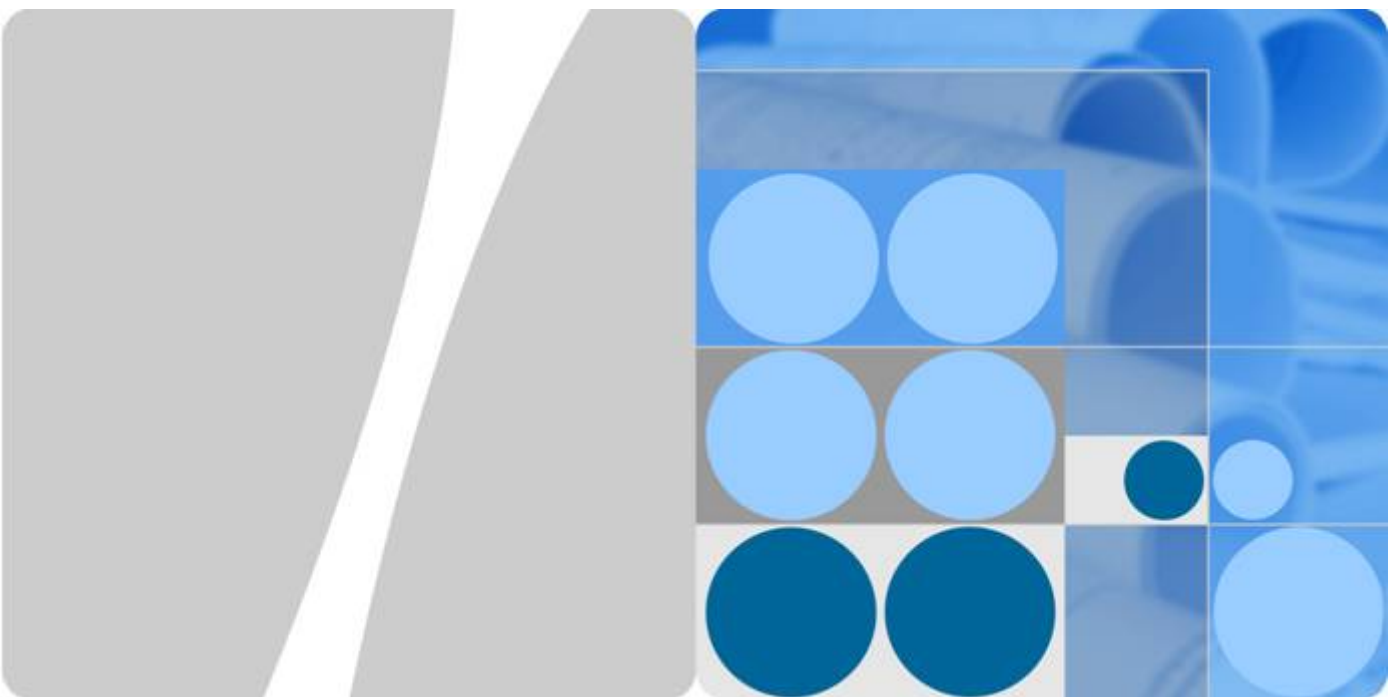




WLAN AC

V200R023C00

RESTful API Reference and Usage Guide

Issue **01**

Date **2024-04-02**

HUAWEI TECHNOLOGIES CO., LTD.



