



(12) 发明专利申请

(10) 申请公布号 CN 101814084 A

(43) 申请公布日 2010.08.25

(21) 申请号 201010033882.3

(22) 申请日 2010.01.11

(71) 申请人 北京世纪高通科技有限公司

地址 100088 北京市海淀区知春路 6 号锦秋
国际大厦 B 座 1601

(72) 发明人 付升荣 郑小华 曹家齐 薛明

(74) 专利代理机构 北京中博世达专利商标代理
有限公司 11274

代理人 申健

(51) Int. Cl.

G06F 17/30(2006.01)

G06F 9/44(2006.01)

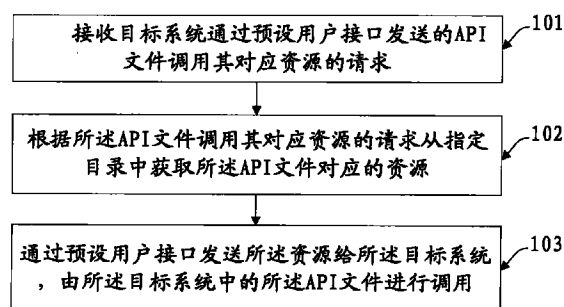
权利要求书 2 页 说明书 6 页 附图 4 页

(54) 发明名称

API 管理方法及系统

(57) 摘要

本发明公开一种 API 管理方法及系统,涉及地理信息系统领域,解决了现有技术中所述 API 文件与采用所述 API 文件开发的系统之间耦合过紧的问题。API 服务端统一保存 API 文件和其对应的资源,该方法包括:接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求,所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统;根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源;通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。本发明实施例主要应用于地理信息系统中。



1. 一种 API 管理方法,其特征在于,API 服务端统一保存 API 文件和其对应的资源,该方法包括:

接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求,所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统;

根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源;

通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。

2. 根据权利要求 1 所述的 API 管理方法,其特征在于,所述目标系统为用户端采用所述 API 文件开发得到的系统,所述方法还包括:

接收所述用户端通过所述预设用户接口发送的所需 API 类型和版本号;

从所述指定目录中获取所述 API 类型和版本号对应的 API 文件;

通过所述预设用户接口发送所述获取到的 API 文件给所述用户端。

3. 根据权利要求 2 所述的 API 管理方法,其特征在于,还包括:

接收所述用户端的注册信息;

在所述用户端注册完成后,通过所述预设用户接口发送许可 Key 给所述用户端;

在从所述指定目录中获取所述 API 类型和版本号对应的 API 文件之前,接收所述用户端通过所述预设用户接口发送的所述许可 Key。

4. 根据权利要求 1 所述的 API 管理方法,其特征在于,还包括:

依次接收 API 开发端发布的每个 API 文件和其对应的资源;

在所述指定目录中的类型子目录下为所述接收到的每个 API 文件创建对应的目标版本目录;

将所述接收到的每个 API 文件和其对应的资源依次保存到各自对应的所述目标版本目录下。

5. 一种 API 管理系统,其特征在于,包括:

目标系统,采用从 API 服务端获取的 API 文件开发得到的,用于通过预设用户接口向所述 API 服务端发送所述 API 文件调用其对应资源的请求;

API 服务端,用于统一保存 API 文件和其对应的资源,并根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源,通过预设用户接口发送所述资源给所述目标系统;

所述目标系统还用于接收所述 API 服务端返回的所述 API 文件对应的资源,由所述 API 文件进行调用。

6. 根据权利要求 5 所述的 API 管理系统,其特征在于,还包括:

用户端,用于通过所述预设用户接口向 API 服务端发送所需 API 类型和版本号;

所述 API 服务端还用于从指定目录中获取所述 API 类型和版本号对应的 API 文件,通过所述预设用户接口发送所述获取到的 API 文件给所述用户端;

所述用户端还用于利用所述 API 文件开发得到所述目标系统。

7. 根据权利要求 5 所述的 API 管理系统,其特征在于,所述用户端还用于在所述 API 服务端进行注册;

