

Kiloview AVX24-4 Media HUB

WEB API Documentation

Document Version: v1.0

Applicable Product: AVX24-4 Media HUB

Publisher: Changsha Kiloview Electronic Technology Co., Ltd.

Document Description

This document is the HTTP API interface specification for AVX24-4 Media HUB, aiming to provide complete backend interface descriptions covering core functional modules such as user authentication, system management, network configuration, panel control, switch management, and diagnostic alarms. HTTP routes define request methods and paths for each operation, serving system integration and remote management.

Key points include:

- **User Authentication and Session Management**
Provides login, logout, user CRUD, and session validation interfaces, supporting token-based authentication mechanism.
- **System Operation and Firmware Upgrade**
Includes system-level operation interfaces such as firmware upload, fragment update, upgrade status query, log retrieval, reboot, and factory reset.
- **Panel and I/O Control**
Supports panel CRUD, input source discovery and management, output configuration, and cross-switching, enabling flexible scheduling of audio and video signals.
- **Network and VLAN Configuration Management**
Provides functional interfaces for VLAN settings, DHCP management, static routing, PTP clock, LLDP topology update, and KDDS network diagnostics.
- **Alarm Events and Switch Deep Control**
Implements alarm query, configuration, and export, as well as fine-grained management of switch ports, PoE, aggregation, VLAN, and power status.

Version Note: Due to continuous product iteration and upgrade, the product you purchased may differ from the description in this manual. In case of inconsistency, the physical product and accompanying instructions in the packaging shall prevail.

Legal Notice

By receiving this document from Changsha Kiloview Electronic Technology Co., Ltd. (hereinafter referred to as "Kiloview"), you are deemed to have agreed to the following terms. If you do not agree to the following terms, please stop using this document.

This document is copyrighted by Changsha Kiloview Electronic Technology Co., Ltd., and reserves any rights not expressly granted in this document. The document contains proprietary information of Kiloview. Without prior written permission from Kiloview, no unit or individual may copy, transmit, distribute, use, or disclose this document or any pictures, tables, data, or other information contained herein.

Kiloview's registered trademarks, product names, and logos are owned by Kiloview. Other product or company names mentioned in this document may be trademarks or registered trademarks of their respective owners. Reading this document does not imply, by estoppel or otherwise, grant the reader any right to use any marks appearing in this document without the prior written consent of Kiloview or the third-party rights holder.

This product meets relevant design requirements for environmental protection and personal safety. Storage, use, and disposal of the product should comply with the product manual, relevant contracts, or the laws and regulations of the country/region where it is located.

This document is provided "as is" and "as available". The information in this document will be continuously updated with the advancement of Kiloview products and technology, and Kiloview will no longer separately notify such updates.

For matters not covered in this document, please visit Kiloview's official website www.kiloview.com for relevant information and technical support.

目录

1 User Authentication	1
1.1 User Management / Users	1
1.1.1 User Login	1
1.1.2 Delete User	2
1.1.3 Add User	3
1.1.4 Modify User	4
1.1.5 Get All User List	6
2 System Management	7
2.1 System Settings	7
2.1.1 Set System Name	7
2.1.2 Restore Factory Settings	8
2.1.3 Get System Log	9
2.1.4 Reboot System	10
2.1.5 Firmware Upload	11
2.1.6 Firmware Upgrade	12
2.1.7 Get Runtime Log	14
2.1.8 Export Operation Log	15
2.2 System Time	16
2.2.1 Get System Time and Timezone	16
2.2.2 Set Timezone	17
2.2.3 Set System Time	18
2.3 PTP Clock	20
2.3.1 Query PTP Clock Status	20
2.3.2 Save PTP Clock Start/Stop	22
2.3.3 Query PTP Clock Details	24
2.3.4 Modify PTP Clock Configuration	25
3 Access Policy / Preset Configuration	28
3.1 Access Policy	28
3.1.1 Get Backup List	28
3.1.2 Save Backup	29
3.1.3 Read Backup	30
3.1.4 Delete Backup	31
3.1.5 Modify Backup Name	33
3.2 Virtual NIC Interface	34
3.2.1 Query Virtual Interface Configuration for Specified VLAN	34
3.2.2 Query All Virtual NIC Interface IP List	36
3.2.3 Set Virtual NIC Interface Configuration for Specified VLAN	37
3.2.4 Query All VLAN Virtual Interface IP Information	39

3.3 Mode Switch	40
3.3.1 Query Switch Layer 2/Layer 3 Mode	40
3.3.2 Set Switch Layer 2/Layer 3 Mode	41
4 Panel Management	42
4.1 Input Source Discovery	42
4.1.1 Add Server Discovery	42
4.1.2 Add Manual Discovery	43
4.1.3 Add Auto Discovery	45
4.1.4 Server Discovery Query	46
4.1.5 Manual Discovery Query	48
4.1.6 Auto Discovery Query	50
4.1.7 Delete Discovery Item	52
4.1.8 Get All Discovery Items	53
4.2 Input Management	55
4.2.1 Modify Input Source	55
4.3 Output Management	56
4.3.1 Add Output	56
4.3.2 Delete Output	57
4.3.3 Modify Output	59
4.3.4 Query Output	60
4.3.5 Get All Outputs	62
4.3.6 Get Default Output	65
4.4 Panel Basic Operations	66
4.4.1 Add Panel	66
4.4.2 Delete Panel	67
4.4.3 Modify Panel	68
4.4.4 Expand Panel	69
4.4.5 Query Panel	70
4.4.6 Get All Panels	71
4.5 Panel Input Management	73
4.5.1 Add Panel Input	73
4.5.2 Remove Panel Input	75
4.5.3 Sort Panel Inputs	76
4.6 Panel Output Management	77
4.6.1 Add Panel Output	77
4.6.2 Remove Panel Output	79
4.6.3 Sort Panel Outputs	80
4.7 Panel Cross Switching	81
4.7.1 Get panel details	81

4.7.2 Output Cross Input	86
4.7.3 Input Selection	87
4.8 Panel Configuration	88
4.8.1 Get All Input Configurations	88
4.8.2 Get All Output Configurations	90
4.8.3 Apply Input Configuration	91
4.8.4 Apply Output Configuration	92
4.8.5 Set Physical Panel Enable Status	93
4.8.6 Query Physical Panel Enable Status	95
5 KLS Panel Management	96
5.1 KLS Input Source Discovery Management	96
5.1.1 KLS Add Server Discovery	96
5.1.2 KLS Add Manual Discovery	97
5.1.3 KLS Add Auto Discovery	98
5.1.4 KLS Delete Discovery Item	99
5.2 KLS Output Management	101
5.2.1 KLS Add Output	101
5.2.2 KLS Delete Output	103
5.2.3 KLS Modify Output	105
5.2.4 KLS Query Output	106
5.2.5 KLS Get All Outputs	107
5.3 KLS Panel Basic Operations	109
5.3.1 KLS Add Panel	109
5.3.2 KLS Delete Panel	110
5.3.3 KLS Switch Enabled	111
5.3.4 KLS Delete Device	117
5.4 KLS Panel Input Management	118
5.4.1 KLS Add Panel Input	118
5.4.2 KLS Remove Panel Input	120
5.5 KLS Panel Output Management	121
5.5.1 KLS Add Panel Output	121
5.5.2 KLS Remove Panel Output	122
5.6 KLS Panel Cross Switching	123
5.6.1 KLS Get Panel Details	123
5.6.2 KLS Output Cross Input	127
5.6.3 KLS Select All Inputs	128
6 KLS Binding Management	130
6.1 Bind KLS	130
6.2 Unbind KLS	131

6.3 Query KLS VLAN	132
6.4 Get Binding Information	133
6.5 Get Binding Status	134
7 Icon Management	135
7.1 Upload Icon	135
7.2 Modify Icon	137
7.3 Get All Icons	138
7.4 Notify Icon Update	139
8 Front Panel Operation	141
8.1 Input Preview	141
8.2 Execute Switch	142
8.3 Query Front Panel	143
8.4 Get Front Panel Details	144
8.5 Stop NDI	147
9 VLAN Configuration	148
9.1 Get All VLAN Configuration	148
9.2 Modify VLAN Configuration	149
10 Topology Management	150
10.1 Query Topology Structure	150
10.2 Query Topology Status	155
10.3 Query Device Information	161
11 Network Diagnostic	166
11.1 Diagnostic Management	166
11.1.1 KLS Cross-Device Start End-to-End Diagnostic	166
11.1.2 Get Current Switch All Device List	168
11.1.3 End Diagnostic Task Deletion	170
11.1.4 Diagnostic Device Result Query	171
11.1.5 Start End-to-End Diagnostic	176
11.1.6 Query Diagnostic Tasks (by Type)	178
12 Switch Management	180
12.1 System Information	180
12.1.1 Query Switch System Information	180
12.1.2 Set Device Alias	182
12.2 Port Management	183
12.2.1 Query All Port Status	183
12.2.2 Port Control	190
12.2.3 Set Port	191
12.2.4 Batch Set Ports	193
12.2.5 Set Port Alias	196

12.2.6 Query Port Traffic	197
12.2.7 Query Port Traffic Top	199
12.2.8 Set Watched Port	202
12.2.9 Remove Watched Port	203
12.2.10 Query Watched Port Information	205
12.2.11 Remove Device	209
12.3 Port Aggregation	210
12.3.1 Create Link Aggregation Group	210
12.3.2 Delete Link Aggregation Group	212
12.3.3 Query All Link Aggregation Groups	213
12.3.4 Query Specified or All Link Aggregation Groups	214
12.3.5 Append Ports to Aggregation Group	216
12.3.6 Modify Link Aggregation Group Configuration	217
12.4 VLAN Management	219
12.4.1 Query All VLAN Information	219
12.4.2 Add VLAN	223
12.4.3 Set VLAN	225
12.4.4 Query VLAN Information	227
12.4.5 Delete VLAN	229
12.4.6 Set VLAN Port	230
12.5 DHCP	231
12.5.1 Query DHCP Client List	231
12.5.2 Get VLAN Interface and DHCP Configuration for Specified VLAN	233
12.5.3 Set VLAN Interface and DHCP Configuration for Specified VLAN	235
12.5.4 Enable or Disable DHCP Service for Specified VLAN	237
12.5.5 Get Static IP Binding List	238
12.5.6 Batch Set Static IP Binding	240
12.6 Power Management	242
12.6.1 Query Power Information	242
12.7 Static Route	244
12.7.1 Add Static Route	244
12.7.2 Delete Static Route	245
12.7.3 Get All Static Routes List	246
12.8 Network Save	248
12.8.1 Save Network Configuration	248
12.8.2 Query Network Configuration	249
13 Alarm Events	250
13.1 Alarm Event Management	250
13.1.1 Query Alarm Events by Device	250

13.1.2 Delete Alarm Events	251
13.1.3 Paginated Query of Alarm Events	252
13.1.4 Filter Event List by Alarm Level	255
13.1.5 Batch Set Alarm Event Status	257
13.1.6 Download Alarm Export File	258
13.1.7 Export Alarm List and Get File Path	259
13.2 Alarm Configuration	261
13.2.1 Get Alarm Policy Configuration	261
13.2.2 Set Alarm Policy Configuration	262
13.3 Alarm Debug	264
13.3.1 Debug: Query All Port Alarms (DEBUG)	264
13.3.2 Debug: Query Port Alarm (DEBUG)	266
14 Production Test	268
14.1 Production Test and Device Information	268
14.1.1 Restore Factory Settings	268
14.1.2 Get Fan List	269
14.1.3 Get Device Resources and Version Information	270
14.1.4. Query Switch Port Status	273
14.1.5 Set Serial Number	274
14.1.6 Start Fan Test	275
14.1.7 Stop Fan Test	277
14.1.8 Query USB Port Status	278
15 AVX Page Interface	279
15.1 Get System Information	279
15.2 Get Network Interface	281
16 AVX Reverse Proxy Device Page Interface	282
16.1 Get Child Device iframe Page Interface	282

1 User Authentication

1.1 User Management / Users

1.1.1 User Login

Interface Description

Login authentication; upon successful login, issues a token for interface authentication.

Request URL

/v1/user/login

Request Method

POST

Request Parameter Example

```
1 {  
2   "username": "admin",  
3   "password": "admin"  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
username	string	True	User login account
password	string	True	User login password

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {  
5     "alias": "admin",  
6     "token":
```

```
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluliwiZXhwIjoxNzY5Njc4MzEwLCJp
c3MiOiJtZWRRpYS1zd2l0Y2hlcilslm5iZiI6MTc2OTY2NzY1MH0.MjS7dXpXkSbmnPczye17visjuK-RwZEM
oHoddwH9YnQ"
```

```
7 }
```

```
8 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Login response core data
data.alias	string	User-set alias / nickname
data.token	string	Interface authentication token value issued after successful login

1.1.2 Delete User

Interface Description

Delete specified system users, supports batch deletion.

Request URL

/v1/user/delete

Request Method

POST

Request Parameter Example

```
1 {
2   "ids": [2, 3, 4]
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
----------------	----------------	------------------	-------------

ids	array	True	Array of user IDs to be deleted
-----	-------	------	---------------------------------

Response Example

```
1 {  
2   "code": 200,  
3   "message": "用户删除成功",  
4   "data": null  
5 }
```

Response parameter description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Operation result description
data	null	Return data; null when no business data

1.1.3 Add User

Interface Description

Add a new system login user, supports setting administrator / regular user role.

Request URL

/v1/user/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "alias": "csadmin",  
3   "password": "123",  
4   "role": "admin",  
5   "username": "csadmin"
```

6 }

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
alias	string	True	User nickname / alias
password	string	True	User login password
role	string	True	User role. Enum: common - regular user, admin - administrator
username	string	True	User login account

Response Example

```
1 {  
2   "code": 200,  
3   "message": "用户创建成功",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Operation result description
data	null	Return data; null when no business data

1.1.4 Modify User

Interface Description

Modify basic user information such as nickname, password, role, etc.

Request URL

/v1/user/modify

Request Method

POST

Request Parameter Example

```
1 {  
2   "alias": "csadmin_new",  
3   "id": 0,  
4   "password": "newpassword123",  
5   "role": "common"  
6 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
alias	string	True	Modified user nickname / alias
id	string	True	Target user ID to be modified
password	string	True	New password after user modification
role	string	True	Modified user role. Enum: common - regular user, admin - administrator

Response Example

```
1 {  
2   "code": 200,  
3   "message": "修改用户信息成功",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Operation result description

data	null	Return data; null when no business data
------	------	---

1.1.5 Get All User List

Interface Description

Get the list and detailed information of all registered users in the current system.

Request URL

/v1/user/userlist

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "id": 1,
7       "username": "admin",
8       "alias": "admin",
9       "role": "admin",
10      "online": true,
11      "first_login": true,
12      "create_time": "2026-02-04 18:51:27",
13      "update_time": "2026-02-04 18:51:27"
```

```

14     }
15 ]
16 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	System user list array
data.id	int32	User ID
data.username	string	User login account
data.alias	string	User nickname / alias
data.role	string	User role. Enum: common - regular user, admin - administrator
data.online	bool	Whether user is currently online
data.first_login	bool	Whether user is logging in for the first time
data.create_time	string	User creation time, format: yyyy-MM-dd HH:mm:ss
data.update_time	string	User information last update time, format: yyyy-MM-dd HH:mm:ss

2 System Management

2.1 System Settings

2.1.1 Set System Name

Interface Description

Modify the system hostname of the device.

Request URL

/v1/sys/sethostname

Request Method

GET

Request Parameter Example

Request URL example: /v1/sys/sethostname?hostname=AVX-Core-Switch-001

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
hostname	string	True	System hostname, Query parameter

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

2.1.2 Restore Factory Settings

Interface Description

Restore all device configurations to factory default state.

Request URL

/v1/sys/refactory

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

2.1.3 Get System Log

Interface Description

Get the system log text content of the device operation.

Request URL

/v1/sys/syslog

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": "2024-01-01 00:00:00 system boot success\n2024-01-01 00:00:05 network interface
up\n2024-01-01 10:00:00 diagnose task 2001 start\n2024-01-01 10:03:00 diagnose task 2001
finished"
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	string	Complete system log text content

2.1.4 Reboot System

Interface Description

Execute device system soft reboot operation.

Request URL

/v1/sys/reboot

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

2.1.5 Firmware Upload

Interface Description

Upload firmware file, submitted in JSON format, file content transmitted in base64 encoding.

Request URL

/v1/sys/upload

Request Method

POST

Request Parameter Example

```
1 {  
2   "u_file": "base64-encoded-bytes",  
3   "filename": "AVX-Firmware.bin"  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
u_file	string	True	Uploaded firmware file content, base64 encoded format
filename	string	True	Firmware file name

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data

2.1.6 Firmware Upgrade

Interface Description

Trigger device firmware upgrade operation; the upgrade package must first be uploaded via the firmware upload interface.

Request URL

/v1/sys/upgrade

Request Method

POST

Request Parameter Example

```
1 {  
2   "filename": "AVX-Firmware.bin"  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
filename	string	True	Firmware upgrade package file name

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

2.1.7 Get Runtime Log

Interface Description

Get the device system operation log text content.

Request URL

/v1/sys/getlog

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": "log content..."  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	string	Complete system operation log text content

2.1.8 Export Operation Log

Interface Description

Package system operation logs into a txt file, return file storage path for subsequent download.

Request URL

/v1/sys/operation/export

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": "/data/operation/operation.txt"  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	string	File storage path after operation log export

2.2 System Time

2.2.1 Get System Time and Timezone

Interface Description

Get current device system time, timezone, NTP configuration, and other related information.

Request URL

/v1/systime/get

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "",  
4   "data": {  
5     "Timezone": "Asia/Shanghai",  
6     "ntp": "time.windows.com time.apple.com pool.ntp.org",  
7     "offset": 8,  
8     "time": "2026-03-18 15:54:48",  
9     "timetype": "pc"  
10  }  
11 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	System time and timezone core data
data.Timezone	string	System current timezone
data.ntp	string	System configured NTP server address
data.offset	int32	System time offset
data.time	string	Current system time, format: yyyy-MM-dd HH:mm:ss
data.timetype	string	System time type. Enum: pc - sync with PC, manual - manual setting, ntp - NTP time synchronization

2.2.2 Set Timezone

Interface Description

Set system timezone and time offset.

Request URL

/v1/systime/setLocation

Request Method

POST

Request Parameter Example

```
1 {  
2   "timezone": "Asia/Shanghai",  
3   "offset": 8  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
timezone	string	True	Target timezone
offset	int32	True	Time offset corresponding to target timezone

Response Example

```
1 {  
2   "code": 200,  
3   "message": "设置地区时间成功",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Operation result description
data	null	Return data; null when no business data

2.2.3 Set System Time

Interface Description

Set system time type, manual time, NTP sync address, and other time-related configurations.

Request URL

/v1/systime/setLocation

Request Method

POST

Request Parameter Example

NTP time synchronization configuration example:

```
1 {  
2   "timetype": "ntp",  
3   "offset": 8,  
4   "ntp": "time.windows.com time.apple.com pool.ntp.org"  
5 }
```

Manual time setting example:

```
1 {  
2   "timetype": "manual",  
3   "offset": 8,  
4   "time": "2026-04-16 15:07:44"  
5 }
```

PC time synchronization example:

```
1 {  
2   "timetype": "pc",  
3   "offset": 8  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
timetype	string	True	Time type. Enum: manual - manual setting, ntp - NTP time synchronization, pc - sync with PC time
offset	int32	True	System time offset
time	string	False	Manually set system time, format: yyyy-MM-dd HH:mm:ss; not required when timetype is ntp
ntp	string	False	NTP time synchronization server

			address; only required when timetype is ntp
--	--	--	---

Response Example

```
1 {
2   "code": 200,
3   "message": "设置时间成功",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Operation result description
data	null	Return data; null when no business data

2.3 PTP Clock

2.3.1 Query PTP Clock Status

Interface Description

Query PTP clock status and port binding information.

Request URL

/v1/ptp/query

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {  
5     "ports": [],  
6     "ptps": []  
7   }  
8 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return core data
data.ports	array	Port list (SWPort)
data.ports.id	int32	Port ID
data.ports.ptp_id	int32	Bound PTP clock ID; 0 means unbound
data.ports.mode	string	Port mode. Enum: e2e/p2p
data.ports.master	bool	Whether primary port
data.ports.vlanInfo	object	Port VLAN information
data.ports.vlanInfo.type	string	VLAN mode
data.ports.vlanInfo.pvid	int32	VLAN PVID
data.ports.vlanInfo.vid	string	VLAN range
data.ports.vlanInfo.alias	string	VLAN alias
data.ports.vlanInfo.color	string	VLAN color
data.ports.vlanInfo.config_file	string	VLAN configuration file
data.ptps	array	PTP clock instance list (SWPtp)
data.ptps.id	int32	Clock instance ID
data.ptps.enable	bool	Whether clock is enabled

data.ptps.domain	int32	Clock domain
data.ptps.type	string	Clock type. Enum: BC/TC/stop
data.ptps.identity	string	Clock identifier
data.ptps.pport_id	string	Parent port ID
data.ptps.status	int32	Clock running status. Enum: 0 - Stopped, 1 - Running, 3 - Configuration error
data.ptps.two_step	bool	Whether two-step mode is enabled
data.ptps.priority1	int32	Clock priority 1
data.ptps.priority2	int32	Clock priority 2
data.ptps.leap_s	bool	Leap second switch
data.ptps.de_mainclock	string	Offset from master clock
data.ptps.average_delay	string	Average path delay
data.ptps.upstream_equipment	string	Upstream device name
data.ptps.upstream_port	string	Upstream port ID
data.ptps.ports	array	List of ports bound to the clock, structure same as data.ports

2.3.2 Save PTP Clock Start/Stop

Interface Description

Save PTP clock enabled status, supports batch configuration of multiple clock instances start/stop.

Request URL

/v1/ptp/save

Request Method

POST

Request Parameter Example

```
1 {
2   "ptps": [
3     {
4       "id": 1,
```

```

5         "enable": true
6     },
7     {
8         "id": 2,
9         "enable": false
10    }
11 ]
12 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ptps	array	True	Clock start/stop configuration list
ptps.id	int32	True	Clock instance ID, value range 1-3
ptps.enable	bool	True	Whether this clock instance is enabled

Response Example

```

1 {
2     "code": 200,
3     "message": "ok",
4     "data": {}
5 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

2.3.3 Query PTP Clock Details

Interface Description

Query detailed configuration information for the specified PTP clock instance.

