回调函数，并将所述第二参数作为所述目标回调函数的输入参数，运行所述目标回调函数得到输出结果，进而将所述输出结果作为所述接口调用请求的调用结果返回给客户端脚本。

25 在另一种实施例中，同步返回单元 2061 具体用于：

根据所述请求标识，从所述回调数据库中查询所述请求标识对应的回调信息；

将所述回调信息指定的回调函数确定为目标回调函数。

在另一种实施例中，方式确定模块 202 具体用于：

30 根据所述接口标识，从预设的接口数据库中查询所述接口标识对应的接口信息，所述接口信息包括接口标识和调用方式；

将所述接口数据库中该接口标识对应的调用方式确定为所述目标接口的调用方式。

在本发明实施例描述的接口调用装置中，接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，只需开发该扩展接口对应的接口函数，提高开发效率。

请参见图 5，图 5 是本发明实施例提供的一种接口调用装置的又一结构示意图，在本发明实施例中，该装置可以包括：参数转换器 301、接口代理 302、接口转发器 303、接口通知器 304、接口注册器 305、接口数据库 306 和调用请求数据库 307。

具体实现中，若接口调用指令所指定的目标接口的调用方式为异步调用，则该接口调用装置还可以包括回调数据库 308 和回调器 309。开发人员在添加接口时，需要向接口注册器 305 提供接口的接口标识和调用方式等注册信息，为方便说明，假定接口标识为接口名称，其中，接口函数的具体实现在主进程中提供。接口注册器 305 记录这些注册信息，并在浏览器创建渲染进程后，把这些注册信息通过进程间通信的方式发送给渲染进程，渲染进程把这些注册信息缓存在接口数据库 306 中，从而，接口数据库 306 中记录有各接口的接口信息，例如，接口数据库的存储形式可以如表 1 所示。回调数据库 308 用于调用方式为异步调用时，存储该异步调用的请求标识和该请求标识所标定的接口调用请求中携带的回调函数，从而，回调器 309 可以根据请求标识去回调数据库 308 中查询该请求标识对应的目标回调函数，通过调用该目标回调函数的方式向脚本返回调用结果。调用请求数据库 307 用于存储调用的请求标识和调用方式，从而，接口通知器 304 获取到目标接口函数运行完毕后的输出参数后，根据该输出参数中的请求标识去调用请求数据库 307 中查询该请求标识对应的调用方式，若为同步调用，则直接将输出参数发送给接口代理 302；若为异步调用，则将输出参数发送给回调器 309。值得说明的是，该接口调用装置可以用于单进程的浏览器中，也可以用于多进程的浏览器中，但为方便说明，以多进程的浏览器为例。

通常，浏览器在运行过程中大多表现为多进程，所述多进程至少包括渲染进程和主进程，其中，渲染进程用于处理网页内容的渲染，主进程用于处理用户界面和管理所有其它的进程

等。由于渲染进程只能执行网页，而不能执行其它操作，因此，扩展接口的具体实现（即接口函数的实现）必须在主进程中提供。为方便说明，假设浏览器在运行过程中启动了渲染进程和主进程，则在实现客户端脚本的接口调用流程中，渲染进程接收客户端脚本的接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口中的任一扩展接口。由于渲染进程本身不可以运行目标接口对应的目标接口函数，因此，根据客户端脚本传入的接口标识和接口参数，重新构造接口调用请求，向主进程发送该接口调用请求，从而，由主进程运行该接口调用请求所指定的目标接口函数，并将运行结果返回给渲染进程和/或客户端脚本，实现脚本的调用。

5 本发明实施例中，参数转换器 301 接收客户端脚本 201 的接口调用请求，所述接口调用请求携带有客户端脚本 201 需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接之一。参数转换器 301 查询接口数据库 306 中是否存在该接口标识对应的接口信息，若不存在，则说明接口调用错误，直接返回预设的错误信息；若存在，则确定该接口标识对应的接口名称和该接口标识对应的目标接口的调用方式。进一步的，根据该调用方式和所述接口参数确定目标接口对应的目标接口函数的输入参数中的第一参数。参数转换器 301 对第一参数进行格式转换，转换后的第一参数的数据类型为预设的第二数据类型，例如，KeyValuePair 类型或 Value 类型。

10 接口代理 302 (Interface Proxy) 从参数转换器 301 处获取接口调用请求所需的调用信息，或者是参数转换器 301 将调用信息发送给接口代理 302，所述调用信息包括接口调用请求的请求标识、接口名称、第一参数和调用方式，若调用方式为异步调用，所述调用信息还包括回调函数。当然，所述请求标识也可以是接口代理 302 确定的或生成的，具体的，本发明实施例不作限制。若调用方式为异步调用，则接口代理 302 根据所获取到的调用信息，将请求标识和与该请求标识对应的回调函数添加至回调数据库 308 中。以便回调器 309 接收到主进程中的接口通知器 304 返回的目标接口函数的输出参数后，根据该输出参数中的请求标识从回调数据库 308 中查询该请求标识对应的目标回调函数。同时，接口代理 302 向位于主进程中的接口转发器 303 发送接口调用请求，所述接口调用请求可以如代码 4 所示。

20 例如，若需要调用的目标接口的调用方式为异步调用，假设目标接口的接口名称为 recordHistory，则客户端脚本调用该目标接口的代码可以为：`recordHistory('http://www.liebao.cn', '2014-07-20 20:00:00', onRecordHistoryDone)`，可知，第一参数包括“`http://www.liebao.cn`”、“`2014-07-20 20:00:00`”，回调函数的函数名称为 `onRecordHistoryDone`。因此，参数转换器 301 接收客户端脚本调用目标接口 recordHistory 的接口调用请求，确定所述接口调用请求的请求标识和所述目标接口的调用方式；根据所述

调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数。参数转换器 303 对第一参数进行格式转换，格式转换后的第一参数可以如表 2 所示。接口代理 302 可以将标识该接口调用请求的请求标识 (request_id)、回调函数对象 (这里是 onRecordHistoryDone 函数) 以及正常执行该回调函数所需的环境变量等信息添加到回调数据库 308 中，保证回调器可以根据回调数据库 308 中保存的信息将调用结果成功返回给脚本。同时，接口代理 302 根据所获取到的调用信息重新构造接口调用请求，并通过进程间通信的方式将该接口调用请求发送给浏览器主进程中的接口转发器 303，接口的调用请求可以如代码 4 所示，该接口的调用请求可以携带请求标识 request_id、接口名称 interface、第一参数 arguments 和调用方式 sync，其中，第一参数的数据类型为预设的第二数据类型。

代码 4 中举例描述了异步接口的调用请求，其中，request_id 用来标识接口调用请求，interface 表示要调用目标接口对应的接口标识，arguments 是经过参数转换器 301 转换后得到的数据类型为预设的第一数据类型的第一参数，sync 用来表示目标接口的调用方式。

代码 4

```
//接口的调用请求
15  {
    "request_id" : 123456,
    "interface" : "recordHistory",
    "arguments" : {
        "1": "http:\\www.liebao.cn",
20     "2": "2014-07-20 20:00:00",
        "url": "http:\\www.liebao.cn",
        "time": "2014-07-20 20:00:00"
    },
    "sync" : false
25  }
```

