ACTAtek REST API Manual

Version 0.1

ACTAtek (UK) Ltd.

Last Update :2025.December

ACTAtek REST API Manual

Offices:

Asia and the Rest of the World:

Unit 913-914, 9/F., Worldwide Industrial Centre, 43-47 Shan Mei Street, Fotan, Shatin, N.T., Hong Kong

Tel: (852) 2319 1333

E-mail: info@actatek.com

America:

ACTAtek Technologies Inc.

411-1221 HOMER ST,VANCOUVER BC V6B 1C5,Canada

Phone: (+1) 604 314 7628

E-mail: info@actatek.com

Europe:

ACTAtek (UK) Ltd.

118 Pall Mall, London SW1Y 5EA, U.K.

Phone: (44) 7344 2638 81

E-mail: info@actatek.com

1. Session Establishment

1.1 Login

Description: To get session ID for further action. session ID will automatically timeout after 30mins no future action.

REQUEST

Method	POST
Path	/cgi-bin/restapi/login
Example URI	/cgi-bin/restapi/login

Example JSON	{
	“username”:”xxx”,
	“password”:”xxxxxx”
	}

Request Parameters:

name	type	description
username	string	User ID
password	string	user password

RESPONSE

Response Code	200

example	{
	“status”:”success”,
	“sessionid”:xxxxxxx
	}

Response data

name	type	description

status	string	request status
sessionid	long	ID for further action

1.2. Logout

Description: remove the sessionid

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/logout/{sessionid}
Example URI	/cgi-bin/restapi/logout/15887756

Request Parameters:

name	type	description
{sessionid}: session id want to disable	long	session id want to disable

RESPONSE

Response Code	200

example	{
	“status”:”success”
	}

Response data

name	type	description
status	string	status response

2. Log Management

2.1. getLogs

Description: Get event logs from device

REQUEST

Method	GET
Path	/cgi-bin/restapi/getLogs/{sessionid} ?from &to &employeeName &employeeID &departmentID &trigger &terminalSN
Example URI	/cgi-bin/restapi/getLogs/1607225982?from=2025-07-16T00:00:00&to=2025-07-20T00:00:00 &employeeName=John&employeeID=A123& departmentID=General&trigger=REMOTE DOOR OPEN&terminalSN=00111D000000

Request Parameters in Query:

name	type	description
{sessionid}:	long	session id , required
from	xsd:datetime	if datatime contain 'Z', it is UTC time, e.g. 2025-07-16T10:24:13Z if without 'Z', it is local time, e.g. 2025-07-16T10:24:13 it will affect the output log time format.
to	xsd:datetime	same as (from), if both from and to is UTC time, the output event logs will use UTC time
employeeName	string	name of employee
employeeID	string	ID of employee
departmentID	integer	department ID
trigger	string	event name
terminalSN	string	SN of device, device may

		contain other device eventlogs.

RESPONSE

Response Code	200

example	[
	{
	"logID": 1,
	"userID": "A999",
	"timestamp": "2025-07-16T10:24:13",
	"trigger": "REMOTE DOOR OPEN",
	"terminalSN": "00111D000000"
	},
	{
	"logID": 2,
	"userID": "A999",
	"timestamp": "2025-07-16T10:26:43",
	"trigger": "REMOTE DOOR OPEN",
	"terminalSN": "00111D000000"
	}
	]

Response data: Array of event logs.

name	type	description
logID	integer	id of log
userID	string	ID of employee
timestamp	xsd:datetime	time of event log
trigger	string	event name
terminalSN	string	SN belong to this event log

2.2. getFullLogs

Description: Get event logs from device with details

REQUEST

Method	GET
Path	/cgi-bin/restapi/getLogs/{sessionid}/{limit} ?from &to &employeeName &employeeID &departmentID &trigger &terminalSN
Example URI	/cgi-bin/restapi/getLogs/1607225982/10?from =2025-07-16T00:00:00&to=2025-07-20T00:00 :00&employeeName=John&employeeID=A12 3&departmentID=General&trigger=REMOTE DOOR OPEN&terminalSN=00111D000000

Request Parameters in Query:

name	type	description
{sessionid}:	long	session id , required
{limit}:	integer	limit the count of output response array, if not present, mean no limit.
from	xsd:datetime	if datatime contain ‘Z’, it is UTC time, e.g. 2025-07-16T10:24:13Z if without ‘Z’, it is local time, e.g. 2025-07-16T10:24:13 it will affect the output log time format.
to	xsd:datetime	same as (from), if both from and to is UTC time, the output event logs will use UTC time
employeeName	string	name of employee
employeeID	string	ID of employee
departmentID	integer	department ID
trigger	string	event name
terminalSN	string	SN of device, device may

		contain other device eventlogs.

RESPONSE

Response Code	200
example	[{ "logID": 9355, "userID": "111", "userName": "Edwin Li", "departmentName": "General", "timestamp": "2020-04-06T19:11:16", "trigger": "IN", "terminalSN": "00111D000000", "terminalName": "ACTAtek", "jpegPhoto": “xxxx", "accessMethod": "FACE", "remarks": "#FACE#" }, { "logID": 9356, "userID": "222", "userName": "Sing Chiu", "departmentName": "General", "timestamp": "2020-04-06T19:24:47", "trigger": "IN", "terminalSN": "00111D000000", "terminalName": "ACTAtek", "jpegPhoto": "xxxxxx", "accessMethod": "FACE", "remarks": "#FACE#" },]

Response data: Array of event logs.

name	type	description
logID	integer	id of log
userID	string	ID of employee
userName	string	combine with User Firstname and Lastname
departmentName	string	department name, only the first one will be shown
timestamp	xsd:datetime	time of event log
trigger	string	event name
terminalSN	string	SN belong to this event log
terminalName	string	terminal name
jpegPhoto	Base64 encoded string	JPEG data, if no photo, this element will not be represented
accessMethod	string	Access method description
remarks	string	remark of event log

2.3. getLogPhoto

Description:get the event log photo only by the timestamp

REQUEST

Method	GET
Path	/cgi-bin/restapi/getLogPhoto/{sessionid}/{timestamp} ?terminalSN
Example URI	/cgi-bin/restapi/getLogPhoto/1607225982/2025-07-20T00:00:00?terminalSN=00111D000000

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{timestamp}	xsd:datetime	time of event log, required
terminalSN	string	SN of device, device may contain other device eventlogs. If not present, default device SN will be used

RESPONSE

Response Code	200

example	{
	"status": “success”,
	"jpegPhoto": “xxxx",
	}

Response data:

name	type	description
status	string	status response
jpegPhoto	Base64 encoded string	JPEG photo data

2.4. addLog

Description: Add single event log

REQUEST

Method	POST
Path	/cgi-bin/restapi/addLog/{sessionid}/{timestamp}/{employeeID}/{trigger}
Example URI	/cgi-bin/restapi/addLog/1805350299/2025-07-20T10:20:30/105/IN

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{timestamp}	xsd:datetime	time of event log, required
{employeeID}	string	user ID of the event, required
{trigger}	string	event name, required

RESPONSE

Response Code	200

example	{
	"status": “success”,
	}

Response data:

name	type	description
status	string	status response

2.5. deleteLogs

Description: delete logs before the days to keep.

