

ビジターカウンター**DC8000**

HTTP プロトコルインターフェイス 仕様書



V2.1.8(3)

ダンボネット・システムズ株式会社

目次

1.機器管理.....	1
1.1 基本情報設定.....	1
1.1.1 時刻設定.....	1
1.1.2 時刻同期(NTP)サーバの設定.....	2
1.1.3 NTP サーバ同期間隔の設定.....	4
1.1.4 OSD 表示メッセージの設定.....	6
1.1.5 RS485 メッセージの設定.....	8
1.2 基本情報取得.....	10
1.2.1 時刻取得.....	10
1.2.2 時刻同期(NTP)サーバ名の取得.....	11
1.2.3 NTP サーバ同期間隔の取得.....	13
1.2.4 シリアルナンバーの取得.....	15
1.2.5 MAC アドレスの取得.....	16
1.2.6 ファームウェアのバージョン番号取得.....	18
1.2.7 デバイスタイプ取得.....	19
1.2.8 カメラ内部パラメータの取得.....	21
1.2.9 OSD 表示情報の取得.....	23
1.2.10 RS485 プロトコル情報取得.....	25
1.3 ネットワーク情報の設定.....	27
1.3.1 ネットワークパラメータ設定.....	27
1.4 ネットワーク情報の取得.....	29
1.4.1 ネットワークパラメータ取得.....	29
1.5 メンテナンス.....	32
1.5.1 デバイスを再起動する.....	32
1.5.2 工場出荷時の設定に戻す.....	33
1.5.3 ネットワークパラメータをデフォルトに戻す.....	35
1.5.4 人流パラメータをデフォルトに戻す.....	36
1.5.5 人流統計データリセット.....	38
1.5.6 空の人流フローデータの状況を問い合わせる.....	39
1.5.7 ファームウェアのアップデート.....	41
1.5.8 ファームウェアアップデートの検証.....	43
1.5.9 ログ取得.....	45
2.トラフィック統計.....	47
2.1 人流ベースパラメータ設定.....	47
2.1.1 高さ設定.....	47
2.1.2 検知領域の設定.....	49
2.1.3 検知ラインの設定.....	51
2.1.4 検知方向設定.....	53
2.1.5 検知スイッチの設定.....	54
2.1.6 HTTP JSON パラメータの設定.....	56
2.1.7 HTTP xml パラメータの設定.....	58
2.2 人流ベースパラメータ取得.....	60

2.2.1 検知高パラメータ取得	60
2.2.2 検知エリア範囲取得	63
2.2.3 検知領域取得	65
2.2.4 検知ライン範囲取得	67
2.2.5 検知ラインの位置取得	69
2.2.6 検知方向取得	71
2.2.7 検知スイッチの状態取得	72
2.2.8 HTTP JSON パラメータの取得	74
2.2.9 HTTP xml パラメータの取得	76
2.3 人流データインターフェイス	78
2.3.1 人流データ照会	78
2.3.2 人流データの取得	80
2.3.3 人流追跡情報の取得	83
2.3.4 人流統計値の取得	86
2.3.5 人流統計履歴データ取得	88
3.画像スキャンインターフェイス	90
3.1 リアルタイムでのイメージのキャプチャ	90
3.2 Base64 型式でのイメージのキャプチャ	92
3.3 Base64 型式でのデバッグイメージのキャプチャ	94
4.パブリックリターンコード	96

1.機器管理

1.1 基本情報設定

1.1.1 時刻設定

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターのシステム時刻とタイムゾーンを設定する			
名称	passengerFlow/deviceManage/deviceTimeSet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	SystemTime	String	100	yyyyMMddHHmmss 形式
	TimeZone	int	4 バイト	TimeZone 値
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列)

				}
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(SystemTime20231118020000TimeZone91096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/deviceManage/deviceTimeSet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: EAC9978B573E6FB9D93586D7DCB9CCAE</p> <p>送信例:</p> <pre>{ "SystemTime": "20231118020000", "TimeZone": 9 }</pre> <p>受信例:</p> <pre>{ "result": { "isError": false, "code": "0", "message": "Set deviceTime success!" } }</pre>			

1.1.2 時刻同期(NTP)サーバの設定

送信者	データ要求装置
レシーバー	ビジターカウンター

インターフェイス定義				
説明	ビジターカウンターに時刻同期(NTP)サーバを設定する			
名称	passengerFlow/deviceManage/timeSyncServerSet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	TimeServer	String	100	NTP サーバ FQDN(.com に限定)
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(TimeServertime.google.com1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/timeSyncServerSet Host:127.0.0.1:8020 Content-Length: ***			

	Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: <pre>{ "TimeServer": "time.google.com" }</pre> 受信例: <pre>{ "result": { "isError": false, "code": "0", "message": "Set time sync Server success!" } }</pre>
--	--

1.1.3 NTP サーバ同期間隔の設定

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターに時刻同期(NTP)サーバと時刻を同期させる間隔を設定する
名称	passengerFlow/deviceManage/timeSyncIntervalSet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;

バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの。 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	TimeSyncInterval	int	4 バイト	実効値 1~60*24*7 単位:分
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(TimeSyncInterval14401096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***:/passengerFlow/deviceManage/timeSyncIntervalSet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: <pre>{ "TimeSyncInterval":1440 }</pre> 受信例:			

	<pre> { "result":{ "isError": false, "code": "0", "message": " Set time sync interval success!" } } </pre>
--	--

