



(12) 发明专利

(10) 授权公告号 CN 110347380 B

(45) 授权公告日 2023. 08. 29

(21) 申请号 201910649652.0

(22) 申请日 2019.07.18

(65) 同一申请的已公布的文献号  
申请公布号 CN 110347380 A

(43) 申请公布日 2019.10.18

(73) 专利权人 中国联合网络通信集团有限公司  
地址 100033 北京市西城区金融大街21号

(72) 发明人 张洺棋 陆文成 张雪明 唐国宇  
王敏 吴宏宇 彭体伟 潘晓磊  
刘超

(74) 专利代理机构 北京同立钧成知识产权代理  
有限公司 11205  
专利代理师 朱颖 刘芳

(51) Int. Cl.  
G06F 8/20 (2018.01)  
H04L 67/60 (2022.01)

(56) 对比文件

- CN 107103016 A, 2017.08.29
- CN 107645508 A, 2018.01.30
- CN 109857375 A, 2019.06.07
- US 2017180222 A1, 2017.06.22
- CN 103502899 A, 2014.01.08
- CN 107771383 A, 2018.03.06
- US 2018321137 A1, 2018.11.08
- CN 109547430 A, 2019.03.29
- CN 106657197 A, 2017.05.10
- CN 107396199 A, 2017.11.24
- CN 103986663 B, 2018.03.27
- CN 106302747 A, 2017.01.04
- US 9432484 B1, 2016.08.30
- WO 2019134306 A1, 2019.07.11

梁新刚. 软件测试管理系统核心框架及过程的研究与实现.《中国优秀硕士学位论文全文数据库 信息科技辑》.2015, I138-296.

审查员 张帆

权利要求书3页 说明书13页 附图4页

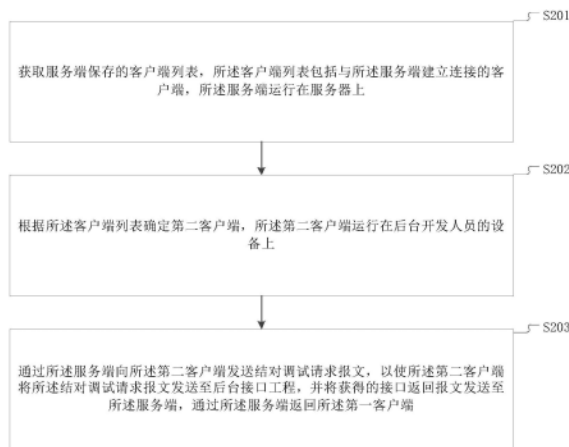
(54) 发明名称

前后端开发方法及设备

(57) 摘要

本申请实施例提供一种前后端开发方法及设备,该方法应用于第一客户端,所述第一客户端运行在前端开发人员的设备上,包括:获取服务端保存的客户端列表,所述客户端列表包括与服务端建立连接的客户端,所述服务端运行在服务器上;根据所述客户端列表确定第二客户端,第二客户端运行在后台开发人员的设备上;通过所述服务端向第二客户端发送结对调试请求报文,以使第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至服务端,通过服务端返回第一客户端,能够使前后端开发人员在代码集成测试之前随时进行结对联调,降低接口开发人员对接口开发存在分歧的情况,并能及时发现接口问题,提高前后台开发效率。

CN 110347380 B



1. 一种前后端开发方法,其特征在于,所述方法应用于第一客户端,所述第一客户端运行在前端开发人员的设备上,所述方法包括:

获取服务端保存的客户端列表,所述客户端列表包括与所述服务端建立连接的客户端,所述服务端运行在服务器上;

根据所述客户端列表确定第二客户端,所述第二客户端运行在后台开发人员的设备上;

通过所述服务端向所述第二客户端发送结对调试请求报文,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端,通过所述服务端返回所述第一客户端;

所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

所述方法还包括:

从所述服务端存储的报文中确定待检查报文;

将所述待检查报文发送至所述报文检查模块,以使所述报文检查模块根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

2. 根据权利要求1所述的方法,其特征在于,所述接口规范包括基于 json语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

3. 根据权利要求1所述的方法,其特征在于,还包括:

与所述服务端建立连接;

经过预设时间间隔发送心跳数据至所述服务端,并判断是否接收到所述服务端返回的心跳数据;

若没有接收到所述服务端返回的心跳数据,则重新执行所述与所述服务端建立连接的步骤。

4. 一种前后端开发方法,其特征在于,所述方法应用于服务端,所述服务端运行在服务器上,所述方法包括:

发送客户端列表至第一客户端,以使所述第一客户端根据所述客户端列表确定第二客户端,所述客户端列表包括与所述服务端建立连接的客户端,所述第一客户端运行在前端开发人员的设备上,所述第二客户端运行在后台开发人员的设备上;

接收所述第一客户端向所述第二客户端发送的结对调试请求报文;

将所述结对调试请求报文发送至所述第二客户端,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端;

将所述接口返回报文返回所述第一客户端;

所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

所述方法还包括:

所述报文检查模块接收所述第一客户端发送的待检查报文,所述待检查报文为所述第一客户端从所述服务端存储的报文中确定的待检查报文;

根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

5. 根据权利要求4所述的方法,其特征在于,所述接口规范包括基于json语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

6. 一种前后端开发设备,其特征在于,所述设备应用于第一客户端,所述第一客户端运行在前端开发人员的设备上,所述设备包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机执行指令,所述处理器执行所述计算机执行指令时实现如下步骤:

获取服务端保存的客户端列表,所述客户端列表包括与所述服务端建立连接的客户端,所述服务端运行在服务器上;

根据所述客户端列表确定第二客户端,所述第二客户端运行在后台开发人员的设备上;

通过所述服务端向所述第二客户端发送结对调试请求报文,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端,通过所述服务端返回所述第一客户端;

所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

所述处理器执行所述计算机执行指令时还实现如下步骤:

从所述服务端存储的报文中确定待检查报文;

将所述待检查报文发送至所述报文检查模块,以使所述报文检查模块根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

7. 一种前后端开发设备,其特征在于,所述设备应用于服务端,所述服务端运行在服务器上,所述设备包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机执行指令,所述处理器执行所述计算机执行指令时实现如下步骤:

发送客户端列表至第一客户端,以使所述第一客户端根据所述客户端列表确定第二客户端,所述客户端列表包括与所述服务端建立连接的客户端,所述第一客户端运行在前端开发人员的设备上,所述第二客户端运行在后台开发人员的设备上;

接收所述第一客户端向所述第二客户端发送的结对调试请求报文;

将所述结对调试请求报文发送至所述第二客户端,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端;

将所述接口返回报文返回所述第一客户端;

所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

所述处理器执行所述计算机执行指令时还实现如下步骤:

所述报文检查模块接收所述第一客户端发送的待检查报文,所述待检查报文为所述第一客户端从所述服务端存储的报文中确定的待检查报文;