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/deleteLogs/{sessionid}/{daysToKeep}
Example URI	/cgi-bin/restapi/deleteLogs/1805350299/365

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

{daysToKeep}	integer	Only keep logs from the last X days, required

RESPONSE

Response Code	200

example	{
	"status": “success”,
	}

Response data:

name	type	description
status	string	status response

2.6. resetLog

Description: remove all logs

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/resetLog/{sessionid}
Example URI	/cgi-bin/restapi/resetLog/1805350299

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	{

	"status": “success”, }

Response data:

name	type	description
status	string	status response

3. User Management

3.1 addUser

Description: Add a single user with user structure

REQUEST

Method	POST
Path	/cgi-bin/restapi/user/{sessionid}
Example URI	/cgi-bin/restapi/user/1805350299

Example JSON	{
	"userID": "A0001",
	"lastName": "John",
	"firstName": "Ben",
	"otherName": "Teddy",
	"adminLevel": "PERSONALUSER",
	"password": "123555",
	"groupID": 0,
	"departmentID": 1,
	"status": {
	"active": true,
	"autoMatch": true,
	"smartCard": true,
	"fingerprint": true,

```
    "password": true,
    "facial": true,
    "facialAutoMatch": false
  },
  "fingerprints": "Base64 Enoded",
  "keyrecords":    {
    "keytype": 1,
    "keystring": "According to keytype",
    "keyid": "unique id for this key",
    "keyenabled": true
  },
  "facialrecords":    {
    "FacialTemplateType": 1,
    "FacialTemplate": "xxx",
    "FacialPhoto": "base64 encode"
  },
  "cardsn": "00003",
  "ScoreThreshold": 0,
  "expirydate": 0,
  "message": "user message"
}
```


Request Parameters:

name	type	description
{sessionid}	long	session id, required
userID	string	User ID, required according to the User ID restriction.
lastName	string	last name
firstName	string	First name
otherName	string	other name
adminLevel	string	either ”PERSONALUSER”, ”NETWORKADMIN”, “USERADMIN”, “SUPERADMIN” required if not present or other words,

		default to “PERSONALUSER”
password	string	user defined password
groupID	integer or array of integer	id(s) of Group, if not present add a default 0 id
departmentID	integer or array of integer	id(s) of Department, if not present add a default 0 id
status	single object of status	required
active	boolean	if active is false, this user is suspended.required
autoMatch	boolean	true when using fingerprint automatch. required
smartCard	boolean	true when using smartcard. required
fingerprint	boolean	true when using fingerprint. required
password	boolean	true when using password. required
facial	boolean	true when using facial. required
facialAutoMatch	boolean	true when using face automatch. required
fingerprints	string	base64 string of fingerprint template data
keyrecords	single or array of keyrecords object	
keytype	integer	key type id, required
keystring	string	key value, required
keyid	string	unique id for this key, required
keyenabled	boolean	true to enable this key, required
facialrecords	single or array of	

	facialrecords object	
FacialTemplateType	integer	face template type id, required
FacialTemplate	string	base64 encoded face template data, required
FacialPhoto	string	base64 encoded jpeg photo of this face, required
cardsn	string	smart card number
ScoreThreshold	integer	setup an individual score for facial
expirydate	xsd:datetime	set if have expiry date, 0 to disable.
message	string	user message

RESPONSE

Response Code	200

example	{
	"status": “success”,
	“message”:”User A0001 Added”
	}

Response data:

name	type	description
status	string	status response
message	string	return added userID

3.2 addUsers

Description: add a batch of users

REQUEST

same as addUser but in array form

Method	POST
Path	/cgi-bin/restapi/user/{sessionid}
Example URI	/cgi-bin/restapi/user/1805350299
Example JSON	[USER1, USER2…..]

Request Parameters :

*Refer to AddUser

RESPONSE

Response Code	200
example	[RESPONSE1, RESPONSE2,.....]

Response data:
*Refer to AddUser

3.3 getUsers

Description: Get User(s) by given criteria

REQUEST

Method	GET

Path	/cgi-bin/restapi/user/{sessionid}/?userID &partialUserID &lastName &firstName &otherName
Example URI	/cgi-bin/restapi/user/{sessionid}/?userID=A123&partialUserID=A123&lastName=John&firstName=Wade&otherName=Bee

Request Parameters in Query:

name	type	description
{sessionid}:	long	session id , required
userID	string	full employee ID
partialUserID	string	partial employee ID
lastName	string	full last name
firstName	string	full first name
otherName	string	full other name

RESPONSE

Response Code	200

example	[
	{
	"userID": "111",
	"lastName": "Li",
	"firstName": "Edwin",
	"adminLevel": "PERSONALUSER",
	"groupID": [0],
	"departmentID": [0],
	"status": {
	"active": true,
	"autoMatch": true,
	"smartCard": true,
	"fingerprint": true,
	"password": false,
	"facial": true,

```
      "facialAutoMatch": true
    },
    "facialrecords": [     {
      "FacialTemplateType": 2,
      "FacialTemplate": Base64 something="
    }],
    "cardsn": "AAAAAAAA",
    "ScoreThreshold": 0
  },
    {
    "userID": "222",
    "lastName": "Chiu",
    "firstName": "Sing",
    "adminLevel": "PERSONALUSER",
    "groupID": [0],
    "departmentID": [0],
    "status":       {
      "active": true,
      "autoMatch": true,
      "smartCard": false,
      "fingerprint": true,
      "password": false,
      "facial": true,
      "facialAutoMatch": true
    },
    "facialrecords": [     {
      "FacialTemplateType": 2,
      "FacialTemplate": "Base64 Something.."
    }],
    "ScoreThreshold": 0
  }
]
```


Response data: Array of event logs.

name	type	description
userID	string	Employee ID

lastName	string	last name
firstName	string	First name
otherName	string	other name
adminLevel	string	either ”PERSONALUSER”, ”NETWORKADMIN”, “USERADMIN”, “SUPERADMIN”
password	string	user defined password
groupID	array of integer	Access Group ID
departmentID	array of integer	Department ID
status	single object of status	
active	boolean	user status, false means user is suspended
autoMatch	boolean	fingerprint automatch status
smartCard	boolean	use smartcard status
fingerprint	boolean	use fingerprint status
password	boolean	use password status
facial	boolean	use facial status
facialAutoMatch	boolean	facial automatch status
fingerprints	string	base64 string of fingerprint template data
keyrecords	single or array of keyrecords object	
keytype	integer	key type id
keystring	string	key value
keyid	string	unique id for this key
keyenabled	boolean	key enabled status

facialrecords	single or array of facialrecords object	
FacialTemplateType	integer	face template type id
FacialTemplate	string	base64 encoded face template data
FacialPhoto	string	base64 encoded jpeg photo of this face
cardsn	string	smart card number
ScoreThreshold	integer	individual score for facial
expirydate	xsd:datetime	expiry date, 0 is disabled
message	string	user message string

3.4 updateUser

Description: update single user with user structure

REQUEST

Method	PUT
Path	/cgi-bin/restapi/user/{sessionid}
Example URI	/cgi-bin/restapi/user/1805350299

Example JSON	{
	"userID": "A0001",
	"lastName": "John",
	"firstName": "Ben",
	"otherName": "Teddy",
	"adminLevel": "PERSONALUSER",
	"password": "123555",
	"groupID": 0,
	"departmentID": 1,
	"status": {
	"active": true,

<table>
<tr><td></td><td><pre>
    "autoMatch": true,
    "smartCard": true,
    "fingerprint": true,
    "password": true,
    "facial": true,
    "facialAutoMatch": false
  },
  "fingerprints": "Base64 Enoded",
  "keyrecords":   {
    "keytype": 1,
    "keystring": "According to keytype",
    "keyid": "unique id for this key",
    "keyenabled": true
  },
  "facialrecords":   {
    "FacialTemplateType": 1,
    "FacialTemplate": "xxx",
    "FacialPhoto": "base64 encode"
  },
  "cardsn": "00003",
  "ScoreThreshold": 0,
  "expirydate": 0,
  "message": "user message"
}
</pre></td></tr>
</table>

Request Parameters:

name	type	description
{sessionid}	long	session id, required
userID	string	User ID, required according to the User ID restriction.
lastName	string	last name
firstName	string	First name
otherName	string	other name
adminLevel	string	either ”PERSONALUSER”, ”NETWORKADMIN”, “USERADMIN”,