1.1.4 OSD 表示メッセージの設定

送信者	データ要求装置												
レシーバー	ビジターカウンター												
インターフェイス定義													
説明	ビジターカウンターへ OSD 表示情報を設定する												
名称	passengerFlow/deviceManage/OSDInfoSet												
プロトコル	HTTP Restful POST												
ポート番号	8020 固定												
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;												
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする												
パラメータ	<table border="1"> <thead> <tr> <th>名称</th> <th>タイプ</th> <th>最大長</th> <th>説明</th> </tr> </thead> <tbody> <tr> <td>osdStation</td> <td>string</td> <td>255</td> <td>ポイント情報</td> </tr> <tr> <td>osdTimeEnable</td> <td>int</td> <td>32</td> <td>時刻情報表示の有効化:0 無効、1 有効</td> </tr> </tbody> </table>	名称	タイプ	最大長	説明	osdStation	string	255	ポイント情報	osdTimeEnable	int	32	時刻情報表示の有効化:0 無効、1 有効
	名称	タイプ	最大長	説明									
	osdStation	string	255	ポイント情報									
osdTimeEnable	int	32	時刻情報表示の有効化:0 無効、1 有効										

	osdPflowEnable	int	32	搭乗者数表示の有効化:0 無効、1 有効
	osdStationEnable	int	32	サイト情報表示の有効化:0は無効、1は有効
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例: POST:***:/passengerFlow/deviceManage/OSDInfoSet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例: { "osdStation": "TestPoint", "osdTimeEnable": 1, "osdPflowEnable": 1, "osdStationEnable": 1 }</p>			

	受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": " OSD Info Set success" } }</pre>
--	---

1.1.5 RS485 メッセージの設定

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターへ RS485 情報を設定する			
名称	passengerFlow/deviceManage/RS485Set			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	protocol	int	4	契約の種類 0:プロトコルオンなし

				1:電流制限型 LED プロトコル 2:Modbus プロトコル
	baudRate	int	4	ボーレート
	address	int	4	Modbus アドレス
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/ RS485Set Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: { "address":100, "baudRate":9600, "protocol":0 } 受信例: { "result":{			

	<pre> "isError": false, "code": "0", "message": "RS485 set success" } } </pre>
--	--

1.2 基本情報取得

1.2.1 時刻取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターからシステム時刻とタイムゾーンを取得する			
名称	passengerFlow/deviceManage/deviceTimeGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{

				<pre> "isError": true/false "code": "String" "message": "String"(文字列) } </pre>
	SystemTime	String	100	yyyyMMddHHmmss 形式
	TimeZone	int	4 バイト	TimeZone 値
使用例	<pre> ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***//passengerFlow/deviceManage/deviceTimeGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "result":{ "isError": false, "code": "0", "message": "get DeviceTime success!" }. "SystemTime":"202306230145 ", "TimeZone":9 } </pre>			

1.2.2 時刻同期(NTP)サーバ名の取得

送信者	データ要求装置
-----	---------

レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから時刻同期(NTP)サーバ名を取得する			
名称	passengerFlow/deviceManage/timeSyncServerGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの。 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	TimeServer	String	100	NTP サーバ FQDN
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/timeSyncServerGet			

	Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": "get timeSyncServer success!" }, "TimeServer": "time.google.com" }</pre>
--	---

1.2.3 NTP サーバ同期間隔の取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターから時刻同期(NTP)サーバとの同期間隔を取得する
名称	passengerFlow/deviceManage/timeSyncIntervalGet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化)

	1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
	TimeSyncInterval	int	4 バイト	単位:分
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***:/passengerFlow/deviceManage/timeSyncIntervalGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": "get time sync interval success" } }</pre>			

	<pre> "TimeSyncInterval":1440 } </pre>
--	--

1.2.4 シリアルナンバーの取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターのシリアルナンバーを取得する			
名称	passengerFlow/deviceManage/deviceSNGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre> { "isError": true/false "code": "String" "message": "String"(文字列) } </pre>

	deviceSN	string	32	
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/deviceManage/deviceSNGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": "get deviceSn success!" } "deviceSN":"1030001906250059" }</pre>			

1.2.5 MACアドレスの取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターから MAC アドレスを取得する
名称	passengerFlow/deviceManage/deviceMacAddrGet
プロトコル	HTTP Restful POST
ポート番号	8020 固定

ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	macAddr	String	32	MAC アドレス型式
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/deviceMacAddrGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例:			

	<pre> { "result":{ "isError": "false", "code": "0", "message": "Get deviceMac success!" } "macAddr": " 4C:BC:98:60:03:AE " } </pre>
--	---

1.2.6 ファームウェアのバージョン番号取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターからファームウェアのバージョン番号を取得する			
名称	passengerFlow/deviceManage/deviceFirmwareVerGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明

	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	FirmwareVer	String	32	
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例: POST:***/passengerFlow/deviceManage/deviceFirmwareVerGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例: { "result":{ "isError": false, "code": "0", "message": "get device firmware Ver success" } "FirmwareVer": "V6.7.4" }</p>			

1.2.7 デバイスタイプ取得

送信者	データ要求装置
レシーバー	ビジターカウンター

インターフェイス定義				
説明	ビジターカウンターからデバイスタイプを取得する			
名称	passengerFlow/deviceManage/deviceTypeGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	DeviceType	int	32	機器には 2 種類ある: 1:140°広角デバイス 2:100°共通機器
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例:			

	POST:***//passengerFlow/deviceManage/deviceTypeGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: <pre> { "result":{ "isError": false, "code": "0", "message": "Get device type success" } "DeviceType":2 } </pre>
--	---

1.2.8 カメラ内部パラメータの取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターからカメラ内部パラメータを取得する
名称	passengerFlow/deviceManage/cameraInnerParamGet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;

バリデーションパラメータ	バリデーションパラメータ: Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
	CameraInnerParam	JSON	256	カメラの内部パラメータは以下の通り: <pre>{ "fx": float, "fy": float, "cu": float, "cv": float }</pre>
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/deviceManage/cameraInnerParamGet Host:127.0.0.1:8020			

	Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": "Get camera inner param success" } } CameraInnerParam: { "fx":250.3925323486328, "fy":250.3925323486328, "cu":320.0, "cv":200.0 } }</pre>
--	---

