USER MANUAL

ZKBioSecurity 3rd Party API

Version: 1.5

Date: January 2022

Software Version: ZKBioSecurity V5000 4.0.0 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.....I

1. Interface Description.....5

1.1 Authorize API Level.....5

1.2 Interface Parameter.....7

2. Interface Details.....8

2.1 Access Control Interface.....8

2.1.1 Delete Person Level [accLevel/deleteLevel].....8

2.1.2 Get Level by Id [accLevel/get byId{id}].....8

2.1.3 Get Level by Name [accLevel/get ByName{name}].....9

2.1.4 Get Access Level/Llist [accLevel/list].....10

2.1.5 Sync Level Info to Device [accLevel/syncLevel].....10

2.1.6 Sync Person Level [accLevel/syncPerson].....11

2.2 Elevator Control Level Interface.....11

2.2.1 Delete person Level [eleLevel/deleteLevel].....11

2.2.2 Get Ele Level by Id[eleLevel/getById/{id}.....12

2.2.3 Get Ele Level by Name[eleLevel/getByName/{name}.....12

2.2.4 Get Ele Level Group List[[eleLevel/list].....13

2.2.5 Sync Ele Level Group Info to Device[eleLevel/syncLevel].....14

2.2.6 Sync Person Ele Level [eleLevel/syncPerson].....14

2.3 Park Authorization Interface.....15

2.3.1 Authorize Park Black and White List [parkAuthorize/addBlackWhite].....15

2.3.2 Delete Park Black and White List[parkAuthorize/delBlackWhite].....15

2.3.3 Edit park Black and White List[parkAuthorize/editBlackWhite].....16

2.4 Card Interface.....16

2.4.1 Get Card List by Pin [card/getCards/{pin}].....16

2.4.2 Set Person Card Info[card/set].....17

2.5 Department Interface.....18

2.5.1 Add /Edit Department[department/add]....18
2.5.2 Delete Department by Code [department/delete/{code}]....18
2.5.3 Get Department Info by Code [department/get/{code}]....19
2.6 Media Interface....20
2.6.1 Add or Update Media Info[adMedia/add]....20
2.6.2 Delete Media by Name[adMedia/delete/{name}]....20
2.6.3 Get Media Object by Name[adMedia/get/{name}]....21
2.7 Area Interface....22
2.7.1 Get Area Object by Area Code[area/get/{code}]....22
2.7.2 Get Area List [area/list]....23
2.8 Person Interface of Att Area....24
2.8.1 Delete Area Person [attAreaPerson/delete]....24
2.8.2 Set Area Person [attAreaPerson/set]....24
2.9 Access Control Device Interface....25
2.9.1 Get Device List [device/accList]....25
2.9.2 Get Device Info by Sn [device/getAcc/{sn}]....26
2.10 Ele Device Interface....27
2.10.1 Get Ele Device List [eleDevice/eleList]....27
2.10.2 Get Ele Device Info by Sn [eleDevice/getEle/{sn}]....28
2.11 Door Interface....29
2.11.1 Get the State of All Doors [door/all DoorState]....29
2.11.2 Get Door State by Door Id [door/doorState ById]....31
2.11.3 Get Door State by the Sn of Device [door/doorStateBy{Sn}]....33
2.11.4 Get Door Info by Door Id [door/get]....34
2.11.5 Get Door List [door/list]....35
2.11.6 Remote Close by Door Id [door/remoteCloseById]....36
2.11.7 Remote Close by Door Name [door/remoteCloseByName]....36
2.11.8 Remote Open by Door Id [door/remoteOpenById]....37
2.11.9 Remote Open by Door Name[door/remoteOpenByName]....37

2.12 Floor interface....38
2.12.1 Get Floor Info List[floor/list]....38
2.12.2 Remote Close by Floor Id[floor/remoteCloseById]....39
2.12.3 Remote Open by Floor Id[floor/remoteOpenById]....40
2.13 Person Interface....40
2.13.1 Add/ Edit Person Info [person/add]....40
2.13.2 Delete Person Info by Pin [person/delete/{pin}]....41
2.13.3 Get person Info by Pin [person/get/{pin}]....43
2.14 Reader Interface....44
2.14.1 Get Reader Info List[reader/accList]....44
2.14.2 Get Reader Info by Reader Id [reader/getAcc]....45
2.15 Visitor Reservation Interface....46
2.15.1 Add Visitor Reservation[visReservation/add]....46
2.15.2 Delete Visitor Reservation[visReservation/del]....47
2.16 Visitor Level Interface....48
2.16.1 Get Visitor Level Group List[visLevel/list]....48
2.17 Visitor Registration Check Out Interface....49
2.17.1 Add Visitor Registration[visRegistration/add]....49
2.17.2 Visitor Check Out [visRegistration/exit]....50
2.18 Access Control transaction Interface....51
2.18.1 Get Transaction by Device Sn [transaction/device/{deviceSn}]....51
2.18.2 Get Access Transaction List[transaction/list]....53
2.18.3 Get Access Real-Time Monitoring Transaction[transaction/monitor]....55
2.18.4 Get Door Transaction by Pin [transaction/person/{pin}]....56
2.18.5 Get Personnel First In and Last Out record by Pin [transaction/firstInAndLastOut/{pin}]58
2.19 Ele Transaction Interface....60
2.19.1 Get Ele Transaction [eleTransaction/list]....60
2.20 Park Transaction Interface....63
2.20.1 Get Park Check-in Transaction [parkTransaction/listParkRecordin]....63

