Passenger Flow Camera HTTP Protocol Interface Document

Version : V2.1.6

Version Information

|  |  |  |
| --- | --- | --- |
| Version | Date | Description |
| V2.0.0 | 20190814 | first edition |
| V2.0.1 | 20190819 | Supplementary content of log acquisition interface and historical passenger flow data acquisition interface |
| V2.0.2 | 20190820 | The content of the optimized passenger flow data interface is the following interfaces: query, acquisition and deletion |
| V2.0.3 | 20190902 | The deletion interface in the passenger flow data interface has been cancelled.  Modified some nested josn in the setting instructions to prevent difficult processing during encryption. |
| V2.0.4 | 20191105 | Added firmware upgrade status query interface |
| V2.0.5 | 20191121 | Added an interface to clear the database passenger flow data, and restored factory settings to also add this function. |
| V2.0.6 | 20200103 | Added an interface to query and clear the passenger flow data status of the database. At the same time, after the data is cleared, the statistical values of each counting event will be cleared. |
| V.2.0.7 | 20211115 | Added support for strabismus mode parameters |
| V.2.0.8 | 20211117 | Change the Chinese return information of the interface to English return information |
| V.2.0.9 | 20211217 | Corrected rotation angle and installation height restriction instructions |
| V.2.1.0 | 20211228 | Added device installation type acquisition interface, device installation type setting interface, and human body tracking result acquisition interface. |
| V.2.1.1 | 20220406 | Added depth map switch setting interface, depth map switch status acquisition interface, and camera internal parameter acquisition interface. |
| V.2.1.2 | 20220507 | Added client passenger flow statistical result acquisition interface and historical passenger flow data acquisition interface. |
| V.2.1.3 | 20220512 | Added http json parameter setting and obtaining interface, added http xml parameter setting and obtaining interface |
| V.2.1.4 | 20220517 | Added device model acquisition interface and OSD information setting interface |
| V.2.1.5 | 20220602 | Optimize format |
| V.2.1.6 | 20220714 | Added RS485 setting and acquisition interface |

**Catalogue**

[Version Information 2](#_Toc15576)

[1.Device management 5](#_Toc22157)

[1.1Basic device information settings 5](#_Toc12360)

[1.1.1 Time setting 5](#_Toc4769)

[1.1.2 Time synchronization server settings 7](#_Toc1579)

[1.1.3 Time synchronization interval setting 9](#_Toc24635)

[1.1.4 Device installation type setting 11](#_Toc26292)

[1.1.5 OSD display information settings 13](#_Toc15678)

[1.1.6 RS485 information setting 15](#_Toc17393)

[1.2 Obtain basic equipment information 17](#_Toc30800)

[1.2.1 Time acquisition 17](#_Toc2942)

[1.2.2 Time synchronization server acquisition 19](#_Toc25600)

[1.2.3 Time synchronization interval acquisition 21](#_Toc31623)

[1.2.4 Device SN acquisition 23](#_Toc3135)

[1.2.5 Obtain device MAC address 25](#_Toc25825)

[1.2.6 Obtain device firmware version number 27](#_Toc18572)

[1.2.7 Device Type acquisition 29](#_Toc27404)

[1.2.8 Device installation type acquisition 31](#_Toc6265)

[1.2.9 Obtain camera internal parameters 33](#_Toc6422)

[1.2.10 Get device model 35](#_Toc25984)

[1.2.11 RS485 protocol information acquisition 37](#_Toc6836)

[1.3 Device network information settings 39](#_Toc23604)

[1.3.1 Network parameter settings 39](#_Toc8975)

[1.4 Obtain device network information 42](#_Toc24665)

[1.4.1 Network parameter acquisition 42](#_Toc5914)

[1.5 Equipment maintenance and upgrades 45](#_Toc31339)

[1.5.1 Reboot the device 45](#_Toc32616)

[1.5.2 Restore the device to factory settings 47](#_Toc21140)

[1.5.3 Restore default network parameters 49](#_Toc9688)

[1.5.4 Restore default passenger flow parameters 51](#_Toc11302)

[1.5.5 Clear passenger flow data 53](#_Toc9627)

[1.5.6 Query the status of cleared passenger flow data 55](#_Toc11795)

[1.5.7 Firmware upgrade 57](#_Toc10326)

[1.5.8 Check firmware upgrade status 59](#_Toc26619)

[1.5.9 Log acquisition 61](#_Toc8563)

[2.Passenger flow statistics 63](#_Toc16579)

[2.1Passenger flow basic parameter settings 63](#_Toc8415)

[2.1.1 Height setting 63](#_Toc8917)

[2.1.2 Detection area settings 66](#_Toc24771)

[2.1.3 Detection line settings 69](#_Toc7036)

[2.1.4 Detection direction setting 71](#_Toc3578)

[2.1.5 Detect switch settings 73](#_Toc3572)

[2.1.6 Http json parameter settings 75](#_Toc2391)

[2.1.7 Http xml parameter settings 77](#_Toc24616)

[2.2 Obtaining basic passenger flow parameters 80](#_Toc4087)

