

# フェイス&サーモプロトコル

Version: 1.14

## 内容

1. サーバー設定例.....	2
2. EDI コマンド .....	4
3. 機器の操作.....	5
4. スナップショット&画像情報のアップロード .....	11
5. 通過ストラテジー .....	12
2. ストラテジーネットワークプロトコル .....	13
6. 失敗したアクセス記録の時限再送.....	18
設定変更時の結果 .....	19
クエリの結果： .....	20
7. 人事情報の一括インポート.....	21
プロトコルのプロセス .....	21
8. ソフトウェアアップグレード .....	24
9. パラメータ設定 .....	25
10. ログ操作 .....	32
11. デバイス再起動.....	33
12. 機器制御の指示 .....	33
操作指示を出す.....	33
13. 個人情報の比較・アップロード: .....	34

## 1. サーバー設定例

The screenshot shows the 'Megapixel IP Camera' web interface. The 'Setting' menu is open, and the 'Upload Platform' sub-menu is selected. The 'Server' field is set to '192.168.1.9', the 'Port' is '8016', and the 'Heart Interval' is '60'. There are 'Save' and 'Cancel' buttons at the bottom of the form.

### 説明:

フェイス&サーモでは顔認証や検温時の情報をサーバーに送信できます。通信プロトコルは HTTP、フォーマットは JSON です。始めにブラウザ（推奨：Internet Explorer）にフェイス&サーモの IP アドレスを入力し、設定画面にログインしてください。

ログイン ID：admin

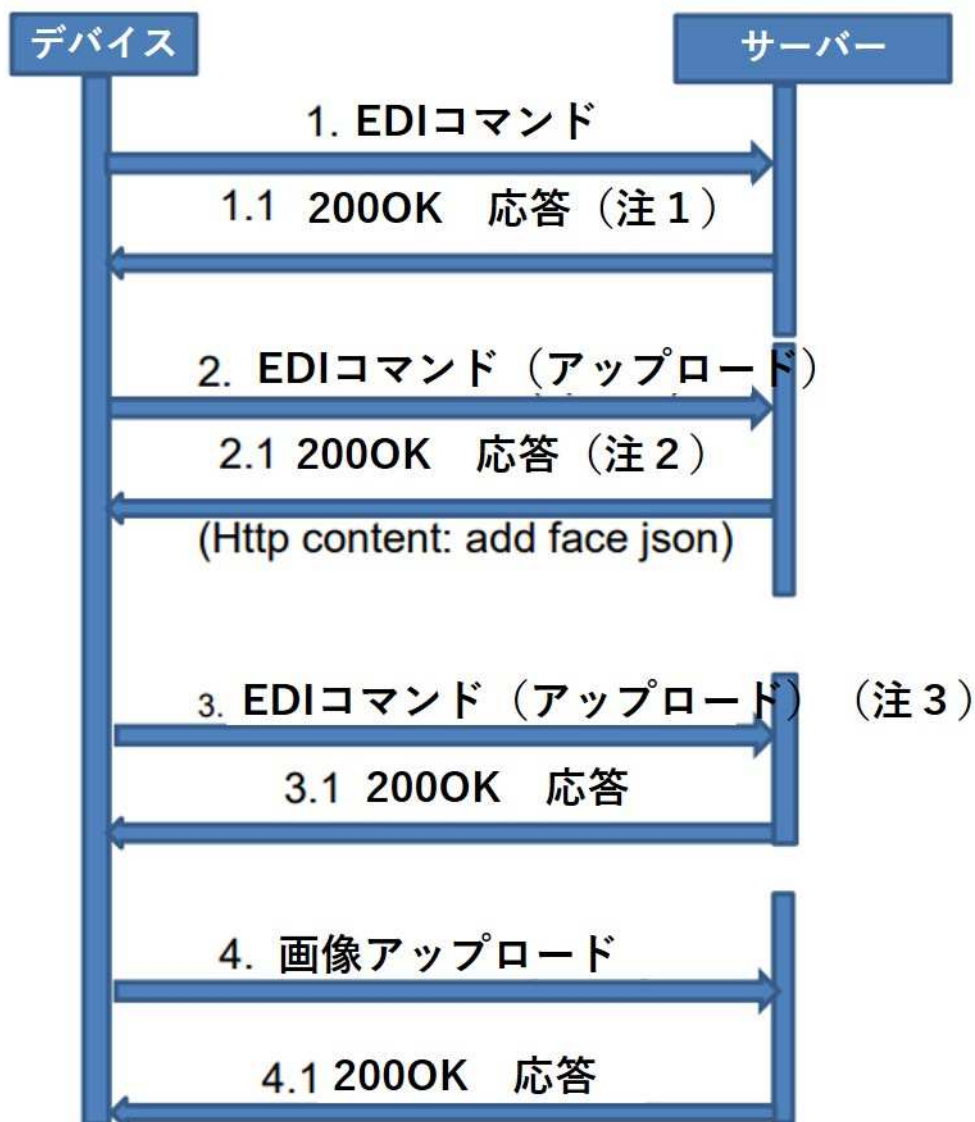
パスワード：admin

**Server**：データの送信先を指定します。example.com などのドメイン名または 192.168.1.9 などの IP アドレス、および/api /face などのアクセスパスを設定します。

**Port**：ポート番号を指定します。初期値は 80 です。

**Heart Interval**：設定した秒数ごとに端末情報を送信します。

例: 顔ライブラリの追加:



(注: 200OK とは、HTTP レスポンスヘッダの 1 行目の値であり、データ本文などではありません。HTTP / 1.1 200OK 200 は正常値を示し、それ以外の値を異常値とします。異常値の場合はデータ本文を読み込みません。)

注 1: サーバーは、HTTP プロトコルを使用して通信するために、コンテンツは json 形式で、プラットフォームは、デバイスの EDI パッケージを受信し、それは 200OK を返信する必要がありますが、処理されるサービスがない場合は、http の内容は空です。

注 2: デバイスはサーバーに EDI コマンドを送信します。サーバーは、EDI 接続を介して http200OK の内容を、顔の追加、顔の削除などの設定が必要な json の内容として返します。

注 3: 顔データベースの追加または削除操作の結果は、次の EDI 時に EDI プロトコルの seq によって通知されます。seq が顔データベース操作プロトコルの seq と一致していれば操作は成功し、一致していなければ操作は失敗する。

プロトコルは、http プロトコルヘッダ+json 形式のコンテンツを使用して、httppost を使用してサーバーにコンテンツを送信します。このドキュメントでは、json 形式のコンテンツについてのみ説明します。

## 2. EDI コマンド

デバイスはプラットフォームサーバーに EDI コマンドを送信し、サーバーは 200OK を返す必要があります。

項目	型	説明
app_version	string	デバイスバージョン番号
cid	string	デバイス識別番号
seq	Int	EDI 送信時間、または顔ライブラリの最後の正常な動作の確認。注 3 をご確認ください。
uptime	Int	本コマンド送信時の UTC 時刻
type	Int	EDI プロトコルタイプ : 3

例 :

```
{  
  "app_version": "1.0.1.1",  
  "cid": "0012b516a202",  
  "seq": 1569497378,  
  "uptime": 1569497378,  
  "type": 3  
}
```

type の数字は以下のコマンドを示しています。

項目	型	説明
type	int	1: スナップショット、アップロード 2: EDI 3: 顔データベースのクエリ結果 4: プッシュ結果のアップグレード 5: ストラテジクエリ結果/コンフィグレーションストラテジー結果 6: 再送時間の問い合わせ結果 7: 一括インポートの結果 8: コンフィグレーションクエリ結果/コンフィグレーション配信確認 9: ログ操作の結果を報告する 10: リモートコントロール

### 3. 機器の操作

EDI コマンドを受信した後、プラットフォームは 200OK を返信し、対応するプロトコルの json 内容を表示します。

#### 1. サーバーアドレスとポートの設定

サーバーは 200OK+次の json コンテンツで応答し、デバイスは、新しい URL アドレスにコンテンツをアップロードします。

項目説明:

項目	型	説明
http_server_url	string	サーバーの URL
http_server_port	Int	サーバーのポート番号

例: .

```
{  
  "http_server_url": "http:// example.com/api/face",  
  "http_server_port": 8080  
}
```

## 2. EDI 期間の設定

プラットフォームは 200OK+以下の json コンテンツで返信し、デバイスが送信する EDI 間隔時間を設定された長さの時間に変更します。

項目説明:

項目	型	説明
time_interval	Int	機器が EDI コマンドを送信する間隔時間

例: .

```
{  
  "time_interval": 3  
}
```

## 3. デバイスの時間設定

サーバーは 200OK+以下の json コンテンツで応答し、設定に応じてデバイスのローカルタイムを変更することができます。

項目説明:

項目	型	説明
time_stamp	Int	サーバーは、UTC 時間を設定します。(単位: 秒)
time_zone	int	サーバーは、タイムゾーンを設定します。(単位: 秒)

例: .

```
{  
  "time_set": {  
    "time_stamp": 1566459660,  
    "time_zone": 28800  
  }  
}
```

#### 4. 顔データベースの操作

顔データベースの操作には、顔データベースの照会、顔データベースの追加・修正、顔データベースの削除などがあります。EDI サイクル設定機能は、顔ライブラリの操作と併用することができ、次のEDI 時間を先送りしたり、前倒ししたりするのに便利です。以下の例では、EDI サイクル設定を持ってきて、必要なければこの項目を削除することができます。

顔データベースのクエリ

項目説明:

項目	型	説明
facelib_manage	object	顔データベース操作の識別
type	int	5 は顔データベースのクエリを示します。
seq	int	サーバー操作シリアル番号

例：.

```
{
  "facelib_manage": {
    "type": 5,
    "seq": 1566358435
  },
  "time_interval": 3
}
```

注：デバイスが受信した問い合わせ要求には、次の EDI での問い合わせ結果が添付されます。

項目説明:

項目	型	説明
cid	string	デバイス識別番号
type	int	4 はクエリの結果を示します。
time	int	本コマンド送信時の UTC 時刻
total_face	int	クエリ結果の数
person_id	string	人員 ID、最大長 35 バイト
person_name	string	人名、最大長 63 バイト
age	int	職員の年齢
sex	string	性別、男性/女性
strategyId	int	人物の戦略 ID
cardNo	String	カード番号、最大長 35 バイト

## Return

```
{
  "cid": "0012b516a202",
  "type": 4,
  "time": 1542430377,
  "total_face": 2,
  "face": [
    {
      "person_id": "20190629153538m3CH6Jdte2DHnANaab",
      "person_name": "Alice",
      "age": 25,
      "sex": "male",
      "strategyId": 1,
      "cardNo": "123015"
    },
    {
      "person_id": "20190629153538MfwkbpkVeRXQVCmiYi",
      "person_name": "Lily",
      "age": 26,
      "sex": "female",
      "strategyId": 2,
      "cardNo": "123016"
    }
  ]
}
```

## 追加または変更

項目	型	説明
facelib_manage	object	顔データベース操作の識別
type	int	1 は顔データベースの追加または変更です
seq	int	サーバー操作シリアル番号
id	string	ID が既に存在する場合は変更操作、存在しない場合は追加操作になります
name	string	人名、最大 35 バイト
age	int	職員の年齢
sex	string	性別、男性/女性



image	String array	人物を追加するには、 <b>base64</b> エンコードされた画像を含める必要があります。修正する場合、この項目は画像が修正されているかどうかで判断することができます。
strategyId	int	その人が属するストラテジー(古いプロトコルとの互換性を保つために、 <b>strategyId</b> は必須フィールドではありません。このフィールドのない人の値はゼロに設定され、ストラテジーは実行されません。) 詳細はストラテジーセクションの説明を参照してください)
cardNo	string	カード番号、最大 <b>63</b> バイト

例 :

```
{
  "facelib_manage": {
    "type": 1,
    "seq": 1566358435,
    "face": {
      "id": "1530327995_hW7UO03",
      "name": "LILEI",
      "age": 26,
      "sex": "male",
      "strategyId": 2,
      "cardNo": "11223344",
      "image": [
        "base64data ..."
      ]
    }
  },
  "time_interval": 3
}
```

注 : デバイスはこの操作の結果を次の EDI で返します。EDI の次の **seq** 項目がこのコマンドの **seq** 項目と等しい場合、操作は成功し、そうでない場合は失敗します。

## 顔データベースの削除

項目説明:

項目	型	説明
facelib_manage	object	顔データベース操作の識別
type	int	3は指定されたID顔データを削除します。 4は全ての顔データベースを削除します。
seq	int	サーバー操作シリアル番号
person_ids	string	指定したIDの人物情報を削除する場合、カンマ区切りで複数のIDを一度に指定し、一度に20件まで削除することができます。 全ての顔ライブラリを削除する場合は、この項目は含まれません。

**Delete the face library corresponding to person\_id, 例 :**

```
{  
  "facelib_manage": {  
    "type": 3,  
    "seq": 1566358435,  
    "person_ids": "1530327995_hW7UO03, 1530327995_hW7UO04"  
  },  
  "time_interval": 3}
```

**Delete all face database, 例 :**

```
{  
  "facelib_manage": {  
    "type": 4,  
    "seq": 1566358435  
  },  
  "time_interval": 3  
}
```

注 : デバイスはこの操作の結果を次の EDI で返します。EDI の次の seq 項目がこのコマンドの seq 項目と等しい場合、操作は成功し、そうでない場合は失敗します。

## 4. スナップショット&画像情報のアップロード

デバイスは人員識別を検出し、関連するスナップショット情報をサーバーにアップロードします。サーバーはスナップショット情報を受信し、確認のために **200 OK HTTP** ヘッダを返します。HTTP はデータを運ぶ必要はありません。

備考。機器が写真をアップロードする際の **HTTP** ヘッダと **JSON** コンテンツは **2** 段階で送信します。まず、**Expect: 100-continue** フィールドを持つ **HTTP** ヘッダを送信し、サーバーからの応答を待ち、次に **JSON** を送信し、最後にサーバーからの **200OK** を待ちます。

サーバーは **Expect: 100-continue** フィールドに応答することができず、デバイスはその後の **JSON** をアップロードします。 .