根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,

根据所述接口规范确定所述待检查报文的有误位置。

8. 一种计算机可读存储介质,其特征在於,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如权利要求1至3任一项所述的前后端开发方法。

9. 一种计算机可读存储介质,其特征在於,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如权利要求4至5任一项所述的前后端开发方法。

## 前后端开发方法及设备

### 技术领域

[0001] 本申请实施例涉及互联网技术领域,尤其涉及一种前后端开发方法及设备。

### 背景技术

[0002] 随着社会的不断发展进步,Web页面的美学观感与交互体验对互联网用户的影响越加显著。现代浏览器流畅的渲染速度、Html5和CSS3丰富的表现力、Javascript灵活的功能,已经引起互联网用户的关注。

[0003] 目前,为了满足用户对产品体验的追求,前端开发的复杂度迅速增加,后台开发兼顾前端页面的开发模式被淘汰,演进出前后端分离的开发模式。新的开发模式使得前端与后台并行开发,最终通过restful协议接口将表现与数据结合起来。

[0004] 然而,前端与后台并行开发,可能会带来前端与后台开发工程师对接口开发不一致问题,这个问题到集成测试阶段才暴露,需要修改代码重新集成测试,开发效率低。

### 发明内容

[0005] 本申请实施例提供一种前后端开发方法及设备,以克服现有前端与后台并行开发,无法及时发现接口问题,导致开发效率低的问题。

[0006] 第一方面,本申请实施例提供一种前后端开发方法,所述方法应用于第一客户端,所述第一客户端运行在前端开发人员的设备上,所述方法包括:

[0007] 获取服务端保存的客户端列表,所述客户端列表包括与所述服务端建立连接的客户端,所述服务端运行在服务器上;

[0008] 根据所述客户端列表确定第二客户端,所述第二客户端运行在后台开发人员的设备上;

[0009] 通过所述服务端向所述第二客户端发送结对调试请求报文,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端,通过所述服务端返回所述第一客户端。

[0010] 在一种可能的设计中,所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

[0011] 所述方法还包括:

[0012] 从所述服务端存储的报文中确定待检查报文;

[0013] 将所述待检查报文发送至所述报文检查模块,以使所述报文检查模块根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

[0014] 在一种可能的设计中,所述接口规范包括基于json语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0015] 在一种可能的设计中,上述的方法,还包括:

[0016] 与所述服务端建立连接；

[0017] 经过预设时间间隔发送心跳数据至所述服务端，并判断是否接收到所述服务端返回的心跳数据；

[0018] 若没有接收到所述服务端返回的心跳数据，则重新执行所述与所述服务端建立连接的步骤。

[0019] 第二方面，本申请实施例提供另一种前后端开发方法，所述方法应用于服务端，所述服务端运行在服务器上，所述方法包括：

[0020] 发送客户端列表至第一客户端，以使所述第一客户端根据所述客户端列表确定第二客户端，所述客户端列表包括与所述服务端建立连接的客户端，所述第一客户端运行在前端开发人员的设备上，所述第二客户端运行在后台开发人员的设备上；

[0021] 接收所述第一客户端向所述第二客户端发送的结对调试请求报文；

[0022] 将所述结对调试请求报文发送至所述第二客户端，以使所述第二客户端将所述结对调试请求报文发送至后台接口工程，并将获得的接口返回报文发送至所述服务端；

[0023] 将所述接口返回报文返回所述第一客户端。

[0024] 在一种可能的设计中，所述第一客户端包括接口设计模块，所述接口设计模块包括所述前端和所述后台的接口规范，所述服务端包括报文检查模块；

[0025] 所述方法还包括：

[0026] 所述报文检查模块接收所述第一客户端发送的待检查报文，所述待检查报文为所述第一客户端从所述服务端存储的报文中确定的待检查报文；

[0027] 根据所述接口规范判断所述待检查报文是否正确，并在判定所述待检查报文有误时，根据所述接口规范确定所述待检查报文的有误位置。

[0028] 在一种可能的设计中，所述接口规范包括基于json语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0029] 第三方面，本申请实施例提供一种前后端开发设备，所述设备应用于第一客户端，所述第一客户端运行在前端开发人员的设备上，所述设备包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机执行指令，所述处理器执行所述计算机执行指令时实现如下步骤：

[0030] 获取服务端保存的客户端列表，所述客户端列表包括与所述服务端建立连接的客户端，所述服务端运行在服务器上；

[0031] 根据所述客户端列表确定第二客户端，所述第二客户端运行在后台开发人员的设备上；

[0032] 通过所述服务端向所述第二客户端发送结对调试请求报文，以使所述第二客户端将所述结对调试请求报文发送至后台接口工程，并将获得的接口返回报文发送至所述服务端，通过所述服务端返回所述第一客户端。

[0033] 在一种可能的设计中，所述第一客户端包括接口设计模块，所述接口设计模块包括所述前端和所述后台的接口规范，所述服务端包括报文检查模块；

[0034] 所述处理器执行所述计算机执行指令时还实现如下步骤：

[0035] 从所述服务端存储的报文中确定待检查报文；

[0036] 将所述待检查报文发送至所述报文检查模块，以使所述报文检查模块根据所述接

口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

[0037] 在一种可能的设计中,所述接口规范包括基于json语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0038] 在一种可能的设计中,所述处理器执行所述计算机执行指令时还实现如下步骤:

[0039] 与所述服务端建立连接;

[0040] 经过预设时间间隔发送心跳数据至所述服务端,并判断是否接收到所述服务端返回的心跳数据;

[0041] 若没有接收到所述服务端返回的心跳数据,则重新执行所述与所述服务端建立连接的步骤。

[0042] 第四方面,本申请实施例提供另一种前后端开发设备,所述设备应用于服务端,所述服务端运行在服务器上,所述设备包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机执行指令,所述处理器执行所述计算机执行指令时实现如下步骤:

[0043] 发送客户端列表至第一客户端,以使所述第一客户端根据所述客户端列表确定第二客户端,所述客户端列表包括与所述服务端建立连接的客户端,所述第一客户端运行在前端开发人员的设备上,所述第二客户端运行在后台开发人员的设备上;

[0044] 接收所述第一客户端向所述第二客户端发送的结对调试请求报文;

[0045] 将所述结对调试请求报文发送至所述第二客户端,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端;

[0046] 将所述接口返回报文返回所述第一客户端。

[0047] 在一种可能的设计中,所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

[0048] 所述处理器执行所述计算机执行指令时还实现如下步骤:

[0049] 所述报文检查模块接收所述第一客户端发送的待检查报文,所述待检查报文为所述第一客户端从所述服务端存储的报文中确定的待检查报文;