1.2.9 OSD 表示情報の取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターから OSD 表示情報を取得する
名称	passengerFlow/deviceManage/OSDInfoGet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8,

	Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ: Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意したもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	osdStation	string	255	ポイント情報
	osdTimeEnable	int	32	時刻情報表示の有効化: 0 無効、1 有効
	osdPflowEnable	int	32	搭乗者数表示の有効化: 0 無効、1 有効
	osdStationEnable	int	32	サイト情報表示の有効化: 0 は無効、1 は有効
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***//passengerFlow/deviceManage/OSDInfoGet Host:127.0.0.1:8020			

	<p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": "OSD Info Get success" } "osdPflowEnable": "1", "osdStation": "TestPoint", "osdStationEnable": "1", "osdTimeEnable": "1" }</pre>
--	--

1.2.10 RS485 プロトコル情報取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターから RS485 プロトコル情報を取得する
名称	passengerFlow/deviceManage/RS485Get
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	<p>HTTP 標準パケットヘッダフィールド Accept:application/json,</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;</p> <p>Content-Length:***;</p>

バリデーションパラメータ	バリデーションパラメータ: Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
	protocol	string	-	プロトコルタイプ:(int 数値文字列) "0": プロトコルオンなし "1": 電流制限型 LED プロトコル "2": Modbus プロトコル
	baudRate	string	-	ボーレート(int 数値文字列)、例:"9600"
	address	string	-	Modbus アドレス(int 数値文字列)、例:"1"
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/RS485Get Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8,			

	Sign: Calculated from (メッセージヘッダに配置) 受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": "RS485 Get success" } "address": "1", "baudRate": "9600", "protocol": "1" }</pre>
--	--

1.3 ネットワーク情報の設定

1.3.1 ネットワークパラメータ設定

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターに対してネットワークパラメータを設定する
名称	passengerFlow/deviceManage/NetParamSet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化)

	1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	NetType	String	32	ネットワークタイプ 正規の値は "static" と "DHCP" のみ。リーガル IP、サブネットマスク、ゲートウェイは静的設定時のみ必要
	Ip	String	32	ネットワーク IP アドレス(オプション、static 時は必須)
	SubnetMask	String	32	サブネットマスク(オプション、static の場合は必須)
	Gateway	String	32	ゲートウェイ(オプション、static の場合は必須)
	DNS1	String	32	プライマリ DNS(オプション)
	DNS2	String	32	バックアップ DNS(オプション)
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/deviceManage/NetParamSet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8,			

	<p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <pre>{ "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" }</pre> <p>受信例:</p> <pre>{ "result": { "isError": false, "code": "0", "message": "Set netparam success!" } }</pre>
--	--

1.4 ネットワーク情報の取得

1.4.1 ネットワークパラメータ取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターからネットワークパラメータを取得する
名称	passengerFlow/deviceManage/NetParamGet
プロトコル	HTTP Restful POST
ポート番号	8020 固定

ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	NetParam	JSON	512	ここで、パラメータは以下のように読み取れます。 { "NetType":(String), "Ip":(String), "SubnetMask":(String), "Gateway":(文字列), "DNS1":(文字列), "DNS2":(文字列) } 注:DNS はオプションのフィールドです。 DNS がない場合は報告されない
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }

<p>使用例</p>	<pre> ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/deviceManage/NetParamGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "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" } } </pre>
------------	---

1.5 メンテナンス

1.5.1 デバイスを再起動する

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターを再起動する			
名称	passengerFlow/deviceManage/DeviceReboot			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign:			

	<p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***:/passengerFlow/deviceManage/DeviceReboot</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": " Set device reboot start! " } }</pre>
--	---

1.5.2 工場出荷時の設定に戻す

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターを工場出荷時の設定に戻す
名称	passengerFlow/deviceManage/DeviceFactoryReset
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	<p>HTTP 標準パケットヘッダフィールド Accept:application/json,</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX;</p> <p>Content-Length:***;</p>

バリデーションパラメータ	バリデーションパラメータ: Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret の暗号化。 デバイスを工場出荷状態に戻す:すべてのパラメータをデフォルトに戻す
戻り値)	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/deviceManage/DeviceFactoryReset Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: <pre>{ "result":{ "isError": false,</pre>			

	<pre> "code": "0", "message": " Set device factory reset! " } } </pre>
--	--

1.5.3 ネットワークパラメータをデフォルトに戻す

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターへデフォルトのネットワークパラメータを設定する			
名称	passengerFlow/deviceManage/NetParamReset			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret の暗号化。 デバイスは、デフォルトのネットワークパラメータのみを復元する
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false

				<pre> "code": "String" "message": "String"(文字列) } </pre>
使用例	<pre> ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/NetParamReset Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "result":{ "isError": false, "code": "0", "message": "set net param reset start" } } </pre>			

1.5.4 人流パラメータをデフォルトに戻す

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターへデフォルトのパラメータを設定する
名称	passengerFlow/deviceManage/passengerFlowParamReset
プロトコル	HTTP Restful POST

ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret は暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret の暗号化。 機器は、デフォルトの乗客フローパラメータのみを復元する
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/passengerFlowParamReset Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8,			

	<p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message":"set passengerflow param reset start" } }</pre>
--	---

1.5.5 人流統計データリセット

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンター内の人流データのリセットを行う			
名称	passengerFlow/deviceManage/passengerFlowDataReset			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。

				機器は、デフォルトの乗客フローパラメータのみを復元する
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "errorCode": true/false "code": "String" "message": "String"(文字列) }
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***//passengerFlow/deviceManage/ passengerFlowDataReset</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "errorCode": false, "code": "0", "message": "set passengerflow data reset start" } }</pre>			