項目説明:

項目	型	説明
cid	string	デバイス ID
type	int	1: スナップショットアップロード結果
time	int	コマンド送信時の UTC 時間
faces	String array	base64 エンコードされたキャプチャーまたはアップロード画像
result	Obj	マッチした顔の情報
person_id	string	人事 ID, 最大長 35 バイト
person_name	string	人名、最大長 63 バイト
age	int	職員の年齢
sex	string	性別、男性/女性
confidence	float	類似度/信頼度、一致しない場合は 0
person_type	int	0: 比較に失敗しました 1: 比較成功 2: 間隔をおいて撮影した画像
temperature	float	温度値 (小数点以下 1 桁まで正確)
facemask	int	0: マスクなし 1: マスク着用

例：

```
{
"cid": "294300000111",
"time": 1568101332,
"faces": [
"base64..."
],
"result": {
"age": 0,
"confidence": 0.9300000071525574,
"person_id": "1567910643_dgQdv2x9",
"person_name": "胡昊",
"person_type": 1,
"sex": " male"
"temperature":36.5,
"facemask":1
},
"type": 1,
}
```

## 5. 通過ストラテジー

### ストラテジールール

ストラテジーを策定することで、人員の通過を柔軟に管理することが可能です。各ストラテジーには複数の時間ルール（ルール）があり、ルールは開始日と終了日、週、複数の時間帯から発効するように設定でき、時間帯内での許可・禁止が可能です。

人物情報を編集することで、人物ごとに1つずつストラテジーを割り当てることができます。デバイスは対応する人物を認識した後、対応するストラテジーを実行します。現在時刻がストラテジーの実行時刻に含まれる場合、ストラテジーのルールに従って実行されます。人物が属するストラテジーが存在しない場合やストラテジーが有効になっていない場合は、通過を禁止します。

### ストラテジー互換設計。

ストラテジーのない旧バージョンの Protokol との互換性を考慮し、**strategyId** の 0 を特別予約値として使用しています。**strategyId = 0** であれば、ストラテジーによる制限を受けません。**strategyId = 0** のストラテジーは失敗します。ただし、人物を追加する際には、人物情報の **strategyId** を 0 に設定できます。このように設定しておけば、その人物はストラテジー機構によって制限されません。

### ストラテジー配信の追加。

ストラテジーの変更は、完全に更新してカバーする方法を採用しており、再発行の度にそれまでのポリシーは全て破棄されます。したがって、ストラテジーの変更は一度に完全に配信される必要があります。正常に追加

されたストラテジーのみが配信の度に保持されるので、一度に配信されたコンテンツ内のストラテジーがどれも合法でなければ、ローカルストラテジーはクリアされます（例えば、**strategyId=0** を 1 つだけ発行するなど）。

配信が完了すると、E D I 間隔に関係なく、各ストラテジーが正常に追加されたかどうか（ストラテジー I D にパースできないストラテジーを除く）を即座に報告します。その結果は、デバイスが新たに開始した接続で報告されます。

## 2. ストラテジーネットワークプロトコル

項目	型	説明
strategy_manage	{}	ストラテジー操作用の固定値
strategyList	{}	ストラテジーコンテンツリスト(最大 16 個)
strategyId	int	ストラテジーのユニーク ID
name	String (最大 31 文字)	ストラテジー名
state	int	ストラテジーステータス 1 : オープン 0 : クローズ
strategyRuleList	Array	ストラテジーのタイムルールのセット(各ストラテジー16個まで)
ruleId	int	ルール ID
begDate	string	開始日 フォーマット例: 2019/01/01
endDate	string	終了日 フォーマット例: 2019/11/11
timeRange	string	時刻範囲: 0-24、「,」区切り、最大 16 範囲 例: 08: 00: 00-08: 15: 00, 08: 15: 00-12: 00: 00
week	string	週範囲: 1-7, 「,」区切り、7 は日曜日 例: 1, 2, 6, 7
type	int	タイムルール内での通過の許可: 1 : 許可 0 : 許可しない

ストラテジー発行例 :

```
{
  "strategy_manage": {
    "strategyList": [
      {
        "strategyId": 1,
        "name": " Test strategy 1",
        "state": 1,
        "strategyRuleList": [
          {
```

```
"ruleId": 1,
"begDate": "2020/03/01",
"endDate": "2019/6/30",
"timeRange": "17:30:00-23:59:59,00:00:00-08:30:00",
"week": "1,2,3,4,5,6,7",
"type": 0
},
{
"ruleId": 2,
"begDate": "2020/03/01",
"endDate": "2019/6/30",
"timeRange": "08:30:00-17:30:00",
"week": "1,2,3,4,5,6,7",
"type": 1
}
]
},
{
"strategyId": 2,
"name": " Test strategy 2",
"state": 1,
"strategyRuleList": [
{
"ruleId": 1,
"begDate": "2020/03/01",
"endDate": "2019/6/30",
"timeRange": "17:30:00-23:59:59,00:00:00-08:30:00",
"week": "1,2,3,4,5,6,7",
"type": 0
},
{
"ruleId": 2,
"begDate": "2020/03/01",
"endDate": "2019/6/30",
"timeRange": "08:30:00-17:30:00",
"week": "1,2,3,4,5,6,7",
"type": 1
}
]
}
}
```