[0050] 根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

[0051] 在一种可能的设计中,所述接口规范包括基于json语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0052] 第五方面,本申请实施例提供一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如上第一方面以及第一方面各种可能的设计所述的前后端开发方法。

[0053] 第六方面,本申请实施例提供另一种计算机可读存储介质,所述计算机可读存储介质中存储有计算机执行指令,当处理器执行所述计算机执行指令时,实现如上第二方面以及第二方面各种可能的设计所述的前后端开发方法。

[0054] 本申请实施例提供的前后端开发方法及设备,该方法应用于第一客户端,第一客户端运行在前端开发人员的设备上,通过获取服务端保存的客户端列表,其中,客户端列表

包括与服务端建立连接的客户端,服务端运行在服务器上;再根据上述客户端列表确定第二客户端,第二客户端运行在后台开发人员的设备上;最后通过服务端向第二客户端发送结对调试请求报文,以使第二客户端将结对调试请求报文发送至后台接口工程,并将获得的接口返回报文通过服务端返回第一客户端,能够使前后端开发人员在代码集成测试之前随时进行结对联调,降低接口开发人员对接口开发存在分歧的情况,并能及时发现接口问题,提高前后台开发效率。

### 附图说明

[0055] 为了更清楚地说明本申请实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作一简单地介绍,显而易见地,下面描述中的附图是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0056] 图1为本申请实施例提供的前后端开发系统架构示意图;

[0057] 图2为本申请实施例提供的前后端开发方法的流程示意图一;

[0058] 图3为本申请实施例提供的前后端开发方法的流程示意图二;

[0059] 图4为本申请实施例提供的前后端开发设备的结构示意图一;

[0060] 图5为本申请实施例提供的前后端开发设备的结构示意图二;

[0061] 图6为本申请实施例提供的电子设备的结构示意图。

### 具体实施方式

[0062] 为使本申请实施例的目的、技术方案和优点更加清楚,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0063] 目前,为了满足用户对产品体验的追求,前端开发的复杂度迅速增加,后台开发兼顾前端页面的开发模式被淘汰,演进出前后端分离的开发模式。新的开发模式使得前端与后台并行开发,最终通过restful协议接口将表现与数据结合起来。然而,前端与后台并行开发,可能会带来前端与后台开发工程师对接口开发不一致问题,这个问题到集成测试阶段才暴露,需要修改代码重新集成测试,开发效率低。

[0064] 因此,考虑到上述问题,本申请提供一种前后端开发方法,该方法应用于第一客户端,第一客户端运行在前端开发人员的设备上,通过获取服务端保存的客户端列表,其中,客户端列表包括与服务端建立连接的客户端,服务端运行在服务器上;再根据上述客户端列表确定第二客户端,第二客户端运行在后台开发人员的设备上;最后通过服务端向第二客户端发送结对调试请求报文,以使第二客户端将结对调试请求报文发送至后台接口工程,并将获得的接口返回报文通过服务端返回第一客户端,能够使前后端开发人员在代码集成测试之前随时进行结对联调,降低接口开发人员对接口开发存在分歧的情况,并能及时发现接口问题,提高前后台开发效率。

[0065] 本申请提供的一种前后端开发方法,可以适用于图1所示的前后端开发系统架构示意图,如图1所示,第一客户端101运行在前端开发人员的设备上,服务端102运行在服务

器上,示例性的,第一客户端101运行在前端开发人员的本地机器上,服务端102运行在远程服务器上,服务端102通过内部局域网与第一客户端101建立数据连接。第一客户端101与前端工程连接,其中,前端工程表示正运行在前端本地机器上的前端工程。第一客户端101可以获取服务端102保存的客户端列表,所述客户端列表包括与所述服务端102建立连接的客户端。第一客户端101可以根据上述客户端列表确定第二客户端103,其中,第二客户端103运行在后台开发人员的设备上。示例性的,第二客户端103运行在后台开发人员的本地机器上,服务端102通过内部局域网与第二客户端103建立数据连接。第二客户端103与后台接口工程连接,其中,后台接口工程表示正运行在后台本地机器上的restful接口工程。第一客户端101还可以通过所述服务端102向所述第二客户端103发送结对调试请求报文,以使所述第二客户端103将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端102,通过所述服务端102返回所述第一客户端101。

[0066] 这里,客户端(Client)也可称为用户端,是指与服务器相对应,为客户提供本地服务的程序。服务端是为客户端服务的,服务的内容诸如向客户端提供资源,保存客户端数据等。Restful是一种软件架构风格、设计风格,而不是标准,只是提供了一组设计原则和约束条件,其主要用于客户端和服务端交互类的软件,基于这个风格设计的软件可以更简洁,更有层次,更易于实现缓存等机制。

[0067] 下面以具体地实施例对本申请的技术方案以及本申请的技术方案如何解决上述技术问题进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例中不再赘述。下面将结合附图,对本申请的实施例进行描述。

[0068] 图2为本申请实施例提供的前后端开发方法的流程示意图一,本申请实施例的执行主体可以为图1所示实施例中的第一客户端。如图2所示,该方法可以包括:

[0069] S201:获取服务端保存的客户端列表,所述客户端列表包括与所述服务端建立连接的客户端,所述服务端运行在服务器上。

[0070] 可选地,上述的方法,还包括:

[0071] 与所述服务端建立连接;

[0072] 经过预设时间间隔发送心跳数据至所述服务端,并判断是否接收到所述服务端返回的心跳数据;

[0073] 若没有接收到所述服务端返回的心跳数据,则重新执行所述与所述服务端建立连接的步骤。

[0074] 示例性的,在前端开发人员本地机器M1上启动第一客户端C1,启动时可以提示相关人员输入一个启动信息,例如客户端表示,C1与服务端S建立连接Connect1,C1经过预设时间间隔,例如每间隔1秒,与服务端S交换一次心跳数据,心跳数据包括启动时输入的启动信息,如果C1没有收到服务端S返回的心跳数据,C1立即重新建立与S的连接,如果无法建立连接,再次重新建立,如果连续多数,例如3次,都无法建立连接,C1提示“无法连接服务器”。

[0075] 服务端S可以保存所有已经与其建立Socket连接的客户端的IP地址、启动信息等,例如保存的是C1的IP地址与启动信息。客户端C1通过之前建立的连接向服务端S查询其他客户端的列表,然后显示在C1的界面上。

[0076] 客户端C1可以经过预设时间间隔,例如每5分钟,向服务端S查询上述列表,然后保存到客户端的内存中,在本实施例中上述列表包括“/rest/query”接口。

[0077] S202:根据所述客户端列表确定第二客户端,所述第二客户端运行在后台开发人员的设备上。

[0078] 第一客户端根据上述客户端列表确定第二客户端,示例性的,第一客户端显示上述客户端列表以供相关人员选择,并在接收到选择指令后,该选择指令携带客户端相关信息,例如客户端标识,根据上述选择指令确定第二客户端,发起结对调试。