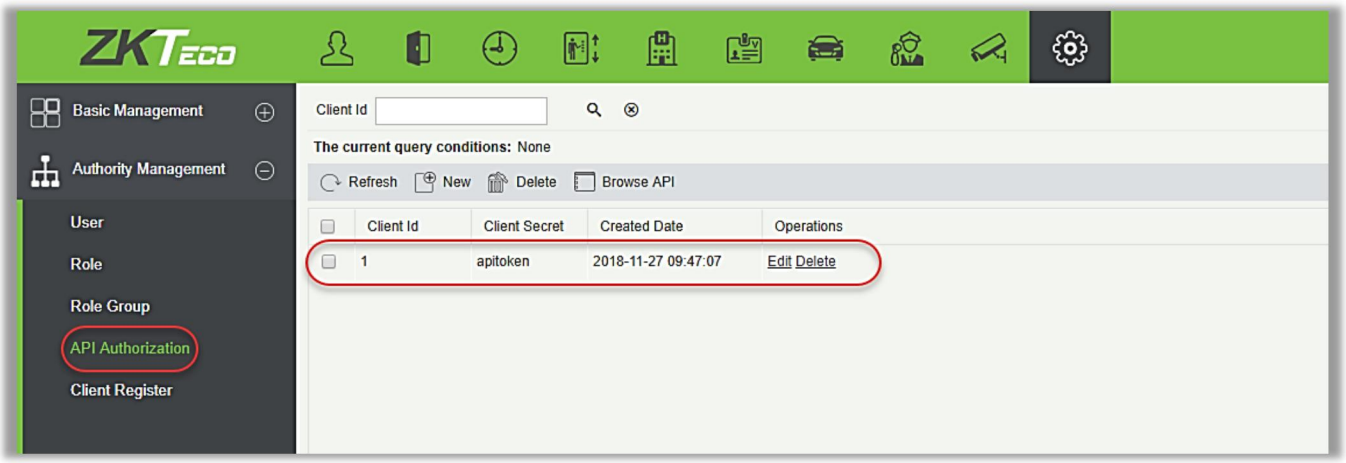
2.20.2 Get park Check-Out Transaction[parkTransaction/listParkRecordout]....64
2.21 Att Transaction Interface....67
2.21.1 Get Att Transaction [transaction/listAttTransaction]....67
2.22 Att Device Interface....70
2.22.1 Get Att Transaction [transaction/listAttTransaction]....70
2.23 Person BioTemplate Interface....71
2.23.1 Add Personnel BioTemplate....71
2.23.2 Delete By PersonId And TemplateNo....71
2.23.3 Delete Personnel BioTemplate By PersonId....72
2.23.4 Get Personnel BioTemplate By PersonId....72
2.24 Temperature transactions Interface....75
2.24.1 Get Hep Transaction [api/hep/transaction/list]....75
2.24.2 Get Mask and Temperature Real-Time Monitor events [api//hep/transaction/monitor]77
2.25 FaceKiosk Device Interface....79
2.25.1 Get Facekiosk device list [api/device/attAdDeviceList]....79
2.26 VMS Face Recognition Report Interface....81
2.26.1 Get VMS face capture records by list [/api/captureRecord/list]....81
3. Appendix....83
3.1 Error Code....83

