



(12)发明专利申请

(10)申请公布号 CN 109558290 A

(43)申请公布日 2019.04.02

(21)申请号 201811341726.6

(22)申请日 2018.11.12

(71)申请人 平安科技(深圳)有限公司

地址 518000 广东省深圳市福田区福田街
道福安社区益田路5033号平安金融中
心23楼

(72)发明人 唐可

(74)专利代理机构 深圳市沃德知识产权代理事
务所(普通合伙) 44347

代理人 高杰 于志光

(51)Int.Cl.

G06F 11/26(2006.01)

G06F 11/36(2006.01)

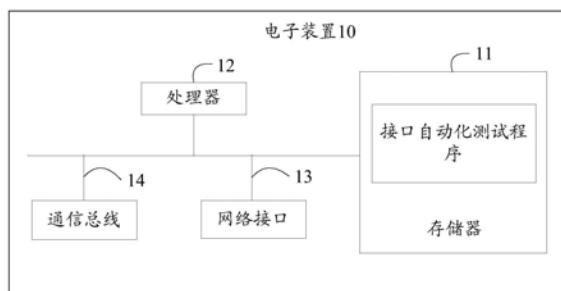
权利要求书2页 说明书11页 附图2页

(54)发明名称

服务器、接口自动化测试方法及存储介质

(57)摘要

本发明涉及功能测试,提出一种接口自动化测试方法,包括:接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;对所述XML测试配置文件进行解析,生成接口的测试脚本;调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。能够解决接口测试过程中数据分离带来的巨大工作量,提高接口测试效率。此外,本发明还提出一种服务器和存储介质。



1. 一种服务器,其特征在于,所述服务器包括存储器、及与所述存储器连接的处理器,所述处理器用于执行所述存储器上存储的接口自动化测试程序,所述接口自动化测试程序被所述处理器执行时实现如下步骤:

A1、接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;

A2、将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;

A3、对所述XML测试配置文件进行解析,生成接口的测试脚本;

A4、调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。

2. 如权利要求1所述的服务器,其特征在于,在所述步骤A2中,所述预定义的XML数据结构包括通过XML语言编写的携带有接口通信参数标签语句模板和目标测试参数标签语句模板的XML配置文件。

3. 如权利要求2所述的服务器,其特征在于,所述步骤A3包括:

将XML测试配置文件中的接口通信参数标签语句进行解析,得到接口测试脚本的请求头;

将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点。

4. 如权利要求3所述的服务器,其特征在于,所述将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点的步骤包括:

解析目标测试参数标签语句中的测试用例,将所述测试用例中需要处理的数据作为接口测试脚本的请求体,将所述测试用例中的断言参数作为接口测试脚本的断言点。

5. 如权利要求1-4任一所述的服务器,其特征在于,所述测试接口通信参数表示接口的连接访问信息,所述目标测试参数表示测试接口的测试代码的输入,用于实现测试接口的测试场景的测试。

6. 一种接口自动化测试方法,其特征在于,所述方法包括如下步骤:

S1、接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;

S2、将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;

S3、对所述XML测试配置文件进行解析,生成接口的测试脚本;

S4、调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。

7. 如权利要求6所述的接口自动化测试方法,其特征在于,在所述步骤S2中,所述预定义的XML数据结构包括通过XML语言编写的携带有接口通信参数标签语句模板和目标测试参数标签语句模板的XML配置文件。

8. 如权利要求7所述的接口自动化测试方法,其特征在于,所述步骤S3包括:

将XML测试配置文件中的接口通信参数标签语句进行解析,得到接口测试脚本的请求头;

将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点。

9. 如权利要求8所述的接口自动化测试方法,其特征在于,所述将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点的步骤包括:

解析目标测试参数标签语句中的测试用例,将所述测试用例中需要处理的数据作为接口测试脚本的请求体,将所述测试用例中的断言参数作为接口测试脚本的断言点。

10. 一种计算机可读存储介质,所述计算机可读存储介质存储有接口自动化测试程序,所述接口自动化测试程序可被至少一个处理器执行,以使所述至少一个处理器执行如权利要求6-9中任一项所述的接口自动化测试方法的步骤。