所述 API 服务端还用于在所述用户端注册完成后,通过所述预设用户接口发送许可 Key 给所述用户端;

所述用户端还用于通过所述预设用户接口向所述 API 服务端发送所述许可 Key。

8. 根据权利要求 5 所述的 API 管理系统,其特征在于,还包括:

API 开发端,用于开发 API 文件,并将确定版本号的 API 文件和其对应的资源发布到 API 服务端;

所述 API 服务端还用于在指定目录中的类型子目录下为所述发布的每个 API 文件创建对应的目标版本目录,将所述发布的每个 API 文件和其对应的资源保存到各自对应的所述目标版本目录下。

9. 一种 API 服务端,其特征在于,API 服务端统一保存 API 文件和其对应的资源,所述 API 服务端包括:第一接收单元,用于接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求,所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统;

第一获取单元,用于根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源;

第一发送单元,用于通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。

10. 根据权利要求 9 所述的 API 服务端,其特征在于,所述目标系统为用户端采用所述 API 文件开发得到的系统,所述 API 服务端还包括:

第二接收单元,用于接收所述用户端通过所述预设用户接口发送的所需 API 类型和版本号;

第二获取单元,用于从所述指定目录中获取所述 API 类型和版本号对应的 API 文件;

第二发送单元,用于通过所述预设用户接口发送所述获取到的 API 文件给所述用户端。

API 管理方法及系统

技术领域

[0001] 本发明涉及地理信息系统领域,尤其涉及一种 API 管理方法及系统。

背景技术

[0002] 目前,基于互联网的地图搜索服务随着 Ajax 技术的提出,已经从性能上得到很大的提升,不仅在速度上还是用户体验上,它都已经超越了传统的方式。但是开发一个完整的地图搜索服务系统,需要在人力和技术上投入很高的成本,同时由于其专业性,使很多企业望其项背。这样的情况下,地图 API 应运而生。

[0003] 地图 API(Application Programming Interface,应用程序编程接口)通过编程语言把各种搜索服务、地图显示、地图操作工具以控件的形式提供给用户,这些控件通过不同的指定接口名称来调用。这样就把系统的复杂性隐藏在控件下面,而用户只需要的面对一个简单的 API,这样就大大加快了开发的效率和难度。

[0004] 然而发明人发现现有技术中在使用地图 API 进行二次开发时,地图 API 的提供方都是直接将该地图 API 文件提供给用户,用户接收到所述地图 API 文件后,将其写入进行开发的系统中,然后用户配置相关信息来获取 API 提供的各种功能,同时由于用户开发后的系统运行时需要调用所述 API 对应的资源,因而上述地图 API 的提供方也需要把地图 API 所使用的资源如图片、控制页面显示的样式等一起提供给用户,导致所述 API 文件与用户开发的系统耦合过紧。

发明内容

[0005] 本发明的实施例提供一种 API 管理方法及系统,在用户使用 API 进行二次开发时,实现 API 文件与用户采用所述 API 文件开发的系统解耦。

[0006] 为达到上述目的,本发明的实施例采用如下技术方案:

[0007] 一种 API 管理方法,API 服务端统一保存 API 文件和其对应的资源,该方法包括:

[0008] 接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求,所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统;

[0009] 根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源;

[0010] 通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。

[0011] 一种 API 管理系统,包括:

[0012] 目标系统,采用从 API 服务端获取的 API 文件开发得到的,用于通过预设用户接口向所述 API 服务端发送所述 API 文件调用其对应资源的请求;

[0013] API 服务端,用于统一保存 API 文件和其对应的资源,并根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源,通过预设用户接口发送所述资源给所述目标系统;

[0014] 所述目标系统还用于接收所述 API 服务端返回的所述 API 文件对应的资源,由所述 API 文件进行调用。

[0015] 一种 API 服务端,API 服务端统一保存 API 文件和其对应的资源,所述 API 服务端包括:第一接收单元,用于接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求,所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统;