Request URL

/v1/ptp/detail/query

Request Method

GET

Request Parameter Example

Request URL example: /v1/ptp/detail/query?id=1

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int32	True	Clock instance ID, value range 1-3, Query parameter

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {  
5     "id": 1,  
6     "enable": true,  
7     "domain": 0,  
8     "type": "BC",  
9     "two_step": true,  
10    "priority1": 128,  
11    "priority2": 128,  
12    "leap_s": false,
```

```

13     "ports": []
14 }
15 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Clock detail core data
data.id	int32	Clock instance ID
data.enable	bool	Whether clock is enabled
data.domain	int32	Clock domain
data.type	string	Clock type. Enum: BC/TC/stop
data.two_step	bool	Whether two-step mode is enabled
data.priority1	int32	Clock priority 1, value range 0-255
data.priority2	int32	Clock priority 2, value range 0-255
data.leap_s	bool	Leap second switch
data.ports	array	List of ports bound to the clock (SWPort)
data.ports.id	int32	Port ID
data.ports.ptp_id	int32	Bound PTP clock ID
data.ports.mode	string	Port mode. Enum: e2e/p2p
data.ports.master	bool	Whether primary port
data.ports.vlanInfo	object	Port VLAN information, structure same as 4.2.1 interface definition

2.3.4 Modify PTP Clock Configuration

Interface Description

Modify the basic configuration and port binding rules for the specified PTP clock instance.

Request URL

/v1/ptp/modify

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "type": "BC",  
4   "domain": 0,  
5   "two_step": true,  
6   "priority1": 128,  
7   "priority2": 128,  
8   "leap_s": false,  
9   "ports": [  
10    {  
11      "id": 1,  
12      "mode": "e2e",  
13      "master": true  
14    }  
15  ]  
16 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int32	True	Clock instance ID, value range 1-3
type	string	True	Clock type. Enum: BC/TC/stop
domain	int32	False	Clock domain; default value is id-1 if not provided
two_step	bool	False	Whether two-step

			mode is enabled
priority1	int32	False	Clock priority 1, value range 0-255
priority2	int32	False	Clock priority 2, value range 0-255
leap_s	bool	False	Leap second switch
ports	array	False	Port configuration list
ports.id	int32	True	Port ID
ports.ptp_id	int32	False	Bound PTP clock ID
ports.mode	string	True	Port mode. Enum: e2e/p2p
ports.master	bool	False	Whether it is the primary port; forced to false in TC mode

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

3 Access Policy / Preset Configuration

3.1 Access Policy

3.1.1 Get Backup List

Interface Description

Get the list of all saved configuration presets in the system.

Request URL

/v1/preset/getlist

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": [  
5     {  
6       "prname": "default_config",  
7       "version": "1.0.0",  
8       "apply": true  
9     },  
10    {  
11      "prname": "performance_mode",
```

```

12         "version": "1.2.0",
13         "apply": false
14     }
15 ]
16 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Preset list
data[].prname	string	Preset name
data[].version	string	Preset version number
data[].apply	bool	Whether preset is available

3.1.2 Save Backup

Interface Description

Save the current running configuration of the device as a new preset template.

Request URL

/v1/preset/save

Request Method

POST

Request Parameter Example

```

1 {
2     "prname": "high_performance_v2"
3 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
prname	string	True	Preset name , is not allowed to contain "-" characters

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

3.1.3 Read Backup

Interface Description

Read the specified preset and apply its configuration to the current device.

Request URL

/v1/preset/read

Request Method

POST

Request Parameter Example

```
1 {
```

```
2   "prname": "default_config"
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
prname	string	True	Preset name

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

3.1.4 Delete Backup

Interface Description

Delete specified presets, supports batch deletion.

Request URL

/v1/preset/delete

Request Method

POST

Request Parameter Example

```
1 {
2   "namelist": ["old_config", "test_mode_v1", "invalid_preset"]
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
namelist	array	True	Preset name array

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "delete_info": {
5     "delete_list": ["old_config", "test_mode_v1"],
6     "notfound": ["invalid_preset"],
7     "failed": [],
8     "message": "2 个预设删除成功, 1 个预设未找到"
9   }
10 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
delete_info	object	Deletion result details
delete_info.delete_list	array	List of successfully deleted preset names
delete_info.notfound	array	List of preset names not found
delete_info.failed	array	List of preset names that failed to be deleted
delete_info.message	string	Deletion result text description

3.1.5 Modify Backup Name

Interface Description

Modify the name of an existing preset.

Request URL

/v1/preset/modify

Request Method

POST

Request Parameter Example

```
1 {  
2   "old_name": "performance_mode",  
3   "new_name": "high_performance_v1"  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
old_name	string	True	Original preset name
new_name	string	True	New preset name

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success

message	string	Business prompt message
data	object	Return data, may be empty

3.2 Virtual NIC Interface

3.2.1 Query Virtual Interface Configuration for Specified VLAN

Interface Description

Query virtual interface configuration information for the specified VLAN

Request URL

/v1/preset/vlan/interface/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_id	int32	True	VLAN ID (location: query)

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "interface": {
6       "enable": true,
7       "ip": "192.168.1.1",
8       "netmask": "255.255.255.0",
9       "type": "static",
10      "referenced": true,
```

```

11         "is_default_route": true
12     },
13     "recommend": {
14         "dhcp": {
15             "is_recommend": false,
16             "recommend_type": "dhcp"
17         },
18         "manual": {
19             "is_recommend": true,
20             "recommend_type": "manual",
21             "vlanif_ip": "192.168.1.254",
22             "recommend_range": "192.168.1.1-192.168.1.200"
23         }
24     }
25 }
26 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
interface	object	Virtual interface configuration information
interface.enable	bool	Virtual interface enable switch
interface.ip	string	Virtual interface IP address
interface.netmask	string	Subnet mask
interface.type	string	Interface type
interface.referenced	bool	Whether referenced by matrix
interface.is_default_route	bool	Whether set as default route
recommend	object	Recommended configuration info (DHCP/manual configuration), this field is only for frontend recommendation hints
recommend.dhcp	object	Business parameter
recommend.dhcp.is_recommend	bool	Boolean switch value
recommend.dhcp.recommend_type	string	Returned type identifier

recommend.manual	object	Business parameter
recommend.manual.is_recommend	bool	Boolean switch value
recommend.manual.recommend_type	string	Returned type identifier
recommend.manual.vlanif_ip	string	Returned IP address
recommend.manual.recommend_range	string	String parameter

- GET /v1/preset/vlan/interface/list/get - Get Virtual NIC Interface List
- POST /v1/preset/vlan/interface/set - Set Virtual NIC Interface

3.2.2 Query All Virtual NIC Interface IP List

Interface Description

Query IP configuration list for all virtual NICs (with alias and color information)

Request URL

/v1/preset/vlan/interface/list/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "vlan_id": 1,
```

```
7      "ip": "string",
8      "netmask": "string",
9      "alias": "string",
10     "color": "string"
11   }
12 ]
13 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	VLAN interface IP configuration list
data.vlan_id	int32	VLAN ID
data.ip	string	IP address
data.netmask	string	Subnet mask
data.alias	string	VLAN alias
data.color	string	VLAN color

3.2.3 Set Virtual NIC Interface Configuration for Specified VLAN

Interface Description

Set Virtual NIC Configuration for Specified VLAN

Request URL

/v1/preset/vlan/interface/set

Request Method

POST

Request Parameter Example

```
1 {
```

```

2    "vlan_id": 10,
3    "interface": {
4        "enable": true,
5        "ip": "192.168.10.1",
6        "netmask": "255.255.255.0",
7        "type": "static",
8        "is_default_route": false
9    }
10 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_id	int32	True	VLAN ID
interface	object	True	Virtual interface configuration information
interface.enable	bool	True	Virtual interface enable switch
interface.ip	string	True	Virtual interface IP address
interface.netmask	string	True	Subnet mask
interface.type	string	False	Interface type
interface.is_default_route	bool	False	Whether set as default route

Response Example

```

1 {
2     "code": 200,
3     "message": "ok"
4 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

3.2.4 Query All VLAN Virtual Interface IP Information

Interface Description

Query all VLANIF IP address information
Provide small screen with interface to get all virtual NIC IPs

Request URL

/v1/switch/interface/ip/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "interface_ip_list": [
6       {
7         "vlan_id": 0,
8         "ip": "192.168.0.168"
9       },
10      {
11        "vlan_id": 1,
12        "ip": "192.168.1.1"
```

```

13     }
14   ]
15 }
16 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	VLAN interface IP information
data.interface_ip_list	array	VLAN-to-IP mapping list
data.interface_ip_list.vlan_id	int32	VLAN ID (0 represents physical interface eth0), i.e., the OOB management port
data.interface_ip_list.ip	string	IP address

3.3 Mode Switch

3.3.1 Query Switch Layer 2/Layer 3 Mode

Interface Description

Query whether the current switch is in Layer 2 mode or Layer 3 mode

Request URL

/v1/preset/mode/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {  
5     "mode": "layer3"  
6   }  
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data.mode	string	Current mode: layer2 (Layer 2 mode) or layer3 (Layer 3 mode)

3.3.2 Set Switch Layer 2/Layer 3 Mode

Interface Description

Switch between Layer 2/Layer 3 mode of the switch

Request URL

/v1/preset/mode/set

Request Method

POST

Request Parameter Example

```
1 {  
2   "mode": "layer3"  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
mode	string	True	Target mode: layer2 (Layer 2 mode) or layer3 (Layer 3 mode)

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok"  
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

4 Panel Management

4.1 Input Source Discovery

4.1.1 Add Server Discovery

Interface Description

Add server discovery

Request URL

/v1/input/server/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "dis_g_name": "public",  
3   "dis_s_ip": "192.168.30.125"  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
dis_g_name	string	Yes	Discovery group name
dis_s_ip	string	Yes	Discovery server IP address

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.1.2 Add Manual Discovery

Interface Description

Add manual discovery

Request URL

/v1/input/manual/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "manuals": [  
3     {  
4       "dis_g_name": "public",  
5       "dis_d_ip": "192.168.40.23"  
6     }  
7   ]  
8 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
manuals	array	Yes	Manual discovery information array
manuals[].dis_g_name	string	Yes	Discovery group name
manuals[].dis_d_ip	string	Yes	Discovered device IP address

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success

message	string	Operation result description
data	null	No data returned

4.1.3 Add Auto Discovery

Interface Description

Add automatic discovery

Request URL

/v1/input/auto/add

Request Method

POST

Request Parameter Example

```
1 {
2     "dis_g_name": "public"
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
dis_g_name	string	Yes	Discovery group name

Response Example

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success

message	string	Operation result description
data	null	No data returned

4.1.4 Server Discovery Query

Interface Description

Server input discovery

Request URL

/v1/input/server/discovery

Request Method

GET

Request Parameter Example

```
1 {
2   "dis_g_name": "group1",
3   "dis_s_ip": "192.168.1.1"
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
dis_g_name	string	Yes	Discovery group name
dis_s_ip	string	Yes	Discovery server IP address

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "inputs": [
6       {
7         "name": "input1",
```

```

8      "alias": "输入 1",
9      "c_name": "通道 1",
10     "d_name": "设备 1",
11     "d_ip": "192.168.1.100",
12     "d_port": 554,
13     "url": "192.168.1.100:554",
14     "enable": true,
15     "dis_method": "server",
16     "dis_g_name": "group1",
17     "dis_d_ip": "",
18     "dis_s_ip": "192.168.1.1",
19     "vlan_id": 100,
20     "switcher_id": "00000000000000"
21   }
22 ]
23 }
24 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.inputs	array	Input list
data.inputs[].name	string	Input name
data.inputs[].alias	string	Input alias
data.inputs[].c_name	string	Channel name
data.inputs[].d_name	string	Device name
data.inputs[].d_ip	string	Device IP address
data.inputs[].d_port	int	Device port
data.inputs[].url	string	Input URL
data.inputs[].enable	bool	Whether enabled
data.inputs[].dis_method	string	Discovery method
data.inputs[].dis_g_name	string	Discovery group name

data.inputs[].dis_d_ip	string	Discovered device IP address
data.inputs[].dis_s_ip	string	Discovery server IP address
data.inputs[].vlan_id	int	VLAN ID
data.inputs[].switcher_id	string	Switch serial number

4.1.5 Manual Discovery Query

Interface Description

Manual input discovery

Request URL

/v1/input/manual/discovery

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "inputs": [
6       {
7         "name": "input2",
8         "alias": "输入 2",
9         "c_name": "通道 2",
10        "d_name": "设备 2",
```

```

11         "d_ip": "192.168.1.101",
12         "d_port": 554,
13         "url": "192.168.1.101:554",
14         "enable": true,
15         "dis_method": "manual",
16         "dis_g_name": "group1",
17         "dis_d_ip": "192.168.1.101",
18         "dis_s_ip": "",
19         "vlan_id": 100,
20         "switcher_id": "00000000000000"
21     }
22 ]
23 }
24 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.inputs	array	Input list
data.inputs[].name	string	Input name
data.inputs[].alias	string	Input alias
data.inputs[].c_name	string	Channel name
data.inputs[].d_name	string	Device name
data.inputs[].d_ip	string	Device IP address
data.inputs[].d_port	int	Device port
data.inputs[].url	string	Input URL
data.inputs[].enable	bool	Whether enabled
data.inputs[].dis_method	string	Discovery method
data.inputs[].dis_g_name	string	Discovery group name
data.inputs[].dis_d_ip	string	Discovered device IP address
data.inputs[].dis_s_ip	string	Discovery server IP address
data.inputs[].vlan_id	int	VLAN ID
data.inputs[].switcher_id	string	Switch serial number

4.1.6 Auto Discovery Query

Interface Description

Automatic input discovery

Request URL

/v1/input/auto/discovery

Request Method

GET

Request Parameter Example

```
1 {
2     "dis_g_name": "group1"
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
dis_g_name	string	Yes	Discovery group name

Response Example

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "inputs": [
6             {
7                 "name": "input3",
8                 "alias": "输入 3",
9                 "c_name": "通道 3",
10                "d_name": "设备 3",
11                "d_ip": "192.168.1.102",
12                "d_port": 554,
```

```

13         "url": "192.168.1.102:554",
14         "enable": true,
15         "dis_method": "auto",
16         "dis_g_name": "group1",
17         "dis_d_ip": "192.168.1.102",
18         "dis_s_ip": "",
19         "vlan_id": 100,
20         "switcher_id": "00000000000000"
21     }
22 ]
23 }
24 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.inputs	array	Input list
data.inputs[].name	string	Input name
data.inputs[].alias	string	Input alias
data.inputs[].c_name	string	Channel name
data.inputs[].d_name	string	Device name
data.inputs[].d_ip	string	Device IP address
data.inputs[].d_port	int	Device port
data.inputs[].url	string	Input URL
data.inputs[].enable	bool	Whether enabled
data.inputs[].dis_method	string	Discovery method
data.inputs[].dis_g_name	string	Discovery group name
data.inputs[].dis_d_ip	string	Discovered device IP address
data.inputs[].dis_s_ip	string	Discovery server IP address
data.inputs[].vlan_id	int	VLAN ID
data.inputs[].switcher_id	string	Switch serial number

4.1.7 Delete Discovery Item

Interface Description

Delete discovered input

Request URL

/v1/input/discovery/delete

Request Method

POST

Request Parameter Example

```
1 {  
2   "ids": [1, 2, 3]  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ids	array	Yes	Array of discovery IDs to delete

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description

data	null	No data returned
------	------	------------------

4.1.8 Get All Discovery Items

Interface Description

Get all discovered inputs

Request URL

/v1/input/discovery/all

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "manuals": [
6       {
7         "dis_id": 3,
8         "dis_g_name": "public",
9         "dis_d_ip": "192.168.40.23"
10      }
11    ],
12    "auto_servers": [
13      {
```

```

14         "dis_id": 1,
15         "dis_method": "auto",
16         "dis_g_name": "public",
17         "dis_d_ip": "",
18         "dis_s_ip": ""
19     },
20     {
21         "dis_id": 2,
22         "dis_method": "auto",
23         "dis_g_name": "public1",
24         "dis_d_ip": "",
25         "dis_s_ip": ""
26     },
27     {
28         "dis_id": 4,
29         "dis_method": "server",
30         "dis_g_name": "public",
31         "dis_d_ip": "",
32         "dis_s_ip": "192.43.56.2"
33     }
34 ]
35 }
36 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.manuals	array	Manual discovery list
data.manuals[].dis_id	int	Discovery ID
data.manuals[].dis_g_name	string	Discovery group name
data.manuals[].dis_d_ip	string	Discovered device IP address
data.auto_servers	array	Automatic server discovery list

data.auto_servers[].dis_id	int	Discovery ID
data.auto_servers[].dis_method	string	Discovery method
data.auto_servers[].dis_g_name	string	Discovery group name
data.auto_servers[].dis_d_ip	string	Discovered device IP address
data.auto_servers[].dis_s_ip	string	Discovery server IP address

4.2 Input Management

4.2.1 Modify Input Source

Interface Description

Modify input source

Request URL

/v1/input/modify

Request Method

POST

Request Parameter Example

```
1 {
2     "id": 1,
3     "alias": "新输入别名"
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Input ID
alias	string	No	Input alias

Response Example

```
1 {
2     "code": 200,
3     "message": "Success",
4 }
```

```
4     "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.3 Output Management

4.3.1 Add Output

Interface Description

Add output

Request URL

/v1/output/add

Request Method

POST

Request Parameter Example

```
1 {
2     "outputs": [
3         {
4             "name": "output1",
5             "alias": "输出 1",
6             "c_name": "通道 1",
7             "g_name": "group1",
8             "router": true,
9             "dis_s_ip": "192.168.1.1",
10            "vlan_id": 100
11         }
12     ]
13 }
```

```

11     }
12 ]
13 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
outputs	array	Yes	Output list
outputs[].name	string	Yes	Output name
outputs[].alias	string	Yes	Output alias
outputs[].c_name	string	Yes	Channel name
outputs[].g_name	string	Yes	Group name
outputs[].router	bool	Yes	Whether route
outputs[].dis_s_ip	string	Yes	Discovery server IP address
outputs[].vlan_id	int	Yes	VLAN ID

Response Example

```

1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.3.2 Delete Output

Interface Description

Delete output

Request URL

/v1/output/delete

Request Method

POST

Request Parameter Example

```
1 {  
2   "ids": [1, 2, 3]  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ids	array	Yes	Array of output IDs to delete

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.3.3 Modify Output

Interface Description

Modify output

Request URL

/v1/output/modify

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "name": "new_name",  
4   "g_name": "new_group",  
5   "router": false,  
6   "vlan_id": 200  
7 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Output ID
name	string	No	Output name
g_name	string	No	Group name
dis_s_ip	string	No	Discovery server IP address
router	bool	No	Whether route
vlan_id	int	No	VLAN ID

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",
```

```
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.3.4 Query Output

Interface Description

Query output

Request URL

/v1/output/query

Request Method

GET

Request Parameter Example

```
1 {
2   "id": 1
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Output ID

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
```

```

4   "data": {
5       "id": 1,
6       "name": "output1",
7       "alias": "输出 1",
8       "c_name": "通道 1",
9       "g_name": "group1",
10      "smooth": 0,
11      "router": true,
12      "codec_k": 0,
13      "dis_s_ip": "192.168.1.1",
14      "locked": false,
15      "enable": true,
16      "vlan_id": 100,
17      "vlan_alias": "VLAN100",
18      "vlan_color": "#0000FF",
19      "switcher_id": "00000000000000",
20      "connect_num": 0,
21      "create_time": "2026-04-21 00:00:00.000000",
22      "update_time": "2026-04-21 00:00:00.000000"
23  }
24 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Output information
data.id	int	Output ID
data.name	string	Output name
data.alias	string	Output alias
data.c_name	string	Channel name
data.g_name	string	Group name
data.smooth	int	Smoothness
data.router	bool	Whether route

data.codec_k	int	Codec keep-alive
data.dis_s_ip	string	Discovery server IP address
data.locked	bool	Whether locked
data.enable	bool	Whether enabled
data.vlan_id	int	VLAN ID
data.vlan_alias	string	VLAN alias
data.vlan_color	string	VLAN color
data.switcher_id	string	Switch serial number
data.connect_num	int	Connection count
data.create_time	string	Creation time
data.update_time	string	Update time

4.3.5 Get All Outputs

Interface Description

Query all outputs

Request URL

/v1/output/query/all

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
```

```

5      "outputs": [
6          {
7              "id": 1,
8              "name": "output1",
9              "alias": "输出 1",
10             "c_name": "通道 1",
11             "g_name": "group1",
12             "smooth": 0,
13             "router": true,
14             "codec_k": 0,
15             "dis_s_ip": "192.168.1.1",
16             "locked": false,
17             "enable": true,
18             "vlan_id": 100,
19             "vlan_alias": "VLAN100",
20             "vlan_color": "#0000FF",
21             "switcher_id": "00000000000000",
22             "connect_num": 0,
23             "create_time": "2026-04-21 00:00:00.000000",
24             "update_time": "2026-04-21 00:00:00.000000"
25         },
26         {
27             "id": 2,
28             "name": "output2",
29             "alias": "输出 2",
30             "c_name": "通道 2",
31             "g_name": "group1",
32             "smooth": 0,
33             "router": true,
34             "codec_k": 0,
35             "dis_s_ip": "192.168.1.1",
36             "locked": false,
37             "enable": true,
38             "vlan_id": 100,

```

```

39         "vlan_alias": "VLAN100",
40         "vlan_color": "#0000FF",
41         "switcher_id": "00000000000000",
42         "connect_num": 0,
43         "create_time": "2026-04-21 00:00:00.000000",
44         "update_time": "2026-04-21 00:00:00.000000"
45     }
46 ]
47 }
48 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.outputs	array	Output list
data.outputs[].id	int	Output ID
data.outputs[].name	string	Output name
data.outputs[].alias	string	Output alias
data.outputs[].c_name	string	Channel name
data.outputs[].g_name	string	Group name
data.outputs[].smooth	int	Smoothness
data.outputs[].router	bool	Whether route
data.outputs[].codec_k	int	Codec keep-alive
data.outputs[].dis_s_ip	string	Discovery server IP address
data.outputs[].locked	bool	Whether locked
data.outputs[].enable	bool	Whether enabled
data.outputs[].vlan_id	int	VLAN ID
data.outputs[].vlan_alias	string	VLAN alias
data.outputs[].vlan_color	string	VLAN color
data.outputs[].switcher_id	string	Switch serial number
data.outputs[].connect_num	int	Connection count
data.outputs[].create_time	string	Creation time
data.outputs[].update_time	string	Update time

4.3.6 Get Default Output

Interface Description

Query default output

Request URL

/v1/output/query/default

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": {  
5     "name": "output1"  
6   }  
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Default output information
data.name	string	Default output name

4.4 Panel Basic Operations

4.4.1 Add Panel

Interface Description

Add panel

Request URL

/v1/panel/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "name": "panel1"  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
name	string	Yes	Panel name

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success

message	string	Operation result description
data	null	No data returned

4.4.2 Delete Panel

Interface Description

Delete panel

Request URL

/v1/panel/delete

Request Method

POST

Request Parameter Example

```
1 {  
2   "ids": [1, 2, 3]  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ids	array	Yes	Panel ID list