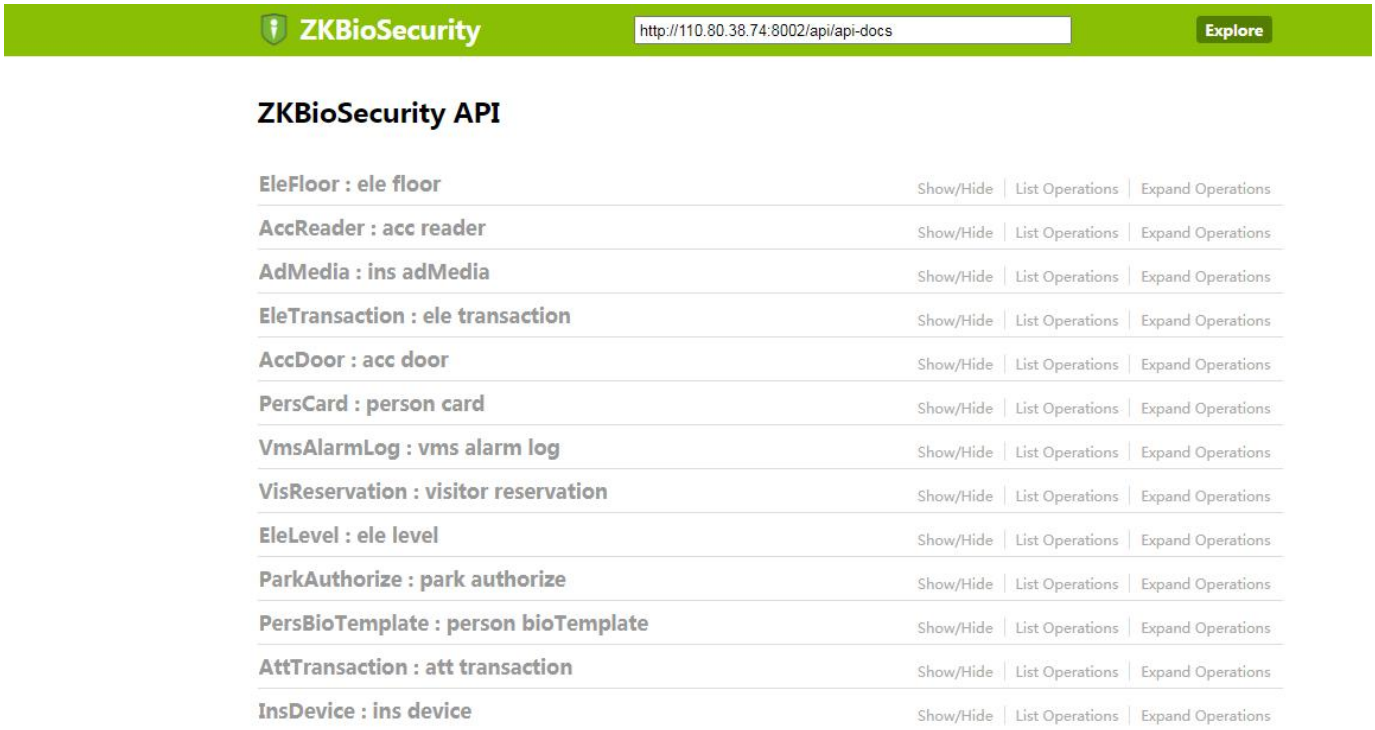
1. Interface Description

ZKBioSecurity 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:





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

serverIP: ZKBioSecurity server or computer IP, such as: 192.168.1.100;

serverPort: The server port of ZKBioSecurity, 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

Public Response Result	{
	"code": 0,
	"message": "string",
	"data": {}
	}

Public Response Result Parameter Description	code: error code, less than 0 means failure, more than 0 means success; msg: Error message data: can refer to each interface description
Basic Parameter Description	Sn: device serial number Pin: pin pageSize: page size; pageNo: page no;

2. Interface Details

2.1 Access Control Interface

2.1.1 Delete Person Level [accLevel/deleteLevel]

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

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

Post Request URL	http://serverIP:serverPort/api/accLevel/getById/{id}?access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	Id: level group id, required, string; such as “402856aa6c3bbde1016c3bc2329200ff”

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": “402856aa6c3bbde1016c3bc2329200ff”,

	"name": "public level group" } }
Response Result Description	Data: level group object; Id: level group id; Name: level group name;

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

Post Request URL	http://serverIP:serverPort/api/accLevel/getByName/{name}?access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	Name: level group name, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": "402856aa6c3bbde1016c3bc2329200ff",
	"name": "public level group"
	}
	}

Response Result Description	Data: level group object; Id: level group id, string; such as '402856aa6c3bbde1016c3bc2329200ff' Name: level group name;

2.1.4 Get Access Level/Llist [accLevel/list]

Post Request URL	http://serverIP:serverPort/api/accLevel/list?pageNo={pageNo}&pageSize={pageSize}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	PageNo: page no, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0

Response Result	{
	"code": 0,
	"message": "success",
	"data": [
	{
	"id": "402856aa6c3bbde1016c3bc2329200ff",
	"name": "public level group"
	}
	]
	}

Response Result Description	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]

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

Description	"402856aa6c3bbde1016c3bc2329200ff"
Response Result	Refer to public response result, for any other result please refer to Appendix-Error code
Response Result Description	Public response parameter description

2.1.6 Sync Person Level [accLevel/syncPerson]

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

2.2 Elevator Control Level Interface

2.2.1 Delete person Level [eleLevel/deleteLevel]

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

Response Result	Refer to public Response Result
Response Result Description	public response parameter description

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

Post Request URL	http://serverIP:serverPort/api/eleLevel/getById/{id}?access_token={apitoken}
Request Mode	get
Request Content	None
Request Parameter Description	id: level group, required, string; such as: “402856aa6c3bbde1016c3bc2329200ff”

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": “402856aa6c3bbde1016c3bc2329200ff”,
	"name": "public level group"
	}
	}

Response Result Description	data: level group object; id: level group id; name: level group name;

