

ICS 号
中国标准文献分类号

团 体 标 准

T/JSHLW ###-####

云端融合工业互联网平台技术

Technical standards for industrial Internet platform of cloud integrated

(征求意见稿)

####-##-##发布

####-##-##实施

江苏省互联网协会 发布

目录

前 言.....	1
1 范围.....	1
2 规范性引用文件.....	1
3 术语、定义和缩略语.....	1
3.1 术语和定义.....	1
3.2 缩略语.....	2
4 接口设计原则.....	4
5 平台框架开发接口.....	4
5.1 平台基础服务模块.....	3
5.1.1 设备管理模块.....	5
5.1.2 数据管理模块.....	5
5.1.3 容器服务模块.....	5
5.1.4 用户/权限管理模块.....	6
5.1.5 安全和认证.....	6
5.2 云端融合工业互联网平台关键技术.....	6
5.2.1 面向复杂工业环境的智能感知技术.....	6
5.2.2 工业环境网络互联技术.....	8
5.2.3 工业云平台数据处理和反馈控制技术.....	8
附录 A（规范性附录）基础服务接口规范.....	11

前 言

本标准按照GB/T 1.1-2009《标准化工作导则第一部分：标准结构与编写》规则起草。

本标准规定了“感、联、知、控”的云端融合新型工业互联网平台框架模型和数据接入接口技术要求，以及对数据格式、网络协议和应用接入接口协议提出规范性要求。

本标准由江苏省互联网协会提出。

本标准由江苏省互联网协会团体标准化技术委员会归口。

本标准起草单位：东南大学、清华大学、中国科学技术大学、北京邮电大学、江苏金恒信息科技股份有限公司、中国电信股份有限公司南京分公司。

本标准主要起草人：东南大学：罗军舟、东方、熊润群、金嘉晖、张竞慧、沈典、单冯、方效林、董恺、凌振、吴巍炜、王帅、丁玎、徐祝庆、谢玮、陈慈媛；清华大学：何源、王需；中国科学技术大学：张兰、刘慧琦、耿佳宁；北京邮电大学：刘亮、郑霄龙、罗梓琿；江苏金恒信息科技股份有限公司：孙茂杰、李福存、王苏扬、杨猛、马超；中国电信股份有限公司南京分公司：陈于锋。

征求意见稿

云端融合工业互联网技术

1 范围

本标准规定了云端融合工业互联网平台框架模型、数据接入接口技术要求等内容。

本标准适用于云端融合工业互联网平台的应用开发、系统子模块的接入、数据采集和处理接口开发。

其他场合的云端融合工业互联网平台框架可参照本标准。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 8566-2007 信息技术 软件生存周期过程；

GB/T 34068-2017 物联网总体技术 智能传感器接口规范；

GB/T 26335-2010 工业企业信息化集成系统规范；

GB/T 31916.2-2018 信息技术 云数据存储和管理；

AII/001-2018 工业互联网平台 接口模型；

GB/T 19582.1-2008 基于 Modbus 协议的工业自动化网络规范 第 1 部分：Modbus 应用协议；

GB/T 19582.3-2008 基于 Modbus 协议的工业自动化网络规范 第 3 部分：Modbus 协议在 TCP/IP 上的实现指南；

GB/T 33137-2016 基于传感器的产品监测软件集成接口规范；

GB/T 34068-2017 物联网总体技术 智能传感器接口规范；

GB/T 34069-2017 物联网总体技术 智能传感器特性与分类；

GB/T 28181-2016 标准公共安全视频监控联网系统信息传输、交换、控制技术要求。

3 术语、定义和缩略语

3.1 术语和定义

GB/T 8566-2007、GB/T 34068-2017、GB/T 26335-2010、GB/T 31916.2-2018 以及 AII/001-2018、GB/T 19582.1-2008、GB/T 19582.3-2008、GB/T 33137-2016、GB/T 34068-2017、GB/T 34069-2017、GB/T 28181-2016 界定的以及下列术语和定义适用于本文件。