		“SUPERADMIN” required if other words, default to “PERSONALUSER”
password	string	user defined password
groupID	integer or array of integer	id(s) of Group, if not present add a default 0 id
departmentID	integer or array of integer	id(s) of Department, if not present add a default 0 id
status	single object of status	
active	boolean	if active is false, this user is suspended.required
autoMatch	boolean	true when using fingerprint automatch. required
smartCard	boolean	true when using smartcard. required
fingerprint	boolean	true when using fingerprint. required
password	boolean	true when using password. required
facial	boolean	true when using facial. required
facialAutoMatch	boolean	true when using face automatch. required
fingerprints	string	base64 string of fingerprint template data
keyrecords	single or array of keyrecords object	
keytype	integer	key type id, required
keystring	string	key value, required
keyid	string	unique id for this key, required
keyenabled	boolean	true to enable this key,

		required
facialrecords	single or array of facialrecords object	
FacialTemplateType	integer	face template type id, required
FacialTemplate	string	base64 encoded face template data, required
FacialPhoto	string	base64 encoded jpeg photo of this face, required
cardsn	string	smart card number
ScoreThreshold	integer	setup an individual score for facial
expirydate	xsd:datetime	set if have expiry date, 0 to disable.
message	string	user message

RESPONSE

Response Code	200

example	{
	"status": “success”,
	“message”:”User A0001 Updated”
	}

Response data:

name	type	description
status	string	status response
message	string	return updated userID

3.5 updateUsers

Description: update a batch of users

REQUEST

same as addUser but in array form

Method	PUT
Path	/cgi-bin/restapi/user/{sessionid}
Example URI	/cgi-bin/restapi/user/1805350299
Example JSON	[USER1, USER2…..]

Request Parameters :

*Refer to UpdateUser

RESPONSE

Response Code	200
example	[RESPONSE1, RESPONSE2,.....]

Response data:
*Refer to UpdateUser

3.6 deleteUser

Description: delete single user

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/user/{sessionid}/{userID}

Example URI	/cgi-bin/restapi/deleteLogs/1805350299/A123

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{userID}	string	employee ID, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "User A123 Deleted"
	}

Response data:

name	type	description
status	string	status response
message	string	show deleted employee ID

3.7 activateUser

Description: set user status active

REQUEST

Method	POST
Path	/cgi-bin/restapi/activateUser/{sessionid}/{userID}

Example URI	/cgi-bin/restapi/activateUser/1805350299/A123

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{userID}	string	employee ID, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	}

Response data:

name	type	description
status	string	status response

3.8 deactivateUser

Description: set user status inactive

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/activateUser/{sessionid}/{userID}
Example URI	/cgi-bin/restapi/activateUser/1805350299/A123

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{userID}	string	employee ID, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	}

Response data:

name	type	description
status	string	status response

3.9 get UserMessage

Description: get User Message

REQUEST

Method	GET
Path	/cgi-bin/restapi/UserMessage/{sessionid}/{userID}
Example URI	/cgi-bin/restapi/UserMessage/1805350299/A123

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

{userID}	string	employee ID, required

RESPONSE

Response Code	200

example	{
	"usermessage": "this is test",
	}

Response data:

name	type	description
usermessage	string	user message content

3.10 Set UserMessage

Description: set user message

REQUEST

Method	POST
Path	/cgi-bin/restapi/UserMessage/{sessionid}/{userID}
Example URI	/cgi-bin/restapi/UserMessage/1805350299/A123

Example JSON	{
	"usermessage": "ABC"
	}

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

{userID}	string	employee ID, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	}

Response data:

name	type	description
status	string	status response

3.11 delete UserMessage

Description: delete user message

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/UserMessage/{sessionid}/{userID}
Example URI	/cgi-bin/restapi/UserMessage/1805350299/A123

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{userID}	string	employee ID, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	}

Response data:

name	type	description
status	string	status response

3.12 get UserCount

Description: get total number of Users

REQUEST

Method	GET
Path	/cgi-bin/restapi/UserCount/{sessionid}
Example URI	/cgi-bin/restapi/UserMessage/1805350299

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	{ "usercount": 16

	}

Response data:

name	type	description
usercount	integer	number of users

3.13. get UserBatch

Description: get total number of Users

REQUEST

Method	GET
Path	/cgi-bin/restapi/UserBatch/{sessionid}/{offset}/{limit}
Example URI	/cgi-bin/restapi/UserMessage/1805350299/0/10

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{offset}	integer	offset of batch users, required
{limit}	integer	limit of batch users, required

RESPONSE

Response Code	200
example	[USER1, USER2…..]

*USER1, USER2 is user json structure , please refer to getUsers
Response data:

name	type	description
USER*	ns:User*	User json structure*

*please refer to getUsers

3.14. get User API TOTP

Description: get API TOTP code for login(Time-based One-Time Password)

REQUEST

Method	GET
Path	/cgi-bin/restapi/UserAPITOTP/{sessionid}/{userID}/{TOTPType}
Example URI	/cgi-bin/restapi/UserMessage/1805350299/A33322/QR_PNG

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{userID}	string	user ID, required
{TOTPType}	string	5 types allowed required “QR_PNG”, “QR_SVG”, “QR_ANSI”, “QR_PLAINTEXT”, “OTP_PASSCODE” any wrong pattern default as “PNG”

RESPONSE

Response Code	200
example	{ "format": "QR_PNG",

	"expiry": "2025-08-20T04:14:40Z", "data": Base64 encoded PNG data }

*USER1, USER2 is user json structure , please refer to getUsers
Response data:

name	type	description
format	string	5 types of {TOTPType}
expiry	xsd:datetime	UTC time of expiry of TOTP
data	complex	when format QR_PNG, QR_SVG, QR_ANSI: it is base64 encoded data when QR_PLAINTEXT: it is the raw text of the QRcode contain(includeing a header and a json content) when OTP_PASSCODE: it is 8 digits TOTP passcode.

4. Department Management

4.1 addDepartment

Description: add new department

REQUEST

Method	POST
Path	/cgi-bin/restapi/department/{sessionid}
Example URI	/cgi-bin/restapi/department/1805350299
Example JSON	{ "id": 21, "name": "ABC",

	"description": "LONG" } or in array form: [JSON1, JSON2,...]

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of department, required, need handle by user
name	string	Department Name, required
description	string	Department Description, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "department ID 21 Added"
	}
	or in array

Response data:

name	type	description
status	string	status response
message	string	details description

4.2 updateDepartment

Description: update exist department

REQUEST

Method	PUT
Path	/cgi-bin/restapi/department/{sessionid}
Example URI	/cgi-bin/restapi/department/1805350299

Example JSON	{
	"id": 21,
	"name": "ABC",
	"description": "LONG"
	}
	or in array form:
	[JSON1, JSON2,...]

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of department, required
name	string	Department Name
description	string	Department Description

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "department ID 21 Updated"
	}
	or in array.

Response data:

name	type	description

status	string	status response
message	string	details description

4.3 getDepartments

Description: get exist department, if no parameters provided, all department will be listed.

REQUEST

Method	GET
Path	/cgi-bin/restapi/department/{sessionid}?id &name &description
Example URI	/cgi-bin/restapi/department/1805350299?id=0 &name=General&description=General

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of department
name	string	Department Name
description	string	Department Description

RESPONSE

Response Code	200

example	{
	"id": 21,
	"name": "ABC",
	"description": "LONG"
	}
	
	or in array.

Response data:

name	type	description
id	integer	ID of department
name	string	Department Name
description	string	Department Description

4.4 deleteDepartment

Description: delete department, at least 1 parameters provided

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/department/{sessionid}?id &name &description
Example URI	/cgi-bin/restapi/department/1805350299?id=0 &name=General&description=General

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of department
name	string	Department Name
description	string	Department Description

RESPONSE

Response Code	200
example	{ "status": "success",

	"message": "department ID 21 Deleted" } or in array.