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

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

Request Parameter Description	name: level group name, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": “402856aa6c3bbde1016c3bc2329200ff”,
	"name": "public level group"
	}
	}

Response Result Description	data: level group object; id: level group id; name: level group name;

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

Post Request URL	http://serverIP:serverPort/api/eleLevel/list?pageNo={pageNo}&pageSize={pageSize}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	pageNo: page, required, must be greater than 0 pageSize: The number of per page, required, must be greater than 0

Response Result	{
	"code": 0,
	"message": "success",
	"data": [
	{
	"id": “402856aa6c3bbde1016c3bc2329200ff”,
	"name": "public level group"
	}
	]

	}
Response Result Description	data: level group array list; id: level group id; name: level group name;

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

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

2.2.6 Sync Person Ele Level [eleLevel/syncPerson]

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

2.3 Park Authorization Interface

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

Post Request URL	http://serverIP:serverPort/api/parkAuthorize/addBlackWhite?type={type}&carNumber={carNumber}&fromTime={fromTime}&toTime={toTime}&access_token={apitoken}
Request Mode	POST
Request Content	None
Request Parameter Description	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
Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

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

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

Description	

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

Post Request URL	http://serverIP:serverPort/api/parkAuthorize/editBlackWhite?type={type}&carNumber={carNumber}&oldCarNumber={oldCarNumber}&fromTime={fromTime}&toTime={toTime}&access_token={apitoken}
Request Mode	POST
Request Content	None
Request Parameter Description	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)
Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

2.4 Card Interface

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

Post Request URL	http://serverIP:serverPort/api/card/getCards/{pin}?access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	pin: pin, required;
Response Result	{

	"code": 0,
	"message": "success",
	"data": [
	{
	"pin": "1234",
	"cardNo": "111",
	"cardType": "0"
	}
	]
	}

Response Result Description	data: card number List; pin: pin; cardNo: card no; 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

2.4.2 Set Person Card Info[card/set]

Post Request URL	http://serverIP:serverPort/api/card/set?access_token={apitoken}
Request Mode	POST

Request Content	{
	"pin": "1234",
	"cardNo": "212121",
	"cardType": "0"
	}

Request Parameter Description	Pin: pin, required cardNo: card number, required; cardType: card type, 0: master card, 1: sub card, the system does not open multi-card function, sub card setting invalid
Response Result	Refer to public response result, other can refer to Appendix-Error code

Response Result Description	public response parameter description

2.5 Department Interface

2.5.1 Add /Edit Department[department/add]

Post Request URL	http://serverIP:serverPort/api/department/add?access_token={apitoken}
Request Mode	POST

Request Content	{
	"name": "department name",
	"code": "456",
	"parentCode": "123"
	}

Request Parameter Description	Name: department name, required; code: department code, required; parentCode: parent department number;
Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

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

Post Request URL	http://serverIP:serverPort/api/department/delete/{code}?access_token={apitoken}
Request Mode	DELETE
Request Content	None
Request Parameter Description	code: department code, required;

Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

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

Post Request URL	http://serverIP:serverPort/api/department/get/{code}?access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	Code: department code, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"name": "4444",
	"code": "3333",
	"parentCode": "2"
	}
	}

Response Result Description	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]

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

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

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

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

Post Request URL	http://serverIP:serverPort/api/adMedia/get/{name}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	name: media name, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": [
	{
	"id": “402856aa6c3bbde1016c3bc2329200ff”,
	
	"mediaName": "111",
	"mediaSize": "21KB",
	"mediaSuffix": "png",
	"mediaType": "1",
	"mediaPath": ""/att/picture/1535520932078.png""
	}
	]
	}

Response Result Description	Data: media object Id: media id; such as：“402856aa6c3bbde1016c3bc2329200ff” mediaName: media name; mediaSize: media size; mediaType: media type; 1: picture; 2: video mediaPath: media path

2.7 Area Interface

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

Post Request URL	http://serverIP:serverPort/api/attAreaPerson/area/get/{code}?code=1&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	Code: department code, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": 1,
	"code": "402856aa6c3bbde1016c3bc2329200ff ",
	"name": "area name"
	}
	}

Response Result Description	Data: area object; Id: area id; Code: area code; name: area name;

2.7.2 Get Area List [area/list]

Post Request URL	http://serverIP:serverPort/api/attAreaPerson/area/list?pageNo=1&pageSize=5&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	pageNo: page, required, must be greater than 0 pageSize, the number of each page, required, must be greater than 0

Response Result	{
	"code": 0,
	"message": "success",
	"data": [
	{
	"id": "402856aa6c3bbde1016c3bc2329200ff",
	"code": "1",
	"name": "area name"
	},
	{
	"id": "402856aa6c3bbde1016c3bc211e5005a",
	"code": "2",
	"name": "2"
	}
	]
	}

Response Result Description	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]

Post Request URL	http://serverIP:serverPort/api/attAreaPerson/delete?access_token={apitoken}
Request Mode	DELETE