3.1.1

工业互联网平台 industrial internet platform

工业互联网平台是面向工业数字化、网络化、智能化需求，构建基于数据采集、汇聚、分析的服务体系，支撑工业资源在连接、弹性供给、高效配置的工业云平台。

3.1.2

传感设备 sensor device

连接传感器，具有通信接口的数据采集硬件设备。

3.1.3

采集模块 acquisition module

从传感设备通信接口获取信息的计算机程序。

3.1.4

表述性状态转移 representational state transfer

一种针对网络应用的设计和开发方式，指的是一组架构约束条件和原则，可以降低开发的复杂性，提高系统的可伸缩性。

3.1.5

远程过程调用 Remote Procedure Call

一种通过网络从远程计算机程序上请求服务，通过接口调用远程服务进程中函数的方式。

3.1.6

容器镜像 Container image

虚拟集群中，应用使用的容器镜像名称。

3.1.7

角色身份 Principals

即主体的标识属性，可以是任何属性，用户名、邮箱等。

3.1.8

证明/凭证 Credentials

只有主体知道的安全值，密码/数字证书等。

3.1.9

工业串行通信协议 Modbus TCP/IP

运行在 TCP/IP 协议之上的 Modbus 工业串行通信协议。

3.1.10

Mesh 节点 Mesh Node

无线网状网节点。

3.2 缩略语

下列缩略语适用于本文件

HTTP，超文本传输协议(HyperText Transfer Protocol)

REST，表述性状态转移(Representational State Transfer)

RPC，远程过程调用 (Remote Procedure Cal)

URI，统一资源标识(Uniform Resource Identifier)

JSON JavaScript, 对象简谱格式 (JavaScript Object Notation)

XML, 统一资源标识(Uniform Resource Identifier)

SSL, 安全套接层 (Secure Sockets Layer)

TSL, 传输层安全 (Transport Layer Security)

PLC, 可编程逻辑控制器 (Programmable Logic Controller)

NIC, 网络接口控制器 (Network Interface Controller)

mmWave 雷达, 毫米波雷达 (Millimeter-Wave Radar)

IPC, 网络摄像机 (IP Camera)

RTP, 实时传输协议 (Real-time Transport Protocol)

RTCP, 实时传输控制协议 (Real-time Transport Control Protocol)

4 接口设计原则

1) 松耦合原则, 包括: a) 接口方式与信息模型的松耦合, 无论采取何种接口方式或技术, 其交互的信息应遵循统一的信息模型; b) 工业互联网服务与外部资源的松耦合, 通过接口的信息交互, 工业互联网服务与外部资源之间不应存在强依赖的关系。

2) 可靠性原则, 包括: a) 无论采取何种接口方式, 都应保证所传递的信息是可靠、完整和一致的。接口可靠性不仅要求交互方式的可靠与稳定, 还要求接口的实现不能对参与接口的系统的可靠性造成任何不良影响, 如当采集链路或平台功能组件异常时, 不应造成数据丢失以及接口相关系统工作受影响; b) 当接口失效时, 应当在一定时间段内保持传输信息不丢失; c) 接口在异常发生后, 具有一定的自我恢复能力。

3) 安全性原则, 包括: a) 工业 APP 与外部资源通过接口交互的信息应当考虑相关安全措施, 如采用加密传输等方式完成信息交互。

4) 高效性原则, 包括: a) 采用的接口方式与实现技术应当保证接口的高效传输; b) 应当采取措施减少待交互信息的积压, 降低延迟时间。

5) 扩展性原则, 包括: a) 平台组件功能的可扩展性:接口的定义不应限制工业 APP 的功能实现, 并且应保证在将来工业互联网服务组件功能发生改变时(增加或调整),接口方式应能继续提供支持; b) 外部资源的可扩展性:当外部资源的种类和具体指标发生改变时, 接口应能很好地支持这些变化。

6) 成熟性原则, 包括: a) 接口方式与实现宜采用成熟的先进技术, 使系统得到较好的投资保护。



图 1 云端融合工业互联网平台架构图

5 平台框架开发接口

本标准定义了工业应用接入云端融合工业互联网平台的接口标准和方法流程, 设计了系统子模块的接口标准和框架标准, 包括基础服务标准和关键技术标准。

5.1 平台基础服务模块

平台基础服务模块为云平台提供基本的系统支撑保障, 为各类工业应用接入提供设备、数据、服务和安全访问管理等的基础功能。基础框架目前采用 Spring 架构但不限于此架构, 其亦可扩展到 Spring Boot、Spring Cloud 等分布式服务框架。基础服务主要由设备管理服务、服务接口规范、数据管理、容器服务、权限管理服务组成, 其技术标准如下。