[2.2.1 Height parameter acquisition 80](#_Toc30261)

[2.2.2 Get detection area range 82](#_Toc27435)

[2.2.3 Detection area acquisition 85](#_Toc15144)

[2.2.4 Get detection line range 88](#_Toc3272)

[2.2.5 Detection line acquisition 90](#_Toc10388)

[2.2.6 Detection direction acquisition 92](#_Toc25307)

[2.2.7 Detection switch acquisition 94](#_Toc3059)

[2.2.8 Http json parameter acquisition 96](#_Toc17938)

[2.2.9 Http xml parameter acquisition 98](#_Toc32751)

[2.3 Passenger flow data interface 101](#_Toc16475)

[2.3.1 Passenger flow data query 101](#_Toc256)

[2.3.2 Passenger flow data acquisition 103](#_Toc5508)

[2.3.3 Obtaining passenger flow tracking results 107](#_Toc13)

[2.3.4 Obtain client flow statistics results 110](#_Toc1512)

[2.3.5 Acquisition of passenger flow historical data 112](#_Toc10983)

[3.Video image class 115](#_Toc17256)

[3.1 Capture image interface 115](#_Toc8670)

[3.2 Capture base64 image interface 117](#_Toc4926)

[3.3 Depth map switch setting interface 119](#_Toc19139)

[3.4 Depth map switch status acquisition interface 121](#_Toc11127)

[4.Public return code 124](#_Toc13572)

### 1.Device management

#### 1.1Basic device information settings

##### 1.1.1 Time setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data Requesting Device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | The data requesting device sets the system time and time zone to the passenger flow device | | | |
| Interface name | passengerFlow/deviceManage/deviceTimeSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Verification parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| SystemTime | String | 100 | yyyyMMddHHmmss format |
| TimeZone | int | 4 bytes | -12—13 time zone |
| interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex (parameter sorting &1096931dc28e4ecc9ddcb14e8760d53c), converted to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/deviceTimeSet  Host:127.0.0.1:8020  Content-Length:\*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Calculated (put in the message header)  Send sample parameters as follows:  {  "SystemTime":"20190814175550",  "TimeZone":8  }  The returned sample parameters are as follows:  {  "result":{  "isError": false,  "code":"0",  "message": "Set deviceTime success!"  }  } | | | |

##### 1.1.2 Time synchronization server settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | The data requesting device sets the time synchronization server to the passenger flow device | | | |
| Interface name | passengerFlow/deviceManage/timeSyncServerSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Verification parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| TimeServer | String | 100 |  |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/timeSyncServerSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "TimeServer":"time.7x24s.com"  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set time sync Server success!"  }  } | | | |

##### 1.1.3 Time synchronization interval setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets the interval for synchronizing time with the time server to Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/timeSyncIntervalSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| TimeSyncInterval | int | 4 bytes | Legal value 1——60\*24\*7 Unit: minutes |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/timeSyncIntervalSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "TimeSyncInterval":1440  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set time sync interval success!"  }  } | | | |

##### 1.1.4 Device installation type setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets the installation Type of the device to Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/installationTypeSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| InstallationType | int | 4 bytes | Legal values are 1 and 2:  1: Horizontal installation  2: Diagonal installation |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/installationTypeSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "InstallationType":2  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": " Set installation type success"  }  } | | | |

##### 1.1.5 OSD display information settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets OSD display information to Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/OSDInfoSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| Station | string | 255 | Site Information |
| GPS | string | 255 | GPS information |
| Speed | string | 255 | Speed information in km/h |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/OSDInfoSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "Station":"No. 21-777-upward-Yinchuan Road",  "GPS":"1111,1111",  "Speed":"25.6"  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": " Set osd info success"  }  } | | | |

##### 1.1.6 RS485 information setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets RS485 information to Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/RS485Set | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| protocol | int | 4 | Protocol Type:  0: Do not open the protocol  1: Current-limiting LED protocol  2: Modbus protocol |
| baudRate | int | 4 | Baud Rate |
| address | int | 4 | modbus address |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/ RS485Set  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "address":100,  "baudRate":9600,  "protocol":0  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": " RS485 set success"  }  } | | | |

#### 1.2 Obtain basic equipment information

##### 1.2.1 Time acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains system time and time zone from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/deviceTimeGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | 0 | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| SystemTime | String | 100 | yyyyMMddHHmmss format |
| TimeZone | int | 4 bytes | -12—12 time zone |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/deviceTimeGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get DeviceTime success!"  },  "SystemTime":"201908141755",  "TimeZone":8  } | | | |

##### 1.2.2 Time synchronization server acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the time synchronization server from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/timeSyncServerGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | 0 | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| TimeServer | String | 100 |  |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/timeSyncServerGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get timeSyncServer success"  },  "TimeServer":"time.7x24s.com"  } | | | |

##### 1.2.3 Time synchronization interval acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the synchronization time interval with the time server from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/timeSyncIntervalGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| TimeSyncInterval | int | 4 bytes | Unit: minutes |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/timeSyncIntervalGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get time sync interval success"  }，  "TimeSyncInterval":1440  } | | | |