Copyright © Huawei Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <https://e.huawei.com>

Contents

1 Overview of APIs.....

1

1.1 Constraints and Precautions.....

1

1.2 Service Usage Method.....

2

2 API Reference.....

3

2.1 Token Configuration.....

3

2.1.1 Obtaining a Token.....

3

2.2 AC Configuration.....

5

2.2.1 Configuring AC Account Information.....

5

2.2.1.1 Changing the Password.....

5

2.2.1.2 Adding an Account.....

7

2.2.1.3 Deleting an Account.....

8

2.3 AP Configuration.....

9

2.3.1 Configuring an AP.....

10

2.3.1.1 Registering an AP.....

10

2.3.2 Managing Certificates.....

11

2.3.2.1 Requesting to Obtain the P10 File.....

11

2.3.2.2 Installing an AP-WAPI Certificate.....

12

2.3.3 Managing Radios.....

14

2.3.3.1 Configuring a Radio.....

14

2.3.4 Managing Wireless Services.....

15

2.3.4.1 Obtaining a Wireless Service Template.....

16

2.3.4.2 Binding a Wireless Service Template.....

18

2.3.4.3 Configuring a Whitelist.....

19

2.3.4.4 Configuring a Blacklist.....

20

2.4 STA Management.....

22

2.4.1 Managing STAs.....

22

2.4.1.1 Forcibly Disconnecting STAs.....

22

2.5 System Maintenance.....

23

2.5.1 Configuring Backup Management.....

23

2.5.1.1 Backing Up Configurations.....

23

2.5.1.2 Restoring Configuration.....

24

1 Overview of APIs

About This Chapter

[1.1 Constraints and Precautions](#)

[1.2 Service Usage Method](#)

1.1 Constraints and Precautions

Before using application programming interfaces (APIs) in this document, carefully read the following information:

- All APIs in this document can only be connected to the integrated network management system (NMS) of the power grid.
- To invoke APIs in this document, you need to use the HTTPS 1.1 protocol and the UTF-8 character set.
- The data returned by the APIs in this document is in JSON format.
- Token authentication is required to invoke APIs.
 - Token request requirements: Tokens are generated by device vendors. Before the configuration, the NMS obtains the token values through the account and password request. The token parameters are contained in the HTTPS request headers.
 - Token verification requirements: Each service interface (except the token obtaining interface) verifies the token validity before processing services. If a token expires or is invalid, the unified JSON structure of the NMS is returned.

```
Data: {  
  "code":333,  
  "msg": "Token is invalid!"  
}
```
- Encryption algorithm: This document uses the electronic codebook (ECB) mode of the SM4 encryption algorithm. The following lists information required to implement the algorithm:
 - Algorithm name: SM4

- Mode: ECB
- Key length: 128 bits
- Padding mode: PKCS7 padding
- Encryption result encoding mode: Base64
- Key configuration: The key is generated by the NMS and delivered to the AC through SNMP. The corresponding MIB object is csgDot11SecretKey with the OID of 1.2.156.11235.6001.60.7.2.75.1.1.2.33.

1.2 Service Usage Method

Representational State Transfer (REST) determines resources distributed over the entire network by Uniform Resource Identifier (URI). Applications on clients obtain resources through uniform resource locator (URL).

The HTTP protocol defines various request methods, such as GET, PUT, POST, and DELETE, to specify an action to be performed on a specified resource.

Table 1-1 List of request methods

Method	Description
GET	Requests a server to return specified resources.
PUT	Requests a server to update specified resources.
POST	Requests a server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, such as objects.

2 API Reference

About This Chapter

[2.1 Token Configuration](#)

[2.2 AC Configuration](#)

[2.3 AP Configuration](#)

[2.4 STA Management](#)

[2.5 System Maintenance](#)

2.1 Token Configuration

2.1.1 Obtaining a Token

Typical Scenario

The NMS sends the account and password encrypted using a key to the device. The password is encrypted using the SM4 algorithm in ECB mode to obtain a binary array, and then converted into a string using Base64. After receiving the binary array, the device decrypts and verifies the binary array. After successful decryption and verification, the device returns the token value to the NMS. Changing the account password does not affect the generated valid token values. When concurrent requests occur, multiple token values obtained by the NMS take effect simultaneously.

Invoke Method

POST

Request Address

`https://dest_ip:26868/wapi_config/get_token`

Request Parameters

Table 2-1 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
acAccountName	Yes	String	1-64	Account name.
acAccountPassword	Yes	String	16-128	Base64-encoded password encrypted using the SM4 algorithm.