5.1.1 设备管理模块

设备管理模块负责工业互联网平台用户设备的管理, 支持用户自定义的设备注册认证、设备信息增删改、设备状态转变和多功能视角列表功能, 并且, 提供基于多层级用户角色权限的设备管理功能, 主要包括: 权限验证, 分组统计, 设备操作审核, 运行维护记录查询, 可视化实时监控等功能。其服务层从 MES 系统接口获取数据提供给应用层使用和管理。

5.1.2 数据管理模块

数据服务的主要功能是管理工业互联网平台中的各类数据。通过建立数据统一访问接口, 屏蔽底层异构的数据格式和数据存储方式, 使得数据访问操作统一化, 从而提高平台效率和兼容性。

5.1.3 容器服务模块

面向工业互联网应用实现服务管理功能，包括：定制服务编排，提供微服务级的业务抽象，基于容器弹性支撑上层业务；全局资源总览，对平台资源进行可视化管理，形成资源运行报表，提供全局的资源视图；实时服务监控，对于平台上运行的工业互联网应用，提供微服务级的实时监控。其服务层对接 Kubernetes 等容器应用接口，实现对工业应用的容器管理。

5.1.4 用户/权限管理模块

用户管理模块负责工业互联网平台用户相关管理，支持用户创建、用户审核、角色创建、角色权限分配、用户角色分配，用户信息更改及用户相关信息展示等基础功能，并且，通过角色访问控制技术实现用户对资源的限制访问。

用户管理服务中，用户相关的用户审核注册、用户角色分配、角色权限分配须符合 RBAC1 规范：用户管理系统对象主要包含资源、权限、用户、角色，主要操作为：角色权限分配、用户角色分配。

权限为访问资源与访问策略的绑定，角色权限分配将权限与角色形成映射，用户角色分配将用户与角色进行关联，从而实现用户对资源的控制访问。

5.1.5 安全和认证

5.1.5.1 安全类接口

安全类应包含身份认证类、权限管理类、访问控制类、密钥管理类、数据加解密类等类型接口；其中：

身份认证类至少包含：获取认证信息、上传认证信息等接口。

权限管理类至少包含：获取权限、授予权限、删除权限等接口。

访问控制类至少包含：获取权限资源等接口。

密钥管理类至少包含：密钥上传、密钥删除等接口。

数据加解密类至少包含：数据加密、数据解密等接口。

接口支持包含但不限于 SSL、TSL、SHA2、对称加密算法和方式。

5.1.5.2 设备/产品认证接口

设备/产品认证类等类型接口；其中：

设备/产品认证类至少包含：获取认证信息、删除认证信息、验证认证信息等接口。

5.2 云端融合工业互联网平台关键技术

在综合考虑云端融合工业互联网平台基础服务的基础上，按照智感、效联、迅知谐控三层构架原则，本标准定义了平台关键技术标准和方法流程。

5.2.1 面向复杂工业环境的智能感知技术

5.2.1.1 工业场景传统传感器感知技术

本平台对传统传感器信息接口的对接、信息获取和联网规则，可使用传统工业传感器技术标准以及工业物联相关行业的传感器技术规范和标准，做到尽可能统一使用目前现有工业场景的传感器通用技术标准。

5.2.1.2 非传感器感知技术规范

针对工业场景中环境复杂多变、人员流动性强、干扰严重等挑战，创新性提出非传感器感知技术，设计技术规范，方便应用层服务调用其数据和接口。

5.2.1.2.1 工业现场振动监测技术

非接触式毫米波测振通过发送毫米波频段射频信号，并接收其视场角范围内振动物体反射的毫米波信号，从接收的毫米波信号中提取振动物体的振动信号，并将原始振动信号及频谱分析等结果通过有线或无线网络发送给远程控制台。该解决方案具有非接触式测量、多点监测两大优势，能够为工业现场提供高精度的振动检测服务。

