



(12) 发明专利

(10) 授权公告号 CN 109901938 B

(45) 授权公告日 2021. 11. 19

(21) 申请号 201910141734.4

(22) 申请日 2019.02.26

(65) 同一申请的已公布的文献号
申请公布号 CN 109901938 A

(43) 申请公布日 2019.06.18

(73) 专利权人 北京华夏电通科技股份有限公司
地址 100094 北京市海淀区丰秀中路3号院
6号楼5层101

(72) 发明人 张琪 刘燕飞 阚海北 单衍景

(74) 专利代理机构 北京三友知识产权代理有限公司 11127

代理人 杨冬英 王涛

(51) Int. Cl.

G06F 9/54 (2006.01)

G06F 16/955 (2019.01)

(56) 对比文件

CN 105635259 A, 2016.06.01

CN 105635259 A, 2016.06.01

CN 106527885 A, 2017.03.22

CN 107992279 A, 2018.05.04

CN 109189596 A, 2019.01.11

CN 103414785 A, 2013.11.27

CN 108762482 A, 2018.11.06

CN 108683936 A, 2018.10.19

US 2016260319 A1, 2016.09.08

审查员 宋秀之

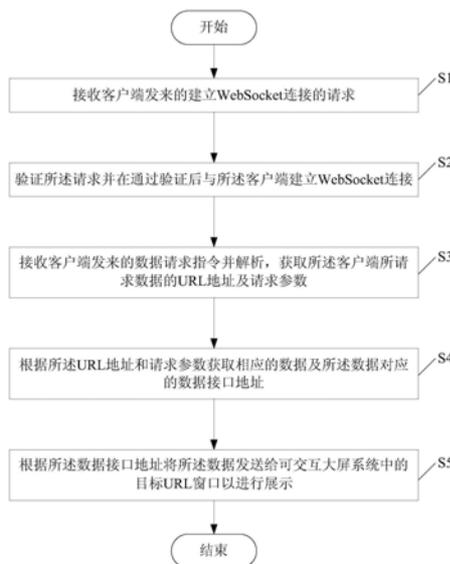
权利要求书1页 说明书7页 附图2页

(54) 发明名称

基于WebSocket通信的可交互大屏系统及可视化展示方法

(57) 摘要

本发明提供了基于WebSocket通信的可交互大屏系统及可视化展示方法,所述方法包括:接收客户端发来的建立WebSocket连接请求;验证所述请求并在通过验证后与所述客户端建立WebSocket连接;接收客户端发来的数据请求指令并解析,获取所述客户端所请求数据的URL地址及请求参数;根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址;根据所述数据接口地址将所述数据发送给可交互大屏系统中的目标URL窗口以进行展示。本发明通过客户端向可交互大屏系统发送需求指令,可交互大屏系统接收到指令后对其进行解析,最后将获取的目标数据传送到目标URL窗口,实现了数据的实时交互。



1. 一种基于WebSocket通信的可交互大屏系统可视化展示方法,其特征在于,所述方法包括:

可交互大屏系统服务器接收客户端发来的建立WebSocket连接请求;

可交互大屏系统服务器验证所述请求并在通过验证后与所述客户端建立WebSocket连接;

可交互大屏系统服务器接收客户端发来的数据请求指令并解析,获取所述客户端所请求数据的URL地址及请求参数;

可交互大屏系统服务器根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址;其中,所述数据包括多个相关数据;所述数据接口地址为可交互大屏系统中用于展示所述多个相关数据的URL窗口的地址;

可交互大屏系统服务器根据所述数据接口地址将所述多个相关数据发送给所述URL窗口以进行展示;或者

可交互大屏系统服务器根据展示需求,在可交互大屏系统的屏幕上创建URL窗口,并为所创建的URL窗口配置数据接口地址。

2. 根据权利要求1所述的基于WebSocket通信的可交互大屏系统可视化展示方法,其特征在于,所述客户端包括可交互大屏系统的URL窗口。

3. 根据权利要求1所述的基于WebSocket通信的可交互大屏系统可视化展示方法,其特征在于,所述客户端包括与可交互大屏系统通信的外部系统。

4. 根据权利要求1所述的基于WebSocket通信的可交互大屏系统可视化展示方法,其特征在于,所述方法还包括:所述URL窗口接收所述数据并更新当前展示页面。

5. 一种基于WebSocket通信的可交互大屏系统,其特征在于,可应用于可交互大屏系统服务器上,所述基于WebSocket通信的可交互大屏系统包括:

WebSocket交互单元,用于接收客户端发来的建立WebSocket连接请求,验证所述请求并在通过验证后与所述客户端建立WebSocket连接;

指令解析单元,用于接收客户端发来的数据请求指令并解析,获取所述客户端所请求数据的URL地址及请求参数;

数据获取单元,用于根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址;所述数据接口地址为可交互大屏系统中用于展示所述数据的URL窗口的地址;

数据传输单元,用于根据所述数据接口地址将所述数据发送给所述URL窗口以进行展示;

配置单元,用于根据展示需求在可交互大屏系统的屏幕上创建URL窗口,并为所创建的URL窗口配置数据接口地址。

6. 根据权利要求5所述的基于WebSocket通信的可交互大屏系统,其特征在于,所述客户端包括可交互大屏系统的URL窗口。

7. 根据权利要求5所述的基于WebSocket通信的可交互大屏系统,其特征在于,所述客户端包括与可交互大屏系统通信的外部系统。

8. 根据权利要求5所述的基于WebSocket通信的可交互大屏系统,其特征在于,所述URL窗口接收所述数据并更新当前展示页面。

基于WebSocket通信的可交互大屏系统及可视化展示方法

技术领域

[0001] 本发明涉及通信技术领域,尤其涉及一种基于WebSocket通信的可交互大屏系统及可视化展示方法。

背景技术

[0002] 现有的大屏可视化系统的缺点是:1) 大屏可视化系统作为一个独立的整体,PC端系统与大屏展示数据实时交互性较差,对大屏展示界面操控效果不理想;2) 大屏展示页面局部窗口进行跳转(如进入下一级页面),难以保证相关的其他窗口随着实时联动。因此,如何保证大屏可视化系统内部及其与其他系统实现实时、定时数据交互是目前迫切需要解决的问题。

发明内容

[0003] 本发明的技术任务是提供一种基于WebSocket通信的可交互大屏系统及可视化展示方法,来解决大屏可视化系统内部及其与其他系统实现实时、定时数据交互的问题,提高系统的可用性,灵活性。

[0004] 为更好的实现上述目标,本发明实施例一方面提供了一种基于WebSocket通信的可交互大屏系统可视化展示方法,所述方法包括:

[0005] 接收客户端发来的建立WebSocket连接请求;

[0006] 验证所述请求并在通过验证后与所述客户端建立WebSocket连接;

[0007] 接收客户端发来的数据请求指令并解析,获取所述客户端所请求数据的URL地址及请求参数;

[0008] 根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址;

[0009] 根据所述数据接口地址将所述数据发送给可交互大屏系统中的目标URL窗口以进行展示。

[0010] 在一实施例中,所述方法还包括:

[0011] 根据展示需求,在可交互大屏系统的屏幕上创建URL窗口,并为所创建的URL窗口配置数据接口地址。

[0012] 在一实施例中,所述客户端包括可交互大屏系统的URL窗口。

[0013] 在一实施例中,所述客户端包括与可交互大屏系统通信的外部系统。

[0014] 在一实施例中,所述方法还包括:所述目标URL窗口接收所述数据并更新当前展示页面。

[0015] 另一方面,本发明实施例还提供了一种基于WebSocket通信的可交互大屏系统,所述系统包括:

[0016] WebSocket交互单元,用于接收客户端发来的建立WebSocket连接请求,验证所述请求并在通过验证后与所述客户端建立WebSocket连接;

[0017] 指令解析单元,用于接收客户端发来的数据请求指令并解析,获取所述客户端所

请求数据的URL地址及请求参数；

[0018] 数据获取单元,用于根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址；

[0019] 数据传输单元,用于根据所述数据接口地址将所述数据发送给可交互大屏系统中的目标URL窗口以进行展示。

[0020] 在一实施例中,所述系统还包括：

[0021] 配置单元,用于根据展示需求在可交互大屏系统的屏幕上创建URL窗口,并为所创建的URL窗口配置数据接口地址。

[0022] 在一实施例中,所述客户端包括可交互大屏系统的URL窗口。

[0023] 在一实施例中,所述客户端包括与可交互大屏系统通信的外部系统。

[0024] 在一实施例中,所述目标URL窗口接收所述数据并更新当前展示页面。

[0025] 本发明通过客户端向可交互大屏系统发送需求指令,可交互大屏系统接收到指令后对其进行解析,最后将获取的目标数据传送到目标URL窗口,实现了数据的实时交互。

