

# USER MANUAL

## ZKBio CVSecurity 3rd Party API

---

Version: 1.0

Date: December 20, 2022

Software Version: ZKBio CVSecurity 5.2.0 or above Version

## Important Statements

Firstly, thank you for choosing our product. Before using the product, please read this manual carefully to avoid any unnecessary damage. Proper operation of the product will result in better performance and faster verification.

None of the content of this document shall be copied or delivered in any forms or by any means without the prior written consent of our company.

The product described in the manual may include the software whose copyrights are shared by the licensors, including our company. No one shall copy, distribute, revise, modify, extract, decompile, disassemble, decrypt, reverse engineering, lease, transfer, sub-license the software, or perform other acts of copyright infringement, unless such restrictions are prohibited by applicable laws or such actions are approved by respective copyright holders.



Information provided in this manual may differ from actual technical specifications due to the constant development of products. Our company claims no responsibility for any disputes arising out of any discrepancy between actual technical parameters and those described in this document. The document is subject to change without prior notice.

# Table of Contents

Important Statements .....	1
1. Interface Description .....	7
1.1 Authorize API Level .....	7
1.2 Interface Parameter .....	9
2. Interface Details .....	10
2.1 Access Control Interface .....	10
2.1.1 Delete Person Level [accLevel/deleteLevel] .....	10
2.1.2 Get Level by Id [accLevel/get byId{id}] .....	10
2.1.3 Get Level by Name [accLevel/get ByName{name}] .....	11
2.1.4 Get Access Level/Llist [accLevel/list] .....	12
2.1.5 Sync Level Info to Device [accLevel/syncLevel] .....	12
2.1.6 Sync Person Level [accLevel/syncPerson] .....	13
2.2 Elevator Control Level Interface .....	13
2.2.1 Delete person Level [eleLevel/deleteLevel] .....	13
2.2.2 Get Ele Level by Id[eleLevel/getById/{id}] .....	14
2.2.3 Get Ele Level by Name[eleLevel/getByName/{name}] .....	14
2.2.4 Get Ele Level Group List[[eleLevel/list] .....	15
2.2.5 Sync Ele Level Group Info to Device[eleLevel/syncLevel] .....	16
2.2.6 Sync Person Ele Level [eleLevel/syncPerson] .....	16
2.3 Park Authorization Interface .....	17
2.3.1 Authorize Park Black and White List [parkAuthorize/addBlackWhite] .....	17
2.3.2 Delete Park Black and White List[parkAuthorize/delBlackWhite] .....	17
2.3.3 Edit park Black and White List[parkAuthorize/editBlackWhite] .....	18
2.4 Card Interface .....	18
2.4.1 Get Card List by Pin [card/getCards/{pin}] .....	18
2.4.2 Set Person Card Info[card/set] .....	19
2.5 Department Interface .....	20

2.5.1 Add /Edit Department[department/add] .....	20
2.5.2 Delete Department by Code [department/delete/{code}] .....	20
2.5.3 Get Department Info by Code [department/get/{code}] .....	21
2.6 Media Interface .....	22
2.6.1 Add or Update Media Info[adMedia/add] .....	22
2.6.2 Delete Media by Name[adMedia/delete/{name}] .....	22
2.6.3 Get Media Object by Name[adMedia/get/{name}] .....	23
2.7 Area Interface .....	24
2.7.1 Get Area Object by Area Code[area/get/{code}] .....	24
2.7.2 Get Area List [area/list] .....	25
2.8 Person Interface of Att Area .....	26
2.8.1 Delete Area Person [attAreaPerson/delete] .....	26
2.8.2 Set Area Person [attAreaPerson/set] .....	26
2.9 Access Control Device Interface .....	27
2.9.1 Get Device List [device/acclist] .....	27
2.9.2 Get Device Info by Sn [device/getAcc/{sn}] .....	28
2.10 Ele Device Interface .....	29
2.10.1 Get Ele Device List [eleDevice/eleList] .....	29
2.10.2 Get Ele Device Info by Sn [eleDevice/getEle/{sn}] .....	30
2.11 Door Interface .....	31
2.11.1 Get the State of All Doors [door/all DoorState] .....	31
2.11.2 Get Door State by Door Id [door/doorState ById] .....	33
2.11.3 Get Door State by the Sn of Device [door/doorStateBy{Sn}] .....	35
2.11.4 Get Door Info by Door Id [door/get] .....	36
2.11.5 Get Door List [door/list] .....	37
2.11.6 Remote Close by Door Id [door/remoteCloseById] .....	38
2.11.7 Remote Close by Door Name [door/remoteCloseByName] .....	38
2.11.8 Remote Open by Door Id [door/remoteOpenById] .....	39
2.11.9 Remote Open by Door Name[door/remoteOpenByName] .....	39

2.12 Floor interface .....	40
2.12.1 Get Floor Info List[floor/list] .....	40
2.12.2 Remote Close by Floor Id[floor/remoteCloseById] .....	41
2.12.3 Remote Open by Floor Id[floor/remoteOpenById] .....	42
2.13 Person Interface .....	42
2.13.1 Add/ Edit Person Info [person/add] .....	42
2.13.2 Delete Person Info by Pin [person/delete/{pin}] .....	43
2.13.3 Get person Info by Pin [person/get/{pin}]Get person Info by Pin [person/get/{pin}] ....	44
2.13.4 Get personnel information according to the personnel identification number array and the department code array .....	45
2.13.5 Get dynamic QR code by personal identification number .....	48
2.13.6 Resignation .....	49
2.13.7 Add / Edit personnel basic information .....	50
2.13.8 Update personnel Photo by PIN .....	53
2.14 Access Control Reader Interface .....	53
2.14.1 Get Access Control Reader Info List[reader/acclist] .....	53
2.14.2 Get Access Control Reader Info by Reader Id [reader/getAcc] .....	55
2.15 Visitor Reservation Interface .....	55
2.15.1 Add Visitor Reservation[visReservation/add] .....	55
2.15.2 Delete Visitor Reservation[visReservation/del] .....	57
2.16 Visitor Level Interface .....	58
2.16.1 Get Visitor Level Group List[visLevel/list] .....	58
2.17 Visitor Registration Check Out Interface .....	59
2.17.1 Add Visitor Registration[visRegistration/add] .....	59
2.17.2 Visitor Check Out [visRegistration/exit] .....	60
2.17.3 Get Visitor Dynamic QR code .....	60
2.18 Access Control transaction Interface .....	61
2.18.1 Get Transaction by Device Sn [transaction/device/{deviceSn}] .....	61
2.18.2 Get Access Transaction List[transaction/list] .....	63
2.18.3 Get Access Real-Time Monitoring Transaction[transaction/monitor] .....	65

2.18.4 Get Door Transaction by Pin [transaction/person/{pin}] .....	66
2.18.5 Get Personnel First In and Last Out record by Pin [transaction/firstInAndLastOut/{pin}]	69
2.18.6 Obtain door transaction, daily transaction, personnel access transactions [transaction/getDoorTransactions] .....	70
2.18.7 Get Doors transaction details [transaction/getDoorTransactionDetail] .....	72
2.19 Elevator Transaction Interface .....	74
2.19.1 Get Elevator Transaction [eleTransaction/list] .....	74
2.20 Park Transaction Interface .....	77
2.20.1 Get Car Entry/Exit record pictures [parkTransaction/getParkRecordPicture] .....	77
2.20.2 Get Park Check-in Transaction [parkTransaction/listParkRecordin] .....	77
2.20.3 Get park Check-Out Transaction[parkTransaction/listParkRecordout] .....	79
2.21 Att Transaction Interface .....	81
2.21.1 Get Att Transaction [transaction/listAttTransaction] .....	81
2.22 Att Device Interface .....	84
2.22.1 Get Att Device [device/getAtt] .....	84
2.23 Apply for abnormal attendance .....	85
2.23.1 Apply for Re-clock-in .....	85
2.23.2 Apply for overtime .....	85
2.23.3 Ask for leave .....	86
2.23.4 Get leave type list .....	88
2.23.5 Upload leave picture .....	89
2.23.6 Calculate the total duration of leave .....	89
2.24 Edit Attendance personnel information .....	91
2.24.5 Edit Personal information .....	91
2.25 Person BioTemplate Interface .....	93
2.25.1 Add Personnel BioTemplate .....	93
2.25.2 Delete By PersonId And TemplateNo .....	93
2.25.3 Delete Personnel BioTemplate By PersonId .....	94
2.25.4 Get Personnel BioTemplate By PersonId .....	94
2.26 Temperature transactions Interface .....	97

2.26.1	Get Hep Transaction [api/hep/transaction/list] .....	97
2.26.2	Get Mask and Temperature Real-Time Monitor events [api//hep/transaction/monitor]	99
2.27	Edit FaceKiosk Personal information .....	101
2.27.1	Edit Personal information .....	101
2.28	FaceKiosk Area Personal .....	102
2.28.1	Delete Area Personnel .....	102
2.28.2	Add Area Personnel .....	102
2.28.3	Get a list of attendance zones .....	103
2.29	FaceKiosk Device Interface .....	105
2.29.1	Get Facekiosk device list [api/device/attAdDeviceList] .....	105
2.30	Entrance Control Device Interface .....	107
2.30.1	Get entrance control device list [api/psgdDvice/list] .....	107
2.30.2	Get entrance control device info by SN [/api/psgDevice/getBySn] .....	108
2.31	Entrance Control Gate Interface .....	109
2.31.1	Get the State of All gates [psgGate/allGateState] .....	109
2.31.2	Get Entrance Control Gate State by Gate Id [psgGate/gateStateById] .....	110
2.31.3	Get Entrance Control Gate State by the SN of Device [psgGate/gateStateBySn] .....	112
2.31.4	Get Entrance Control Gate Info by gate Id [psgGate/getById] .....	113
2.31.5	Get Entrance Control Gate List [door/list] .....	114
2.31.6	Remote Open by Gate Id [psgGate/remoteOpenById] .....	114
2.31.7	Remote Open by Gate Name [psgGate/remoteOpenByName] .....	115
2.32	Entrance Control Gate Interface .....	115
2.32.1	Delete Person Level [psgLevel/deleteLevel] .....	115
2.32.2	Get Gate Control Level by Id [psgLevel/getById/{id}] .....	116
2.32.3	Get Entrance Access Level by Name [psgLevel/getByName/{name}] .....	116
2.32.4	Get Entrance Control Access Level [/psgLevel/list] .....	118
2.32.5	Sync Level Info to Device [psgLevel/syncLevel] .....	118
2.32.6	Sync Person Level [/psgLevel/syncPerson] .....	119
2.33	Entrance Control Gate Reader Interface .....	119

2.33.1 Get Entrance Control Gate Reader Info List[psgReader/list] .....	119
2.33.2 Get Entrance Control Gate Reader Info by Reader Id [psgReader/getByld] .....	120
2.34 Entrance Control Device transaction Interface .....	121
2.34.1 Get Entrance Control Transaction by Device Sn [psgTransaction/device/{deviceSn}]	121
2.34.2 Get Entrance Control Gate Transaction List[psgTransaction/list] .....	123
2.34.3 Get Entrance Control Device Real-Time Monitoring Transaction[psgTransaction/monitor] .....	125
2.34.4 Get Personnel Entrance Control Access Transaction by Pin [transaction/person/{pin}]	126
3. Appendix .....	129
3.1 Error Code .....	129

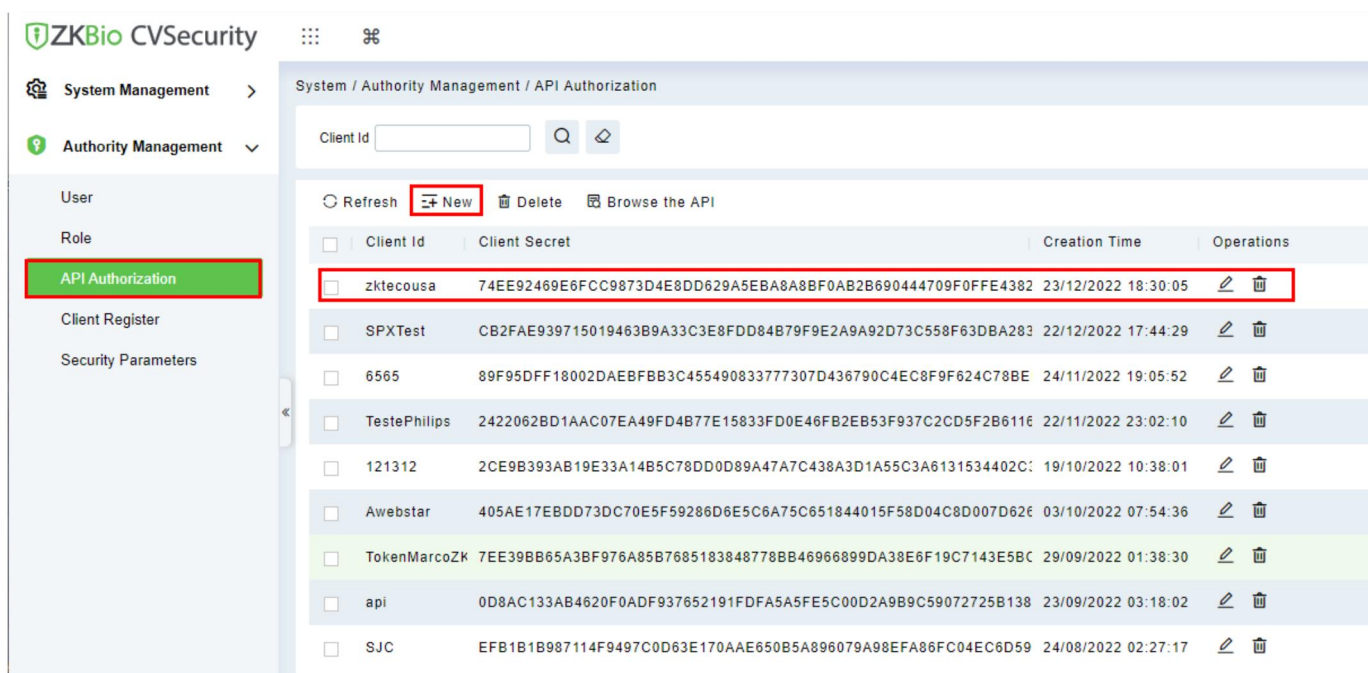


# 1. Interface Description

**ZKBio CVSecurity API** is dedicated to the platform data connection for the third-party system. The third-party system can read and set business data flexibly, which effectively reduces the complexity of the third-party system business integration and provides convenient, fast standard connection mode and data structure.