[0016] 第一获取单元,用于根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源;

[0017] 第一发送单元,用于通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。

[0018] 本发明实施例提供的 API 管理方法,由 API 服务端统一保存 API 文件和其对应的资源,采用从所述 API 服务端获取的 API 文件开发得到目标系统,通过接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求,并根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源,然后通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。本技术方案在实现时通过所述目标系统采用的 API 文件从所述 API 服务端获得,而所述目标系统运行时,所述 API 文件也是从所述 API 服务端中调用对应的资源,因而解决了现有技术中由于直接将所述 API 文件对应的资源给使用所述 API 文件进行目标系统开发的用户,导致所述 API 文件与采用所述 API 文件开发的系统之间耦合过紧的问题。同时,由于本方案无需将所述 API 文件对应的资源直接给使用 API 文件的开发者,也就避免了现有技术中每次有新版本的 API 时,就需要重新将所述新版本的 API 文件及其对应资源传给 API 文件的使用者,从而导致对 API 文件进行管理时不便捷的问题。

附图说明

[0019] 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。

[0020] 图 1 为实施例 1 中 API 管理方法的流程图;

[0021] 图 2 为实施例 1 中 API 管理系统的结构图;

[0022] 图 3 为实施例 1 中 API 服务端的结构框图;

[0023] 图 4 为实施例 2 中 API 管理方法的流程图;

[0024] 图 5 为实施例 2 中 API 管理系统的结构图;

[0025] 图 6 为实施例 2 中 API 服务端的结构框图。

具体实施方式

[0026] 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,都属于本发明保护的范围。

[0027] 实施例 1：

[0028] 本发明实施例提供一种 API 管理方法，其中 API 服务端统一保存 API 文件和其对应的资源，如图 1 所示，该方法包括如下步骤：

[0029] 101、接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求，所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统。

[0030] 由所述 API 服务端对所述 API 文件及其对应的资源进行统一管理。与现有技术直接所述 API 文件及其对应的资源给使用者相比，在有 API 文件的版本更新时，便于对所述 API 文件及其对应的资源进行管理。

[0031] 102、根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源。所述资源如 API 文件需要调用的图片或控制页面显示的样式，在所述目标系统运行时调用该资源来保证该目标系统可以正确的运行。所述 API 文件及其对应的资源都保存在所述 API 服务端的指定目录中。所述 API 文件调用其对应资源的请求中包含了所述 API 文件对应的类型和版本号，通过所述 API 文件对应的类型和版本号就可以从指定目录中获取所述 API 文件对应的资源。

[0032] 103、通过预设用户接口发送所述资源给所述目标系统，由所述目标系统中的所述 API 文件进行调用。在所述目标系统运行时，由所述 API 文件通过预设用户接口从 API 服务端调用其对应的资源，解决了现有技术中由于直接将所述 API 文件对应的资源给使用所述 API 文件进行目标系统开发的人员，导致所述 API 文件与采用所述 API 文件开发的系统之间耦合过紧的问题。

[0033] 为了实现上述方法，本实施例还提供一种 API 管理系统，如图 2 所示，所述 API 管理系统包括：目标系统 21 和 API 服务端 22。

[0034] 目标系统 21 采用从 API 服务端获取的 API 文件开发得到的，用于通过预设用户接口向所述 API 服务端发送所述 API 文件调用其对应资源的请求。所述 API 文件及其对应的资源都保存在所述 API 服务端的指定目录中。API 服务端 22 用于统一保存 API 文件和其对应的资源，并根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源，通过预设用户接口发送所述资源给所述目标系统。

[0035] 所述目标系统 21 还用于接收所述 API 服务端返回的所述 API 文件对应的资源，由所述 API 文件进行调用。由于所述目标系统采用的 API 文件从所述 API 服务端获得，而所述目标系统运行时，所述 API 文件也是从所述 API 服务端中调用对应的资源，解决了现有技术中由于直接将所述 API 文件对应的资源给使用所述 API 文件进行目标系统开发的用户，导致所述 API 文件与采用所述 API 文件开发的系统之间耦合过紧的问题。