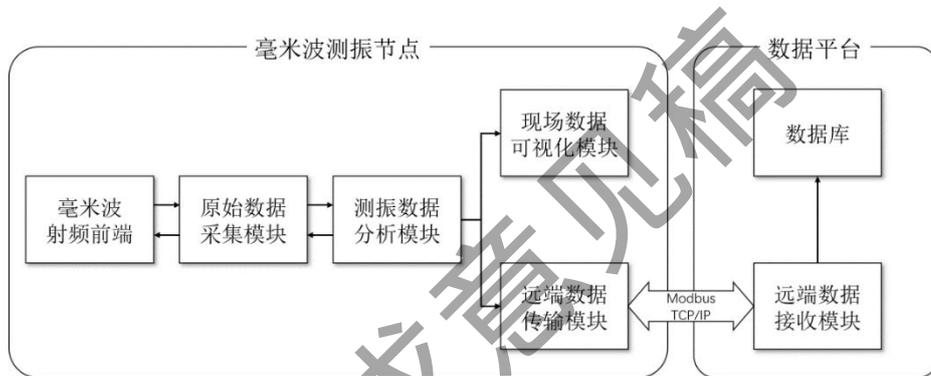


图 2 毫米波测振技术路线图

5.2.1.2.2 工业环境多媒体感知技术

工业环境多媒体感知技术通过在工厂环境中部署监控摄像头等多媒体感知设备、无线 Mesh 网络等联网设备和智能视频分析服务器，实现对大规模工业监控视频的感知、传输、分析等功能，为不同的工业应用提供实时语义化的多媒体感知信息。

5.2.1.2.3 工业环境非侵入式传感技术

此标准旨在针对工业生产中人员活动的复杂性和粒度差异，提出一种与活动相关、位置无关的普适特征提取方法；针对工业生产中复杂的人机交互需求，提出一种基于微多普勒效应实现高精度、细粒度的手势识别方法，以及基于信号周期变化特性的多人协同工作评估方法。

5.2.1.3 平台设备/产品标识

工业设备、传感器和网络设备等标识和设备参数应符合工业互联网产业联盟的工业互联网标准体系 2.0 和标识解析体系等工业行业规范的要求。传感器接口支持包括 IEEE1451 协议族等。采集数据应支持包含但不限于 XML、JSON、Protocol Buffer、二进制等数据编码格式。

标识类应包含设备/产品标识类、网关标识类、数据采集点标识类等类型接口；其中：设备/产品标识类至少包含：获取设备/产品信息等接口。网关标识类至少包含：获取网关信息等接口。采集点标识类至少包含：获取设备/产品数据采集点信息、获取网关数据采集点信息等接口。

5.2.2 工业环境网络互联技术

5.2.2.1 平台效联模式

工业网络各层的数据通过效连层技术来汇集与互联，通过以太网网关、网关服务器中间件，把企业级网络与云平台汇集系统相连接，在软件侧部署中间件 MQ 消息处理，为系统应用 REST/RPC 等提供服务，提高异构系统间的兼容性。

5.2.2.2 平台接口模块

平台接口模块可分为以下几类：平台网络接口、平台无线通信接口。

数据网关和边缘计算网关应支持包含但不限于 http、TCP/IP、AMQP、MQTT、UDP 等网络协议。使用软件定义网络技术（OpenVSwitch、OpenFlow）为上层计算机点搭建大二层虚拟网络平台，通过下发 OpenFlow 流表为上层提供路由管理服务，ARP 服务，通过 IP 地址管理服务。为计算单元的动态创建、迁移、销毁提供对应的网络配置，最终达到动态传算的目的。

异构网关涵盖 USB 等常见硬件通讯端口，支持 WIFI, ZigBee, Bluetooth, Lora 这些通讯方式，兼容 modbus、http、https、mqtt 和 Zeromq 等总线及网络传输协议。预留 mini-PCIe 扩展接口，可对采集模块、通讯模块、存储模块和计算模块进行扩展。

5.2.3 工业云平台数据处理和反馈控制技术

云端融合工业互联网平台通过“讯知谐控”层来进行数据处理，提供应用服务间和子系统模块间的对接。

5.2.3.1 工业云平台数据处理接口

云端融合工业互联网平台包含下列几类接口：

1) **数据预处理接口**：工业互联网平台采用 python 中的 pandas、numpy 库协助进行数据获取与预处理，支持 centOS、Ubuntu、Windows 多种操作系统的运行环境，可兼容 Tensorflow、Pytorch 框架的调用。

2) **服务集合构建接口**：平台的工业互联网服务采用 Docker 封装，支持 centOS、Ubuntu、Windows、fedora、alpine 等多种操作系统的运行环境，支持 HTTP、HTTPS、TCP/IP、MQTT 等网络通信协议，兼容 Docker compose、Docker swarm、Kubernetes、Dubbo 等服务框架的调用。