服务器、接口自动化测试方法及存储介质

技术领域

[0001] 本发明涉及接口测试领域,尤其涉及一种服务器、接口自动化测试方法及存储介质。

背景技术

[0002] 随着微服务架构的发展,接口测试变得越来越重要。这是由于系统间的通信几乎都是通过接口来进行数据的交互,甚至同一系统的前后端也需要通过接口来进行数据的传递,甚至是通过SDK包来提供服务的系统,如游戏领域、第三方依赖库(语言库、人工智能库等)、jar包服务等均需要通过接口进行交互。

[0003] 目前,接口测试工具很多,但大部分是通用型工具,在接口测试过程中需要进行脚本和数据的分离,导致测试效率不高,且对于异常测试不易维护。

发明内容

[0004] 有鉴于此,本发明提出一种服务器、接口自动化测试方法及存储介质,通过引入XML测试配置文件,避免脚本和数据的分离,能够提高测试效率,且便于异常测试的维护。

[0005] 首先,为实现上述目的,本发明提出一种服务器,所述服务器包括存储器、及与所述存储器连接的处理器,所述处理器用于执行所述存储器上存储的接口自动化测试程序,所述接口自动化测试程序被所述处理器执行时实现如下步骤:

[0006] A1、接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;

[0007] A2、将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;

[0008] A3、对所述XML测试配置文件进行解析,生成接口的测试脚本;

[0009] A4、调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。

[0010] 优选地,在所述步骤A2中,所述预定义的XML数据结构包括通过XML语言编写的携带有接口通信参数标签语句模板和目标测试参数标签语句模板的XML配置文件。

[0011] 优选地,所述步骤A3包括:

[0012] 将XML测试配置文件中的接口通信参数标签语句进行解析,得到接口测试脚本的请求头;

[0013] 将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点。

[0014] 优选地,所述将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点的步骤包括:

[0015] 解析目标测试参数标签语句中的测试用例,将所述测试用例中需要处理的数据作为接口测试脚本的请求体,将所述测试用例中的断言参数作为接口测试脚本的断言点。

[0016] 优选地,所述测试接口通信参数表示接口的连接访问信息,所述目标测试参数表

示测试接口的测试代码的输入,用于实现测试接口的测试场景的测试。

[0017] 此外,为了实现上述目的,本发明还提出一种接口自动化测试方法,所述方法包括如下步骤:

[0018] S1、接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;

[0019] S2、将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;

[0020] S3、对所述XML测试配置文件进行解析,生成接口的测试脚本;

[0021] S4、调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。

[0022] 优选地,在所述步骤S2中,所述预定义的XML数据结构包括通过XML语言编写的携带有接口通信参数标签语句模板和目标测试参数标签语句模板的XML配置文件。

[0023] 优选地,所述步骤S3包括:

[0024] 将XML测试配置文件中的接口通信参数标签语句进行解析,得到接口测试脚本的请求头;

[0025] 将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点。

[0026] 优选地,所述将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点的步骤包括:

[0027] 解析目标测试参数标签语句中的测试用例,将所述测试用例中需要处理的数据作为接口测试脚本的请求体,将所述测试用例中的断言参数作为接口测试脚本的断言点。

[0028] 此外,为实现上述目的,本发明还提出一种计算机可读存储介质,所述计算机可读存储介质存储有接口自动化测试程序,所述接口自动化测试程序可被至少一个处理器执行,以使所述至少一个处理器执行如上所述的接口自动化测试方法的步骤。

[0029] 本发明所提出的服务器、接口自动化测试方法及存储介质,通过接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;对所述XML测试配置文件进行解析,生成接口的测试脚本;调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。能够解决接口测试过程中数据分离带来的巨大工作量,提高接口测试效率。

附图说明

[0030] 图1是本发明提出的服务器一可选的硬件架构的示意图;

[0031] 图2是本发明服务器一实施例中接口自动化测试的程序模块示意图;

[0032] 图3是本发明接口自动化测试方法较佳实施例的实施流程图。

[0033] 本发明目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

具体实施方式

[0034] 为了使本发明的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本发明进行进一步详细说明。应当理解,此处所描述的具体实施例仅用以解释本发明,并不