[0036] 本发明实施例还提供一种 API 服务端，API 服务端统一保存 API 文件和其对应的资源，如图 3 所示，所述 API 服务端包括：第一接收单元 31、第一获取单元 32 和第一发送单元 33。

[0037] 第一接收单元 31 用于接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求，所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统。所述预设用户接口是提供给所述 API 文件的使用者和采用所述 API 文件开发得到的目标系统进行访问 API 服务端的唯一接口。第一获取单元 32 用于根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源。

[0038] 第一发送单元 33 用于通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。由于无需将所述 API 文件对应的资源也给使用所述 API 文件进行目标系统开发的人员,实现了 API 文件与用户采用所述 API 文件开发的系统解耦。

[0039] 实施例 2:

[0040] 本发明实施例详细介绍一种 API 管理方法,如图 4 所示,该方法包括如下步骤:

[0041] 401、在用户端从 API 服务端调用所需 API 文件进行二次开发之前,在 API 服务端提供的注册页面中进行用户端的信息注册。

[0042] 所述 API 服务端接收所述用户端的注册信息,并在所述用户端注册完成后,通过所述预设用户接口发送许可 Key 给所述用户端。每个所述许可 Key 唯一对应一个用户端。用户端接收到所述许可 Key 后将其保存,并在从所述 API 服务端调用所需 API 文件前发送给所述 API 服务端。

[0043] 402、所述用户端通过所述预设用户接口发送的所述许可 Key、所需 API 类型和版本号给所述 API 服务端。所述 API 类型可以为 JavaScript 或者 Flex,根据开发所述 API 文件时采用的编程语言而定。所述 API 版本号可以为 1.0、1.02 等,根据所述 API 文件版本的更新情况,由所述 API 文件的开发者确定。

[0044] 403、所述 API 服务端接收所述用户端通过所述预设用户接口发送的所述许可 Key、所需 API 类型和版本号。然后从所述指定目录中获取所述 API 类型和版本号对应的 API 文件并通过所述预设用户接口发送所述获取到的 API 文件给所述用户端。

[0045] 所述用户端接收到所述 API 文件后,利用所述 API 文件进行开发得到目标系统。由于所述 API 服务端对所述 API 文件及其对应的资源进行统一管理。与现有技术直接所述 API 文件及其对应的资源给使用者相比,在有 API 文件的版本更新时,便于对所述 API 文件及其对应的资源进行管理。

[0046] 404、所述目标系统在运行时,所述 API 文件调用其对应的资源之前,所述 API 服务端接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求。

[0047] 405、所述 API 服务端根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源。所述 API 文件及其对应的资源都保存在所述 API 服务端的指定目录中。所述 API 文件调用其对应资源的请求中包含了所述 API 文件对应的类型和版本号,通过所述 API 文件对应的类型和版本号就可以从指定目录中获取所述 API 文件对应的资源。

[0048] 406、所述 API 服务端通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。在所述目标系统运行时,由所述 API 文件通过预设用户接口从 API 服务端调用其对应的资源,解决了现有技术中由于直接将所述 API 文件对应的资源给使用所述 API 文件进行目标系统开发的人员,导致所述 API 文件与采用所述 API 文件开发的系统之间耦合过紧的问题。

[0049] 在用户使用所述目标系统进行查询时,所述 API 服务端还提供一种查询接口,所述目标系统中 API 文件可以通过该查询接口调用地图引擎接口进行相关查询操作。如:用户在使用所述目标系统时希望查询北京饭店相关的信息,则当用户在所述目标系统上输入北京饭店进行查询时,所述目标系统中 API 文件会通过预设用户接口访问所述 API 服务端,

然后通过调用所述 API 服务端中上述查询接口访问所述地图引擎,当从该地图引擎上获取到北京饭店的相关信息后,返回给用户。该查询过程属于 API 文件内部调用过程,对于用户来说,该过程是透明的,也就是用户利用所述 API 文件进行二次开发时,不需要关心所述查询过程。