##### 1.2.4 Device SN acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the device SN from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/deviceSNGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError": true/false  "code":"String"  "message": "String"  } |
| deviceSN | String | 32 |  |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/deviceSNGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get deviceSn success!"  }，  "deviceSN": "1030001906250059"  } | | | |

##### 1.2.5 Obtain device MAC address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the device MAC address from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/deviceMacAddrGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError": true/false  "code":"String"  "message": "String"  } |
| macAddr | String | 32 |  |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/deviceMacAddrGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get deviceMac success!"  }，  "macAddr": "4c-bc-98-60-00-7b"  } | | | |

##### 1.2.6 Obtain device firmware version number

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the device firmware version number from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/deviceFirmwareVerGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError": true/false  "code":"String"  "message": "String"  } |
| FirmwareVer | String | 32 |  |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/deviceFirmwareVerGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get device firmware Ver success"  }，  "FirmwareVer": "V1.0.0\_190814\_R"  } | | | |

##### 1.2.7 Device Type acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the device Type from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/deviceTypeGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError": true/false  "code":"String"  "message": "String"  } |
| DeviceType | int | 32 | Devices are divided into two types:  1: 140° wide angle equipment  2: 100° ordinary equipment. |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/deviceTypeGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get device type success"  }，  "DeviceType": 1  } | | | |

##### 1.2.8 Device installation type acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the device installation Type from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/installationTypeGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError": true/false  "code":"String"  "message": "String"  } |
| InstallationType | int | 32 | Legal values are 1 and 2:  1: Horizontal installation  2: Diagonally mounted |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/installationTypeGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get installation type success"  }，  "InstallationType": 1  } | | | |

##### 1.2.9 Obtain camera internal parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains camera internal parameters from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/cameraInnerParamGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError": true/false  "code":"String"  "message": "String"  } |
| CameraInnerParam | Json | 256 | The internal parameters of the camera are as follows:  {  "fx"：float,  "fy"：float,  "cu"：float,  "cv"：float  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/cameraInnerParamGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get installation type success"  }，  CameraInnerParam": {  "fx":250.3925323486328,  "fy":250.3925323486328,  "cu":320.0,  "cv":200.0  }  } | | | |

##### 1.2.10 Get device model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the camera model from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/deviceModelGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError": true/false  "code":"String"  "message": "String"  } |
| DeviceModel | string | 256 | Device model information |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/deviceModelGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get device model success"  }，  "DeviceModel":"PEA1-060201-201"  } | | | |

##### 1.2.11 RS485 protocol information acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains RS485 protocol information from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/RS485Get | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| protocol | string | - | Protocol Type: (int numeric string)  "0": Do not enable the protocol  "1": Current limiting LED protocol  "2": Modbus protocol |
| baudRate | string | - | Baud Rate（intnumeric string）,such as:"9600" |
| address | string | - | modbus地址（intnumeric string）, such as:"1" |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/RS485Get  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get device model success"  }，  "address":"100",  "baudRate":"9600",  "protocol":"0"  } | | | |

#### 1.3 Device network information settings

##### 1.3.1 Network parameter settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets device network parameters to Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/NetParamSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| NetType | String | 32 | The legal values of Network Type are only "static" and "DHCP"  Only legal IP, subnet mask and gateway are required when set to static. |
| Ip | String | 32 | Network IP address (optional, required when static) |
| SubnetMask | String | 32 | Subnet mask (optional, required when staic) |
| Gateway | String | 32 | Gateway (optional, required when staic is used) |
| DNS1 | String | 32 | Primary DNS (optional) |
| DNS2 | String | 32 | Backup DNS (optional) |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/NetParamSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "NetType":"static",  "Ip":"192.168.1.100",  "SubnetMask":"255.255.255.0",  "Gateway":"192.168.1.1",  " DNS1":"192.168.1.1",  " DNS2":"192.168.1.2"  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set netparam success!"  }  } | | | |

#### 1.4 Obtain device network information

##### 1.4.1 Network parameter acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains device network parameters from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/NetParamGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | 0 | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| NetParam | JSON | 512 | The parameters are as follows  {  "NetType": (String),  "Ip": (String),  "SubnetMask":（String）  "Gateway": （String）  "DNS1": （String）  " DNS2": （String）  }  Note: DNS is optional. Do not report when there is no DNS |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/NetParamGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get device NetParam success"  }，  NetParam{  "NetType":"static",  "Ip":"192.168.1.100",  "SubnetMask":"255.255.255.0",  "Gateway":"192.168.1.1",  " DNS1":"192.168.1.1",  " DNS2":"192.168.1.2"  }  } | | | |

#### 1.5 Equipment maintenance and upgrades

##### 1.5.1 Reboot the device

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains device network parameters from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/DeviceReboot | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/DeviceReboot  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set device reboot start!"  }  } | | | |

##### 1.5.2 Restore the device to factory settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device restores factory settings to Passenger flow equipment settings | | | |
| Interface Name | passengerFlow/deviceManage/DeviceFactoryReset | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret.  Restore the device to factory settings: set all parameters to default |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/DeviceFactoryReset  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set device factory reset!"  }  } | | | |