## 1.1 Authorize API Level

The API Authorization menu will only be displayed after the API license is activated. Open the corresponding authorized API menu and add the corresponding API client information as shown below:



ZKBio CVSecurity		<a href="https://cvsdemo.zkteco.com/api/api-docs">https://cvsdemo.zkteco.com/api/api-docs</a>	<a href="#">Explore</a>
<b>AccDevice : acc device</b>	Show/Hide	List Operations	Expand Operations
<b>AccDoor : acc door</b>	Show/Hide	List Operations	Expand Operations
<b>AccLevel : acc level</b>	Show/Hide	List Operations	Expand Operations
<b>AccReader : acc reader</b>	Show/Hide	List Operations	Expand Operations
<b>AccTransaction : acc transaction</b>	Show/Hide	List Operations	Expand Operations
<b>AdMedia : ins adMedia</b>	Show/Hide	List Operations	Expand Operations
<b>AttApply : att apply</b>	Show/Hide	List Operations	Expand Operations
<b>AttAreaPerson : att area person</b>	Show/Hide	List Operations	Expand Operations
<b>AttDevice : att device</b>	Show/Hide	List Operations	Expand Operations
<b>AttPerson : att person</b>	Show/Hide	List Operations	Expand Operations
<b>AttTransaction : att transaction</b>	Show/Hide	List Operations	Expand Operations
<b>EleDevice : ele device</b>	Show/Hide	List Operations	Expand Operations
<b>EleFloor : ele floor</b>	Show/Hide	List Operations	Expand Operations
<b>EleLevel : ele level</b>	Show/Hide	List Operations	Expand Operations

Based on the standard http interface definition style, in consistent with RESTFUL API style, supports https. Parameter descriptions are as following:

serverIP: ZKBio CVSecurity server or computer IP, such as: 192.168.1.100;

serverPort: The server port of ZKBio CVSecurity, such as: 8088;

access\_token: API access token is to check whether the requested permission is allowed or denied. All request should pass through a valid token, such as: apitoken;

## 1.2 Interface Parameter

<b>Public Response Result</b>	<pre>{   "code": 0,   "message": "string",   "data": {} }</pre>
<b>Public Response Result Parameter Description</b>	<p>code: error code, less than 0 means failure, more than 0 means success;  msg: Error message  data: can refer to each interface description</p>
<b>Basic Parameter Description</b>	<p>Sn: device serial number  Pin: pin  pageSize: page size;  pageNo: page no;</p>

## 2. Interface Details

### 2.1 Access Control Interface

#### 2.1.1 Delete Person Level [accLevel/deleteLevel]

<b>Post Request URL</b>	http://serverIP:serverPort/api/accLevel/deleteLevel?pin={pin}&levelIds={levelIds}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Pin: pin, required Level Ids: level group id List, multiple comma separated, required, string; such as 402856aa6c3bbde1016c3bc2329200ff,402856aa6c3bbde1016c3bc232930100
<b>Response Result</b>	Refer to public response results
<b>Response Result Description</b>	Public response parameter description

#### 2.1.2 Get Level by Id [accLevel/get byId{id}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/accLevel/getById/{id}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Id: level group id, required, string; such as "402856aa6c3bbde1016c3bc2329200ff"
<b>Response Result</b>	{ "code": 0, "message": "success", "data": { "id": "402856aa6c3bbde1016c3bc2329200ff", "name": "public level group" }}

	}
<b>Response Result Description</b>	Data: level group object; Id: level group id; Name: level group name;

### 2.1.3 Get Level by Name [accLevel/get ByName{name}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/accLevel/getByName/{name}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Name: level group name, required;
<b>Response Result</b>	{ "code": 0, "message": "success", "data": { "id": "402856aa6c3bbde1016c3bc2329200ff", "name": "public level group" } }
<b>Response Result Description</b>	Data: level group object; Id: level group id, string; such as '402856aa6c3bbde1016c3bc2329200ff' Name: level group name;

## 2.1.4 Get Access Level/List [accLevel/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/accLevel/list?pageNo={pageNo}&pageSize={pageSize}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page no, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3bbde1016c3bc2329200ff",       "name": "public level group"     }   ] }</pre>
<b>Response Result Description</b>	Data: level group array list; Id: level group id, string; such as "402856aa6c3bbde1016c3bc2329200ff" Name: level group name;

## 2.1.5 Sync Level Info to Device [accLevel/syncLevel]

<b>Post Request URL</b>	http://serverIP:serverPort/api/accLevel/syncLevel?levelId={levelId}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Level Id: level group id, required, string; such as "402856aa6c3bbde1016c3bc2329200ff"

<b>Response Result</b>	Refer to public response result, for any other result please refer to Appendix- <a href="#">Error code</a>
<b>Response Result Description</b>	Public response parameter description

## 2.1.6 Sync Person Level [accLevel/syncPerson]

<b>Post Request URL</b>	http://serverIP:serverPort/api/accLevel/syncPerson?pin={pin}&levelIds={levelIds}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Level Ids: level group id list, multiple comma separated, required, string; such as "402856aa6c3bbde1016c3bc2329200ff" Pin: pin, required
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	Public response parameter description

## 2.2 Elevator Control Level Interface

### 2.2.1 Delete person Level [eleLevel/deleteLevel]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleLevel/deleteLevel?pin={pin}&levelIds={levelIds}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Pin: pin, required level Ids: level Ids: level group id List, multiple comma separated, required, string; such as "402856aa6c3bbde1016c3bc2329200ff"
<b>Response Result</b>	Refer to public Response Result
<b>Response Result Description</b>	public response parameter description

## 2.2.2 Get Ele Level by Id[eleLevel/getById/{id}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleLevel/getById/{id}?access_token={apitoken}
<b>Request Mode</b>	get
<b>Request Content</b>	None
<b>Request Parameter Description</b>	id: level group, required, string; such as: "402856aa6c3bbde1016c3bc2329200ff"
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "402856aa6c3bbde1016c3bc2329200ff",     "name": "public level group"   } }</pre>
<b>Response Result Description</b>	data: level group object; id: level group id; name: level group name;

## 2.2.3 Get Ele Level by Name[eleLevel/getByName/{name}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleLevel/getByName/{name}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	name: level group name, required;
<b>Response Result</b>	<pre>{   "code": 0,</pre>



	<pre>"message": "success", "data": {   "id": "402856aa6c3bbde1016c3bc2329200ff",   "name": "public level group" } }</pre>
<b>Response Result Description</b>	<p>data: level group object;  id: level group id;  name: level group name;</p>

## 2.2.4 Get Ele Level Group List[[eleLevel/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleLevel/list?pageNo={pageNo}&page Size={pageSize}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>pageNo: page, required, must be greater than 0  pageSize: The number of per page, required, must be greater than 0</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3bbde1016c3bc2329200ff",       "name": "public level group"     }   ] }</pre>
<b>Response Result Description</b>	<p>data: level group array list;  id: level group id;  name: level group name;</p>

## 2.2.5 Sync Ele Level Group Info to Device[eleLevel/syncLevel]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleLevel/syncLevel?levelId={levelId}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	levelid: level group, required, string; such as: "402856aa6c3bbde1016c3bc2329200ff"
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.2.6 Sync Person Ele Level [eleLevel/syncPerson]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleLevel/syncPerson?pin={pin}&levelIds={levelIds}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	LevelIds: level group id list, multiple comma separated, required, string; such as: "402856aa6c3bbde1016c3bc2329200ff" pin: pin, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.3 Park Authorization Interface

### 2.3.1 Authorize Park Black and White List [parkAuthorize/addBlackWhite]

<b>Post Request URL</b>	http://serverIP:serverPort/api/parkAuthorize/addBlackWhite?type={type}&carNumber={carNumber}&fromTime={fromTime}&toTime={toTime}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	type: Black and White List type, 0: White List, 1:Black list carNumber: license plate number, such as: A32212 fromTime: start time, 2018-08-17 toTime: end time, 2019-08-16
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.3.2 Delete Park Black and White List[parkAuthorize/delBlackWhite]

<b>Post Request URL</b>	http://serverIP:serverPort/api/parkAuthorize/delBlackWhite?carNumber={carNumber}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	carNumber: license plate number, such as: A32212
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.3.3 Edit park Black and White List[parkAuthorize/editBlackWhite]

<b>Post Request URL</b>	http://serverIP:serverPort/api/parkAuthorize/editBlackWhite?type={type}&carNumber={carNumber}&oldCarNumber={oldCarNumber}&fromTime={fromTime}&toTime={toTime}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	type: Black and White List type, 0: White List, 1:Black list carNumber: new license plate number, such as A32212 oldCarNumber: old license plate number, such as A32212 fromTime: start tim, 2018-08-17(White List need to be filled in) toTime: end time 2019-08-16(White List need to be filled in)
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.4 Card Interface

### 2.4.1 Get Card List by Pin [card/getCards/{pin}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/card/getCards/{pin}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pin: pin, required;
<b>Response Result</b>	{ "code": 0, "message": "success", "data": [ {

	<pre>"pin": "1234", "cardNo": "111", "cardType": "0" } ] }</pre>
<b>Response Result Description</b>	<p>data: card number List;</p> <p>pin: pin;</p> <p>cardNo: card no;</p> <p>cardType: card type, 0 : master card, 1: sub card, a person only uses one card when the system does not open the function of one more card</p>

## 2.4.2 Set Person Card Info[card/set]

<b>Post Request URL</b>	http://serverIP:serverPort/api/card/set?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{ "pin": "1234", "cardNo": "212121", "cardType": "0" }</pre>
<b>Request Parameter Description</b>	<p>Pin: pin, required</p> <p>cardNo: card number, required;</p> <p>cardType: card type, 0: master card, 1: sub card, the system does not open multi-card function, sub card setting invalid</p>
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.5 Department Interface

### 2.5.1 Add /Edit Department[department/add]

<b>Post Request URL</b>	http://serverIP:serverPort/api/department/add?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "name": "department name", "code": "456", "parentCode": "123" }
<b>Request Parameter Description</b>	Name: department name, required; code: department code, required; parentCode: parent department number;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.5.2 Delete Department by Code [department/delete/{code}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/department/delete/{code}?access_token={apitoken}
<b>Request Mode</b>	DELETE
<b>Request Content</b>	None
<b>Request Parameter Description</b>	code: department code, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.5.3 Get Department Info by Code [department/get/{code}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/department/get/{code}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Code: department code, required;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "name": "4444",     "code": "3333",     "parentCode": "2"   } }</pre>
<b>Response Result Description</b>	Data: department object Name: department name; Code: department code; parentCode: parent department code

## 2.6 Media Interface

### 2.6.1 Add or Update Media Info[adMedia/add]

<b>Post Request URL</b>	http://serverIP:serverPort/api/adMedia/add&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	File: "bayue.png", mediaName: "Media name" (be included in a form data to submit)
<b>Request Parameter Description</b>	File: Media file, required mediaName:media name, required (be included in form data to send)
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.6.2 Delete Media by Name[adMedia/delete/{name}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/adMedia/delete/{name}&access_token={apitoken}
<b>Request Mode</b>	DELETE
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Name: media name, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description



### 2.6.3 Get Media Object by Name[adMedia/get/{name}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/adMedia/get/{name}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	name: media name, required;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3bbde1016c3bc2329200ff",       "mediaName": "111",       "mediaSize": "21KB",       "mediaSuffix": "png",       "mediaType": "1",       "mediaPath": ""/att/picture/1535520932078.png""     }   ] }</pre>
<b>Response Result Description</b>	<p>Data: media object</p> <p>Id: media id; such as: "402856aa6c3bbde1016c3bc2329200ff"</p> <p>mediaName: media name;</p> <p>mediaSize: media size;</p> <p>mediaType: media type; 1: picture; 2: video</p> <p>mediaPath: media path</p>

## 2.7 Area Interface

### 2.7.1 Get Area Object by Area Code[area/get/{code}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/attAreaPerson/area/get/{code}?code=1&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Code: department code, required;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": 1,     "code": "402856aa6c3bbde1016c3bc2329200ff ",     "name": "area name"   } }</pre>
<b>Response Result Description</b>	Data: area object; Id: area id; Code: area code; name: area name;

## 2.7.2 Get Area List [area/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/attAreaPerson/area/list?pageNo=1&pageSize=5&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pageNo: page, required, must be greater than 0 pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3bbde1016c3bc2329200ff",       "code": "1",       "name": "area name"     },     {       "id": "402856aa6c3bbde1016c3bc211e5005a",       "code": "2",       "name": "2"     }   ] }</pre>
<b>Response Result Description</b>	Data: area object List; Id: area id; code: area code; name: area name;