.....

```
]
}
}
```

ストラテジー操作結果の応答

項目	型	説明
cid	String	デバイス ID
type	int	デバイス応答動作結果。固定値 : 7
time	int	時間
strategy_manage	{}	固定フィールド
resultList	array	操作結果一覧
strategyId	int	ストラテジーのユニーク ID
result	int	操作結果: 0: 成功 1: 失敗

例 :

```
{
  "cid": "042082065161",
  "type": 7,
  "time": 1584424711,
  "strategy_manage": {
    "resultList": [
      {
        "strategyId": 1,
        "result": 0
      },
      {
        "strategyId": 2,
        "result": 0
      },
      ...
    ]
  }
}
```

サーバーは、デバイスの EDI 時間に応答してストラテジークエリ要求を発行します。

項目	型	説明
strategy_manage	{}	固定フィールド
operType	int	固定フィールド、値は 2 に固定

ストラテジークエリ発行例：

```
{
  "strategy_manage":
  {
    "operType": 2
  }
}
```

デバイスは、次回の EDI 時にポリシーリストの内容を送信します。

項目	型	説明
cid	String	デバイス ID
type	int	クエリ結果に対するデバイスの返信。固定値：7
time	int	時間
total_strategy	int	ストラテジー数
strategyList	Array	ストラテジーリスト
strategyId	int	ストラテジーのユニーク ID
name	string	ストラテジー名
state	int	ストラテジーステータス： 1：オープン 0：クローズ
strategyRuleList	Array	ストラテジーのタイムルールのセット(各ストラテジー16個まで)
ruleId	int	ルール ID
begDate	string	開始日 フォーマット例: 2019/01/01
endDate	string	終了日 フォーマット例: 2019/11/11
timeRange	string	時刻範囲: 0-24、「,」区切り、最大 16 範囲 例: 08: 00: 00-08: 15: 00, 08: 15: 00-12: 00: 00
week	string	週範囲: 1-7, 「,」区切り、7 は日曜日 例: 1, 2, 6, 7
type	int	タイムルール内での通過の許可： 1：許可 0：許可しない

クエリ返信例：

```
{
  "cid": "294300000005",
  "type": 7,
```



```
"time": 1542430377,
"total_strategy": 2,
"strategyList": [
  {
    "strategyId": 1,
    "name": "策略 1",
    "state": 1,
    "strategyRuleList": [
      {
        "ruleId": 1,
        "begDate": "2020/03/01",
        "endDate": "2020/04/02",
        "timeRange": "07:30:40-10:30:20",
        "week": "6,7",
        "type": 0
      },
      {
        "ruleId": 2,
        "begDate": "2020/03/03",
        "endDate": "2020/05/04",
        "timeRange": "07:30:00-08:30:00,07:30:00-08:30:00",
        "week": "1,2,3,4,5",
        "type": 1
      }
    ]
  },
  {
    "strategyId": 2,
    "name": "test_strategy_2",
    "state": 0,
    "strategyRuleList": [
      {
        "ruleId": 1,
        "begDate": "2020/03/01",
        "endDate": "2020/04/02",
        "timeRange": "07:30:40-10:30:20",
        "week": "1,2,3",
        "type": 1
      },
      {
        "ruleId": 2,
        "begDate": "2020/03/03",
```

```

        "endDate": "2020/05/04",
        "timeRange": "8:59:00-9:30:00,18:29:00-19:00:00",
        "week": "4,5,6",
        "type": 1
    }
]
},
.....
]
}

```

## 6. 失敗したアクセス記録の時限再送

通過失敗時のキャプチャ画像は保存され、選択した時間に再送信されます。デバイスは最大 **2000** 枚の通過失敗時の画像をローカルに保存することができ、上限を超えると最も古いものは破棄されます。

再送信には **2** つのモードがあります。:

1. スケジュールされた再送モード: スケジュールされた時間に達した後、古いものから新しいものへと順に **1** つずつ再送します。再送失敗時はスキップして次の送信を続けます。失敗した画像は保持されます。
2. アイドル再送信モード: **3** 分以内に新しいパスレコードが生成されなかった場合、古いものから新しいものへと順に **1** 枚ずつ再送信します。再送失敗時は、次の試行を待ちます。アップロード処理中、通りすがりの人が映った場合、再送を停止し、新たに **3** 分待ちます。