Request Example

```
Data: {
  "acAccountName": "xxxxx",
  "acAccountPassword ": "xxxxx"
}
```

Response Parameters

Table 2-2 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
code	Yes	Int	-	• 200: success • 201: failure • 202: decryption failed because the key is invalid or not configured
token	Yes	String	64	Token character string.
expiryTime	No	Date	-	Expiration time in the yyyy/MM/dd HH:mm:ss format.
msg	No	String	1-255	Description of a success or failure.

Response Example

```
# Return a success result.
Data: {
  "code": 200,
```

```
"token": "",  
"expiryTime": "",  
"msg": "reason....."  
}
```

Return a failure result.

```
Data: {  
  "code": 201,  
  "token": null,  
  "expiryTime": "",  
  "msg": "reason....."  
}
```

Return an encryption failure result when the key is invalid or not configured.

```
Data: {  
  "code": 202,  
  "token": null,  
  "expiryTime": "",  
  "msg": "reason....."  
}
```

2.2 AC Configuration

2.2.1 Configuring AC Account Information

2.2.1.1 Changing the Password

Typical Scenario

This API is used to change the password of an AC account. The password transmitted over the API is encrypted using the SM4 encryption algorithm in ECB mode and the key configured on the NMS to obtain a binary array. Then, the binary array is converted into the string format using Base64.

Invoke Method

POST

Request Address

https://dest_ip:26868/ac_config/update_password

Request Parameters

Table 2-3 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
acAccountName	Yes	String	1-64	Account name.
acAccountPassword	Yes	String	16-128	Old Base64-encoded password encrypted using the SM4 algorithm.
acAccountNewPassword	Yes	String	16-128	New Base64-encoded password encrypted using the SM4 algorithm.

Request Example

```
Data: {
  "acAccountName": "xxxxx",
  "acAccountPassword": "xxxxx",
  "acAccountNewPassword": "xxxxx"
}
```

Response Parameters

Table 2-4 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	Yes	String	1-255	Description of a success or failure.

Response Example

```
# Return a success result.

Data: {
  "code": 200,
  "msg": "reason....."
}

# Return a failure result.

Data: {
  "code": 201,
```

```
    "msg": "reason....."
}
```

2.2.1.2 Adding an Account

Typical Scenario

This API is used to add an AC account. The password transmitted over the API is encrypted using the SM4 encryption algorithm in ECB mode and the key configured on the NMS to obtain a binary array. Then, the binary array is converted into the string format using Base64.

Invoke Method

POST

Request Address

https://dest_ip:26868/ac_config/create_account

Request Parameters

Table 2-5 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
acAccountNa me	Yes	String	1–64	Account name.
acAccountPas sword	Yes	String	16–128	Base64-encoded password encrypted using the SM4 algorithm.
acAccountRol e	Yes	Int	-	Role. • 1: common administrator • 2: enterprise administrator • 3: super administrator

Request Example

```
Data: {
  "acAccountName": "xxxxx",
  "acAccountPassword": "xxxxx",
  "acAccountRole": x
}
```

Response Parameters

Table 2-6 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	No	String	1-255	Description of a success or failure.

Response Example

```
# Return a success result.  
Data: {  
  "code": 200,  
  "msg": "reason....."  
}  
  
# Return a failure result.  
Data: {  
  "code": 201,  
  "msg": "reason....."  
}
```

2.2.1.3 Deleting an Account

Typical Scenario

This API is used to delete an AC account.

Invoke Method

POST

Request Address

https://dest_ip:26868/ac_config/delete_account

Request Parameters

Table 2-7 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
acAccountName	Yes	String	1–64	Account name.

Request Example

```
Data: {
  "acAccountName": "xxxxx"
}
```

Response Parameters

Table 2-8 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	No	String	1–255	Description of a success or failure.

