

ICS 35.240.50

CCS J 07



**NATIONAL STANDARD OF THE
PEOPLE'S REPUBLIC OF CHINA**

GB/T 43960-2024

**Open application program interface requirements for cloud
manufacturing service platform**

云制造服务平台开放接口要求

Issued on: April 25, 2024

Implemented on: November 1, 2024

**Issued by: State Administration for Market Regulation;
Standardization Administration of the PRC**

Table of Contents

1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	1
5 Open interface model	2
6 Interface types	3
7 Requirements for writing interfaces	9
8 Requirements for designing interfaces	10
9 Requirements for registering and calling interfaces	10
10 Packet format	11
Annex A (informative) Examples of open interfaces of a cloud manufacturing service platform	13

3 Terms and definitions

The terms defined in GB/T 29826-2013 apply, together with three terms defined here.

3.1 open application program interface: the open application programming interface that a cloud manufacturing service platform offers to the outside.

3.2 application program interface for industrial application: the open application programming interface offered on a cloud manufacturing service platform to support the development and management of industrial applications.

3.3 application program interface for manufacturing resource/capability import: the open application programming interface offered on a cloud manufacturing service platform to support data transmission, resource management and identifier management when cloud manufacturing resources and capabilities are brought onto the platform.

5 Open interface model

5.1 General. The model architecture is shown in Figure 1 and holds two families of open interface: industrial application service interfaces, and manufacturing resource and capability access interfaces. Manufacturing resource and capability access shall comply with Clauses 6 to 8 of GB/T 39471-2020. Interface security shall comply with OWASP and with the relevant provisions of GB/T 39403-2020. Both families may use transport protocols such as HTTP and HTTPS and should follow the OpenAPI Specification, service level

agreements and similar conventions.

5.2 Industrial application service interfaces are divided into industrial application development interfaces and industrial application management interfaces. The development interfaces cover data interfaces, model interfaces and micro-service interfaces, and shall make clear the data, models and micro-services that industrial application development calls. The management interfaces shall include application management interfaces and security interfaces, and shall provide interfaces for application deployment, monitoring management, identity authentication, access control and permission management.

5.3 Manufacturing resource and capability access interfaces are divided into access service interfaces and access management interfaces. The access service interfaces shall include device data interfaces, device control interfaces, system data interfaces and access security interfaces, and shall make clear the interfaces for data exchange between industrial equipment or enterprise information systems and the platform, for issuing device control commands, and for securing access. The access management interfaces shall include device management interfaces, edge management interfaces and identifier interfaces, providing for the management and identification of cloud manufacturing resources such as industrial equipment and edge gateways.

6 Interface types

6.1 The open interfaces of a cloud manufacturing service platform are mainly of seven kinds: data, model, micro-service, security, identifier, management and control. Examples are given in Annex A.

Each interface in Clause 6 is presented in a table with five fields: interface name, interface description, interface method, parameter description and return value. The parameter lists are stated as minimum sets, and each named parameter is given with its data type in the printed text.

6.2.1 Data interfaces. The data retrieval interface shall support queries on the state of a data resource; it uses POST, takes at least the data resource ID, enterprise ID, user ID and an authentication parameter, and returns data resource information such as data volume, data update time and data permissions. The data analysis interface shall support calling a data analysis task and obtaining its result; it uses POST, takes at least the enterprise ID, user ID, authentication parameter and data analysis task ID, and returns the analysis result.

6.2.2 Model interfaces. The model registration interface shall let models of different kinds be registered on the platform; it uses POST, takes at least the user ID, authentication parameter, model name, model description and model class, and returns whether registration succeeded. The model status query interface shall let the running state of models of different kinds be queried; it uses GET, takes at least the user ID, authentication parameter and model ID, and returns the running state data of the model.

6.2.3 Micro-service interfaces. The micro-service registration interface shall publish micro-service information; it uses POST, takes at least the user ID, authentication parameter, micro-service name, micro-service description and data centre address, and returns the micro-service information. The micro-service status query interface shall query the running state; it uses GET, takes at least the user ID, authentication parameter, micro-service ID, task ID and the user account that started the micro-service, and returns the running state data.