##### 1.5.3 Restore default network parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device restores default network parameters to Passenger flow equipment settings | | | |
| Interface Name | passengerFlow/deviceManage/NetParamReset | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret.  The device only restores the default network parameters |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/NetParamReset  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "set net param reset start"  }  } | | | |

##### 1.5.4 Restore default passenger flow parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device Set and restore default passenger flow parameters to Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/passengerFlowParamReset | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret.  The device only restores the default passenger flow parameters |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/passengerFlowParamReset  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "set passengerflow param reset start"  }  } | | | |

##### 1.5.5 Clear passenger flow data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device clears passenger flow data to Passenger flow equipment settings | | | |
| Interface Name | passengerFlow/deviceManage/passengerFlowDataReset | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret.  The device only restores the default passenger flow parameters |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/ passengerFlowDataReset  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "set passengerflow data reset start"  }  } | | | |

##### 1.5.6 Query the status of cleared passenger flow data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device queries Passenger flow equipment to clear passenger flow data status | | | |
| Interface Name | passengerFlow/deviceManage/passengerFlowDataResetStateGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be capitalized | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | 0 | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| passengerFlowDataResetStateValue | int | 4 bytes | 0，-1，-2 |
| passengerFlowDataResetStateInformation | String | 100 |  |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/ passengerFlowDataResetStateGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get PassengerFlowDataResetState success!"  },       "passengerFlowDataResetStateValue": 0,  "passengerFlowDataResetStateInformation": "Camera PassengerFlow Data Reset Succeed"  } | | | |

##### 1.5.7 Firmware upgrade

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets firmware upgrade to Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/FirmwareUpdate | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret.  Data request device can call this interface by placing the upgrade package in the /home/firmware path of the device. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/FirmwareUpdate  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set Fireware Update start"  }  } | | | |

##### 1.5.8 Check firmware upgrade status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 定义方 |  | | | |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device queries Passenger flow equipment for firmware upgrade status | | | |
| Interface Name | passengerFlow/deviceManage/FirmwareUpdateStateGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be capitalized | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | 0 | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| updateResultValue | int | 4 bytes | 0，-1，-4 |
| updateStateInformation | String | 100 |  |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/FirmwareUpdateStateGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get FirmwareUpdateState success!"  },  "updateResultValue":0  "updateStateInformation":"Camera Firmware Update Succeed",  } | | | |

##### 1.5.9 Log acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains logs at a specified time from Passenger flow equipment | | | |
| Interface Name | passengerFlow/deviceManage/logGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be capitalized | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| logTime | String | 100 | yyyyMMddHHmmss format  Note: Just fill in 0 for HHmmss. The basic unit for obtaining logs is days. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| logFileName | String | 64 | The format is SN\_TIME.log |
| logFile | File | Maximum 1M | Returns the binary data of File |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/deviceManage/logGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get log success"  }  "logFileName":"1030001903120041\_20190819.log";  "logFile":File content binary data;  } | | | |

### 2.Passenger flow statistics

#### 2.1Passenger flow basic parameter settings

##### 2.1.1 Height setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets height to Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/HeightParamSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| DetectionHeight | int | 4 | Detection height legal range  Large wide-angle equipment: 190cm~350cm  Ordinary equipment: 220cm~600cm |
| FilterHeight | int | 4 | Filter height legal range 0-150 unit cm |
| RotateDirection | int | 4 | Rotation direction legal range 0-3  0: x-axis  1: y-axis  2: x-axis aisle  3: y-axis aisle |
| RotateAngle | float | 4 | Legal range of rotation angle  Large wide-angle equipment: 0~20 degrees  Ordinary equipment: 0~50 degrees |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/HeightParamSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "DetectionHeight":300,  "FilterHeight":120，  "RotateDirection":0,  "RotateAngle":0.0  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "setHightParam success"  }  } | | | |

##### 2.1.2 Detection area settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets detection area to Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionAreaParamSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| LeftUpPointX | int | 4 | Starting coordinate X of the upper left corner of the detection area |
| LeftUpPointY | int | 4 | Starting coordinate Y of the upper left corner of the detection area |
| RightUpPointX | int | 4 | Starting coordinate X of the upper right corner of the detection area |
| RightUpPointY | int | 4 | Starting coordinate Y of the upper right corner of the detection area |
| RightDownPointX | int | 4 | Starting coordinate X of the lower right corner of the detection area |
| RightDownPointY | int | 4 | Starting coordinate Y of the lower right corner of the detection area |
| LeftDownPointX | int | 4 | Starting coordinate X of the lower left corner of the detection area |
| LeftDownPointY | int | 4 | Starting coordinate Y of the lower left corner of the detection area |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionAreaParamSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "LeftUpPointX": 260,  "LeftUpPointY": 180,  "RightUpPointX": 340,  "RightUpPointY": 180,  "RightDownPointX": 360,  "RightDownPointY": 240,  "LeftDownPointX": 260,  "LeftDownPointY": 240  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "setDetectionArea success"  }  } | | | |
|  |  | | | |

