

---

# 目录

<b>1</b>	<b>引言 .....</b>	<b>4</b>
1.1	编写目的.....	4
1.2	背景.....	4
1.3	定义.....	5
1.4	参考资料.....	5
<b>2</b>	<b>应用程序接口及使用说明 .....</b>	<b>6</b>
2.1	原子服务相关接口及使用说明.....	6
2.1.1	原子服务部署.....	6
2.1.2	原子服务反部署.....	7
2.1.3	原子服务调用.....	7
2.2	组合服务相关接口与使用说明.....	9
2.2.1	组合服务部署.....	10
2.2.2	组合服务反部署.....	10
2.2.3	组合服务调用.....	11
2.2.4	组合服务监控.....	11
2.3	WEB 应用相关接口及说明 .....	12
2.3.1	Web 应用部署.....	12
2.3.2	Web 应用反部署.....	13
<b>3</b>	<b>总结 .....</b>	<b>13</b>

---

## 表目

表 1 原子服务部署 .....	6
表 2 原子服务反部署 .....	7
表 3 原子服务调用 (1) .....	7
表 4 原子服务调用 (2) .....	8
表 5 原子服务调用 (3) .....	9
表 6 组合服务部署 (1) .....	10
表 7 组合服务部署 (2) .....	10
表 8 组合服务反部署 .....	10
表 9 组合服务调用 .....	11
表 10 组合服务监控 (1) .....	11
表 11 组合服务监控 (2) .....	12
表 12 组合服务监控 (3) .....	12
表 13 WEB 应用部署 .....	12
表 14 WEB 应用反部署 .....	12

---

## 图目

图 1 原子服务调用文件格式.....	9
---------------------	---

---

# 应用程序接口及使用说明

## 1 引言

### 1.1 编写目的

本文是平台的应用程序接口及使用说明，阐述了用户使用该平台向用户提供的应用程序接口及具体的使用方法。

### 1.2 背景

随着互联网的快速发展，企业应用呈爆发式增长。由于业务需求的多样性和复杂性，企业之间开始寻求一种跨企业的应用集成方法。面向服务的软件开发方法由此应运而生，它致力于制定企业应用的标准开放接口、通信协议，并以之为基础解决跨企业的应用集成问题。随着面向服务软件开发方法的进一步发展，它逐渐演变成为一种新的分布式软件开发方法。这种开发方法讨论如何按需、动态的集成现有的服务组件，从而快速开发出满足新需求的分布式软件，这个过程也称之为服务组合。以业务流程为核心的服务组合是面向服务软件开发方法的主流实现之一，类比传统的软件开发方法，它需要业务建模、组合服务编制、组合服务验证、组合服务测试等几个阶段。其中，在组合服务测试阶段，由于需要相对真实的分布式部署、运行和测试环境，一方面需要测试人员安装、配置、维护分布式测试环境，工作量大；另一方面，测试环境的隔离特性、多变特性也要求测试环境满足动态、按需以及可信等特征。这都使得传统的服务化软件测试平台无法满足测试需求，需要研发新的服务化软件测试平台。

近年来，云计算无论在工业界还是在学术界都已经被公认为一种新的计算模式，它强调通过因特网来进行软件的透明分发，软件以服务的方式提供使用。从服务模式的角度划分，云计算应用可以分为 SaaS（软件即服务）、PaaS（平台即服务）、IaaS（基础设施即服务）三种。无论是那一种服务模式，用户都可以通过因特网透明的即时访问云计算应用，而不用关心后台的具体实现细节以及物理设施的具体位置。采用云计算技术来构建一个服务化软件测试平台可以满足服务化软件测试的新需求，它支持服务化软件测试环境的云管理、自动化构建和服务

---

化软件的即时部署和测试。

## 1.3 定义

**SOAP:** Simple Object Access Protocol 简单对象访问控制协议是一种简单的基于 XML 的协议，它使应用程序通过 HTTP 来交换信息。

**BPMN:** Business Process Modeling Notation 业务流程建模标注是一个全新的业务流程建模和 web 服务标准。

**组合服务:** 随着 web 服务技术的日益成熟，越来越多的稳定易用 web 服务共享在网络上。但单个的 web 服务能够提供的功能有限，为了更加充分地利用共享的 web 服务，有必要将共享的 web 服务组合起来，这种由原子服务组合而成的服务就称为组合服务。

**服务部署:** 指将开发后的服务装载到服务容器中，使服务能够被用户调用的过程。广义的服务部署还包括服务的反部署，即将服务从服务容器中卸载的过程，以及服务的重部署，即更新已经部署的服务的过程。

## 1.4 参考资料

[1] M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann, "Service-Oriented Computing: State of the Art and Research Challenges," in IEEE Computer. vol. 40 (11), 2007, pp. 64-71.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. "Above the clouds: A Berkeley view of cloud computing". Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[3] IBM Smart Business Development and Test Cloud. <http://www-935.ibm.com/services/us/index.wss/offering/middleware/a1030965>.