用于限定本发明。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。

[0035] 需要说明的是，在本发明中涉及“第一”、“第二”等的描述仅用于描述目的，而不能理解为指示或暗示其相对重要性或者隐含指明所指示的技术特征的数量。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括至少一个该特征。另外，各个实施例之间的技术方案可以相互结合，但是必须是以本领域普通技术人员能够实现为基础，当技术方案的结合出现相互矛盾或无法实现时应当认为这种技术方案的结合不存在，也不在本发明要求的保护范围之内。

[0036] 参阅图1所示，是本发明提出的服务器一可选的硬件架构示意图。本实施例中，服务器10可包括，但不仅限于，可通过通信总线14相互通信连接存储器11、处理器12、网络接口13。需要指出的是，图1仅示出了具有组件11-14的服务器10，但是应理解的是，并不要求实施所有示出的组件，可以替代的实施更多或者更少的组件。

[0037] 其中，存储器11至少包括一种类型的计算机可读存储介质，计算机可读存储介质包括闪存、硬盘、多媒体卡、卡型存储器（例如，SD或DX存储器等）、随机访问存储器（RAM）、静态随机访问存储器（SRAM）、只读存储器（ROM）、电可擦除可编程只读存储器（EEPROM）、可编程只读存储器（PROM）、磁性存储器、磁盘、光盘等。在一些实施例中，存储器11可以是服务器10的内部存储单元，例如服务器10的硬盘或内存。在另一些实施例中，存储器11也可以是服务器10的外包存储设备，例如服务器10上配备的插接式硬盘，智能存储卡（Smart Media Card, SMC），安全数字（Secure Digital, SD）卡，闪存卡（Flash Card）等。当然，存储器11还可以既包括服务器10的内部存储单元也包括其外包存储设备。本实施例中，存储器11通常用于存储安装于服务器10的操作系统和各类应用软件，例如接口自动化测试程序等。此外，存储器11还可以用于暂时地存储已经输出或者将要输出的各类数据。

[0038] 处理器12在一些实施例中可以是中央处理器（Central Processing Unit, CPU）、控制器、微控制器、微处理器、或其他数据处理芯片。处理器12通常用于控制服务器10的总体操作。本实施例中，处理器12用于运行存储器11中存储的程序代码或者处理数据，例如运行的接口自动化测试程序等。

[0039] 网络接口13可包括无线网络接口或有线网络接口，网络接口13通常用于在服务器10与其他电子设备之间建立通信连接。

[0040] 通信总线14用于实现组件11-13之间的通信连接。

[0041] 图1仅示出了具有组件11-14以及接口自动化测试的服务器10，但是应理解的是，并不要求实施所有示出的组件，可以替代的实施更多或者更少的组件。

[0042] 可选地，服务器10还可以包括用户接口（图1中未示出），用户接口可以包括显示器、输入单元比如键盘，其中，用户接口还可以包括标准的有线接口、无线接口等。

[0043] 可选地，在一些实施例中，显示器可以是LED显示器、液晶显示器、触控式液晶显示器以及OLED触摸器等。进一步地，显示器也可称为显示屏或显示单元，用于显示在服务器10中处理信息以及用于显示可视化的用户界面。

[0044] 可选地，在一些实施例中，服务器10还可以包括音频单元（音频单元图1中未示出），音频单元可以在服务器10处于呼叫信号接收模式、通话模式、记录模式、语音识别模式、广播接收模式等等模式下时，将接收的或者存储的音频数据转换为音频信号；进一步

地,服务器10还可以包括音频输出单元,音频输出单元将音频单元转换的音频信号输出,而且音频输出单元还可以提供与服务器10执行的特定功能相关的音频输出(例如呼叫信号接收声音、消息接收声音等等),音频输出单元可以包括扬声器、蜂鸣器等等。

[0045] 可选地,在一些实施例中,服务器10还可以包括警报单元(图中未示出),警报单元可以提供输出已将事件的发生通知给服务器10。典型的事件可以包括呼叫接收、消息接收、键信号输入、触摸输入等等。除了音频或者视频输出之外,警报单元可以以不同的方式提供输出以通知事件的发生。例如,警报单元可以以震动的形式提供输出,当接收到呼叫、消息或一些其他可以使服务器10进入通信模式时,警报单元可以提供触觉输出(即,振动)以将其通知给用户。

[0046] 在一实施例中,存储器11中存储的接口自动化测试程序被处理器12执行时,实现如下操作:

[0047] A1,接收接口测试请求,所述测试请求包括测试接口通信参数和目标测试参数;

[0048] 具体地,当需要进行接口测试时,服务器会接收到测试请求,测试请求中包括测试接口通信参数和目标测试参数。测试接口通信参数包括接口通信协议类型、接口通信模式和数据传输格式中的至少一种,用来表示接口的连接访问信息;目标测试参数用来表示测试接口的测试代码的输入,用于实现测试接口的测试场景的测试。

[0049] A2,将所述测试接口通信参数和所述目标测试参数分别代入预定义的

[0050] XML数据结构,生成XML测试配置文件;

[0051] 具体地,XML(Extensible Markup Language,可扩展标记语言)是一种用于标记文件使其具有结构性的标记语言,所述预定义的XML数据结构包括通过XML语言编写的携带有接口通信参数标签语句模板和目标测试参数标签语句模板的XML配置文件。

[0052] 进一步地,所述预定义的xml数据结构的编写规则包括:

[0053] 1) 编写所述接口通信参数标签语句模板,包括开始标签和结束标签;所述接口通信参数标签语句模板包括各通信参数的标签;例如通信协议类型标签、接口通信模式标签和数据传输格式标签等,具体地,涉及到接口名称需填写接口的中文名称;涉及到接口连接信息的域名和路径则根据不同的通信环境填写;涉及到接口方法名称则填写该接口定义的准确的名称等。

[0054] 2) 所述目标测试参数标签语句模板由测试用例标签组成,每条测试用例使用“testcase”标签来表示(“<testcase>,</testcase>”),而测试用例标签下面又分为“data”目标测试参数和“assert”断言参数,“data”标签下以“键值对”形式表示需要填写接口的测试参数与值,断言参数需要根据接口实际响应返回的情况正常填写;通常设置有多条测试用例对应的多个“testcase”标签,进一步地,欲构造目标测试参数传参key为空时,则直接不写目标测试参数的标签即可,不用像现有技术中的方式独立建立一个接口测试脚本,使用极其方便。

[0055] 进一步地,所述预定义的XML数据结构还包括嵌套规则标签,这是由于目前的接口测试工具通常支持三层嵌套(共四层数据),在XML数据结构中在需要嵌套的标签语句中加属性type=“object”和isBatchQuantity=“yes”(嵌套的支持批量填写yes,不支持批量填no),其中第一层嵌套数据需要填写在XML数据结构中“OneBatchQuantity”元素标签下,第二层嵌套数据需要填写在XML数据结构中“TwoBatchQuantity”元素标签下,第三层嵌套数

据需要填写在XML数据结构中“ThreeBatchQuantity”元素标签下,并在该标签下填写嵌套的各个参数键值对。例如,在一实施方式中,假设生成的XML测试配置文件的名称为“getAccessToken.xml”,对应的数据结构如下:

```
<?xml version="1.0" encoding=" UTF-8"?>
```

[0056]

```
<interface>
```

```
<interfacemethod>get</interfacemethod>
```


[0057]

```
<interfacename>test-ESBX_OAUTH-获取应用口令接口</interfacename>
<interfaceservername>esg-oauth-stg.paic.com.cn </interfaceservername>
<interfacepath>/oauth/oauth2/access_token </interfacepath>
<!--接口是否支持批量，即 data={}为非批量，需要写 no;data=[{}]为批量，
      需要填写 yes. -->
<isBatchQuantity>yes</isBatchQuantity>
<!--第一条测试用例-->
<testcase>
  <data>
    <client_id>P_web2_pussy</client_id>
    <grant_type>client_credentials</grant_type>
    <client_secret>Rep586Ke</client_secret>
  </data>
  <assert>"ret":"0"</assert>
</testcase>
<!--第二条测试用例-->
<testcase>
  <data>
    <client_id>P_dmz_pussy</client_id>
    <grant_type>client_credentials</grant_type>
    <client_secret>Rep586K1</client_secret>
  </data>
  <assert>"ret":"12305"</assert>
</testcase>
```

[0058] 本实施例中，XML的第一行为xml的声明语句，通常声明语句前面没有空格，若构造业务参数传参key为空时，则直接不写业务参数的标签即可，不需要独立建立一个接口测试脚本，极其方便。