## 2.8 Person Interface of Att Area

### 2.8.1 Delete Area Person [attAreaPerson/delete]

<b>Post Request URL</b>	http://serverIP:serverPort/api/attAreaPerson/delete?access_token={apitoken}
<b>Request Mode</b>	DELETE
<b>Request Content</b>	{ "code": "1", "pins": "5" }
<b>Request Parameter Description</b>	Code: area code, required; Pins: pin, required
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.8.2 Set Area Person [attAreaPerson/set]

<b>Post Request URL</b>	http://serverIP:serverPort/api/attAreaPerson/set?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "code": "1", "pins": ["5"] }
<b>Request Parameter Description</b>	Code: area code, required; Pins: pin, required
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

<b>Description</b>	
--------------------	--

## 2.9 Access Control Device Interface

### 2.9.1 Get Device List [device/acclist]

<b>Post Request URL</b>	http://serverIP:serverPort/api/device/acclist?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pageNo: pageno, required, the value must be greater than 0; pageSize: the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3bbde1016c3bc2329200ff",       "sn": "0564140100195",       "name": "192.168.218.104",       "type": "inBIO160",       "state": "1",       "module": "acc"     },     {       "id": "402856aa6c3bbde1016c3bc211e50059",       "sn": "0566141900209",       "name": "192.168.218.11",       "type": "inBIO460",       "state": "1",       "module": "acc"     }   ] }</pre>

	<pre> } ] } </pre>
<b>Response Result Description</b>	<p>Data: device array list;</p> <p>Id: device id;</p> <p>Sn: device sn;</p> <p>Name: device name;</p> <p>state: device state, 1 enabled, 0 disabled;</p> <p>Module: module, access, attendance, elevator;</p>

## 2.9.2 Get Device Info by Sn [device/getAcc/{sn}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/device/getAcc?sn={sn}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Sn:device sn, required;
<b>Response Result</b>	<pre> { "code": 0, "message": "success", "data": { "id": "402856aa6c3bbde1016c3bc2329200ff", "sn": "5662012052868", "name": "192.168.214.15", "type": "inibo460", "state": "0", "module": "acc" } } </pre>
<b>Response Result Description</b>	<p>Data: device object;</p> <p>Id: device id;</p>

	Sn: device sn Name: device name; Type: device model type; state: device state, 1 enabled, 0 disabled; Module: module, access, attendance, elevator;
--	---

## 2.10 Ele Device Interface

### 2.10.1 Get Ele Device List [eleDevice/eleList]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleDevice/eleList?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pageNo: page no, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre> {   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3bbde1016c3bc2329200ff",       "sn": "0564140100195",       "name": "192.168.218.104",       "type": "EC-100",       "state": "1",       "module": "ele"     },     {       "id": "402856aa6c3bbde1016c3bc210d90016",       "sn": "0566141900209", </pre>

	<pre> "name": "192.168.218.11",   "type": "EC-100",   "state": "1",   "module": "ele" } ] } </pre>
<b>Response Result Description</b>	<p>Data: Device array List;          Id: device id;          Sn: device serial number;          Name: device name;          Type: device model name;          state: state: device state, 1 enabled, 0 disabled;          Module: module, access, attendance, elevator;</p>

### 2.10.2 Get Ele Device Info by Sn [eleDevice/getEle/{sn}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleDevice/getEle?sn={sn}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Sn: device sn, required;
<b>Response Result</b>	<pre> {   "code": 0,   "message": "success",   "data": {     "id": "402856aa6c3bbde1016c3bc2329200ff",     "sn": "5662012052868",     "name": "192.168.214.15",     "type": "EC-100",     "state": "0",     "module": "ele"   } } </pre>



	<pre> } } </pre>
<b>Response Result Description</b>	<p>Data: device object;</p> <p>id: device id;</p> <p>sn: device sn;</p> <p>name: device name;</p> <p>type: device model name;</p> <p>state: state: device state, 1 enabled, 0 disabled;</p> <p>Module: module, access, attendance, elevator;</p>

## 2.11 Door Interface

### 2.11.1 Get the State of All Doors [door/all DoorState]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/allDoorState?timestamp={timestamp}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Timestamp: timestamp, millisecond, such as 1494467276539, required; Access frequency cannot be too high, it is recommended to obtain at least 10 seconds or more;
<b>Response Result</b>	<pre> {   "code": 0,   "message": "success",   "data": [     {       "id": door id, string; such as:       "402856aa6c3bbde1016c3bc210d90016"       "name": "door name",       "deviceId": device id, string; such as:       "402856aa6c3bbde1016c3bc211e5005a"       "connect": device connection state (0 offline, 1 online, 2 disabled),       "sensor": sensor state (unknown, no sensor, off, open),       "alarm": alarm state (none, alarm, timeout, be opened accidentally, </pre>

tampering, duress password opening, duress fingerprint opening, duress opening, normal open alarm, low battery voltage, replace battery, illegal operation, backup power, unknown)

"relay": relay state (off, open, unknown)

},

{

"id": "402856aa6c3bbde1016c3bc211e50059",

"name": "192.168.218.11-1",

"deviceId": "402856aa6c3bbde1016c3bc211e5005a",

"connect": "1",

"sensor": "no sensor",

"alarm": "None",

"relay": "off"

},

{

"id": "402856aa6c3bbde1016c3bc211e5005a",

"name": "192.168.218.11-2",

"deviceId": "402856aa6c3bbde1016c3bc211e5005a",

"connect": "1",

"sensor": "no sensor",

"alarm": "none",

"relay": "off"

},

{

"id": "402856aa6bff4b1a016bff5374800c0e",

"name": "192.168.218.11-3",

"deviceId": "402856aa6bff4b1a016bff5374800c0e",

"connect": "1",

"sensor": "no sensor",

"alarm": "None",

"relay": "off"

},

{

"id": "402856aa6c083827016c08582f8f0012",

"name": "192.168.218.11-4",

"deviceId": "402856aa6c3bbde1016c3bc211e5005a",

	<pre>"connect": "1", "sensor": "No sensor", "alarm": "None", "relay": "off" } ] }</pre>
<b>Response Result Description</b>	<p>Data: door state array List;</p> <p>Id: door id;</p> <p>Name: door name;</p> <p>Deviceld: device id;</p> <p>Connection: connection state, 0 offline, 1 online, 2 disabled;</p> <p>Sensor: sensor description (unknown, no sensor, off, open)</p> <p>Alarm: alarm description (none, alarm, timeout, be opened accidentally, tampering, duress password opening, duress fingerprint opening, duress opening, normal open alarm, low battery voltage, replace battery, illegal operation, backup power, unknown)</p> <p>relay: Relay state (off, open, unknown)</p>

### 2.11.2 Get Door State by Door Id [door/doorState ById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/doorStateById?doorId=1&timest amp={timestamp}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Timestamp: timestamp, millisecond, such as 1494467276539, required; Access frequency cannot be too high, it is recommended to obtain at least 10 seconds or more;
<b>Response Result</b>	<pre>{ "code": 0, "message": "success", "data": { "id": door id, string; such as: 402856aa6c083827016c08582f8f0012</pre>

	<pre> "name": "door name", "deviceId": device id, string; such as: "402856aa6c3bbde1016c3bc211e5005a" "connect": device connection state, "sensor": sensor state description, "alarm": alarm state description, "relay": relay state } } </pre>
<p><b>Response Result Description</b></p>	<p>data: door state object</p> <p>id: door id;</p> <p>name: door name;</p> <p>deviceId: device id;</p> <p>connect: connection state (0 offline, 1 online, 2 disabled);</p> <p>sensor: sensor description (unknown, no sensor, off, open);</p> <p>alarm: alarm state (none, alarm, timeout, be opened accidentally, tampering, duress password opening, duress fingerprint opening, duress opening, normal open alarm, low battery voltage, replace battery, illegal operation, backup power, unknown)</p> <p>relay: Relay state (off, open, unknown)</p>

### 2.11.3 Get Door State by the Sn of Device [door/doorStateBy{Sn}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/doorStateBySn?deviceSn={sn}&timestamp={timestamp}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Timestamp: timestamp, millisecond, such as 1494467276539, required; Access frequency cannot be too high, it is recommended to obtain at least 10 seconds or more; Sn: device sn
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": door id, string; such as:     "402856aa6c083827016c08582f8f0012",     "name": "door name",     "deviceId": device id,     "connect": device state description,     "sensor": sensor state description,     "alarm": alarm state description     "relay": relay state description   } }</pre>
<b>Response Result Description</b>	<p>data: door state object;</p> <p>id: door id;</p> <p>name: door name;</p> <p>deviceId: device id;</p> <p>connect: device connection state (0 offline, 1 online, 2 disabled);</p> <p>sensor: sensor state (unknown, no sensor, off, open);</p> <p>alarm: alarm state (none, alarm, timeout, be opened accidentally, tampering, duress password opening, duress fingerprint opening, duress opening, normal open alarm, low battery voltage, replace battery, illegal operation, backup power, unknown)</p> <p>relay: relay state (off, open, unknown)</p>

## 2.11.4 Get Door Info by Door Id [door/get]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/get?id={id}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	id: door id;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "402856aa6c083827016c08582f8f0012",     "name": "192.168.214.15-1",     "deviceId": "402856aa6c3bbde1016c3bc211e5005a",   } }</pre>
<b>Response Result Description</b>	data: door info object; id: doorid; name: door name; deviceId: deviceid;

## 2.11.5 Get Door List [door/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/list?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c083827016c08582f8f0012",       "name": "192.168.214.15-1",       "deviceId": "402856aa6c3bbde1016c3bc211e5005a",     },     {       "id": "402856aa6c08e7e4016c090be94a000a",       "name": "192.168.214.15-2",       "deviceId": "402856aa6c3bbde1016c3bc211e5005a",     },     {       "id": "402856aa6c08e7e4016c092400580021",       "name": "192.168.214.15-3",       "deviceId": "402856aa6c3bbde1016c3bc211e5005a",     },     {       "id": "402856aa6c3baabb016c3bb37365000a",       "name": "192.168.214.15-4",       "deviceId": "402856aa6c3baabb016c3bb37365000a"     }   ] }</pre>

	}
<b>Response Result Description</b>	data: door array list; id: doorid; name: door name; deviceId: device id;

### 2.11.6 Remote Close by Door Id [door/remoteCloseById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/remotecloseById?doorId={doorId}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	doorId: door id, required, string; such as: "402856aa6c3baabb016c3bb37365000a"
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.11.7 Remote Close by Door Name [door/remoteCloseByName]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/remotecloseByName?doorName={doorName}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	doorName: door name, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description



### 2.11.8 Remote Open by Door Id [door/remoteOpenById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/remoteOpenById?doorId={doorId}&interval={interval}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	doorId: door id, required, string; such as: "402856aa6c3baabb016c3bb37365000a", Interval: door opening time, seconds, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.11.9 Remote Open by Door Name [door/remoteOpenByName]

<b>Post Request URL</b>	http://serverIP:serverPort/api/door/remoteOpenByName?doorName={doorName}&interval={interval}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	doorName: door name, required; interval: door opening time, seconds, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.12 Floor interface

### 2.12.1 Get Floor Info List[floor/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/floor/list?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c08e7e4016c092400580021",       "floorNo": 1,       "name": "192.168.214.15-1",       "deviceId": "402856aa6c3c5063016c3cb3a0360005",     },     {       "id": "402856aa6c08e7e4016c090be94a000a",       "floorNo": 2,       "name": "192.168.214.15-2",       "deviceId": "402856aa6c3c5063016c3cb3a0360005"     },     {       "id": "402856aa6c083827016c08582f8f0012",       "floorNo": 3,       "name": "192.168.214.15-3",       "deviceId": "402856aa6c3c5063016c3cb3a0360005",     },     {</pre>

	<pre>"id": "402856aa6c3baabb016c3bb37365000a",   "floorNo":4,   "name": "192.168.214.15-4",   "deviceId":"402856aa6c3c5063016c3cb3a0360005",   } ] }</pre>
<b>Response Result Description</b>	<p>data: floor array list;  id: floor id;  floorNo: floor no;  name: floor name;  deviceId: device id;</p>

### 2.12.2 Remote Close by Floor Id[floor/remoteCloseById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/floor/remotecloseById?floorId={floorId}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	floorId: floorid, required, string; such as: "402856aa6c3c5063016c3cb3a0360005"
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.12.3 Remote Open by Floor Id[floor/remoteOpenById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/floor/remoteOpenById?floorId={floorId}&interval={interval}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	floorId: floor id, required, string; such as: "402856aa6c3c5063016c3cb3a0360005" interval: door opening time, seconds, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.13 Person Interface

### 2.13.1 Add/ Edit Person Info [person/add]

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/add?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "pin": "1234", "deptCode": "1", "name": "name", "lastName": "person name (English)", "gender": "F", "cardNo": "card no", "personPhoto": "picture-base64", "accLevelIds": "access level group ids", "accStartTime": "the start time of person valid access ", "accEndTime": "the end time of person valid access " }