3) **应用分析模型类服务接口**：此接口可对工业应用中处理后的数据进行建模，支持工

业数据的预测。

4) 应用服务请求和路由规范：服务请求和路由规范定义了一种通用的接口服务发现机制,支持分布式工业应用和微服务接口,服务请求和路由包含但不限于 a)服务提供者列表(应用列表) ServiceProviderCatalog、b)服务提供者(应用) ServiceProvider、和 c)服务 Service 三种基本服务请求, 支持 RESTful API、RPC 等规范。

5) 云分布式服务接口

接入的应用服务结构应支持微服务、数据流量负载均衡框架,兼容 Spring、Spring Cloud、Ngnix、RocketMQ/Kafka、DB2/Mysql/Oracle/NoSql 类等服务器集群框架。支持 docker、kubenetes 容器服务框架和接口。

征求意见稿

附录 A 基础服务接口规范

A.1 设备管理服务接口

1. 设备操作审核

接口路由地址	/DeviceController/reviewPass
--------	------------------------------

接口功能	在数据库中修改设备操作记录的审核信息
参数	Ids(string):设备id ApproveStatus(string):设备状态 OperatingSort(string):操作类别 RefuseReason(string):审核不通过原因
返回	MessageString(string):设备审核状态反馈信息

2. 设备注册审核

接口路由地址	/DeviceController/registerReview
接口功能	在数据库中修改设备注册操作的审核信息
参数	ApproveStatus(string):设备状态 Ids(string):设备id Operating(string):操作类型 RefuseReason(string):审核不通过原因
返回	无

3. 设备信息修改审核

接口路由地址	/DeviceController/modifyReview
接口功能	在数据库中修改设备信息修改操作的审核信息
参数	ApproveStatus(string):设备状态 Ids(string):设备id Operating(string):操作类型 RefuseReason(string):审核不通过原因
返回	无

4. 按功能获取设备信息

接口路由地址	/DeviceController/equipment
接口功能	获取按功能划分的设备页面
参数	ListId(integer):列表 ID; date(integer):日期
返回	EQUIPMENT(JSP):设备功能视角页面

5. 获取设备厂区分布信息

接口路由地址	/MaintenanceController/deviceMapList
接口功能	获取设备在各厂区的分布数据

参数	无
返回	MAINTENANCEOVERVIEW(JSP):设备厂区地图

6. 返回设备厂区页面

接口路由地址	/MaintenanceController/regionDeviceInfoList
接口功能	返回设备厂区页面
参数	无
返回	REGIONDEVICEINFOLIST(JSP):厂区地图页面

7. 获取设备维护信息

接口路由地址	/MaintenanceController/malfunctionInfoList
接口功能	获取设备维护列表
参数	Malfunction(Object):设备维护记录实体 RegionId(string):厂区 id
返回	MALFUNCTIONINFO(JSP):厂区内设备维护记录

8. 获取设备故障信息

接口路由地址	/MaintenanceController/overhaulInfoList
接口功能	获取设备故障信息列表
参数	Overhal(Object):设备故障记录实体 RegionId(string):厂区 id
返回	OVERHAULINFO(JSP):厂区内设备故障记录

9. 获取设备消息

接口路由地址	/DeviceController/message
接口功能	获取设备消息弹窗
参数	id(string):设备 id
返回	MESSAGE(JSP):设备消息弹窗

10. 设置设备消息已读

接口路由地址	/DeviceController/alreadyRead
接口功能	设置设备消息已读
参数	id(string):设备 id
返回	无

11. 获取设备类型数据

接口路由地址	/DeviceController/persconv
接口功能	获取设备类型数据
参数	无
返回	PERSCONV(JSP):设备类型数据

12. 获取设备详情列表

接口路由地址	/DeviceController/mydetail
接口功能	获取设备详情列表
参数	Device(Object):设备实体
返回	MYDETAIL(JSP):设备详情列表

13. 获取设备新增页面

接口路由地址	/DeviceController/preCreate
接口功能	获取设备新增页面
参数	无
返回	CREATE(JSP):设备新增页面

14. 获取设备批量导入页面

接口路由地址	/DeviceController/import
接口功能	获取设备批量导入页面
参数	无
返回	IMPORT(JSP):设备批量导入页面

15. 完成设备批量导入功能

接口路由地址	/DeviceController/upload
接口功能	完成设备批量导入功能
参数	eqSort(String):设备类型
返回	无

16. 判定设备编号是否重复

接口路由地址	/DeviceController/eqNoIsRepeat
接口功能	判定设备编号是否重复
参数	Device(Object):设备实体 eqSort(String):设备类型
返回	True/False