- [0059] A3、对所述XML测试配置文件进行解析,生成接口测试脚本;
- [0060] 具体地,所述对所述XML测试配置文件进行解析,生成测试脚本的步骤包括:
- [0061] 将XML测试配置文件中的接口通信参数标签语句进行解析,得到接口测试脚本的请求头;
- [0062] 将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点。
- [0063] 例如,在本实施例的一实施方式中,将XML测试配置文件中的接口通信参数标签语句中的接口通信参数进行拼接(interface method+interface name+interface server name+interface path),以得到对应的唯一键,将该唯一键作为接口测试脚本的请求头。
- [0064] 需要说明的是,XML测试配置文件中的接口通信标签语句的标签名以及主要标签名不可更改,基于这几个固定标签名进行分类解析,如标签名以interface开头的则视为接口通信标签,其解析结果直接组装成接口测试脚本的主体,所述接口测试脚本的主体即为请求头。
- [0065] 进一步地,将XML测试配置文件中的目标测试参数标签语句进行解析,具体的,解析目标测试参数标签语句中的testcase(测试用例),其解析结果作为接口测试脚本的请求体和断言点。具体地,将Testcase中的data(需要处理的数据,目标测试参数)和assert(断言点参数)标签,当存在多个testcase时,视为接口通信标签不变,基于多个testcase生成不同的请求体和断言点。
- [0066] 进一步地,当testcase中存在多层嵌套标签时,则进行多层嵌套解析,依次获取每一层嵌套的标签名和值,并进行包装传递。
- [0067] A4、调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。
- [0068] 由上述事实例可知,本发明提出的服务器,通过接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;对所述XML测试配置文件进行解析,生成接口的测试脚本;调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。能够解决接口测试过程中数据分离带来的巨大工作量,提高接口测试效率。
- [0069] 此外,本发明的接口自动化测试依据其各部分所实现的功能不同,可用具有相同功能的程序模块进行描述。请参阅图2所示,是本发明服务器一实施例中接口自动化测试的程序模块示意图。本实施例中,接口自动化测试依据其各部分所实现的功能的不同,可以被分割成接收模块201、第一生成模块202、第二生成模块203、测试模块204以及确定模块205。由上面的描述可知,本发明所称的程序模块是指能够完成特定功能的一系列计算机程序指令段,比程序更适合于描述接口自动化测试程序在服务器10中的执行过程。所述模块201-204所实现的功能或操作步骤均与上文类似,此处不再详述,示例性地,例如其中:
- [0070] 接收模块201用于接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;
- [0071] 第一生成模块202用于将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;

[0072] 第二生成模块203用于对所述XML测试配置文件进行解析,生成接口的测试脚本;

[0073] 测试模块204用于调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。

[0074] 此外,本发明还提出一种接口自动化测试方法,请参阅图3所示,所述接口自动化测试方法包括如下步骤:

[0075] S301,接收接口测试请求,所述测试请求包括测试接口通信参数和目标测试参数;

[0076] 具体地,当需要进行接口测试时,服务器会接收到测试请求,测试请求中包括测试接口通信参数和目标测试参数。测试接口通信参数包括接口通信协议类型、接口通信模式和数据传输格式中的至少一种,用来表示接口的连接访问信息;目标测试参数用来表示测试接口的测试代码的输入,用于实现测试接口的测试场景的测试。

[0077] S302,将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;

[0078] 具体地,XML (Extensible Markup Language,可扩展标记语言)是一种用于标记文件使其具有结构性的标记语言,所述预定义的XML数据结构包括通过XML语言编写的携带有接口通信参数标签语句模板和目标测试参数标签语句模板的XML配置文件。

[0079] 进一步地,所述预定义的xml数据结构的编写规则包括:

[0080] 1) 编写所述接口通信参数标签语句模板,包括开始标签和结束标签;所述接口通信参数标签语句模板包括各通信参数的标签;例如通信协议类型标签、接口通信模式标签和数据传输格式标签等,具体地,涉及到接口名称需填写接口的中文名称;涉及到接口连接信息的域名和路径则根据不同的通信环境填写;涉及到接口方法名称则填写该接口定义的准确的名称等。