Response data:

name	type	description
status	string	status response
message	string	details description

5. Access Right Management

5.1 add AccessRight

Description: add new Access Right

REQUEST

Method	POST
Path	/cgi-bin/restapi/accessright/{sessionid}
Example URI	/cgi-bin/restapi/accessright/1805350299
Example JSON	{ "id": 14, "quickaccess": true, "terminalSN": "00111D000000", "name": "officer", "description": “night shift", "timespec": [{"weekday": "SUNDAY", "timePattern":"11111111111111111111111111111111111111111111111"},

	{"weekday": "MONDAY", "timePattern":"1111111111111111111111111111111111111111111111111" }, {"weekday": "TUESDAY", "timePattern":"1111111111111111111111111111111111111111111111111"}, {"weekday": "WEDNESDAY", "timePattern":"1111111111111111111111111111111111111111111111111"}, {"weekday": "THURSDAY", "timePattern":"1111111111111111111111111111111111111111111111111"}, {"weekday": "FRIDAY", "timePattern":"1111111111111111111111111111111111111111111111111" }, {"weekday": "SATURDAY", "timePattern":"1111111111111111111111111111111111111111111111111" }, {"weekday": "HOLIDAY", "timePattern":"1111111111111111111111111111111111111111111111111" }] } or in array form: [JSON1, JSON2,...]

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of access right, if not present will auto create
quickaccess	boolean	enable quick access function
terminalSN	string	terminal SN for this access right.
name	string	name of access right
description	string	detail of access right

timespec		only need if the weekday have any slot enabled
weekday	string	fixed string from 8 pattern SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, HOLIDAY no default value
timePattern	special format*	48-byte strings of '0'(off) and/or '1'(on)) Remark: First byte representing timeslot [00:00-00:29], 2nd byte [00:30-00:59] etc.

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "access right ID 14 Added"
	}
	or in array

Response data:

name	type	description
status	string	status response
message	string	details description

5.2 get AccessRight

Description: get exist access right, if no parameters provided, all access righ will be listed.

REQUEST

Method	GET
Path	/cgi-bin/restapi/accessright/{sessionid}?id
Example URI	/cgi-bin/restapi/accessright/1805350299?id=14

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of access right if not provided, all access right will get.

RESPONSE

Response Code	200
example	{ "id": 14, "quickaccess": true, "terminalSN": "00111D000000", "name": "officer", "description": “night shift", "timespec": [{"weekday": "SUNDAY", "timePattern":"11111111111111111111111111111111111111111111111111111111"}, {"weekday": "MONDAY", "timePattern":"11111111111111111111111111111111111111111111111111111111" }, {"weekday": "TUESDAY", "timePattern":"11111111111111111111111111111111111111111111111111111111"}, {"weekday": "WEDNESDAY", "timePattern":"11111111111111111111111111111111111111111111111111111111"}, {"weekday": "THURSDAY", "timePattern":"11111111111111111111111111111111111111111111111111111111"},

	{"weekday": "FRIDAY", "timePattern":"11111111111111111111111111111111111111111111111" }, {"weekday": "SATURDAY", "timePattern":"11111111111111111111111111111111111111111111111" }, {"weekday": "HOLIDAY", "timePattern":"11111111111111111111111111111111111111111111111" }] } or in array form: [JSON1, JSON2,...]

Response data:

name	type	description
id	integer	ID of access right, if not present will auto create
quickaccess	boolean	enable quick access function
terminalSN	string	terminal SN for this access right.
name	string	name of access right
description	string	detail of access right
timespec		only show if the weekday have any slot enabled
weekday	string	fixed string from 8 pattern SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, HOLIDAY no default value
timePattern	special format*	48-byte strings of '0'(off) and/or '1'(on)) Remark: First byte

		representing timeslot [00:00-00:29], 2nd byte [00:30-00:59] etc.

5.3 update AccessRight

Description: same as add access right but the accessright id must be provided

REQUEST

Method	PUT
Path	/cgi-bin/restapi/accessright/{sessionid}
Example URI	/cgi-bin/restapi/accessright/1805350299
Example JSON	{ "id": 14, "quickaccess": true, "terminalSN": "00111D000000", "name": "officer", "description": “night shift", "timespec": [{"weekday": "SUNDAY", "timePattern":"11111111111111111111111111111111111111111111111"}, {"weekday": "MONDAY", "timePattern":"11111111111111111111111111111111111111111111111" }, {"weekday": "TUESDAY", "timePattern":"11111111111111111111111111111111111111111111111"}, {"weekday": "WEDNESDAY", "timePattern":"11111111111111111111111111111111111111111111111"}, {"weekday": "THURSDAY", "timePattern":"11111111111111111111111111111111111111111111111"}, {"weekday": "FRIDAY", "timePattern":"11111111111111111111111111111111111111111111111" },

	{"weekday": "SATURDAY", "timePattern":"11111111111111111111111111111111111111111111111" }, {"weekday": "HOLIDAY", "timePattern":"11111111111111111111111111111111111111111111111" }] } or in array form: [JSON1, JSON2,...]

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of access right, required
quickaccess	boolean	enable quick access function
terminalSN	string	terminal SN for this access right.
name	string	name of access right
description	string	detail of access right
timespec		only need if the weekday have any slot enabled
weekday	string	fixed string from 8 pattern SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, HOLIDAY no default value
timePattern	special format*	48-byte strings of '0'(off) and/or '1'(on)) Remark: First byte representing timeslot [00:00-00:29], 2nd byte [00:30-00:59] etc.

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "access right ID 14 Updated"
	}
	or in array

Response data:

name	type	description
status	string	status response
message	string	details description

5.4 delete AccessRight

Description: delete access right

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/accessright/{sessionid}/{id}
Example URI	/cgi-bin/restapi/accessright/1805350299/14

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{id}	integer	ID of department, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "access right ID 14 Deleted"
	}
	or in array.

Response data:

name	type	description
status	string	status response
message	string	details description

6. Access Group Management

6.1 add Access Group

Description: add new Access Group

REQUEST

Method	POST
Path	/cgi-bin/restapi/accessgroup/{sessionid}
Example URI	/cgi-bin/restapi/accessgroup/1805350299

Example JSON	{
	"id": 14,
	"name": "new manager3",
	"departmentid": 1
	“accessrightid”: [1, 2]
	}

	or in array form: [JSON1, JSON2,...]

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of access group, if not present will auto create
name	string	name of access group
departmentid	integer	associated department ID, required
accessrightid	integer or array of integer	associated access right id(s)

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "access group ID 14 Added"
	}
	or in array

Response data:

name	type	description
status	string	status response
message	string	details description

6.2 get Access Groups

Description: get exist access right, if no parameters provided, all access righ will be listed.

REQUEST

Method	GET
Path	/cgi-bin/restapi/accessgroup/{sessionid}?groupid &deptid
Example URI	/cgi-bin/restapi/accessgroup/1805350299groupid=0&deptid=0

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
groupid	integer	ID of access group if not provided, all access groups will get.
deptid	integer	associated department ID

RESPONSE

Response Code	200

example	[
	{
	"id": 0,
	"name": "General Staff",
	"departmentid": "0",
	"accessrightid": [0]
	}
	]
	
	or in array form:
	[JSON1, JSON2,...]

Response data:

name	type	description
id	integer	ID of access group
name	string	name of access group
departmentid	integer	associated department ID, required
accessrightid	integer or array of integer	associated access right id(s)

6.3 update Access Group

Description: update Access Right

REQUEST

Method	PUT
Path	/cgi-bin/restapi/accessgroup/{sessionid}
Example URI	/cgi-bin/restapi/accessgroup/1805350299

Example JSON	{
	"id": 14,
	"name": "new manager3",
	"departmentid": 1
	“accessrightid”: [1, 2]
	}
	
	or in array form:
	[JSON1, JSON2,...]