17. 按设备分类编码获取设备分类信息接口

接口路由地址	/device/chooseDeviceSort
接口功能	选择设备分类
参数	model (Model): 模型, eqSort (String): 设备分类编码
返回	CREATEPRODUCTDEVICE (device/createProductDevice): 设备分类中文

18. 按设备注册信息返回设备审核结果接口

接口路由地址	/device/createDevice
接口功能	注册设备审核
参数	model (Model): 模型, deviceReview (DeviceReview): 设备审核
返回	ajaxObject.toString(): 设备注册审核结果信息

19. 按设备 ID 信息返回设备停用审核结果接口

接口路由地址	/device/unUsed/{id}
接口功能	停用设备审核
参数	ids (String): 设备 ID
返回	ajaxObject.toString(): 停用设备审核结果信息

20. 按设备 ID 信息返回设备投用审核结果接口

接口路由地址	/device/used/{id}
接口功能	投用设备审核
参数	ids (String): 设备 ID
返回	ajaxObject.toString(): 投用设备审核结果信息

21. 按设备 ID 信息和 http 请求返回设备预更新结果接口

接口路由地址	/device/preUpdate/{id}
接口功能	返回设备更新的预结果页面
参数	ids (String): 设备 ID, request (HttpServletRequest): http 请求
返回	INFORMATION ("device/deviceInfo"): 设备信息

22. 将 DB2 数据库读取的字符串日期改为标准格式接口

接口路由地址	None
接口功能	将字符串转化为日期格式

参数	dateFromDB2 (String): 来自 DB2 的日期
返回	datetime: 标准日期格式

23. 按设备审核信息和 http 请求返回设备更新结果接口

接口路由地址	/device/update
接口功能	返回设备更新结果
参数	deviceReview, request: 设备审核信息和 http 请求
返回	ajaxObject.toString(): 设备审核更新信息

A.2 数据管理服务接口

1. 根据条件查询数据源

接口路由地址	/querySource/param
接口功能	获取对应数据源信息
参数	param (string): 数据源查询条件;
返回	ListOfSource(Serialized Object): 数据源信息列表

2. 新增数据源信息

接口路由地址	/addSource/param
接口功能	新增数据源信息
参数	param (Serialized Object): 数据源信息;
返回	result(Boolean): 数据源新增成功与否

3. 新增数据源信息

接口路由地址	/alterSource/param
接口功能	修改数据源信息
参数	param (Serialized Object): 数据源信息;
返回	result(Boolean): 数据源修改成功与否

4. 删除数据源信息

接口路由地址	/sourceName/deleteSource
接口功能	删除数据源信息
参数	sourceName (String): 数据源名称;
返回	result(Boolean): 数据源信息删除成功与否

5. 获取数据源中接口信息

接口路由地址	/sourceName/getAllDataModel
--------	-----------------------------

接口功能	获取数据源中的所有接口信息
参数	SourceName(string):数据源名称;
返回	ListOfDataModel(Serialized Object):数据接口列表

6. 数据源数据表摘要信息展示

接口路由地址	/sourceName/tableName/getDigest
接口功能	获取数据源数据表摘要信息
参数	sourceName (String):数据源名称; tableName(String): 表名称
返回	ContentOfTableDigest(Serialized Object):数据表摘要信息

7. 数据源数据表信息预览

接口路由地址	/sourceName/tableName/preView
接口功能	数据源数据表信息预览
参数	sourceName (String):数据源名称; tableName(String): 表名称
返回	ContentOfTableDigest(Serialized Object):数据表信息

8. 操作指定数据模板

接口路由地址	/dataModelID/executeModel/param
接口功能	执行特定数据模板
参数	dataModelID (Integer):数据接口 id; param(JSON):数据接口参数
返回	ModelInfo(Serialized Object):数据接口执行结果

A.3 容器服务接口

A.3.1 状态监控部分接口:

1. 资源总览信息获取接口

接口路由地址	/serviceOverview/getPie
接口功能	获取资源总览信息
参数	无
返回	rslt(List<String>): 获取的资源总览信息