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success

message	string	Operation result description
data	null	No data returned

4.4.3 Modify Panel

Interface Description

Modify panel

Request URL

/v1/panel/modify

Request Method

POST

Request Parameter Example

```
1 {
2     "id": 1,
3     "name": "new_panel_name"
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
name	string	Yes	Panel name

Response Example

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.4.4 Expand Panel

Interface Description

Expand panel

Request URL

/v1/panel/expand

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "expand": true  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
expand	bool	Yes	Whether expanded

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",
```

```
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.4.5 Query Panel

Interface Description

Query panel

Request URL

/v1/panel/query

Request Method

GET

Request Parameter Example

```
1 {
2   "id": 1
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
```

```

4    "data": {
5        "id": 1,
6        "name": "panel1",
7        "input_num": 0,
8        "output_num": 0,
9        "create_time": "2026-04-21 00:00:00.000000",
10       "update_time": "2026-04-21 00:00:00.000000"
11    }
12 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Panel information
data.id	int	Panel ID
data.name	string	Panel name
data.input_num	int	Panel input count
data.output_num	int	Panel output count
data.create_time	string	Creation time
data.update_time	string	Update time

4.4.6 Get All Panels

Interface Description

Query all panels

Request URL

/v1/panel/query/all

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": {  
5     "panels": [  
6       {  
7         "id": 1,  
8         "name": "Default",  
9         "input_num": 2,  
10        "output_num": 1,  
11        "create_time": "2026-04-20 21:29:57.832819",  
12        "update_time": "2026-04-20 21:29:57.832820"  
13      },  
14      {  
15        "id": 2,  
16        "name": "ccc",  
17        "input_num": 0,  
18        "output_num": 0,  
19        "create_time": "2026-04-21 13:46:41.195032",  
20        "update_time": "2026-04-21 13:46:41.195032"  
21      },  
22      {  
23        "id": 3,  
24        "name": "xxx",  
25        "input_num": 0,  
26        "output_num": 0,
```

```
27         "create_time": "2026-04-21 13:55:07.570200",
28         "update_time": "2026-04-21 13:55:07.570200"
29     }
30 ]
31 }
32 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.panels	array	Panel list
data.panels[].id	int	Panel ID
data.panels[].name	string	Panel name
data.panels[].input_num	int	Panel input count
data.panels[].output_num	int	Panel output count
data.panels[].create_time	string	Creation time
data.panels[].update_time	string	Update time

4.5 Panel Input Management

4.5.1 Add Panel Input

Interface Description

Add panel input

Request URL

/v1/panel/input/add

Request Method

POST

Request Parameter Example

```
1 {
2   "id": 1,
3   "inputs": [
4     {
5       "name": "input1",
6       "alias": "输入 1",
7       "c_name": "通道 1",
8       "g_name": "group1",
9       "d_ip": "192.168.1.100",
10      "d_port": 554,
11      "url": "192.168.1.100:554",
12      "enable": true,
13      "dis_method": "manual",
14      "dis_g_name": "public",
15      "dis_d_ip": "192.168.1.100",
16      "dis_s_ip": "",
17      "vlan_id": 100,
18      "switcher_id": "00000000000000"
19    }
20  ]
21 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
inputs[]	array	Yes	Input list
inputs[].name	string	Yes	Name
inputs[].alias	string	Yes	Alias
inputs[].c_name	string	Yes	Channel
inputs[].g_name	string	Yes	Group name
inputs[].d_ip	string	Yes	Device IP
inputs[].d_port	int	Yes	Device port
inputs[].url	string	Yes	url

inputs[].enable	bool	Yes	Whether available
inputs[].dis_method	string	Yes	Discovery method
inputs[].dis_g_name	string	Yes	Discovery group
inputs[].dis_d_ip	string	Yes	Discovered device IP
inputs[].dis_s_ip	string	Yes	Discovery server IP
inputs[].vlan_id	int	Yes	vlanID
inputs[].switcher_id	string	Yes	Switch serial number

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.5.2 Remove Panel Input

Interface Description

Remove panel input

Request URL

/v1/panel/input/remove

Request Method

POST

Request Parameter Example

```
1 {
2   "id": 1,
3   "input_ids": [1, 2, 3]
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
input_ids	array	Yes	Input ID array

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.5.3 Sort Panel Inputs

Interface Description

Panel input sort

Request URL

/v1/panel/input/sort

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "input_ids": [3, 1, 2]  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
input_ids	array	Yes	Sorted input ID array

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.6 Panel Output Management

4.6.1 Add Panel Output

Interface Description

Add panel output

Request URL

/v1/panel/output/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "output_ids": [1, 2, 3]  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
output_ids	array	Yes	Output ID array

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.6.2 Remove Panel Output

Interface Description

Remove panel output

Request URL

/v1/panel/output/remove

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "output_ids": [1, 2]  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
output_ids	array	Yes	Output ID array

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success

message	string	Operation result description
data	null	No data returned

4.6.3 Sort Panel Outputs

Interface Description

Panel output sort

Request URL

/v1/panel/output/sort

Request Method

POST

Request Parameter Example

```
1 {
2     "id": 1,
3     "output_ids": [3, 1, 2]
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
output_ids	array	Yes	Sorted output ID array

Response Example

```
1 {
2     "code": 200,
3     "message": "Success",
4     "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.7 Panel Cross Switching

4.7.1 Get panel details

Interface Description

Get panel details

Request URL

/v1/panel/detail

Request Method

GET

Request Parameter Example

```
1 {  
2   "id": 1  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",
```

```

4  "data": {
5      "id": 1,
6      "name": "Default",
7      "expand": false,
8      "panel_inputs": [
9          {
10             "id": 1,
11             "name": "AVX24-4 (hhyoutput1)",
12             "alias": "AVX24-4 (hhyoutput1)",
13             "c_name": "AVX24-4 (hhyoutput1)",
14             "g_name": "",
15             "d_ip": "172.30.46.14",
16             "d_port": 5961,
17             "url": "172.30.46.14:5961",
18             "enable": true,
19             "dis_method": "auto",
20             "dis_g_name": "public",
21             "dis_d_ip": "",
22             "dis_s_ip": "",
23             "vlan_id": 1,
24             "switcher_id": "000000000234D2",
25             "resolution": "0x0@0HZ",
26             "vb": "0.00Mbps",
27             "ab": "0.00Mbps",
28             "multicast": true,
29             "icon_type": "",
30             "icon_name": ""
31         }
32     ],
33     "panel_outputs": [
34         {
35             "id": 1,
36             "name": "output1",
37             "alias": "output1",

```

```

38         "c_name": "output1",
39         "g_name": "public",
40         "router": false,
41         "enable": true,
42         "dis_s_ip": "",
43         "vlan_id": 1,
44         "vlan_alias": "vlan1",
45         "vlan_color": "#1D2024",
46         "switcher_id": "000000000234D2",
47         "connect_num": 0,
48         "lock": false,
49         "resolution": "0x0@0HZ",
50         "vb": "0.00Mbps",
51         "ab": "0.00Mbps",
52         "input_id": 0,
53         "linked": false,
54         "r_id": 0,
55         "r_loop": false,
56         "r_src_id": 0,
57         "r_src_uuid": "",
58         "r_src_progress": 0,
59         "icon_type": "",
60         "icon_name": "",
61         "d_ip": "172.30.46.175"
62     }
63 ],
64     "other_inputs": [],
65     "panel_templates": [],
66     "rotationlists": []
67 }
68 }

```

Response Parameter Description

Parameter name	Parameter type	Description
----------------	----------------	-------------

code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Panel details
data.id	int	Panel ID
data.name	string	Panel name
data.expand	bool	Whether expanded
data.panel_inputs	array	Panel input list
data.panel_inputs[].id	int	Input ID
data.panel_inputs[].name	string	Input name
data.panel_inputs[].alias	string	Input alias
data.panel_inputs[].c_name	string	Channel name
data.panel_inputs[].g_name	string	Group name
data.panel_inputs[].d_ip	string	Device IP address
data.panel_inputs[].d_port	int	Device port
data.panel_inputs[].url	string	Input URL
data.panel_inputs[].enable	bool	Whether enabled
data.panel_inputs[].dis_method	string	Discovery method
data.panel_inputs[].dis_g_name	string	Discovery group name
data.panel_inputs[].dis_d_ip	string	Discovered device IP address
data.panel_inputs[].dis_s_ip	string	Discovery server IP address
data.panel_inputs[].vlan_id	int	VLAN ID
data.panel_inputs[].switcher_id	string	Switch serial number
data.panel_inputs[].resolution	string	Resolution
data.panel_inputs[].vb	string	Video bitrate
data.panel_inputs[].ab	string	Audio bitrate
data.panel_inputs[].multicast	bool	Whether multicast
data.panel_inputs[].icon_type	string	Icon type
data.panel_inputs[].icon_name	string	Icon name
data.panel_outputs	array	Panel output list
data.panel_outputs[].id	int	Output ID
data.panel_outputs[].name	string	Output name
data.panel_outputs[].alias	string	Output alias
data.panel_outputs[].c_name	string	Channel name
data.panel_outputs[].g_name	string	Group name
data.panel_outputs[].router	bool	Whether route
data.panel_outputs[].enable	bool	Whether enabled
data.panel_outputs[].dis_s_ip	string	Discovery server IP address

data.panel_outputs[].vlan_id	int	VLAN ID
data.panel_outputs[].vlan_alias	string	VLAN alias
data.panel_outputs[].vlan_color	string	VLAN color
data.panel_outputs[].switcher_id	string	Switch serial number
data.panel_outputs[].connect_num	int	Connection count
data.panel_outputs[].lock	bool	Whether locked
data.panel_outputs[].resolution	string	Resolution
data.panel_outputs[].vb	string	Video bitrate
data.panel_outputs[].ab	string	Audio bitrate
data.panel_outputs[].input_id	int	Input ID
data.panel_outputs[].linked	bool	Whether linked
data.panel_outputs[].r_id	int	Route ID
data.panel_outputs[].r_loop	bool	Whether route loop
data.panel_outputs[].r_src_id	int	Route source ID
data.panel_outputs[].r_src_uuid	string	Route source UUID
data.panel_outputs[].r_src_progress	int	Route source progress
data.panel_outputs[].icon_type	string	Icon type
data.panel_outputs[].icon_name	string	Icon name
data.panel_outputs[].d_ip	string	Device IP address
data.other_inputs	array	Other input list
data.other_inputs[].id	int	Input ID
data.other_inputs[].name	string	Input name
data.other_inputs[].alias	string	Input alias
data.other_inputs[].c_name	string	Channel name
data.other_inputs[].g_name	string	Group name
data.other_inputs[].d_ip	string	Device IP address
data.other_inputs[].d_port	int	Device port
data.other_inputs[].url	string	Input URL
data.other_inputs[].enable	bool	Whether enabled
data.other_inputs[].dis_method	string	Discovery method
data.other_inputs[].dis_g_name	string	Discovery group name
data.other_inputs[].dis_d_ip	string	Discovered device IP address
data.other_inputs[].dis_s_ip	string	Discovery server IP address
data.other_inputs[].vlan_id	int	VLAN ID
data.other_inputs[].switcher_id	string	Switch serial number
data.other_inputs[].resolution	string	Resolution
data.other_inputs[].vb	string	Video bitrate
data.other_inputs[].ab	string	Audio bitrate

data.other_inputs[].multicast	bool	Whether multicast
data.other_inputs[].icon_type	string	Icon type
data.other_inputs[].icon_name	string	Icon name
data.panel_templates	array	Panel template list
data.rotationlists	array	Carousel list

4.7.2 Output Cross Input

Interface Description

Panel output-input crosspoint

Request URL

/v1/panel/cross/output/input

Request Method

POST

Request Parameter Example

```
1 {
2   "id": 1,
3   "output_id": 1,
4   "input_id": 2
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
output_id	int	Yes	Output ID
input_id	int	Yes	Input ID

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
}
```

```
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.7.3 Input Selection

Interface Description

Panel input selection

Request URL

/v1/panel/cross/input/select

Request Method

POST

Request Parameter Example

```
1 {
2   "id": 1,
3   "input_id": 1,
4   "select": true
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
input_id	int	Yes	Input ID
select	bool	Yes	Whether select all

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.8 Panel Configuration

4.8.1 Get All Input Configurations

Interface Description

Get all input configurations

Request URL

/v1/panel/cfg/input/all

Request Method

GET

Request Parameter Example

```
1 {  
2   "id": 1  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "cfg_inputs": [
6       {
7         "id": 1,
8         "name": "input1",
9         "alias": "输入 1",
10        "icon_id": 1,
11        "front_content": "input1"
12      }
13    ]
14  }
15 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.cfg_inputs	array	Input configuration list
data.cfg_inputs[].id	int	Input ID
data.cfg_inputs[].name	string	Input name
data.cfg_inputs[].alias	string	Input alias
data.cfg_inputs[].icon_id	int	Icon ID
data.cfg_inputs[].front_content	string	Frontend content

4.8.2 Get All Output Configurations

Interface Description

Get all output configurations

Request URL

/v1/panel/cfg/output/all

Request Method

GET

Request Parameter Example

```
1 {  
2   "id": 1  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": {  
5     "outputs": [  
6       {  
7         "id": 1,  
8         "name": "output1",  
9         "alias": "输出 1",  
10        "icon_id": 1,  
11        "front_content": "output1"  
12      }  
5     ]  
4   }  
3 }  
2 }  
1 }
```

```

13     ]
14   }
15 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.cfg_outputs	array	Output configuration list
data.cfg_outputs[].id	int	Output ID
data.cfg_outputs[].name	string	Output name
data.cfg_outputs[].alias	string	Output alias
data.cfg_outputs[].icon_id	int	Icon ID
data.cfg_outputs[].front_content	string	Frontend content

4.8.3 Apply Input Configuration

Interface Description

Apply input configuration

Request URL

/v1/panel/cfg/input/apply

Request Method

POST

Request Parameter Example

```

1 {
2   "id": 1,
3   "input_id": 1,
4   "icon_id": 1,
5   "front_content": "input1"

```

6 }

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
input_id	int	Yes	Panel input ID
icon_id	int	Yes	Icon ID
front_content	string	Yes	Frontend content

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.8.4 Apply Output Configuration

Interface Description

Apply output configuration

Request URL

/v1/panel/cfg/output/apply

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "output_id": 1,  
4   "icon_id": 1,  
5   "front_content": "output1"  
6 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
output_id	int	Yes	Panel output ID
icon_id	int	Yes	Icon ID
front_content	string	Yes	Frontend content

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.8.5 Set Physical Panel Enable Status

Interface Description

Set enable status

Request URL

/v1/panel/cfg/enable/set

Request Method

POST

Request Parameter Example

```
1 {  
2   "id": 1,  
3   "enable": true  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID
enable	bool	Yes	Enable status

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

4.8.6 Query Physical Panel Enable Status

Interface Description

Query enable status

Request URL

/v1/panel/cfg/enable/query

Request Method

GET

Request Parameter Example

```
1 {  
2   "id": 1  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Panel ID

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": {  
5     "enable": true  
6   }  
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success

message	string	Operation result description
data	object	Enable status information
data.enable	bool	Enable status

5 KLS Panel Management

5.1 KLS Input Source Discovery Management

5.1.1 KLS Add Server Discovery

Interface Description

Add KLS server input

Request URL

/v1/kls/input/server/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "serial_number": "KLSSERIAL001",  
3   "dis_g_name": "public",  
4   "dis_s_ip": "192.168.1.1"  
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
serial_number	string	Yes	Serial number
dis_g_name	string	Yes	Discovery group name
dis_s_ip	string	Yes	Discovery server IP address

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.1.2 KLS Add Manual Discovery

Interface Description

Manually add KLS input

Request URL

/v1/kls/input/manual/add

Request Method

POST

Request Parameter Example

```
1 {
2   "serial_number": "KLSSERIAL001",
3   "manuals": [
4     {
5       "dis_g_name": "public",
6       "dis_d_ip": "192.168.1.100"
7     }
9   ]
10 }
```

```
8   ]
9 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
serial_number	string	Yes	Serial number
manuals	array	Yes	Manual discovery information array
manuals[].dis_g_name	string	Yes	Discovery group name
manuals[].dis_d_ip	string	Yes	Discovered device IP address

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.1.3 KLS Add Auto Discovery

Interface Description

Auto-add KLS input

Request URL

/v1/kls/input/auto/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "serial_number": "KLSSERIAL001",  
3   "dis_g_name": "public"  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
serial_number	string	Yes	Serial number
dis_g_name	string	Yes	Discovery group name

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.1.4 KLS Delete Discovery Item

Interface Description

Delete discovered KLS input

Request URL

/v1/cls/input/discovery/delete

Request Method

POST

Request Parameter Example

```
1 {
2   "serial_number": "KLSSERIAL001",
3   "discoverys": [
4     {
5       "dis_method": "manual",
6       "dis_g_name": "public",
7       "dis_d_ip": "192.168.1.100",
8       "dis_s_ip": ""
9     }
10  ]
11 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
serial_number	string	Yes	Serial number
discoverys	array	Yes	Array of discovery data to delete
discoverys[].dis_method	string	Yes	Discovery method
discoverys[].dis_g_name	string	Yes	Discovery group name
discoverys[].dis_d_ip	string	Yes	Discovered device IP address
discoverys[].dis_s_ip	string	Yes	Discovery server IP address

Response Example

```
1 {
2   "code": 200,
```

```
3  "message": "Success",
4  "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.2 KLS Output Management

5.2.1 KLS Add Output

Interface Description

Add KLS output

Request URL

/v1/kls/output/add

Request Method

POST

Request Parameter Example

```
1 {
2   "outputs": [
3     {
4       "uuid": "485441684654546484654",
5       "name": "kls-output-1",
6       "alias": "KLS 输出 1",
7       "c_name": "通道 1",
8       "g_name": "group1",
9       "router": true,
```

```

10         "dis_s_ip": "192.168.1.1",
11         "vlan_id": 100,
12         "switcher_id": "00000000000000"
13     }
14 ]
15 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
outputs	array	Yes	Output list
outputs[].uuid	string	Yes	Output UUID
outputs[].name	string	Yes	Output name
outputs[].alias	string	Yes	Output alias
outputs[].c_name	string	Yes	Channel name
outputs[].g_name	string	Yes	Group name
outputs[].router	bool	Yes	Whether route
outputs[].dis_s_ip	string	Yes	Discovery server IP address
outputs[].vlan_id	int	Yes	VLAN ID
outputs[].switcher_id	string	Yes	Switch serial number

Response Example

```

1 {
2     "code": 200,
3     "message": "Success",
4     "data": {
5         "outputs": [
6             {
7                 "uuid": "485441684654546484654",
8                 "name": "kls-output-1",
9                 "alias": "KLS 输出 1",
10                "c_name": "通道 1",
11                "g_name": "group1",
12                "router": true,
13                "dis_s_ip": "192.168.1.1",

```

```

14         "vlan_id": 100,
15         "switcher_id": "00000000000000"
16     }
17 ],
18     "matrixs": []
19 }
20 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.outputs	array	Added output list
data.outputs[].uuid	string	Output UUID
data.outputs[].name	string	Output name
data.outputs[].alias	string	Output alias
data.outputs[].c_name	string	Channel name
data.outputs[].g_name	string	Group name
data.outputs[].router	bool	Whether route
data.outputs[].dis_s_ip	string	Discovery server IP address
data.outputs[].vlan_id	int	VLAN ID
data.outputs[].switcher_id	string	Switch serial number
data.matrixs	array	Matrix information list
data.matrixs.output_uuid	string	Output UUID
data.matrixs[].input_name	string	Input name
data.matrixs[].rl_uuid	string	Playlist UUID
data.matrixs[].rl_loop	bool	Whether playlist loops

5.2.2 KLS Delete Output

Interface Description

Delete KLS output

Request URL

/v1/cls/output/delete

Request Method

POST

Request Parameter Example

```
1 {  
2   "uuids": ["485441684654546484654"]  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
uuids	array	Yes	Array of output UUIDs to delete

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.2.3 KLS Modify Output

Interface Description

Modify KLS output

Request URL

/v1/kls/output/modify

Request Method

POST

Request Parameter Example

```
1 {  
2   "uuid": "485441684654546484654",  
3   "name": "new-name",  
4   "vlan_id": 200,  
5   "g_name": "new-group",  
6   "dis_s_ip": "192.168.1.2"  
7 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
uuid	string	Yes	Output UUID
name	string	Yes	Output name
vlan_id	int	Yes	VLAN ID
g_name	string	Yes	Group name
dis_s_ip	string	Yes	Discovery server IP address

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",
```

```
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.2.4 KLS Query Output

Interface Description

KLS query output

Request URL

/v1/cls/output/query

Request Method

GET

Request Parameter Example

```
1 {
2   "uuid": "485441684654546484654"
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
uuid	string	Yes	Output UUID

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
```

```

4    "data": {
5        "uuid": "485441684654546484654",
6        "name": "kls-output-1",
7        "alias": "KLS 输出 1",
8        "c_name": "通道 1",
9        "g_name": "group1",
10       "router": true,
11       "dis_s_ip": "192.168.1.1",
12       "vlan_id": 100,
13       "switcher_id": "00000000000000"
14   }
15 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Output information
data.uuid	string	Output UUID
data.name	string	Output name
data.alias	string	Output alias
data.c_name	string	Channel name
data.g_name	string	Group name
data.router	bool	Whether route
data.dis_s_ip	string	Discovery server IP address
data.vlan_id	int	VLAN ID
data.switcher_id	string	Switch serial number

5.2.5 KLS Get All Outputs

Interface Description

KLS get all outputs

Request URL

/v1/kls/output/query/all

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "outputs": [
6       {
7         "uuid": "485441684654546484654",
8         "name": "kls-output-1",
9         "alias": "KLS 输出 1",
10        "c_name": "通道 1",
11        "g_name": "group1",
12        "router": true,
13        "dis_s_ip": "192.168.1.1",
14        "vlan_id": 100,
15        "switcher_id": "00000000000000"
16      }
17    ]
18  }
19 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates

		success
message	string	Operation result description
data	object	Return data
data.outputs	array	Output list
data.outputs[].uuid	string	Output UUID
data.outputs[].name	string	Output name
data.outputs[].alias	string	Output alias
data.outputs[].c_name	string	Channel name
data.outputs[].g_name	string	Group name
data.outputs[].router	bool	Whether route
data.outputs[].dis_s_ip	string	Discovery server IP address
data.outputs[].vlan_id	int	VLAN ID
data.outputs[].switcher_id	string	Switch serial number

5.3 KLS Panel Basic Operations

5.3.1 KLS Add Panel

Interface Description

Add KLS panel

Request URL

/v1/kls/panel/add

Request Method

POST

Request Parameter Example

```
1 {
2   "name": "kls-panel-1",
3   "uuid": "panel-uuid-1",
4   "serial_number": "KLSSERIAL001"
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
name	string	Yes	Panel name
uuid	string	Yes	Panel UUID
serial_number	string	Yes	Serial number

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.3.2 KLS Delete Panel

Interface Description

Delete KLS panel

Request URL

/v1/cls/panel/delete

Request Method

POST

Request Parameter Example