Response Example

```
# Return a success result.

Data: {
  "code": 200,
  "msg": "reason....."
}

# Return a failure result.

Data: {
  "code": 201,
  "msg": "reason....."
}
```

2.3 AP Configuration

2.3.1 Configuring an AP

2.3.1.1 Registering an AP

Typical Scenario

This API is used to register an AP when the AP needs to access the network or the AC is replaced.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/set_registration

Request Parameters

Table 2-9 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
apSn	Yes	String	1–128	AP SN.

Request Example

```
Data: {  
  "apSn": "xxxxx"  
}
```

Response Parameters

Table 2-10 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	No	String	1–255	Description of a success or failure.

Response Example

Return a success result.

```
Data: {
  "code": 200,
  "msg": "reason....."
}
```

Return a failure result.

```
Data: {
  "code": 201,
  "msg": "reason....."
}
```

2.3.2 Managing Certificates

2.3.2.1 Requesting to Obtain the P10 File

Typical Scenario

This API is used to obtain the P10 file of an AP.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/get_p10

Request Parameters

Table 2-11 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
apSn	Yes	String	1-128	AP SN.
deviceDomain	Yes	String	1-50	Device domain.
manageDomain	Yes	String	1-50	Management domain.
networkDomain	Yes	String	1-50	Network domain.

Request Example

```
Data: {
  "apSn": "xxxxx",
}
```

```
"deviceDomain": "xxxxx",
"manageDomain": "xxxxx",
"networkDomain": "xxxxx"
}
```

Response Parameters

Table 2-12 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200: success 201: failure
certCsr	No	String	1-3000	P10 file content.
msg	Yes	String	1-255	Description of a success or failure.

Response Example

Return a success result.

```
Data: {
  "code": 200,
  "certCsr": "-----BEGIN CERTIFICATE REQUEST-----...",
  "msg": "reason....."
}
```

Return a failure result.

```
Data: {
  "code": 201,
  "certCsr": "null",
  "msg": "reason....."
}
```

2.3.2.2 Installing an AP-WAPI Certificate

Typical Scenario

This API is used to install the WAPI certificate generated by the ASU before an AP accesses the network.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/set_cert

Request Parameters

Table 2-13 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
apSn	Yes	String	1-128	AP SN.
certFileName	Yes	String	1-200	Certificate name.
certFileRoot	Yes	String	1-3000	Root certificate content.
certFileInfom ation	Yes	String	1-3000	User certificate content.

Request Example

```
Data: {  
  "apSn": "xxxxx",  
  "certFileName": "xxxxx",  
  "certFileRoot": "xxxxx",  
  "certFileInfomation": "xxxxx"  
}
```

Response Parameters

Table 2-14 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200: success 201: failure
msg	Yes	String	1-255	Description of a success or failure.

Response Example

```
# Return a success result.  
  
Data: {  
  "code": 200,  
  "msg": "reason....."  
}  
  
# Return a failure result.  
  
Data: {  
  "code": 201,
```

```
    "msg": "reason....."  
}
```

2.3.3 Managing Radios

2.3.3.1 Configuring a Radio

Typical Scenario

This API is used to configure radio parameters for an AP.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/set_radio

Request Parameters

Table 2-15 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
apSn	Yes	String	1–128	AP SN.
radioType	Yes	Int	-	Radio type. • 1: 2.4 GHz • 2: 5.2 GHz • 3: 5.8 GHz Only one of the 5.2 GHz and 5.8 GHz frequency bands takes effect at a time. If the 5.2 GHz and 5.8 GHz frequency bands are both configured for a radio, the latest configuration takes effect.
openFlag	Yes	Int	-	Whether to enable a radio. • 1: enabled • 0: disabled
bandWidth	Yes	Int	-	Channel bandwidth, in MHz.

Parameter	Mandatory or Not	Type	Value Range	Description
channel	Yes	Int	-	Channel (0, 1, 2, 3, 4, etc.). The value 0 indicates automatic channel selection.
outputPower	Yes	Int	-	Output power, in dBm.