[0081] 2) 所述目标测试参数标签语句模板由测试用例标签组成,每条测试用例使用“testcase”标签来表示(“<testcase>,</testcase>”),而测试用例标签下面又分为“data”目标测试参数和“assert”断言参数,“data”标签下以“键值对”形式表示需要填写接口的测试参数与值,断言参数需要根据接口实际响应返回的情况正常填写;通常设置有多条测试用例对应的多个“testcase”标签,进一步地,欲构造目标测试参数传参key为空时,则直接不写目标测试参数的标签即可,不用像现有技术中的方式独立建立一个接口测试脚本,使用极其方便。

[0082] 进一步地,所述预定义的XML数据结构还包括嵌套规则标签,这是由于目前的接口测试工具通常支持三层嵌套(共四层数据),在XML数据结构中在需要嵌套的标签语句中加属性type=“object”和isBatchQuantity=“yes”(嵌套的支持批量填写yes,不支持批量填写no),其中第一层嵌套数据需要填写在XML数据结构中“OneBatchQuantity”元素标签下,第二层嵌套数据需要填写在XML数据结构中“TwoBatchQuantity”元素标签下,第三层嵌套数据需要填写在XML数据结构中“ThreeBatchQuantity”元素标签下,并在该标签下填写嵌套的各个参数键值对。例如,在一实施方式中,假设生成的XML测试配置文件的名称为“getAccessToken.xml”,对应的数据结构如下:

[0083]

```
<?xml version="1.0" encoding=" UTF-8"?>
    <interface>
        <interfacemethod>get</interfacemethod>
        <interfacename>test-ESBX_OAUTH-获取应用口令接口</interfacename>
        <interfaceservername>esg-oauth-stg.paic.com.cn </interfaceservername>
        <interfacepath>/oauth/oauth2/access_token </interfacepath>
        <!--接口是否支持批量，即 data={}为非批量，需要写 no;data=[{}]为批量，
            需要填写 yes. -->
        <isBatchQuantity>yes</isBatchQuantity>
        <!--第一条测试用例-->
        <testcase>
            <data>
                <client_id>P_web2_pussy</client_id>
                <grant_type>client_credentials</grant_type>
                <client_secret>Rep586Ke</client_secret>
            </data>
            <assert>"ret":"0"</assert>
        </testcase>
        <!--第二条测试用例-->
        <testcase>
            <data>
                <client_id>P_dmz_pussy</client_id>
                <grant_type>client_credentials</grant_type>
```

```
<client_secret>Rep586K1</client_secret>
```

```
</data>
```

[0084]

```
<assert>"ret":"12305"</assert>
```

```
</testcase>
```

[0085] 本实施例中,XML的第一行为xml的声明语句,通常声明语句前面没有空格,若构造业务参数传参key为空时,则直接不写业务参数的标签即可,不需要独立建立一个接口测试脚本,极其方便。

[0086] S303,对所述XML测试配置文件进行解析,生成接口测试脚本;

[0087] 具体地,所述对所述XML测试配置文件进行解析,生成测试脚本的步骤包括:

[0088] 将XML测试配置文件中的接口通信参数标签语句进行解析,得到接口测试脚本的请求头;

[0089] 将所述XML测试配置文件中的目标测试参数标签语句进行解析,得到接口测试脚本的请求体和断言点。

[0090] 例如,在本实施例的一实施方式中,将XML测试配置文件中的接口通信参数标签语句中的接口通信参数进行拼接(interface method+interface name+interface server name+interface path),以得到对应的唯一键,将该唯一键作为接口测试脚本的请求头。

[0091] 需要说明的是,XML测试配置文件中的接口通信标签语句的标签名以及主要标签名不可更改,基于这几个固定标签名进行分类解析,如标签名以interface开头的则视为接口通信标签,其解析结果直接组装成接口测试脚本的主体,所述接口测试脚本的主体即为请求头。

[0092] 进一步地,将XML测试配置文件中的目标测试参数标签语句进行解析,具体的,解析目标测试参数标签语句中的testcase(测试用例),其解析结果作为接口测试脚本的请求体和断言点。具体地,将Testcase中的data(需要处理的数据,目标测试参数)和assert(断言点参数)标签,当存在多个testcase时,视为接口通信标签不变,基于多个testcase生成不同的请求体和断言点。