```
1 {  
2   "uuid": "panel-uuid-1",
```

```
3   "serial_number": "KLSSERIAL001"
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
uuid	string	Yes	Panel UUID
serial_number	string	Yes	Serial number

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.3.3 KLS Switch Enabled

Interface Description

Enable KLS switch

Request URL

/v1/cls/panel/switcher/enable

Request Method

POST

Request Parameter Example

```
1 {
2   "serial_number": "KLSSERIAL001",
3   "discoverys": [
4     {
5       "dis_method": "auto",
6       "dis_g_name": "public",
7       "dis_d_ip": "192.168.1.100",
8       "dis_s_ip": "192.168.1.1"
9     }
10  ],
11  "inputs": [
12    {
13      "name": "input1",
14      "alias": "输入 1",
15      "channel_name": "通道 1",
16      "group_name": "group1",
17      "dev_ip": "192.168.1.100",
18      "dev_port": 5961,
19      "url": "192.168.1.100:5961",
20      "enable": true,
21      "dis_method": "auto",
22      "dis_g_name": "public",
23      "dis_d_ip": "192.168.1.100",
24      "dis_s_ip": "192.168.1.1",
25      "vlan_id": 1,
26      "switcher_id": "KLSSERIAL001"
27    }
28  ],
29  "outputs": [
30    {
31      "uuid": "output-uuid-1",
32      "name": "output1",
```

```

33     "alias": "输出 1",
34     "channel_name": "通道 1",
35     "group_name": "group1",
36     "router": true,
37     "codec_keepconn": 0,
38     "dis_s_ip": "192.168.1.1",
39     "locked": false,
40     "enable": true,
41     "vlan_id": 1,
42     "switcher_id": "KLSSERIAL001",
43     "multi_add": "239.0.0.1",
44     "multi_mask": "255.255.255.0",
45     "ttl_time": 16
46 }
47 ],
48 "panels": [
49     {
50         "uuid": "panel-uuid-1",
51         "name": "Default",
52         "icon": ""
53     }
54 ],
55 "panel_inputs": [
56     {
57         "panel_uuid": "panel-uuid-1",
58         "input": {
59             "name": "input1",
60             "alias": "输入 1",
61             "channel_name": "通道 1",
62             "group_name": "group1",
63             "dev_ip": "192.168.1.100",
64             "dev_port": 5961,
65             "url": "192.168.1.100:5961",
66             "enable": true,

```

```

67         "dis_method": "auto",
68         "dis_g_name": "public",
69         "dis_d_ip": "192.168.1.100",
70         "dis_s_ip": "192.168.1.1",
71         "vlan_id": 1,
72         "switcher_id": "KLSSERIAL001"
73     }
74 }
75 ],
76 "panel_outputs": [
77     {
78         "panel_uuid": "panel-uuid-1",
79         "output": {
80             "uuid": "output-uuid-1",
81             "name": "output1",
82             "alias": "输出 1",
83             "channel_name": "通道 1",
84             "group_name": "group1",
85             "router": true,
86             "codec_keepconn": 0,
87             "dis_s_ip": "192.168.1.1",
88             "locked": false,
89             "enable": true,
90             "vlan_id": 1,
91             "switcher_id": "KLSSERIAL001",
92             "multi_add": "239.0.0.1",
93             "multi_mask": "255.255.255.0",
94             "ttl_time": 16
95         }
96     }
97 ],
98 "panel_matrixs": [
99     {
100         "panel_uuid": "panel-uuid-1",

```

```

101         "output_uuid": "output-uuid-1",
102         "input_name": "input1",
103         "rl_uuid": "",
104         "rl_loop": false
105     }
106 ]
107 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
serial_number	string	Yes	Serial number
discoverys	array	Yes	Discovery list
discoverys[].dis_method	string	Yes	Discovery method
discoverys[].dis_g_name	string	Yes	Discovery group name
discoverys[].dis_d_ip	string	Yes	Discovery IP
discoverys[].dis_s_ip	string	Yes	Discovery switch IP
inputs	array	Yes	Input list
inputs[].name	string	Yes	Input name
inputs[].alias	string	Yes	Input alias
inputs[].channel_name	string	Yes	Input channel name
inputs[].group_name	string	Yes	Input group name
inputs[].dev_ip	string	Yes	Input IP
inputs[].dev_port	int	Yes	Input port
inputs[].url	string	Yes	Input URL
inputs[].enable	bool	Yes	Input enable status
inputs[].dis_method	string	Yes	Input discovery method
inputs[].dis_g_name	string	Yes	Input discovery group name
inputs[].dis_d_ip	string	Yes	Input discovery IP
inputs[].dis_s_ip	string	Yes	Input discovery switch IP
inputs[].vlan_id	int	Yes	VLAN ID
inputs[].switcher_id	string	Yes	Serial number
outputs	array	Yes	Output list
outputs[].uuid	string	Yes	Output UUID
outputs[].name	string	Yes	Output name

outputs[].alias	string	Yes	Output alias
outputs[].channel_name	string	Yes	Output channel name
outputs[].group_name	string	Yes	Output group name
outputs[].router	bool	Yes	Output route
outputs[].codec_keepconn	int	Yes	Output keep-alive
outputs[].dis_s_ip	string	Yes	Output discovery switch IP
outputs[].locked	bool	Yes	Output lock
outputs[].enable	bool	Yes	Output enable status
outputs[].vlan_id	int	Yes	VLAN ID
outputs[].switcher_id	string	Yes	Serial number
outputs[].multi_add	string	Yes	Multicast address
outputs[].multi_mask	string	Yes	Multicast mask
outputs[].ttl_time	int	Yes	TTL time
panels	array	Yes	Panel list
panels[].uuid	string	Yes	Panel UUID
panels[].name	string	Yes	Panel name
panels[].Icon	string	Yes	Panel icon
panel_inputs	array	Yes	Panel input list
panel_inputs[].panel_uuid	string	Yes	Panel UUID
panel_inputs[].input	object	Yes	Input object
panel_outputs	array	Yes	Panel output list
panel_outputs[].panel_uuid	string	Yes	Panel UUID
panel_outputs[].output	object	Yes	Output object
panel_matrixs	array	Yes	Panel matrix list
panel_matrixs[].panel_uuid	string	Yes	Panel UUID
panel_matrixs[].output_uuid	string	Yes	Output UUID
panel_matrixs[].input_name	string	Yes	Input name
panel_matrixs[].rl_uuid	string	Yes	Playlist UUID
panel_matrixs[].rl_loop	bool	Yes	Playlist loop

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.3.4 KLS Delete Device

Interface Description

Delete KLS device

Request URL

/v1/kls/panel/device/delete

Request Method

POST

Request Parameter Example

```
1 {  
2   "serial_number": "KLSSERIAL001"  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
serial_number	string	Yes	Serial number

Response Example

```
1 {  
2   "code": 200,  
3   "message": "删除 KLS 设备成功",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.4 KLS Panel Input Management

5.4.1 KLS Add Panel Input

Interface Description

Add KLS panel input

Request URL

/v1/kls/panel/input/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "panel_uuid": "panel-uuid-1",  
3   "inputs": [  
4     {  
5       "name": "input1",  
6       "alias": "输入 1",  
7       "c_name": "通道 1",  
8       "g_name": "group1",  
9       "d_ip": "192.168.1.100",  
10      "d_port": 5961,  
11      "url": "192.168.1.100:5961",  
12      "enable": true,
```

```

13         "dis_method": "auto",
14         "dis_g_name": "public",
15         "dis_d_ip": "192.168.1.100",
16         "dis_s_ip": "192.168.1.1",
17         "vlan_id": 1,
18         "switcher_id": "KLSSERIAL001"
19     }
20 ]
21 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_uuid	string	Yes	Panel UUID
inputs	array	Yes	Input object array
inputs[].name	string	Yes	Input name
inputs[].alias	string	Yes	Input alias
inputs[].c_name	string	Yes	Input channel name
inputs[].g_name	string	Yes	Input group name
inputs[].d_ip	string	Yes	Input IP
inputs[].d_port	int	Yes	Input port
inputs[].url	string	Yes	Input URL
inputs[].enable	bool	Yes	Input enable status
inputs[].dis_method	string	Yes	Input discovery method
inputs[].dis_g_name	string	Yes	Input discovery group name
inputs[].dis_d_ip	string	Yes	Input discovery IP
inputs[].dis_s_ip	string	Yes	Input discovery switch IP
inputs[].vlan_id	int	Yes	Input VLAN ID
inputs[].switcher_id	string	Yes	Serial number

Response Example

```

1 {
2     "code": 200,
3     "message": "Success",

```

```
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.4.2 KLS Remove Panel Input

Interface Description

Remove KLS panel input

Request URL

/v1/cls/panel/input/remove

Request Method

POST

Request Parameter Example

```
1 {
2   "panel_uuid": "panel-uuid-1",
3   "input_names": ["input1", "input2"]
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_uuid	string	Yes	Panel UUID
input_names	array	Yes	Input name array

Response Example

```
1 {
```

```
2  "code": 200,  
3  "message": "Success",  
4  "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.5 KLS Panel Output Management

5.5.1 KLS Add Panel Output

Interface Description

Add KLS panel output

Request URL

/v1/kls/panel/output/add

Request Method

POST

Request Parameter Example

```
1 {  
2  "panel_uuid": "panel-uuid-1",  
3  "output_uuids": ["output-uuid-1", "output-uuid-2"]  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_uuid	string	Yes	Panel UUID

output_uuids	array	Yes	Output UUID array
--------------	-------	-----	-------------------

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.5.2 KLS Remove Panel Output

Interface Description

Remove KLS panel output

Request URL

/v1/cls/panel/output/remove

Request Method

POST

Request Parameter Example

```
1 {
2   "panel_uuid": "panel-uuid-1",
3   "output_uuids": ["output-uuid-1"]
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_uuid	string	Yes	Panel UUID
output_uuids	array	Yes	Output UUID array

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.6 KLS Panel Cross Switching

5.6.1 KLS Get Panel Details

Interface Description

Get KLS panel details

Request URL

/v1/kls/panel/detail

Request Method

GET

Request Parameter Example

```
1 {
2   "panel_uuid": "panel-uuid-1"
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_uuid	string	Yes	Panel UUID

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "uuid": "panel-uuid-1",
6     "panel_inputs": [
7       {
8         "name": "input1",
9         "alias": "输入 1",
10        "c_name": "通道 1",
11        "g_name": "group1",
12        "d_ip": "192.168.1.100",
13        "d_port": 5961,
14        "url": "192.168.1.100:5961",
15        "enable": true,
16        "dis_method": "auto",
17        "dis_g_name": "public",
18        "dis_d_ip": "192.168.1.100",
19        "dis_s_ip": "192.168.1.1",
20        "vlan_id": 1,
21        "switcher_id": "KLSSERIAL001",
22        "resolution": "1920x1080@60HZ",
23        "vb": "50.00Mbps",
```

```

24         "ab": "1.50Mbps",
25         "multicast": true
26     }
27 ],
28 "panel_outputs": [
29     {
30         "uuid": "output-uuid-1",
31         "name": "output1",
32         "alias": "输出 1",
33         "c_name": "通道 1",
34         "g_name": "group1",
35         "router": true,
36         "enable": true,
37         "dis_s_ip": "192.168.1.1",
38         "vlan_id": 1,
39         "vlan_alias": "VLAN1",
40         "vlan_color": "#0000FF",
41         "switcher_id": "KLSSERIAL001",
42         "connect_num": 0,
43         "lock": false,
44         "resolution": "1920x1080@60HZ",
45         "vb": "50.00Mbps",
46         "ab": "1.50Mbps",
47         "input_name": "input1",
48         "linked": false,
49         "r_uuid": "",
50         "r_loop": false,
51         "r_src_id": "",
52         "r_src_uuid": "",
53         "r_src_progress": 0,
54         "d_ip": "192.168.1.200"
55     }
56 ],
57 "other_inputs": [],

```

```

58     "panel_templates": [],
59     "rotationlists": []
60 }
61 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Panel details
data.uuid	string	Panel UUID
data.panel_inputs	array	Panel input list
data.panel_inputs[].name	string	Input name
data.panel_inputs[].alias	string	Input alias
data.panel_inputs[].c_name	string	Channel name
data.panel_inputs[].g_name	string	Group name
data.panel_inputs[].d_ip	string	Device IP address
data.panel_inputs[].d_port	int	Device port
data.panel_inputs[].url	string	Input URL
data.panel_inputs[].enable	bool	Whether enabled
data.panel_inputs[].dis_method	string	Discovery method
data.panel_inputs[].dis_g_name	string	Discovery group name
data.panel_inputs[].dis_d_ip	string	Discovered device IP address
data.panel_inputs[].dis_s_ip	string	Discovery server IP address
data.panel_inputs[].vlan_id	int	VLAN ID
data.panel_inputs[].switcher_id	string	Serial number
data.panel_inputs[].resolution	string	Resolution
data.panel_inputs[].vb	string	Video bitrate
data.panel_inputs[].ab	string	Audio bitrate
data.panel_inputs[].multicast	bool	Whether multicast
data.panel_outputs	array	Panel output list
data.panel_outputs[].uuid	string	Output UUID
data.panel_outputs[].name	string	Output name
data.panel_outputs[].alias	string	Output alias
data.panel_outputs[].c_name	string	Channel name
data.panel_outputs[].g_name	string	Group name

data.panel_outputs[].router	bool	Whether route
data.panel_outputs[].enable	bool	Whether enabled
data.panel_outputs[].dis_s_ip	string	Discovery server IP address
data.panel_outputs[].vlan_id	int	VLAN ID
data.panel_outputs[].vlan_alias	string	VLAN alias
data.panel_outputs[].vlan_color	string	VLAN color
data.panel_outputs[].switcher_id	string	Serial number
data.panel_outputs[].connect_num	int	Connection count
data.panel_outputs[].lock	bool	Whether locked
data.panel_outputs[].resolution	string	Resolution
data.panel_outputs[].vb	string	Video bitrate
data.panel_outputs[].ab	string	Audio bitrate
data.panel_outputs[].input_name	string	Input name
data.panel_outputs[].linked	bool	Whether linked
data.panel_outputs[].r_uuid	string	Playlist UUID
data.panel_outputs[].r_loop	bool	Whether playlist loops
data.panel_outputs[].r_src_id	string	Carousel source UUID
data.panel_outputs[].r_src_uuid	string	Carousel source UUID
data.panel_outputs[].r_src_progress	int	Carousel source progress
data.panel_outputs[].d_ip	string	Device IP address
data.other_inputs	array	Other input list
data.panel_templates	array	Panel template list
data.rotationlists	array	Carousel list

5.6.2 KLS Output Cross Input

Interface Description

KLS panel output-input crosspoint

Request URL

/v1/kls/panel/cross/output/input

Request Method

POST

Request Parameter Example

```
1 {  
2   "panel_uuid": "panel-uuid-1",  
3   "output_uuid": "output-uuid-1",  
4   "input_name": "input1"  
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_uuid	string	Yes	Panel UUID
output_uuid	string	Yes	Output UUID
input_name	string	Yes	Input name

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

5.6.3 KLS Select All Inputs

Interface Description

KLS panel input select

Request URL

/v1/kls/panel/cross/input/select

Request Method

POST

Request Parameter Example

```
1 {  
2   "panel_uuid": "panel-uuid-1",  
3   "input_name": "input1",  
4   "select": true  
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_uuid	string	Yes	Panel UUID
input_name	string	Yes	Input name
select	bool	Yes	Whether select all

Response Example

```
1 {  
2   "code": 200,  
3   "message": "KLS 面板输入选择成功",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

6 KLS Binding Management

6.1 Bind KLS

Interface Description

Bind KLS

Request URL

/v1/bindkls/tobind

Request Method

POST

Request Parameter Example

```
1 {
2   "ip": "192.168.1.100",
3   "ifname": "1", // eth1.1
4   "location": 1
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ip	string	Yes	IP address
ifname	string	Yes	Interface name
location	int	Yes	Position ID

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

6.2 Unbind KLS

Interface Description

Unbind KLS

Request URL

/v1/bindkls/unbind

Request Method

GET

Request Parameter Example

```
1 {  
2   "id": 1  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	Yes	Position ID

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

6.3 Query KLS VLAN

Interface Description

Get KLS VLAN

Request URL

/v1/bindkls/vlan

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "获取 KLS VLAN 成功",  
4   "data": ["eth1.1"]  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
----------------	----------------	-------------

code	int	Status code; 200 indicates success
message	string	Operation result description
data	array	NIC name list

6.4 Get Binding Information

Interface Description

Get binding information

Request URL

/v1/bindkls/bindinfo

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "获取绑定信息成功",
4   "data": {
5     "status": 1,
6     "bind_list": [
7       {
8         "id": 1,
9         "ip": "192.168.1.100",
10        "ifname": "1", // eth1.1
```

```

11         "bind_status": 1
12     }
13 ]
14 }
15 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Binding information
data.status	int	Binding status: 1 - Bound, 2 - Binding in progress, 3 - Binding failed, 4 - Unbound
data.bind_list	array	Binding list
data.bind_list[].id	int	Position ID
data.bind_list[].ip	string	IP address
data.bind_list[].ifname	string	Interface name
data.bind_list[].bind_status	int	Binding status: 1 - Bound, 2 - Binding in progress, 3 - Binding failed, 4 - Unbound

6.5 Get Binding Status

Interface Description

Get binding status

Request URL

/v1/bindkls/status

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": {  
5     "status": 1  
6   }  
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Binding status information
data.status	int	Binding status: 1 - Bound, 2 - Binding in progress, 3 - Binding failed, 4 - Unbound

7 Icon Management

7.1 Upload Icon

Interface Description

Upload icon

Request URL

/v1/icon/upload

Request Method

POST

Request Parameter Example

```
1 {
2   "index": 1,
3   "icon_file": "文件流",
4   "icon_name": "custom_icon.png",
5   "mime_type": "image/png",
6   "width": 64,
7   "height": 64
8 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
index	int	Yes	Icon index
icon_file	file	Yes	Icon file
icon_name	string	No	Icon name
mime_type	string	No	MIME type
width	int	No	Icon width
height	int	No	Icon height

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
----------------	----------------	-------------

code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

7.2 Modify Icon

Interface Description

Modify custom icon

Request URL

/v1/icon/modify

Request Method

POST

Request Parameter Example

```
1 {
2   "ids": [1, 2, 3]
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ids	array	Yes	Icon ID array

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
----------------	----------------	-------------

code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

7.3 Get All Icons

Interface Description

Get all icons

Request URL

/v1/icon/all

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "official_icons": [
6       {
7         "id": 1,
8         "name": "icon1.png",
9         "type": "official"
10      },
```

```

11      {
12          "id": 2,
13          "name": "icon2.png",
14          "type": "official"
15      }
16  ],
17  "custom_icons": [
18      {
19          "id": 3,
20          "name": "icon3.png",
21          "type": "custom"
22      }
23  ]
24  }
25 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Icon list
data.official_icons	array	Official Icon Array
data.official_icons[].id	int	Icon ID
data.official_icons[].name	string	Icon name
data.official_icons[].type	string	Icon type: official/custom
data.custom_icons	array	Custom icon array
data.custom_icons[].id	int	Icon ID
data.custom_icons[].name	string	Icon name
data.custom_icons[].type	string	Icon type: official/custom

7.4 Notify Icon Update

Interface Description

Notify icon update

Request URL

/v1/icon/notify/update

Request Method

POST

Request Parameter Example

```
1 {  
2   "icon_name": "custom_icon.png"  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
icon_name	string	Yes	Icon name

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

8 Front Panel Operation

8.1 Input Preview

Interface Description

Input preview

Request URL

/v1/front/input/preview

Request Method

POST

Request Parameter Example

```
1 {  
2   "input_id": 1,  
3   "enable": true  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
input_id	int	Yes	Input ID
enable	bool	Yes	Whether preview is enabled

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

8.2 Execute Switch

Interface Description

Input-output switch

Request URL

/v1/front/input/output/take

Request Method

POST

Request Parameter Example

```
1 {  
2   "panel_id": 1,  
3   "input_id": 1,  
4   "output_ids": [1, 2, 3]  
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_id	int	Yes	Front panel ID
input_id	int	Yes	Input ID
output_ids	array	Yes	Output ID array

Response Example

```
1 {  
2   "code": 200,
```

```
3  "message": "Success",
4  "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

8.3 Query Front Panel

Interface Description

Query front panel

Request URL

/v1/front/panel/query

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2  "code": 200,
3  "message": "Success",
4  "data": {
```

```

5      "panels": [
6          {
7              "id": 1,
8              "name": "panel1",
9              "alias": "panel1"
10         },
11         {
12             "id": 2,
13             "name": "panel2",
14             "alias": "panel2"
15         }
16     ]
17 }
18 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Data object
data.panels	array	Panel array
data.panels[].id	int	Panel ID
data.panels[].name	string	Panel name
data.panels[].alias	string	Alias name

8.4 Get Front Panel Details

Interface Description

Get front panel details

Request URL

/v1/front/panel/detail

Request Method

GET

Request Parameter Example

```
1 {  
2   "panel_id": 1  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
panel_id	int	Yes	Front panel ID

Response Example

```
1 {  
2   "code": 200,  
3   "message": "Success",  
4   "data": {  
5     "id": 1,  
6     "name": "panel1",  
7     "alias": "panel1",  
8     "panel_inputs": [  
9       {  
10        "id": 1,  
11        "name": "input1",  
12        "alias": "input1",  
13        "icon_name": "input1",  
14        "front_content": "input1"  
15      },  
16      {  
17        "id": 2,  
18        "name": "input2",  
19        "alias": "input2",  
20        "icon_name": "input2",
```

```

21         "front_content": "input2"
22     }
23 ],
24     "panel_outputs": [
25         {
26             "id": 1,
27             "name": "output1",
28             "alias": "output1",
29             "lock": false,
30             "input_id": 1,
31             "linked": false,
32             "icon_name": "output1",
33             "front_content": "output1 "
34         }
35     ]
36 }
37 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Data object
data.id	int	Panel ID
data.name	string	Panel name
data.alias	string	Alias name
data.panel_inputs	array	Input array
data.panel_inputs[].id	int	Input ID
data.panel_inputs[].name	string	Input name
data.panel_inputs[].alias	string	Input alias
data.panel_inputs[].icon_name	string	Input icon name
data.panel_inputs[].front_content	string	Input frontend content
data.panel_outputs	array	Output array
data.panel_outputs[].id	int	Output ID

data.panel_outputs[].name	string	Output name
data.panel_outputs[].alias	string	Output alias
data.panel_outputs[].lock	bool	Whether output is locked
data.panel_outputs[].input_id	int	Input ID
data.panel_outputs[].linked	bool	Whether input is linked
data.panel_outputs[].icon_name	string	Output icon name
data.panel_outputs[].front_content	string	Output frontend content