[0079] 这里,第二客户端需要与服务端建立连接,示例性的,在后台开发人员本地机器M2上启动装置的第二客户端C2,启动时必须输入一个启动信息;C2与服务端S建立连接Connect2,C2经过预设时间间隔,例如每间隔1秒,与服务端S交换一次心跳数据,心跳数据包括启动时输入的启动信息,如果C2没有收到服务端S返回的心跳数据,C2立即重新建立与S的连接,如果无法建立连接,再次重新建立,如果连续多次,例如连续3次,都无法建立连接,C2提示“无法连接服务器”。

[0080] 同上,服务端S也可以C2的IP地址与启动信息。客户端C2通过之前建立的连接向服务端S查询其他客户端的列表,然后显示在C2的界面上。客户端C2也可以经过预设时间间隔,例如每5分钟,向服务端S查询上述列表,然后保存到客户端的内存中,在本实施例中上述列表包括“/rest/query”接口。

[0081] S203:通过所述服务端向所述第二客户端发送结对调试请求报文,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端,通过所述服务端返回所述第一客户端。

[0082] 这里,前端开发人员在客户端C1的界面上选择客户端C2发起结对调试,指令通过连接Connect1发送给S,然后又通过连接Connect2中转给C2,C2的界面上提示结对调试请求。如果C2拒绝请求,本次结对调试立即结束。如果C2同意请求,C2可以提示输入本地机器M2上运行的后台接口工程,例如restful接口的端口号,C2将端口号保存在内存中。示例性的,前端开发人员可以在本地机器M1上运行前端工程,在页面上操作功能按钮,按钮触发请求:http://localhost:8080/rest/query,客户端C1拦截本地机器M1发出的所有请求,截取端口号以后的部分与内存中保存的列表进行匹配:如果匹配不成功,C1将请求释放给M1的操作系统处理。如果匹配成功,例如与C2匹配成功,C1将请求发送给服务端S,服务端S将请求存储到数据库中,然后中转给C2。C2将请求发送给M2上运行的restful接口,获得的接口返回报文发送给服务端S,服务端S将返回结果存储到数据库中,然后中转给C1,C1将返回结果发送给页面进行展示。

[0083] 前端开发人员可以通过第一客户端调用后台接口工程,在第一客户端记录下每次调用的请求报文与返回报文的全部内容、时间,然后把这些数据传送给服务端保存下来。

[0084] 可选地,所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

[0085] 所述方法还包括:

[0086] 从所述服务端存储的报文中确定待检查报文;

[0087] 将所述待检查报文发送至所述报文检查模块,以使所述报文检查模块根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

[0088] 可选地,所述接口规范包括基于json语法描述的所述前端和所述后台的接口的所

有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0089] 这里,第一客户端和第二客户端均可以包括上述接口设计模块,上述接口设计模块包括基于json语法描述接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系、集合节点数量约束等。其可以作为前后端开发人员共同遵守的接口规范,也可以用于报文检查模块判断报文的正确性。

[0090] 示例性的,一个接口设计模块样例如下所示。

```

{
  "RestName": "商品查询接口",
  "RestUrl": "/shop/queryShopInfo",
  "RestDesc": "根据店铺ID 查询店铺的名称",
  "RestReqFields": [
    {
      "FieldName": "shopId",
      "FieldDesc": "店铺ID",
      "FieldPath": "$.shopId",
      "FieldRequired": true,
      "FieldType": "String",
      "FieldSize": 8,
      "FieldEnum": null,
      "FieldRelations": null
    }
  ],
  "RestRspFields": [
    {
      "FieldDesc": "店铺信息",
      "FieldPath": "$.shopInfo",
      "FieldRequired": true,
      "FieldType": "Object",
      "FieldSize": null,
      "FieldEnum": null,
      "FieldRelations": null
    },
    {
      "FieldDesc": "店铺名称",
      "FieldPath": "$.shopInfo.shopName",
      "FieldRequired": true,
      "FieldType": "string",
      "FieldSize": 64,
      "FieldEnum": null,
      "FieldRelations": null
    },
    {
      "FieldDesc": "店铺类型",
      "FieldPath": "$.shopInfo.shopType",
      "FieldRequired": true,
      "FieldType": "string",
      "FieldSize": 8,
      "FieldEnum": ["suning","jd"],
    }
  ]
}

```

[0091]

```

        "FieldRelations": [
            {
                "RelatedField": "$.shopInfo.shopName",
                "RelatedType": "coexist"
            }
        ]
    }
}
]
}

```

[0093] 其中,描述接口的对象的第一层有5个键值对:

[0094] (1)RestName键对应的值是接口名称;

[0095] (2)RestUrl键对应的值是接口地址后缀;

[0096] (3)RestDesc键对应的值是接口的功能表述;

[0097] (4)RestReqFields键对应的值是接口请求报文所有字段的描述;

[0098] (5)RestRspFields键对应的值是接口返回报文所有字段的描述。

[0099] RestReqFields、RestRspFields键的值都是一个数组,数组中包含若干子对象,每个子对象有7个键值对:

[0100] (1)FieldDesc键对应的值是字段意义的说明;

[0101] (2)FieldPath键对应的值是字段的路径表达式,例如\$.shopInfo.shopName表示shopName节点在shopInfo节点下级,shopInfo节点在接口报文的根节点下级;

[0102] (3)FieldRequired键对应的值是字段是否必填,true表示必填,false表示非必填;

[0103] (4)FieldType键对应的值是字段类型,例如字符串、整数;

[0104] (5)FieldSize键对应的值是字段长度,例如64;

[0105] (6)FieldEnum键对应的值是字段可选值数组,例如["suning","jd"];

[0106] (7)FieldRelations键对应的值是字段的与其他字段的关联约束;

[0107] FieldRelations的值是数组,数组中包含若干子对象,每个子对象有2个键值对:

[0108] (1)RelatedField键的对应值是关联字段的路径表达式;

[0109] (2)RelatedType键的对应值是关联约束类型,可选值:coexist表示RelatedField键值所指的字段与本字段必须同时不为空,或者同时为空,size\_size表示RelatedField键值所指的字段是数组类型,本字段的值也是数组类型,并且两个字段的数组长度相同;count\_size表示RelatedField键值所指的字段是一个整数类型,本字段的值是数组类型,并且前者的数值就是后者的数组长度。

[0110] 在上述结对调试的过程中,如果前端开发人员对接口的报文有异议时,可以通过第一客户端在服务端存储的调用历史中找到当时的报文记录,提交给报文检查模块检查其正确与否并给出解释。

[0111] 示例性的,以请求报文为例,第一客户端从服务端存储的报文中确定待检查报文,读取报文内容,提取报文中所有字段的路径,再从上述接口设计模块的RestReqFields节点提取所有字段的路径,如果后者不完全包含前者,记录下未包含的字段在待检查报文中的路径。