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
id	integer	ID of access group, required

name	string	name of access group
departmentid	integer	associated department ID, required
accessrightid	integer or array of integer	associated access right id(s)

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "access group ID 14 Added"
	}
	or in array

Response data:

name	type	description
status	string	status response
message	string	details description

6.4 delete Access Group

Description: delete access group

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/accessgroup/{sessionid}/{id}
Example URI	/cgi-bin/restapi/accessgroup/1805350299/14

Request Parameters in Query:

name	type	description

{sessionid}	long	session id, required
{id}	integer	ID of access group, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "access group ID 14 Deleted"
	}
	or in array.

Response data:

name	type	description
status	string	status response
message	string	details description

7. Trigger Maintenance Functions

7.1 getTriggers

Description: Get Trigger with the trigger schedule

REQUEST

Method	GET
Path	/cgi-bin/restapi/trigger/{sessionid}?type
Example URI	/cgi-bin/restapi/trigger/1805350299?type=IN

Request Parameters in Query:

name	type	description

{sessionid}	long	session id, required
type	string	Trigger type name: IN, OUT, F1,F2…., F40

RESPONSE

Response Code	200

example	[{
	"type": "IN",
	"name": "IN",
	"enable": true,
	“freeJobcode”:false
	"timespec": [{
	"weekday": "SUNDAY",
	"timePattern":
	"1000000000000000000000000000000000000000000000000"
	}]
	
	}]
	
	or in array form:
	[JSON1, JSON2,...]

Response data:

name	type	description
type	string	Trigger type name: IN, OUT, F1,F2…., F40
name	string	name of Trigger
enable	boolean	enable status
freeJobcode	boolean	only F1-F10
timespec		only need if having a schedule

weekday	string	fixed string from 8 pattern SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, HOLIDAY no default value
timePattern	special format*	48-byte strings of '0'(off) and/or '1'(on)) Remark: First byte representing timeslot [00:00-00:29], 2nd byte [00:30-00:59] etc.

7.2 updateTrigger

Description: UpdateTrigger with the trigger schedule

REQUEST

Method	PUT
Path	/cgi-bin/restapi/trigger/{sessionid}
Example URI	/cgi-bin/restapi/trigger/1805350299

Example JSON	{
	"type": "F32",
	"name": "F32",
	"enable": true,
	"timespec": [{
	"weekday": "SUNDAY",
	"timePattern":
	"100000000000000000000000000000000000000000000000"
	}]
	}

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

type	string	Trigger type name: IN, OUT, F1,F2…., F40, required
name	string	name of Trigger
enable	boolean	enable status
freeJobcode	boolean	only F1-F10
timespec		only need if having a schedule
weekday	string	fixed string from 8 pattern SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, HOLIDAY no default value
timePattern	special format*	48-byte strings of '0'(off) and/or '1'(on)) Remark: First byte representing timeslot [00:00-00:29], 2nd byte [00:30-00:59] etc.

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "trigger type F32 schedule Updated"
	}

Response data:

name	type	description

status	string	status response
message	string	details description

7.3 clearTrigger

Description: Clear trigger schedule of trigger

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/trigger/{sessionid}/{type}
Example URI	/cgi-bin/restapi/trigger/1805350299/F40

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
type	string	Trigger type name: IN, OUT, F1,F2…., F40, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "trigger type F40 schedule cleared"
	}

Response data:

name	type	description

status	string	status response
message	string	details description

8. Personal Functions

8.1 getMyself

Description: Get current login user

REQUEST

Method	GET
Path	/cgi-bin/restapi/myself/{sessionid}/
Example URI	/cgi-bin/restapi/user/{sessionid}/

Request Parameters in Query:

name	type	description
{sessionid}:	long	session id , required

RESPONSE

Response Code	200

example	{
	"userID": "111",
	"lastName": "Li",
	"firstName": "Edwin",
	"adminLevel": "PERSONALUSER",
	"groupID": [0],
	"departmentID": [0],
	"status": {
	"active": true,
	"autoMatch": true,
	"smartCard": true,

	"fingerprint": true,
	"password": false,
	"facial": true,
	"facialAutoMatch": true
	},
	"facialrecords": [{
	"FacialTemplateType": 2,
	"FacialTemplate": Base64 something="
	}],
	"cardsn": "AAAAAAAA",
	"ScoreThreshold": 0
	}

Response data: Array of event logs.

name	type	description
userID	string	Employee ID
lastName	string	last name
firstName	string	First name
otherName	string	other name
adminLevel	string	either ”PERSONALUSER”, ”NETWORKADMIN”, “USERADMIN”, “SUPERADMIN”
password	string	user defined password
groupID	array of integer	Access Group ID
departmentID	array of integer	Department ID
status	single object of status	
active	boolean	user status, false means user is suspended

autoMatch	boolean	fingerprint automatch status
smartCard	boolean	use smartcard status
fingerprint	boolean	use fingerprint status
password	boolean	use password status
facial	boolean	use facial status
facialAutoMatch	boolean	facial automatch status
fingerprints	string	base64 string of fingerprint template data
keyrecords	single or array of keyrecords object	
keytype	integer	key type id
keystring	string	key value
keyid	string	unique id for this key
keyenabled	boolean	key enabled status
facialrecords	single or array of facialrecords object	
FacialTemplateType	integer	face template type id
FacialTemplate	string	base64 encoded face template data
FacialPhoto	string	base64 encoded jpeg photo of this face
cardsn	string	smart card number
ScoreThreshold	integer	individual score for facial
expirydate	xsd:datetime	expiry date, 0 is disabled
message	string	user message string

8.2 updateMyself

Description: update current login user, only can update first name, last name, other name, card sn and password.

REQUEST

Method	PUT
Path	/cgi-bin/restapi/myself/{sessionid}
Example URI	/cgi-bin/restapi/myself/1805350299

Example JSON	{
	"userID": "A0001",
	"lastName": "John",
	"firstName": "Ben",
	"otherName": "Teddy",
	"password": "123555",
	"cardsn": "00003",
	}

Request Parameters:

name	type	description
{sessionid}	long	session id, required
userID	string	User ID, required must be the same as current login user
lastName	string	last name
firstName	string	First name
otherName	string	other name
password	string	user defined password
cardsn	string	smart card number

RESPONSE

Response Code	200

example	{
	"status": “success”,
	“message”:”User A0001 Updated”
	}

Response data:

name	type	description
status	string	status response
message	string	return updated userID

9. Agent

9.1 register Agent

Description: Register new Agent

REQUEST

Method	POST
Path	/cgi-bin/restapi/agent/{sessionid}
Example URI	/cgi-bin/restapi/agent/1805350299

Example JSON	{
	"url": "https://anc.cde.com",
	"agentVersion": "1.2",
	"magic": "1",
	"initSync": false
	}

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
url	string	URL of Agent endpoint,required
agentVersion	string	Agent version control, if not present, “1.2” will be used
magic	string	agent magic string, according to agent setting, if not present, “” empty string will be used
initSync	boolean	if true, all logs will be sent to agent after register.