<b>Request Parameter Description</b>	<p>Pin: pin, required</p> <p>deptCode: department code;</p> <p>name: name;</p> <p>lastName: person name (English)</p> <p>gender: gender, male, female;</p> <p>cardNo: card no;</p> <p>personPhoto :personnel comparison picture. "picture turn into base64 format"</p> <p>accLevelIds: access level groupid List, multiple comma separated, string; such as : 402856aa6c3c5063016c3cb3a0360005,402856aa6bff4b1a016bff5374800c0e,402856aa6c3baabb016c3bb37365000a;</p> <p>accStartTime: access start time, such as :2017-08-04 15:45:00;</p> <p>accEndTime: access end time, such as :2017-08-04 23:59:59;</p>
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.13.2 Delete Person Info by Pin [person/delete/{pin}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/delete/{pin}?access_token={apitoken}
<b>Request Mode</b>	DELETE
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pin: pin, required;
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.13.3 Get person Info by Pin [person/get/{pin}]Get person Info by Pin [person/get/{pin}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/get/{pin}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pin: pin, required;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "pin": "pin",     "deptCode": department code,     "deptName": "department name",     "name": "name",     "lastName": "person name (English)",     "gender": "gender",     "cardNo": "card no"   } }</pre>
<b>Response Result Description</b>	<p>data: person data info;</p> <p>pin: pin;</p> <p>gender: gender, M: male, F: female;</p> <p>deptCode: department code;</p> <p>deptName: department name;</p> <p>name: name;</p> <p>lastName: person name (English);</p> <p>cardNo: card no;</p>

## 2.13.4 Get personnel information according to the personnel identification number array and the department code array

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/getPersonList?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "pins": "1,2,3",   "deptCodes": "1,2",   "pageNo": 1,   "pageSize": 10 }</pre>
<b>Request Parameter Description</b>	<p>pageNo: page number.(Page number is necessary and must be greater than zero.)</p> <p>pageSize: Number of entries per page is necessary and must be greater than zero.;</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "pin": "PIN Personal identification number",     "deptCode": "Department Code",     "deptName": "Department Name",     "name": "First Name",   } }</pre>

	<pre> "lastName": "Last Name", "gender": "Gender", "cardNo": Card number, "personPhoto": Personal photo( base64 photo), "vislightPhoto":Facial recognition photo(base64 photo), "vislightPhotoPath": "/upload/pers/user/cropface/789789/789789.jpg" } } </pre>
<p><b>Response Result Description</b></p>	<p>data: Personnel information date;</p> <p>pin: Personal identification number;</p> <p>gender: gender, M is male, F is female;</p> <p>deptCode: Department Code;</p> <p>deptName: Department Name;</p> <p>name: First Name;</p> <p>lastName: Last Name;</p> <p>cardNo: Card number;</p> <p>personPhoto: personal photo (base64 photo);</p> <p>vislightPhoto: Facial recognition photo (base64 photo);</p>



vislightPhotoPath: Facial recognition photo path;

## 2.13.5 Get dynamic QR code by personal identification number

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/getQrCode/{pin}?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "pin": "123" }
<b>Request Parameter Description</b>	pin: Personal identification number and it is necessary
<b>Response Result</b>	{ "code":0, "message": "success", "data": "2#6SQLlaSLhprGhpkCgFrhHfdyZXFJxK2DLy+oLVolmol=" }
<b>Response Result Description</b>	data: dynamic QR code (base64 data);

## 2.13.6 Resignation

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/leave?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "leaveDate": "2019-06-10",   "leaveType": "1",   "pin": "123" }</pre>
<b>Request Parameter Description</b>	<p>leaveDate: Leave Date</p> <p>leaveType: Leave Type</p> <p>pin: Personal identification number and it is necessary</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": null }</pre>
<b>Response Result Description</b>	http://serverIP:serverPort/api/person/leave?access_token={apitoken}

## 2.13.7 Add / Edit personnel basic information

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/addPersonnelBasicInfo?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "pin": "1234567",   "deptCode": "1",   "birthday": "2016-07-15",   "cardNo": "123456789",   "certNumber": "123456",   "certType": "2",   "email": "123@zkteco.com",   "gender": "F",   "hireDate": "2019-06-10",   "isDisabled": false,   "isSendMail": true,   "lastName": "lastName",   "mobilePhone": "15123456789",   "name": "max",   "personPwd": "123456",   "ssn": "111111",   "supplyCards": "987643", }</pre>

**Request Parameter Description**

pin: Personal identification number and it is necessary;

deptCode: Department Code;

name: First Name;

lastName: Last Name;

gender: Gender, M: male, F: female;

cardNo: card number. Only cardNO is setted as an empty string the card number be deleted.If it is not set, the original card number will be maintained. The format is: "cardno":

birthday: Birthday;

certNumber: Certificate number;

certType: certificate type.1 : Second generation ID card, 2: Passport, 3: Driver's License, 4: Work Permit, 5: Citizen Card, 6: Driver's License; 1000: Xiamen Social Security Card, 1001: Hong Kong, Macao passport;

email: Email address, such as 123@zkteco.com;

hireDate: employee's entry date,such as: " 2019-06-10";

isDisabled: whether it is a prohibited list. True: Yes; False: No

isSendMail: whether to send mail. True: send; False: do not send;

mobilePhone: phone number,such as: "15123456789",

	<p>personPwd: device authentication password such as: "123456";</p> <p>ssn: social security number;</p> <p>supplyCards: Secondary cards. To use this function, you need to set up one person multiple cards in the personnel parameter setting. You can set multiple cards, which are separated by commas, such as: "987643, 2233452";</p>
<b>Response Result</b>	Refer to the general response results, others refer to Appendix - Exception Dictionary
<b>Response Result Description</b>	General response parameter description

## 2.13.8 Update personnel Photo by PIN

<b>Post Request URL</b>	http://serverIP:serverPort/api/person/updatePersonnelPhoto ?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{  "pin": "1234567",  "personPhoto": "string",  }
<b>Request Parameter Description</b>	pin: Personal identification number and it is necessary;  personPhoto: Personnel photos and it is in Base64 format.If no value is transferred, no processing will be performed
<b>Response Result</b>	Refer to the general response results, others refer to Appendix - Exception Dictionary
<b>Response Result Description</b>	General response parameter description

## 2.14 Access Control Reader Interface

### 2.14.1 Get Access Control Reader Info List[reader/acclist]

<b>Post Request URL</b>	http://serverIP:serverPort/api/reader/acclist?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET

<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre> {   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3baabb016c3bb37365000a",       "doorId": "402856aa6bff4b1a016bff5374800c0e",       "name": "192.168.214.15-1-in",       "readerNo": 1,       "readerState": 0     },     {       "id": "402856aa6c3c5063016c3cb3a0360005",       "doorId": "402856aa6bff4b1a016bff5374800c0e",       "name": "192.168.214.15-1-out",       "readerNo": 2,       "readerState": 1     }   ] } </pre>
<b>Response Result Description</b>	Data: reader array List; Id: reader id; DoorId: door id; Name: reader name; ReaderNo: readerNo; ReaderState: reader state, 0: in, 1: out;



## 2.14.2 Get Access Control Reader Info by Reader Id [reader/getAcc]

<b>Post Request URL</b>	http://serverIP:serverPort/api/reader/getAcc?id={id}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	id: reader id, required;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "402856aa6c3c5063016c3cb3a0360005",     "doorId": "402856aa6bff4b1a016bff5374800c0e",     "name": "192.168.214.15-1-in",     "readerNo": 1,     "readerState": 0   } }</pre>
<b>Response Result Description</b>	data: reader info object; id: reader id; doorId: doorid; name: reader name; readerNo: reader no; readState: reader state, 0: in, 1:out;

## 2.15 Visitor Reservation Interface

### 2.15.1 Add Visitor Reservation[visReservation/add]

<b>Post Request URL</b>	http://serverIP:serverPort/api/visReservation/add?access_token={apitoken}
-------------------------	---

<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "persPersonPin": "5",   "certType": "3",   "certNum": "4512",   "visitEmpName": "Mary",   "visitEmpPhone": "13015648651",   "company": "ZKTeco",   "visitReason": "visit",   "visitDate": "2018-08-31 14:40:00"   "visitEndDate": "2018-08-31 23:59:59" }</pre>
<b>Request Parameter Description</b>	<p>persPersonPin: interviewee pin, required;  certType: certificate type, required; details:  {  2:national ID card; 3:Passport; 4:Driver's license; 5:Work Permit;  6:Real-name card; 7:driving license; 8:others;  }  certNum: certificate no, required;  visitEmpName: interviewee name, required;  visitEmpPhone: visitor phone;  company: company;  visitReason: visit reason, required;  visitDate: visit date,yyyy-MM-dd mm:hh:ss, required;  visitEndDate: visit date,yyyy-MM-dd mm:hh:ss, required;</p>
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.15.2 Delete Visitor Reservation[visReservation/del]

<b>Post Request URL</b>	http://serverIP:serverPort/api/visReservation/del?certType={certType}&certNum={certNum}&visitDate={visitDate}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>certType: certificate type, required; details:</p> <pre>{   2: national ID card; 3: Passport; 4: Driver's license; 5:Work Permit;   6: Real-name card; 7: driving license; 8:others; }</pre> <p>certNum: certificate no, required;</p> <p>visitDate: visit date, required;</p> <p>[The system will find out the only visitor reservation information and delete it according to the certificate type, ID number, and visit date]</p>
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.16 Visitor Level Interface

### 2.16.1 Get Visitor Level Group List[visLevel/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/visLevel/list?pageNo={pageNo}&pageSize={pageSize}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6bff4b1a016bff5374800c0e",       "name": "public level group"     }   ] }</pre>
<b>Response Result Description</b>	data: level group array list; id: level group id; name: level groupname;

## 2.17 Visitor Registration Check Out Interface

### 2.17.1 Add Visitor Registration[visRegistration/add]

<b>Post Request URL</b>	http://serverIP:serverPort/api/visRegistration/add?&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "persPersonPin": "1",   "certType": "8",   "certNum": "456888",   "visEmpName": "max",   "visitEmpPhone": "",   "company": "",   "visitReason": "visit",   "visitorCount": "1",   "startTime": "2018-08-31 14:40:00",   "endTime": "2018-08-31 23:59:59",   "cardNo": "",   "visLevels": "" }</pre>
<b>Request Parameter Description</b>	<p>"persPersonPin": "Interviewee pin", required  certType: certificate type, required; details:  {  2:national ID card; 3:Passport; 4:Driver's license; 5:Work Permit;  6:Real-name card; 7:driving license; 8:others;  }  "certNum": visitor certificate no, required  "visitEmpName": visitor name, required  "visitEmpPhone": visitor mobile phone,  "company" : company,  "visitReason": visit reason, required  visitorCount: the number of visitors, required</p>

	<p>startTime: start time, yyyy-MM-dd mm:hh:ss, required</p> <p>endTime: end time, yyyy-MM-dd mm:hh:ss, required</p> <p>cardNo: card no</p> <p>visLevels: visitor level group id, string; such as: 402856aa6bff4b1a016bff5374800c0e,402856aa6c3c5063016c3c b3a0360005</p>
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.17.2 Visitor Check Out [visRegistration/exit]

<b>Post Request URL</b>	http://serverIP:serverPort/api/visRegistration/exit?certType={certType}&certNum={certNum}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>certType:certificate type,required;details:</p> <pre>{   2: national ID card;3:Passport;4:Driver's license;5:Work Permit;   6: real-name card;7:driving license;8:others; }</pre> <p>"certNum": visitor certificate no, required</p>
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.17.3 Get Visitor Dynamic QR code

<b>Post Request URL</b>	http://serverIP:serverPort/api/visRegistration/getQrCode/{pin}?access_token={apitoken}
-------------------------	--

<b>Request Mode</b>	POST
<b>Request Content</b>	{ "pin": "80000001"
<b>Request Parameter Description</b>	Pin: visitor number, required
<b>Response Result</b>	{ "code": 0, "data": {}, "message": "string" }
<b>Response Result Description</b>	Data: visitor's dynamic qr code, base 64 data;

## 2.18 Access Control transaction Interface

### 2.18.1 Get Transaction by Device Sn [transaction/device/{deviceSn}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/transaction/device/{deviceSn}?pageNo=1&pageSize=20&acc_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: pageNo, required, the value must be greater than 0; PageSize: required, the value must be greater than 0, not more than 1000 per page; DeviceSn: device sn;
<b>Response Result</b>	{ "code": 0, "message": "success", "data": [ { "id": "402856aa6c3c5063016c3cb3a0360005",

	<pre> "eventTime": "2017-08-04 15:59:12", "pin": "", "name": "", "lastName": "", "deptName": "", "areaName": "area name", "cardNo": "", "devSn": "5662012052868", "verifyModeName": "others", "eventName": "enable device", "eventPointName": "", "readerName": "Main door-In Reader", "devName": "192.168.214.15", "logId": 11734 }, { "id": "402856aa6c028430016c02d48ac10003", "eventTime": "2017-08-04 15:58:49", "pin": "", "name": "", "lastName": "", "deptName": "", "areaName": " area name", "cardNo": "", "devSn": "5662012052868", "verifyModeName": "others", "eventName": "connect server successfully", "eventPointName": "", "readerName": "others", "devName": "192.168.214.15", "logId": 11733 } ] } </pre>
<b>Response Result</b>	data: transaction array list;



<b>Description</b>	<p>id: "402856aa6c032ca3016c03351d2600fd";</p> <p>eventTime: record device trigger time;</p> <p>pin: pin;</p> <p>name: person name;</p> <p>lastName: person name, (English);</p> <p>deptName: department name;</p> <p>areaName: area name;</p> <p>cardNo: card no;</p> <p>devSn: device sn;</p> <p>verifyModeName: verification mode name;</p> <p>eventName: transaction description;</p> <p>eventPointName: transaction point name;</p> <p>readerName: reader name;</p> <p>devName: device name;</p> <p>logId: transaction index value, there will be a value if device support it;</p>
--------------------	--