[0112] 遍历接口设计模块的RestReqFields节点的所有子对象,提取FieldRequired键值

等于true的接口字段的路径,如果待检查报文中该字段为空或不存在,记录下该字段在报文中的路径。

[0113] 提取待检查报文的每个字段,再找到接口设计模块中RestReqFields节点描述该字段的子对象,根据FieldType键值检查该字的类型,根据FieldSize键值检查该字段的长度,根据FieldEnum键值检查该字的值是否在枚举范围内。如果存在检查未通过的项目,记录字段在待检查报文中的路径、不满足的检查类型。

[0114] 提取接口设计模块中RestReqFields节点下级的每个FieldRelations数组节点,遍历每个FieldRelations数组节点的所有子对象,提取子对象的RelatedField键值,以下称为A值,提取子对象的RelatedType键值,以下称为C值,提取子对象所在FieldRelations数组的同级FieldName键值,以下称为B值。根据A值找到所指字段在待检查报文中的字段值,根据B值找到所指字段在待检查报文中的字段值。得到两个字段值后,再根据C值,按照规则检查A、B值的关系:若C值等于coexist,A值与B值必须同时不为空,或者同时为空;若C值等于size, A值是数组类型,B值也是数组类型,并且数组长度相同;若C值等于countsize,A值是一个整数类型,B值是数组类型,并且前者的数值就是后者的数组长度。如果有检查不通过的项目,记录A值得字段路径、B值的字段路径、C值。

[0115] 收集以上步骤中记录的所有未通过检查的项目,作为输出结果显示。

[0116] 以上是对请求报文的检查步骤,如果是响应报文的检查,将RestReqFields节点换成RestRspFields。

[0117] 另外,在上述结对调试的过程中,如果后台开发人员对接口的报文有异议时,可以通过第二客户端在服务端存储的调用历史中找到当时的报文记录,提交给报文检查模块检查其正确与否并给出解释。具体步骤可以参照上述对请求报文的检查步骤。

[0118] 从上述描述可知,本申请实施例使用接口设计模块描述接口设计,前后端开发人员参照接口设计模块进行开发工作,在代码集成测试之前随时进行远程结对联调,联调过程产生的接口报文备份到服务器中,如果前后端开发人员就接口报文产生争议,报文检查模块进行报文正确性检查,并指出有误的具体位置,由此确认哪方对接口理解有误,降低接口设计人员人工处理开发人员分歧的情况,提高前后台分离开发模式的工作效率。

[0119] 图3为本申请实施例提供的前后端开发方法的流程示意图二,本申请实施例的执行主体可以为图1所示实施例中的服务端。如图3所示,该方法包括:

[0120] S301:发送客户端列表至第一客户端,以使所述第一客户端根据所述客户端列表确定第二客户端,所述客户端列表包括与所述服务端建立连接的客户端,所述第一客户端运行在前端开发人员的设备上,所述第二客户端运行在后台开发人员的设备上。

[0121] S302:接收所述第一客户端向所述第二客户端发送的结对调试请求报文。

[0122] S303:将所述结对调试请求报文发送至所述第二客户端,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端。

[0123] S304:将所述接口返回报文返回所述第一客户端。

[0124] 可选地,所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块;

[0125] 所述方法还包括:

[0126] 所述报文检查模块接收所述第一客户端发送的待检查报文,所述待检查报文为所述第一客户端从所述服务端存储的报文中确定的待检查报文;

[0127] 根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

[0128] 这里,上述报文检查模块还可以接收第二客户端发送的待检查报文,该待检查报文为第二客户端从服务端存储的报文中确定的待检查报文。根据上述接口规范判断上述待检查报文是否正确,并在判定上述待检查报文有误时,根据上述接口规范确定上述待检查报文的有误位置。

[0129] 可选地,所述接口规范包括基于 json 语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0130] 本申请实施例提供的前后端开发方法,应用于服务端,所述服务端运行在服务器上,通过发送客户端列表至第一客户端,以使第一客户端根据客户端列表确定第二客户端,其中,客户端列表包括与服务端建立连接的客户端,第一客户端运行在前端开发人员的设备上,第二客户端运行在后台开发人员的设备上;接收第一客户端向第二客户端发送的结对调试请求报文;将结对调试请求报文发送至第二客户端,以使第二客户端将结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至服务端;将接口返回报文返回第一客户端,能够使前后端开发人员在代码集成测试之前随时进行结对联调,降低接口开发人员对接口开发存在分歧的情况,并能及时发现接口问题,提高前后台开发效率。

[0131] 对应于上文实施例的前后端开发方法,图4为本申请实施例提供的前后端开发设备的结构示意图。为了便于说明,仅示出了与本申请实施例相关的部分。图4为本申请实施例提供的前后端开发设备的结构示意图一。如图4所示,该前后端开发设备40应用于第一客户端,所述第一客户端运行在前端开发人员的设备上,所述设备包括:列表获取模块401、客户端确定模块402以及结对调试模块403。

[0132] 其中,列表获取模块401,用于获取服务端保存的客户端列表,所述客户端列表包括与所述服务端建立连接的客户端,所述服务端运行在服务器上。

[0133] 客户端确定模块402,用于根据所述客户端列表确定第二客户端,所述第二客户端运行在后台开发人员的设备上。

[0134] 结对调试模块403,用于通过所述服务端向所述第二客户端发送结对调试请求报文,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端,通过所述服务端返回所述第一客户端。

[0135] 可选地,所述第一客户端包括接口设计模块404,所述接口设计模块404包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块。

[0136] 所述设备还包括报文确定模块405和报文检查模块406。

[0137] 其中,报文确定模块405,用于从所述服务端存储的报文中确定待检查报文。

[0138] 报文检查模块406,用于将所述待检查报文发送至所述报文检查模块,以使所述报文检查模块根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

[0139] 可选地,所述接口规范包括基于 json 语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0140] 可选地,所述设备还包括连接建立模块407和数据发送模块408。

[0141] 其中,连接建立模块407,用于与所述服务端建立连接。

[0142] 数据发送模块408,用于经过预设时间间隔发送心跳数据至所述服务端,并判断是否接收到所述服务端返回的心跳数据;

[0143] 所述连接建立模块407,还用于若没有接收到所述服务端返回的心跳数据,则重新执行所述与所述服务端建立连接的步骤。

[0144] 本申请实施例提供的设备,可用于执行上述图2方法实施例的技术方案,其实现原理和技术效果类似,本申请实施例此处不再赘述。

[0145] 图5为本申请实施例提供的前后端开发设备的结构示意图二。如图5所示,该前后端开发设备50应用于服务端,所述服务端运行在服务器上,所述设备包括:列表发送模块501、报文接收模块502、报文发送模块503和报文返回模块504。