1.5.6 空の人流フローデータの状況を問い合わせる

送信者	データ要求装置
-----	---------

レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターに空になった人流データの状態を問い合わせる			
名称	passengerFlow/deviceManage/passengerFlowDataResetStateGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	passengerFlowDataResetStateValue	int	4 バイト	0, -1, -2
	passengerFlowDataResetStateValue	String	100	

	tStateInfor mation			
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/deviceManage/ passengerFlowDataResetStateGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": " get PassengerFlowDataResetState success!" }, "passengerFlowDataResetStateValue": 0, "passengerFlowDataResetStateInformation": " Camera PassengerFlow Data Reset Succeed" }</pre>			

1.5.7 ファームウェアのアップデート

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	

説明	ビジターカウンターへファームウェアのアップロードを行う			
名称	passengerFlow/deviceManage/FirmwareUpload			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret の暗号化。
データのアップロード	なし	バイナリ	—	ファームウェアパッケージのデータをバイナリでアップロードする
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/deviceManage/FirmwareUpload Host:127.0.0.1:8020			

	Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": " Upload firmware success" } }</pre>
--	--

1.5.8 ファームウェアアップデートの検証

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターへのファームウェアアップデートを検証する
名称	passengerFlow/deviceManage/FirmwareCheckAndUpgrade
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXX; Content-Length:***;
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする

	名称	タイプ	最大長	説明
パラメータ	MD5	string	32	ファームウェアアップグレードパッケージの 32 バイト MD5 値
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***:/passengerFlow/deviceManage/FirmwareCheckAndUpgrade</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <pre>{ "MD5": "97f8cf81cad33a11a1d07723bdd34998" }</pre> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": "Upgrade firmware success, device will reboot" } }</pre>			

	<pre> } } </pre>
--	------------------

1.5.9 ログ取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから指定された時間のログを取得する			
名称	passengerFlow/deviceManage/logGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	logTime	String	100	yyyyMMddHHmmss 形式 注)HHmmss には 0 を入れることができる。ログ取得の基本単位は日。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre> { "isError": true/false "code": "String" "message": "String"(文字列) } </pre>

	logFileN ame	String	64	ファイル名
	logFile	TEXT	最大 1M	ログのテキストデータ
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/deviceManage/logGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": "get log success" } "logFileName": "Application-App-Log_2023-11-23", "logFile":ログ内容 }</pre>			

2.トラフィック統計

2.1 人流ベースパラメータ設定

2.1.1 高さ設定

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターに対して高さを設定する			
名称	passengerFlow/passengerFlowParam/HeightParamSet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	Detection Height	int	4	高度に合法的な範囲を検知する 大型広角装置:190cm~350cm 一般装備:220cm~600cm
	FilterHeight	int	4	フィルター高さ 法定範囲 0~150 単位 cm
	RotateDirection	int	4	回転方向 適正範囲 0~3 0: X 軸 1: Y 軸

				2:X 軸の通路 3:Y 軸のアイル
	RotateAngle	float	4	回転角度の法定範囲 大型広角装置:0~20 度 一般機器:0~50 度
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/passengerFlowParam/HeightParamSet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: { "DetectionHeight":300, "FilterHeight":120, "RotateDirection":0, "RotateAngle":0.0 }			

	受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": "setHightParam success" } }</pre>
--	---

2.1.2 検知領域の設定

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターに検知領域を設定する			
名称	passengerFlow/passengerFlowParam/DetectionAreaParamSet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	leftUpPointX	int	4	検知領域の左上隅の開始座標 X
	LeftUpPointY	int	4	検知領域の左上隅の開始座標 Y
	RightUpPointX	int	4	検知エリア右上のスタート座標 X

	RightUpPointY	int	4	検知エリア右上のスタート座標 Y
	RightDownPointX	int	4	検知エリア右下のスタート座標 X
	RightDownPointY	int	4	検知領域の右下隅の開始座標 Y
	LeftDownPointX	int	4	検知エリア左下のスタート座標 X
	LeftDownPointY	int	4	検知領域の左下隅の開始座標 Y
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***/passengerFlow/passengerFlowParam/DetectionAreaParamSet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <pre>{ "LeftUpPointX":260, "LeftUpPointY":180, "RightUpPointX":340, "RightUpPointY":180,</pre>			

	<pre> "RightDownPointX":360, "RightDownPointY": 240, "LeftDownPointX":260, "LeftDownPointY":240 } 受信例: { "result":{ "isError": false, "code": "0", "message": "setDetectionArea success" } } </pre>

2.1.3 検知ラインの設定

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターへ検知ラインを設定する
名称	passengerFlow/passengerFlowParam/DetectionLineParamSet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;

バリデーションパラメータ	バリデーションパラメータ: Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	DetectionLine	int	4 バイト	注:カメラ左画像の検知ラインの Y 画素値
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/passengerFlowParam/DetectionLineParamSet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: <pre>{ "DetectionLine ":200 }</pre> 受信例: <pre>{</pre>			

	<pre> "result":{ "isError": false, "code": "0", "message": "setDetectionLine success" } } </pre>
--	--

2.1.4 検知方向設定

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターに対して検知方向を設定する			
名称	passengerFlow/passengerFlowParam/DetectionDirectionSet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	DetectionDirection	int	4 バイト	0 : 逆方向 1 : 正方向
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{

				<pre> "isError": true/false "code": "String" "message": "String"(文字列) } </pre>
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/passengerFlowParam/DetectionDirectionSet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <pre> { "DetectionDirection":0 } </pre> <p>受信例:</p> <pre> result { "isError": false, "code": "0", "message": "setDetectionDirection success" } </pre>			