附图说明

[0026] 为了更清楚地说明本发明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。

[0027] 图1为本发明提供的基于WebSocket通信的可交互大屏系统可视化展示方法的流程示意图；

[0028] 图2为本发明提供的基于WebSocket通信的可交互大屏系统的结构示意图。

具体实施方式

[0029] 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,所描述的实施例仅仅是本发明一部分实施例,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。

[0030] 图1为本发明实施例提供的一种基于WebSocket通信的可交互大屏系统可视化展示方法的流程示意图。如图1所示,上述方法主要包括以下步骤：

[0031] 步骤S1、接收客户端发来的建立WebSocket连接请求。

[0032] 本方法的实施主体为可交互大屏系统的服务器,客户端要想与可交互大屏系统通信,需先向交互大屏系统的服务器发起建立WebSocket连接请求,二者建立WebSocket连接后,客户端便可与可交互大屏系统进行通信。

[0033] 步骤S2、验证所述请求并在通过验证后与所述客户端建立WebSocket连接。

[0034] 步骤S3、接收客户端发来的数据请求指令并解析,获取所述客户端所请求数据的URL地址及请求参数。

[0035] 步骤S4、根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址。

[0036] 为方便展示起见,通常为数据指定访问地址及数据接口地址,访问地址即为URL地址,数据接口地址即为可交互大屏系统中用于展示该数据的相应的URL窗口的地址。

[0037] 步骤S5、根据所述数据接口地址将所述数据发送给可交互大屏系统中的目标URL窗口以进行展示。

[0038] 本发明实施例通过客户端向可交互大屏系统发送需求指令,可交互大屏系统接收到指令后对其进行解析,最后将获取的目标数据传送到目标URL窗口,实现了数据的实时交互。

[0039] 在一实施例中,在利用图1中所示方法进行可交互大屏系统可视化展示之前,通常会根据展示需求,在可交互大屏系统的屏幕上创建URL窗口,并为所创建的URL窗口配置数据接口地址。如此,可实现大屏展示页面个性化定制,根据展示的内容调整窗口的大小,实现灵活创建URL窗口的目的。

[0040] 在一实施例中,在将所述数据发送给可交互大屏系统中的目标URL窗口后,目标URL窗口会根据所接收的数据更新当前展示页面。

[0041] 本方法的实施主体为可交互大屏系统的服务器,首先需要在可交互大屏系统的服务器端注册一个STOMP端点,例如“/endpointDisplay”。不同的Websocket连接定义的地址不一样,具体实施时视实际情况而定。服务器端配置代码如表1所示:

[0042] 表1服务器端代码参考示例

```
[0043] @Override
public void registerStompEndpoints(StompEndpointRegistry registry) {
    registry.addEndpoint("/endpointDisplay").addInterceptors().withSockJS();
}
```

[0044] 如客户端要想同服务器连接,则需要先连接该端点,即“/endpointDisplay”,那么客户端需要进行如表2所示配置:

[0045] 表2客户端代码参考示例

```
[0046] var stompClient = null;
var socket = new SockJS('/endpointDisplay');
stompClient = Stomp.over(socket);
```

[0047] 在进行如上设置后,客户端便可服务器连接通信了。

[0048] 服务器端定义接收和推送消息的前缀,该前缀可以随意指定。例如,定义“/topic”为服务器端给客户端发送消息的地址前缀,定义“/push”为客户端向服务器发送数据时要加的前缀,具体配置见表3。

[0049] 表3配置消息前缀代码参考示例

```
[0050] @Override
public void configureMessageBroker(MessageBrokerRegistry registry) {
    registry.enableSimpleBroker("/topic");
    registry.setApplicationDestinationPrefixes("/push");
    //registry.setUserDestinationPrefix("/user/");
}
```

[0051] 表4为客户端向服务器发送消息的一示例。Send之后的括号内部为/push打头的地址,queryMap为参数。

[0052] 表4配置消息前缀代码参考示例

```
function sendMessage() {
  var queryMap = $("input[name='radio']:checked").val();
  stompClient.send("/push/notice/sendMessage", {}, queryMap);
  stompClient.send("/push/notice/wahahahahaha", {}, queryMap);
}
```