[0146] 其中,列表发送模块501,用于发送客户端列表至第一客户端,以使所述第一客户端根据所述客户端列表确定第二客户端,所述客户端列表包括与所述服务端建立连接的客户端,所述第一客户端运行在前端开发人员的设备上,所述第二客户端运行在后台开发人员的设备上。

[0147] 报文接收模块502,用于接收所述第一客户端向所述第二客户端发送的结对调试请求报文。

[0148] 报文发送模块503,用于将所述结对调试请求报文发送至所述第二客户端,以使所述第二客户端将所述结对调试请求报文发送至后台接口工程,并将获得的接口返回报文发送至所述服务端。

[0149] 报文返回模块504,用于将所述接口返回报文返回所述第一客户端。

[0150] 可选地,所述第一客户端包括接口设计模块,所述接口设计模块包括所述前端和所述后台的接口规范,所述服务端包括报文检查模块505。

[0151] 所述报文检查模块505接收所述第一客户端发送的待检查报文,所述待检查报文为所述第一客户端从所述服务端存储的报文中确定的待检查报文;

[0152] 根据所述接口规范判断所述待检查报文是否正确,并在判定所述待检查报文有误时,根据所述接口规范确定所述待检查报文的有误位置。

[0153] 可选地,所述接口规范包括基于json语法描述的所述前端和所述后台的接口的所有节点名称、类型、长度限制、是否必填、枚举值、节点依赖关系和集合节点数量约束。

[0154] 本申请实施例提供的设备,可用于执行上述图3方法实施例的技术方案,其实现原理和技术效果类似,本申请实施例此处不再赘述。

[0155] 参考图6,其示出了适于用来实现本申请实施例的电子设备600的结构示意图,该电子设备600可以为终端设备或服务器。其中,终端设备可以包括但不限于诸如移动电话、笔记本电脑、数字广播接收器、个人数字助理(Personal Digital Assistant,简称PDA)、平板电脑(Portable Android Device,简称PAD)、便携式多媒体播放器(Portable Media Player,简称PMP)、车载终端(例如车载导航终端)等等的移动终端以及诸如数字TV、台式计算机等等的固定终端。图6示出的电子设备仅仅是一个示例,不应对本申请实施例的功能和使用范围带来任何限制。

[0156] 如图6所示,电子设备600可以包括处理装置(例如中央处理器、图形处理器等)

601,其可以根据存储在只读存储器(Read Only Memory,简称ROM)602中的程序或者从存储装置608加载到随机访问存储器(Random Access Memory,简称RAM)603中的程序而执行各种适当的动作和处理。在RAM 603中,还存储有电子设备600操作所需的各种程序和数据。处理装置601、ROM 602以及RAM 603通过总线604彼此相连。输入/输出(I/O)接口605也连接至总线604。

[0157] 通常,以下装置可以连接至I/O接口605:包括例如触摸屏、触摸板、键盘、鼠标、摄像头、麦克风、加速度计、陀螺仪等的输入装置606;包括例如液晶显示器(Liquid Crystal Display,简称LCD)、扬声器、振动器等的输出装置607;包括例如磁带、硬盘等的存储装置608;以及通信装置609。通信装置609可以允许电子设备600与其他设备进行无线或有线通信以交换数据。虽然图6示出了具有各种装置的电子设备600,但是应理解的是,并不要求实施或具备所有示出的装置。可以替代地实施或具备更多或更少的装置。

[0158] 特别地,根据本申请的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本申请的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信装置609从网络上被下载和安装,或者从存储装置608被安装,或者从ROM602被安装。在该计算机程序被处理装置601执行时,执行本申请实施例的方法中限定的上述功能。

[0159] 需要说明的是,本申请上述的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本申请中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本申请中,计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读信号介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:电线、光缆、RF(射频)等等,或者上述的任意合适的组合。

[0160] 上述计算机可读介质可以是上述电子设备中所包含的;也可以是单独存在,而未装配入该电子设备中。

[0161] 上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被该电子设备执行时,使得该电子设备执行上述实施例所示的方法。

[0162] 可以以一种或多种程序设计语言或其组合来编写用于执行本申请的操作的计算机程序代码,上述程序设计语言包括面向对象的程序设计语言——诸如Java、Smalltalk、C++,还包括常规的过程式程序设计语言——诸如“C”语言或类似的程序设计语言。程序代码可

以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括局域网(Local Area Network,简称LAN)或广域网(Wide Area Network,简称WAN)——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0163] 附图中的流程图和框图,图示了按照本申请各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,该模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0164] 描述于本申请实施例中所涉及到的单元可以通过软件的方式实现,也可以通过硬件的方式来实现。

[0165] 最后应说明的是:以上各实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述各实施例对本申请进行了详细的说明,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申请各实施例技术方案的范围。