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "agent registered"
	}

Response data:

name	type	description
status	string	status response
message	string	details description

9.2 unregister Agent

Description: Unregister Agent

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/agent/{sessionid}?url
Example URI	/cgi-bin/restapi/agent/1805350299?url=https://anc.cde.com

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
url	string	URL of Agent endpoint,required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "Unregistration Agent Success"
	}

Response data:

name	type	description
status	string	status response
message	string	details description

9.3 get Registered Agents

Description: Get all registered Agent

REQUEST

Method	GET

Path	/cgi-bin/restapi/agent/{sessionid}
Example URI	/cgi-bin/restapi/agent/1805350299

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200

example	[
	{
	"url":
	"http://abc.def.com:8080/AccessServer/AgentService.asmx",
	"registrationTime":
	"2025-08-29T07:11:04Z",
	"lastUpdate": "1970-01-01T00:00:00Z",
	"lastCheck": "1970-01-01T00:00:00Z",
	"lastCheckStatus": false,
	"active": true
	},
	{
	"url": "https://anc.cde.com",
	"registrationTime":
	"2025-08-29T09:24:21Z",
	"lastUpdate": "1970-01-01T00:00:00Z",
	"lastCheck": "1970-01-01T00:00:00Z",
	"lastCheckStatus": false,
	"active": true
	}
	]

Response data:

name	type	description

url	string	URL of Agent endpoint
registrationTime	xsd:datetime	time of the registeration
lastUpdate	xsd:datetime	last update time
lastCheck	xsd:datetime	last check connection time
lastCheckStatus	boolean	last check is ok/fail
active	boolean	active flag

10. Time Settings

10.1 get NTP Server setting

Description: get NTP Setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/ntp/{sessionid}
Example URI	/cgi-bin/restapi/ntp/1805350299

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	{ "serveraddr": "0.arch.pool.ntp.org 1.arch.pool.ntp.org", "enable": true

	}

Response data:

name	type	description
serveraddr	string	NTP server (more than 1 server can separate them with space
enable	boolean	NTP enable status

10.2 set NTP Server setting

Description: update NTP setting

REQUEST

Method	PUT
Path	/cgi-bin/restapi/ntp/{sessionid}
Example URI	/cgi-bin/restapi/ntp/1805350299

Example JSON	{
	"serveraddr": "0.arch.pool.ntp.org",
	"enable": true
	}

Request Parameters:

name	type	description
{sessionid}	long	session id, required
serveraddr	string	NTP server (more than 1 server can separate them with space
enable	boolean	NTP enable status

RESPONSE

Response Code	200

example	{
	"status": “success”,
	“message”:”NTP Server setting updated”
	}

Response data:

name	type	description
status	string	status response
message	string	return message

10.3 get Terminal DateTime Info

Description: get Terminal Datetime information

REQUEST

Method	GET
Path	/cgi-bin/restapi/time/{sessionid}
Example URI	/cgi-bin/restapi/time/1805350299

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200

example	{
	"currentTime": "2025-09-02T16:29:14",
	"timezone": "UTC"
	}

Response data:

name	type	description
currentTime	xsd:datetime	current datetime, must be without 'Z'
timezone	string	timezone: according to "TZ identifier" from https://en.wikipedia.org/wiki/List_of_tz_database_time_zones. if not found, please try the identifier in 'Notes'

10.4 set Terminal DateTime Info

Description: update terminal datetime info

REQUEST

Method	PUT
Path	/cgi-bin/restapi/ntp/{sessionid}
Example URI	/cgi-bin/restapi/ntp/1805350299

Example JSON	{
	"currentTime": "2025-09-02T16:29:14",
	"timezone": "UTC"
	}

Request Parameters:

name	type	description

{sessionid}	long	session id, required
currentTime	xsd:datetime	datetime to be setup ,if datatime contain ‘Z’, it is UTC time, e.g. 2025-07-16T10:24:13Z if without ‘Z’, it is local time, e.g. 2025-07-16T10:24:13
timezone	string	timezone: according to “TZ identifier” from https://en.wikipedia.org/wiki/List_of_tz_database_time_zones. if not found, please try the identifier in ‘Notes’

RESPONSE

Response Code	200

example	{
	"status": “success”,
	“message”:”Time setting updated”
	}

Response data:

name	type	description
status	string	status response
message	string	return message

11. Terminal Functions

11.1 open Door

Description: open door stike

REQUEST

Method	POST
Path	/cgi-bin/restapi/opendoor/{sessionid}
Example URI	/cgi-bin/restapi/opendoor/1805350299

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "Main door opened"
	}

Response data:

name	type	description
status	string	status response
message	string	details description

11.2 get APIVersion

Description: get API version

REQUEST

Method	GET
Path	/cgi-bin/restapi/apiversion
Example URI	/cgi-bin/restapi/apiversion

RESPONSE

Response Code	200

example	{
	"name": "A4 REST API",
	"version": "0.1"
	}

Response data:

name	type	description
name	string	API name
version	string	API version string

11.3 get Terminal Status

Description: Get Terminal Status

REQUEST

Method	GET
Path	/cgi-bin/restapi/terminalstatus/{sessionid}
Example URI	/cgi-bin/restapi/terminalstatus/1805350299

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	{

	"serialNumber": "00111D000000", "FAMVersion": "NO FAM", "description": "ACTAtek", "registeredUsers": 16, "maximumUsers": 1000, "automatchUsers": 12, "maximumAutomatchUsers": 0, "currentStatus": "Online", "firmwareVersion": "jakinid_trunk" }

Response data:

name	type	description
serialNumber	string	serial number
FAMVersion	string	Fingerprint version
description	string	terminal description
registeredUsers	integer	registered user
maximumUsers	integer	maximum user support
automatchUsers	integer	fingerprint automatch user count
maximumAutomatchUsers	integer	maximum fingerprint automatch support
currentStatus	string	terminal status
firmwareVersion	string	terminal firmware version

11.4 Relay action

Description: control relay state

REQUEST

Method	POST
Path	/cgi-bin/restapi/relay/{sessionid}/{relayaction}/{relayid}
Example URI	/cgi-bin/restapi/relay/1805350299/open/ALL

Request Parameters in Query:

name	type	description
{sessionid}	long	session id, required
{relayaction}	string	relay action, required: 'open' - open the relay 'close' - close the relay
{relayid}	string	ID of relay, required: 'ONE' - Door strike 1 'TWO' - Door strike 2 'ALL' - Both door strike

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "Relay ALL opened"
	}

Response data:

name	type	description
status	string	status response
message	string	details description

11.5 set Relay Setting

Description: set the relay setting

REQUEST

Method	PUT
Path	/cgi-bin/restapi/relaysetting/{sessionid}/
Example URI	/cgi-bin/restapi/relaysetting/1805350299

Example JSON	{
	"id": "ONE",
	"option": "disable",
	"delay": 11
	}

Request Parameters:

name	type	description
{sessionid}	long	session id, required
id	string	ID of relay, required: 'ONE' - Door strike 1 'TWO' - Door strike 2
option	string	either 'disable' or 'access_denied' for Door strike 1 'disable', 'access_granted', 'access_denied', 'doorbell' or 'bell_schedule' for Door strike 2
delay	integer	in second Delay must between 1 - 1800 seconds when the door open schedule is enabled. 1 - 20 seconds when the door open schedule is disabled

RESPONSE

Response Code	200
example	{ "status": "success",

	"message": "relay setting ONE updated" }

Response data:

name	type	description
status	string	status response
message	string	return message

11.6 get Relay Setting

Description: get the relay setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/relaysetting/{sessionid}/{relayid}
Example URI	/cgi-bin/restapi/relaysetting/1805350299/ONE

Request Parameters:

name	type	description
{sessionid}	long	session id, required
{relayid}	string	‘ONE’ - Door strike 1 ‘TWO’ - Door strike 2 ‘ALL’ - Both door strikes

RESPONSE

Response Code	200

example	{
	"id": "ONE",
	"option": "disable",
	"delay": 11
	}

Response data:

name	type	description
id	string	ID of relay, required: ‘ONE’ - Door strike 1 ‘TWO’ - Door strike 2
option	string	either ‘disable’ or ‘access_denied’ for Door strike 1 ‘disable’, ‘access_granted’, ‘access_denied’, ‘doorbell’ or ‘bell_schedule’ for Door strike 2
delay	integer	in second Delay must between 1 - 1800 seconds when the door open schedule is enabled. 1 - 20 seconds when the door open schedule is disabled