Request Content	{
	"code": "1",
	"pins": "5"
	}

Request Parameter Description	Code: area code, required; Pins: pin, required
Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

2.8.2 Set Area Person [attAreaPerson/set]

Post Request URL	http://serverIP:serverPort/api/attAreaPerson/set?access_token={apitoken}
Request Mode	POST

Request Content	{
	"code": "1",
	"pins": "5"
	}

Request Parameter Description	Code: area code, required; Pins: pin, required
Response Result	Refer to public response result, other can refer to Appendix-Error code

Response Result Description	public response parameter description

2.9 Access Control Device Interface

2.9.1 Get Device List [device/accList]

Post Request URL	http://serverIP:serverPort/api/device/accList?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	pageNo: pageno, required, the value must be greater than0; pageSize: the number of each page, required, must be greater than 0
Response Result	{ "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" }] }
Response Result Description	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;

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

Post Request URL	http://serverIP:serverPort/api/device/getAcc?sn={sn}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	Sn:device sn, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": "402856aa6c3bbde1016c3bc2329200ff",
	"sn": "5662012052868",
	"name": "192.168.214.15",
	"type": "inibo460",
	"state": "0",
	"module": "acc"
	}

	}
Response Result Description	Data: device object; Id: device id; Sn: device sn Name: device name; Type: device model type; state: device state, 1enabled, 0 disabled; Module: module, access, attendance, elevator;

2.10 Ele Device Interface

2.10.1 Get Ele Device List [eleDevice/eleList]

Post Request URL	http://serverIP:serverPort/api/eleDevice/eleList?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	pageNo: page no, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
Response Result	{ "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",
	"name": "192.168.218.11",
	"type": "EC-100",
	"state": "1",
	"module": "ele"
	}
	]
	}

Response Result Description	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;

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

Post Request URL	http://serverIP:serverPort/api/eleDevice/getEle?sn={sn}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	Sn: device sn, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {

	"id": “402856aa6c3bbde1016c3bc2329200ff”, "sn": "5662012052868", "name": "192.168.214.15", "type": "EC-100", "state": "0", "module": "ele" } }
Response Result Description	Data: device object; id: device id; sn: device sn; name: device name; type: device model name; state: state: device state, 1 enabled, 0 disabled; Module: module, access, attendance, elevator;

2.11 Door Interface

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

Post Request URL	http://serverIP:serverPort/api/door/allDoorState?timestamp={timestamp}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	Timestamp: timestamp, millisecond, such as 1494467276539, required; Access frequency cannot be too high, it is recommended to obtain at least 10 seconds or more;
Response Result	{ "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, 1online, 2 disabled), "sensor": sensor state (unknown, no sensor, off, open), "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) "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",
	"connect": "1",
	"sensor": "No sensor",
	"alarm": "None",
	"relay": "off"
	}
	]
	}

Response Result Description	Data: door state array List; Id: door id; Name: door name; DeviceId: device id; Connection: connection state, 0 offline, 1online, 2 disabled; Sensor: sensor description (unknown, no sensor, off, open) 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) relay: Relay state (off, open, unknown)

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

Post Request URL	http://serverIP:serverPort/api/door/doorStateById?doorId=1&timestamp={timestamp}&access_token={apitoken}

Request Mode	GET
Request Content	None
Request Parameter Description	Timestamp: timestamp, millisecond, such as 1494467276539, required; Access frequency cannot be too high, it is recommended to obtain at least 10 seconds or more;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": door id, string; such as: 402856aa6c083827016c08582f8f0012
	"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
	}
	}

Response Result Description	data: door state object id: door id; name: door name; deviceId: device id; connect: connection state (0 offline, 1 online, 2 disabled); sensor: sensor description (unknown, no sensor, off, open); 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) relay: Relay state (off, open, unknown)

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

Post Request URL	http://serverIP:serverPort/api/door/doorStateBySn?deviceSn={sn} &timestamp={timestamp}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	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

Response Result	{
	"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
	}
	}

Response Result Description	data: door state object; id: door id; name: door name; deviceId: device id; 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, 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)

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

Post Request URL	http://serverIP:serverPort/api/door/get?id={id}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	id: door id;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": "402856aa6c083827016c08582f8f0012",
	"name": "192.168.214.15-1",
	"deviceId":"402856aa6c3bbde1016c3bc211e5005a",
	}
	}

Response Result Description	data: door info object; id: doorid; name: door name; deviceId: deviceid;

2.11.5 Get Door List [door/list]

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

	}] }
Response Result Description	data: door array list; id: doorid; name: door name; deviceId: device id;