Request Example

```
Data: {  
  "apSn": "xxxxx",  
  "radioType": x,  
  "bandWidth": x,  
  "channel": x,  
  "outputPower": "xxxxx"  
}
```

Response Parameters

Table 2-16 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
code	Yes	Int	-	• 200: success • 201: failure
msg	Yes	String	1–255	Description of a success or failure.

Response Example

Return a success result.

```
Data: {  
  "code": 200,  
  "msg": "reason....."  
}
```

Return a failure result.

```
Data: {  
  "code": 201,  
  "msg": "reason....."  
}
```

2.3.4 Managing Wireless Services

2.3.4.1 Obtaining a Wireless Service Template

Typical Scenario

This API is used to obtain the wireless service template list preset by the AC vendor before you can configure AP wireless services.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/get_template

Request Parameters

Table 2-17 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
ssidTemplateName	No	String	1-128	Wireless service template name. If this parameter is left empty, data of all templates is obtained. If this parameter is set, data of the specified template is obtained.

Request Example

```
Data: {  
  "ssidTemplateName": "XXXX",  
}
```

Response Parameters

Table 2-18 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
templateList	Yes	Array List	1-128	Wireless service template list.
ssidTemplateName	Yes	String	1-128	Wireless service template name.

Parameter	Man dator y or Not	Type	Value Range	Description
ssidName	Yes	String	1-64	Wireless service name.
authenType	Yes	Int	-	Authentication mode. • 1: WAPI certificate authentication • 2: WAPI-PSK authentication • 3: other
vlan	Yes	Int	-	Default VLAN.
staWhiteMacList	No	Array List	-	Whitelist.
staBlackMacList	No	Array List	-	Blacklist.
code	Yes	Int	-	200 : success; 201 : failure.
msg	Yes	String	1-255	Description of a success or failure.

Response Example

Return a success result.

```
Data: {
  "templateList": [
    {
      "ssidTemplateName": "xxxx",
      "ssidName": "xxxx",
      "authenType": x,
      "vlan": x,
      "staWhiteMacList": [aa,bb,cc,...],
      "staBlackMacList": [aa,bb,cc,...]
    },{
      "ssidTemplateName": "xxxx",
      "ssidName": "xxxx",
      "authenType": x,
      "vlan": x,
      "staWhiteMacList": [aa,bb,cc,...],
      "staBlackMacList": [aa,bb,cc,...]
    },.....
  ],
  "code": 200,
  "msg": "reason....."
}
```

Return a failure result.

```
Data: {
  "templateList": {},
  "code": 201,
  "msg": "reason....."
}
```

2.3.4.2 Binding a Wireless Service Template

Typical Scenario

This API is used to bind or unbind an AP to or from a wireless service template.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/set_template

Request Parameters

Table 2-19 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
ssidTemplateName	Yes	String	1–128	Template name.
apSn	Yes	String	1–128	AP SN.
type	Yes	Int	-	1: binding 2: unbinding

Request Example

```
Data: {
  "ssidTemplateName": "xxxxx",
  "apSn": "xxxxx",
  "type": x
}
```

Response Parameters

Table 2-20 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200: success; 201: failure.
msg	Yes	String	1–255	Description of a success or failure.

Response Example

```
# Return a success result.

Data: {
  "code": 200,
  "msg": "reason....."
}

# Return a failure result.

Data: {
  "code": 201,
  "msg": "reason....."
}
```

2.3.4.3 Configuring a Whitelist

Typical Scenario

This API is used to configure the AP whitelist on a WLAN.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/set_white

Request Parameters

Table 2-21 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
ssidTemplateName	Yes	String	1–128	Wireless service template name.
staWhiteMacList	Yes	Array List	-	Whitelist.
type	Yes	Int	-	1: add 2: delete

Request Example

```
Data: {
  "ssidTemplateName": "XXXX",
```

```
"staWhiteMacList ": [aa,bb,cc,...],
"type": x
}
```

Response Parameters

Table 2-22 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	Yes	String	1–255	Description of a success or failure.