6.2.4 Application management interfaces. The application deployment and release interface shall support creating and releasing an application; it uses POST, takes at least the user ID, authentication parameter, application ID, application name and application version, and returns the application ID together with a release status value. The application health monitoring interface shall obtain the running state of an application; it uses GET, takes at least the user ID, authentication parameter and application ID, and returns whether the application is running normally.

6.2.5 Application security interfaces shall support identity authentication of application management users and the obtaining of access permissions. The application security authentication interface uses GET, takes at least the user ID, authentication parameter and application ID, and returns the authorisation status value of the application.

6.3 Manufacturing resource and capability access interfaces. The device status interface shall query the current state of a device, which is one of running, standby, fault or offline; it uses GET, takes at least the user ID, authentication parameter and device ID, and returns whether the call succeeded. The device data interface shall collect device running data and obtain the list of data points; it uses POST, takes at least the user ID, authentication parameter, gateway ID, device ID and a data key-value pair, and returns whether the call succeeded. The device control interface shall issue device control commands; it uses POST, takes at least the user ID, authentication parameter, device ID and the command content, and returns whether the command was sent and what the result of execution was.

The system data interfaces shall support the connection of enterprise information systems and data transmission between the platform and the connected system, and shall cover interfaces for exchange with ERP, PLM and MES systems; the worked example is a production material form saving interface, which uses POST, takes at least the user ID, authentication parameter and form name, and returns whether the call succeeded. The access security interfaces authenticate the identity of connected resources; the device security authentication interface uses POST, takes at least the user ID, authentication parameter, device ID and device certificate, and returns whether authentication succeeded.

The device management interfaces shall support creating, querying, grouping, editing, deleting, importing and exporting devices and managing device templates; the device query interface uses GET, takes at least the user ID, authentication parameter and device ID, and returns whether the call succeeded. The edge management interfaces shall support edge

gateway management, cloud-edge coordination node management and edge application management; the edge gateway creation interface uses POST, takes at least the user ID, authentication parameter, gateway name, communication protocol code, communication protocol address and gateway type code, and returns whether the call succeeded. The identifier interfaces cover a device identifier saving interface and a gateway identifier saving interface, both using POST and both taking, in addition to the user ID and authentication parameter, the device or gateway ID, the identifier class ID, the identifier template version and the identifier template attributes.

7 Requirements for writing interfaces

7.1 Practicality. The data format shall use a cross-platform and general-purpose file format, for example JSON or ProtoBuf; XML and TXT may also be used as an aid during development. The data volume shall be allocated as needed.

7.2 Usability. Interface and parameter names shall be accurate, shall carry the necessary content and shall be easy to understand. For both successful and failed transmission, the interface shall give clear data status information and a return value. The interface shall be extensible so that functions can be adjusted or added.

7.3 Security. An agreed algorithm shall be used to validate and match the parameter values passed. When a user name and password are passed, a proper encryption algorithm shall be used for the data communication request, and domestic encryption algorithms are recommended. Interfaces shall be version controlled, the interface version shall be updated when the platform or the software and hardware are upgraded, and the status of new and old interface versions should be dealt with periodically.

8 Requirements for designing interfaces

8.1 General. Interface design shall keep parameter passing to a minimum and pass parameters according to the request. It shall avoid duplicate interfaces, raise interface reuse, cut the number of HTTP requests and improve program stability.

8.2 Data type conventions. The data type of each parameter shall be annotated, together with whether it may be empty and whether it is a default field. The data type of a parameter shall match the corresponding return data type; mismatched parameter fields readily cause data type parsing exceptions in the application.

8.3 Encoding conventions. The encoding of the interface shall be annotated during development, and UTF-8 should be used.

8.4 Request method. The request method shall be annotated when the interface is written; the usual methods are GET, POST, PUT and DELETE. GET is used for small amounts of data, and POST shall be used when the data volume exceeds 1024 bytes, to avoid failed or abnormal requests.