注: 再送中の画像情報の時間フィールドは、時間の再計算を行わず、パスレコードが生成された時間のままになります。

### 設定変更:

項目	型	説明
reupload	{ }	再送操作の固定フィールド
type	int	操作タイプ: 1 設定変更 2 クエリ設定
seq	int	操作シリアル番号
reupload_time	String	定時再送時間、時:分:秒
mode	int	再送方法。 1: スケジュール再送 2: 3分間のアイドル再送-[初期値]

設定例 :

```
{
  "reupload": {
    "type": 1,
    "seq": 11223456,
    "reupload_time": "19:45:10",
    "mode": 2
  }
}
```

## 設定変更時の結果

設定を変更した後は、すぐに結果が返ってきます。

項目	型	説明
type	Int	固定値 : 8
cid	string	デバイス ID
time	int	送信時間
reupload	{ }	固定フィールド
seq	int	送信時のシリアル番号
errcode	int	0 : 成功 1 : 入力パラメータエラー

結果例 :

```
{
  "cid": "042071005218",
  "type": 8,
  "time": 1586159775,
  "reupload": {
    "seq": 11223456,
    "errcode": 0
  }
}
```

クエリ例 :

```
{
  "reupload": {
    "type": 2
  }
}
```

## クエリの結果：

クエリ結果はすぐに返されます。

項目	型	説明
type	Int	固定値：8
cid	string	デバイスID
time	int	送信時間
reupload	{ }	固定フィールド
reupload_time	String	定時再送時間、時：分：秒
mode	int	再送方法。 1：スケジュール再送 2：3分間のアイドル再送-[初期値]
records_num	int	デバイスのローカルに保存されている再送回数

```
{  
  "cid": "042071005218",  
  "type": 8,  
  "time": 1586151346,  
  "reupload": {  
    "reupload_time": "00:00:00",  
    "mode": 2,  
    "records_num": 28  
  }  
}
```

## 7. 人事情報の一括インポート



### プロトコルのプロセス

1. デバイスは EDI を送信し、サーバーが EDI に応答すると、人事情報を送信します。
2. デバイスは人事情報データベースの操作が完了した後、直ちに EDI 間隔に関係なく操作結果を送信します。
3. 操作結果受信後、サーバーは次の人事情報を送信し、デバイスは人事情報を更新します。これらのやり取りが完了後、デバイスの EDI 間隔は再開します。

備考：人事情報の一括処理は短期コネクションです。デバイスが一度送信し、サーバーが返信して完結します。デバイスは、各人事情報操作の結果を報告します（人事 ID フィールドのパーズに失敗した場合を除く）。

### サーバーからの人事情報

サーバーは人事情報と操作内容を一括して送信します。人事情報の ID、ストラテジーID、写真は必須項目です。ストラテジーが不要の場合はストラテジーID (strategyId) を 0 にしてください。

項目	型	説明
facelib_batch	{}	固定フィールド
personList	Array	人事リスト、複数の人事情報の一括運用をサポート。最大 20 人分を送信できます。

operType	Int	操作タイプ: 1 : 人事情報が存在しない場合は追加処理を行います。存在する場合は変更処理を行います。 3 : 削除(削除した場合、人事 ID のみ有効)
id	String(最大 35 文字)	人事 ID。ユニーク ID です。
cardNo	String(最大 35 文字)	カード番号
name	String(最大 63 文字)	名前
age	Int	年齢
sex	String	性別、男性/女性
strategyId	int	人事用ストラテジーID
image	String	Base64 エンコードされた画像 (備考: シングルフェイスのインポートとは異なり、ここでは角括弧は使用されません。)

例 :

Platform transfer personnel information, 例 :

```
{
  "facelib_batch": {
    "personList": [
      {
        "operType": 1,
        "id": "1530327995_hW7UO01",
        "cardNo": "123",
        "name": "name1",
        "age": 24,
        "sex": "female",
        "strategyId": 1,
        "image": "base64data ..."
      },
      {
        "operType": 3,
        "id": "1530327995_hW7UO02"
      },
      {
        "operType": 1,
        "id": "1530327995_hW7UO03",
        "cardNo": "789",
        "name": "name3",
        "age": 16,
        "sex": "female",
        "strategyId": 3,

```

```

        "image": "base64data ..."
    },
    .....
]
}
}

```

## デバイスの応答

デバイスがインポート結果に応答する

項目	型	説明
cid	String	デバイス ID
type	int	デバイス返信インポート結果 固定値 : 9
time	int	時間
facelib_batch	{}	固定フィールド
resultList	array	操作結果一覧
operType	Int	操作タイプ: 1 : 人事情報が存在しない場合は追加処理を行います。存在する場合は変更処理を行います。 3 : 削除(削除した場合、人事 ID のみ有効)
id	String	人事 ID。ユニーク ID です。
result	int	操作結果 0: 成功 1: 失敗 2: 人事情報が間違っています 3: 画像が間違っています 4: スタッフプールがいっぱいです 5: インポート結果の取得がタイムアウト 6: システムビジー