8.5 Stop NDI

Interface Description

Stop NDI

Request URL

/v1/front/ndi/stop

Request Method

POST

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
----------------	----------------	-------------

code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

9 VLAN Configuration

9.1 Get All VLAN Configuration

Interface Description

Get all VLAN configurations

Request URL

/v1/vlan/cfg/all

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "vlan_cfgs": [
6       {
7         "id": 1,
8         "type": true,
```

```
9         "name": "NDI",
10        "priority": "Highest",
11        "description": "Default VLAN NDI",
12        "vlans": []
13    }
14 ]
15 }
16 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Data object
data.id	int	Panel ID
data.name	string	Panel name
data.type	bool	Type (whether official)
data.priority	string	Priority
data.description	string	Description
data.vlans	array	VLAN list

9.2 Modify VLAN Configuration

Interface Description

Modify VLAN configuration

Request URL

/v1/vlan/cfg/modify

Request Method

POST

Request Parameter Example

```
1 {
2   "id": 1,
3   "vlan_ids": "1"
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	int	True	ID
vlan_ids	array	True	VLAN ID list

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

10 Topology Management

10.1 Query Topology Structure

Interface Description

Query topology structure

Request URL

/v1/topology/query

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "nodes": [
6       {
7         "id": "18:C3:F4:2F:FB:EE",
8         "name": "AVX24-4-chen666",
9         "type": "kiloview",
10        "label": "AVX24-4",
11        "parent_id": "",
12        "shape": "router",
13        "image": "./static/img/AVX24_4.png",
14        "valid": false,
15        "x": 320,
16        "y": 560,
17        "connection_port": 0
18      },
19      {
20        "id": "V:18:C3:F4:2F:FB:EE-3",
```

```

21         "name": "unknownSwitch",
22         "type": "other",
23         "label": "unknown",
24         "parent_id": "18:C3:F4:2F:FB:EE",
25         "shape": "device",
26         "image": "./static/img/unknown_switch.png",
27         "valid": false,
28         "x": 325,
29         "y": 690,
30         "connection_port": 0
31     },
32     {
33         "id": "18:C3:F4:2F:FB:F3",
34         "name": "AVX24-4-lion",
35         "type": "kiloview",
36         "label": "AVX24-4",
37         "parent_id": "V:18:C3:F4:2F:FB:EE-3",
38         "shape": "routerTerminal",
39         "image": "./static/img/AVX24_4.png",
40         "valid": false,
41         "x": 175,
42         "y": 820,
43         "connection_port": 1
44     },
45     {
46         "id": "18:C3:F4:2F:FB:E7",
47         "name": "AVX24-4",
48         "type": "kiloview",
49         "label": "AVX24-4",
50         "parent_id": "V:18:C3:F4:2F:FB:EE-3",
51         "shape": "routerTerminal",
52         "image": "./static/img/AVX24_4.png",
53         "valid": false,
54         "x": 325,

```

```

55         "y": 820,
56         "connection_port": 1
57     },
58     {
59         "id": "10:78:56:34:12:61",
60         "name": "AVX24-4",
61         "type": "kiloview",
62         "label": "AVX24-4",
63         "parent_id": "V:18:C3:F4:2F:FB:EE-3",
64         "shape": "routerTerminal",
65         "image": "./static/img/AVX24_4.png",
66         "valid": false,
67         "x": 475,
68         "y": 820,
69         "connection_port": 19
70     },
71     {
72         "id": "18:C3:F4:25:6A:D2",
73         "name": "unknown",
74         "type": "other",
75         "label": "unknown",
76         "parent_id": "18:C3:F4:2F:FB:EE",
77         "shape": "device",
78         "image": "./static/img/unknown.png",
79         "valid": false,
80         "x": 625,
81         "y": 690,
82         "connection_port": 0
83     }
84 ],
85 "wires": [
86     {
87         "id": "18:C3:F4:2F:FB:EE:3",
88         "shape": "edge",

```

```

89         "color": ""
90     },
91     {
92         "id": "V:18:C3:F4:2F:FB:EE-3:35",
93         "shape": "edge",
94         "color": ""
95     },
96     {
97         "id": "V:18:C3:F4:2F:FB:EE-3:42",
98         "shape": "edge",
99         "color": ""
100    },
101    {
102        "id": "V:18:C3:F4:2F:FB:EE-3:85",
103        "shape": "edge",
104        "color": ""
105    },
106    {
107        "id": "18:C3:F4:2F:FB:EE:11",
108        "shape": "edge",
109        "color": ""
110    }
111 ]
112 }
113 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.nodes	array	Node list

data.nodes[].id	string	Node ID
data.nodes[].name	string	Node name
data.nodes[].type	string	Node type
data.nodes[].label	string	Node label
data.nodes[].parent_id	string	Parent node ID
data.nodes[].shape	string	Node shape
data.nodes[].image	string	Node image
data.nodes[].valid	bool	Whether node is valid
data.nodes[].x	int	Node X coordinate
data.nodes[].y	int	Node Y coordinate
data.nodes[].connection_port	int	Node connection port
data.wires	array	Line list
data.wires[].id	string	Line ID
data.wires[].shape	string	Line shape
data.wires[].color	string	Line color

10.2 Query Topology Status

Interface Description

Query topology status

Request URL

/v1/topology/status/query

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "data": {
3     "nodes": [
4       {
5         "id": "18:C3:F4:25:6A:D2",
6         "isMain": false,
7         "status": "online",
8         "alarm_status": "",
9
10        "ports": []
11      },
12      {
13        "id": "10:78:56:34:12:61",
14        "isMain": false,
15        "status": "online",
16        "alarm_status": "",
17
18        "ports": []
19      },
20      {
21        "id": "18:C3:F4:2F:FB:F3",
22        "isMain": false,
23        "status": "online",
24        "alarm_status": "",
25
26        "ports": []
27      },
28      {
29        "id": "18:C3:F4:2F:FB:E7",
```

```

33         "isMain": false,
34         "status": "online",
35         "alarm_status": "",
36
37         "ports": []
38
39     },
40     {
41         "id": "V:18:C3:F4:2F:FB:EE-3",
42         "isMain": false,
43         "status": "online",
44         "alarm_status": "",
45
46         "ports": []
47
48     },
49     {
50         "id": "18:C3:F4:2F:FB:EE",
51         "isMain": true,
52         "status": "online",
53         "alarm_status": "",
54         "ports": [
55             {
56                 "port_id": "3",
57                 "status": "online"
58             },
59             {
60                 "port_id": "11",
61                 "status": "online"
62             }
63         ]
64     }
65 ],
66 "edges": [

```

```

67     {
68         "id": "18:C3:F4:2F:FB:EE:11",
69         "fill": "",
70         "status": "online",
71         "source": {
72             "is_dev": false,
73             "cell": "18:C3:F4:2F:FB:EE",
74             "port": "11"
75         },
76         "target": {
77             "is_dev": true,
78             "cell": "18:C3:F4:25:6A:D2",
79             "port": ""
80         }
81     },
82     {
83         "id": "18:C3:F4:2F:FB:EE:3",
84         "fill": "",
85         "status": "online",
86         "source": {
87             "is_dev": false,
88             "cell": "18:C3:F4:2F:FB:EE",
89             "port": "3"
90         },
91         "target": {
92             "is_dev": true,
93             "cell": "V:18:C3:F4:2F:FB:EE-3",
94             "port": ""
95         }
96     },
97     {
98         "id": "V:18:C3:F4:2F:FB:EE-3:85",
99         "fill": "",
100        "status": "online",

```

```

101         "source": {
102             "is_dev": false,
103             "cell": "V:18:C3:F4:2F:FB:EE-3",
104             "port": "85"
105         },
106         "target": {
107             "is_dev": true,
108             "cell": "10:78:56:34:12:61",
109             "port": ""
110         }
111     },
112     {
113         "id": "V:18:C3:F4:2F:FB:EE-3:35",
114         "fill": "",
115         "status": "online",
116         "source": {
117             "is_dev": false,
118             "cell": "V:18:C3:F4:2F:FB:EE-3",
119             "port": "35"
120         },
121         "target": {
122             "is_dev": true,
123             "cell": "18:C3:F4:2F:FB:F3",
124             "port": ""
125         }
126     },
127     {
128         "id": "V:18:C3:F4:2F:FB:EE-3:42",
129         "fill": "",
130         "status": "online",
131         "source": {
132             "is_dev": false,
133             "cell": "V:18:C3:F4:2F:FB:EE-3",
134             "port": "42"

```

```

135         },
136         "target": {
137             "is_dev": true,
138             "cell": "18:C3:F4:2F:FB:E7",
139             "port": ""
140         }
141     }
142 ]
143 }
144 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	Return data
data.nodes	array	Node list
data.nodes[].id	string	Node ID
data.nodes[].isMain	bool	Whether primary node
data.nodes[].status	string	Node status
data.nodes[].alarm_status	string	Alarm status
data.nodes[].ports	array	Port list
data.nodes[].ports[].port_id	string	Port ID
data.nodes[].ports[].status	string	Port status
data.wires	array	Line list
data.wires[].id	string	Line ID
data.wires[].fill	string	Line fill
data.wires[].status	string	Line status
data.wires[].source	object	Line source
data.wires[].source.is_dev	bool	Whether it is a device
data.wires[].source.cell	string	Source device ID
data.wires[].source.port	string	Source port
data.wires[].target	object	Line target
data.wires[].target.is_dev	bool	Whether it is a device

data.wires[].target.is_dev	bool	Whether it is a device
data.wires[].target.cell	string	Target device ID
data.wires[].target.port	string	Target port

10.3 Query Device Information

Interface Description

Query topology device details.

Request URL

/v1/topology/dev/info/query

Request Method

POST

Request Parameter Example

```
1 {
2     "data": {
3         "port_id": 13,
4         "dev_mac": ""
5     }
6 }
7
8 或者
9 {
10    "data": {
11        "port_id": 0,
12        "dev_mac": "18:C3:F4:20:64:E1"
13    }
14 }
```

Request Parameter Description

The following only describes request body fields (excluding HTTP headers).

Description conventions

1. data is an object type (corresponding to topologyDeviceReq), its sub-fields are separately listed in the table below.
2. port_id and dev_mac: in practice, usually only one is used as the query key – when querying by port, fill in a valid port_id and dev_mac can be an empty string; when querying by MAC, fill in a valid dev_mac and port_id can be 0.

data (topologyDeviceReq) object field description (data.port_id, data.dev_mac)

Parameter name	Parameter type	Whether required	Description
port_id	Number	True	Switch port ID; pass 0 when querying by MAC
dev_mac	String	True	Device MAC (uppercase/lowercase subject to gateway implementation); can be empty string "" when querying by port

Response Example

```

1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "switch_id": 0,
6     "port_id": 13,
7     "dev_name": "18:C3:F4:20:64:E1",
8     "dev_mac": "18:C3:F4:20:64:E1",
9     "dev_ip": "192.168.3.225",
10    "is_static_ip": false,
11    "image": "./static/img/N50.png",
12    "rx": {
13      "packets": "2",
14      "bytes": "1203",
15      "error": "0",
16      "loss": "0"

```

```

17     },
18     "tx": {
19         "packets": "1",
20         "bytes": "491",
21         "error": "0",
22         "loss": "0"
23     },
24     "media": [],
25     "model": "N50",
26     "sn": "32021120064E1",
27     "software_version": "2.43.0011.kdds.0297",
28     "hardware_version": "",
29     "company": "Kiloview",
30     "alias": "N50-32021120064E1",
31     "sw_serial_number": "18c3f425640e",
32     "diagnose_status": 0,
33     "parent_id": "",
34     "cpu_usage": "16.20%",
35     "disk_usage": "48.75%",
36     "mem_usage": "21.44%",
37     "temperature": "",
38     "server_count": 1,
39     "dev_proxy_url": "",
40     "switch_link_status": "online"
41 }
42 }

```

Response Parameter Description

The top-level code and message are consistent with other interfaces: code is the business code (e.g., 200 indicates success), message is the description string. The following only describes data (topologyDevInfo) and its nested object fields.

Description conventions

1. Object fields in data (rx, tx): first list sub-fields in a separate table in this section; in the final "data object field description" table, the corresponding row writes "see xxx" without repeating the

expansion.

2. media: an object array, element type is DevMediaInfo, described separately in "media array element field description".
3. rx / tx: identical structure (both are SwitchPortRate), sharing one "data.rx / data.tx object field description".

data.rx / data.tx object field description (data.rx, data.tx, both of type SwitchPortRate)

Parameter name	Parameter type	Description
packets	Number / String	Packet count (may be JSON number or string depending on serialization implementation)
bytes	Number / String	Byte count
error	Number / String	Error packet count
loss	Number / String	Discarded packet count

media array element field description (data.media[], type DevMediaInfo)

Parameter name	Parameter type	Description
ndi_src	String	NDI source identifier
url	String	Media URL or address
video_codec	String	Video encoding format
height	Number	Video height (pixels)
width	Number	Video width (pixels)
audio_codec	String	Audio encoding format
sampling	Number	Sampling rate related parameters
channel_number	Number	Channel count
bitrate	Number	Bitrate (float)
framerate	Number	Frame rate

data object field description (topologyDevInfo)

Parameter name	Parameter type	Description
switch_id	Number	Switch identifier; usually 0 in standalone scenarios
port_id	Number	Port ID where the device is located
dev_name	String	Topology/device name (often

		related to MAC or display name)
dev_mac	String	Device MAC
dev_ip	String	Device IP
is_static_ip	Boolean	Whether it is a static IP
image	String	Device icon or image path
rx	Object	Receive direction traffic statistics, see "data.rx / data.tx object field description"
tx	Object	Send direction traffic statistics, see "data.rx / data.tx object field description"
media	Array	Media information list, elements see "media array element field description"
model	String	Device model
sn	String	Device serial number
software_version	String	Software version
hardware_version	String	Hardware version
company	String	Vendor
alias	String	Device alias (for display)
sw_serial_number	String	Serial number of this switch or associated gateway serial number (subject to actual return value)
diagnose_status	Number	Diagnostic status: 0 - No diagnostic, 1 - Diagnosing
parent_id	String	Parent node or topology association ID; empty string if unavailable
cpu_usage	String	CPU usage display string, e.g., 16.20%
disk_usage	String	Disk usage display string
mem_usage	String	Memory usage display string
temperature	String	Temperature display string; can be empty string when no data available
server_count	Number	Number of servers reporting

		simultaneously
dev_proxy_url	String	Reverse proxy URL for accessing device web; empty string if unavailable
switch_link_status	String	Link status with the switch, e.g., online, offline, etc.

11 Network Diagnostic

11.1 Diagnostic Management

11.1.1 KLS Cross-Device Start End-to-End Diagnostic

Interface Description

Start KLS end-to-end diagnostic task (KLS issued to AVX)

Request URL

/v1/diagnose/task/rfppadd

Request Method

POST

Request Parameter Example

Example with only required parameters (using defaults):

```
1 {
2   "task_id": 1001,
3   "type": 0,
4   "from_dev": "AVX-001",
5   "to_dev": "DEV-002",
6   "from_mac": "00:11:22:33:44:55",
7   "to_mac": "66:77:88:99:AA:BB",
8   "from_ip": "192.168.1.10",
9   "to_ip": "192.168.1.20",
10  "from_type": 0,
```

```

11     "to_type": 1
12 }

```

Full parameter configuration example:

```

1 {
2     "task_id": 1001,
3     "type": 2,
4     "role": "1222333",
5     "packet_len": 1400,
6     "speed": 900,
7     "duration": 120,
8     "from_dev": "AVX-001",
9     "to_dev": "DEV-002",
10    "from_mac": "00:11:22:33:44:55",
11    "to_mac": "66:77:88:99:AA:BB",
12    "from_ip": "192.168.1.10",
13    "to_ip": "192.168.1.20",
14    "from_type": 0,
15    "to_type": 1
16 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
task_id	int32	True	KLS task ID
type	int32	True	Diagnostic type. Enum: 0 - Connectivity, 1 - Quick, 2 - Performance
role	string	False	Enter RF02SN
packet_len	int32	False	Packet length, default 1400
speed	int32	False	Bandwidth, unit Mbps, default 900
duration	int32	False	Diagnostic duration, unit seconds, default 120

from_dev	string	True	Source device name
to_dev	string	True	Target device name
from_mac	string	True	Source device MAC address
to_mac	string	True	Target device MAC address
from_ip	string	True	Source device IP address
to_ip	string	True	Target device IP address
from_type	int32	True	Source device type. Enum: 0 - AVX, 1 - Regular device
to_type	int32	True	Target device type. Enum: 0 - AVX, 1 - Regular device

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

11.1.2 Get Current Switch All Device List

Interface Description

Get the current list of all switch devices (dedicated for end-to-end diagnostics).

Request URL

/v1/diagnose/task/alldev/list

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": [  
5     {  
6       "name": "AVX-001",  
7       "mac": "00:11:22:33:44:55",  
8       "serial_number": "SN20240001",  
9       "ip": "192.168.1.10"  
10    }  
11  ]  
12 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Device list
data[].name	string	Device name

data[].mac	string	Device MAC address
data[].serial_number	string	Device serial number
data[].ip	string	Device IP Address

11.1.3 End Diagnostic Task Deletion

Interface Description

Delete specified diagnostic tasks, supports batch deletion.

Request URL

/v1/diagnose/task/del

Request Method

POST

Request Parameter Example

```
1 {  
2   "ids": [2001, 2002, 2003]  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ids	array	True	Diagnostic task ID array

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
----------------	----------------	-------------

code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data, may be empty

11.1.4 Diagnostic Device Result Query

Interface Description

Query diagnostic device results, used for diagnostic progress polling.

Request URL

/v1/diagnose/task/device

Request Method

POST

Request Parameter Example

```
1 {
2   "task_id": 2001,
3   "task_type": 0
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
task_id	int32	True	Diagnostic task ID
task_type	int32	True	Task type. Enum: 0 - Global diagnostic, 1 - End-to-end diagnostic

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "total": 5,
```

```

5  "data": {
6      "result_msg": "诊断完成",
7      "device_info_list": [
8          {
9              "id": 1,
10             "task_id": 2001,
11             "result_type": 0,
12             "from_dev": "AVX-001",
13             "to_dev": "DEV-002",
14             "ping_result": 1,
15             "delay": "5ms",
16             "jitter": "1ms",
17             "loss": "0.1%",
18             "rx": "950Mbps",
19             "tx": "940Mbps",
20             "ip": "192.168.1.20",
21             "devmac": "66:77:88:99:AA:BB",
22             "switcher_sn": "SN20240001",
23             "create_time": "2024-01-01 10:00:00",
24             "update_time": "2024-01-01 10:03:00",
25             "check_device_data": {
26                 "check_delay": true,
27                 "check_loss": true,
28                 "check_tx": true,
29                 "check_device": true,
30                 "check_ping": true
31             },
32             "total_info": "1000/1",
33             "dev_type": "普通设备",
34             "identify": 1
35         }
36     ],
37     "device_count": 5,
38     "done": 100,

```

```
39     "diagnose_time": "2024-01-01 10:03:00",
40     "all_er": {
41         "er_delay": 0,
42         "er_loss": 0,
43         "er_tx": 0,
44         "er_device": 0,
45         "er_ping": 0
46     },
47     "normal": true,
48     "diagnose_info": {
49         "abnormal": 0,
50         "dtime": "180s",
51         "from_dev": "AVX-001",
52         "from_ip": "192.168.1.10",
53         "id": 1,
54         "mac_process": 100,
55         "macping_test": 1,
56         "normal": 5,
57         "ping_process": 100,
58         "ping_test": 1,
59         "recognize": 5,
60         "unrecognized": 0,
61         "step": 2,
62         "task_id": 2001,
63         "to_dev": "DEV-002",
64         "to_ip": "192.168.1.20",
65         "network_diagnose": 1
66     }
67 }
68 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200

		indicates success
message	string	Business prompt message
total	int32	Total number of diagnostic devices
data	object	Diagnostic result core data
data.result_msg	string	Diagnostic result text
data.device_info_list	array	Device diagnostic list
data.device_info_list[].id	int32	Record ID
data.device_info_list[].task_id	int32	Task ID
data.device_info_list[].result_type	int32	Result type
data.device_info_list[].from_dev	string	Source device name
data.device_info_list[].to_dev	string	Target device name
data.device_info_list[].ping_result	int32	Ping status: 0 - Not connected, 1 - Connected, 2 - Not tested (new version only pings end-to-end)
data.device_info_list[].delay	string	Ping latency
data.device_info_list[].jitter	string	Jitter (temporarily unused)
data.device_info_list[].loss	string	Packet loss rate
data.device_info_list[].rx	string	Average rate
data.device_info_list[].tx	string	Receive rate (temporarily unused)
data.device_info_list[].ip	string	Device IP Address
data.device_info_list[].devmac	string	Device MAC address
data.device_info_list[].switcher_sn	string	Switch serial number
data.device_info_list[].create_time	string	Creation time
data.device_info_list[].update_time	string	Update time
data.device_info_list[].check_device_data	object	Diagnostic verification result
data.device_info_list[].check_device_data.check_delay	bool	Whether latency is normal
data.device_info_list[].check_device_data.check_loss	bool	Whether packet loss is normal
data.device_info_list[].check_device_data.check_tx	bool	Whether sending is normal
data.device_info_list[].check_device_data.check_device	bool	Whether device is normal
data.device_info_list[].check_device_data.check_ping	bool	Whether Ping is normal
data.device_info_list[].total_info	string	Total packets / lost packets information (deprecated)
data.device_info_list[].dev_type	string	Device type
data.device_info_list[].identify	int32	Whether identified: 0 -

		Unidentified, 1 - Identified
data.device_count	int32	Total number of diagnostic devices (currently unused)
data.done	int32	Diagnostic progress bar
data.diagnose_time	string	Diagnostic time
data.all_er	object	Abnormal statistics (currently unused)
data.all_er.er_delay	int32	Number of latency anomalies
data.all_er.er_loss	int32	Number of packet loss anomalies
data.all_er.er_tx	int32	Number of rate anomalies
data.all_er.er_device	int32	Number of abnormal devices
data.all_er.er_ping	int32	Number of Ping anomalies
data.normal	bool	Whether current diagnostic result is normal (currently unused)
data.process	object	Diagnostic process (deprecated)
data.diagnose_info	object	New version global diagnostic information
data.diagnose_info.abnormal	int32	Number of abnormal devices (global diagnostic)
data.diagnose_info.dtime	string	Diagnostic time
data.diagnose_info.from_dev	string	Source device name
data.diagnose_info.from_ip	string	Source device IP address
data.diagnose_info.id	int32	Diagnostic information ID
data.diagnose_info.mac_process	int32	Network connectivity test progress
data.diagnose_info.macping_test	int32	Network connectivity test status: 0 - Testing, 1 - Success, 2 - Failed, 3 - Not tested
data.diagnose_info.normal	int32	Number of normal devices (global diagnostic)
data.diagnose_info.ping_process	int32	IP connectivity test progress
data.diagnose_info.ping_test	int32	IP connectivity test status: 0

		- Testing, 1 - Success, 2 - Failed, 3 - Not tested
data.diagnose_info.recognize	int32	Number of identified devices (global diagnostic)
data.diagnose_info.unrecognized	int32	Number of Unidentified Devices (Global Diagnostic)
data.diagnose_info.step	int32	Current progress (both end-to-end and global have two steps)
data.diagnose_info.task_id	int32	Event ID
data.diagnose_info.to_dev	string	Target device name
data.diagnose_info.to_ip	string	Target device IP address
data.diagnose_info.network_diagnose	int32	Whether network diagnostic is complete: 1 - Complete

11.1.5 Start End-to-End Diagnostic

Interface Description

Start generic end-to-end diagnostic task.