[0093] 进一步地,当testcase中存在多层嵌套标签时,则进行多层嵌套解析,依次获取每一层嵌套的标签名和值,并进行包装传递。

[0094] S304,调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。

[0095] 由上述事实例可知,本发明提出的接口自动化测试方法,通过接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构,生成XML测试配置文件;对所述XML测试配置文件进行解析,生成接口的测试脚本;调用与所述测试脚本对应的测试工具,触发所述测试脚本进行接口测试,并基于所述测试结果输出测试报告。能够解决接口测试过程中数据分离带来的巨大工作量,提高接口测试效率。

[0096] 此外,本发明还提出一种计算机可读存储介质,所述计算机可读存储介质上存储有接口自动化测试程序,所述接口自动化测试程序被处理器执行时实现如下操作:

[0097] 接收接口测试请求,测试请求包括测试接口通信参数和目标测试参数;

[0098] 将所述测试接口通信参数和所述目标测试参数分别代入预定义的XML数据结构，生成XML测试配置文件；

[0099] 对所述XML测试配置文件进行解析，生成接口的测试脚本；

[0100] 调用与所述测试脚本对应的测试工具，触发所述测试脚本进行接口测试，并基于所述测试结果输出测试报告。

[0101] 本发明计算机可读存储介质的具体实施过程，与上述服务器以及接口自动化测试方法的具体实施过程类似，在此不再赘述。

[0102] 通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现，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一个存储介质（如ROM/RAM、磁碟、光盘）中，包括若干指令用以使得一台终端设备（可以是手机，计算机，服务器，空调器，或者网络设备）执行本发明各个实施例所述的方法。以上仅为本发明的优选实施例，并非因此限制本发明的专利范围，凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换，或直接或间接运用在其他相关的技术领域，均同理包括在本发明的专利保护范围内。