[0050] 另外,本发明实施例中的所述 API 文件可以通过下述开发步骤获得:首先 API 开发端的开发小组在 SVN 服务端建立 API 开发的工程,然后在小组内共享并协同开发,开发过程中不断进行 API 的更新。所述 API 工程基本开发完成后,由测试人员从 SVN 服务端获取,测试人员从 SVN 服务端获取到最后版本的 API 文件时并与开发人员迭代进行测试,测试通过后确定所述 API 文件的版本号,并把所述 API 文件和其所使用的资源发布到所述 API 服务端。每个版本的 API 文件都提供一个相应的 API 文档,为用户在使用所述 API 文件进行二次开发时提供参考。然后进行下一版本的开发与测试,如此循环。

[0051] 所述 API 服务端可以采用步骤 407 至步骤 409(图未示)管理所述 API 文件及其对应的资源:

[0052] 407、依次接收 API 开发端发布的每个 API 文件和其对应的资源。

[0053] 408、在所述指定目录中的类型子目录下为所述接收到的每个 API 文件创建对应的目标版本目录。所述指定目录的结构可以采用但不限于下述目录结构:{server}/{type}/{version}/{API 文件及相关资源};其中,serve 表示该系统中的指定目录;type 为 API 的类型目标,表示所述 API 文件对应的类型;version 为版本号目录,所述 API 文件及相关资源存放于所述 API 文件对应的类型目录下的版本子目录中。

[0054] 409、将所述接收到的每个 API 文件和其对应的资源依次保存到各自对应的所述目标版本目录下。

[0055] 本实施例还提供一种 API 管理系统,如图 5 所示,所述 API 管理系统包括:目标系统 51、API 服务端 52、用户端 53。

[0056] 所述用户端 53 用于在所述 API 服务端进行注册;所述 API 服务端 52 用于在所述用户端 53 注册完成后,通过所述预设用户接口发送许可 Key 给所述用户端 53。所述用户端 53 还用于通过所述预设用户接口向所述 API 服务端发送所述许可 Key、所需 API 类型和版本号。

[0057] 然后,所述 API 服务端 52 还用于从指定目录中获取所述 API 类型和版本号对应的 API 文件,通过所述预设用户接口发送所述获取到的 API 文件给所述用户端 53。所述用户端 53 还用于利用所述 API 文件开发得到所述目标系统 51。

[0058] 目标系统 51 采用从 API 服务端获取的 API 文件开发得到的,用于通过预设用户接口向所述 API 服务端发送所述 API 文件调用其对应资源的请求。所述 API 文件及其对应的资源都保存在所述 API 服务端 52 的指定目录中。API 服务端 52 还用于统一保存 API 文件和其对应的资源,并根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源,通过预设用户接口发送所述资源给所述目标系统 51。

[0059] 所述目标系统 51 还用于接收所述 API 服务端返回的所述 API 文件对应的资源,由所述 API 文件进行调用。由于所述目标系统采用的 API 文件从所述 API 服务端获得,而所述目标系统运行时,所述 API 文件也是从所述 API 服务端中调用对应的资源,解决了现有技术中由于直接将所述 API 文件对应的资源给使用所述 API 文件进行目标系统开发的用户,

导致所述 API 文件与采用所述 API 文件开发的系统之间耦合过紧的问题。同时也方便对所述 API 文件在有版本更新时进行管理。

[0060] 所述 API 管理系统,可选的,还可以包括:API 开发端 54,用于开发 API 文件,并将确定版本号 API 文件和其对应的资源发布到 API 服务端 52。所述 API 服务端 52 还用于在指定目录中的类型子目录下为所述发布的每个 API 文件创建对应的目标版本目录,将所述发布的每个 API 文件和其对应的资源保存到各自对应的所述目标版本目录下。

[0061] 本发明实施例还提供一种 API 服务端,如图 6 所示,所述 API 服务端包括:第一接收单元 61、第一获取单元 62、第一发送单元 63、第二接收单元 64、第二获取单元 65、第二发送单元 66。