Request URL

/v1/diagnose/task/p2pdev/add

Request Method

POST

Request Parameter Example

Example with only required parameters (using defaults):

```
1 {
2   "from_dev": "AVX-001",
3   "from_ip": "192.168.1.10",
4   "from_mac": "00:11:22:33:44:55",
5   "to_dev": "DEV-002",
6   "to_ip": "192.168.1.20",
```

```

7   "to_mac": "66:77:88:99:AA:BB",
8   "type": 0
9 }

```

Full parameter configuration example:

```

1 {
2   "from_dev": "AVX-001",
3   "from_ip": "192.168.1.10",
4   "from_mac": "00:11:22:33:44:55",
5   "to_dev": "DEV-002",
6   "to_ip": "192.168.1.20",
7   "to_mac": "66:77:88:99:AA:BB",
8   "type": 2,
9   "role": "sendrecv",
10  "speed": 1000,
11  "duration": 180,
12  "packet_len": 1500
13 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
from_dev	string	True	Device 1 name (source device)
from_ip	string	True	Device 1 IP address
from_mac	string	True	Device 1 MAC address
to_dev	string	True	Device 2 Name (Target Device)
to_ip	string	True	Device 2 IP address
to_mac	string	True	Device 2 MAC address
type	int32	True	Diagnostic type. Enum: 0 - Connectivity, 1 - Quick, 2 - Performance
role	string	False	Diagnostic send/receive mode. Enum:

			recv/send/sendrecv
speed	int32	False	Bandwidth, unit Mbps
duration	int32	False	Diagnostic duration, unit seconds
packet_len	int32	False	Packet length

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "task_id": 3001
6   }
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Return data
data.task_id	int32	Diagnostic task ID

11.1.6 Query Diagnostic Tasks (by Type)

Interface Description

Query historical diagnostic task list by task type.

Request URL

/v1/diagnose/task/query/type

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "id": 3001,
7       "name": "端到端诊断任务",
8       "type": 1,
9       "from_dev": "AVX-001",
10      "to_dev": "DEV-002",
11      "seq": 1,
12      "speed": 1000,
13      "duration": 180,
14      "pkg_len": 1500,
15      "vlan_id": 100,
16      "status": 1,
17      "switcher_sn": "SN20240001",
18      "create_time": "2024-01-01 10:00:00",
19      "update_time": "2024-01-01 10:03:00"
20    }
21  ]
22 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200

		indicates success
message	string	Business prompt message
data	array	Diagnostic task list
data[].id	int32	Task ID
data[].name	string	Task name
data[].type	int32	Task type. Enum: 0 - Global diagnostic, 1 - End-to-end diagnostic
data[].from_dev	string	Source device name
data[].to_dev	string	Target device name
data[].seq	int32	Serial number
data[].speed	int32	Speed test rate, unit Mbps
data[].duration	int32	Diagnostic duration, unit seconds
data[].pkg_len	int32	Packet length
data[].vlan_id	int32	VLAN ID
data[].status	int32	Task status. Enum: 0 - In progress, 1 - Completed, 2 - Task partially failed, 3 - Task failed
data[].switcher_sn	string	Switch serial number
data[].create_time	string	Creation time
data[].update_time	string	Update time

12 Switch Management

12.1 System Information

12.1.1 Query Switch System Information

Interface Description

Get basic switch information, device information, model, alias, serial number, KiloLink connection status, etc.

Request URL

/v1/switch/sys/info

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "basic_info": {
6       "loc_mac": "18:C3:F4:2F:FB:F3",
7       "uptime": "1d 07:10:14",
8       "soft_version": "0.03.0268",
9       "hw_version": "V0.0-V1-0.0-0.0"
10    },
11    "dev_info": {
12      "cpu_usage": "10%",
13      "mem_usage": "12%",
14      "temperature": "73",
15      "fan": "87%"
16    },
17    "model": "AVX24-4",
18    "alias": "AVX24-4",
19    "sn": "18c3f425640e",
20    "kls_link_status": "disconnected",
21    "hostname": "AVX24-4-lion",
22    "used_port": 6,
23    "total_port": 28,
```

```
24     "mode": "layer3"
25   }
26 }
```

Response Parameter Description

Parameter name	Parameter type	Description
basic_info	Object	Runtime information (MAC, uptime, version)
dev_info	Object	CPU/Memory/Temperature/Fan
model	String	Model
alias	String	Alias
sn	String	Serial number
kls_link_status	String	KiloLink status

12.1.2 Set Device Alias

Interface Description

Modify the device display alias in the topology by device MAC.

Request URL

/v1/switch/dev/alias/set

Request Method

POST

Request Parameter Example

```
1 {
2   "mac": "AA:BB:CC:DD:EE:FF",
3   "alias": "Camera-01"
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
mac	String	True	Device MAC

alias	String	True	New alias
-------	--------	------	-----------

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2 Port Management

12.2.1 Query All Port Status

Interface Description

Query all switch port status (VLAN, Link, Aggregation, Traffic, EEE, PoE, etc.).

Request URL

/v1/switch/port/dump

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
```

```

5      {
6          "port": 11,
7          "alias": "port11",
8          "vlan": {
9              "type": "access",
10             "pvid": 3,
11             "vid": "3",
12             "alias": "vlan3",
13             "color": "#4298ED",
14             "config_file": ""
15         },
16         "link": {
17             "enable": true,
18             "link": true,
19             "media": "copper",
20             "speed_duplex": "1G",
21             "autonego": false,
22             "flow_control": true
23         },
24         "aggregation": {
25             "aggre_id": 0,
26             "type": "-"
27         },
28         "rate": {
29             "rx": {
30                 "packets": "5",
31                 "bytes": "1398",
32                 "error": "0",
33                 "loss": "0"
34             },
35             "tx": {
36                 "packets": "3",
37                 "bytes": "620",
38                 "error": "0",

```

```

39         "loss": "0"
40     }
41 },
42     "eee": {
43         "actiphy": false,
44         "perfect_reach": false,
45         "enable": false
46     },
47     "poe": {
48         "enable": true,
49         "type": "PoE",
50         "power": "14.85",
51         "voltage": "54.02",
52         "current": "275"
53     },
54     "bind_ip": {
55         "enable": false,
56         "ip": ""
57     },
58     "igmp": {
59         "port": 11,
60         "router": false,
61         "uplink": true
62     },
63     "ptp_clock_domain": 0,
64     "device_ip": "192.168.3.38",
65     "model": "",
66     "image": "",
67     "company": ""
68 },
69 ...
70 ]
71 }

```

Response Parameter Description

data is a port information array, each element corresponds to a port's configuration and status. The following only describes fields within data (excluding top-level code and message).

Description conventions

1. Object fields in data (such as vlan, link, rate, etc.): first list their sub-fields in a separate table near the beginning of this section; in the final "data array element field description" table, the corresponding row only writes "see xxx" without repeating the sub-field expansion.
2. Non-Object fields (such as port, alias, ptp_clock_domain, etc.): only described in the "data array element field description" table.
3. rate: rx and tx under data[].rate are both objects with identical structure, therefore two tables are used – "rate object field description" and "rate.rx / rate.tx object field description".

vlan object field description (data[].vlan)

Parameter name	Parameter type	Description
type	String	VLAN type: access or trunk
pvid	Number	PVID / Native VLAN ID, value range 1-4096
vid	String	For access mode, it can be a placeholder or a PVID-related string; for trunk mode, it is a Tagged VLAN list expression of allowed VLANs (e.g., 1,3,10-4000)
alias	String	Alias of this VLAN
color	String	Color of this VLAN (display string, e.g., #1D2024)
config_file	String	Associated configuration file identifier; empty string if unavailable

link object field description (data[].link)

Parameter name	Parameter type	Description
enable	Boolean	Whether the port is enabled (logical on/off)
link	Boolean	Whether the link is Up (true means online)

media	String	Media type: copper / fiber
speed_duplex	String	Current or negotiated speed duplex, e.g., 1G, 10G, auto, etc.
autonego	Boolean	Whether auto-negotiation is enabled
flow_control	Boolean	Whether flow control is enabled

aggregation object field description (data[].aggregation)

Parameter name	Parameter type	Description
aggre_id	Number	Associated aggregation group ID; 0 means non-aggregated port
type	String	Aggregation mode or display placeholder (e.g., static, lacp_active, -, etc., subject to actual return value)

rate object field description (data[].rate)

Parameter name	Parameter type	Description
rx	Object	Receive direction traffic statistics, sub-fields see "rate.rx / rate.tx object field description"
tx	Object	Send direction traffic statistics, sub-fields same as rx

rate.rx / rate.tx object field description (data[].rate.rx, data[].rate.tx)

Parameter name	Parameter type	Description
packets	String	Packet count (decimal number string in the example)
bytes	String	Byte count (string in example)
error	String	Error packets
loss	String	Packet loss count

eee object field description (data[].eee)

Parameter name	Parameter type	Description
actiphy	Boolean	Actiphy energy-saving mode switch
perfect_reach	Boolean	PerfectReach energy-saving mode switch
enable	Boolean	EEE master switch (consistent with implementation/display)

poe object field description (data[].poe)

Parameter name	Parameter type	Description
enable	Boolean	Whether PoE power supply is enabled
type	String	PoE level, e.g., PoE, PoE+, PoE++
power	String	Power; can be empty string when not counted
voltage	String	Voltage; can be empty string when not counted
current	String	Current; can be "0" or empty string

bind_ip object field description (data[].bind_ip)

Parameter name	Parameter type	Description
enable	Boolean	Whether to enable port-bound static IP
ip	String	Bound IPv4 address; can be empty string when not enabled

igmp object field description (data[].igmp)

Parameter name	Parameter type	Description
port	Number	Port number (consistent with entry port)
router	Boolean	IGMP router related role/capability identifier
uplink	Boolean	Topology role identifier such as whether it is an uplink port

data array element field description

Parameter name	Parameter type	Description
port	Number	Port ID: 1-24 are electrical ports, 33-36 are optical ports
alias	String	Port display alias
vlan	Object	See "vlan object field description"
link	Object	See "link object field description"
aggregation	Object	See "aggregation object field description"
rate	Object	See "rate object field description" and "rate.rx / rate.tx object field description"
eee	Object	See "eee object field description"
poe	Object	See "poe object field description"
bind_ip	Object	See "bind_ip object field description"
igmp	Object	See "igmp object field description"
ptp_clock_domain	Number	PTP clock domain; can be 0 when not in use
device_ip	String	Peer or associated device IP; empty string if unavailable
model	String	Associated device model; empty string if unavailable
image	String	Associated device icon/image identifier; empty string if unavailable
company	String	Associated device vendor and other information; empty string if unavailable
connect_mode	String	(If returned) Connection method description: regular port or aggregation group

		name, etc.; consistent with implementation extensions
--	--	---

12.2.2 Port Control

Interface Description

Execute down / up / restart on the port; does not affect the PoE power supply policy itself.

Request URL

/v1/switch/port/control

Request Method

POST

Request Parameter Example

```
1 {
2   "switch_id": 0,
3   "port_id": 1,
4   "command": "restart"
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
switch_id	Number	False	Can fill 0 in standalone scenario
port_id	Number	True	Port ID
command	String	True	down / up / restart

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2.3 Set Port

Interface Description

Set the alias, static IP, VLAN, Link, EEE, PoE, etc. for the specified port.

Request URL

/v1/switch/port/set

Request Method

POST

Request Parameter Example

```
1 {  
2   "msg": {  
3     "alias": "port1",  
4     "eee": {  
5       "actiphy": false,  
6       "perfect_reach": false,  
7       "enable": true  
8     },  
9     "poe": {  
10      "enable": true,  
11      "type": "PoE",  
12      "power": "",  
13      "voltage": "",  
14      "current": "0"  
15    },  
16    "link": {  
17      "enable": true,  
18      "link": true,  
19      "media": "copper",  
20      "speed_duplex": "1G",  
21      "autonego": false,
```

```

22         "flow_control": true
23     },
24     "vlan": {
25         "type": "access",
26         "pvid": 1,
27         "vid": "1",
28         "alias": "vlan1",
29         "color": "#1D2024",
30         "config_file": ""
31     },
32     "ip": {},
33     "igmp": {
34         "port": 1,
35         "router": false,
36         "uplink": true
37     },
38     "port": 1
39 }
40 }

```

Request body is {"msg": { ... }}, the following parameters are all within the msg object (consistent with the example above).

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
port	Number	True	Port number to be set
alias	String	False	Port display alias
vlan	Object	False	Please refer to "vlan object field description" in 12.2.1 Query All Port Information (switchPortDumpApi).
link	Object	False	Please refer to "link object field description" in 12.2.1.

eee	Object	False	Please refer to "eee object field description" in 12.2.1.
poe	Object	False	Please refer to "poe object field description" in 12.2.1.
ip	Object	False	Static IP / Binding; sub-fields include enable, addr, op (binding group). When comparing with "bind_ip object field description" (12.2.1) on the query side, note: the query uses bind_ip, while this interface request field name is ip, and includes an additional op field.
igmp	Object	False	Please refer to "igmp object field description" in 12.2.1.

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2.4 Batch Set Ports

Interface Description

Batch set enable, speed, EEE, PoE, etc. for multiple ports.

Request URL

/v1/switch/port/set/batch

Request Method

POST

Request Parameter Example

```
1 {  
2   "msg": {  
3     "eee_enable": true,  
4     "poe_enable": true,  
5     "port_enable": true,  
6     "speed_duplex": "auto",  
7     "uplink": false,  
8     "vlan": {  
9       "alias": "",  
10      "color": "",  
11      "config_name": "",  
12      "pvid": 1,  
13      "type": "access",  
14      "vid": "",  
15      "interface": {}  
16    },  
17    "ports": [  
18      1,  
19      3  
20    ]  
21  }  
22 }
```

Request Parameter Description

Request body is {"msg": { ... }}, the following parameters are all within msg.

msg.vlan object field description

Parameter name	Parameter type	Whether required	Description
(Structure and meaning)	-	-	For type, pvid, vid, alias, color, etc., please refer to "vlan object field description" in 12.2.1 Query All Port Information (switchPortDumpApi); config_name in batch examples may differ from config_file naming on the query side, subject to the gateway.
config_name	String	False	Configuration name (example field)
interface	Object	False	VLAN Layer 3 interface; for sub-fields, please refer to "interface object field description" in 12.4.1 Get All VLANs (switchVlanDumpApi).

msg root-level field description

Parameter name	Parameter type	Whether required	Description
ports	Number[]	True	List of port IDs to be batch set
port_enable	Boolean	False	Port enable
speed_duplex	String	False	auto, 10G, 1G, 100MF, 10MF, etc.
eee_enable	Boolean	False	EEE switch
poe_enable	Boolean	False	PoE switch
uplink	Boolean	False	Whether it is an uplink port, etc. (example

			field, subject to gateway implementation)
vlan	Object	False	See "msg.vlan object field description" above

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2.5 Set Port Alias

Interface Description

Only modify port alias.

Request URL

/v1/switch/port/alias/set

Request Method

POST

Request Parameter Example

```
1 {
2   "port": 1,
3   "alias": "port-1"
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
----------------	----------------	------------------	-------------

port	Number	True	Port number
alias	String	True	Alias

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.2.6 Query Port Traffic

Interface Description

Query send/receive traffic statistics for the specified port.

Request URL

/v1/switch/port/traffic

Request Method

POST

Request Parameter Example

```
1 {
2   "port": 1
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
port	Number	True	Port number

Response Example

```
1 {
2   "code": 200,
```

```

3  "message": "ok",
4  "data": {
5      "port": "1",
6      "traffic": [
7          {
8              "rx": {
9                  "packets": "8438",
10                 "bytes": "12236088",
11                 "error": "0",
12                 "loss": "0"
13             },
14             "tx": {
15                 "packets": "29",
16                 "bytes": "10757",
17                 "error": "0",
18                 "loss": "0"
19             }
20         },
21         ...
22     ]
23 }
24 }

```

Response Parameter Description

data.traffic[] element field description (each item in the data.traffic array)

Parameter name	Parameter type	Description
rx	Object	Receive statistics; for sub-fields, please refer to "rate.rx / rate.tx object field description" in 12.2.1 (packets, bytes, error, loss, etc.).
tx	Object	Send statistics; sub-fields same as above.

data object field description

Parameter name	Parameter type	Description
port	String	Queried port number (string in example)
traffic	Array	Traffic statistics sequence (e.g., multi-period/multi-group); elements see "data.traffic[]" element field description"
Parameter name	Parameter type	Description
code	Number	Business status code
message	String	Prompt message
data	Object	See "data object field description" above

12.2.7 Query Port Traffic Top

Interface Description

Query traffic statistics list for all ports.

Request URL

/v1/switch/port/traffic/query/top

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
```

```
4  "data": {
5      "sum_rx_rate": 93.395,
6      "sum_tx_rate": 0.24,
7      "desc_stat": [
8          {
9              "port": "1",
10             "rx_rate": 93.318,
11             "tx_rate": 0.036,
12             "top_rate": 93.318,
13             "top_usage": 0.093
14         },
15         {
16             "port": "23",
17             "rx_rate": 0.032,
18             "tx_rate": 0.09,
19             "top_rate": 0.09,
20             "top_usage": 0
21         },
22         {
23             "port": "4",
24             "rx_rate": 0.019,
25             "tx_rate": 0.06,
26             "top_rate": 0.06,
27             "top_usage": 0
28         },
29         {
30             "port": "33",
31             "rx_rate": 0.003,
32             "tx_rate": 0.045,
33             "top_rate": 0.045,
34             "top_usage": 0
35         },
36         {
37             "port": "11",
```

```

38         "rx_rate": 0.016,
39         "tx_rate": 0.005,
40         "top_rate": 0.016,
41         "top_usage": 0
42     },
43     {
44         "port": "13",
45         "rx_rate": 0.007,
46         "tx_rate": 0.004,
47         "top_rate": 0.007,
48         "top_usage": 0
49     }
50 ]
51 }
52 }

```

Response Parameter Description

data.desc_stat[] element field description

Parameter name	Parameter type	Description
port	String	Port number
rx_rate	Number	Receive rate (example unit consistent with gateway, e.g., Mbps)
tx_rate	Number	Send rate
top_rate	Number	Peak/composite rate used for sorting
top_usage	Number	Occupancy or utilization related metrics

data object field description

Parameter name	Parameter type	Description
sum_rx_rate	Number	Receive rate summary for each port
sum_tx_rate	Number	Send rate summary for each port

desc_stat	Array	Top port list, elements see "data.desc_stat[] element field description"
Parameter name	Parameter type	Description
code	Number	Business status code
message	String	Prompt message
data	Object	See "data object field description" above

12.2.8 Set Watched Port

Interface Description

Set watched port list

Request URL

/v1/switch/port/display/set

Request Method

POST

Request Parameter Example

```
1 {
2   "type": "select",
3   "ports": [
4     1,
5     11,
6     13,
7     23,
8     33,
9     4
10  ]
11 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
type	String	True	Focus type; example is select, subject to gateway conventions
ports	Number[]	True	List of port IDs to set as watched

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": null  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	Number	Business status code; 200 indicates success
message	String	Prompt message
data	Null	Can be null on success

12.2.9 Remove Watched Port

Interface Description

Remove ports from the watch list

Request URL

/v1/switch/port/display/del

Request Method

POST

Request Parameter Example

```
1 {
2   "type": "select",
3   "ports": [
4     4
5   ]
6 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
type	String	True	Focus type; example is select, must be consistent with the setting
ports	Number[]	True	List of port IDs to remove from the watch list

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	Number	Business status code; 200 indicates success
message	String	Prompt message
data	Null	Can be null on success

12.2.10 Query Watched Port Information

Interface Description

Query watched port status

Request URL

/v1/switch/port/display/query

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "cascade_ports": [],
3   "select_ports": [
4     {
5       "port": 11,
6       "alias": "port11",
7       "vlan": {
8         "type": "access",
9         "pvid": 3,
10        "vid": "3",
11        "alias": "vlan3",
12        "color": "#4298ED",
13        "config_file": ""
14      },
15      "link": {
```

```

16         "enable": true,
17         "link": true,
18         "media": "copper",
19         "speed_duplex": "1G",
20         "autonego": false,
21         "flow_control": true
22     },
23     "aggregation": {
24         "aggre_id": 0,
25         "type": "-"
26     },
27     "rate": {
28         "rx": {
29             "packets": "6",
30             "bytes": "1789",
31             "error": "0",
32             "loss": "0"
33         },
34         "tx": {
35             "packets": "4",
36             "bytes": "800",
37             "error": "0",
38             "loss": "0"
39         }
40     },
41     "eee": {
42         "actiphy": false,
43         "perfect_reach": false,
44         "enable": false
45     },
46     "poe": {
47         "enable": true,
48         "type": "PoE",
49         "power": "14.48",

```