例 :

```

{
  "cid": "042082065161",
  "type": 9,
  "time": 1584424711,
  "facelib_batch": {
    "resultList": [
      {
        "operType": 1,
        "id": "1530327995_hW7UO01",
        "result": 0
      }
    ]
  }
}

```

```

    },
    {
        "operType":3,
        "id": "1530327995_hW7UO02",
        "result":1
    },
    .....
]
}
}

```

## 8. ソフトウェアアップグレード

プロトコルプロセス:

1. デバイスは正常に EDI に報告し、サーバーは EDI の応答でアップグレードパッケージのバージョン番号とダウンロードアドレスを発行します。
2. デバイスは EDI とスナップショットのアップロードを中断します。
3. デバイスはバージョン番号を確認し、バージョン番号が異常な場合は、エラーメッセージを返し、EDI とスナップショットのアップロードを再開します。
4. デバイスは、アップグレードパッケージを取得するためにダウンロードアドレスに接続します。ダウンロードに失敗した場合は、エラーメッセージを返し、EDI とスナップショットのアップロードを再開します。
5. アップグレードパッケージを受信した後、デバイスは確認情報をサーバーに返信します。
6. デバイスはアップグレードされ、再起動します。

### Upgrade commands issued by the platform

項目	型	説明
fw_upgrade	{}	アップグレード用の固定フィールド
version	String	バージョン番号
download_url	String	アップグレードパッケージダウンロードアドレス

例 :

```

{
    "fw_upgrade": {
        "version": "V1.2.1.2",
        "download_url": "http://xxxxxxxx/xxx.zip"
    }
}

```

デバイスはアップデート後、EDI を待たずにすぐに復帰します。

項目	型	説明
----	---	----



app_version	string	デバイスのバージョン番号
cid	string	デバイスID
seq	int	EDI 送信時間または顔データベースの最後の正常な操作の確認については、詳細は注 3 を参照してください。
uptime	int	デバイス起動時の UTC 時間
type	int	返信メッセージの更新。固定値：6
error	Int	エラーコード: 1: アップグレードサーバに接続できません 2: ファイルが取得できません 3: ファイルサイズが大きすぎます 4: アップグレード接続が中断されました 5: アップグレードファイルが一致しません 6: アップグレード後の再起動の準備完了

例：

```
{
  "app_version": "1.0.1.1",
  "cid": "0012b516a202",
  "seq": 1569497378,
  "uptime": 1569497378,
  "type": 6,
  "error": 6
}
```

## 9. パラメータ設定

プロセス:

1. デバイスは EDI を報告し、サーバーは EDI に返信する際に設定パラメータを送信します。
2. デバイスアプリケーションの構成パラメータは、次回 EDI は確認のためのサーバーに返信されます。

### Parameter delivery

項目	型	説明
config	{}	設定パラメータの固定フィールド
opType	Int	操作タイプ: 0: 設定パラメータ 1: クエリパラメータ (以下のフィールドはクエリ中には入力されません)

seq	int	操作シリアル番号、opType=0 の場合、必須項目
face_rec	{}	顔認識設定項目
face_cap	{}	設定項目のキャプチャ
facemask	{}	マスク設定項目
temperature	{}	体温設定項目
relay	{}	リレー構成項目
misc	{}	その他の項目

#### 顔認識の設定 “face\_rec”:{}

項目	型	説明
enable	int	顔認識スイッチ: 0: オフ 1: オン-[初期値]
voice	int	音声出力 0: 音声出力なし 1: 音声出力("authorized", "unauthorized") 2: 異常時のみ音声出力("Unauthorized")-[初期値]
relay*	int	リレー出力: 0: オフ 1: オン認証なしでドアオープン不可[初期値]
liveness	float	顔認証時の生態認識値 (0-1, 値が大きいほど生態認識が強い)-[初期値 0.2]
score	float	顔認識スコア (類似度) (0-100) -[初期値 70]
times	int	同一人物であると判定された回数の閾値 (1 ~ 10) この回数に満たない場合、別人であると判定されます。
		超過 -[初期値 3]
distance	int	認識距離のしきい値。 0: 0.5m 1: 1m-[初期値] 2: 1.5m 3: 2m 4: 2.5m
face_expos	int	顔の露出しきい値(0-100)-[ 初期値 50]

顔のキャプチャ “face\_cap”:{}

項目	型	説明
interval	int	写真を撮影する時間間隔 (単位 : 秒) [初期値 3]
goodness	float	キャプチャ画質パラメータ (0-1、値が大きいほど画質が良い) -[初期値 0.7]
blur	float	キャプチャ画像のぼかしパラメータ (0-1、値が小さいほど鮮明) -[初期値 0.3]
time_overlay	int	スナップ画像に時刻情報のオーバーレイ: 0: オフ-[初期値] 1: オン
upload_mode	int	一般的なスナップショットのアップロードモード: 1: データベース登録済の人物のみアップロード 2: 登録済の人物と訪問者をアップロード-[初期値]