[0062] 其中,第二接收单元 64 用于接收所述用户端通过所述预设用户接口发送的所需 API 类型和版本号。第二获取单元 65 用于从所述指定目录中获取所述 API 类型和版本号对应的 API 文件。第二发送单元 66 用于通过所述预设用户接口发送所述获取到的 API 文件给所述用户端。所述用户端获取到所述 API 文件后,采用所述 API 文件进行二次开发得到目标系统。

[0063] 然后,所述目标系统运行时,第一接收单元 61 用于接收目标系统通过预设用户接口发送的 API 文件调用其对应资源的请求,所述目标系统为采用从所述 API 服务端获取的 API 文件开发得到的系统。所述预设用户接口是提供给所述 API 文件的使用者和采用所述 API 文件开发得到的目标系统进行访问 API 服务端的唯一接口。第一获取单元 62 用于根据所述 API 文件调用其对应资源的请求从指定目录中获取所述 API 文件对应的资源。

[0064] 第一发送单元 63 用于通过预设用户接口发送所述资源给所述目标系统,由所述目标系统中的所述 API 文件进行调用。由于无需将所述 API 文件对应的资源也给使用所述 API 文件进行目标系统开发的人员,实现了 API 文件与用户采用所述 API 文件开发的系统解耦。

[0065] 本发明实施例主要应用于地理信息系统,实现了 API 文件与用户采用所述 API 文件开发的系统解耦。

[0066] 通过以上的实施方式的描述,所属领域的技术人员可以清楚地了解到本发明可借助软件加必需的通用硬件的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在可读取的存储介质中,如计算机的软盘,硬盘或光盘等,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本发明各个实施例所述的方法。