若需要调用的目标接口为同步接口，则接口调用请求中的参数 sync 为 true，同理，接口代理 302 通过进程间通信的方式把所构造的接口的调用请求发送给浏览器主进程中的接口转发器 303。若调用方式为同步调用，则接口代理 302 向接口转发器 303 发送调用请求后，接口代理 302 会暂停执行，直到浏览器主进程执行完成并传递返回值后才继续。若调用方式为异步调用，则接口代理 302 向接口转发器 303 发送调用请求后，接口代理 302 会立即返回，从而，渲染进程可以继续执行其它操作，网页的执行效率高。

接口转发器 303 接收到接口代理发送的调用请求后, 根据该调用请求中所携带的调用信息, 将标识该调用请求的请求标识和调用方式添加至调用请求数据库 307 中, 从而, 位于主进程中的调用请求数据库 307 保存有各请求标识与调用请求所指定的目标接口的调用方式之间的对应关系。接口转发器 303 根据接口标识去接口注册器 305 中查询与该接口名称对应的目标接口函数, 获取目标接口对应的目标接口函数 401 的位置, 将请求标识 request_id 和第一参数 arguments 作为所查询到的目标接口函数 401 的输入参数, 调用该目标接口函数, 其中, 目标接口函数的实现方式可以是异步实现方式也可以是同步实现方式。

通常, 接口函数运行完毕后, 需要返回运行结果, 如操作是否成功, 要求读取的数据等等, 这些数据为接口函数的输出参数, 该输出参数的数据类型为预设的第一数据类型, 如 KeyValueType 类型或 Value 类型等。那么, 目标接口函数在执行完毕后, 会把输出参数传递给接口通知器 304, 其中, 输出参数包括请求标识和第二参数, 第二参数为所述请求标识对应的接口调用请求的调用结果, 从而, 接口通知器 304 根据请求标识在调用请求数据库 307 中查找所对应的调用请求和调用方式。

若调用方式为同步调用, 则接口通知器 304 将请求标识和第二参数发送给接口代理 302, 从而, 接口代理 302 接收到接口通知器 304 返回的目标接口函数的输出参数后, 继续执行其它操作。参数转换器 301 对输出参数中的第二参数进行格式转换, 转换后的第二参数的数据类型为预设的第三数据类型, 例如, 脚本内部的 Object 对象, 进而, 返回该请求标识对应的调用结果给脚本, 同步调用完成。在同步调用完成后, 根据请求标识删除调用请求数据库 307 中的对应项, 从而降低数据量, 提高查询效率。

若调用方式为同步调用, 接口代理 302 给接口转发器 303 发送接口的调用请求后会立即返回, 继续执行脚本的其它部分。而调用接口的返回结果需要通过调用请求所指定的回调函数来返回。因此, 接口通知器 304 根据输出参数中的请求标识从调用请求数据库 307 中查询得该请求标识对应的目标接口为异步调用后, 将请求标识和第二参数发送给渲染进程中的回调器 309, 回调器 309 采用该请求标识从回调数据库 308 中查询该请求标识对应的目标回调函数, 同时, 调用参数转换器 301 将第二参数进行格式转换, 转换后的第二参数的数据类型为预设的第三数据类型, 例如, 脚本的 Object 对象, 将转换后的第二参数作为目标回调函数的输入参数, 运行脚本提供的目标回调函数, 从而, 返回调用结果, 异步接口调用完成。在异步调用完成后, 根据请求标识删除调用请求数据库 307 和回调数据库 308 中的对应项, 从而降低数据量, 提高查询效率。

在本发明实施例描述的接口调用装置中, 接收接口调用请求, 所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数, 所述目标接口为浏览器所提供

的扩展接口之一；根据所述接口标识，确定目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，只需开发该扩展接口对应的接口函数，提高开发效率。

10 请参见图 6，图 6 是本发明实施例提供的一种终端的结构示意图，在本发明实施例中，终端 10 包括如上所述的接口调用装置。在本发明实施例中，所述终端可以包括：智能手机、电脑、平板电脑、笔记本电脑、掌上电脑以及移动互联网设备（MID）、PAD 等终端设备。具体的，本发明实施例不作限制。

值得说明的是，终端 10 包括如上所述的任意一种接口调用装置，该终端接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数；将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，只需开发该扩展接口对应的接口函数，提高开发效率。

25 请参见图 7，图 7 是本发明实施例提供的一种接口调用装置的再一结构示意图，用于执行本发明实施例公开的接口调用方法，该装置可以包括：至少一个处理器 401，例如 CPU，至少一个网络接口 404 或者其他用户接口 403，存储器 405，至少一个通信总线 402。通信总线 402 用于实现这些组件之间的连接通信。其中，用户接口 403 可选的可以包括 USB 接口以及其他标准接口、有线接口。网络接口 404 可选的可以包括 Wi-Fi 接口以及其他无线接口。存储器 405 可能包含高速 RAM 存储器，也可能还包括非不稳定的存储器（non-volatile memory），例如至少一个磁盘存储器。存储器 405 可选的可以包含至少一个位于远离前述处

理器 401 的存储装置。如图 7 所示,作为一种计算机存储介质的存储器 405 中可以包括操作系统 4051 及应用程序 4052。

在一些实施方式中,存储器 405 存储了如下的元素,可执行模块或者数据结构,或者他们的子集,或者他们的扩展集:

- 5 操作系统 4051, 包含各种系统程序, 用于实现各种基础业务以及处理基于硬件的任务;
应用程序 4052, 包含目标网络数据的数据分块参数的设置程序、目标数据块的划分程序、目标数据块与数据库中的数据块的比较程序以及目标数据块的删除程序等各种应用程序, 用于实现各种应用业务。

具体地, 处理器 401 用于调用存储器 405 中存储的程序, 执行以下操作:

- 10 接收接口调用请求, 所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数, 所述目标接口为浏览器所提供的扩展接口之一;

根据所述接口标识, 确定所述目标接口的调用方式;

根据所述调用方式和所述接口参数, 确定所述目标接口对应的目标接口函数的输入参数中的第一参数;

- 15 将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数, 调用所述目标接口函数;

获取调用所述目标接口函数后返回的输出参数, 所述输出参数包括所述请求标识和第二参数;

- 20 基于所述第二参数, 将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

在一种实施例中, 所述输出参数的数据类型为预设的第一数据类型;

处理器 401 执行所述根据所述调用方式和所述接口参数, 确定所述目标接口对应的目标接口函数的输入参数中的第一参数的操作之后, 还包括执行以下操作:

将所述第一参数进行格式转换, 转换后的第一参数的数据类型为预设的第二数据类型;

- 25 处理器 401 执行所述获取调用所述目标接口函数后返回的输出参数, 所述输出参数包括所述请求标识和第二参数的操作之后, 还包括执行以下操作:

将所述第二参数进行格式转换, 转换后的第二参数的数据类型为预设的第三数据类型。

在一种实施例中, 处理器 401 执行所述根据所述调用方式和所述接口参数, 确定所述目标接口函数的输入参数中的第一参数的操作, 包括执行以下操作:

- 30 若所述调用方式为同步调用, 将所述接口参数确定为所述目标接口函数的输入参数中的第一参数;

若所述调用方式为异步调用,将所述接口参数中除预设参数外的其它参数确定为所述目标接口函数的输入参数中的第一参数,所述预设参数用于标识回调函数。

在一种实施例中,处理器 401 执行所述基于所述第二参数,将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本,包括执行以下操作:

5 若所述调用方式为同步调用,将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本;

若所述调用方式为异步调用,确定所述请求标识对应的目标回调函数,并将所述第二参数作为所述目标回调函数的输入参数,运行所述目标回调函数得到输出结果,进而将所述输出结果作为所述接口调用请求的调用结果返回给客户端脚本。

10 在一种实施例中,处理器 401 执行所述根据所述接口标识,确定所述目标接口的调用方式的操作之后,还包括执行以下操作:

若所述调用方式为异步调用,将回调信息保存至回调数据库中,所述回调信息包括所述请求标识和所述预设参数所标识的回调函数;

15 其中,处理器 401 执行所述确定所述请求标识对应的目标回调函数的操作,包括执行以下操作:

若所述调用方式为异步调用,根据所述请求标识从所述回调数据库中查询所述请求标识对应的回调信息;

将所述回调信息指定的回调函数确定为目标回调函数。

20 在一种实施例中,处理器 401 执行所述将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本的操作之后,还包括执行以下操作:

删除所述回调数据库中所述请求标识对应的回调信息。

在一种实施例中,处理器 401 执行根据所述接口标识,确定所述目标接口的调用方式的操作,包括执行以下操作:

25 根据所述接口标识,从预设的接口数据库中查询所述接口标识对应的接口信息,所述接口信息包括接口标识和调用方式;

将所述接口数据库中该接口标识对应的调用方式确定为所述目标接口的调用方式。

30 在本发明实施例描述的技术方案中,接收接口调用请求,所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数,所述目标接口为浏览器所提供的扩展接口之一;根据所述接口标识,确定目标接口的调用方式;根据所述调用方式和所述接口参数,确定所述目标接口对应的目标接口函数的输入参数中的第一参数;将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数,调用所述目标接口函数;

获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。采用本发明实施例，通过规范化接口函数的设计和接口调用，浏览器可以采用统一的接口调用方法来实现客户端脚本对任一扩展接口的调用。从而，开发人员开发扩展接口时，

5 只需开发该扩展接口对应的接口函数，提高开发效率。

在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中，对上述术语的示意性表述不一定是必须针对相同的实施例或示例。而且，描述的具体特征、结构、材料或者特点可以在任一个或多个实施例或示例中以合适的方式结合。此外，在不相互矛盾的情况下，本领域的技术人员可以将本说明书中描述的不同实施例或示例以及不同实施例或示例的特征进行结合和组合。

10

本领域的技术人员可以将本说明书中描述的不同实施例以及不同实施例的特征进行结合和组合。本发明所有实施例中的模块或单元，可以通过通用集成电路，例如 CPU (Central Processing Unit, 中央处理器)，或通过 ASIC (Application Specific Integrated Circuit, 专用集成电路) 来实现。

15

本发明所有实施例方法中的步骤可以根据实际需要进行顺序调整、合并和删减；本发明所有实施例装置中的模块或单元可以根据实际需要进行合并、划分和删减。

20 流程图中或在此以其他方式描述的任何过程或方法描述可以被理解为，表示包括一个或多个用于实现特定逻辑功能或过程的步骤的可执行指令的代码的模块、片段或部分，并且本发明的优选实施方式的范围包括另外的实现，其中可以不按所示出或讨论的顺序，包括根据所涉及的功能按基本同时的方式或按相反的顺序，来执行功能，这应被本发明的实施例所属技术领域的技术人员所理解。

25 在流程图中表示或在此以其他方式描述的逻辑和/或步骤，例如，可以被认为是用于实现逻辑功能的可执行指令的定序列表，可以具体实现在任何计算机可读介质中，以供指令执行系统、装置或设备（如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统）使用，或结合这些指令执行系统、装置或设备而使用。就本说明书而言，“计算机可读介质”可以是任何可以包含、存储、通信、传播或传输程序以供指令执行系统、装置或设备或结合这些指令执行系统、装置或设备而使用的装置。计算机可读介质的更具体的示例（非穷尽性列表）包括以下：具有一个或多个布线的电连接部（电

30

子装置), 便携式计算机盘盒(磁装置), 随机存取存储器(RAM), 只读存储器(ROM), 可擦除可编辑只读存储器(EPROM或闪速存储器), 光纤装置, 以及便携式光盘只读存储器(CDROM)。另外, 计算机可读介质甚至可以是可在其上打印所述程序的纸或其他合适的介质, 因为可以例如通过对纸或其他介质进行光学扫描, 接着进行编辑、解译或必要时以其他合适方式进行处理来以电子方式获得所述程序, 然后将其存储在计算机存储器中。

本技术领域的普通技术人员可以理解实现上述实施例方法携带的全部或部分步骤是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, 该程序在执行时, 包括方法实施例的步骤之一或其组合。

此外, 在本发明各个实施例中的各功能单元可以集成在一个处理模块中, 也可以是各个单元单独物理存在, 也可以两个或两个以上单元集成在一个模块中。上述集成的模块既可以采用硬件的形式实现, 也可以采用软件功能模块的形式实现。所述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用, 也可以存储在一个计算机可读取存储介质中。

上述提到的存储介质可以是只读存储器, 磁盘或光盘等。尽管上面已经示出和描述了本发明的实施例, 可以理解的是, 上述实施例是示例性的, 不能理解为对本发明的限制, 本领域的普通技术人员在本发明的范围内可以对上述实施例进行变化、修改、替换和变型。

以上所揭露的仅为本发明的较佳实施例而已, 当然不能以此来限定本发明之权利范围, 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流程, 并依本发明权利要求所作的等同变化, 仍属于发明所涵盖的范围。

权利要求书

1、一种接口调用方法，其特征在于，包括：

接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；

根据所述接口标识，确定所述目标接口的调用方式；

根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数；

将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；

获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；

基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

2、如权利要求 1 所述的方法，其特征在于，所述输出参数的数据类型为预设的第一数据类型；

所述根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数之后，所述方法还包括：

将所述第一参数进行格式转换，转换后的第一参数的数据类型为预设的第二数据类型；

所述获取调用所述目标接口函数后返回的输出参数之后，所述方法还包括：

将所述第二参数进行格式转换，转换后的第二参数的数据类型为预设的第三数据类型。

3、如权利要求 1 或 2 所述的方法，其特征在于，所述根据所述调用方式和所述接口参数，确定所述目标接口对应的目标接口函数的输入参数中的第一参数包括：

若所述调用方式为同步调用，将所述接口参数确定为所述目标接口函数的输入参数中的第一参数；

若所述调用方式为异步调用，将所述接口参数中除预设参数外的其它参数确定为所述目标接口函数的输入参数中的第一参数，所述预设参数用于标识回调函数。

4、如权利要求 1 至 3 中任一项所述的方法，其特征在于，所述基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本包括：

若所述调用方式为同步调用，将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本；

若所述调用方式为异步调用，确定所述请求标识对应的目标回调函数，并将所述第二参数作为所述目标回调函数的输入参数，运行所述目标回调函数得到输出结果，进而将所述输出结果作为所述接口调用请求的调用结果返回给所述客户端脚本。