##### 2.1.3 Detection line settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets detection line to Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionLineParamSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| DetectionLine | int | 4 bytes | Note: The Y pixel value of the detection line in the left image of the camera |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionLineParamSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  " DetectionLine ":200  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "setDetectionLine success"  }  } | | | |

##### 2.1.4 Detection direction setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets the detection direction to Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionDirectionSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| DetectionDirection | int | 4 bytes | 0: Reverse 1: Forward |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionDirectionSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "DetectionDirection":0  }  Request message example:  result  {  "isError": false,  "code":"0",  "message": "setDetectionDirection success"  } | | | |

##### 2.1.5 Detect switch settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets the detection switch to Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionSwitchSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| DetectionSwitch | int | 4 | 0: Detection is off 1: Detection is on |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionSwitchSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "DetectionSwitch":0  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "setDetectionSwitch success"  }  } | | | |

##### 2.1.6 Http json parameter settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets http json parameters to Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/HttpJsonParamSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| Server1 | string | - | Server address of data push 1 |
| Port1 | int | 32 | Port number of data push 1 |
| Interval1 | int | 32 | Push interval for data push 1 |
| Enable1 | init | 32 | Enable flag of data push 1 |
| Server2 | string | - | Server address of data push 2 |
| Port2 | int | 32 | Port number of data push 2 |
| Interval2 | int | 32 | Push interval of data push 2 |
| Enable2 | init | 32 | Enable flag of data push 2 |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/HttpJsonParamSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "Server1":"192.168.8.101",  "Port1":5001,  "Interval1":11,  "Enable1":0,  "Server2":"192.168.8.102",  "Port2":5002,  "Interval2":12,  "Enable2":0  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set http json param success"  }  } | | | |

##### 2.1.7 Http xml parameter settings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets http xml parameters to Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/HttpXmlParamSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| PostUrl | string | - | Data address for data push |
| HeartUrl | string | - | Heartbeat address for data push |
| DeviceId | string | 20 | Device ID for data push |
| Interval | int | 32 | Push interval for data push |
| IntervalMode | int | 32 | Data push interval mode |
| RealTimeMode | int | 32 | Data push real-time mode |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/HttpXmlParamSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "PostUrl":"http://192.168.8.107:5002/api/camera/dataUpload",  "HeartUrl":"http://192.168.8.107:5002/api/camera/heartBeat",  "DeviceId":"2010012112250305",  "Interval":2,  "IntervalMode":0,  "RealTimeMode":0  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Set http xmlparam success"  }  } | | | |

#### 2.2 Obtaining basic passenger flow parameters

##### 2.2.1 Height parameter acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains height parameters from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/HeightParamGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| DetectionHeight | int | 4 | Legal range of detection height:  Large wide-angle equipment: 190cm~350cm  Ordinary equipment: 220cm~600cm |
| FilterHeight | int | 4 | Filter height legal range 0-150 unit cm |
| RotateDirectionValue | int | 4 | Rotation direction legal range 0-3  0: x-axis  1: y-axis  2: x-axis aisle  3: y-axis aisle |
| RotateAngleValue | float | 4 | Legal range of rotation angle  Large wide-angle equipment: 0~20 degrees  Ordinary equipment: 0~50 degrees |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/HeightParamGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get getHightParam success! "  },  "DetectionHeight": 300,  "FilterHeight":120，  "RotateDirectionValue":0,  "RotateAngleValue":1.5  } | | | |

##### 2.2.2 Get detection area range

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | The Data request device obtains the legal range of the detection area corresponding to the current height from the Passenger flow equipment (different detection heights correspond to different detection ranges. When the detection height is set, the detection range should be obtained and reset) | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionAreaRangeGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| DetectionAreaMax | JSON | 512 | The parameters are as follows  {  "LeftUpPointX": (int),  "LeftUpPointY": (int),  "RightUpPointX": (int),  "RightUpPointY": (int),  "RightDownPointX": (int),  "RightDownPointY": (int),  "LeftDownPointX": (int),  "LeftDownPointY": (int)  } |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionAreaRangeGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": " Get DetectionAreaRange success! "  },  "DetectionAreaMax": {  "LeftUpPointX":260,  "LeftUpPointY":180,  "RightUpPointX":340,  "RightUpPointY":180,  "RightDownPointX":360,  "RightDownPointY":240,  "LeftDownPointX":260,  "LeftDownPointY":240  }  **Remark:**  **The relevant parameters of the detection area are based on the 640\*400 pixels in the left image as a reference** | | | |

##### 2.2.3 Detection area acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the detection area from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionAreaGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| DetectionArea | JSON | 512 | The parameters are as follows  {  "LeftUpPointX": (int),  "LeftUpPointY": (int),  "RightUpPointX": (int),  "RightUpPointY": (int),  "RightDownPointX": (int),  "RightDownPointY": (int),  "LeftDownPointX": (int),  "LeftDownPointY": (int)  }, |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionAreaGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get getDetectionArea success! "  },  "DetectionArea": {  "LeftUpPointX":260,  "LeftUpPointY":180,  "RightUpPointX":340,  "RightUpPointY":180,  "RightDownPointX":360,  "RightDownPointY":240,  "LeftDownPointX":260,  "LeftDownPointY":240  }  } | | | |