### 2.18.2 Get Access Transaction List[transaction/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/transaction/list?pageNo=1&pageSi ze=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>personPin: personPin, [not required]</p> <p>startDate: startDate, [not required]</p> <p>endDate: endDate, [not required]</p> <p>PageNo: page no, required, the value must be greater than 0;</p> <p>pageSize, the number of each page, required, must be greater than 0, not more than 1000 per page;</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [</pre>

```
{
  "id": "402856aa6c032ca3016c03362ec10104",
  "eventTime": "2017-08-04 15:59:12",
  "pin": "",
  "name": "",
  "lastName": "",
  "deptName": "",
  "areaName": " area name",
  "cardNo": "",
  "devSn": "5662012052868",
  "verifyModeName": "others",
  "eventName": "enable device",
  "eventPointName": "",
  "readerName": "others",
  "accZone": null,
  "devName": "192.168.214.15",
  "logId": 11734
},
{
  "id": "402856aa6c082c51016c082ef4e70001",
  "eventTime": "2017-08-04 15:58:49",
  "pin": "",
  "name": "",
  "lastName": "",
  "deptName": "",
  "areaName": "area name",
  "cardNo": "",
  "devSn": "5662012052868",
  "verifyModeName": "others",
  "eventName": "connect server successfully",
  "eventPointName": "",
  "readerName": "others",
  "accZone": null,
  "devName": "192.168.214.15",
  "logId": 11733
}
```

	<pre> } ] } </pre>
<b>Response Result Description</b>	<p>data: transaction array list;</p> <p>id:5;</p> <p>eventTime: record device trigger time;</p> <p>pin: pin;</p> <p>name: person name;</p> <p>lastName: person name, (English);</p> <p>deptName: department name;</p> <p>areaName: area name;</p> <p>cardNo: card no;</p> <p>devSn: device sn;</p> <p>verifyModeName: verification mode name;</p> <p>eventName: transaction description;</p> <p>eventPointName: transaction point name;</p> <p>readerName: reader name;</p> <p>accZone: access area name;</p> <p>devName: device name;</p> <p>logId: transactionindex value, there will be a value if device support it;</p>

### 2.18.3 Get Access Real-Time Monitoring Transaction[transaction/monitor]

<b>Post Request URL</b>	http://serverIP:serverPort/api/transaction/monitor?timestamp={timestamp}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Timestamp: timestampmillisecond, such as 1494467276539, required; Access frequency cannot be too high, it is recommended to obtain at least 10 seconds or more; there is no need to change after timestamp value set the first request, it is allowed to keep using the first request
<b>Response Result</b>	{

	<pre> "code": 0, "message": "success", "data": [ { "eventTime": "2017-05-15 16:35:41", "pin": "", "areaName": " area name", "cardNo": "9606650", "devSn": "0564140100195", "verifyModeName": "card or fingerprint", "eventName": "card is not registered", "eventPointName": "192.168.218.11-1", "readerName": "192.168.218.11-1- out", "devName": "192.168.218.11(0566141900209)" } ] } </pre>
<p><b>Response Result Description</b></p>	<p>data:transaction monitoring record array List;  eventTime: record device trigger time;  pin:pin;  areaName: area name;  cardNo: card no;  devSn: device sn;  verifyModeName: verification mode name;  eventName: transaction description;  eventPointName: transaction point name;  readerName: reader name;  devName: device name;</p>

## 2.18.4 Get Door Transaction by Pin [transaction/person/{pin}]

<p><b>Post Request URL</b></p>	<p>http://serverIP:serverPort/api/transaction/person/{pin}?pageNo=1&amp;pageSize=20&amp;access_token={apitoken}</p>
<p><b>Request Mode</b></p>	<p>GET</p>

<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>PageNo: page no, required, the value must be greater than 0;</p> <p>pageSize, the number of each page, required, must be greater than 0, not more than 1000 per page;</p> <p>pin: pin, required;</p>
<b>Response Result</b>	<pre> {   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c17518e016c175385fd0001",       "eventTime": "2017-08-04 15:59:12",       "pin": "",       "name": "",       "lastName": "",       "deptName": "",       "areaName": "area name",       "cardNo": "",       "devSn": "5662012052868",       "verifyModeName": "others",       "eventName": " enable device",       "eventPointName": "",       "readerName": "others",       "devName": "192.168.214.15",       "logId": 11734     },     {       "id": 5,       "eventTime": "2017-08-04 15:58:49",       "pin": "",       "name": "",       "lastName": "",       "deptName": "",       "areaName": "area name", </pre>

	<pre> "cardNo": "", "devSn": "5662012052868", "verifyModeName": "others", "eventName": "connect server successfully", "eventPointName": "", "readerName": "others", "devName": "192.168.214.15", "logId": 11733 } ] } </pre>
<p><b>Response Result Description</b></p>	<p>data:transaction array list;  id:5;  eventTime: record device trigger time;  pin: pin;  name: name;  lastName: person name (English);  deptName: department name;  areaName: area name;  cardNo: card no;  devSn: device sn;  verifyModeName: verification mode name;  eventName: transaction description;  eventPointName: transaction point name;  readerName: reader name;  accZone: access area name;  devName: device name;  logId: transactionindex value, there will be a value if device support it;</p>

## 2.18.5 Get Personnel First In and Last Out record by Pin

[transaction/firstInAndLastOut/{pin}]

<b>Post Request URL</b>	<b>http://serverIP:serverPort/api/transaction/firstInAndLastOut/{pin}?pageNo=1&amp;pageSize=20&amp;access_token={apitoken}</b>
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>PageNo: page no, required, the value must be greater than 0;</p> <p>pageSize, the number of each page, required, must be greater than 0, not more than 1000 per page;</p> <p>pin: pin, required;</p> <p>startDate: startDate, [not required]</p> <p>endDate: endDate, [not required]</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "40285ab374521d14017452820eb50005",       "pin": "147",       "name": "",       "lastName": "",       "readerNameIn":       "192.168.218.183-1-Out",       "firstInTime": "2020-09-03 13:48:07",       "readerNameOut": null,       "lastOutTime": null,       "deptCode": "1",       "deptName": "Department Name"     }   ] }</pre>
<b>Response Result</b>	data:transaction array list;

<b>Description</b>	id:5; pin: pin; name: name; lastName: person name (English); deptCode: department code; deptName: department name; readerNameIn: First In reader name; firstInTime: First In Time; readerNameOut: Last Out reader name; lastOutTime: Last Out Time;
--------------------	--

## 2.18.6 Obtain door transaction, daily transaction, personnel access transactions [transaction/getDoorTransactions]

<b>Post Request URL</b>	<b>http://serverIP:serverPort/api/transaction/getDoorTransactions?access_token={apitoken}</b>
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "devSn": "5662012052868",   "startTime": "2021-11-19 18:00:00",   "endTime": "2021-11-19 18:00:00",   "filter": "Name",   "type": "normal",   "eventNo": "0",   "personPin": "320022",   "pageNo": "1",   "pageSize": "100", }</pre>
<b>Request Parameter Description</b>	devsn: Device Serial Number [Not required] startTime: Start Time [Not required] endTime: End Time [Not required] filter: Filter conditions [Not required, filter by name, last name, PIN] (Fuzzy query) type: Event Type [Not required, alarm events, normal event,



	<p>warning events]</p> <p>eventNo: Events Number [Not required]</p> <p>personPin: personnel ID number[Not required] [Accurate Query]</p> <p>page: page number</p> <p>pageSize: record number of current page</p>
<p><b>Response Result</b></p>	<pre>{   "ret": "ok",   "msg": "operation succeed",   "data": [     {       "id": "402856a7695c716301695c81290d0006",       "eventNo": 8,       "eventName": "remote opening",       "eventTime": "2019-03-08 16:52:38",       "doorName": "192.168.214.197-1",       "personPin": "",       "personName": "",       "areaName": "Area Name",       "personLastName": "",       "eventLevel": "0"     },     {       "id": "402856a7695c716301695c7fe7620004",       "eventNo": 8,       "eventName": "remote opening",       "eventTime": "2019-03-08 16:51:14",       "doorName": "192.168.214.197-1",       "personPin": "",       "personName": "",       "areaName": "Area Name",       "personLastName": "",       "eventLevel": "0"     }   ],   "i18nArgs": null, }</pre>

	<pre> "page": 1, "pageSize": 10, "total": 2, "success": true } {   "ret": "fail",   "msg": "failed info",   "data": null,   "i18nArgs": null } </pre>
<b>Response Result Description</b>	<p>data: transaction array list;          Id: event id          eventNo: event number          eventName: event name          eventTime: event time          eventLevel: 0 normal 1 warning 2 alarm</p>

### 2.18.7 Get Doors transaction details [transaction/getDoorTransactionDetail]

<b>Post Request URL</b>	<b>http://serverIP:serverPort/api/transaction/getDoorTransactionDetail?access_token={apitoken}</b>
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre> {   "id": "402856a7695c716301695c81290d0006", } </pre>
<b>Request Parameter Description</b>	<pre> {   "content":{     id: event ID   } } </pre>
<b>Response Result</b>	{

	<pre> "ret": "ok", "msg": "operation succeed", "data": {   "id": "402856a7694641ff0169468394b2000d",   "eventNo": 8, event id   "eventName": "remote opening", event name   "eventTime": "2019-03-04 10:23:40", event time   "doorName": null, door name   "personPin": null, personnel ID   "personName": null, First Name   "areaName": "Area 1", Area Name   "personLastName": null, Last Name   "devName": "192.168.214.197", Device Name   "cardNo": null, Card Number   "readerName": "others", reader name   "verifyModeName": "others"  Verification mode }, "i18nArgs": null, "success": true } {   "ret": "fail",   "msg": "failed info",   "data": null,   "i18nArgs": null } </pre>
<p><b>Response Result Description</b></p>	<p>data:transaction array list;  id: event id;  eventNo: event number  eventName: event name  eventTime: event time</p>

## 2.19 Elevator Transaction Interface

### 2.19.1 Get Elevator Transaction [eleTransaction/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/eleTransaction/list?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>personPin:personPin, [not required]</p> <p>startDate:startDate, [not required]</p> <p>endDate:endDate, [not required]</p> <p>PageNo: page no, required, the value must be greater than 0;</p> <p>pageSize, the number of each page, required, must be greater than 0, not more than 1000 per page;</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c1778b3016c1a695c860013",       "eventTime": "2017-08-04 15:59:12",       "pin": "",       "name": "",       "lastName": "",       "deptName": "",       "areaName": "area name",       "cardNo": "",       "devSn": "5662012052868",       "verifyModeName": "others",       "eventName": "enable device",       "eventPointName": "",       "readerName": "others",       "accZone": null,     }   ] }</pre>

	<pre> "devName": "192.168.214.15", "logId": 11734 }, { "id": 402856aa6c1778b3016c1a695c860016, "eventTime": "2017-08-04 15:58:49", "pin": "", "name": "", "lastName": "", "deptName": "", "areaName": "areaname", "cardNo": "", "devSn": "5662012052868", "verifyModeName": "others", "eventName": "connect server successfully", "eventPointName": "", "readerName": "others", "devName": "192.168.214.15", "logId": 11733 } ] } </pre>
<p><b>Response Result Description</b></p>	<p>data: transaction array list;  id: 5;  eventTime: record device trigger time;  pin: pin;  name: name;  lastName: person name (English);  deptName: department name;  areaName: area name;  cardNo: card no;  devSn: device sn;  verifyModeName: verification mode name;  eventName: transaction description;  eventPointName: transaction point name;</p>

readerName: reader name;  
devName: device name;  
logId: transaction index value, there will be a value if device support it;

## 2.20 Park Transaction Interface

### 2.20.1 Get Car Entry/Exit record pictures

#### [parkTransaction/getParkRecordPicture]

<b>Post Request URL</b>	http://serverIP:serverPort/api/parkTransaction/getParkRecordPicture?recordId={recordId}&recordType={recordType}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	recordId: Car entry/exit records ID number, [Required]; recordType: Car Entry/Exit park lot, [Required], 1: Entry 2: Exit
<b>Response Result</b>	{ "code": 0, "message": "success", "data": "/9j/4AAQSkZJRgABAQAAQABAAD/2wDFAAMCAgMCAgMDAwMEAwMEBQgFBQQEB="
<b>Response Result Description</b>	data: Picture , base 64;

### 2.20.2 Get Park Check-in Transaction [parkTransaction/listParkRecordin]

<b>Post Request URL</b>	http://serverIP:serverPort/api/parkTransaction/listParkRecordin?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	carNumber:carNumber, [not required] startDate:startDate, [not required] endDate:endDate, [not required]

	<p>PageNo: page no, required, the value must be greater than 0;          pageSize, the number of each page, required, must be greater than 0, not more than 1000 per page;</p>
<p><b>Response Result</b></p>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c1778b3016c1a695c860016",       "parkAreaName": " park area",       "name": " driver name",       "carNumber": "car licence plate number",       "carTypeName": "car type",       "channelName": "channel name",       "channelState": " in and out state",       "eventType": "transaction type",       "checkInTime": "check-in time"     },     {       "id": "402856aa6c1778b3016c1a695c860018",       "parkAreaName": "park area",       "name": " driver name",       "carNumber": "car licence plate number",       "carTypeName": "car type",       "channelName": "channel name",       "channelState": "in and out state",       "eventType": "transactiontype",       "checkInTime": "check-in time"     }   ] }</pre>
<p><b>Response Result Description</b></p>	<p>data: transaction array list;          id: "402856aa6c1778b3016c1a695c860018",          parkAreaName: park area;          name: driver name;</p>



	carNumber: car licence plate number; carTypeName: car type; channelName: channel name; channelState: in and out state; eventType: transaction type; checkInTime: check-in time
--	---