[0054] 因为有"/push",所以服务器端会认为是客户端发送给自己的消息,然后解析客户端发来的数据请求指令后得到的URL地址"/notice/sendMessage"。然后服务器端会调用参数为queryMap、地址为"/notice/sendMessage"对应的程序,进行数据请求处理,获取最终请求的数据结果。

[0055] 服务器给对应的目标URL窗口发送消息("/topic/acceptanceClose"为目标URL窗口的地址,resultMap为得到的目标数据),服务器往外发消息必须以"/topic"为前缀(因上文中定义了该前缀,所以必须一致)。服务器发送给目标URL窗口的消息见表5所示。

[0056] 表5服务器发向目标URL窗口发送消息参考示例

```
simpMessagingTemplate.convertAndSend("/topic/acceptanceClose",resultMap);
```

[0058] 目标URL窗口会订阅一个地址,若和服务器发送的地址一致,就会接收消息,如不一致则拒收。表6给出了指定目标URL窗口接收消息的路径,定义回调函数的示例。

[0059] 表6服务器发向目标URL窗口发送消息参考示例

```
stompClient.connect({}, function(frame) {
  stompClient.subscribe('/topic/twelveMonthCaseClose', function(msg) {
  });
});
```

[0061] 其中,发起数据请求指令的客户端可以为可交互大屏系统的URL窗口,也可以为与可交互大屏系统通信的外部系统,例如PC端。

[0062] 如果发起数据请求指令的客户端为可交互大屏系统内部的URL窗口,在进行WebSocket通信时,URL窗口根据WebSocket协议给可交互大屏系统的服务器端发送相应的数据请求指令,服务器解析该数据请求指令,并根据指令要求对数据进行相应处理,再将处理后获得的数据发送给目标URL窗口,目标URL窗口接收到数据后,重新加载、刷新页面。例如,其中一个URL窗口展示的内容为河北省的地图,其他URL窗口展示的是河北省的相关数据。点击地图上的一个市如石家庄市,那么其他URL窗口展示的数据也会变成石家庄市的数据,点击其他市,其他URL窗口的数据还会随着变化。

[0063] 利用本发明实施例提供的技术方案,外部系统也可以与可交互大屏系统进行数据交互,此时发起数据请求指令的客户端为其他的与当前可交互大屏系统建立WebSocket的外部系统。外部系统根据WebSocket协议给当前可交互大屏系统的服务器发送相应的数据请求指令,服务器解析指令,并根据指令要求从其他系统中获取、处理所需要的数据,再将处理后的数据发送给可交互大屏系统中的目标URL窗口进行反馈,目标URL窗口接收到数据后,重新加载、刷新页面。

[0064] 例如,利用本发明可在电脑端或智能电视端通过互联网或内部网络对可交互大屏系统进行操作达到远程操控大屏页面随其联动的目的。具体实施时,在为可交互大屏系统

创建URL窗口并配置数据接口地址后,通过电脑端或智能电视端系统向大屏可视化系统的服务器发送请求建立WebSocket连接。可交互大屏系统的服务器验证电脑端或智能电视端系统发送的请求信息,并返回验证后的消息,验证通过后确立建立WebSocket连接。电脑端或智能电视端系统可随时向大屏可视化系统的服务端发送请求信息,如触发式请求(点击某按钮实现大屏页面展示数据随指令要求变化)、定时请求(设置定时推送消息给大屏任意URL窗口)。可交互大屏系统的服务器根据电脑端或智能电视端系统的请求进行相应处理,并通知相关大屏窗口执行指令,刷新页面,完成数据展示。

[0065] 通过本发明提供的基于WebSocket通信的可交互大屏可视化展示方法,可保证大屏可视化系统内部及其与其他系统实现实时、定时数据交互,系统具有较高的灵活性、可用性、兼容性。

[0066] 在一实施例中,Ventuz软件由于其功能强大、流程简单,而且很容易集成到复杂的硬件和软件系统中,非常适合应用于大屏可视化大数据展示领域。因此,本发明的基于WebSocket通信的可交互大屏可视化展示方法可基于Ventuz软件实现。在利用该软件实现可视化展示时,在为所创建的URL窗口配置数据接口地址时,通常在Excel中间文档中配置编写好的数据接口地址,然后由Ventuz软件的解析器解析所配置的各数据接口,然后按照图1所示步骤即可实现可交互大屏系统的可视化展示。URL窗口与其调用的数据接口通过Excel中间文档实现分离部署。

[0067] 为更清楚的对本发明的技术方案进行说明,下面给出一依托Ventuz软件实现本发明技术方案的示例,主要包括以下步骤:

[0068] (1) 根据需要展示的数据项进行页面设计,确定数据的展示形式,在可交互大屏系统中创建满足需求的URL窗口。

[0069] (2) 按照创建的URL窗口,配置数据接口地址,生成Excel中间文档。

[0070] (3) Ventuz软件解析器解析所配置的各数据接口地址。

[0071] (4) 客户端与可交互大屏系统的服务器建立WebSocket连接后,客户端基于WebSocket协议向服务器发起数据请求指令。

[0072] (5) 可交互大屏系统的服务器根据客户端的请求进行相应处理,并通知涉及到的其他URL窗口进行相应的变化。具体实施时,服务器解析客户端发来的数据请求指令,获取客户端所请求数据的URL地址及请求参数,根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址,按照数据接口地址将所获取的数据加载到可交互大屏系统中的目标URL窗口以进行展示。

[0073] 重复步骤(4)和(5),可以进行URL窗口页面的刷新。

[0074] 基于先进的3D展示技术Ventuz,本发明实施例创新性地将数据模块、信息模块和应用模块进行整合,支持自定义编辑组合。可有效地传达思想概念,实现艺术、数据、业务逻辑功能齐头并进,通过直观地传达关键的方面与特征,从而实现对于相当稀疏而又复杂的数据集的深入洞察。

[0075] 基于与图1所示的基于WebSocket通信的可交互大屏系统可视化展示方法相同的发明构思,本申请实施例还提供了一种基于WebSocket通信的可交互大屏系统可视化展示方法,如下面实施例所述。由于该系统解决问题的原理与图1中基于WebSocket通信的可交互大屏系统可视化展示方法相似,因此该系统的实施可以参见图1中方法的实施,重复之处

不再赘述。

[0076] 在另一实施例中,本发明还提供了一种基于WebSocket通信的可交互大屏系统,其结构如图2所示,该系统主要包括:WebSocket交互单元10、指令解析单元20、数据获取单元30和数据传输单元40。

[0077] 其中,WebSocket交互单元10用于接收客户端发来的建立WebSocket连接请求,验证所述请求并在通过验证后与所述客户端建立WebSocket连接;指令解析单元20用于接收客户端发来的数据请求指令并解析,获取所述客户端所请求数据的URL地址及请求参数;数据获取单元30用于根据所述URL地址和请求参数获取相应的数据及所述数据对应的数据接口地址;数据传输单元40用于根据所述数据接口地址将所述数据发送给可交互大屏系统中的目标URL窗口以进行展示。

[0078] 在一实施例中,上述系统还包括一配置单元50,用于根据展示需求在可交互大屏系统的屏幕上创建URL窗口,并为所创建的URL窗口配置数据接口地址。

[0079] 在一实施例中,所述客户端包括可交互大屏系统的URL窗口。

[0080] 在一实施例中,所述客户端包括与可交互大屏系统通信的外部系统。

[0081] 在一实施例中,所述目标URL窗口接收所述数据并更新当前展示页面。

[0082] 本发明实施例通过客户端向可交互大屏系统发送需求指令,可交互大屏系统接收到指令后对其进行解析,最后将获取的目标数据传送到目标URL窗口,实现了数据的实时交互。

[0083] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0084] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0085] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0086] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0087] 本发明中应用了具体实施例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,