[0067] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以所述权利要求的保护范围为准。

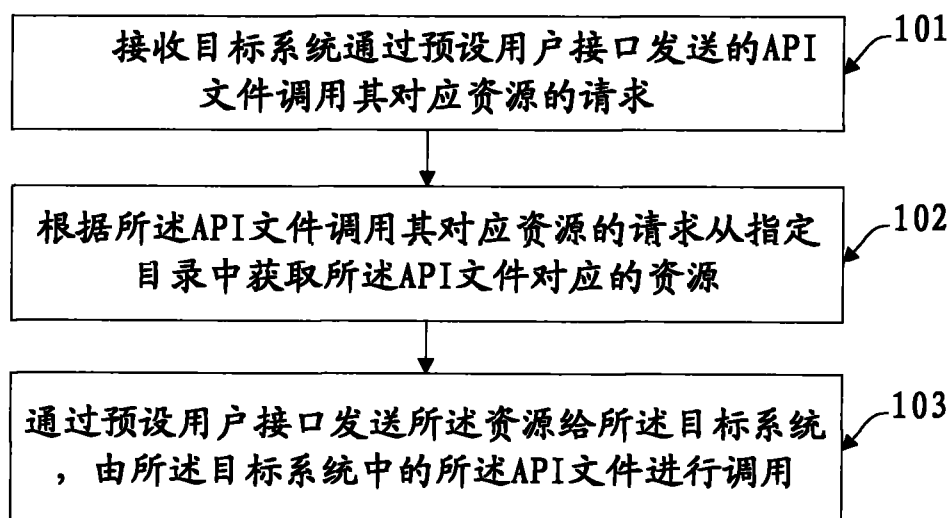


图 1

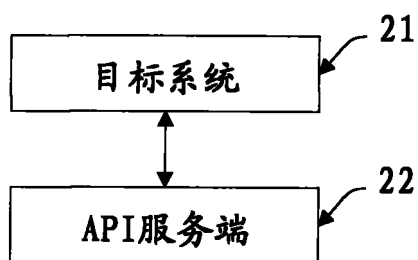


图 2