### 2.20.3 Get park Check-Out Transaction[[parkTransaction/listParkRecordout](#)]

<b>Post Request URL</b>	http://serverIP:serverPort/api/parkTransaction/listParkRecordout?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	carNumber:carNumber, [not required] startDate:startDate, [not required] endDate:endDate, [not required] pageNo: page no, required, the value must be greater than 0; pageSize, the number of each page, required, the value must be greater than 0, not more than 1000 per page;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c1778b3016c1a695c860018",       "parkAreaName": "park area",       "name": "driver name",       "carNumber": "car licence plate number",       "carTypeName": " car type",       "channelName": "channel name",       "channelState": "in and out state",       "eventType": "transaction type",       "checkOutTime": "check-out time"     }   ] }</pre>

	<pre> }, {   "id": "402856aa6c1e1def016c1e2073e80001",   "parkAreaName": "park area",   "name": "driver name",   "carNumber": "car licence plate number",   "carTypeName": "car type",   "channelName": "channel name",   "channelState": "in and out state",   "eventType": "transaction type",   "checkOutTime": "check-out time" } ] } </pre>
<p><b>Response Result Description</b></p>	<p>data: transaction array list;  id: "402856aa6c1e1def016c1e2073e80001",  parkAreaName: park area;  name: driver name;  carNumber: car licence plate number;  carTypeName: car type;  channelName: channel name;  channelState: in and out state;  eventType: transaction type;  checkOutTime: check-out time</p>

## 2.21 Att Transaction Interface

### 2.21.1 Get Att Transaction [transaction/listAttTransaction]

<b>Post Request URL</b>	http://serverIP:serverPort/api/transaction/listAttTransaction?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>personPin:personPin, [not required]</p> <p>startDate:startDate, [not required]</p> <p>endDate:endDate, [not required]</p> <p>PageNo: page no, required, the value must be greater than 0;</p> <p>pageSize, the number of each page, required, the value must be greater than 0, not more than 1000 per page;</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3c5063016c3cb5a88f001b",       "eventTime": "2017-08-04 15:59:12",       "pin": "",       "name": "",       "lastName": "",       "deptName": "",       "areaName": "areaname",       "cardNo": "",       "devSn": "5662012052868",       "verifyModeName": "others",       "eventName": "enable device",       "eventPointName": "",       "readerName": "others",       "accZone": null,     }   ] }</pre>

	<pre> "devName": "192.168.214.15", "logId": 11734 }, { "id": "402856aa6c3c5063016c3cb5f7150021", "eventTime": "2017-08-04 15:58:49", "pin": "", "name": "", "lastName": "", "deptName": "", "areaName": "areaname", "cardNo": "", "devSn": "5662012052868", "verifyModeName": "others", "eventName": " connected to the server successfully", "eventPointName": "", "readerName": "others", "accZone": null, "devName": "192.168.214.15", "logId": 11733 } ] } </pre>
<p><b>Response Result Description</b></p>	<p>data: transaction array List;  id: "402856aa6c3c5063016c3cb5cfa2001d";  eventTime: Record device trigger time;  pin: pin;  name: name;  lastName: person name (English);  deptName: department name;  areaName: area name;  cardNo: card no;  devSn: device sn;  verifyModeName: verification mode name;  eventName: transaction description;</p>

eventPointName: transaction point name;  
readerName: reader name;  
accZone: access area name;  
devName: device name;  
logId: transaction index value, there will be a value if device support it;

## 2.22 Att Device Interface

### 2.22.1 Get Att Device [device/getAtt]

<b>Post Request URL</b>	<b>http://serverIP:serverPort/api/device/getAtt?sn={sn}&amp;access_token={apitoken}</b>
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	sn: Device Serial Number,[required]
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "402856aa6c3bbde1016c3bc210d90016",     "sn": "CEXB193160012",     "name": "CEXB193160012",     "type": "ZK-S1007",     "status": "0",     "module": "att"   } }</pre>
<b>Response Result Description</b>	<p>data: Device object;</p> <p>id: Device id;</p> <p>sn: Device Serial Number;</p> <p>name: Device Name;</p> <p>type: Device Model Name;</p> <p>statu: Device Status, 1 Enable, 0 Disabled;</p> <p>module: Module, acc AccessControl, att Attendance, ele Elevator;</p>

## 2.23 Apply for abnormal attendance

### 2.23.1 Apply for Re-clock-in

<b>Post Request URL</b>	http://serverIP:serverPort/api/attApply/applySign?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "personPin": 1, "remark": "api", "signDatetime": "2021-11-19 18:00:00" }
<b>Request Parameter Description</b>	personPin (string): personal number remark (string): remarks signDatetime(string): Re-clock-in time
<b>Response Result</b>	{ "code":0, "message":"success", "data":null }
<b>Response Result Description</b>	

### 2.23.2 Apply for overtime

<b>Post Request URL</b>	http://serverIP:serverPort/api/attApply/applyOvertime?access_token={apitoken}
-------------------------	---

<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "endDatetime": "2021-11-19 18:00:00",   "personPin": 1,   "remark": "api",   "startDatetime": "2021-11-19 13:00:00" }</pre>
<b>Request Parameter Description</b>	<p>endDatetime (string): End Time</p> <p>personPin (string): Personal Number</p> <p>remark (string): Remarks</p> <p>startDatetime (string): start time</p>
<b>Response Result</b>	<pre>{   "code":0,   "message":"success",   "data":null }</pre>
<b>Response Result Description</b>	

### 2.23.3 Ask for leave

<b>Post Request URL</b>	http://serverIP:serverPort/api/attApply/applyLeave?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "endDatetime": "2021-11-19 18:00:00",</pre>



	<pre>"leavelImagePath": "string", "leaveTypeNo": "L1", "personPin": 1, "remark": "api", "startDatetime": "2021-11-19 13:00:00" }</pre>
<p><b>Request Parameter Description</b></p>	<p>endDatetime (string):End time</p> <p>leavelImagePath (string): Leave photo path (For details/api/attApply/uploadLeavelImage)</p> <p>leaveTypeNo (string): Leave type number(For details/api/attApply/getLeaveTypeList)</p> <p>personPin (string): Personal number</p> <p>remark (string): Remark</p> <p>startDatetime (string): Start time</p>
<p><b>Response Result</b></p>	<pre>{ "code":0, "message":"success", "data":null }</pre>
<p><b>Response Result Description</b></p>	

## 2.23.4 Get leave type list

<b>Post Request URL</b>	http://serverIP:serverPort/api/attApply/getLeaveTypeList?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	
<b>Request Parameter Description</b>	
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [{     "leaveTypeNo": "L1",     "leaveTypeName": "personal leave"   },   {     "leaveTypeNo": "L2",     "leaveTypeName": "marriage leave"   }   ] }</pre>
<b>Response Result Description</b>	<p>leaveTypeNo: Leave type number</p> <p>leaveTypeName: Leave type name</p>

### 2.23.5 Upload leave picture

<b>Post Request URL</b>	http://serverIP:serverPort/api/attApply/uploadLeaveImage?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "base64Image": "base64Image" }
<b>Request Parameter Description</b>	base64Image (string): picture Base64code
<b>Response Result</b>	{ "code":0, "message":"success", "data":"/fileresource/att/leave/image/2021-12-03/1638503394663.jpg" }
<b>Response Result Description</b>	data: picture path

### 2.23.6 Calculate the total duration of leave

<b>Post Request URL</b>	http://serverIP:serverPort/api/attApply/getApplyTimeLong?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "endDatetime": "2021-11-19 18:00:00", }

	<pre> "leaveImagePath": "string",  "leaveTypeNo": "L1",  "personPin": 1,  "remark": "api",  "startDatetime": "2021-11-19 13:00:00"  } </pre>
<p><b>Request Parameter Description</b></p>	<p>endDatetime (string): End time</p> <p>leaveImagePath (string, optional): Leave photo path (For details/api/attApply/uploadLeaveImage)</p> <p>leaveTypeNo (string): Leave type number (For details/api/attApply/getLeaveTypeList)</p> <p>personPin (string): personal number</p> <p>remark (string,): Remark</p> <p>startDatetime (string): Start time</p>
<p><b>Response Result</b></p>	<pre> {"code": 0, "message": "success", "data": null } </pre>
<p><b>Response Result Description</b></p>	

## 2.24 Edit Attendance personnel information

### 2.24.5 Edit Personal information

<b>Post Request URL</b>	http://serverIP:serverPort /api/attPerson/editAttPerson?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "isAttendance": false,   "perDevAuth": 0,   "personPin": 1,   "verifyMode": 0 }</pre>
<b>Request Parameter Description</b>	<p>isAttendance (boolean): Attendance or not (true is Yes、 false is Not)</p> <p>perDevAuth (integer): Device operation authority (0 is ordinary person、 2 is registrar、 6 is administrators 、 14 is super administrators)</p> <p>personPin (string):personal number</p> <p>verifyMode (integer): Verification method (0=Automatic recognition、 1=Fingerprint only 、 2=User ID only 、 3=Password only 、 4=Card only 、 5=Fingerprint or password、 6=Fingerprint or card、 7=Card or</p>

	<p>password、 8=User ID and fingerprint、 9=Fingerprint and password、 10=Card and fingerprint、 11=Card and password、 12=Fingerprint and password and card、 13=User ID and fingerprint and password、 14=(User ID and fingerprint)or(card and fingerprint)、 15=Face、 16=Face and fingerprint、 17=Face and password、 18=Face and card、 19=Face and fingerprint and password、 20=Face and fingerprint and password、 21=Finger vein、 22=Finger vein and password、 23=Finger vein and card、 24=Finger vein and password and card、 25=Palm prints、 26=Palm prints and card、 27=Palm prints and face、 28=Palm prints and fingerprint、 29=Palm prints and fingerprint and face)</p>
<p><b>Response Result</b></p>	<p>{ "code": 0, "message": "success", "data": null }</p>
<p><b>Response Result Description</b></p>	

## 2.25 Person BioTemplate Interface

### 2.25.1 Add Personnel BioTemplate

<b>Post Request URL</b>	http://serverIP:serverPort/api/bioTemplate/add?access_token={apitoken}
<b>Request Mode</b>	Post
<b>Request Content</b>	<pre>{   "personId": "666",   "validType": "1",   "template": "fsfsfsfsfsfsfsfsfsfsfsfsfsfs",   "templateNo": "3" }</pre>
<b>Request Parameter Description</b>	<p>personId: personnel ID, required;          validType: 1:common, 3:duress, required;          Template: personnel fingerprint template, required;          templateNo: finger number, required,range[0-9];</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": null }</pre>
<b>Response Result Description</b>	public response parameter description

### 2.25.2 Delete By PersonId And TemplateNo

<b>Post Request URL</b>	http://serverIP:serverPort/api/bioTemplate/getFgListByPin/{pin}?access_token={apitoken}
<b>Request Mode</b>	Delete
<b>Request Content</b>	none

<b>Request Parameter Description</b>	pin: personnel ID, required; templateNo: finger number, required, range[0-9];
<b>Response Result</b>	{ "code": 0, "message": "success", "data": null }
<b>Response Result Description</b>	public response parameter description

### 2.25.3 Delete Personnel BioTemplate By PersonId

<b>Post Request URL</b>	http://serverIP:serverPort/api/bioTemplate/deleteById/*?access_token={apitoken}&pin={personnelID}
<b>Request Mode</b>	Delete
<b>Request Content</b>	none
<b>Request Parameter Description</b>	pin: personnel ID, required;
<b>Response Result</b>	{ "code": 0, "message": "success", "data": null }
<b>Response Result Description</b>	public response parameter description

### 2.25.4 Get Personnel BioTemplate By PersonId

<b>Post Request URL</b>	<u>http://serverIP:serverPort/api/bioTemplate/getFgListByPin/{pin}?access_token={apitoken}</u>
-------------------------	--



<b>Request Mode</b>	Get
<b>Request Content</b>	none
<b>Request Parameter Description</b>	pin: personnel ID, required;
<b>Response Result</b>	<pre> {   "code": 0,   "message": "success",   "data": [     {       "id": 2,       "personId": "666",       "validType": 1,       "bioType": 1,       "version": "10",       "template": "fsfsfs",       "templateNo": "3",       "templateNoIndex": 0     },     {       "id": 3,       "personId": "666",       "validType": 1,       "bioType": 1,       "version": "10",       "template": "fsfsfstt",       "templateNo": "5",       "templateNoIndex": 0     }   ] } </pre>

	}
<b>Response Result Description</b>	id: dataBase table personnel fingerprint template id; personId: personnel ID; validType: 1:common, 3:duress; bioType: 1:common, 2:fingerprint; version: fingerprint version template: personnel fingerprint template; templateNo: finger number; templateNoIndex: fingerprintNo index;

## 2.26 Temperature transactions Interface

### 2.26.1 Get Hep Transaction [api/hep/transaction/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/hep/transaction/list?personPin={personPin}&startDate={startDate}&endDate={endDate}&pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>personPin: Pin (Not required)</p> <p>startDate: StartTime (Not required, query by time, work with endtime)</p> <p>endDate: EndTime (Not required, query by time, work with starttime)</p> <p>PageNo: page no, required, the value must be greater than 0;</p> <p>pageSize: the number of each page, required, the value must be greater than 0, not more than 1000 per page;</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402855a6723b91d101723b9e3edf0004",       "eventTime": "2020-05-22 17:01:50",       "pin": "0",       "name": "UnregisterUser",       "lastName": "",       "deptName": "NULL",       "areaName": "Area Name",       "cardNo": null,       "devSn": "CKJF201260077",       "verifyModeName": null,       "eventName": "acc_newEventNo_68",     }   ] }</pre>