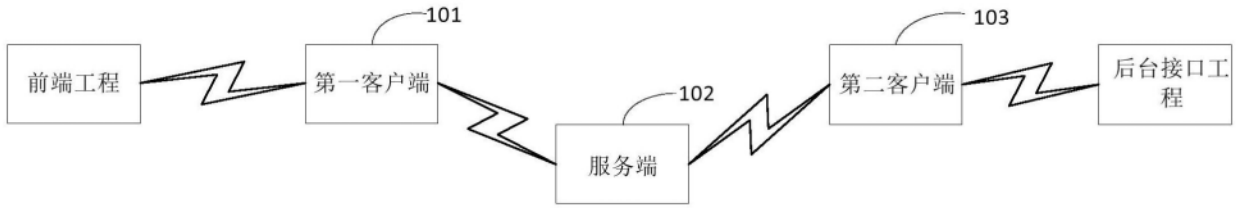


图1

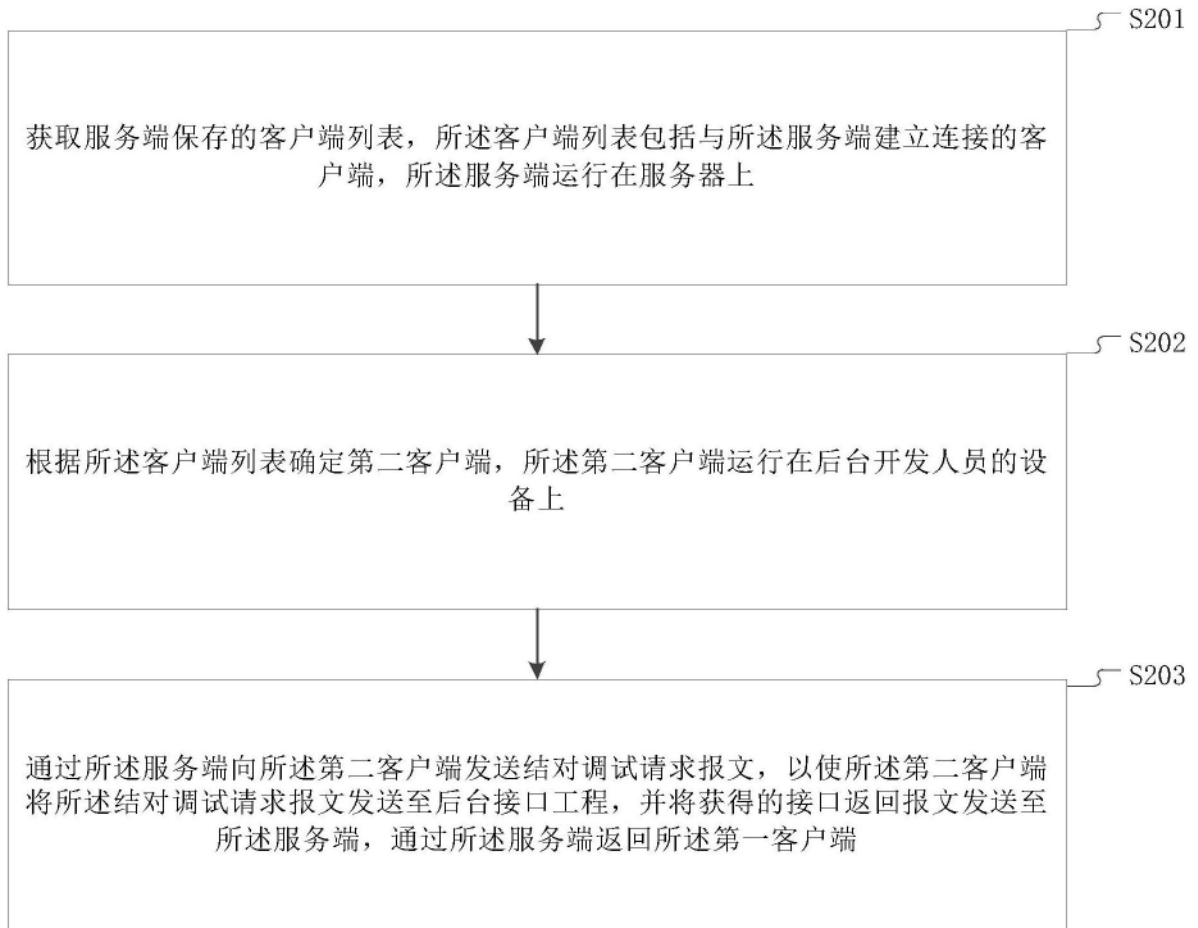


图2

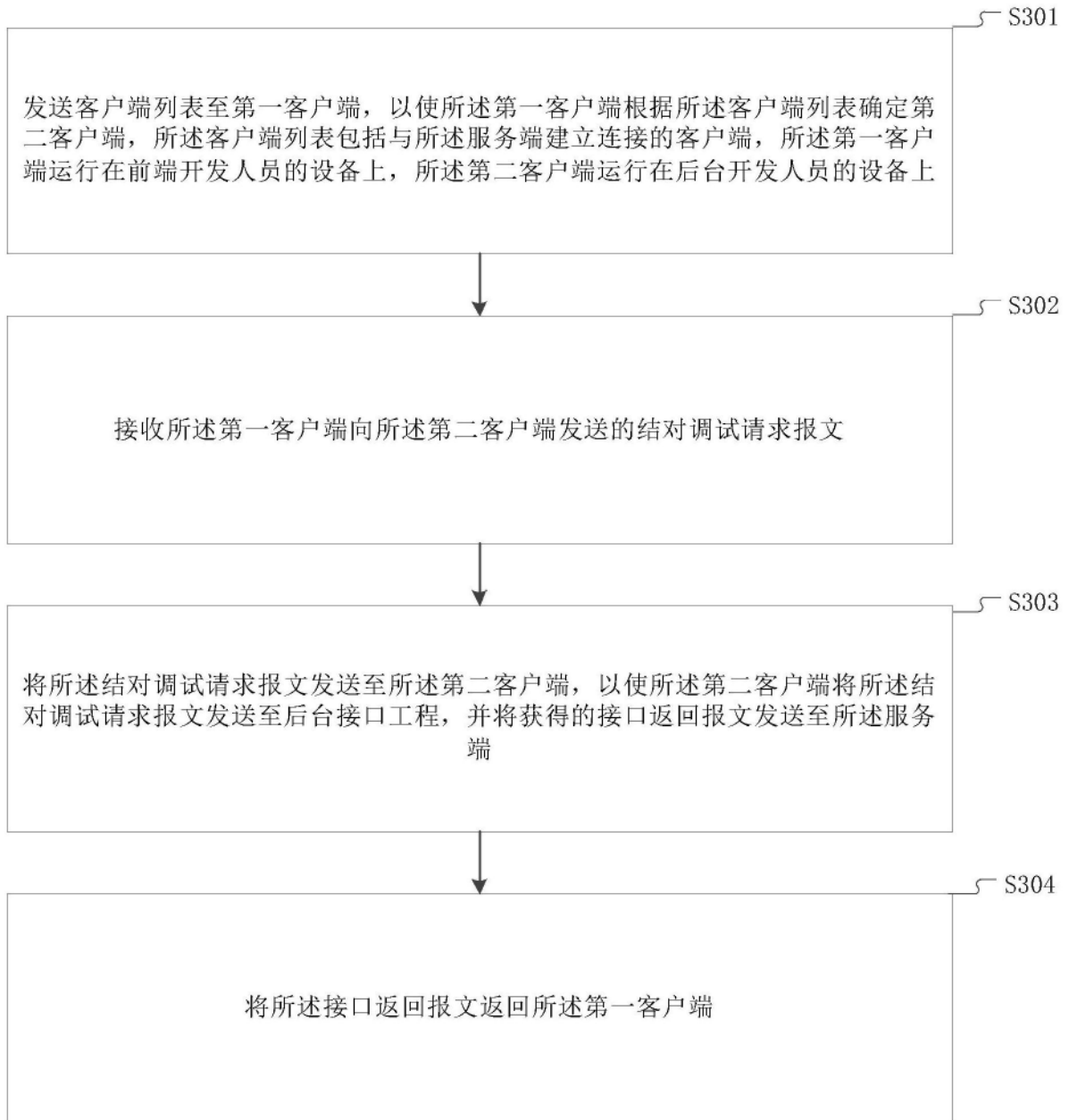


图3