5 5、如权利要求 4 所述的方法，其特征在于，所述根据所述接口标识，确定所述目标接口的调用方式之后，所述方法还包括：

若所述调用方式为异步调用，将回调信息保存至回调数据库中，所述回调信息包括所述请求标识和所述预设参数所标识的回调函数；

所述确定所述请求标识对应的目标回调函数，包括：

10 若所述调用方式为异步调用，根据所述请求标识从所述回调数据库中查询所述请求标识对应的回调信息；

将所述回调信息指定的回调函数确定为目标回调函数。

6、如权利要求 5 所述的方法，其特征在于，所述将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本之后，所述方法还包括：

删除所述回调数据库中所述请求标识对应的回调信息。

15 7、如权利要求 1 至 6 中任一项所述的方法，其特征在于，所述根据所述接口标识，确定所述目标接口的调用方式包括：

根据所述接口标识，从预设的接口数据库中查询所述接口标识对应的接口信息，所述接口信息包括接口标识和调用方式；

将所述接口数据库中该接口标识对应的调用方式确定为所述目标接口的调用方式。

20 8、一种接口调用装置，其特征在于，包括：

接收模块，用于接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；

方式确定模块，用于根据所述接口标识，确定所述目标接口的调用方式；

25 参数确定模块，用于根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数；

调用模块，用于将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；

获取模块，用于获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；

30 返回模块，用于基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

9、如权利要求 8 所述的装置，其特征在于，所述输出参数的数据类型为预设的第一数据类型，所述装置还包括：

第一转换模块，用于将所述第一参数进行格式转换，转换后的第一参数的数据类型为预设的第二数据类型；

5 第二转换模块，用于将所述第二参数进行格式转换，转换后的第二参数的数据类型为预设的第三数据类型。

10、如权利要求 8 或 9 所述的装置，其特征在于，所述参数确定模块具体用于：

若所述调用方式为同步调用，将所述接口参数确定为所述目标接口函数的输入参数中的第一参数；

10 若所述调用方式为异步调用，将所述接口参数中除预设参数外的其它参数确定为所述目标接口函数的输入参数中的第一参数，所述预设参数用于标识回调函数。

11、如权利要求 8 至 10 中任一项所述的装置，其特征在于，所述返回模块包括：

同步返回单元，用于若所述调用方式为同步调用，将所述第二参数作为所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本；以及

15 异步返回单元，用于若所述调用方式为异步调用，确定所述请求标识对应的目标回调函数，并将所述第二参数作为所述目标回调函数的输入参数，运行所述目标回调函数得到输出结果，进而将所述输出结果作为所述接口调用请求的调用结果返回给客户端脚本。

12、如权利要求 8 至 11 中任一项所述的装置，其特征在于，还包括：

20 保存模块，用于若所述调用方式为异步调用，将回调信息保存至回调数据库中，所述回调信息包括所述请求标识和所述预设参数所标识的回调函数，

所述同步返回单元用于：

根据所述请求标识从所述回调数据库中查询所述请求标识对应的回调信息；

将所述回调信息指定的回调函数确定为目标回调函数。

13、如权利要求 12 所述的装置，其特征在于，还包括：

25 删除模块，用于删除所述回调数据库中所述请求标识对应的回调信息。

14、如权利要求 8 至 13 中任一项所述的装置，其特征在于，所述方式确定模块用于：

根据所述接口标识，从预设的接口数据库中查询所述接口标识对应的接口信息，所述接口信息包括接口标识和调用方式；

将所述接口数据库中该接口标识对应的调用方式确定为所述目标接口的调用方式。

30 15、一种接口调用装置，其特征在于，包括：

接口注册器，用于记录接口的注册信息，所述接口的注册信息包括接口的接口标识和调

用方式；

接口数据库，用于存储各个接口的接口信息；

调用请求数据库，用于存储调用的请求标识和调用方式；

- 5 参数转换器，用于接收客户端脚本的接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；根据所述接口标识，确定所述目标接口的调用方式；根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数中的第一参数；

接口代理，用于获取所述接口调用请求所需的调用信息，所述调用信息包括请求标识、接口标识、第一参数和调用方式；并且根据获取到的所述调用信息重构接口调用请求；

- 10 接口转发器，接收所述接口代理发送的重构的所述接口调用请求；将所述接口调用请求中的请求标识和调用方式添加至所述调用请求数据库；根据所述接口标识查询所述接口注册器以获取所述接口标识对应的目标接口函数，获取所述目标接口对应的目标接口函数的位置，将所述请求标识和第一参数作为查询到的目标接口函数的输入参数，调用所述目标接口函数；

- 15 接口通知器，用于获取目标接口函数运行完毕后的输出参数，所述输出参数包括所述请求标识和第二参数，所述第二参数为所述请求标识对应的接口调用请求的调用结果；根据所述请求标识在所述调用请求数据库中查找对应的调用请求和调用方式；以及若所述调用方式为同步调用，将所述输出参数发送给所述接口代理。