	<pre> "eventPointName": "", "readerName": null, "accZone": null, "devName": "192.168.213.160", "logId": null, "eventNumber": "511", "photoPath": "/upload/event/photo/CKJF201260077/20200522170150.jpg", "maskFlag": "1", "temperature": 36.56, "status": "Normal" }, } </pre>
<p><b>Response Result Description</b></p>	<p>data: transaction array list;</p> <p>id: "402856aa6c3c5063016c3cb5cfa2001d"; record id</p> <p>eventTime: Record device trigger time;</p> <p>pin: pin;</p> <p>name: name;</p> <p>lastName: person name (English);</p> <p>deptName: department name;</p> <p>areaName: area name;</p> <p>cardNo: card no;</p> <p>devSn: device sn;</p> <p>verifyModeName: verification mode name;</p> <p>eventName: transaction description;</p> <p>eventPointName: transaction point name;</p> <p>readerName: reader name;</p> <p>accZone: access area name;</p> <p>devName: device name;</p> <p>logId: transaction index value, there will be a value if device support it;</p> <p>eventNumber: event record number</p> <p>photoPath: Instant capture photo storage path</p> <p>maskFlag: weather wear a mask (0: unmask, 1: wear mask, Null:</p>

	Undetected) temperature: temperature status: body temperature status (normal, abnormal, unmeasured)
--	---

## 2.26.2 Get Mask and Temperature Real-Time Monitor events

[api//hep/transaction/monitor]

<b>Post Request URL</b>	http://serverIP:serverPort/api/hep/transaction/monitor?timestamp={timestamp}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Timestamp: Timestamp, millisecond, such as 1494467276539, required; Access frequency cannot be too high, it is recommended to obtain at least 10 seconds or more; there is no need to change after timestamp value set the first request, it is allowed to keep using the first request
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [{     "id": "",     "eventTime": "2020-07-03 16:57:16",     "pin": "1",     "name": "seven",     "lastName": "wu",     "deptName": "Department Name",     "areaName": "Area Name",     "cardNo": null,     "devSn": "CKJF201260077",     "eventName": "Normal Verify Open",     "eventPointName": "192.168.218.202-1",     "devName": "192.168.218.202",     "eventNumber": "315",</pre>

	<pre> "photoPath": "/upload/event/photo/CKJF201260077/20200703165716-1.jpg", "maskFlag": "1", "temperature": 36.6, "status": "Normal"      }}   } </pre>
<p><b>Response Result Description</b></p>	<p>data: transaction array list;</p> <p>id: "empty"; There has empty event id in real time monitor.</p> <p>eventTime: Record device trigger time;</p> <p>pin: pin;</p> <p>name: name;</p> <p>lastName: person family name (English);</p> <p>deptName: department name;</p> <p>areaName: area name;</p> <p>cardNo: card no;</p> <p>devSn: device sn;</p> <p>eventName: transaction description;</p> <p>eventPointName: transaction point name;</p> <p>devName: Device name;</p> <p>eventNumber: event record number;</p> <p>photoPath: Instant capture photo storage path;</p> <p>maskFlag: weather wear a mask (0: unmask, 1: wear mask, Null: Undetected)</p> <p>temperature: temperature</p> <p>status: body temperature status (normal, abnormal, unmeasured)</p>

## 2.27 Edit FaceKiosk Personal information

### 2.27.1 Edit Personal information

<b>Post Request URL</b>	http://serverIP:serverPort /api/attPerson/editInsPerson?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	{ "category": 0, "perDevAuth": 0, "personPin": 1 }
<b>Request Parameter Description</b>	category (string): ID category (0=Ordinary staff、1VIP、=Blacklist) perDevAuth (integer): Device operation authority (0=Ordinary staff,14=super administrators ) personPin (string): Personal number
<b>Response Result</b>	{ "code": 0, "message": "success", "data": null }

## 2.28 FaceKiosk Area Personal

### 2.28.1 Delete Area Personnel

<b>Post Request URL</b>	http://serverIP:serverPort/api/insAreaPerson/delAreaPerson?access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "code": 1,   "pins": "1,2,3" }</pre>
<b>Request Parameter Description</b>	<p>code(string): Area code (For details/api/insAreaPerson/getAreaList)</p> <p>pins (string): Personnel number collection(Multiple personnel are separated by ',')</p>
<b>Response Result</b>	<pre>{   "code":0,   "message":"success",   "data": null }</pre>
<b>Response Result Description</b>	

### 2.28.2 Add Area Personnel

<b>Post Request URL</b>	http://serverIP:serverPort/api/insAreaPerson/setAreaPerson?access_token={apitoken}
-------------------------	--



<b>Request Mode</b>	POST
<b>Request Content</b>	<pre>{   "code": 1,   "pins": "1,2,3" }</pre>
<b>Request Parameter Description</b>	<p>code (string): Area code(For details/api/insAreaPerson/getAreaList)</p> <p>pins (string): personnel number collection(Multiple personnel are separated by ',')</p>
<b>Response Result</b>	<pre>{   "code":0,   "message":"success",   "data": null }</pre>
<b>Response Result Description</b>	

### 2.28.3 Get a list of attendance zones

<b>Post Request URL</b>	http://serverIP:serverPort/api/insAreaPerson/getAreaList?pageNo=1&pageSize=10&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	
<b>Request Parameter Description</b>	
<b>Response Result</b>	{

	<pre> "code":0, "message":"success", "data": [   { "id": "402855b27d4b9eb4017d4ba051690002",     "code": "1",     "name": "Area name"   } ] </pre>
<b>Response Result Description</b>	id: Area ID code: Area code name: Area name

## 2.29 FaceKiosk Device Interface

### 2.29.1 Get Facekiosk device list [api/device/attAdDeviceList]

<b>Post Request URL</b>	http://serverIP:serverPort/api/device/attAdDeviceList?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pageNo: pageno, required, the value must be greater than 0; pageSize: the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402848327562dedd01756d555c726227",       "sn": "6271203100008",       "name": "FaceKiosk-1",       "type": "Robot-Pad",       "status": "1",       "module": "ins"     },     {       "id": "4028483274b4eeb40174d3e117bf4329",       "sn": "174559110004",       "name": "174559110004",       "type": "ZK-S1050",       "status": "1",       "module": "ins"     }   ] }</pre>

<b>Response Result Description</b>	Data: device array list; Id: device id; Sn: device sn; Name: device name; state: device state, 1 enabled, 0 disabled; Module: module, acc- Access, att-Attendance,ele- Elevator, ins-FaceKiosk;
------------------------------------	--

## 2.30 Entrance Control Device Interface

### 2.30.1 Get entrance control device list [api/psgdDvice/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgdDevice/list?pageNo=1&page Size=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	pageNo: pageno, required, the value must be greater than 0; pageSize: the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "8a888e8f820060eb018200e487a50a67",       "sn": "0564140100195",       "name": "192.168.218.104",       "type": "PGIC10",       "state": "1",       "module": "psg",       "gateType": "0"     }   ] }</pre>
<b>Response Result Description</b>	Data: device array list; Id: device id; Sn: device sn; Name: device name; state: device state, 1 enabled, 0 disabled; Module: module, access, attendance, elevator;

	Type: device type name; gateType: Entrance Control device type, 0-unknow(device is not connected to entrance control terminals); 1-Tripod Turnstile; 2-Flap Barrier; 3-Swing Barrier
--	---

### 2.30.2 Get entrance control device info by SN [/api/psgDevice/getBySn]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgDevice/getBySn?sn={sn}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Sn:device sn, required;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "8a888e8f820060eb018200e487a50a67",     "sn": "CKN4213360081",     "name": "10.8.14.234",     "type": "InBio-PGates",     "status": "1",     "module": "psg",     "gateType": "0"   } }</pre>
<b>Response Result Description</b>	Data: device object; Id: device id; Sn: device sn; Name: device name; Type: device model type; state: device state, 1 enabled, 0 disabled;

Module: module, psg: entrance control module;  
 gateType: Entrance Control device type, 0-unknow(device is not connected to entrance control terminals); 1-Tripod Turnstile; 2-Flap Barrier; 3-Swing Barrier

## 2.31 Entrance Control Gate Interface

### 2.31.1 Get the State of All gates [psgGate/allGateState]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgGate/allGateState?access_token={a pitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	None
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "8a888e8f820060eb018200e488650abb",       "name": "10.8.14.234-1",       "deviceId": "8a888e8f820060eb018200e487a50a67",       "connect": "1",       "gateState": "0",       "alarm": "0",       "relay": "0"     }   ] }</pre>
<b>Response Result Description</b>	Data: entrance control gate state array List; Id: gate id; Name: entrance control device name;

	<p>DeviceId: device id;</p> <p>Connection: connection state, 0 offline, 1 online, 2 disabled;</p> <p>gateState: gate state (0 offline-device is not connect to entrance control gate, default:0, 1 Online, 2 Disabled, 3 Lockdown)</p> <p>Alarm: alarm description (0-none, 1-be opened accidentally, 2-tampering, 3-duress password opening,4- duress fingerprint opening, 5-duress opening,6-Infraed ray exception, 7=gate error, None-unknown)</p> <p>relay: Relay state (0-off, 1-Exit open,2-Entry Open, None-unknown)</p>
--	---

### 2.31.2 Get Entrance Control Gate State by Gate Id

[psgGate/gateStateById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgGate/gateStateById?gateId={gateId}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	gateId:Entrance Control gate id, example: "8a888e8f820060eb018200e488650abb"
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "8a888e8f820060eb018200e488650abb",       "name": "10.8.14.234-1",       "deviceId": "8a888e8f820060eb018200e487a50a67",       "connect": "1",       "gateState": "0",       "alarm": "0",       "relay": "0"     }   ] }</pre>



	<pre>] }</pre>
<p><b>Response Result Description</b></p>	<p>Data: entrance control gate state array List;          Id: gate id;          Name: entrance control device name;          DeviceId: device id;          Connection: connection state, 0 offline, 1online, 2 disabled;          gateState: gate state (0 offline-device is not connect to entrance control gate, defaul:0, 1 Online, 2 Disabled, 3 Lockdown)          Alarm: alarm description (0-none, 1-be opened accidentally, 2-tampering, 3-duress password opening,4- duress fingerprint opening, 5-duress opening,6-Infraed ray exception, 7=gate error, None-unknown)          relay: Relay state (0-off, 1-Exit open,2-Entry Open, None-unknown)</p>

### 2.31.3 Get Entrance Control Gate State by the SN of Device

[psgGate/gateStateBySn]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgGate/gateStateBySn?deviceSn={sn}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Sn: device sn
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "8a888e8f820060eb018200e488650abb",       "name": "10.8.14.234-1",       "deviceId":       "8a888e8f820060eb018200e487a50a67",       "connect": "0",       "gateState": "0",       "alarm": "0",       "relay": "0"     }   ] }</pre>
<b>Response Result Description</b>	<p>Data: entrance control gate state array List;</p> <p>Id: gate id;</p> <p>Name: entrance control device name;</p> <p>DeviceId: device id;</p> <p>Connection: connection state, 0 offline, 1online, 2 disabled;</p> <p>gateState: gate state (0 offline-device is not connect to entrance control gate, default:0, 1 Online, 2 Disabled, 3 Lockdown)</p> <p>Alarm: alarm description (0-none, 1-be opened accidentally, 2-tampering, 3-duress password opening,4- duress fingerprint)</p>

	opening, 5-duress opening,6-Infraed ray exception, 7=gate error, None-unknown) relay: Relay state (0-off, 1-Exit open,2-Entry Open, None-unknown)
--	--

### 2.31.4 Get Entrance Control Gate Info by gate Id [psgGate/getById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgGate/getById?id={id}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	id: gate id;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "8a888e8f820060eb018200e488650abb",     "name": "10.8.14.234-1",     "deviceId": "8a888e8f820060eb018200e487a50a67"   } }</pre>
<b>Response Result Description</b>	data: gate info object; id: gate id; name: gate name; deviceId: entrance control device id;

### 2.31.5 Get Entrance Control Gate List [door/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgGate/list?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "8a888e8f820060eb018200e488650abb",       "name": "10.8.14.234-1",       "deviceId": "8a888e8f820060eb018200e487a50a67"     }   ] }</pre>
<b>Response Result Description</b>	data: entrance control gate array list; id: gate id; name: gate name; deviceId: device id;

### 2.31.6 Remote Open by Gate Id [psgGate/remoteOpenById]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgGate/remoteOpenById?gateId={gateId}&openType={openType}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter</b>	gateId: Gate id, required; Example:

<b>Description</b>	"8a888e8f820060eb018200e488650abb" openType: Open gate type, 1-Open(Entry), 2-Open(Exit)
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

### 2.31.7 Remote Open by Gate Name [psgGate/remoteOpenByName]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgGate/remoteOpenByName?gateName={gateName}&openType={openType}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	gateName: entrance control gate name, required; openType: Open gate type, 1-Open(Entry), 2-Open(Exit)
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	public response parameter description

## 2.32 Entrance Control Gate Interface

### 2.32.1 Delete Person Level [psgLevel/deleteLevel]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgLevel/deleteLevel?pin={pin}&levelIds={levelIds}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Pin: pin, required Level Ids: level group id List, multiple comma separated, required, string; such as 402856aa6c3bbde1016c3bc2329200ff,402856aa6c3bbde1016c3bc2

	32930100
<b>Response Result</b>	Refer to public response results
<b>Response Result Description</b>	Public response parameter description

### 2.32.2 Get Gate Control Level by Id [psgLevel/getById/{id}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgLevel/getById/{id}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Id: level group id, required, string; such as "402856aa6c3bbde1016c3bc2329200ff"
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "402856aa6c3bbde1016c3bc2329200ff",     "name": "public level group"   } }</pre>
<b>Response Result Description</b>	Data: level group object; Id: level group id; Name: level group name;