##### 2.2.4 Get detection line range

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the detection line range from Passenger flow equipment (Note: Each time the detection area is set, the detection line range will be changed, so it is necessary to obtain the detection line range and reset it) | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionLineRangeGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| DetectionLineMin | int | 4 | Minimum value of the detection line: The minimum value of the Y coordinate of the detection line in the left picture |
| DetectionLineMax | int | 4 | Maximum value of the detection line: The maximum value of the Y coordinate of the detection line in the left picture |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionLineRangeGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get getDetectionLineRange success!"  },  " DetectionLineMin": 100,  " DetectionLineMax": 300  } | | | |

##### 2.2.5 Detection line acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the detection line position from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionLineGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| DetectionLine | int | 4 | Detection line position: The Y coordinate value of the detection line in the left picture |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionLineGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get getDetectionLine success!"  },  " DetectionLine": 200  } | | | |

##### 2.2.6 Detection direction acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the detection direction from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionDirectionGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| DetectionDirection | int | 4 | 0: Reverse 1: Forward |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionDirectionGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get getDetection Direction success!"  },  " DetectionDirection": 0  } | | | |

##### 2.2.7 Detection switch acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the detection switch from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/DetectionSwitchGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| DetectionSwitch | int | 4 | 0: off 1: on |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/DetectionSwitchGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "get detection switch success"  },  "DetectionSwitch": 0  } | | | |

##### 2.2.8 Http json parameter acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains http json parameters from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/HttpJsonParamGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Server1 | string | - | Server address of data push 1 |
| Port1 | int | 32 | Port number of data push 1 |
| Interval1 | int | 32 | Push interval for data push 1 |
| Enable1 | init | 32 | Enable flag of data push 1 |
| Server2 | string | - | Server address of data push 2 |
| Port2 | int | 32 | Port number of data push 2 |
| Interval2 | int | 32 | Push interval of data push 2 |
| Enable2 | init | 32 | Enable flag of data push 2 |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/HttpJsonParamGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get http json param success"  },  "Server1":"192.168.8.101",  "Port1":5001,  "Interval1":11,  "Enable1":0,  "Server2":"192.168.8.102",  "Port2":5002,  "Interval2":12,  "Enable2":0  } | | | |

##### 2.2.9 Http xml parameter acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains http xml parameters from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowParam/HttpXmlParamGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| PostUrl | string | - | Data address for data push |
| HeartUrl | string | - | Heartbeat address for data push |
| DeviceId | string | 20 | Device ID for data push |
| Interval | int | 32 | Push interval for data push |
| IntervalMode | int | 32 | Data push interval mode |
| RealTimeMode | int | 32 | Data push real-time mode |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowParam/HttpXmlParamGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get http xml param success"  },  "PostUrl":"http://192.168.8.107:5002/api/camera/dataUpload",  "HeartUrl":"http://192.168.8.107:5002/api/camera/heartBeat",  "DeviceId":"2010012112250305",  "Interval":2,  "IntervalMode":0,  "RealTimeMode":0  } | | | |

#### 2.3 Passenger flow data interface

##### 2.3.1 Passenger flow data query

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device queries Passenger flow equipment for passenger flow data, and Passenger flow equipment returns the cumulative number of passenger flow data starting from the query index (historical accumulation) | | | |
| Interface Name | passengerFlow/passengerFlowData/PassengerFlowDataAsk | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| startIndex | int | 4 | Passenger flow data starting index (legal value >= 0) |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| numberOfPassengers | int | 4 | The number of passenger flow data starting from the starting index |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowData/PassengerFlowDataAsk  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "startIndex":100  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "ask passenger flow data success"  },  "numberOfPassengers":200  } | | | |

##### 2.3.2 Passenger flow data acquisition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains passenger flow data in a specified index range from Passenger flow equipment  （The device will cache three months of passenger flow data locally on the camera.） | | | |
| Interface Name | passengerFlow/passengerFlowData/PassengerFlowDataGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| startIndex | int | 4 | Starting index (legal value >= 0) |
| getQuantity | int | 4 | Get the number of passenger flow data (legal value > 0)  If the actual passenger flow data < the value of this parameter, the actual passenger flow data volume will be returned. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| rptQuantity | int | 4 | The actual number of customer flow data returned |
| PassengerFlowData | JSON | Up to 3 days of historical data | [  {  "idIndex": (int)  "timestamp":(long long)  "eventType": (int)  "stayTime":(int)  },  {  "idIndex": (int)  "timestamp":(long long)  "eventType": (int)  "stayTime":(int)  },  …….  ]  备注：  idIndex: ID index (unique)  timestamp: Disappearing timestamp (UNIX timestamp)  eventType: Event Type   |  |  | | --- | --- | | Event Type encoding | Meaning | | 0 | Enter | | 1 | Leave | | 2 | Pass through | | 3 | Return | | -1 | Invalid |   stayTime: length of stay in ms |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/passengerFlowData/PassengerFlowDataGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "startIndex":0,  "getQuantity":100  }  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "ask passenger flow data success"  },  "rptQuantity": 2，//(There are only two pieces of data, so only 2 pieces of data will be replied to.)  "PassengerFlowData":[  {  "idIndex": 0,  "timestamp":1565801855 (disappeared timestamp),  "eventType": 0,  "stayTime":2500 (Stayed for 2500ms)  },  {  "idIndex": 1,  "timestamp":1565801856,  "eventType": 1,  "stayTime":2200  }  ]  } | | | |