16、如权利要求 15 所述的接口调用装置，其特征在于，还包括：

- 20 回调数据库，用于若调用方式为异步调用，存储所述异步调用的请求标识和所述请求标识所标定的接口调用请求中携带的回调函数；

回调器，用于根据所述请求标识查询所述回调数据库以获取所述请求标识对应的目标回调函数，并通过调用所述目标回调函数的方式向所述客户端脚本返回调用结果。

17、一种接口调用装置，其特征在于，包括：

- 25 至少一个处理器；

存储器，用于存储所述至少一个处理器可执行的指令；

所述至少一个处理器用于：

接收接口调用请求，所述接口调用请求携带有客户端脚本需要调用的目标接口对应的接口标识和接口参数，所述目标接口为浏览器所提供的扩展接口之一；

- 30 根据所述接口标识，确定所述目标接口的调用方式；

根据所述调用方式和所述接口参数，确定所述接口标识对应的目标接口函数的输入参数

中的第一参数；

将所述接口调用请求的请求标识和所述第一参数作为所述目标接口函数的输入参数，调用所述目标接口函数；

5 获取调用所述目标接口函数后返回的输出参数，所述输出参数包括所述请求标识和第二参数；

基于所述第二参数，将所述请求标识对应的接口调用请求的调用结果返回给所述客户端脚本。

18、一种终端，其特征在于，包括如权利要求 8-17 任一项所述的接口调用装置。

19、一种计算机可读存储介质，具有存储于其中的指令，当终端的处理器执行所述指令
10 时，所述终端执行如权利要求 1-7 中任一项所述的接口调用方法。

20、一种计算机程序，当其在处理器上运行时，执行如权利要求 1-7 中任一项所述的接口调用方法。

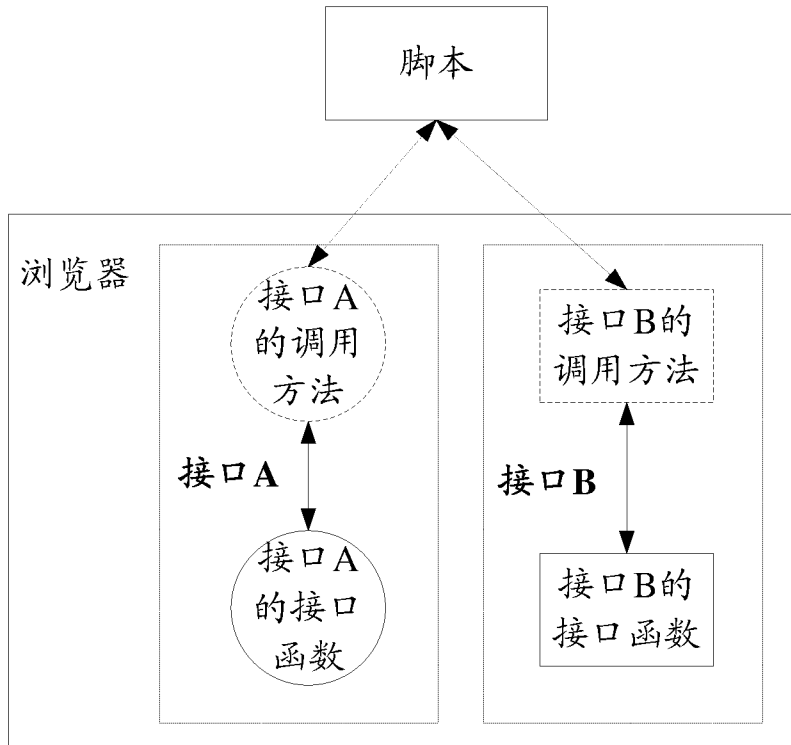


图 1a

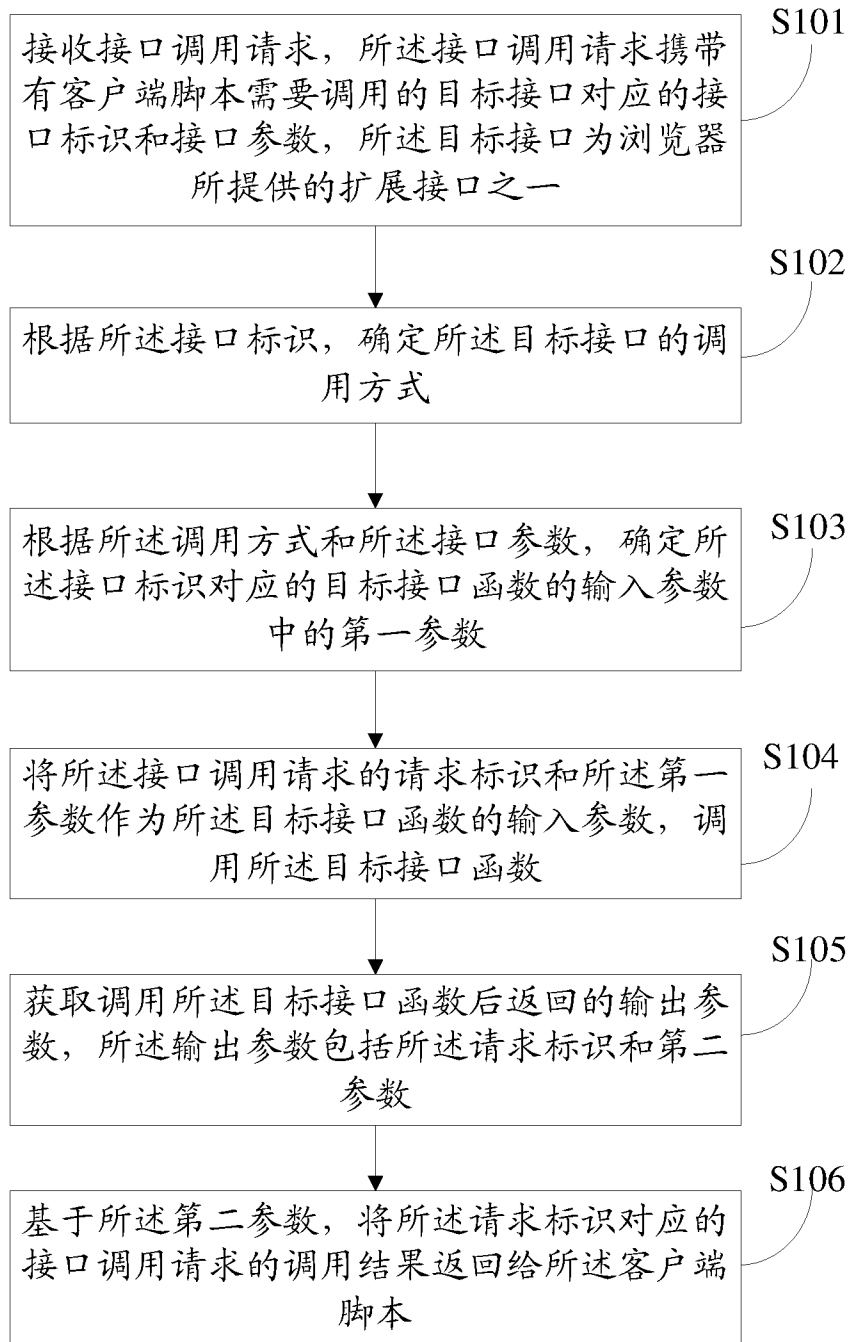


图1

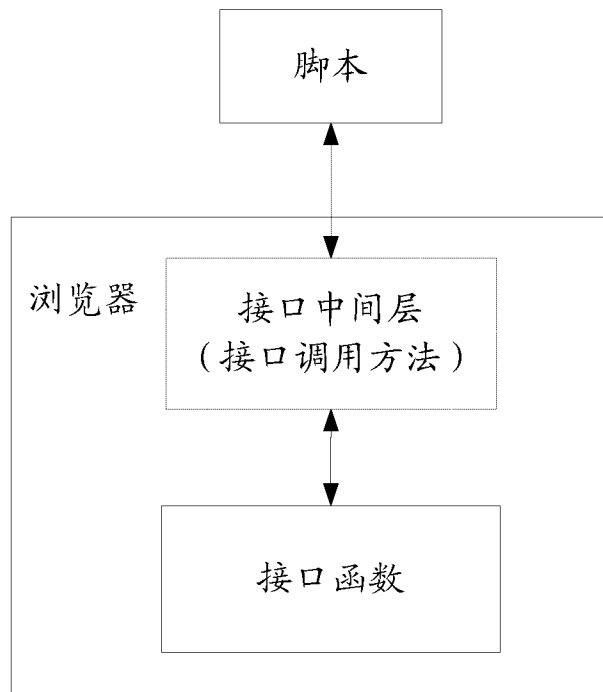


图 1b

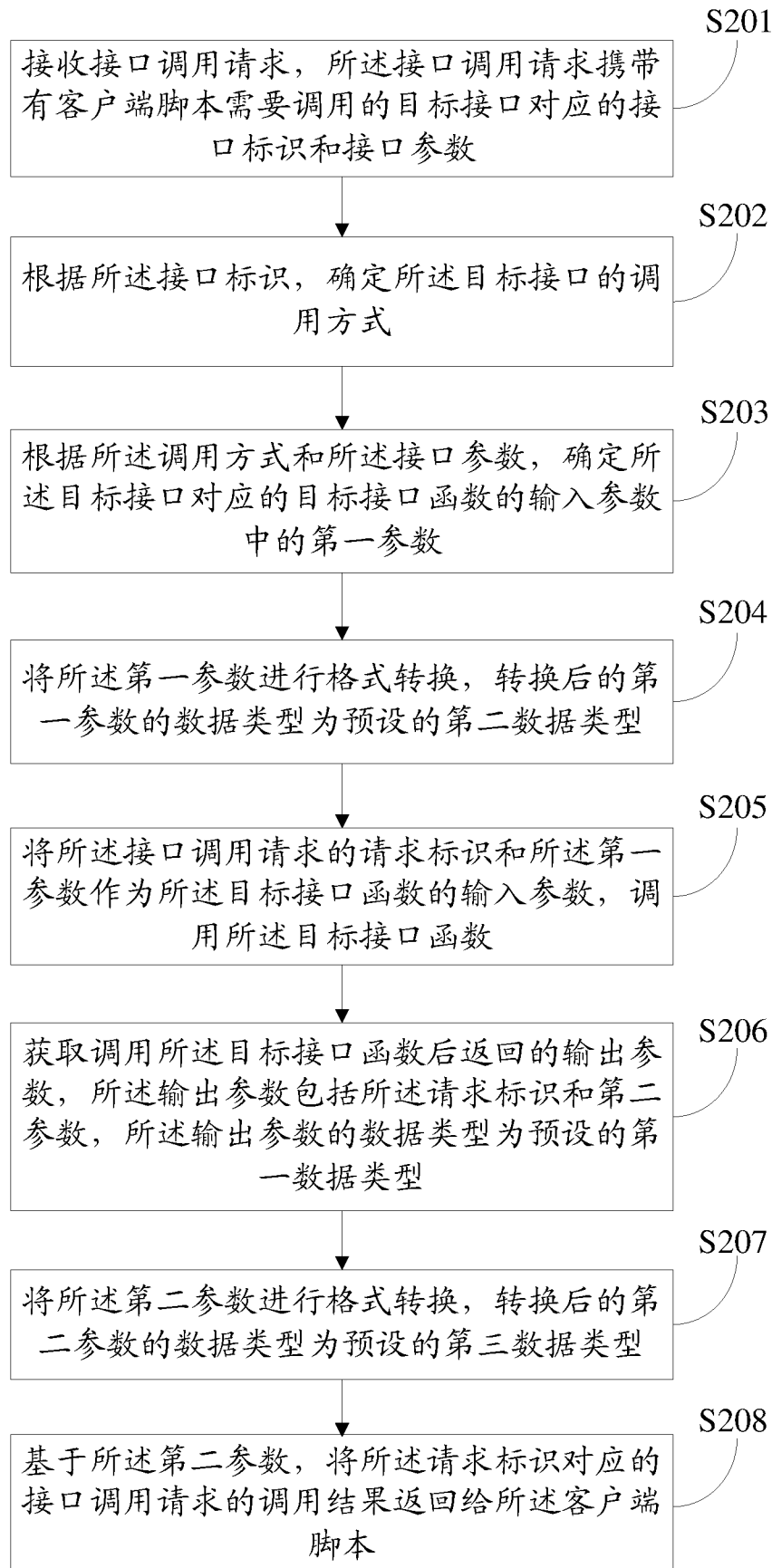


图 2