2.1.5 検知スイッチの設定

送信者	データ要求装置
レシーバー	ビジターカウンター

インターフェイス定義				
説明	ビジターカウンターに検知スイッチを設定する			
名称	passengerFlow/passengerFlowParam/DetectionSwitchSet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	DetectionSwitch	int	4	0: 検知オフ 1: 検知オン
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/passengerFlowParam/DetectionSwitchSet Host:127.0.0.1:8020 Content-Length: ***			

	Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: <pre>{ "DetectionSwitch":0 }</pre> 受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": "setDetectionSwitch success" } }</pre>
--	--

2.1.6 HTTP JSON パラメータの設定

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターに HTTP JSON パラメータを設定する
名称	passengerFlow/passengerFlowParam/HttpJsonParamSet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;

バリデーションパラメータ	バリデーションパラメータ: Sign MD5 暗号化(全パラメータを ASCII 順にソート+ ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	Server1	string	-	データプッシュ 1 用サーバアドレス
	Port1	int	32	データプッシュ用ポート番号 1
	Interval1	int	32	データプッシュのためのプッシュ間隔 1
	Enable1	init	32	データプッシュ用イネーブルフラグ 1
	Server2	string	-	データプッシュ 2 用サーバアドレス
	Port2	int	32	データプッシュ用ポート番号 2
	Interval2	int	32	データプッシュのためのプッシュ間隔 プッシュ 2
	Enable2	init	32	データプッシュ用イネーブルフラグ 2
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/passengerFlowParam/HttpJsonParamSet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json;			

	<p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <pre>{ "Server1": "192.168.8.101", "Port1":5001, "Interval1":11, "Enable1":0, "Server2": "192.168.8.102", "Port2":5002, "Interval2":12, "Enable2":0 }</pre> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": "Set http json param success" } }</pre>
--	--

2.1.7 HTTP xml パラメータの設定

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターに対して http xml パラメータを設定する
名称	passengerFlow/passengerFlowParam/HttpXmlParamSet
プロトコル	HTTP Restful POST

ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	PostUrl	string	-	データプッシュ用データアドレス
	HeartUrl	string	-	データプッシュ用ハートビートアドレス
	DeviceId	string	20	データプッシュ用デバイス ID
	Interval	int	32	データプッシュのプッシュ間隔
	IntervalMode	int	32	データプッシュインターバルモード
	RealTimeMode	int	32	データプッシュのためのリアルタイムモード
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ " isError": true/false " code": "String" " message": "String"(文字列) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***:/passengerFlow/passengerFlowParam/HttpXmlParamSet			

	Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: <pre>{ "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 }</pre> 受信例: <pre>{ "result":{ "isError": false, "code": "0", "message": " Set http xmlparam success" } }</pre>
--	---

2.2 人流ベースパラメータ取得

2.2.1 検知高パラメータ取得

送信者	データ要求装置
レシーバー	ビジターカウンター

インターフェイス定義				
説明	ビジターカウンターから検知高パラメータを取得する			
名称	passengerFlow/passengerFlowParam/HeightParamGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	DetectionHeight	int	4	検知高さ 単位 cm 広角:190cm~350cm 一般:220cm~600cm
	FilterHeight	int	4	フィルター高さ 0~150 単位 cm
	RotateDirectionValue	int	4	回転方向 0~3 0: X 軸 1: Y 軸

				2:X 軸通路方向 3:Y 軸通路方向
	RotateAngle Value	float	4	回転角度 単位 度 広角:0.0~20.0 度 一般:0.0~50.0 度
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***//passengerFlow/passengerFlowParam/HeightParamGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": "Get getHightParam success!" } }. "DetectionHeight": 300, "FilterHeight":120, "RotateDirectionValue":0, "RotateAngleValue":1.5 }</pre>			

2.2.2 検知エリア範囲取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから、現在の高さに対応する検知領域の範囲を取得する(異なる検知の高さが設定された場合検知範囲を取得し再設定する必要がある)			
名称	passengerFlow/passengerFlowParam/DetectionAreaRangeGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }

	DetectionAreaMax	JSON	512	<pre>{ "LeftUpPointX":(int), "LeftUpPointY":(int), "RightUpPointX":(int), "RightUpPointY":(int), "RightDownPointX":(int), "RightDownPointY":(int), "LeftDownPointX":(int), "LeftDownPointY":(int) }</pre>
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***/passengerFlow/passengerFlowParam/DetectionAreaRangeGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message": " Get DetectionAreaRange success!" }. "DetectionAreaMax":{ "LeftUpPointX":260, "LeftUpPointY":180,</pre>			

	<pre> "RightUpPointX":340, "RightUpPointY":180, "RightDownPointX":360, "RightDownPointY":240, "LeftDownPointX":260, "LeftDownPointY":240 } 備考 テストエリアの関連パラメータは、左の640*400ピクセルで表示されます。 </pre>
--	--

2.2.3 検知領域取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから検知領域を取得する			
名称	passengerFlow/passengerFlowParam/DetectionAreaGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。

	名称	タイプ	最大長	戻り値 説明
戻り値	result	JSON	255	{ "errorCode": true/false "code": "String" "message": "String"(文字列) }
	DetectionArea	JSON	512	{ "LeftUpPointX":(int), "LeftUpPointY":(int), "RightUpPointX":(int), "RightUpPointY":(int), "RightDownPointX":(int), "RightDownPointY":(int), "LeftDownPointX":(int), "LeftDownPointY":(int) }
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/*passengerFlow/passengerFlowParam/DetectionAreaGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "result":{ "errorCode": false, 			

	<pre> "code": "0", "message": "Get getDetectionArea success!" }. "DetectionArea": { "LeftUpPointX":260, "LeftUpPointY":180, "RightUpPointX":340, "RightUpPointY":180, "RightDownPointX":360, "RightDownPointY":240, "LeftDownPointX":260, "LeftDownPointY":240 } } </pre>
--	---