[4] M. Hajjat, X. Sun, Y. E. Sung, D. Maltz, S. Rao, K. Sripanidkulchai and M. Tawarmalani, "Cloudward Bound: Planning for Beneficial Migration of Enterprise Applications to the Cloud". SIGCOMM 2010.

[5] H. Sun, X. Liu, X. Li, J. Zeng, Z. Huang, "SOARWare: A Service Oriented Software Production and Running Environment," Proceedings of the 12th

---

International Asia-Pacific Web Conference(APWeb 2010).

[6] SOARProLine. <http://www.trustie.net/projects/project/show/SOARProLine>.

[7] Hailong Sun, Xu Wang, Chao Zhou, Zicheng Huang, Xudong Liu. Early Experience of Building a Cloud Platform for Service Oriented Software Development. IEEE International Conference on Cluster Computing (Cluster 2010)

[8] Google App Engine, <http://code.google.com/appengine/>.

[9] Service Exchange, <http://www.servicexchange.cn>.

[10]Apache ServiceMix, <http://servicemix.apache.org/home.html>.

[11]Apache Axis2, <http://axis.apache.org/>.

## 2 应用程序接口及使用说明

基于云计算技术的服务化软件测试平台平台提供的应用程序接口包括原子服务部署、反部署和执行，以及组合服务的部署、反部署、执行和监控两大类接口。各接口输入输出以及使用说明如下所示。

### 2.1 原子服务相关接口及使用说明

原子服务相关的接口包括原子服务部署，反部署以及执行三方面的接口。

#### 2.1.1 原子服务部署

原子服务部署，即将用户开发的或者引用第三方的原子服务装在到基于云计算技术的服务化软件测试平台平台中的服务容器中，使服务能够被用户调用。参数列表如下。

表 1 原子服务部署

方法名	webServiceDeploy		
输入参数	availability	类型	String
		说明	所要部署的服务的服务的可用性
	serviceArchive	类型	File
		说明	原子服务归档文件
输出	服务部署反馈报文		

如表 1 所示，原子服务部署接口 `webServiceDeploy()` 提供两个参数。其中，`availability` 指用户所部署的服务在平台中的可用性。平台通过用户想要的可用性来决定该服务在平台中的副本个数。这些副本对于用户来说是透明的。服务的副本数  $n$  与 `availability` 的关系是  $availability = 1 - 0.2^n$ ，即当所要部署的一个副本的服务时，令 `availability=0.8` 即可。原子服务部署后，平台将为其分配唯一的标识。通过选择 `availability` 来部署的多个副本具有相同的标识。

原子服务要部署在平台中首先需要将其打包成 `.aar` 文件。其中，`.aar` 文件中包括名为 `services.xml` 的描述文件以及服务的 `.class` 文件。`serviceArchive` 即服务的 `aar` 包。

## 2.1.2 原子服务反部署

原子服务的反部署，即将用户已经部署在平台上的服务从基于云计算技术的服务化软件测试平台平台中的服务容器中删除。服务被反部署后，用户将不能再继续调用此服务。参数列表如下。

表 2 原子服务反部署

方法名	webServiceUndeploy		
输入参数	id	类型	String
		说明	要反部署的服务的 ID
输出	服务反部署报文		

如上表所示，原子服务反部署接口 `webServiceUndeploy()` 需要输入一个参数：需要被反部署的服务的 `id`。该 `id` 是原子服务在平台中的唯一标识，而这一标识可以对应多个原子服务副本。通过原子服务 `id` 来反部署服务可以将改原子服务所对应的所有的副本从平台中删除，使其不能再被用户调用。