##### 2.3.3 Obtaining passenger flow tracking results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains passenger flow tracking results from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowDataGet/TrackingResultGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| timestamp | double | 32 | Returns microsecond-level timestamp information. The decimal point is the second-level timestamp; the decimal point is the microsecond timestamp. |
| totalCnt | int | 32 | The total number of passengers tracked in the current scene |
| TrackingResultData | JSON | Passenger flow tracking result data in the detection area | [  {  "id": (int), // Human body tracking ID  "x\_h":(long long), //head coordinate point x  "y\_h": (int), //head coordinate point y  "x\_f":(int), //Human foot coordinate point x  "y\_f":(int) //Human foot coordinate point y  },  {  "id": (int),  "x\_h":(long long),  "y\_h": (int),  "x\_f":(int),  "y\_f":(int),  }  …  ] |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/PassengerFlowDataGet/TrackingResultGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get tracking result success"  },  "totalCnt": 2,  "TrackingResultData":[  {  "id": 0,  "x\_h":344,  "y\_h":143,  "x\_f":330,  "y\_f":175  },  {  "id": 1,  "x\_h":230,  "y\_h":231,  "x\_f":297,  "y\_f":207  }  ]  } | | | |
|  |  | | | |

##### 2.3.4 Obtain client flow statistics results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the current passenger flow statistics results from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowDataGet/DisplayDataGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| enter | int | 32 | Number of people entering |
| leave | int | 32 | Number of people leaving |
| pass | int | 32 | Number of people passing by |
| return | int | 32 | Number of people returning |
| stay | int | 32 | Number of people staying |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/PassengerFlowDataGet/DisplayDataGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get display data success"  },  "enter": 2,  " leave": 1,  " pass": 5,  "return": 3,  "stay": 2  } | | | |

##### 2.3.5 Acquisition of passenger flow historical data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains historical passenger flow statistics results from Passenger flow equipment | | | |
| Interface Name | passengerFlow/passengerFlowData/HistoryDataGetFromTimestamp | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| startTimestamp | int | 32 | Second level start timestamp |
| endTimestamp | int | 32 | Second level end stamp |
| Interface return value  (String, key-value pair convertible to JSON) | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| enter | int | 32 | Number of people entering |
| leave | int | 32 | Number of people leaving |
| pass | int | 32 | Number of people passing by |
| return | int | 32 | Number of people returning |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/PassengerFlowDataGet/HistoryDataGetFromTimestamp  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Request message example:  {  "result":{  "isError": false,  "code":"0",  "message": "Get history data success"  },  "enter": 2,  " leave": 1,  " pass": 5,  "return": 3  } | | | |

### 3.Video image class

#### 3.1 Capture image interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device captures images in real time from Passenger flow equipment | | | |
| Interface Name | passengerFlow/ImageAndVideo/CapturePicture | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| imageFileName | String | 32 | yyyyMMddHHmmssms.jpg format  Such as 20190814220000025.jpg represents  Picture at 22:00:00:25 milliseconds on August 14, 2019 |
| imageFile | File | 640\*400 | binary data |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/ImageAndVideo/CapturePicture  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  Request message example:  {  result  {  "isError": false,  "code":"0",  "message": "capture picture success!"  },  "imageFileName":"20190814220000025.jpg",  "imageFile":(image data)  } | | | |

#### 3.2 Capture base64 image interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device captures base64 images in real time from the passenger flow camera | | | |
| Interface Name | passengerFlow/ImageAndVideo/CapturePictureBase64 | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None, note that MD5 encryption only encrypts ServiceSecret. |
| Interface return value | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| imageFileName | String | 32 | yyyyMMddHHmmssms.jpg format  Such as 20190814220000025.jpg represents  Picture at 22:00:00:25 milliseconds on August 14, 2019 |
| imageFile | String | - | Base64 data |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/ImageAndVideo/CapturePictureBase64  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  Request message example:  {  result  {  "isError": false,  "code":"0",  "message": "capture picture success!"  }  "imageFileName":"20190814220000025.jpg",  "imageFile":"base64 data"  } | | | |