```

50     "voltage": "54.02",
51     "current": "268"
52 },
53     "bind_ip": {
54         "enable": false,
55         "ip": ""
56     },
57     "igmp": {
58         "port": 11,
59         "router": false,
60         "uplink": true
61     },
62     "ptp_clock_domain": 0,
63     "device_ip": "192.168.3.38",
64     "model": "N60",
65     "image": "./static/img/N60.png",
66     "company": "Kiloview"
67 },
68 ...
69 ]
70 }

```

Response Parameter Description

The example shows cascade_ports / select_ports business payload; if the gateway returns a standard envelope { "code", "message", "data" }, the following two fields are located within data, and the top-level code and message can be supplemented at the end of the table.

select_ports[], cascade_ports[] element field description (both array element structures are identical, consistent with a single port entry in data[] from 12.2.1 Query All Port Information (switchPortDumpApi))

Parameter name	Parameter type	Description
port	Number	Port ID
alias	String	Port alias
vlan	Object	Please refer to "vlan object"

		field description" in 12.2.1 Query All Port Information (switchPortDumpApi).
link	Object	Please refer to "link object field description" in 12.2.1.
aggregation	Object	Please refer to "aggregation object field description" in 12.2.1.
rate	Object	Please refer to "rate object field description" and "rate.rx / rate.tx object field description" in 12.2.1.
eee	Object	Please refer to "eee object field description" in 12.2.1.
poe	Object	Please refer to "poe object field description" in 12.2.1.
bind_ip	Object	Please refer to "bind_ip object field description" in 12.2.1.
igmp	Object	Please refer to "igmp object field description" in 12.2.1.
ptp_clock_domain	Number	PTP clock domain, etc.
device_ip	String	Peer/Device IP
model	String	Peer device model
image	String	Icon or resource path
company	String	Vendor and other display fields
connect_mode	String	(If returned) Connection method description

data (or response root-level) object field description

Parameter name	Parameter type	Description
cascade_ports	Array	Port list for cascade/association display; elements see "select_ports[], cascade_ports[] element field description" above
select_ports	Array	User-selected watch port list; element structure same as above

Parameter name	Parameter type	Description
code	Number	(If standard envelope is returned) Business status code
message	String	(If standard envelope is returned) Prompt Message
data	Object	(If standard envelope is returned) contains cascade_ports, select_ports

12.2.11 Remove Device

Interface Description

Remove the device connected to a switch port (enters removal process with configurable effective duration).

Request URL

/v1/switch/device/remove

Request Method

POST

Request Parameter Example

```
1 {  
2   "switch_sn": "",  
3   "port": 1,  
4   "duration": 10  
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
switch_sn	String	False	Serial number; for standalone, can pass empty string to indicate local device
port	Number	True	Port ID

duration	Number	False	Effective duration (minutes); 0 means use default 10 minutes
----------	--------	-------	--

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.3 Port Aggregation

12.3.1 Create Link Aggregation Group

Interface Description

Create aggregation group (only one aggregation group can be created per request)

Additional notes:

- aggre_id must be unique within the device; it is recommended to use sequential numbering (e.g., 1, 2, 3...); if not provided or set to 0, it will be auto-generated by the backend
- type values: static, lacp_active (LACP active)
- members is the list of port IDs to be added to this aggregation group

Request URL

/v1/switch/aggr/add

Request Method

POST

Request Parameter Example

```
1 {
2   "msg": [
3     {
```

```

4      "aggre_id": 1,
5      "alias": "上联聚合组",
6      "type": "lacp_active",
7      "members": [1, 2]
8    }
9  ]
10 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
msg	array	True	Aggregation group configuration list (only one aggregation group supported per request)
msg.aggre_id	int32	True	Aggregation group ID (passing 0 will auto-calculate the ID)
msg.alias	string	False	Aggregation group alias
msg.type	string	True	Aggregation type: static, lacp_active (LACP active)
msg.members	array	True	List of port IDs to join the aggregation group

Response Example

```

1 {
2   "code": 200,
3   "message": "ok"
4 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200

		indicates success
message	string	Business prompt message

12.3.2 Delete Link Aggregation Group

Interface Description

Delete specified aggregation groups (supports batch deletion)

Request URL

/v1/switch/aggr/del

Request Method

POST

Request Parameter Example

```
1 {
2     "aggre_id": [1, 2]
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
aggre_id	array	True	List of aggregation group IDs to delete

Response Example

```
1 {
2     "code": 200,
3     "message": "ok"
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success

message	string	Business prompt message
---------	--------	-------------------------

12.3.3 Query All Link Aggregation Groups

Interface Description

Query all aggregation group information

Additional notes:

- configured: list of ports already added to this aggregation group at the configuration level
- aggregated: list of ports actually aggregated successfully in running state (may be fewer than configured)

Request URL

/v1/switch/aggr/dump

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "aggre_id": 1,
7       "alias": "聚合组 1",
```

```

8         "type": "static",
9         "configured": [1, 2, 3],
10        "aggregated": [1, 2, 3]
11    }
12 ]
13 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Aggregation group list
data.aggre_id	int32	Aggregation group ID
data.alias	string	Aggregation group alias
data.type	string	Aggregation type: static/lacp_active
data.configured	array	List of Port IDs Configured to Join the Aggregation Group
data.aggregated	array	List of port IDs actually and successfully joined to the aggregation group

12.3.4 Query Specified or All Link Aggregation Groups

Interface Description

Query specified aggregation group information by aggregation group ID (supports batch query with multiple IDs); returns an empty list when aggre_id is not provided or is an empty array. To query all aggregation groups, use the /v1/switch/aggr/dump interface.

Request URL

/v1/switch/aggr/get

Request Method

POST

Request Parameter Example

```
1 {
2   "aggre_id": [1]
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
aggre_id	array	False	List of aggregation group IDs to query; returns empty list when empty, use /v1/switch/aggr/dump to query all

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "aggre_id": 1,
7       "alias": "string",
8       "type": "string",
9       "configured": [
10        1
11      ],
12      "aggregated": [
13        1
14      ]
15    }
16  ]
17 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Business data body
data.aggre_id	int32	Numeric parameter
data.alias	string	Alias
data.type	string	Aggregation type: static/lacp_active
data.configured	array	List of Port IDs Configured to Join the Aggregation Group
data.aggregated	array	List of port IDs actually and successfully joined to the aggregation group

12.3.5 Append Ports to Aggregation Group

Interface Description

Append port members to the specified aggregation group (does not change aggregation group type)

Request URL

/v1/switch/aggr/port/add

Request Method

POST

Request Parameter Example

```
1 {  
2   "aggre_id": 1,  
3   "members": [3, 4]  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
aggre_id	int32	True	Target aggregation group ID
members	array	True	List of port IDs to append to the aggregation group

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {}  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

12.3.6 Modify Link Aggregation Group Configuration

Interface Description

Modify aggregation group configuration (supports modifying alias, type, member ports; only one aggregation group can be modified per request)

Request URL

/v1/switch/aggr/set

Request Method

POST

Request Parameter Example

```
1 {
2   "msg": [
3     {
4       "aggre_id": 1,
5       "alias": "上联聚合组 A",
6       "type": "lacp_passive",
7       "members": [1, 2, 3]
8     }
9   ]
10 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
msg	array	True	Aggregation group configuration list (only one aggregation group supported per request)
msg.aggre_id	int32	True	Aggregation group ID to be modified
msg.alias	string	False	Aggregation group alias
msg.type	string	False	Aggregation type: static/lacp_active
msg.members	array	False	Updated member port ID list

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {}
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

12.4 VLAN Management

12.4.1 Query All VLAN Information

Interface Description

Get all current VLAN attributes and port, interface, and other information.

Request URL

/v1/switch/vlan/dump

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": [  
5     {  
6       "id": 1,  
7       "alias": "vlan1",
```

```

8      "color": "#1D2024",
9      "configs": [],
10     "config_names": [],
11     "ports": [
12         1,
13         2,
14         3,
15         4,
16         5,
17         6,
18         33,
19         34,
20         35,
21         36
22     ],
23     "interface": {
24         "enable": true,
25         "ip": "172.30.46.107",
26         "netmask": "255.255.255.0",
27         "type": "dhcp",
28         "referenced": true,
29         "is_default_route": false
30     },
31     "dhcp_enable": false,
32     "vlan_if": {
33         "vlan_id": 1,
34         "enable": true,
35         "ip": "172.30.46.254",
36         "netmask": "255.255.255.0"
37     },
38     "dhcp_conflict": false
39 },
40 {
41     "id": 3,

```

```
42     "alias": "vlan3",
43     "color": "#4298ED",
44     "configs": [],
45     "config_names": [],
46     "ports": [
47         7,
48         8,
49         9,
50         10,
51         11,
52         12,
53         13,
54         14,
55         15,
56         16,
57         17,
58         18,
59         19,
60         20,
61         21,
62         22,
63         23,
64         24
65     ],
66     "interface": {
67         "enable": true,
68         "ip": "192.168.3.2",
69         "netmask": "255.255.255.0",
70         "type": "dhcp",
71         "referenced": true,
72         "is_default_route": false
73     },
74     "dhcp_enable": true,
75     "vlan_if": {
```

```

76         "vlan_id": 3,
77         "enable": true,
78         "ip": "192.168.3.1",
79         "netmask": "255.255.255.0"
80     },
81     "dhcp_conflict": false
82 }
83 ]
84 }

```

Response Parameter Description

Object type fields are first listed in sub-tables, then referenced in the "data array element" main table.

"interface object field description" (data[].interface, VLAN Layer 3/management interface display)

Parameter name	Parameter type	Description
enable	Boolean	Whether this interface configuration is enabled
ip	String	IPv4 address; can be empty string when not configured
netmask	String	Subnet mask
type	String	Address type, e.g., dhcp, manual
referenced	Boolean	Whether referenced/associated (business display field)
is_default_route	Boolean	Whether it is a default route related interface

"vlan_if object field description" (data[].vlan_if, VLANIF deployment/runtime parameters)

Parameter name	Parameter type	Description
vlan_id	Number	Corresponding VLAN ID
enable	Boolean	Whether VLANIF is enabled
ip	String	VLANIF IP
netmask	String	VLANIF mask

"data array element field description" (each VLAN entry in data[])

Parameter name	Parameter type	Description
id	Number	VLAN ID, value range 1-4096
alias	String	VLAN alias
color	String	Display color string
configs	Number[]	Associated configuration file ID list
config_names	String[]	Associated configuration file name list (corresponding to configs, subject to the gateway)
ports	Number[]	List of port IDs joined to this VLAN
interface	Object	See "interface object field description"
dhcp_enable	Boolean	Whether DHCP is enabled (example field)
vlan_if	Object	See "vlan_if object field description"
dhcp_conflict	Boolean	DHCP conflict and other business flags (example field)
Parameter name	Parameter type	Description
code	Number	Business status code; 200 indicates success
message	String	Prompt message
data	Array	VLAN list; elements see "data array element field description" (consistent with switchVlanAttr in switch_api.go and gateway extended fields)

12.4.2 Add VLAN

Interface Description

Add one or more VLANs; if a VLAN already exists, it will be overwritten.

Request URL

/v1/switch/vlan/add

Request Method

POST

Request Parameter Example

```
1 {
2   "data": [
3     {
4       "id": 10,
5       "configs": [
6         1
7       ],
8       "color": "#38bb70",
9       "alias": "vlan10",
10
11      "ports": [],
12
13      "vlan_if": {
14        "vlan_id": 1,
15        "ip": "",
16        "enable": false,
17        "netmask": "255.255.255.0"
18      },
19      "interface": {
20        "enable": false,
21        "ip": "",
22        "netmask": "255.255.255.0",
23        "type": "dhcp"
24      }
25    }
26  ]
27 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
data	Array	True	VLAN object array; each element's fields are consistent with "data array element field description" in 12.2.1 Query All VLAN Information; among them, interface and vlan_if sub-fields refer to "interface object field description" and "vlan_if object field description" in the same section respectively. Fields not present in the request are handled by the gateway with default values.

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": ""  
5 }
```

On error, code may be 70001 etc., message contains bilingual description.

12.4.3 Set VLAN

Interface Description

Edit the configuration of the specified VLAN.

Request URL

/v1/switch/vlan/set

Request Method

POST

Request Parameter Example

```
1 {
2   "data": {
3     "id": 10,
4     "configs": [
5       1
6     ],
7     "color": "#FFD700",
8     "alias": "vlan10",
9     "ports": [],
10    "vlan_if": {
11      "vlan_id": 10,
12      "enable": false,
13      "ip": "",
14      "netmask": ""
15    }
16  }
17 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
data	Object	True	Single VLAN; fields are consistent with "data array element field description" in 12.2.1 Query All VLAN Information. For interface and vlan_if,

			see "interface object field description" and "vlan_if object field description" in 12.2.1 Query All VLAN Information.
--	--	--	--

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.4.4 Query VLAN Information

Interface Description

Query single VLAN information by VLAN ID.

Request URL

/v1/switch/vlan/query

Request Method

POST

Request Parameter Example

```
1 {
2   "vlan_id": 100
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_id	Number	True	VLAN ID

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "id": 10,
6     "alias": "vlan10",
7     "color": "#38bb70",
8     "configs": [
9       1
10    ],
11    "ports": [33],
12    "vlan_if": {
13      "vlan_id": 10,
14      "enable": false,
15      "ip": "",
16      "netmask": ""
17    }
18  }
19 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	Number	Business status code
message	String	Prompt message
data	Object	Single VLAN; fields are consistent with "data array element field description" in 12.2.1 Query All VLAN Information (query response may omit some extended fields). For interface and vlan_if, see the corresponding sub-tables in 12.2.1 Query All VLAN Information.

12.4.5 Delete VLAN

Interface Description

Delete one or more VLANs.

Request URL

/v1/switch/vlan/del

Request Method

POST

Request Parameter Example

```
1 {
2     "vlan": [
3         10
4     ]
5 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan	Number[]	True	List of VLAN IDs to be deleted

Response Example

```
1 {
2     "code":200,
3     "message":"ok",
4     "data":null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	Number	Business status code

message	String	Prompt message
data	Null	Business data can be null on success

12.4.6 Set VLAN Port

Interface Description

Batch add specified ports to the corresponding VLAN.

Request URL

/v1/switch/vlan/port/set

Request Method

POST

Request Parameter Example

```
1 {
2   "msg": [
3     {
4       "vlan_id": 10,
5       "ports": [
6         36,
7         35
8       ]
9     }
10  ]
11 }
```

Request Parameter Description

Request body is msg array; each item represents adding several ports to the specified VLAN.

msg[] element field description

Parameter name	Parameter type	Whether required	Description
vlan_id	Number	True	VLAN ID
ports	Number[]	True	List of port IDs to be

			added to this VLAN
--	--	--	--------------------

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": ""
5 }
```

12.5 DHCP

12.5.1 Query DHCP Client List

Interface Description

Query DHCP client list (devices that obtained IP addresses)

Request URL

/v1/switch/dhcps/clients/dump

Request Method

POST

Request Parameter Example

```
1 {
2   "vlan_id": []
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_id	array	False	VLAN ID list; if not provided, queries clients for all VLANs

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "vlan_id": 1,
7       "ip": "string",
8       "netmask": "string",
9       "type": "string",
10      "host_name": "string",
11      "mac": "string",
12      "expired_time": {
13        "days": 1,
14        "hours": 1,
15        "minutes": 1
16      },
17      "brand": "string",
18      "mode": "string",
19      "sn": "string",
20      "connect_port": 1,
21      "bind_type": 1,
22      "color": "string"
23    }
24  ]
25 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	DHCP client list
data.vlan_id	int32	VLAN ID to which the client

		belongs
data.ip	string	Allocated IP address
data.netmask	string	Subnet mask
data.type	string	IP allocation type: Automatic or Manual
data.host_name	string	Client hostname
data.mac	string	Client MAC address
data.expired_time	object	Remaining lease time
data.expired_time.days	int32	Remaining days
data.expired_time.hours	int32	Remaining hours
data.expired_time.minutes	int32	Remaining minutes
data.brand	string	Device brand/vendor
data.mode	string	Device model
data.sn	string	Device serial number
data.connect_port	int32	Connected switch port number
data.bind_type	int32	Binding type: 0 - Unbound, 1 - IP binding, 2 - MAC binding, 3 - IP+MAC binding
data.color	string	VLAN color

- GET /v1/switch/dhcps/get - Get DHCP Configuration
- POST /v1/switch/dhcps/set - Set DHCP

12.5.2 Get VLAN Interface and DHCP Configuration for Specified VLAN

Interface Description

Get VLAN interface and DHCP server configuration for the specified VLAN

Request URL

/v1/switch/dhcps/get

Request Method

GET

Request Parameter Example

```
1 /v1/switch/dhcps/get?vlan_id=10
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_id	int32	True	VLAN ID (location: query)

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "vlan_if": {
6       "vlan_id": 1,
7       "enable": true,
8       "ip": "192.168.1.1",
9       "netmask": "255.255.255.0"
10    },
11    "dhcp": {
12      "vlan_id": 1,
13      "enable": true,
14      "ip_pool": {
15        "start": "192.168.1.100",
16        "end": "192.168.1.200"
17      },
18      "gateway": {
19        "type": "default",
20        "ip": "192.168.1.1"
21      },
22      "dns": "8.8.8.8",
23      "lease_time": {
24        "days": 1,
25        "hours": 0,
26        "minutes": 0
27      }
28    }
```

```
29     }  
30 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	VLAN interface and DHCP configuration information
data.vlan_if	object	VLAN interface configuration
data.vlan_if.vlan_id	int32	VLAN ID
data.vlan_if.enable	bool	VLAN interface enable switch
data.vlan_if.ip	string	VLAN interface IP address
data.vlan_if.netmask	string	VLAN interface subnet mask
data.dhcp	object	DHCP server configuration
data.dhcp.vlan_id	int32	VLAN ID
data.dhcp.enable	bool	DHCP server enable switch
data.dhcp.ip_pool	object	DHCP address pool
data.dhcp.ip_pool.start	string	Address pool start IP
data.dhcp.ip_pool.end	string	Address pool end IP
data.dhcp.gateway	object	Gateway configuration
data.dhcp.gateway.type	string	Gateway type: default (use VLAN IP) or manual (manually specified)
data.dhcp.gateway.ip	string	Gateway IP address
data.dhcp.dns	string	DNS server address
data.dhcp.lease_time	object	Lease time
data.dhcp.lease_time.days	int32	Lease days
data.dhcp.lease_time.hours	int32	Lease hours
data.dhcp.lease_time.minutes	int32	Lease minutes

12.5.3 Set VLAN Interface and DHCP Configuration for Specified VLAN

Interface Description

Set VLAN interface and DHCP server configuration for the specified VLAN

Request URL

/v1/switch/dhcps/set

Request Method

POST

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_if	object	True	VLAN interface configuration
vlan_if.vlan_id	int32	True	VLAN ID
vlan_if.enable	bool	True	VLAN interface enable switch
vlan_if.ip	string	True	VLAN interface IP address
vlan_if.netmask	string	True	VLAN interface subnet mask
dhcp	object	True	DHCP server configuration
dhcp.vlan_id	int32	True	VLAN ID
dhcp.enable	bool	True	DHCP server enable switch
dhcp.ip_pool	object	True	DHCP address pool
dhcp.ip_pool.start	string	True	Address pool start IP
dhcp.ip_pool.end	string	True	Address pool end IP
dhcp.gateway	object	True	Gateway configuration
dhcp.gateway.type	string	True	Gateway type: default or manual
dhcp.gateway.ip	string	True	Gateway IP address (required when type is manual)
dhcp.dns	string	False	DNS server address
dhcp.lease_time	object	True	Lease time

dhcp.lease_time.days	int32	True	Lease days
dhcp.lease_time.hours	int32	True	Lease hours
dhcp.lease_time.minutes	int32	True	Lease minutes

Response Example

```
1 {
2     "code": 200,
3     "message": "ok"
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

- POST /v1/switch/dhcps/enable/set - Enable/Disable DHCP

12.5.4 Enable or Disable DHCP Service for Specified VLAN

Interface Description

Enable/disable DHCP server for the specified VLAN

Request URL

/v1/switch/dhcps/enable/set

Request Method

POST

Request Parameter Example

```
1 {
2     "vlan_id": 10,
3     "enable": true
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_id	int32	True	VLAN ID
enable	bool	True	DHCP server switch: true - enabled, false - disabled

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok"  
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

12.5.5 Get Static IP Binding List

Interface Description

Get static IP binding list (manually bound clients)

Request URL

/v1/switch/dhcps/static/get

Request Method

POST

Request Parameter Example

```
1 {  
2   "vlan_id": [10, 20]  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
vlan_id	array	False	VLAN ID list; queries all if not provided

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "vlan_id": 1,
7       "ip": "192.168.1.50",
8       "netmask": "255.255.255.0",
9       "type": "Manual",
10      "host_name": "device-001",
11      "mac": "00:11:22:33:44:55",
12      "expired_time": {
13        "days": 0,
14        "hours": 12,
15        "minutes": 0
16      },
17      "brand": "kiloview",
18      "mode": "DS-2CD2043G2",
19      "sn": "DSGK123456789",
20      "connect_port": 5,
21      "bind_type": 3,
22      "color": "#FF0000"
23    }
24  ]
25 }
```

Response Parameter Description

Parameter name	Parameter type	Description
----------------	----------------	-------------

code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Static IP binding list
data.vlan_id	int32	Associated VLAN ID
data.ip	string	Bound IP address
data.netmask	string	Subnet mask
data.type	string	Allocation type: Manual
data.host_name	string	Client hostname
data.mac	string	Client MAC address
data.expired_time	object	Remaining lease time
data.expired_time.days	int32	Remaining days
data.expired_time.hours	int32	Remaining hours
data.expired_time.minutes	int32	Remaining minutes
data.brand	string	Device brand
data.mode	string	Device model
data.sn	string	Device serial number
data.connect_port	int32	Connected switch port number
data.bind_type	int32	Binding type: 0 - Unbound, 1 - IP binding, 2 - MAC binding, 3 - IP+MAC binding
data.color	string	VLAN color

12.5.6 Batch Set Static IP Binding

Interface Description

Batch set static IP binding (IP/MAC binding) currently only supports port-based binding

Request URL

/v1/switch/dhcps/static/set

Request Method

POST

Request Parameter Example

```
1 {
```

```

2  "data": [
3    {
4      "vlan_id": 10,
5      "ip": "192.168.10.50",
6      "mac": "00:11:22:33:44:55",
7      "bind_port": 5,
8      "op": 1,
9      "bind_type": 3
10   }
11 ]
12 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
data	array	True	Static IP binding list
data.vlan_id	int32	True	VLAN ID
data.ip	string	True	IP address to bind
data.mac	string	True	MAC address to bind
data.bind_port	int32	False	Bound port number (used when bind_type is 1)
data.op	int32	True	Operation type: 1 - Bind, 0 - Unbind
data.bind_type	int32	True	Binding type: 1 - Port-based binding, 2 - MAC-based binding, 3 - Unbound. Currently only port-based binding is supported

Response Example

```

1 {
2   "code": 200,
3   "message": "ok"
4 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

12.6 Power Management

12.6.1 Query Power Information

Interface Description

Get switch power information, etc.

Request URL

/v1/switch/power/status/query

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": [  
5     {  
6       "slot": "slot1",  
7       "name": "AVX-KILOLINK-920-A",
```

```

8      "status": "Slot Empty",
9      "power": ""
10     },
11     {
12         "slot": "slot2",
13         "name": "AVX-KILOLINK-920-B",
14         "status": "Slot Empty",
15         "power": ""
16     },
17     {
18         "slot": "PoE",
19         "name": "PoE Total Power(Real-time)",
20         "status": "Power On",
21         "power": "23.77W"
22     }
23 ]
24 }

```

Response Parameter Description

data is an array of power/power consumption related entries.

data[] element field description

Parameter name	Parameter type	Description
slot	String	Slot or logical position identifier, e.g., slot1, slot2, PoE
name	String	Module or statistics item name
status	String	Status description, e.g., Slot Empty, Power On
power	String	Power string; can be empty string if unavailable (e.g., non-PoE summary items)
Parameter name	Parameter type	Description
code	Number	Business status code; 200 indicates success
message	String	Prompt message
data	Array	The above table is the field

		description for each data[] element
--	--	-------------------------------------

12.7 Static Route

12.7.1 Add Static Route

Interface Description

Add static route

Request URL

/v1/switch/iproutes/add

Request Method

POST

Request Parameter Example

```
1 {
2   "dest_ip": "192.168.2.0",
3   "dest_mask": 24,
4   "next_hop": "192.168.1.1",
5   "priority": 100
6 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
dest_ip	string	True	Target network address (e.g., 192.168.2.0)
dest_mask	int32	True	Target network mask (e.g., 24 represents 255.255.255.0)

next_hop	string	True	Next hop IP address
priority	int32	True	Route priority (lower value = higher priority)

Response Example

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

12.7.2 Delete Static Route

Interface Description

Delete specified static route

Request URL

/v1/switch/iproutes/delete

Request Method

POST

Request Parameter Example

```
1 {
2   "dest_ip": "192.168.2.0",
3   "dest_mask": 24,
4   "next_hop": "192.168.1.1",
5   "priority": 100
6 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
dest_ip	string	True	Target network address
dest_mask	int32	True	Target network mask
next_hop	string	True	Next hop IP address
priority	int32	True	Route priority

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok"  
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

12.7.3 Get All Static Routes List

Interface Description

Get all static route list

Request URL

/v1/switch/iproutes/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "routes": [
5     {
6       "dest_ip": "192.168.2.0",
7       "dest_mask": 24,
8       "next_hop": "192.168.1.1",
9       "priority": 100
10    }
11  ]
12 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
routes	array	Static route list
routes.dest_ip	string	Target network address
routes.dest_mask	int32	Target network mask bits (e.g., 24)
routes.next_hop	string	Next hop IP address
routes.priority	int32	Route priority (lower value = higher priority)

12.8 Network Save

12.8.1 Save Network Configuration

Interface Description

Save network configuration

Request URL

/v1/network/save

Request Method

POST

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": null
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	null	No data returned

12.8.2 Query Network Configuration

Interface Description

Query network configuration

Request URL

/v1/network/query

Request Method

GET

Request Parameter Example

None

Request Parameter Description

None

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "status": false
6   }
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	System information
data.status	bool	Status

13 Alarm Events

13.1 Alarm Event Management

13.1.1 Query Alarm Events by Device

Interface Description

Query alarm events by device.