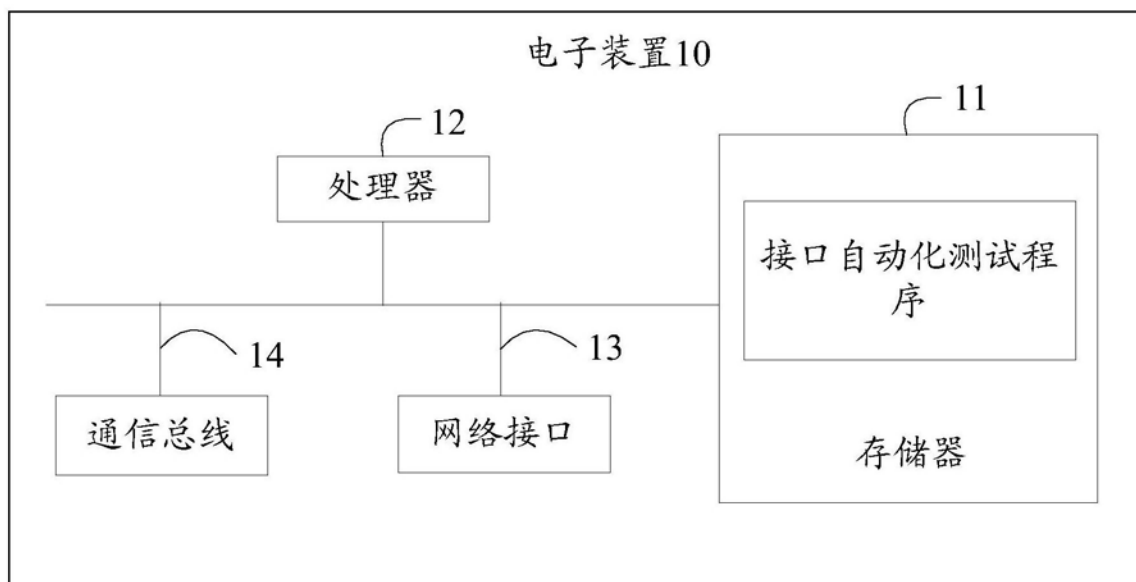


图1

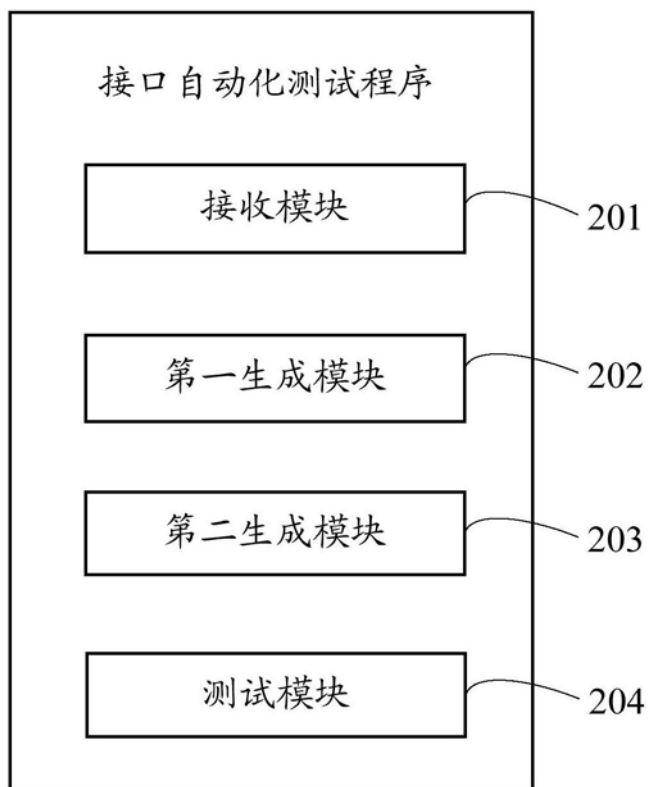


图2

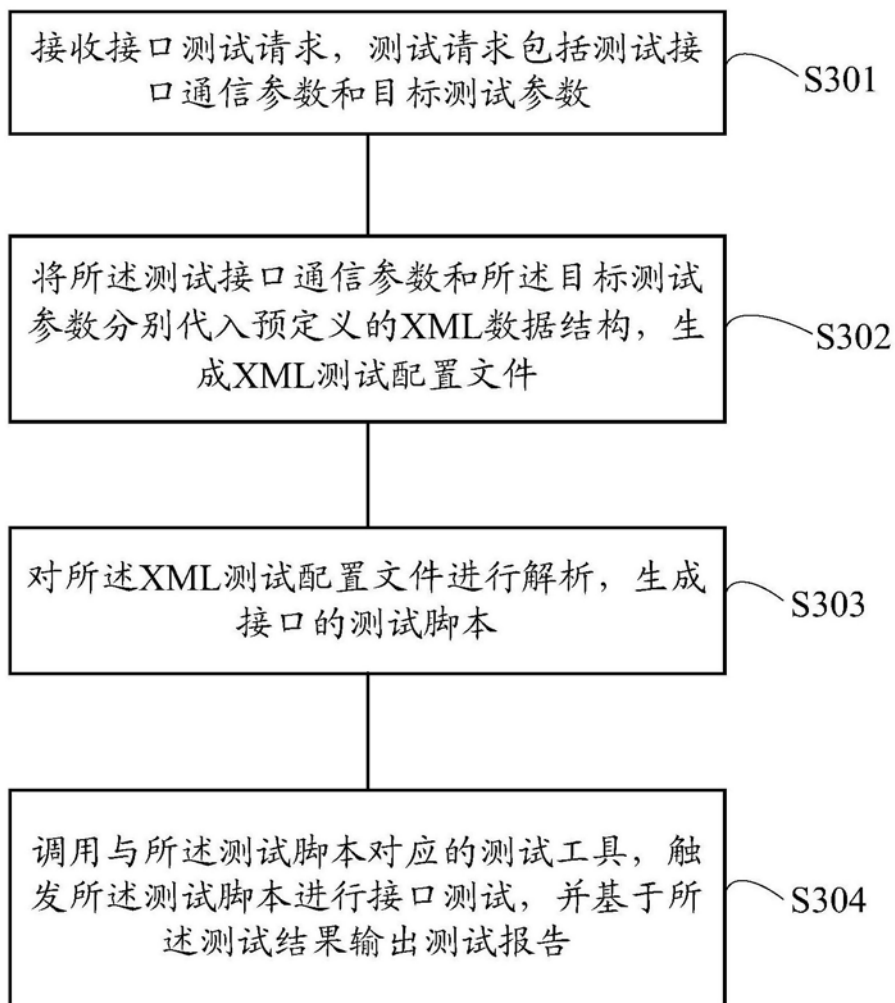


图3