依据本发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

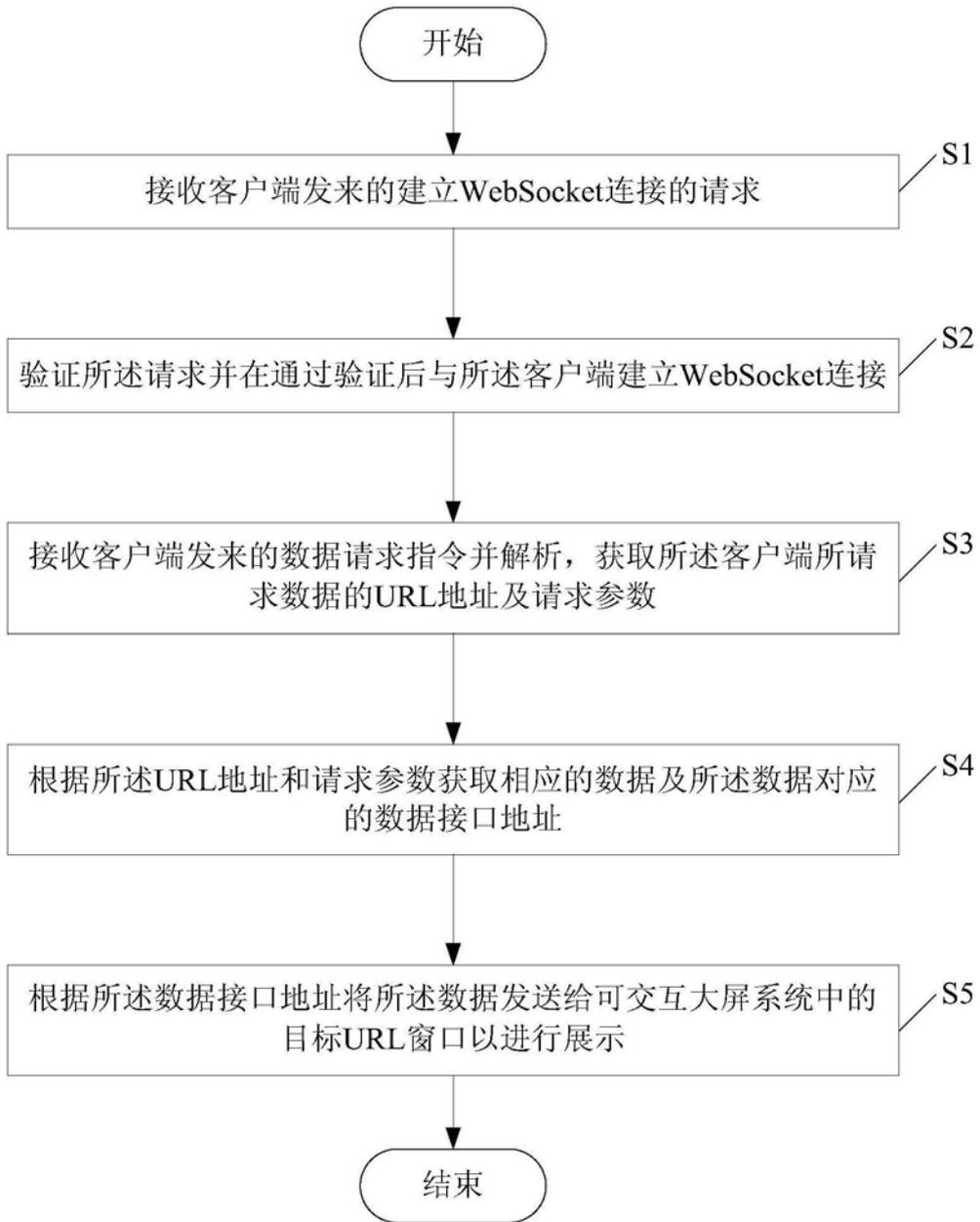


图1

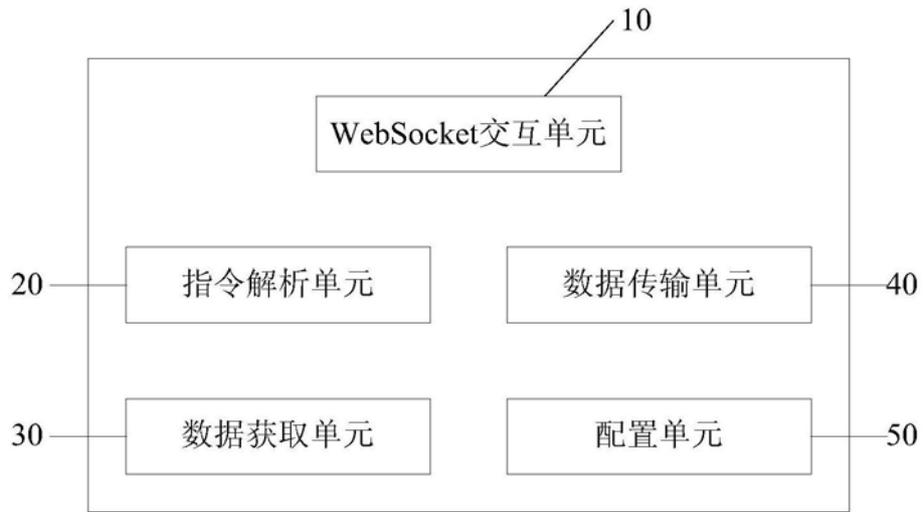


图2