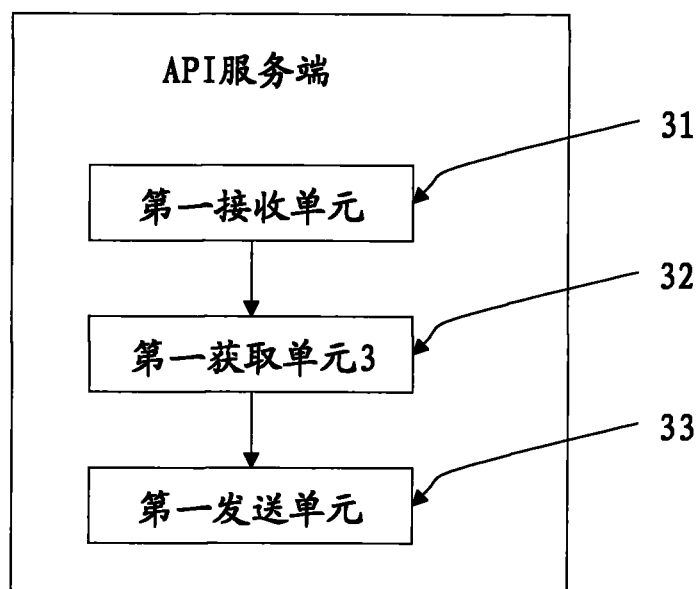


图 3

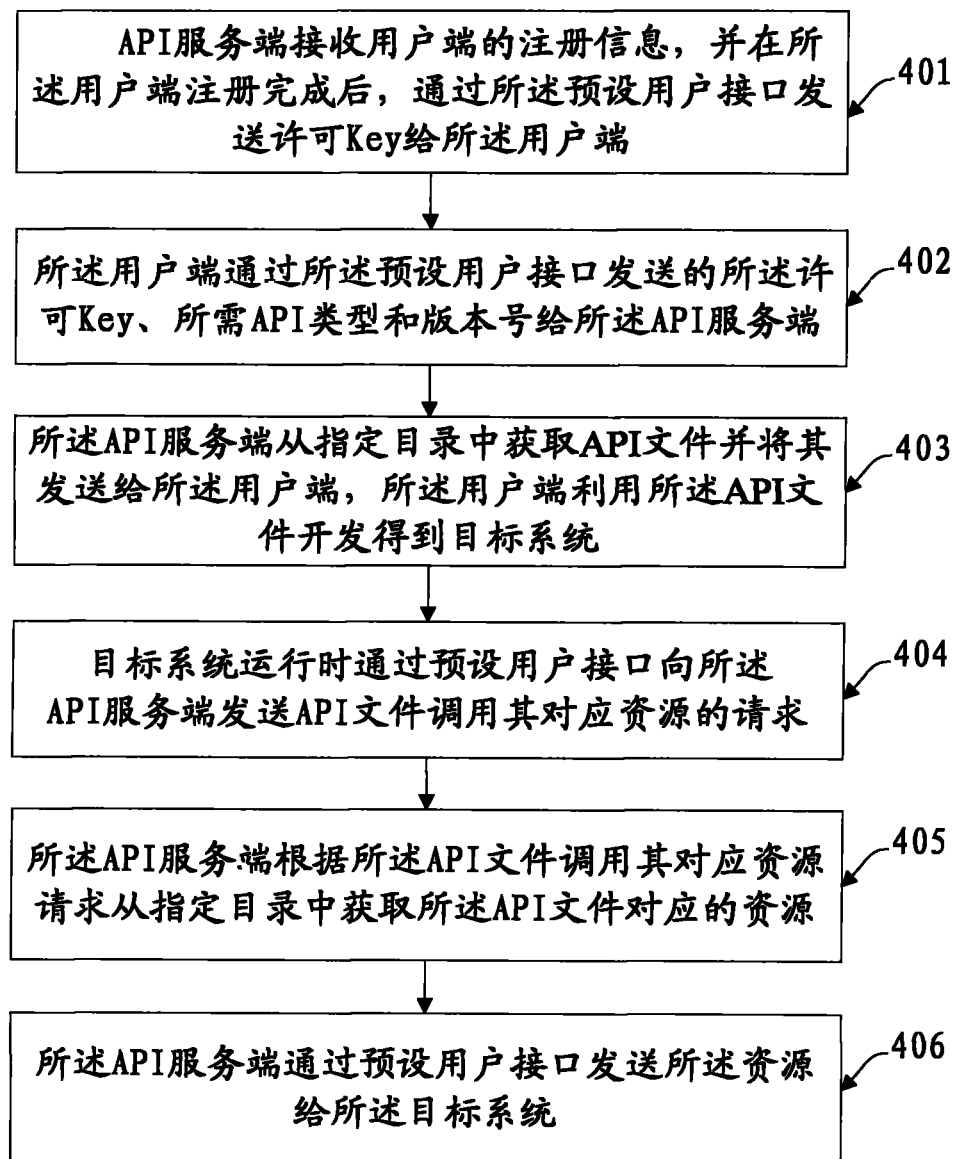


图 4

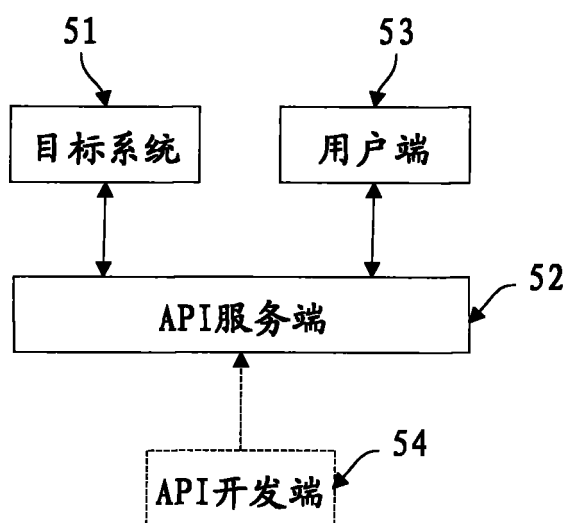


图 5

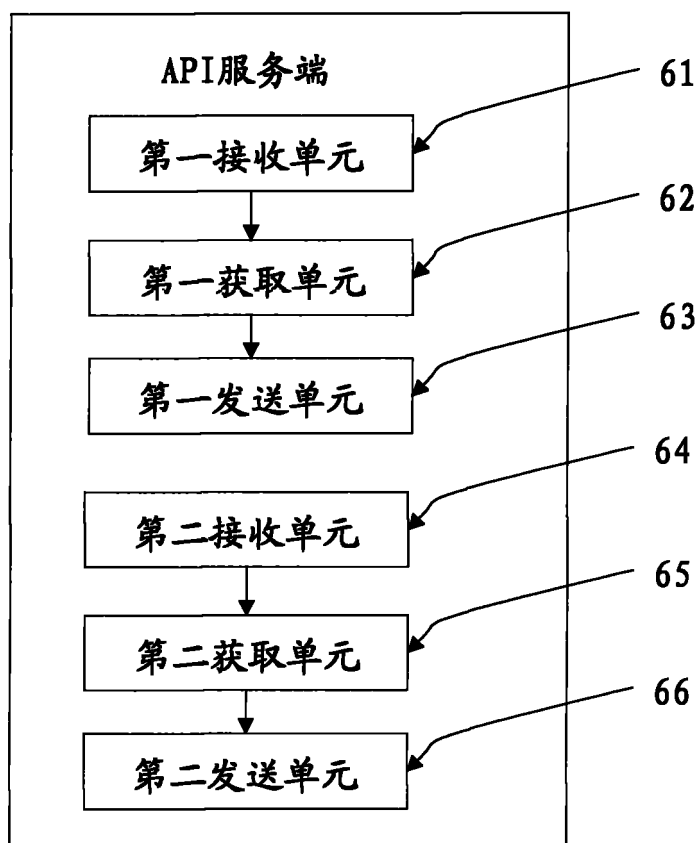


图 6