11.7 setAutoInOut

Description: set AutoInOut setting

REQUEST

Method	PUT
Path	/cgi-bin/restapi/autoInOut/{sessionid}
Example URI	/cgi-bin/restapi/autoInOut/1805350299
Example JSON	{ "mode": "disable", "auto_reset": true, "rejectRepeat_time_sec": 8, "lockout_time_min": 20,

	"crowd_limit_count": 100, "crowd_reset_hhmm": "10:11" }

Request Parameters:

name	type	description
{sessionid}	long	session id, required
mode	string	mode string: 'disable'- Disable 'enable'/'autoInOut' - AutoInOut 'reject_repeat_login' - Reject repeat 'anti_passback' - Anti-passback 'lunch_break' - Lunch break 'crowd_control' - Crowd Control
auto_reset	boolean	in AutoInOut mode, Auto Reset IN/OUT
rejectRepeat_time_sec	integer	in term of second from 1 to 86400
lockout_time_min	integer	in term of minute from 1 to 120
crowd_limit_count	integer	from 1 to 65535
crowd_reset_hhmm	string	in HH:MM format e.g: 12:30

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "autoInOut option updated"
	}

Response data:

name	type	description
status	string	status response

message	string	return message

11.8 getAutoInOut

Description: get AutoInOut setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/autoInOut/{sessionid}/
Example URI	/cgi-bin/restapi/autoInOut/1805350299

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200

example	{
	"mode": "disable",
	"auto_reset": true,
	"rejectRepeat_time_sec": 8,
	"lockout_time_min": 20,
	"crowd_limit_count": 100,
	"crowd_reset_hhmm": "10:11"
	}

Response data:

name	type	description
mode	string	mode string: 'disable'- Disable 'enable'/'autoInOut' - AutoInOut 'reject_repeat_login' - Reject

		repeat 'anti_passback' - Anti-passback 'lunch_break' - Lunch break 'crowd_control' - Crowd Control
auto_reset	boolean	in AutoInOut mode, Auto Reset IN/OUT
rejectRepeat_time_sec	integer	in term of second from 1 to 86400
lockout_time_min	integer	in term of minute from 1 to 120
crowd_limit_count	integer	from 1 to 65535
crowd_reset_hhmm	string	in HH:MM format e.g: 12:30

11.9 set LogUnauthorizedEvent

Description: set LogUnauthorizedEvent

REQUEST

Method	PUT
Path	/cgi-bin/restapi/logUnauthorizedEvent/{sessionid}/{enable}
Example URI	/cgi-bin/restapi/logUnauthorizedEvent/1805350299/true

Request Parameters:

name	type	description
{sessionid}	long	session id, required
enable	string	'enable'/'1'/'true' – enable 'disable'/'0'/'false' – disable

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "logUnauthorizedEvent set false"
	}

Response data:

name	type	description
status	string	status response
message	string	return message

11.10 getLogUnauthorizedEvent

Description: get AutoInOut setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/logUnauthorizedEvent/{sessionid}/
Example URI	/cgi-bin/restapi/logUnauthorizedEvent/1805350299

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200

example	{ "enable": false }

Response data:

name	type	description
enable	boolean	true - enable false - disable

11.11 getCaptureFingerprint

Description: get captured fingerprint picture in jpeg format

REQUEST

Method	GET
Path	/cgi-bin/restapi/`getCaptureFingerprint` /{sessionid}/
Example URI	/cgi-bin/restapi/`getCaptureFingerprint` /1805350299

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	{"jpegPhoto": “xxxxxxx” }

Response data:

name	type	description

jpegPhoto	string	base64 encoded string

11.12 set RTP Setting

Description: set the RTP setting

REQUEST

Method	PUT
Path	/cgi-bin/restapi/rtpsetting/{sessionid}/
Example URI	/cgi-bin/restapi/rtpsetting/1805350299

Example JSON	{
	"mode": "disable",
	"key": "123",
	"target": "abc",
	"fps": 10,
	“stun”:”stun:stun1.l.google.com:3478”
	}

Request Parameters:

name	type	description
{sessionid}	long	session id, required
mode	string	mode of RTP: ‘disable’, ‘RTP’ or ‘SRTP’
key	string	key for SRTP
target	string	RTP destination URL
fps	integer	FPS for video, only accept 5,10,15,20,25,30
stun	string	STUN server setting: e.g. stun:stun1.l.google.com:3478

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "rtp setting updated, wait few second to get latest status"
	}

Response data:

name	type	description
status	string	status response
message	string	return message

11.13 get RTP Setting

Description: get the RTP setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/rtpsetting/{sessionid}/{relayid}
Example URI	/cgi-bin/restapi/rtpsetting/1805350299/ONE

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	{ "mode": "disable", "key": "123", "target": "abc", "fps": 10,

	“stun”:”stun:stun1.l.google.com:3478”, "sdpfile": "https://192.168.111.123/A4.sdp", "status": "Terminated" }

Response data:

name	type	description
mode	string	mode of RTP: ‘disable’, ‘RTP’ or ‘SRTP’
key	string	key for SRTP
target	string	RTP destination URL
fps	integer	FPS for video, only accept 5,10,15,20,25,30
stun	string	stun setting: e.g. stun:stun1.l.google.com:3478
sdpfile	string	path to download SDP file
status	string	working status, either ‘Terminated’ or ‘Running’

11.14 set Consent message Setting

Description: set the Consent message setting

REQUEST

Method	PUT
Path	/cgi-bin/restapi/consent/{sessionid}/{enable}/{display_name}
Example URI	/cgi-bin/restapi/consent/1805350299/enable/JakinID

Request Parameters:

name	type	description

{sessionid}	long	session id, required
{enable}	string	mode of consent message: ‘disable’, ‘enable’
{display_name}	string	name of display in consent message

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "Consent is enble"
	}

Response data:

name	type	description
status	string	status response
message	string	return message

11.15 get Consent message Setting

Description: get the Consent message setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/consent/{sessionid}
Example URI	/cgi-bin/restapi/consent/1805350299

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200

example	{
	"status": "succes"
	“enable”:true,
	"display_name": "oh my"
	}

Response data:

name	type	description
status	string	working status, either ‘Terminated’ or ‘Running’
enable	boolean	true : enable false: disable
display_name	string	

11.16 set Log Setting

Description: set the Log setting

REQUEST

Method	PUT
Path	/cgi-bin/restapi/logsetting/{sessionid}/
Example URI	/cgi-bin/restapi/logsetting/1805350299
Example JSON	{ "log_event": "user_audit_event", "log_size": 75000, "log_unAuth_event": true, "log_Accept_nonRegistered_card": true,

	"log_photo_Auth_event": true, "log_photo_unAuth_event": true, "log_Accept_nonRegistered_facial": true, "log_case_event": true, “log_EM_Option”:”FullSN” }

Request Parameters:

name	type	description
{sessionid}	long	session id, required
log_event	string	mode of log: ‘disable’, ‘user_event’, “audit_event” or ‘user_audit_event’
log_size	integer	log size:10000/75000/100000/500000/1000000
log_unAuth_event	boolean	Log Unauthorized Event
log_Accept_nonRegistered_card	boolean	Accept Unregistered Smartcard
log_photo_Auth_event	boolean	Photo Option for Log: Authorized Event
log_photo_unAuth_event	boolean	Photo Option for Log: Unauthorized Event
log_Accept_nonRegistered_facial	boolean	Accept Unregistered Facial
log_case_event	boolean	Case open/close event log
log_EM_Option	string	EM card model log setting: “FullSN” / “ID” / “FC_ID”

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "log setting updated"
	}

Response data:

name	type	description
status	string	status response
message	string	return message

11.17 get Log Setting

Description: get the Log setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/logsetting/{sessionid}
Example URI	/cgi-bin/restapi/logsetting/1805350299