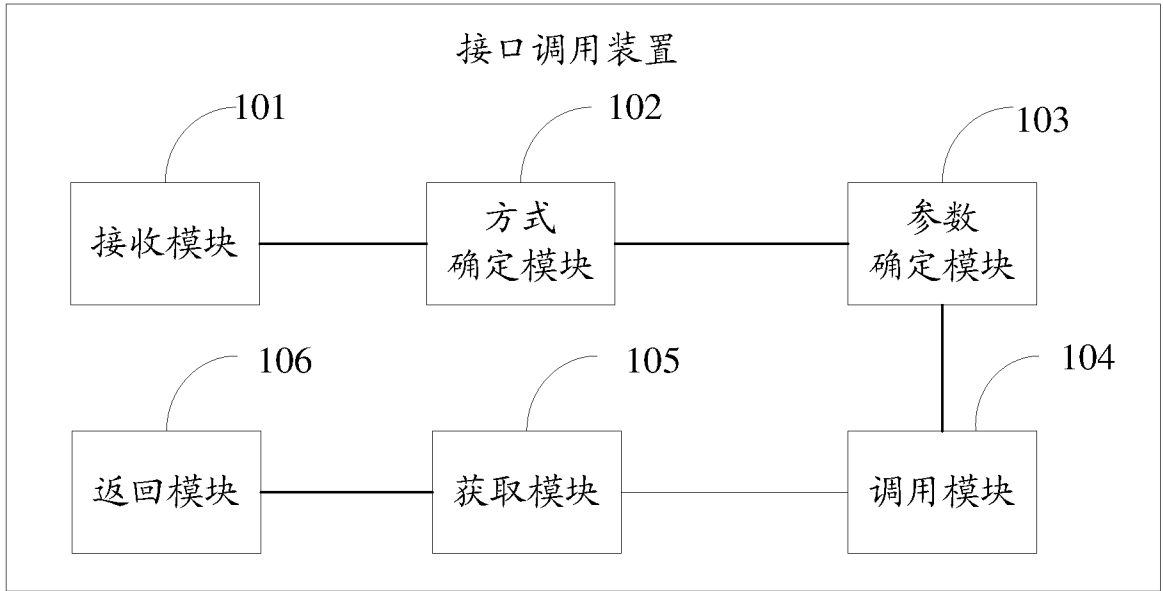


图 3

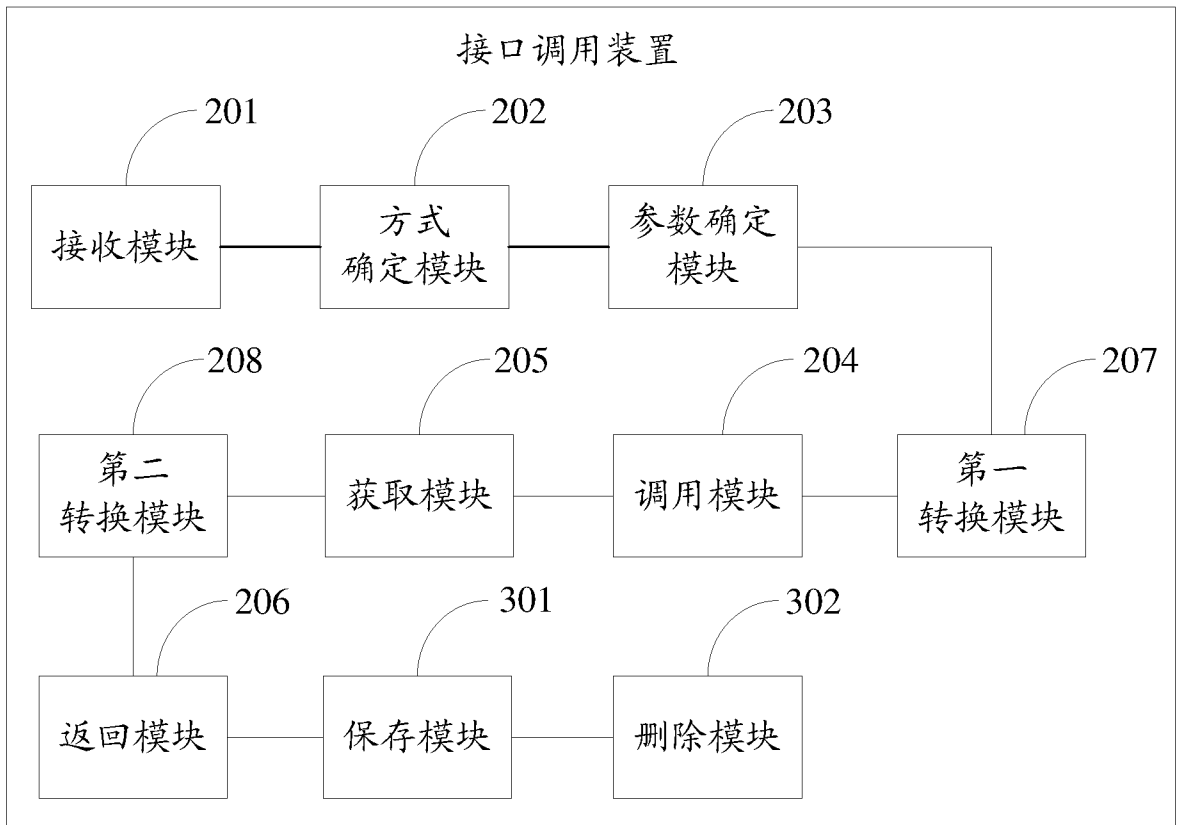


图 4

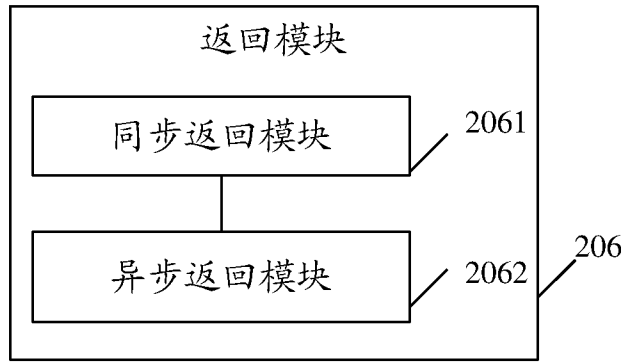


图 4a

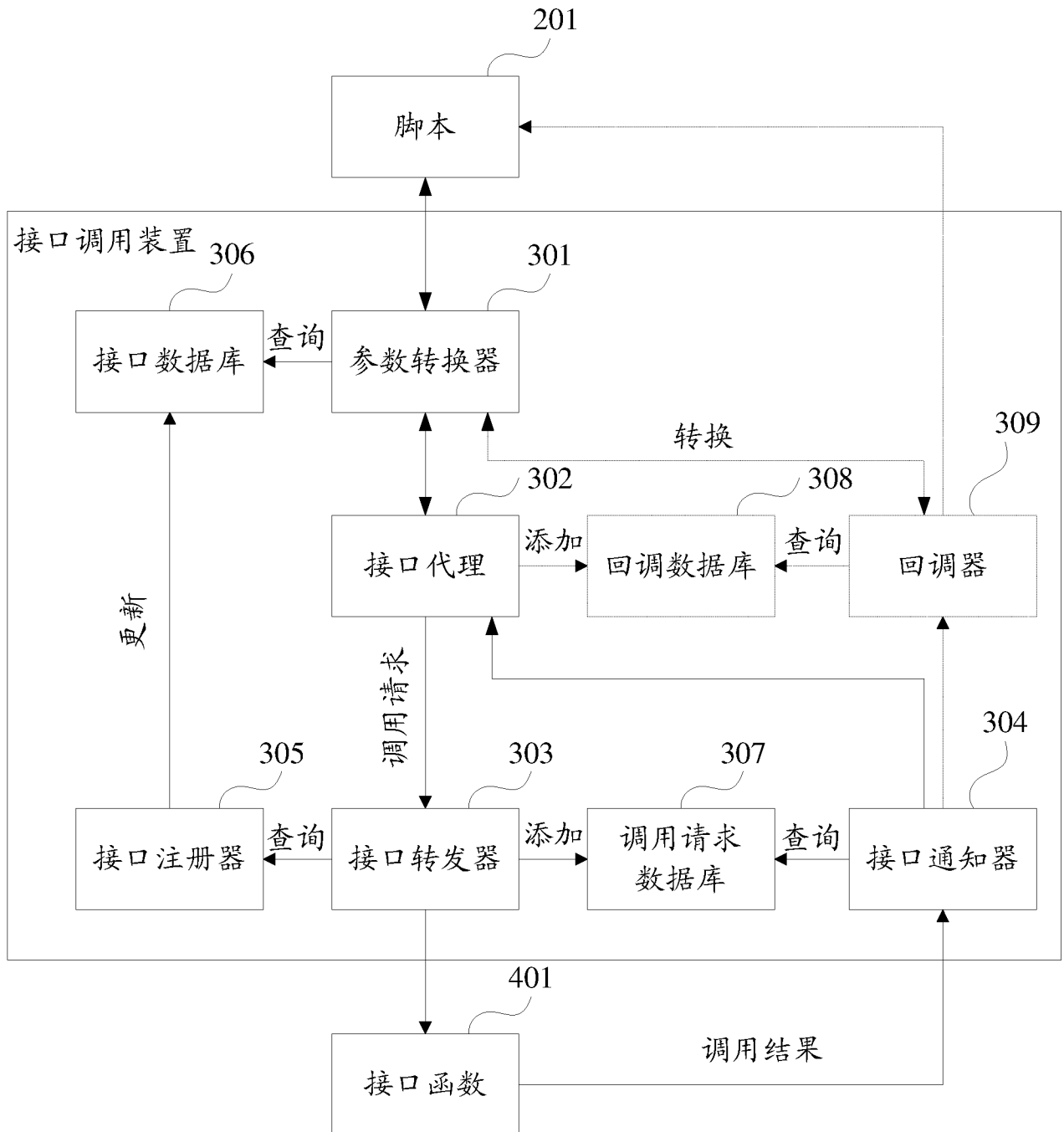


图 5

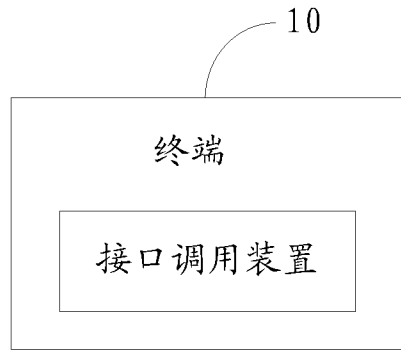


图6

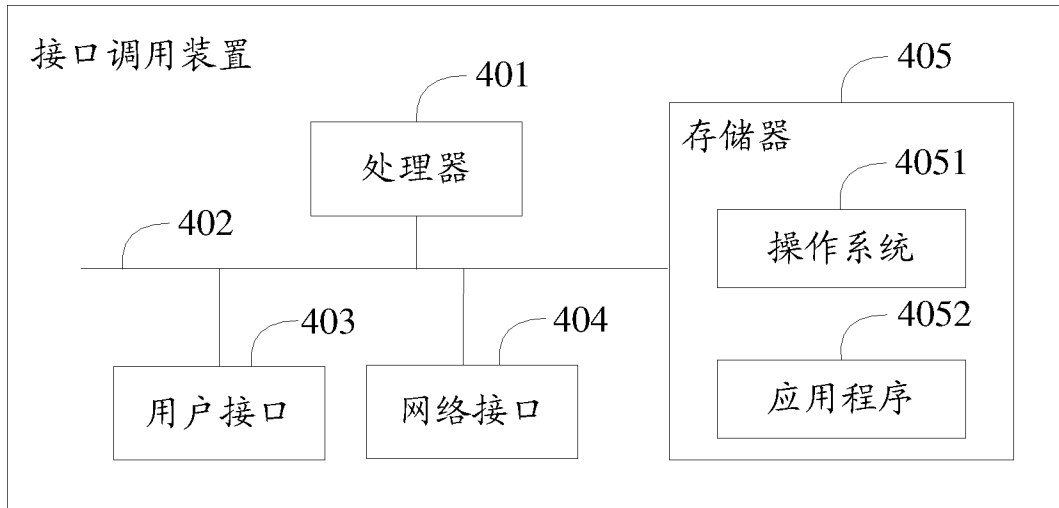


图7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2016/083359

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/54 (2006.01) 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)