Response Example

```
# Return a success result.
Data: {
  "code": 200,
  "msg": "reason....."
}

# Return a failure result.
Data: {
  "code": 201,
  "msg": "reason....."
}
```

2.3.4.4 Configuring a Blacklist

Typical Scenario

This API is used to configure the AP blacklist on a WLAN.

Invoke Method

POST

Request Address

https://dest_ip:26868/ap_config/set_black

Request Parameters

Table 2-23 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
ssidTemplate Name	Yes	String	1–128	Wireless service template name.
staBlackMacLi st	Yes	Array List		Whitelist.
type	Yes	Int		1: add 2: delete

Request Example

```
Data: {  
  "ssidTemplateName": XXXX,  
  "staBlackMacList ": [aa,bb,cc,...],  
  "type": x  
}
```

Response Parameters

Table 2-24 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	Yes	String	1–255	Description of a success or failure.

Response Example

```
# Return a success result.  
  
Data: {  
  "code": 200,  
  "msg": "reason....."  
}  
  
# Return a failure result.  
  
Data: {  
  "code": 201,  
  "msg": "reason....."  
}
```

2.4 STA Management

2.4.1 Managing STAs

2.4.1.1 Forcibly Disconnecting STAs

Typical Scenario

This API is used to forcibly disconnect some STAs on a WLAN.

Invoke Method

POST

Request Address

https://dest_ip:26868/sta_config/set_staout

Request Parameters

Table 2-25 body parameters

Parameter	Mandatory or Not	Type	Value Range	Description
apSn	Y	String	1-128	AP SN.
staMac	Y	String	1-64	STA's MAC address.
time	N	Int	-	The unit is minute.

Request Example

```
Data: {
  "apSn": "xxxxx",
  "staMac": "xxxxx",
  "time": x
}
```

Response Parameters

Table 2-26 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	Yes	String	1-255	Description of a success or failure.

Response Example

Return a success result.

```
Data: {  
  "code": 200,  
  "msg": "reason....."  
}
```

Return a failure result.

```
Data: {  
  "code": 201,  
  "msg": "reason....."  
}
```

2.5 System Maintenance

2.5.1 Configuring Backup Management

2.5.1.1 Backing Up Configurations

Typical Scenario

This API is used to back up the device configuration data of the AC in a specified file.

Invoke Method

POST

Request Address

https://dest_ip:26868/wapi_config/backup

Request Parameters

Table 2-27 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
backupName	Yes	String	1-64	Configuration backup name.

Request Example

```
Data: {  
  "backupName": "xxxxx"  
}
```

Response Parameters

Table 2-28 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	Yes	String	1-255	Description of a success or failure.

Response Example

```
# Return a success result.  
  
Data: {  
  "code": 200,  
  "msg": "reason....."  
}  
  
# Return a failure result.  
  
Data: {  
  "code": 201,  
  "msg": "reason....."  
}
```

2.5.1.2 Restoring Configuration

Typical Scenario

This API is used to restore backup configuration of the AC.

 NOTE

After the configuration is restored, restart the device to make the restored configuration take effect.

Invoke Method

POST

Request Address

https://dest_ip:26868/wapi_config/backup_return

Request Parameters

Table 2-29 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
backupName	Yes	String	1-64	Configuration backup name.

Request Example

```
Data: {  
  "backupName": "xxxxx",  
}
```

Response Parameters

Table 2-30 body parameters

Parameter	Man dator y or Not	Type	Value Range	Description
code	Yes	Int	-	200 : success; 201 : failure.
msg	Yes	String	1-255	Description of a success or failure.

Response Example

Return a success result.

```
Data: {  
  "code": 200,  
  "msg": "reason....."  
}
```

Return a failure result.

```
Data: {  
  "code": 201,  
  "msg": "reason....."  
}
```