## 2.1.3 原子服务调用

原子服务调用接口提供用户通过平台调用自己或者其他人部署的可用原子服务的方法。该部分共包括三个接口，分别如表 3，表 4，表 5 所示。

表 3 原子服务调用（1）

方法名	webServiceInvoke
-----	------------------

输入参数	paraList	类型	OMElement
		说明	所调用的原子服务所需的参数
	id	类型	String
		说明	要调用的原子服务的 id
	userID	类型	String
		说明	原子服务调用者的用户 ID
	userType	类型	String
		说明	服务调用者之于该服务的角色类型
输出	服务调用反馈报文		

如表 3 所示，id 表示所要调用的原子服务在平台中的唯一标识。userID 表示调用者在平台中的唯一标识，userType 表示用户基于该服务的角色，owner 表示该用户是这个服务的所有者。paraList 中包括了调用该服务的某个方法时必须提供的参数列表。

表 4 原子服务调用（2）

方法名	webServiceInvoke		
输入参数	id	类型	String
		说明	原子服务的 id
	operation	类型	String
		说明	要调用原子服务的具体方法名
	paraList	类型	自定义类型 ParameterList4WS[]
		说明	所要调用的服务所需的参数列表
	userName	类型	String
		说明	原子服务调用者的用户 ID
	userType	类型	String
		说明	服务调用者之于该服务的角色类型
输出	服务调用反馈报文		

如表 4 所示，id 表示所要调用的原子服务在平台中的唯一标识，operation 指所要调用的原子服务的方法。在 body 中 userID 表示调用者在平台中的唯一标识，userType 表示用户基于该服务的角色，owner 表示该用户是这个服务的所有



者。paraList 中包括了调用该服务的某个方法时必须提供的参数列表。

表 5 原子服务调用 (3)