CNABS, DWPI, CNKI: interface, call, script, parameter, browser, page, web, synchronization, asynchronism, input, output, format, type

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P, X	CN 104866383 A (BEIJING KINGSOFT CO., LTD.) 26 August 2015 (26.08.2015) the whole document	1-20
A	CN 104281455 A (BEIJING KINGSOFT CO., LTD.) 14 January 2015 (14.01.2015) the whole document	1-20
A	CN 103177040 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 26 June 2013 (26.06.2013) the whole document	1-20
A	CN 103309728 A (TENCENT TECHNOLOGY SHENZHEN CO., LTD.) 18 September 2013 (18.09.2013) the whole document	1-20

Further documents are listed in the continuation of Box C.

See patent family annex.

<p>* Special categories of cited documents:</p> <p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier application or patent but published on or after the international filing date</p> <p>“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)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p>	<p>“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</p> <p>“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</p> <p>“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</p> <p>“&” document member of the same patent family</p>
---	---

Date of the actual completion of the international search

23 August 2016

Date of mailing of the international search report

05 September 2016

Name and mailing address of the ISA
State Intellectual Property Office of the P. R. China
No. 6, Xitucheng Road, Jimenqiao
Haidian District, Beijing 100088, China
Facsimile No. (86-10) 62019451

Authorized officer

ZHANG, Yan

Telephone No. (86-10) 62089421

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2016/083359

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 104866383 A	26 August 2015	None	
CN 104281455 A	14 January 2015	None	
CN 103177040 A	26 June 2013	None	
CN 103309728 A	18 September 2013	CN 103309728 B	03 August 2016

<p>A. 主题的分类</p> <p>G06F 9/54 (2006.01) i</p> <p>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</p>																	
<p>B. 检索领域</p> <p>检索的最低限度文献 (标明分类系统和分类号)</p> <p>G06F</p> <p>包含在检索领域中的除最低限度文献以外的检索文献</p> <p>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))</p> <p>CNABS, DWPI, CNKI: 接口, 调用, 脚本, 参数, 浏览器, 页面, 网页, 同步, 异步, 输入, 输出, 格式, 类型, interface, call, script, parameter, browser, page, web, synchronization, asynchronism, input, output, format, type</p>																	
<p>C. 相关文件</p> <table border="1"> <thead> <tr> <th>类型*</th> <th>引用文件, 必要时, 指明相关段落</th> <th>相关的权利要求</th> </tr> </thead> <tbody> <tr> <td>PX</td> <td>CN 104866383 A (北京金山安全软件有限公司) 2015年 8月 26日 (2015 - 08 - 26) 全文</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>CN 104281455 A (北京金山安全软件有限公司) 2015年 1月 14日 (2015 - 01 - 14) 全文</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>CN 103177040 A (腾讯科技深圳有限公司) 2013年 6月 26日 (2013 - 06 - 26) 全文</td> <td>1-20</td> </tr> <tr> <td>A</td> <td>CN 103309728 A (腾讯科技深圳有限公司) 2013年 9月 18日 (2013 - 09 - 18) 全文</td> <td>1-20</td> </tr> </tbody> </table>			类型*	引用文件, 必要时, 指明相关段落	相关的权利要求	PX	CN 104866383 A (北京金山安全软件有限公司) 2015年 8月 26日 (2015 - 08 - 26) 全文	1-20	A	CN 104281455 A (北京金山安全软件有限公司) 2015年 1月 14日 (2015 - 01 - 14) 全文	1-20	A	CN 103177040 A (腾讯科技深圳有限公司) 2013年 6月 26日 (2013 - 06 - 26) 全文	1-20	A	CN 103309728 A (腾讯科技深圳有限公司) 2013年 9月 18日 (2013 - 09 - 18) 全文	1-20
类型*	引用文件, 必要时, 指明相关段落	相关的权利要求															
PX	CN 104866383 A (北京金山安全软件有限公司) 2015年 8月 26日 (2015 - 08 - 26) 全文	1-20															
A	CN 104281455 A (北京金山安全软件有限公司) 2015年 1月 14日 (2015 - 01 - 14) 全文	1-20															
A	CN 103177040 A (腾讯科技深圳有限公司) 2013年 6月 26日 (2013 - 06 - 26) 全文	1-20															
A	CN 103309728 A (腾讯科技深圳有限公司) 2013年 9月 18日 (2013 - 09 - 18) 全文	1-20															
<p><input type="checkbox"/> 其余文件在C栏的续页中列出。</p> <p><input checked="" type="checkbox"/> 见同族专利附件。</p>																	
<p>* 引用文件的具体类型:</p> <table border="0"> <tr> <td>“A” 认为不特别相关的表示了现有技术一般状态的文件</td> <td>“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</td> </tr> <tr> <td>“E” 在国际申请日的当天或之后公布的在先申请或专利</td> <td>“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</td> </tr> <tr> <td>“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)</td> <td>“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</td> </tr> <tr> <td>“O” 涉及口头公开、使用、展览或其他方式公开的文件</td> <td>“&” 同族专利的文件</td> </tr> <tr> <td>“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</td> <td></td> </tr> </table>			“A” 认为不特别相关的表示了现有技术一般状态的文件	“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	“E” 在国际申请日的当天或之后公布的在先申请或专利	“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	“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)	“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	“O” 涉及口头公开、使用、展览或其他方式公开的文件	“&” 同族专利的文件	“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						
“A” 认为不特别相关的表示了现有技术一般状态的文件	“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																
“E” 在国际申请日的当天或之后公布的在先申请或专利	“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																
“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)	“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																
“O” 涉及口头公开、使用、展览或其他方式公开的文件	“&” 同族专利的文件																
“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																	
<p>国际检索实际完成的日期</p> <p>2016年 8月 23日</p>		<p>国际检索报告邮寄日期</p> <p>2016年 9月 5日</p>															
<p>ISA/CN的名称和邮寄地址</p> <p>中华人民共和国国家知识产权局 (ISA/CN) 中国北京市海淀区蓟门桥西土城路6号 100088</p> <p>传真号 (86-10) 62019451</p>		<p>授权官员</p> <p>张妍</p> <p>电话号码 (86-10) 62089421</p>															

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2016/083359

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	104866383	A	2015年 8月 26日	无	
CN	104281455	A	2015年 1月 14日	无	
CN	103177040	A	2013年 6月 26日	无	
CN	103309728	A	2013年 9月 18日	CN 103309728	B 2016年 8月 3日