2.11.6 Remote Close by Door Id [door/remoteCloseById]

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

2.11.7 Remote Close by Door Name [door/remoteCloseByName]

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

	code
Response Result Description	public response parameter description

2.11.8 Remote Open by Door Id [door/remoteOpenById]

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

2.11.9 Remote Open by Door Name[door/remoteOpenByName]

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

2.12 Floor interface

2.12.1 Get Floor Info List[floor/list]

Post Request URL	http://serverIP:serverPort/api/floor/list?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	PageNo: page, required, the value must be greater than 0; pageSize, the number of each page, required, must be greater than 0
Response Result	{ "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”,
	},
	{
	"id": “402856aa6c3baabb016c3bb37365000a”,
	"floorNo":4,
	"name": "192.168.214.15-4",
	"deviceId":”402856aa6c3c5063016c3cb3a0360005”,
	}
	]
	}

Response Result Description	data: floor array list; id: floor id; floorNo: floor no; name: floor name; deviceId: device id;

2.12.2 Remote Close by Floor Id[floor/remoteCloseById]

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

2.12.3 Remote Open by Floor Id[floor/remoteOpenById]

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

2.13 Person Interface

2.13.1 Add/ Edit Person Info [person/add]

Post Request URL	http://serverIP:serverPort/api/person/add?access_token={apitoken}
Request Mode	POST
Request Content	{ "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 " }
Request Parameter Description	Pin: pin, required deptCode: department code; name: name; lastName: person name (English) gender: gender, male, female; cardNo: card no; personPhoto :personnel comparison picture. “picture turn into base64 format” accLevelIds: access level groupid List, multiple comma separated, string; such as : 402856aa6c3c5063016c3cb3a0360005,402856aa6bff4b1a016bff5374800c0e,402856aa6c3baabb016c3bb37365000a; accStartTime: access start time, such as :2017-08-04 15:45:00; accEndTime: access end time, such as :2017-08-04 23:59:59;
Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

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

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

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

Post Request URL	http://serverIP:serverPort/api/person/get/{pin}?access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	pin: pin, required;

Response Result	{
	"code": 0,
	"message": “success”,
	"data": {
	"pin": "pin",
	"deptCode": department code,
	"deptName": "department name",
	"name": "name",
	"lastName": "person name (English)",
	"gender": "gender",
	"cardNo": “card no”
	}
	}

Response Result Description	data: person data info; pin: pin; gender: gender, M: male, F: female; deptCode: department code; deptName: department name; name: name; lastName: person name (English); cardNo: card no;

2.14 Reader Interface

2.14.1 Get Reader Info List[reader/accList]

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

Response Result	{
	"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
	}
	]
	}

Response Result	Data: reader array List;

Description	Id: reader id; DoorId: door id; Name: reader name; ReaderNo: readerNo; ReaderState: reader state, 0: in, 1: out;

2.14.2 Get Reader Info by Reader Id [reader/getAcc]

Post Request URL	http://serverIP:serverPort/api/reader/getAcc?id={id}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	id: reader id, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": “402856aa6c3c5063016c3cb3a0360005”,
	"doorId": “402856aa6bff4b1a016bff5374800c0e”,
	"name": "192.168.214.15-1-in",
	"readerNo": 1,
	"readerState": 0
	}
	}

Response Result Description	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]

Post Request URL	http://serverIP:serverPort/api/visReservation/add?access_token={apitoken}
Request Mode	POST

Request Content	{
	"persPersonPin": "5",
	"certType": "3",
	"certNum": "4512",
	"visitEmpName": "Mary",
	"visitEmpPhone": "13015648651",
	"company": "ZKTeco",
	"visitReason": "visit",
	"visitDate": "2018-08-31"
	}

Request Parameter Description	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, required;

Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result	public response parameter description

Description	

2.15.2 Delete Visitor Reservation[visReservation/del]

Post Request URL	http://serverIP:serverPort/api/visReservation/del?certType={certType}&certNum={certNum}&visitDate={visitDate}&access_token={apitoken}
Request Mode	POST
Request Content	None

Request Parameter Description	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;
	visitDate: visit date, required;
	[The system will find out the only visitor reservation information and delete it according to the certificate type, ID number, and visit date]

Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

2.16 Visitor Level Interface

2.16.1 Get Visitor Level Group List[visLevel/list]

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

Response Result	{
	"code": 0,
	"message": "success",
	"data": [
	{
	"id": “402856aa6bff4b1a016bff5374800c0e”,
	"name": "public level group"
	}
	]
	}

Response Result Description	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]

Post Request URL	http://serverIP:serverPort/api/visRegistration/add?&access_token={apitoken}
Request Mode	POST

Request Content	{
	"persPersonPin": "1",
	"certType": "8",
	"certNum": "456888",
	"visEmpName": "max",
	"visitEmpPhone": "",
	"company": "",
	"visitReason": "visit",
	"visitorCount": "1",
	"startTime": "2018-08-31",
	"endTime": "2018-08-31",
	"cardNo": "",
	"visLevels": ""
	}

Request Parameter Description	"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 startTime: start time, required endTime: end time, required cardNo: card no visLevels: visitor level group id, string; such as: 402856aa6bff4b1a016bff5374800c0e,402856aa6c3c5063016c3cb3a0360005
Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