方法名	webServiceInvoke		
输入参数	file	类型	File
		说明	服务调用信息的 xml 文件
输出	服务调用反馈报文		

如表 5 所示，输入参数 file 中包括调用该服务所需要的信息。文件的格式如图 1 所示。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <invokeRequest type="serviceID" isEngine="true">
3   <serviceID>WS_927</serviceID>
4   <operation>alert</operation>
5   <requestSoap>
6     <Enve>
7       <Header></Header>
8       <Body userID="ymz" userType="owner">
9         <paraList>
10          <para>
11            <paraName>email</paraName>
12            <paraType>String</paraType>
13            <paraValue>123@163.com</paraValue>
14          </para>
15        </paraList>
16      </Body>
17    </Enve>
18  </requestSoap>
19 </invokeRequest>
```

图 1 原子服务调用文件格式

在上图所示文件中，serviceID 表示所要调用的原子服务在平台中的唯一标识，operation 指所要调用的原子服务的方法。在 body 中 userID 表示调用者在平台中的唯一标识，userType 表示用户基于该服务的角色，owner 表示该用户是这个服务的所有者。paraList 中包括了调用该服务的某个方法时必须提供的参数。

## 2.2 组合服务相关接口与使用说明

组合服务相关接口包括组合服务部署，反部署，执行以及组合服务流程监控四部分。组合服务是将某些原子服务按照一定的逻辑将原子服务组合起来并完成某一系列功能的服务。

## 2.2.1 组合服务部署

与原子服务部署不相同，组合服务部署并不是将服务的 **aar** 包装在到基于云计算技术的服务化软件测试平台平台中去，而是将组合服务的描述文件装载到基于云计算技术的服务化软件测试平台平台中。

表 6 组合服务部署（1）

方法名	bpmnDeploy		
输入参数	name	类型	String
		说明	所要部署的组合服务的名字
	bpmn	类型	File
		说明	组合服务描述文件
输出	服务部署反馈报文		

组合服务描述文件是以**.bpmn** 结尾的文件。在组合服务描述文件中指定了所包含的原子服务的名称，**id**，**url** 以及执行的逻辑判断。

表 7 组合服务部署（2）

方法名	bpmnDeploy		
输入参数	bpmn	类型	File
		说明	组合服务描述文件
输出	服务部署反馈报文		

## 2.2.2 组合服务反部署

同样，组合服务的反部署也不是将具体的服务 **aar** 包从平台中删除，而是将组合服务描述的 **bpmn** 文件从平台中删除。

表 8 组合服务反部署

方法名	bpmnUndeploy		
输入参数	sn	类型	String
		说明	要反部署的服务名
	sId	类型	String
		说明	要反部署的服务的 ID

输出	反部署反馈报文
----	---------

### 2.2.3 组合服务调用

组合服务的调用及平台通过解析.bpmn 文件来执行相对应的原子服务的过程。其参数列表如下表所示。

表 9 组合服务调用

方法名	bpmnInvoke		
输入参数	serviceName	类型	String
		说明	所要调用的服务名
	serviceID	类型	String
		说明	所调用的服务 ID
	ps	类型	List
		说明	调用组合服务时所需要的参数列表，List 的每一个对象须包含参数名称，参数值，参数类型三部分值
输出	服务执行反馈报文		

### 2.2.4 组合服务监控

由于组合服务是由多个原子服务组合而成的，所以每一次执行组合服务的流程可能都不一致。每执行一次组合服务，平台就为该组合服务分配一个 jobId，该 jobId 是组合服务每一次执行的标识，通过该标识可以清楚地看到组合服务执行的中间结果集合。

表 10 组合服务监控（1）

方法名	getJobById		
输入参数	jobId	类型	String
		说明	所要监控的组合服务的 Id
输出	自定义类型 Job		

如上表所示，用户可以通过 jobId 查询组合服务的某一次执行。自定义类型包含了组合服务的名字和组合服务 id 的信息。

为了有效地监控组合服务执行流程，我们需要知道组合服务执行所产生的一些中间数据。通过基于云计算技术的服务化软件测试平台 d 平台，我们可以通过 jobId 来获得这些数据。该方法的参数列表如下。

表 11 组合服务监控（2）

方法名	getMonitorRecordByJobId		
输入参数	jobId	类型	String
		说明	所要监控的组合服务的 Id
输出	自定义类型 MonitorRecord 的列表		

在表 11 中，自定义类型 MonitorRecord 中包含了组合服务每一节点的状态，执行时间，执行是否成功等信息，还包含了每个节点处所执行的原子服务的输入的参数列表。

执行组合服务后，执行的结果我们也可以通过平台来查看。其参数列表及说明如下表所示。

表 12 组合服务监控（3）

方法名	getResultRecordByJobId		
输入参数	jobId	类型	String
		说明	所要监控的组合服务的 Id
输出	自定义类型 ResultRecord		

在表 12 中，自定义类型 ResultRecord 中包含了执行组合服务的输出参数列表以及该次执行的 jobId。

## 2.3 Web 应用相关接口及说明

### 2.3.1 Web 应用部署

Web 应用部署，即将用户开发的或者引用第三方的 web 应用以 .war 文件的装在到基于云计算技术的服务化软件测试平台平台中的 web server 中，使该应用可以在浏览器中被正常访问。参数列表如下。

表 13 web 应用部署

方法名	webAppDeploy		
输入参数		类型	File

	appArchive	说明	web 应用归档文件
输出	web 应用反部署报文		

如表 13 所示, web 应用部署接口 `webAppDeploy()` 提供一个参数, `appArchive`, 及用户所开发的 web 应用的 .war 文件。Web 应用部署后, 平台将为其分配唯一的标识。

### 2.3.2 Web 应用反部署

Web 应用反部署, 即将用户已经部署在平台上的 web 应用 web server 中删除。应用被反部署后, 该应用将不能在从浏览器中被访问。参数列表如下。

表 14 web 应用反部署

方法名	webAppUndeploy		
输入参数	appID	类型	String
		说明	要反部署的 web 应用的 ID
输出	服务反部署报文		

如上表所示, 原子服务反部署接口 `webAppUndeploy()` 需要输入一个参数: 需要被反部署的 web 应用的 `appId`。该 id 是原子服务在平台中的唯一标识。

## 3 总结

在服务化软件测试阶段, 由于需要相对真实的分布式测试环境, 一方面测试人员必须安装、配置、维护分布式测试环境, 工作量大; 另一方面, 测试环境的隔离特性、多变特性也要求测试环境满足动态、按需以及可信等特征。这都使得传统的服务化软件测试平台无法满足测试需求, 需要研发新的服务化软件测试平台。基于云计算技术的服务化软件测试平台提出了支持面向服务软件测试的云平台。它从测试环境的云管理、测试环境的自动构建以及在线的即时部署和测试等三个方面实现了对服务化软件测试的支持, 满足了新的测试需求。本文介绍了基于云计算技术的服务化软件测试平台向用户提供的接口以及具体的使用方法, 为用户使用平台提供的方便。