2.2.4 検知ライン範囲取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターから検知ライン範囲を取得する(注:検知エリアを設定する度に検知ライン範囲が変更されるので、検知ライン範囲の取得と再設定が必要)
名称	passengerFlow/passengerFlowParam/DetectionLineRangeGet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;

バリデーションパラメータ	バリデーションパラメータ: Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
	DetectionLineMin	int	4	テストラインの最小値: 左 Y 座標におけるテストラインの最小値
	DetectionLineMax	int	4	検知ライン最大値: 左 Y 座標における検知ラインの最大値
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/passengerFlowParam/DetectionLineRangeGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置)			

	受信例: <pre>{ "result":{ "isError": false, "code": "0", "message":"Get getDetectionLineRange success!" }, "DetectionLineMin": 100, "DetectionLineMax": 300 }</pre>
--	---

2.2.5 検知ラインの位置取得

送信者	データ要求装置								
レシーバー	ビジターカウンター								
インターフェイス定義									
説明	ビジターカウンターから検知ラインの位置を取得する								
名称	passengerFlow/passengerFlowParam/DetectionLineGet								
プロトコル	HTTP Restful POST								
ポート番号	8020 固定								
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XX; Content-Length:***;								
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする								
パラメータ	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">名称</th> <th style="width: 25%;">タイプ</th> <th style="width: 25%;">最大長</th> <th style="width: 25%;">説明</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	名称	タイプ	最大長	説明				
名称	タイプ	最大長	説明						

	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "errorCode": true/false "code": "String" "message": "String"(文字列) }
	DetectionLine	int	4	テストラインの位置:テストラインは左の Y 座標の値にある
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***:/passengerFlow/passengerFlowParam/DetectionLineGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "result":{ "errorCode": false, "code": "0", "message": "Get getDetectionLine success!" }. " DetectionLine": 200 }			

2.2.6 検知方向取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから検知方向を取得する			
名称	passengerFlow/passengerFlowParam/DetectionDirectionGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	DetectionDirection	int	4	0: 逆方向 1: 正方向

使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/passengerFlowParam/DetectionDirectionGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0", "message":"Get getDetection Direction success!" }, "DetectionDirection": 0 }</pre>
-----	--

2.2.7 検知スイッチの状態取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターから検知スイッチの状態を取得する
名称	passengerFlow/passengerFlowParam/DetectionSwitchGet
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json,

	Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret の暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	DetectionSwitch	int	4	0: オフ 1: オン
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/passengerFlowParam/DetectionSwitchGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例:			

	<pre> { "result":{ "isError": false, "code": "0", "message": " get detection switch success" }. "DetectionSwitch":0 } </pre>
--	--

2.2.8 HTTP JSON パラメータの取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから HTTP の JSON パラメータを取得する			
名称	passengerFlow/passengerFlowParam/HttpJsonParamGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明

	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	Server1	string	-	データプッシュ 1 用サーバアドレス
	Port1	int	32	データプッシュ用ポート番号 1
	Interval1	int	32	データプッシュのためのプッシュ間隔 1
	Enable1	init	32	データプッシュ用イネーブルフラグ 1
	Server2	string	-	データプッシュ 2 用サーバアドレス
	Port2	int	32	データプッシュ用ポート番号 2
	Interval2	int	32	データプッシュのためのプッシュ間隔 プッシュ 2
	Enable2	init	32	データプッシュ用イネーブルフラグ 2
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***//passengerFlow/passengerFlowParam/HttpJsonParamGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre>{ "result":{ "isError": false, "code": "0",</pre>			

	<pre> "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 } </pre>
--	---

2.2.9 HTTP xml パラメータの取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから HTTP xml パラメータを取得する			
名称	passengerFlow/passengerFlowParam/HttpXmlParamGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明

	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	PostUrl	string	-	データプッシュ用データアドレス
	HeartUrl	string	-	データプッシュ用ハートビートアドレス
	DeviceId	string	20	データプッシュ用デバイス ID
	Interval	int	32	データプッシュのプッシュ間隔
	IntervalMode	int	32	データプッシュインターバルモード
	RealTimeMode	int	32	データプッシュのためのリアルタイムモード
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/passengerFlow/passengerFlowParam/HttpXmlParamGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "result":{ "isError": false,			

	<pre> "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 } </pre>
--	---

2.3 人流データインターフェイス

2.3.1 人流データ照会

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	データ要求装置がビジターカウンターに人流データを問い合わせ、ビジターカウンターが問い合わせインデックスを起点とした人流データの累積数(履歴累積数)を返す
名称	passengerFlow/passengerFlowData/PassengerFlowDataAsk
プロトコル	HTTP Restful POST
ポート番号	8020 固定
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化)

	1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	startIndex	int	4	人流データ開始インデックス(法定値 >= 0)
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String" }</pre> (文字列)
	numberOfPassengers	int	4	スタートインデックスからの人流フローデータエントリ数
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(startIndex1001096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/passengerFlowData/PassengerFlowDataAsk</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <pre>{ "startIndex":100 }</pre>			

	受信例: <pre>{ "result":{ "isError": false, "code": "0", "message":"ask passenger flow data success" }, "numberOfPassengers":7259 }</pre>
--	---

2.3.2 人流データの取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから指定された指標範囲の人流データを取得する (People plus traffic は 3 ヶ月分のデータを保存しています)			
名称	passengerFlow/passengerFlowData/PassengerFlowDataGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	startIndex	int	4	開始インデックス(法定値 >= 0)

	getQuantity	int	4	<p>搭乗者データ数取得(法定値>0)</p> <p>実際の搭乗者データ<本パラメータの値であれば、実際の搭乗者データ量を返す</p>
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre>{ "isError": true/false "code": "String" "message": "String"(文字列) }</pre>
	rptQuantity	int	4	パッセンジャーフローデータの実返信数
	PassengerFlowData	JSON	最大3日間のヒストリカルデータ	<pre>[{ "idIndex":(int) "timestamp":(long long) "eventType":(int) "stayTime":(int) }, { "idIndex":(int) "timestamp":(long long) "eventType":(int) "stayTime":(int) }, ]</pre> <p>備考</p>