Request URL

/v1/alarm/dev/events/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
mac	string	False	MAC address (location: query)

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "id": 1,
7       "status": 1,
8       "alarm_level": 1,
9       "time": "2024-01-01 10:00:00",
```

```
10         "message": "设备 CPU 温度过高",
11         "source_port": 0
12     }
13 ]
14 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Device alarm event list
data.id	int32	Alarm event ID
data.status	int32	Alarm status: 1 - Unread, 2 - Read
data.alarm_level	int32	Alarm level: 1 - Error, 2 - Warning, 3 - Info
data.time	string	First occurrence time
data.message	string	Alarm message content
data.source_port	int32	Source port number

13.1.2 Delete Alarm Events

Interface Description

Delete alarm events.

Request URL

/v1/alarm/events/del

Request Method

POST

Request Parameter Example

```
1 {
```

```
2   "ids": [1, 2, 3]
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ids	array	False	List of alarm event IDs to delete, supports batch deletion

Response Example

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

13.1.3 Paginated Query of Alarm Events

Interface Description

Paginated query of alarm events.

Request URL

/v1/alarm/events/get

Request Method

POST

Request Parameter Example

```
1 {
```

```
2  "alarm_level": [1, 2],
3  "start_time": "2024-01-01 00:00:00",
4  "end_time": "2024-01-31 23:59:59",
5  "status": 1,
6  "page_index": 1,
7  "page_size": 20,
8  "port_id": [1, 2],
9  "keyword": "CPU"
10 }
```

When only paginated without other conditions:

```
1 {
2   "page_index": 1,
3   "page_size": 20
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
alarm_level	array<int32>	False	JSON key name; alarm level multi-select. 1 - Error, 2 - Warning, 3 - Info
start_time	string	False	JSON key name; start time, format: yyyy-MM-dd HH:mm:ss
end_time	string	False	JSON key name; end time, format: yyyy-MM-dd HH:mm:ss
status	int32	False	JSON key name; alarm status: 1 - Unread, 2 - Read
page_index	int32	False	JSON key name; page number, recommended to start from 1

page_size	int32	False	JSON key name; items per page
port_id	array<int32>	False	Port number filter
keyword	string	False	JSON key name; fuzzy search for alarm messages, etc.

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "total": 100,
5   "page_index": 1,
6   "page_size": 20,
7   "data": [
8     {
9       "id": 1,
10      "status": 1,
11      "alarm_level": 1,
12      "time": "2024-01-01 10:00:00",
13      "message": "CPU 使用率超过 80%",
14      "source_port": 1
15    }
16  ]
17 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
total	int32	Total number of alarm events
page_index	int32	Current page number
page_size	int32	Data amount per page
data	array	Alarm event list

data.id	int32	Alarm event ID
data.status	int32	Alarm status: 1 - Unread, 2 - Read
data.alarm_level	int32	Alarm level: 1 - Error, 2 - Warning, 3 - Info
data.time	string	First occurrence time
data.message	string	Alarm message content
data.source_port	int32	Source port number

13.1.4 Filter Event List by Alarm Level

Interface Description

Filter event list by alarm level.

Request URL

/v1/alarm/events/levels/get

Request Method

GET

Request Parameter Example

Query example (multi-level):

```
1 /v1/alarm/events/levels/get?alarm_levels=1&alarm_levels=2
```

Error level only:

```
1 /v1/alarm/events/levels/get?alarm_levels=1
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
alarm_levels	array<int32>	False	Query parameter name; multi-select. 1 - Error, 2 - Warning, 3 - Info (location: query, duplicate key: multi)

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "events_num": 50,
5   "error_total": 10,
6   "warning_total": 30,
7   "info_total": 10,
8   "data": [
9     {
10      "id": 1,
11      "status": 1,
12      "alarm_level": 1,
13      "time": "2024-01-01 10:00:00",
14      "message": "CPU 使用率超过 80%",
15      "source_port": 1
16    }
17  ]
18 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
events_num	int32	Total number of alarm events matching filter criteria
data	array	Alarm event list
data.id	int32	Alarm event ID
data.status	int32	Alarm status: 1 - Unread, 2 - Read
data.alarm_level	int32	Alarm level: 1 - Error, 2 - Warning, 3 - Info
data.time	string	First occurrence time
data.message	string	Alarm message content

data.source_port	int32	Source port number
error_total	int32	Number of unread error level alarms
warning_total	int32	Number of unread warning level alarms
info_total	int32	Number of unread info level alarms

13.1.5 Batch Set Alarm Event Status

Interface Description

Batch set alarm event status.

Request URL

/v1/alarm/events/status/set

Request Method

POST

Request Parameter Example

```
1 {  
2   "ids": [1, 2, 3],  
3   "status": 2  
4 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
ids	array	False	List of alarm event IDs to process, supports batch operations
status	int32	False	Target status value: 1 - Unread, 2 - Read

Response Example

```
1 {
2   "code": 200,
3   "message": "ok"
4 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

13.1.6 Download Alarm Export File

Interface Description

Download the alarm exported Excel spreadsheet file

Request URL

/v1/alarm/file/download

Request Method

POST

Request Parameter Example

```
1 {
2   "file_path": "alarm_export_xxx.xlsx"
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
file_path	string	True	File path to download (file_path returned by AlarmEventsListGet interface)

Response Example

Directly returns Excel file download, response headers include:

- Content-Disposition: attachment; filename=alarm_export_xxx.xlsx

Response Parameter Description

Parameter name	Parameter type	Description
-	-	See above (file stream response)

13.1.7 Export Alarm List and Get File Path

Interface Description

Export alarm list and get file path.

Request URL

/v1/alarm/list/export

Request Method

GET

Request Parameter Example

```
1 GET /v1/alarm/list/export
2 ?alarm_level=1
3 &alarm_level=2
4 &start_time=2024-01-01 00:00:00
5 &end_time=2024-12-31 23:59:59
6 &status=1
7 &page_index=1
8 &page_size=100
9 &port_id=1
10 &keyword=error
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
alarm_level	array	False	Alarm level filter, supports multi-select. 1 - Error, 2 - Warning, 3 - Info
start_time	string	False	Start time, format: yyyy-MM-dd HH:mm:ss
end_time	string	False	End time, format: yyyy-MM-dd HH:mm:ss
status	int32	False	Status filter: 1 - Unread, 2 - Read
page_index	int32	False	Page number
page_size	int32	False	Data amount per page
port_id	array	False	Port number filter
keyword	string	False	Keyword search, fuzzy match alarm message content

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {  
5     "file_path": "string"  
6   }  
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Export file information

data.file_path	string	Exported alarm record Excel file path
----------------	--------	---------------------------------------

13.2 Alarm Configuration

13.2.1 Get Alarm Policy Configuration

Interface Description

Get alarm policy configuration.

Request URL

/v1/alarm/config/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "cpu_enabled": true,
6     "mem_enabled": true,
7     "temp_enabled": true,
8     "alarm_interval": 1,
9     "disconnect_alarm_enabled": true,
```

```
10     "disconnect_duration": 1
11 }
12 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Business data body
data.cpu_enabled	bool	CPU alarm switch
data.mem_enabled	bool	Memory alarm switch
data.temp_enabled	bool	Temperature alarm switch
data.alarm_interval	int32	Alarm interval for the same type on the same device (minutes)
data.disconnect_alarm_enabled	bool	Device unexpected disconnection alarm switch
data.disconnect_duration	int32	Device unexpected disconnection detection duration (seconds)

13.2.2 Set Alarm Policy Configuration

Interface Description

Set alarm policy configuration.

Request URL

/v1/alarm/config/set

Request Method

POST

Request Parameter Example

```
1 {
```

```

2  "cpu_enabled": true,
3  "mem_enabled": true,
4  "temp_enabled": true,
5  "alarm_interval": 5,
6  "disconnect_alarm_enabled": true,
7  "disconnect_duration": 30
8 }

```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
cpu_enabled	bool	False	CPU alarm switch
mem_enabled	bool	False	Memory alarm switch
temp_enabled	bool	False	Temperature alarm switch
alarm_interval	int32	False	Alarm interval for the same type on the same device (minutes)
disconnect_alarm_enabled	bool	False	Device unexpected disconnection alarm switch
disconnect_duration	int32	False	Device unexpected disconnection detection duration (seconds); if the device reconnects within the set time, no alarm will be triggered

Response Example

```

1 {
2   "code": 200,
3   "message": "ok"
4 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200

		indicates success
message	string	Business prompt message

13.3 Alarm Debug

13.3.1 Debug: Query All Port Alarms (DEBUG)

Interface Description

This interface is only used for DEBUG debugging, returns alarm event data for all ports

Request URL

/v1/alarm/ports/events/get

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "1": {
6       "has_alarm": true,
7       "has_error": true,
8       "port_alarm_events": [
9         {
10          "id": 1,
```

```
11         "status": 1,
12         "alarm_level": 1,
13         "time": "2024-01-01 10:00:00",
14         "message": "端口 LinkDown",
15         "source_port": 1
16     }
17 ]
18 },
19 "2": {
20     "has_alarm": false,
21     "has_error": false,
22
23     "port_alarm_events": []
24
25 }
26 }
27 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	map<string, PortAlarmItem>	Port alarm information, key is port number, value is the alarm details for that port

| data[].has_alarm | bool | Whether this port has alarm events |

| data[].has_error | bool | Whether this port has error level alarms |

| data[].port_alarm_events | array | Alarm event list for this port |

13.3.2 Debug: Query Port Alarm (DEBUG)

Interface Description

This interface is only used for DEBUG debugging

Request URL

/v1/alarm/debug/port

Request Method

POST

Request Parameter Example

```
1 {
2     "port_id": 1
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
port_id	int32	False	Port number

Response Example

```
1 {
2     "code": 200,
3     "message": "ok",
4     "data": {
5         "code": 1,
6         "message": "string",
7         "has_alarm": true,
8         "has_error": true,
9         "alarm_events": [
10            {
11                "id": 1,
12                "status": 1,
```

```

13         "alarm_level": 2,
14         "time": "2024-01-01 10:00:00",
15         "message": "端口 LinkDown",
16         "source_port": 1
17     }
18 ],
19     "error_events": [
20     {
21         "id": 2,
22         "status": 1,
23         "alarm_level": 1,
24         "time": "2024-01-01 10:00:00",
25         "message": "端口 Error",
26         "source_port": 1
27     }
28 ]
29 }
30 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
has_alarm	bool	Whether this port has alarm events (warning or info level)
has_error	bool	Whether this port has error level alarms
alarm_events	array	Alarm event list for this port (warning and info levels)
alarm_events.id	int32	Alarm event ID
alarm_events.status	int32	Alarm status: 1 - Unread, 2 - Read
alarm_events.alarm_level	int32	Alarm level: 1 - Error, 2 - Warning, 3 - Info

alarm_events.time	string	First occurrence time
alarm_events.message	string	Alarm message content
alarm_events.source_port	int32	Source port number
error_events	array	Error-level alarm event list for this port
error_events.id	int32	Alarm event ID
error_events.status	int32	Alarm status: 1 - Unread, 2 - Read
error_events.alarm_level	int32	Alarm level: 1 - Error, 2 - Warning, 3 - Info
error_events.time	string	First occurrence time
error_events.message	string	Alarm message content
error_events.source_port	int32	Source port number

14 Production Test

14.1 Production Test and Device Information

14.1.1 Restore Factory Settings

Interface Description

Restore factory settings.

Request URL

/v1/test/factory_reset

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
----------------	----------------	------------------	-------------

(None)	-	-	No request parameters
--------	---	---	-----------------------

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "code": 1,
6     "message": "string"
7   }
8 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

14.1.2 Get Fan List

Interface Description

Get fan list.

Request URL

/v1/test/fan_list

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": [  
5     "string"  
6   ]  
7 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Business data body

14.1.3 Get Device Resources and Version Information

Interface Description

Get device resources and version information.

Request URL

/v1/test/get_resources

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "exchange_board": {
6       "cpu_usage": "string",
7       "mem_usage": "string",
8       "temperature": "string",
9       "software_version": "string",
10      "hardware_version": "string"
11    },
12    "ev4": {
13      "cpu_usage": "string",
14      "mem_usage": "string",
15      "temperature": "string",
16      "software_version": "string",
17      "hardware_version": "string"
18    },
19    "a311d2": {
20      "cpu_usage": "string",
21      "mem_usage": "string",
22      "temperature": "string",
23      "software_version": "string",
24      "hardware_version": "string"
```

```

25     },
26     "serial_number": "string",
27     "firmware_version": "string",
28     "hardware_version": "string"
29 }
30 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Business data body
data.exchange_board	object	Switch board resource information
data.exchange_board.cpu_usage	string	CPU usage rate
data.exchange_board.mem_usage	string	Memory usage rate
data.exchange_board.temperature	string	Temperature
data.exchange_board.software_version	string	Software version number
data.exchange_board.hardware_version	string	Hardware version number
data.ev4	object	4EV board resource information
data.ev4.cpu_usage	string	CPU usage rate
data.ev4.mem_usage	string	Memory usage rate
data.ev4.temperature	string	Temperature
data.ev4.software_version	string	Software version number
data.ev4.hardware_version	string	Hardware version number
data.a311d2	object	A311D2 board resource information
data.a311d2.cpu_usage	string	CPU usage rate
data.a311d2.mem_usage	string	Memory usage rate
data.a311d2.temperature	string	Temperature
data.a311d2.software_version	string	Software version number
data.a311d2.hardware_version	string	Hardware version number
data.serial_number	string	Device serial number
data.firmware_version	string	Firmware version number

data.hardware_version	string	Hardware version number
-----------------------	--------	-------------------------

14.1.4. Query Switch Port Status

Interface Description

Query switch port status.

Request URL

/v1/test/port_status

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": [
5     {
6       "port": 1,
7       "speed_duplex": "string",
8       "link": true,
9       "poe_power": "string",
10      "test_result": "string"
11    }
12  ]
13 }
```

```
11     }
12   ]
13 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	array	Business data body
data.port	int32	Returned port number
data.speed_duplex	string	String parameter
data.link	bool	Boolean switch value
data.poe_power	string	String parameter
data.test_result	string	Returned result

14.1.5 Set Serial Number

Interface Description

Set serial number.

Request URL

/v1/test/set_serial_number

Request Method

POST

Request Parameter Example

```
1 {
2   "serial_number": "SN20240001"
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
----------------	----------------	------------------	-------------

serial_number	string	True	Device serial number
---------------	--------	------	----------------------

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "code": 1,
6     "message": "string"
7   }
8 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

14.1.6 Start Fan Test

Interface Description

Start fan test.

Request URL

/v1/test/start_fan_test

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {
2   "code": 200,
3   "message": "ok",
4   "data": {
5     "fans": [
6       {
7         "name": "string",
8         "status": "string",
9         "rpm": 1
10      }
11    ]
12  }
13 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message
data	object	Business data body
data.fans	array	Fan test result list
data.fans.name	string	Returned name
data.fans.status	string	Returned status value
data.fans.rpm	int32	Numeric parameter

14.1.7 Stop Fan Test

Interface Description

Stop fan test.

Request URL

/v1/test/stop_fan_test

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": {  
5     "code": 1,  
6     "message": "string"  
7   }  
8 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success

message	string	Business prompt message
---------	--------	-------------------------

14.1.8 Query USB Port Status

Interface Description

Query USB port status.

Request URL

/v1/test/usb_port_status

Request Method

GET

Request Parameter Example

```
1 {}
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
(None)	-	-	No request parameters

Response Example

```
1 {  
2   "code": 200,  
3   "message": "ok",  
4   "data": "string"  
5 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int32	Business status code; 200 indicates success
message	string	Business prompt message

data	string	Business data body
------	--------	--------------------

15 AVX Page Interface

15.1 Get System Information

Interface Description

Get System Information

Request URL

/api/systemctrl/report/get_system_info

Request Method

GET

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": {
5     "hostname": "AVX24-4",
6     "version": {
7       "hardwareVersion": "V1.0",
8       "softwareVersion": "V2.0.0",
9       "firmwareVersion": "V1.5.0",
10      "serialnumber": "SN0000000000001",
11      "ndiVersion": "NDI 5.5",
12      "product": "AVX24-4",
13      "oem": "Brand"
14    },
15    "mac": "00:11:22:33:44:55",
16    "mode": "L2",
17    "cpu": {
```

```

18         "percent": 25
19     },
20     "temperature": 45,
21     "mem": {
22         "percent": 60,
23         "total": 4096,
24         "used": 2458
25     },
26     "disk": {
27         "percent": 30,
28         "total": 10240,
29         "used": 3072
30     }
31 }
32 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	object	System information
data.hostname	string	Hostname
data.version	object	System version
data.version.hardwareVersion	string	Hardware version
data.version.softwareVersion	string	Software version
data.version.firmwareVersion	string	Firmware version
data.version.serialnumber	string	Serial number
data.version.ndiVersion	string	NDI version
data.version.product	string	Product
data.version.oem	string	OEM
data.mac	string	MAC address
data.mode	string	Mode
data.cpu	object	CPU usage
data.cpu.percent	int	CPU usage rate
data.temperature	int	Temperature

data.mem	object	MEM usage
data.mem.percent	int	MEM usage rate
data.mem.total	int	MEM total size
data.mem.used	int	MEM used size
data.disk	object	Disk usage
data.disk.percent	int	Disk usage rate
data.disk.total	int	Disk total size
data.disk.used	int	Disk used size

15.2 Get Network Interface

Interface Description

Get Network Interface Information

Request URL

/api/networkmanager/report/get_network

Request Method

GET

Response Example

```
1 {
2   "code": 200,
3   "message": "Success",
4   "data": [
5     {
6       "name": "eth1.1",
7       "type": "eth",
8       "mac": "00:11:22:33:44:55",
9       "address": "192.168.1.100",
10      "netmask": "255.255.255.0",
11      "gw": "192.168.1.1",
12      "dns": "8.8.8.8",
13      "method": "static",
```

```

14         "status": "up",
15         "multicast": "yes",
16         "receive": 102400,
17         "transmit": 51200
18     }
19 ]
20 }

```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
message	string	Operation result description
data	array	Network interface information
data[].name	string	Interface name
data[].type	string	Interface type (fixed "eth")
data[].mac	string	MAC address
data[].address	string	IP address
data[].netmask	string	Subnet mask
data[].gw	string	Gateway address
data[].dns	string	DNS server
data[].method	string	IP acquisition method (DHCP/static)
data[].status	string	Interface status (up/down)
data[].multicast	string	Whether multicast is supported
data[].receive	int	Receive traffic (KB)
data[].transmit	int	Send traffic (KB)

16 AVX Reverse Proxy Device Page Interface

16.1 Get Child Device iframe Page Interface

Interface Description

Get child device iframe page interface information

Request URL

/v1/local/klnk/forward

Request Method

GET

Request Parameter Example

```
1 {  
2   "id": "18:C3:F4:21:D7:2E"  
3 }
```

Request Parameter Description

Parameter name	Parameter type	Whether required	Description
id	string	Yes	Child device MAC address

Response Example

```
1 {  
2   "code": 200,  
3   "msg": "Success",  
4   "requestId": "963f4f61-9079-44dd-82b7-820ce11403a5",  
5   "data": [  
6     {  
7       "serial_number": "201413001D72E",  
8       "id": "",  
9       "action": "iframeDrawer",  
10      "path":  
11        "/iframeDrawer/6558d1db07503352fb1211ffee941875/39651/7907/iframeDrawer",  
12      "pathParams": {  
13        "channel": "",  
14        "type": ""  
15      },  
16      "isWorking": true,  
17    }  
18   ]  
19 }
```

```
16         "dev_ip": "",
17         "alias": ""
18     }
19 ]
20 }
```

Response Parameter Description

Parameter name	Parameter type	Description
code	int	Status code; 200 indicates success
msg	string	Operation result description
requestId	string	Request ID
data	array	Network interface information
data[].serial_number	string	Child device serial number
data[].id	string	Device ID
data[].action	string	Operation method
data[].path	string	Reverse proxy path
data[].pathParams	object	Reverse proxy path parameter
data[].pathParams.channel	string	Channel
data[].pathParams.type	string	Type
isWorking	bool	Whether running normally
dev_ip	string	IP address
alias	string	Alias