2.17.2 Visitor Check Out [visRegistration/exit]

Post Request URL	http://serverIP:serverPort/api/visRegistration/exit?certType={certType}&certNum={certNum}&access_token={apitoken}
Request Mode	POST
Request Content	None

Request Parameter Description	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

Response Result	Refer to public response result, other can refer to Appendix-Error code
Response Result Description	public response parameter description

2.18 Access Control transaction Interface

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

Post Request URL	http://serverIP:serverPort/api/transaction/device/{deviceSn}?pageNo=1&pageSize=20&acc_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	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;
Response Result	{ "code": 0, "message": "success", "data": [{ "id": "402856aa6c3c5063016c3cb3a0360005", "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": “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
	}
	]
	}

Response Result Description	data: transaction array list; id: “402856aa6c032ca3016c03351d2600fd”; eventTime: record device trigger time; pin: pin; name: person 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; devName: device name; logId: transaction index value, there will be a value if device support it;

2.18.2 Get Access Transaction List[transaction/list]

Post Request URL	http://serverIP:serverPort/api/transaction/list?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	personPin:personPin, [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, must be greater than 0, not more than 1000 per page;
Response Result	{ "code": 0, "message": "success", "data": [{ "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
	}
	]
	}

Response Result Description	data: transaction array list; id:5; eventTime: record device trigger time; pin: pin; name: person 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;

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

Post Request URL	http://serverIP:serverPort/api/transaction/monitor?timestamp={timestamp}&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	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
Response Result	{ "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)" }] }
Response Result Description	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;

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

Post Request URL	http://serverIP:serverPort/api/transaction/person/{pin}?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	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;
Response Result	{

```
"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",
"cardNo": "",
"devSn": "5662012052868",
"verifyModeName": "others",
"eventName": "connect server successfully",
"eventPointName": "",
"readerName": "others",
```

	"devName": "192.168.214.15", "logId": 11733 }] }
Response Result Description	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;

2.18.5 Get Personnel First In and Last Out record by Pin [transaction/firstInAndLastOut/{pin}]

Post Request URL	**http://serverIP:serverPort/api/transaction/firstInAndLastOut/{pin}?pageNo=1&pageSize=20&access_token={apitoken}**
Request Mode	GET

Request Content	None
Request Parameter Description	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]

Response Result	{
	"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"
	}
	]
	}

Response Result Description	data:transaction array list; 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；

2.19 Ele Transaction Interface

2.19.1 Get Ele Transaction [eleTransaction/list]

Post Request URL	http://serverIP:serverPort/api/eleTransaction/list?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	personPin:personPin, [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, must be greater than 0, not more than 1000 per page;
Response Result	{ "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,
	"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
	}
	]
	}

Response Result Description	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; 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 Park Check-in Transaction [parkTransaction/listParkRecordin]

Post Request URL	http://serverIP:serverPort/api/parkTransaction/listParkRecordin?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	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, must be greater than 0, not more than 1000 per page;
Response Result	{ "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" }] }
Response Result Description	data: transaction array list; 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; checkInTime: check-in time

2.20.2 Get park Check-Out Transaction[parkTransaction/listParkRecordout]

Post Request URL	http://serverIP:serverPort/api/parkTransaction/listParkRecordout ?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	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;

Response Result	{
	"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"
	},
	{
	"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"
	}
	]
	}

Response Result Description	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

2.21 Att Transaction Interface

2.21.1 Get Att Transaction [transaction/listAttTransaction]

Post Request URL	http://serverIP:serverPort/api/transaction/listAttTransaction?pageNo=1&pageSize=20&access_token={apitoken}
Request Mode	GET
Request Content	None
Request Parameter Description	personPin:personPin, [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;
Response Result	{ "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,
	"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
	}
	]
	}

Response Result Description	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; 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 Transaction [transaction/listAttTransaction]

Post Request URL	**http://serverIP:serverPort/api/device/getAtt?sn={sn}&access_token={apitoken}**
Request Mode	GET
Request Content	None
Request Parameter Description	sn: Device Serial Number,[required]

Response Result	{
	"code": 0,
	"message": "success",
	"data": {
	"id": "402856aa6c3bbde1016c3bc210d90016",
	"sn": "CEXB193160012",
	"name": "CEXB193160012",
	"type": "ZK-S1007",
	"status": "0",
	"module": "att"
	}
	}

Response Result Description	data: Device object; id: Device id; sn: Device Serial Number; name: Device Name; type: Device Model Name; statu: Device Status, 1 Enable, 0 Disabled; module: Module, acc AccessControl, att Attendance, ele Elevator;

2.23 Person BioTemplate Interface

2.23.1 Add Personnel BioTemplate

Post Request URL	http://serverIP:serverPort/api/bioTemplate/add?access_token={apitoken}
Request Mode	Post

Request Content	{
	"personId": "666",
	"validType": "1",
	"template": "fsfsfsfsfsfsfsfsfsfsfsfsfs",
	"templateNo": "3"
	}