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	{ "log_event": "user_audit_event", "log_size": 75000, "log_unAuth_event": true, "log_Accept_nonRegistered_card": true, "log_photo_Auth_event": true, "log_photo_unAuth_event": true, "log_Accept_nonRegistered_facial": true,

	"log_case_event": true, “log_EM_Option”:”FullSN” }

Response data:

Same as above set Log Setting

12. Job Code

12.1 get JobCode Settings

Description: get jobcode setting

REQUEST

Method	GET
Path	/cgi-bin/restapi/jobcodesetting/{sessionid
Example URI	/cgi-bin/restapi/jobcodesetting/1805350299/

Request Parameters:

name	type	description
{sessionid}	long	session id, required

RESPONSE

Response Code	200
example	[{ "jobcode": 1, "name": "Job 1111", "enable": "disable" }, { "jobcode": 2, "name": "Occup. 2222",

	"enable": "disable"
	},
	{
	"jobcode": 3,
	"name": "Customise 3333",
	"enable": "disable"
	}
	]

Response data:

name	type	description
jobcode	integer	jobcode setting id either 1,2 or 3
name	string	jobcode setting description
enable	integer	enable status either ‘enable’ or ‘disable’

12.2 update JobCode Settings

Description: update jobcode setting

REQUEST

Method	PUT
Path	/cgi-bin/restapi/jobcodesetting/{sessionid
Example URI	/cgi-bin/restapi/jobcodesetting/1805350299/

example json	{
	"jobcode": 1,
	"name": "Job 1111",
	"enable": "disable"
	}
	or in array form:
	[JSON1, JSON2,...]

Request Parameters:

name	type	description
{sessionid}	long	session id, required
jobcode	integer	jobcode setting id either 1,2 or 3
name	string	jobcode setting description
enable	integer	enable status either ‘enable’ or ‘disable’

*note: if any one of jobcode setting enabled, jobcode will be enable in terminal setting web ui terminal setting page

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "jobcode setting 1 updated"
	}
	or in array form:
	[JSON1, JSON2,...]

Response data:

name	type	description
status	string	status response
message	string	return message

12.3 add JobCodes

Description: add new jobcode

REQUEST

Method	POST
Path	/cgi-bin/restapi/jobcode/{sessionid
Example URI	/cgi-bin/restapi/jobcode/1805350299/

example json	{
	"jobcode": 1,
	“id”: 1
	"name": "Job 1-1",
	"enable": "disable"
	}
	or in array form:
	[JSON1, JSON2,...]

Request Parameters:

name	type	description
{sessionid}	long	session id, required
jobcode	integer	jobcode setting id , required either 1,2 or 3
id	integer	id under this jobcode setting, required 1 to 999999
name	string	jobcode description, required
enable	integer	enable status, required either ‘enable’ ,‘disable’ or ‘default’ Only can 1 default in the same jobcode setting.

RESPONSE

Response Code	200
example	{ "status": "success", "message": "jobcode 1 id 2 added"

	} or in array form: [JSON1, JSON2,...]

Response data:

name	type	description
status	string	status response
message	string	return message

12.4 updateJobCodes

Description: add new jobcode

REQUEST

Method	PUT
Path	/cgi-bin/restapi/jobcode/{sessionid
Example URI	/cgi-bin/restapi/jobcode/1805350299/

example json	{
	"jobcode": 1,
	“id”: 1
	"name": "Job 1-1",
	"enable": "disable"
	}
	or in array form:
	[JSON1, JSON2,...]

Request Parameters:

name	type	description
{sessionid}	long	session id, required
jobcode	integer	jobcode setting id , required either 1,2 or 3

id	integer	id under this jobcode setting, required 1 to 999999
name	string	jobcode description
enable	integer	enable status either ‘enable’ ,‘disable’ or ‘default’ Only can 1 default in the same jobcode setting.

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "jobcode 1 id 2 updated"
	}
	or in array form:
	[JSON1, JSON2,...]

Response data:

name	type	description
status	string	status response
message	string	return message

12.5 delete JobCodes

Description: delete jobcode

REQUEST

Method	DELETE

Path	/cgi-bin/restapi/jobcode/{sessionid}?jobcode &id &name
Example URI	/cgi-bin/restapi/jobcode/1805350299?jobcode =1&id=1&name=ONE

Request Parameters:

name	type	description
{sessionid}	long	session id, required
jobcode	integer	jobcode setting id , required either 1,2 or 3
id	integer	id under this jobcode setting, required 1 to 999999
name	string	jobcode description, required

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "jobcode 1 id 2 deleted"
	}
	or in array form:
	[JSON1, JSON2,...]

Response data:

name	type	description
status	string	status response
message	string	return message

12.6 getJobCodes

Description: get jobcode

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/jobcode/{sessionid}?jobcode &id &name &enable
Example URI	/cgi-bin/restapi/jobcode/1805350299?jobcode =1&id=1&name=ONE&enable=enable

Request Parameters:

name	type	description
{sessionid}	long	session id, required
jobcode	integer	jobcode setting id , required when id provided either 1,2 or 3
id	integer	id under this jobcode setting 1 to 999999
name	string	jobcode description
enable	integer	enable status either ‘enable’ ,‘disable’ or ‘default’ Only can 1 default in the same jobcode setting.

RESPONSE

Response Code	200
example	{ "jobcode": 1, "id": 1,

	"name": "ONE", "enable": "enable" } or in array form: [JSON1, JSON2,...]

Response data:

name	type	description
jobcode	integer	jobcode setting id either 1,2 or 3
id	integer	id under this jobcode setting, required 1 to 999999
name	string	jobcode description
enable	integer	enable status either ‘enable’ ,‘disable’ or ‘default’ Only can 1 default in the same jobcode setting.

13. Holiday

13.1 get Holidays

Description: get Holiday

REQUEST

Method	GET
Path	/cgi-bin/restapi/holiday/{sessionid?date
Example URI	/cgi-bin/restapi/holiday/1805350299/date=2025-0-0

Request Parameters:

name	type	description
{sessionid}	long	session id, required
date	xsd:date	YYYY-MM-DD format value 0 of year/month/day as wildcard

RESPONSE

Response Code	200

example	{
	"dates": ["2027-10-15",
	“2026-05-10”]
	}

Response data:

name	type	description
dates	xsd:date	YYYY-MM-DD format, can be in array

13.2 set Holidays

Description: add new holiday

REQUEST

Method	POST
Path	/cgi-bin/restapi/holiday/{sessionid
Example URI	/cgi-bin/restapi/holiday/1805350299/
example json	{

	["2026-10-12", "2025-13-01"] }

Request Parameters:

name	type	description
{sessionid}	long	session id, required
dates	xsd:date	YYYY-MM-DD format, can be in array

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "some holiday added",
	"invalid_dates": ["2025-13-01"]
	}

Response data:

name	type	description
status	string	status response
message	string	return message
invalid_dates	xsd:date	array of xsd:date for invalid date not added.

13.3 deleteHolidays

Description: delete holiday

REQUEST

Method	DELETE
Path	/cgi-bin/restapi/holiday/{sessionid
Example URI	/cgi-bin/restapi/holiday/1805350299/

example json	{
	["2026-10-12", "2025-13-01"]
	}

Request Parameters:

name	type	description
{sessionid}	long	session id, required
dates	xsd:date	YYYY-MM-DD format, can be in array

RESPONSE

Response Code	200

example	{
	"status": "success",
	"message": "1 holiday deleted",
	"invalid_dates": ["2027-10-125"]
	}

Response data:

name	type	description
status	string	status response
message	string	return message
invalid_dates	xsd:date	array of xsd:date for invalid date not deleted. invalid format or not found

14. Common Error response

Common error response:

Response Code	400/500

example	{
	“status”:”error”,
	“message”:”xxxxxxx”
	}

Response data

name	type	description
status	string	status response
message	string	error description