2. 负载信息获取接口

接口路由地址	/serviceOverview/getCPUload
接口功能	获取节点 CPU 和内存负载信息
参数	无
返回	rslt(List<String>): 获取的节点 CPU 以及内存信息

3. 微服务信息获取接口

接口路由地址	/serviceOverview/getPods
接口功能	获取平台微服务信息
参数	无
返回	rslt(List<String>): 获取的平台微服务信息

4. 容器负载信息获取接口

接口路由地址	/serviceOverview/getPodsCPU
接口功能	获取容器 CPU 和内存负载信息
参数	无
返回	rslt(List<String>): 获取的容器 CPU 和内存负载信息

5. 应用负载获取接口

接口路由地址	/statusMonitor/getCPULoad
接口功能	获取应用的 CPU 以及内存负载信息
参数	deployTemp(List<Map<String, Object>>): 应用部署列表
返回	rslt(List<Map<String, Object>>): 获取的应用 CPU 以及内存负载信息

6. 应用微服务状态获取接口

接口路由地址	/statusMonitor/getPodStatus
接口功能	获取应用部署的微服务状态
参数	deployTemp(List<Map<String, Object>>): 应用部署列表
返回	rslt(List<Map<String, Object>>): 获取应用微服务状态

7. 应用日志获取接口

接口路由地址	/statusMonitor/getLog
接口功能	获取应用的日志信息
参数	deployTemp(List<Map<String, Object>>): 应用部署列表
返回	rslt(List<Map<String, Object>>): 获取的应用日志信息

8. 应用弹性伸缩状态获取接口

接口路由地址	/statusMonitor/getSerDeploy
接口功能	获取应用资源弹性伸缩状态
参数	deployTemp(List<Map<String, Object>>) : 应用部署列表 ;

	serviceTemp(List<Map<String, Object>>): 应用服务列表
返回	rslt(List<Map<String, Object>>): 获取的应用资源弹性伸缩状态列表

A.3.2 服务编排接口:

1. freemarker 接口

接口路由地址	/freemarker/template
接口功能	通过模板生成 yaml 文件
参数	deploymentMap(HashMap) : deployment 属性集 ; deploymentTemplate(template): 模板文件位置
返回	fw(FileWriter): yaml 文件

2. Service 数据库存取接口

接口路由地址	/serviceArrange/preCreateService
接口功能	数据库新增 Service 信息
参数	nodeName(String) : 拓扑节点标识 ; type(String) : 类型 ; relativeID(String): 关联 ID
返回	res(int): 数据库存取结果状态码

3. ServicePort 数据库存取接口

接口路由地址	/serviceArrange/insertServicePort
接口功能	数据库新增 ServicePort 信息
参数	serviceports(servicePorts): 网络服务端口实体类
返回	res(int): 数据库存取结果状态码

4. Deployment 数据库存取接口

接口路由地址	/serviceArrange/preCreateDeployment
接口功能	数据库存取 Deployment 数据
参数	nodeName(String) : 拓扑节点标识 ; type(String) : 类型 ; relativeID(String): 关联 ID; deployment(Deployment): 应用实体类
返回	res(int): 数据库存取结果状态码

5. Pod 数据库存取接口

接口路由地址	/serviceArrange/podService
接口功能	数据库存取 Pod 数据
参数	pod(Pod): Pod 实体类

返回	res(int): 数据库存取结果状态码
----	----------------------

6. 容器数据库存取接口

接口路由地址	/serviceArrange/containerService
接口功能	数据库存取容器数据
参数	container(Container): 容器实体类
返回	res(int): 数据库存取结果状态码

7. 容器端口数据库存取接口

接口路由地址	/serviceArrange/containerPortService
接口功能	数据库存取容器端口数据
参数	containerPort(ContainerPort): 容器端口实体类
返回	res(int): 数据库存取结果状态码

8. 容器环境变量数据库存取接口

接口路由地址	/serviceArrange/environmentService
接口功能	数据库存取容器环境变量数据
参数	environment(Environment): 容器环境变量实体类
返回	res(int): 数据库存取结果状态码

A.4 用户信息管理接口

接口路由地址	/UserController/User
接口功能	在系统中保存用户信息
参数	strName (string):用户名称 strPassword (string):用户密码 strEmail (string):用户邮件 userRoles(string):用户角色
返回	无