			<p>idIndex: ID インデックス(ユニーク)</p> <p>timestamp: 消失タイムスタンプ (UNIX タイムスタンプ)</p> <p>eventType: イベントタイプ</p> <table border="1" data-bbox="965 504 1286 801"> <thead> <tr> <th>イベントタイプ コード</th> <th>意味</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>入場</td> </tr> <tr> <td>1</td> <td>退場</td> </tr> <tr> <td>2</td> <td>通過</td> </tr> <tr> <td>3</td> <td>引き返す</td> </tr> <tr> <td>-1</td> <td>無効</td> </tr> </tbody> </table> <p>stayTime: 滞在時間 単位:ms</p>	イベントタイプ コード	意味	0	入場	1	退場	2	通過	3	引き返す	-1	無効
イベントタイプ コード	意味														
0	入場														
1	退場														
2	通過														
3	引き返す														
-1	無効														
<p>使用例</p>	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign: md5Hex(getQuantity100startIndex01096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例: POST:***:/passengerFlow/passengerFlowData/PassengerFlowDataGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例: { "startIndex":0, "getQuantity":100 }</p> <p>受信例: {</p>														

	<pre> "result":{ "isError": false, "code": "0", "message":"ask passenger flow data success" }. "rptQuantity": 2, //(2つのデータのみ、2つのみに返信) "PassengerFlowData":[(パッセンジャーフローデータ) { "idIndex": 0, "timestamp":1565801855(消失タイムスタンプ), "eventType": 0, "stayTime":2500(2500msの間常駐) }. { "idIndex": 1, "timestamp":1565801856, "eventType": 1, "stayTime":2200 }] } </pre>
--	---

2.3.3 人流追跡情報の取得

送信者	データ要求装置
レシーバー	ビジターカウンター
インターフェイス定義	
説明	ビジターカウンターから人流追跡情報を取得する
名称	passengerFlow/passengerFlowDataGet/TrackingResultGet
プロトコル	HTTP Restful POST

ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	timestamp	double	32	タイムスタンプ情報。小数点以下はマイクロ秒のタイムスタンプ。
	totalCnt	int	32	現在のシナリオで追跡される人流フローの総数
	TrackingResultData	JSON	検知エリアにおける人流追跡の結果に関するデータ	[{ "id": (int), // ヒューマントラッキング ID "x_h":(long long), // 頭部座標点 x "y_h": (int), // 頭部座標点 y "x_f":(int), // 足部座標点 x

				<pre> "y_f":(int) // 足部座標点 y }, ...] </pre>
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***//passengerFlow/passengerFlowDataGet/TrackingResultGet</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>受信例:</p> <pre> { "result":{ "isError": false, "code": "0", "message": "Get tracking result success!" }. "timestamp":1700696746.828159, "totalCnt": 2, "TrackingResultData":[{ "id":0, "x_h":344, "y_h":143, "x_f":330, </pre>			

	<pre> "y_f":175 }, { "id":1, "x_h":230, "y_h":231, "x_f":297, "y_f":207 }] } </pre>
--	--

2.3.4 人流統計値の取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから現在の人流統計値を取得する			
名称	passengerFlow/passengerFlowData/DisplayDataGet			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明

	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	enter	int	32	入場者数
	Leave	int	32	退場者数
	pass	int	32	通過者数
	return	int	32	引き返し者数
	stay	int	32	滞留者数
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:****/ passengerFlow/passengerFlowData/DisplayDataGet Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "result":{ "isError": false, "code": "0", "message": "Get display data success!" } }			

	<pre> }. "enter": 2, "leave ": 1, "pass": 5, "return":3, "stay": 2 } </pre>
--	---

2.3.5 人流統計履歴データ取得

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから過去の人流統計値を取得する			
名称	passengerFlow/passengerFlowData/HistoryDataGetFromTimestamp			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(全パラメータを ASCII 順にソート+ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2:MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	startTimestamp	int	32	開始タイムスタンプ
	endTimestamp	int	32	終了タイムスタンプ
戻り値	名称	タイプ	最大長	戻り値の説明

	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	enter	int	32	入場者数
	leave	int	32	退場者数
	pass	int	32	通過者数
	return	int	32	折り返し者数
使用例	ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(パラメータ+1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***//passengerFlow/PassengerFlowDataGet/HistoryDataGetFromTime stamp Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 受信例: { "result":{ "isError": false, "code": "0", "message": "Get history data success" }. "enter": 2, "leave ": 1,			

	<pre> "pass": 5, "return":3 } </pre>
--	--

3.画像スキャンインターフェイス

3.1 リアルタイムでのイメージのキャプチャ

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターからリアルタイムで画像を取り込む			
名称	passengerFlow/ImageAndVideo/CapturePicture			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret で暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre> { " isError": true/false " code": "String" } </pre>

				<pre>"message": "String"(文字列) }</pre>
	imageFileName	String	32	yyyyMMddHHmmssms.jpg 形式 20190814220000025.jpg の場合、2019年8月14日22時0分0秒25ミリ秒時点の画像。
	imageFile	File	640*400	バイナリデータ
使用例	<pre>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換 リクエストメッセージ例: POST:***/passengerFlow/ImageAndVideo/CapturePicture Host:127.0.0.1:8020 Content-Length: *** Accept:application/json; Content-Type:application/json;charset=utf-8, Sign: Calculated from (メッセージヘッダに配置) 送信例: 受信例: { result { "isError": false, "code": "0", "message": "captue picture success!" } }</pre>			