Request Parameter Description	personId: personnel ID, required; validType: 1:common, 3:duress, required; Template: personnel fingerprint template, required; templateNo: finger number, required,range[0-9];

Response Result	{
	"code": 0,
	"message": "success",
	"data": null
	}

Response Result Description	public response parameter description

2.23.2 Delete By PersonId And TemplateNo

Post Request URL	http://serverIP:serverPort/api/bioTemplate/delete/*?access_token={apitoken}&pin={**personnelID**}&templateNo={**templateNo**}
Request Mode	Delete
Request Content	none

Request Parameter Description	pin: personnel ID, required; templateNo: finger number, required, range[0-9];

Response Result	{
	"code": 0,
	"message": "success",
	"data": null
	}

Response Result Description	public response parameter description

2.23.3 Delete Personnel BioTemplate By PersonId

Post Request URL	http://serverIP:serverPort/api/bioTemplate/deleteById/*?access_token={apitoken}&pin={**personnelID**}
Request Mode	Delete
Request Content	none
Request Parameter Description	pin: personnel ID, required;

Response Result	{
	"code": 0,
	"message": "success",
	"data": null
	}

Response Result Description	public response parameter description

2.23.4 Get Personnel BioTemplate By PersonId

Post Request URL	http://serverIP:serverPort/api/bioTemplate/getFgListById/*?access_token={apitoken}&pin={**personnelID**}
Request Mode	Get

Request Content	none
Request Parameter Description	pin: personnel ID, required;

Response Result	<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>

Response Result Description	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;

2.24 Temperature transactions Interface

2.24.1 Get Hep Transaction [api/hep/transaction/list]

Post Request URL	**http://serverIP:serverPort/api/hep/transaction/list?personPin={personPin}&startDate={startDate}&endDate={endDate}&pageNo=1&pageSize=20**
Request Mode	GET
Request Content	None
Request Parameter Description	personPin: Pin（Not required） startDate: StartTime（Not required, query by time, work with endtime） endDate: EndTime（Not required， query by time, work with starttime） 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;
Response Result	{ "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", "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" }, }
Response Result Description	data： transaction array list; id：”402856aa6c3c5063016c3cb5cfa2001d”； record id 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：transaction index value, there will be a value if device support it; eventNumber：event record number

	photoPath：Instant capture photo storage path maskFlag：weather wear a mask (0: unmasks, 1: wear mask, Null: Undetected) temperature：temperature status：body temperature status（normal，abnormal，unmeasured）

2.24.2 Get Mask and Temperature Real-Time Monitor events [api//hep/transaction/monitor]

Post Request URL	**http://serverIP:serverPort/api/hep/transaction/monitor?timestamp={timestamp}&access_token={apitoken}**
Request Mode	GET
Request Content	None
Request Parameter Description	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

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

2.25 FaceKiosk Device Interface

2.25.1 Get Facekiosk device list [api/device/attAdDeviceList]

Post Request URL	**http://serverIP:serverPort/api/device/attAdDeviceList?pageNo=1&pageSize=20&access_token={apitoken}**
Request Mode	GET
Request Content	None
Request Parameter Description	pageNo: pageno, required, the value must be greater than0; pageSize: the number of each page, required, must be greater than 0

Response Result	<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>

	}
Response Result Description	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;

2.26 VMS Face Recognition Report Interface

2.26.1 Get VMS face capture records by list [/api/captureRecord/list]

Post Request URL	**http://serverIP:serverPort/api/captureRecord/list?includePhoto={includePhoto}&personPin={personPin}&startDate={startDate}&endDate={endDate}&pageNo=1&pageSize=20&access_token={apitoken}**
Request Mode	GET
Request Content	None
Request Parameter Description	includePhoto:Is return the Capture Photo?(true or false) personPin: Pin（Not required） startDate: StartTime（Not required, query by time, work with endtime） endDate: EndTime（Not required， query by time, work with starttime） pageNo: pageno, required, the value must be greater than0; pageSize: the number of each page, required, must be greater than 0,less than 1000.
Response Result	{ "code": 0, "message": "success", "data": [{ "id": "402856f475b5418c0175b6c7102b2208", "captureTime": "2020-11-11 18:08:12", "deviceName": "192.168.214.193", "recordType": "1", "pin": "3403", "name": "Jason", "lastName": "Bill", "capturePhoto": null },

	{
	"id": "402856f475b5418c0175b6c7107b220a",
	"captureTime": "2020-11-11 18:08:12",
	"deviceName": "192.168.214.193",
	"recordType": "1",
	"pin": "3403",
	"name": "Jack",
	"lastName": "White",
	"capturePhoto": null
	}
	]
	}

Response Result Description	Data: events array list; Id: '402856aa6c082c51016c082ef4e70001'; captureTime：record time； deviceName：device name； recordType：record type（-1:Stranger Alarm，0: BlockList Alarm，1:Normal Record）； pin：Pin； name： Name； lastName： Family name; capturePhoto: Capture photo

3. Appendix

3.1 Error Code

-1	**Program error**
-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