### 2.32.3 Get Entrance Access Level by Name [psgLevel/getByName/{name}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgLevel/getByName/{name}?access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None

<b>Request Parameter Description</b>	Name: level group name, required;
<b>Response Result</b>	<pre> { "code": 0, "message": "success", "data": { "id": "402856aa6c3bbde1016c3bc2329200ff", "name": "public level group" } } </pre>
<b>Response Result Description</b>	Data: level group object; Id: level group id, string; such as '402856aa6c3bbde1016c3bc2329200ff' Name: level group name;

### 2.32.4 Get Entrance Control Access Level [/psgLevel/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgLevel/list?pageNo={pageNo}&pageSize={pageSize}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page no, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "402856aa6c3bbde1016c3bc2329200ff",       "name": "public level group"     }   ] }</pre>
<b>Response Result Description</b>	Data: level group array list; Id: level group id, string; such as "402856aa6c3bbde1016c3bc2329200ff" Name: level group name;

### 2.32.5 Sync Level Info to Device [psgLevel/syncLevel]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgLevel/syncLevel?levelId={levelId}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Level Id: level group id, required, string; such as "402856aa6c3bbde1016c3bc2329200ff"



<b>Response Result</b>	Refer to public response result, for any other result please refer to Appendix- <a href="#">Error code</a>
<b>Response Result Description</b>	Public response parameter description

### 2.32.6 Sync Person Level [/psgLevel/syncPerson]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgLevel/syncPerson?pin={pin}&levelIds={levelIds}&access_token={apitoken}
<b>Request Mode</b>	POST
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Level Ids: level group id list, multiple comma separated, required, string; such as "402856aa6c3bbde1016c3bc2329200ff" Pin: pin, required
<b>Response Result</b>	Refer to public response result, other can refer to Appendix-Error code
<b>Response Result Description</b>	Public response parameter description

## 2.33 Entrance Control Gate Reader Interface

### 2.33.1 Get Entrance Control Gate Reader Info List[psgReader/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgReader/list?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	PageNo: page, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
<b>Response Result</b>	{

	<pre> "code": 0, "message": "success", "data": [   {     "id": "8a888e8f820060eb018200e488660abc",     "gateId": "8a888e8f820060eb018200e488650abb",     "name": "10.8.14.234-1-入",     "readerNo": 1,     "readerState": 0   },   {     "id": "8a888e8f820060eb018200e488660abd",     "gateId": "8a888e8f820060eb018200e488650abb",     "name": "10.8.14.234-1-出",     "readerNo": 2,     "readerState": 1   } ] </pre>
<p><b>Response Result Description</b></p>	<p>Data: reader array List;  Id: reader id;  gateId: entrance control gate id;  Name: reader name;  ReaderNo: readerNo;  ReaderState: reader state, 0: in, 1: out;</p>

### 2.33.2 Get Entrance Control Gate Reader Info by Reader Id

[psgReader/getById]

<p><b>Post Request URL</b></p>	<p>http://serverIP:serverPort/api/psgReader/getById?id={id}&amp;access_to ken={apitoken}</p>
<p><b>Request Mode</b></p>	<p>GET</p>

<b>Request Content</b>	None
<b>Request Parameter Description</b>	id: reader id, required;
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": {     "id": "402856aa6c3c5063016c3cb3a0360005",     "gateId": "402856aa6bff4b1a016bff5374800c0e",     "name": "192.168.214.15-1-in",     "readerNo": 1,     "readerState": 0   } }</pre>
<b>Response Result Description</b>	data: reader info object; id: reader id; Gate Id: gate id; name: reader name; readerNo: reader no; readState: reader state, 0: in, 1:out;

## 2.34 Entrance Control Device transaction Interface

### 2.34.1 Get Entrance Control Transaction by Device Sn

[psgTransaction/device/{deviceSn}]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgTransaction/device/{deviceSn}?pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter</b>	pageNo: pageNo, required, the value must be greater than 0; pageSize: required, the value must be greater than 0, not more

<b>Description</b>	<p>than 1000 per page;  DeviceSn: device sn;  startDate: stat time, not required;  endDate: end time, not required;</p>
<b>Response Result</b>	<pre> {   "code": 0,   "message": "success",   "data": [     {       "id": "8a888e8f821044f3018210ce84150029",       "eventTime": "2022-07-18 18:15:42",       "pin": "",       "name": "",       "lastName": "",       "deptName": "",       "areaName": "Area Name",       "cardNo": "",       "devSn": "CKN4213360081",       "verifyModeName": "Others",       "eventName": "The entrance control gate is not connected",       "eventPointName": "10.8.14.234-1",       "readerName": "Others",       "devName": "10.8.14.234",       "logId": 3603,       "eventNo": 110,       "gateType": 0     }   ] } </pre>
<b>Response Result Description</b>	<p>data: transaction array list;  id: event id, "402856aa6c032ca3016c03351d2600fd";  eventTime: record device trigger time;  pin: pin;  name: person name;</p>

	<p>lastName: person last name, (English);</p> <p>deptName: department name;</p> <p>areaName: area name;</p> <p>cardNo: card no;</p> <p>devSn: device sn;</p> <p>verifyModeName: verification mode name;</p> <p>eventName: transaction description;</p> <p>eventPointName: transaction point name;</p> <p>readerName: reader name;</p> <p>devName: device name;</p> <p>logId: transaction index value, there will be a value if device support it;</p> <p>eventNo: Event Number;</p> <p>gateType: Gate Type, 0-Unknown(Device is not connected to Gate);1-Tripod Turnstyle; 2-Flap Barrier;3-Swing Barrier</p>
--	---

### 2.34.2 Get Entrance Control Gate Transaction List[psgTransaction/list]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgTransaction/list?personPin={personPin}&startDate={startDate}&endDate={endDate}&pageNo=1&pageSize=20&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>personPin: personPin, [not required]</p> <p>startDate: startDate, [not required]</p> <p>endDate: endDate, [not required]</p> <p>PageNo: page no, required, the value must be greater than 0;</p> <p>pageSize, the number of each page, required, must be greater than 0, not more than 1000 per page;</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {</pre>

	<pre>         "id": "8a888e8f821044f3018210ce84150029",         "eventTime": "2022-07-18 18:15:42",         "pin": "",         "name": "",         "lastName": "",         "deptName": "",         "areaName": "Area Name",         "cardNo": "",         "devSn": "CKN4213360081",         "verifyModeName": "Other",         "eventName": "The Gate is not connected",         "eventPointName": "10.8.14.234-1",         "readerName": "Others",         "devName": "10.8.14.234",         "logId": 3603,         "eventNo": 110,         "gateType": 0       }     ]   } </pre>
<p><b>Response Result Description</b></p>	<p>data: transaction array list;</p> <p>id:5;</p> <p>eventTime: record device trigger time;</p> <p>pin: pin;</p> <p>name: person name;</p> <p>lastName: last name;</p> <p>deptName: department name;</p> <p>areaName: area name;</p> <p>cardNo: card no;</p> <p>devSn: device sn;</p> <p>verifyModeName: verification mode name;</p> <p>eventName: transaction description;</p> <p>eventPointName: transaction point name;</p> <p>readerName: reader name;</p> <p>devName: device name;</p>

	<p>logId: transactionindex value, there will be a value if device support it;</p> <p>eventNo: Event Number;</p> <p>gateType: Gate Type, 0-Unknown(Device is not connected to Gate);1-Tripod Turnstyle; 2-Flap Barrier;3-Swing Barrier</p>
--	---

### 2.34.3 Get Entrance Control Device Real-Time Monitoring

#### Transaction[psgTransaction/monitor]

<b>Post Request URL</b>	http://serverIP:serverPort/api/psgTransaction/monitor?timestamp={timestamp}&access_token={apitoken}
<b>Request Mode</b>	GET
<b>Request Content</b>	None
<b>Request Parameter Description</b>	Timestamp: timestampmillisecond, such as 1494467276539, required; Get frequency cannot be too high, it is recommended to obtain at least 10 seconds or more; there is no need to change after timestamp value set the first request, it is allowed to keep using the first request
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": null,       "eventTime": "2022-07-18 18:13:33",       "pin": "",       "name": "",       "lastName": "",       "deptName": "",       "areaName": "Area Name",       "cardNo": "",       "devSn": "CKN4213360081",       "verifyModeName": "Other",       "eventName": "Gate is not connected",     }   ] }</pre>

	<pre> "eventPointName": "10.8.14.234-1", "readerName": "Other", "devName": "10.8.14.234", "logId": 3601, "eventNo": 110, "gateType": 0     }     ] } </pre>
<p><b>Response Result Description</b></p>	<p>data:transaction monitoring record array List;  Id: event id(Not stored in database yet, so it is null)  eventTime: record device trigger time;  pin:pin;  areaName: area name;  cardNo: card no;  devSn: device sn;  verifyModeName: verification mode name;  eventName: transaction description;  eventPointName: transaction point name;  readerName: reader name;  devName: device name;  logId: transactionindex value, there will be a value if device support it;  eventNo: Event Number;  gateType: Gate Type, 0-Unknown(Device is not connected to Gate);1-Tripod Turnstyle; 2-Flap Barrier;3-Swing Barrier</p>

### 2.34.4 Get Personnel Entrance Control Access Transaction by Pin

[transaction/person/{pin}]

<p><b>Post Request URL</b></p>	<p>http://serverIP:serverPort/api/psgTransaction/person/{pin}?pageNo=1&amp;pageSize=20&amp;access_token={apitoken}</p>
<p><b>Request Mode</b></p>	<p>GET</p>



<b>Request Content</b>	None
<b>Request Parameter Description</b>	<p>PageNo: page no, required, the value must be greater than 0;          pageSize, the number of each page, required, must be greater than 0, not more than 1000 per page;          pin: pin, required;          startDate: startDate, [not required]          endDate: endDate, [not required]</p>
<b>Response Result</b>	<pre>{   "code": 0,   "message": "success",   "data": [     {       "id": "8a888e8f821044f3018210ce84150029",       "eventTime": "2022-07-18 18:15:42",       "pin": "",       "name": "",       "lastName": "",       "deptName": "",       "areaName": "Area Name",       "cardNo": "",       "devSn": "CKN4213360081",       "verifyModeName": "Other",       "eventName": "The Gate is not connected",       "eventPointName": "10.8.14.234-1",       "readerName": "Others",       "devName": "10.8.14.234",       "logId": 3603,       "eventNo": 110,       "gateType": 0     }   ] }</pre>
<b>Response Result Description</b>	<p>data:transaction monitoring record array List;          Id: event id(Not stored in database yet, so it is null)</p>

eventTime: record device trigger time;  
pin:pin;  
areaName: area name;  
cardNo: card no;  
devSn: device sn;  
verifyModeName: verification mode name;  
eventName: transaction description;  
eventPointName: transaction point name;  
readerName: reader name;  
devName: device name;  
logId: transactionindex value, there will be a value if device support it;  
eventNo: Event Number;  
gateType: Gate Type, 0-Unknown(Device is not connected to Gate);1-Tripod Turnstyle; 2-Flap Barrier;3-Swing Barrier

## 3. Appendix

### 3.1 Error Code

-1	<b>Program error</b>
-10	The department number or name cannot be empty
-12	The department name has already existed
-13	The department number does not exist
-14	Cannot be set as parent department
-15	Cannot delete department
-20	Pin number cannot be empty
-22	The person does not exist
-23	The card number has been used
-24	The level group does not exist
-25	No person under the access level
-26	The password has been used
-27	Invalid personnel photo
-29	PIN exception
-31	The personnel fingerprint already existed!
-32	Input wrong validtype!
-33	Fingerprint template index range is from 0-9!
-34	templateNo not existed!
-40	Authorize access failure
-41	Delete access level failure
-42	Personnel ID or access level id is null
-43	Access level id can't be null
-44	Door id can't be null

-45	Door open duration between 1~254 seconds.
-46	Door name can't be null
-47	Door doesn't exist
-48	Device serial number can't be null
-70	Advertisement doesn't exist
-71	Didn't support this format
-80	Area didn't exist
-81	Pins not allowed to be null
-82	Pins length not allowed over 500
-83	Area ID can't be null
-90	PageNo or pageSize cannot be set less than or equal to 0
-91	PageSize is greater than 1000
-92	PageSize and pageNo cannot be empty
-100	Time format error
-101	Start time cannot be greater than end time
-200	SN is required
-201	Floor Id is required
-202	The door opening time is required
-203	The door opening time needs to be greater than 0
-220	Operation type error
-221	Black and white list type error
-222	Time parameter error
-223	Car license plate is incorrect
-240	Host(visitor) pin cannot be empty
-241	Visitor certificate type cannot be empty
-242	Visitor certificate number cannot be empty

-243	Visitor name cannot be empty
-244	You have made a reservation
-245	No need to make a reservation during the visit
-246	Visit reason cannot be empty
-247	Visit date cannot be empty
-248	No eligible visitor reservation information
-249	Date format error
-250	Invalid certificate type
-251	Please fill in the correct national ID number
-252	Please fill in the correct date
-253	The employee does not exist
-254	User in the block list
-255	Registration has been completed. Please do not register again
-256	Visitor unregistered, cannot be sign-out
-257	Visitor information doesn't exist
-258	Visitor Number Count Must bigger than 0
-259	The operation object is not selected
-260	The visited department info doesn't exist
-261	Contains special characters
-262	The photo is not qualified.
-263	Reader ID cannot empty
-264	Did not exist this reader
-265	The Open Gate Type cannot empty
-266	Open Gate Type incorrect