	<pre> "imageName": "20190814220000025.jpg", "imageFile":(画像データ) } </pre>
--	--

3.2 Base64 型式でのイメージのキャプチャ

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターからリアルタイム画像を base64 型式で取り込む			
名称	passengerFlow/ImageAndVideo/CapturePictureBase64			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし、MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	<pre> { "isError": true/false "code": "String" "message": "String"(文字列) } </pre>

	imageFileName	String	32	yyyyMMddHHmmssms.jpg 形式 20190814220000025.jpg の場合、2019年8月14日22時0分0秒25ミリ秒時点の画像。
	imageFile	String	-	Base64 データ
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign: md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:***/passengerFlow/ImageAndVideo/CapturePictureBase64</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <p>受信例:</p> <pre>{ result { "isError": false, "code": "0", "message": "capture picture success!" } "imageFileName": "20190814220000025.jpg", "imageFile": "base64データ" }</pre>			

3.3 Base64 型式でのデバッグイメージのキャプチャ

送信者	データ要求装置			
レシーバー	ビジターカウンター			
インターフェイス定義				
説明	ビジターカウンターから base64 型式のデバッグ画像をリアルタイムで取り込む			
名称	passengerFlow/ImageAndVideo/CaptureDebugPictureBase64			
プロトコル	HTTP Restful POST			
ポート番号	8020 固定			
ヘッダ定義	HTTP 標準パケットヘッダーフィールド Accept:application/json, Content-Type:application/json;charset=utf-8, Sign: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX; Content-Length:***;			
バリデーションパラメータ	バリデーションパラメータ:Sign MD5 暗号化(ServiceSecret を暗号化) 1: 認証トークン ServiceSecret は当事者間で合意されたもの 2: MD5 値は大文字にする			
パラメータ	名称	タイプ	最大長	説明
	なし	なし	なし	なし。MD5 暗号は ServiceSecret のみ暗号化。
戻り値	名称	タイプ	最大長	戻り値 説明
	result	JSON	255	{ "isError": true/false "code": "String" "message": "String"(文字列) }
	imageFileName	String	32	yyyyMMddHHmmssms.jpg 形式

				20190814220000025.jpg の場合、2019年8月14日22時0分0秒25ミリ秒時点の画像。
	imageFile	String	-	Base64 データ
使用例	<p>ServiceSecret:1096931dc28e4ecc9ddcb14e8760d53c</p> <p>Sign:</p> <p>md5Hex(1096931dc28e4ecc9ddcb14e8760d53c), 大文字変換</p> <p>リクエストメッセージ例:</p> <p>POST:****/passengerFlow/ImageAndVideo/CaptureDebugPictureBase64</p> <p>Host:127.0.0.1:8020</p> <p>Content-Length: ***</p> <p>Accept:application/json;</p> <p>Content-Type:application/json;charset=utf-8,</p> <p>Sign: Calculated from (メッセージヘッダに配置)</p> <p>送信例:</p> <p>受信例:</p> <pre>{ result { "isError": false, "code": "0", "message": "capture picture success!" } "imageName": "20190814220000025.jpg", "imageFile": "base64データ" }</pre>			

4.パブリックリターンコード

注:リターンコード 0 Success , < 0 はカメラ本体の動作異常、> 0 インタラクティブコマンド異常

リターンコード名	リターンコード	リターンコード 説明	解決方法
SUCCESS	0	サクセスコール	
CAMERA_INIT_ERR	-1	カメラの初期化に失敗しました	
CAMERA_OPEN_ERR	-2	カメラの開閉に失敗しました	
CAMERA_START_ERR	-3	カメラオープンフロー不良	
CAMERA_SET_FPS_ERR	-4	カメラがフレームレートの設定に失敗した	
CAMERA_SET_MODE_ERR	-5	カメラ設定モードに失敗した	
CAMERA_SET_AUTO_EXPOUSE_ERR	-6	カメラ設定の自動露出に失敗した	
CAMERA_SET_AUTO_GAIN_ERR	-7	カメラがオートゲインの設定に失敗した	
CAMERA_SET_EXPOUSE_ERR	-8	カメラの設定露出に失敗した	
CAMERA_SET_GAIN_ERR	-9	カメラ設定ゲインに失敗	
CAMERA_GET_VER1_ERR	-10	カメラのバージョン番号の取得に失敗しました 1	
CAMERA_GET_VER2_ERR	-11	カメラはバージョン番号 2 の取得に失敗しました	
CAMERA_GET_SN_ERR	-12	カメラが SN 取得に失敗した	
CAMERA_GET_NET_PARAM_ERR	-13	カメラがネットワークパラメータの取得に失敗した	
CAMERA_GET_IMAGE_FAILED	-14	カメラの画像取得に失敗した	

リターンコード名	リターンコード	リターンコード 説明	解決方法
CAMERA_GET_IMAGE_TIMEOUT	-15	画像取得のためのカメラタイムアウト	
CAMERA_GET_PARAM_ERR	-16	カメラが内部パラメータの取得に失敗した	
CAMERA_UPDATE_ERR	-17	カメラのアップグレードに失敗	
CAMERA_RUN_FPS_ERR	-18	カメラフレームレートが異常な動きをしている	
CAMERA_AMBIENT_BRIGHTNESS_TOO_LOW	-19	カメラ操作のための低周囲輝度	
SIGN_METHOD_NOTNULL	1	シグネチャパラメータが空である	シグネチャーのパラメータを追加する
SIGN_TYPE_NOTEXIST	2	シグネチャタイプが存在しない	現在、MD5 のみ対応
SIGN_WRONG	3	シグネチャーエラー	合意された署名ルールに従って署名を生成する
PARAM_WRONG	4	パラメータエラー	パラメータは、合意された法的範囲に設定する必要があります。
SET_PARAM_FAILED	5	パラメータ設定に失敗しました	
GET_PARAM_FAILED	6	パラメータの取得に失敗しました	