#### 3.3 Depth map switch setting interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device sets the depth map transmission switch to the passenger flow camera | | | |
| Interface Name | passengerFlow/ImageAndVideo/DepthVideoSwitchSet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| DepthVideoSwitch | Int | 4 | Legal value  0: Turn off depth map transmission  1: Enable depth map transmission |
| Interface return value | Name | Type | The maximum length | Return value description |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/ImageAndVideo/DepthVideoSwitchSet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  {  "DepthVideoSwitch":1  }  Request message example:  {  result  {  "isError": false,  "code":"0",  "message": " setDepthVideoSwitch success "  }  } | | | |

#### 3.4 Depth map switch status acquisition interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sender | Data request device | | | |
| Receiver | Passenger flow equipment | | | |
| Interface definition | | | | |
| Interface description | Data request device obtains the depth map switch status from the passenger flow camera | | | |
| Interface Name | passengerFlow/ImageAndVideo/DepthVideoSwitchGet | | | |
| Interface Protocol | HTTP Restful POST | | | |
| Interface calling address | The port is fixed at 8020 | | | |
| Message header definition | HTTP standard header field Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;  Content-Length:\*\*\*; | | | |
| Validation parameters | Validation parameters: Sign  MD5 encryption (all parameters are sorted in ASCII order + ServiceSecret for encryption).  1: Authorization token ServiceSecret agreed upon by both parties;  2: Parameters need to be in uppercase letters; | | | |
| Interface parameters | Name | Type | The maximum length | Interface description |
| None | None | None | None |
| Interface return value | Name | Type | The maximum length | Return value description |
| DepthVideoSwitch | Int | 4 | legal value  0: Turn off depth map transmission  1: Enable depth map transmission |
| result | JSON | 255 | The parameters are as follows  {  "isError":  "code":  "message":  } |
| Interface usage example | ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c  Sign:  md5Hex(Parameter sorting&1096931dc28e4ecc9ddcb14e8760d53c)，Convert to uppercase  Request message example:  POST:\*\*\*\*/passengerFlow/ImageAndVideo/DepthVideoSwitchGet  Host:127.0.0.1:8020  Content-Length: \*\*\*  Accept:application/json;  Content-Type:application/json;charset=utf-8;  Sign: Based on calculation (put in the message header)  Send sample parameters as follows:  Request message example:  {  result  {  "isError": false,  "code":"0",  "message": " setDepthVideoSwitch success "  },  "DepthVideoSwitch":1  } | | | |

### 4.Public return code

Note: Return code 0 is successful, < 0 means the camera itself is running abnormally; > 0 means the interactive command is abnormal.

| **Return code Name** | **Return code** | **Return code description** | **solution** |
| --- | --- | --- | --- |
| SUCCESS | 0 | Call successful |  |
| CAMERA\_INIT\_ERR | -1 | Camera initialization failed |  |
| CAMERA\_OPEN\_ERR | -2 | Camera failed to open |  |
| CAMERA\_START\_ERR | -3 | Camera streaming failed |  |
| CAMERA\_SET\_FPS\_ERR | -4 | Camera setting frame rate failed |  |
| CAMERA\_SET\_MODE\_ERR | -5 | Camera setup mode failed |  |
| CAMERA\_SET\_AUTO\_EXPOUSE\_ERR | -6 | Camera settings auto-exposure failed |  |
| CAMERA\_SET\_AUTO\_GAIN\_ERR | -7 | Camera setting auto gain failed |  |
| CAMERA\_SET\_EXPOUSE\_ERR | -8 | Camera settings exposure failed |  |
| CAMERA\_SET\_GAIN\_ERR | -9 | Camera setup gain failed |  |
| CAMERA\_GET\_VER1\_ERR | -10 | Camera failed to obtain version number 1 |  |
| CAMERA\_GET\_VER2\_ERR | -11 | Camera failed to obtain version number 2 |  |
| CAMERA\_GET\_SN\_ERR | -12 | Camera failed to obtain SN |  |
| CAMERA\_GET\_NET\_PARAM\_ERR | -13 | The camera failed to obtain network parameters |  |
| CAMERA\_GET\_IMAGE\_FAILED | -14 | Camera failed to acquire image |  |
| CAMERA\_GET\_IMAGE\_TIMEOUT | -15 | Camera image acquisition timeout |  |
| CAMERA\_GET\_PARAM\_ERR | -16 | Camera failed to obtain internal parameters |  |
| CAMERA\_UPDATE\_ERR | -17 | Camera upgrade failed |  |
| CAMERA\_RUN\_FPS\_ERR | -18 | Camera frame rate is running abnormally |  |
| CAMERA\_AMBIENT\_BRIGHTNESS\_TOO\_LOW | -19 | The brightness of the camera operating environment is too low |  |
| SIGN\_METHOD\_NOTNULL | 1 | Signature parameter is empty | Add signature parameters |
| SIGN\_TYPE\_NOTEXIST | 2 | Signature Type does not exist | Currently only supports MD5 |
| SIGN\_WRONG | 3 | Signature error | Generate signatures based on agreed signature rules |
| PARAM\_WRONG | 4 | Parameter error | Parameters should be set within the agreed legal range |
| SET\_PARAM\_FALID | 5 | Failed to set parameters |  |
| GET\_PARAM\_FALID | 6 | Failed to get parameters |  |