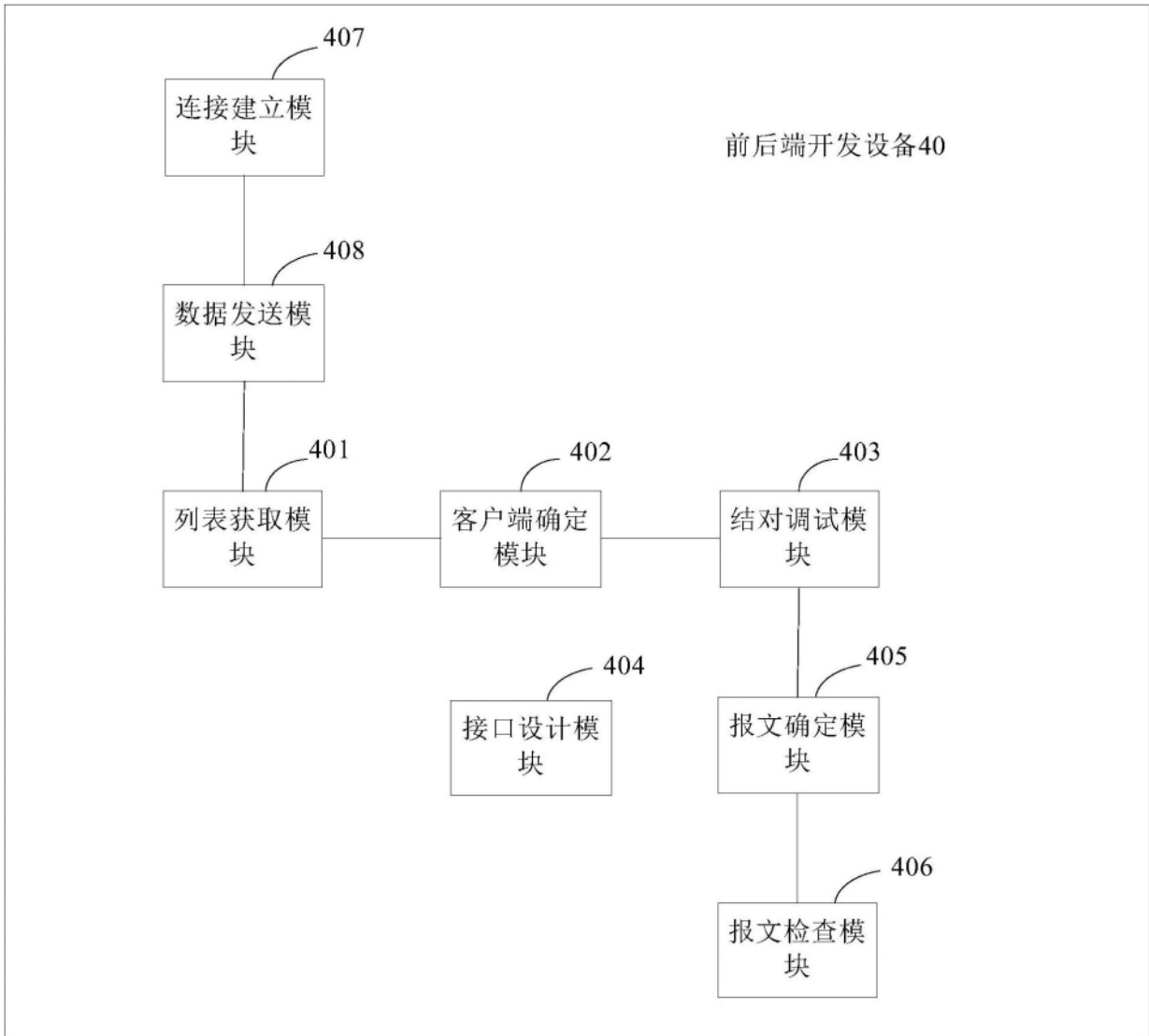


图4

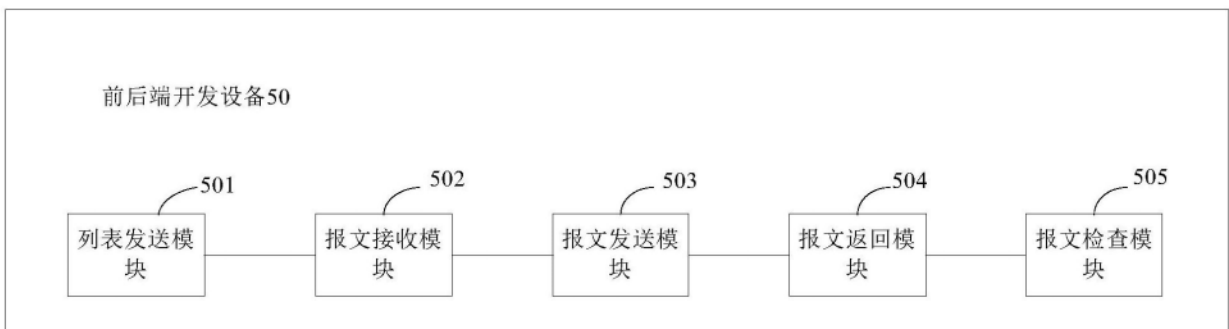


图5

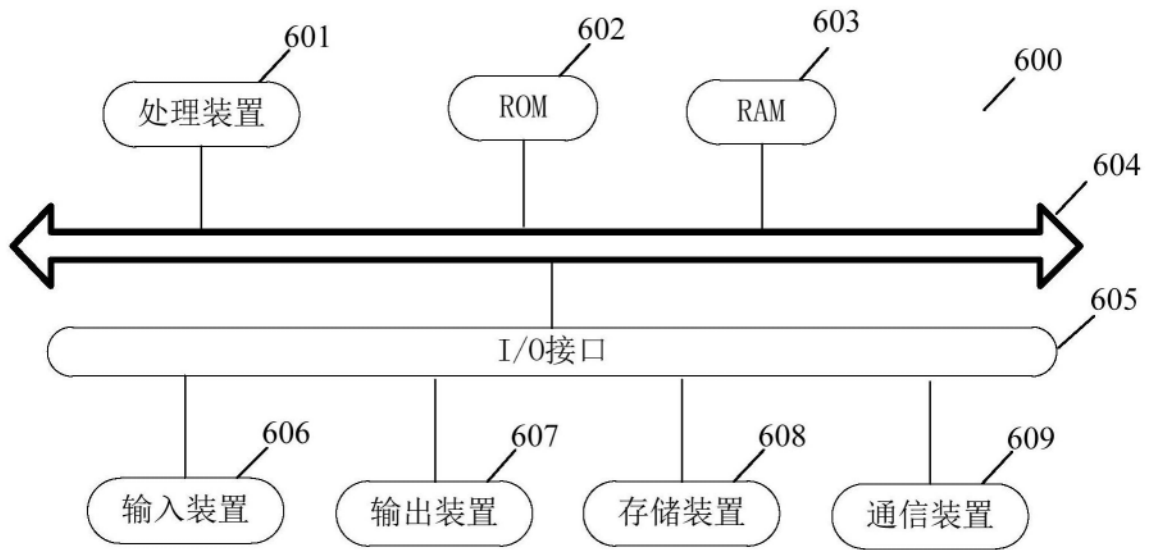


图6