マスク設定 “facemask”:{}

項目	型	説明
enable	int	マスク検知スイッチ: 0: オフ 1: オン-[初期値]
mode	int	マスク検出モードオプション: 0: マスクなし(エピデミックモード)-[初期値] 1: マスク着用(ノンエピデミックモード)
voice	int	音声出力 0: 音声出力なし 1: 異常時のみ音声出力(エピデミックモードでは "マスクの着用をお願いします。", ノンエピデミックモードでは "マスクを外してください")-[初期値]
relay	int	リレー出力: 0: オフ-[初期値] 1: マスク検知異常時はドアを開けない

体温の調整 “temperature” : {}

項目	型	説明
enable	int	体温検知スイッチ: 0: オフ 1: オン-[初期値]
high_threshold	float	体温のしきい値-[初期値 37.3℃]
voice	int	音声出力: 0: 音声出力なし 1: 体温異常時のみ音声出力-[初期値] 2: 正常値、異常値どちらも音声出力
relay	int	リレー出力 0: オフ 1: 体温異常時にはドアを開けない [初期値]
offset	float	温度補正係数
unit	int	温度単位: 0: 摂氏-[初期値] 1: 華氏

リレー出力 “relay” : {}

項目	型	説明
duration	int	リレー信号の継続時間 (1-100) (単位 : 100 ミリ秒) - [初期値 2]
state	int	リレーの動作状態: 0: 状態が正常な場合に接続します-[default] 1: 状態異常時に接続

その他 “misc” : {}

項目	型	説明
version	int	設定パラメータのバージョン (問い合わせのみ可能 で、設定はできません)

パラメータ設定例：

```
{
  "config": {
    "opType": 0,
    "seq": 123456,
    "face_rec": {
      "enable": 1,
      "voice": 1,
      "relay": 1,
      "liveness": 0.2,
      "score": 70,
      "times": 5,
      "distance": 1,
      "face_expos": 50
    },
    "face_cap": {
      "interval": 10,
      "goodness": 0.7,
      "blur": 0.3,
      "time_overlay": 0,
      "upload_mode": 1
    },
    "facemask": {
      "enable": 1,
      "mode": 0,
      "voice": 1,
      "relay": 0
    },
    "temperature": {
      "enable": 1,
      "high_threshold": 37.3,
      "voice": 1,
      "relay": 1,
      "offset": 0,
      "unit": 0
    },
    "relay": {
      "duration": 2,
      "state": 0
    }
  }
}
```

}

デバイスはエラーコードを返します。:

デバイスは EDI 間隔に関係なく、直ちにエラーコードを報告します。

項目	型	説明
cid	string	デバイス ID
type	int	固定値 : 10
time	int	送信時間
config	{}	固定フィールド
seq	int	配信された seq、もし配信された seq に解決策がなければ、戻り値にはフィールドがありません。
errcode	int	エラーコード: 0: 成功 1: 入力パラメータエラー(seq と opType フィールドの入力パラメータエラーもここで分類されます) 2: 設定失敗

例 :

```
{  
  "cid": "042082065161",  
  "type": 10,  
  "time": 1585047588,  
  "config": {  
    "seq": 1584940620,  
    "errcode": 0  
  }  
}
```

クエリ例 :

```
{  
  "config": {  
    "opType": 1  
  }  
}
```

クエリ結果は、次回の EDI 時間に報告されます。

項目	型	説明
type	int	固定値 : 10 設定クエリが戻る際に
cid	string	デバイス ID
time	int	時間

クエリ結果例 :

```
{
  "cid": "042082065161",
  "type": 10,
  "time": 1584945022,
  "config": {
    "face_rec": {
      "enable": 1,
      "voice": 1,
      "relay": 1,
      "liveness": 0.20,
      "score": 70.00,
      "times": 5,
      "distance": 1,
      "face_expos": 50
    },
    "face_cap": {
      "interval": 10,
      "goodness": 0.70,
      "blur": 0.30,
      "time_overlay": 0,
      "upload_mode": 1
    },
    "facemask": {
      "enable": 1,
      "mode": 0,
      "voice": 1,
      "relay": 0
    },
    "temperature": {
      "enable": 1,
      "high_threshold": 37.30,
      "voice": 1,
      "relay": 1,
      "offset": 0.00,
      "unit": 0
    },
    "relay": {
      "duration": 2,
      "state": 0
    }
  },
}
```

```

    "misc": {
        "version": 1
    }
}

```

## 10. ログ操作

操作指示を出す

項目	型	説明
facelog	{ }	固定フィールド
opType	int	操作タイプ: 1: ログのクリア
seq	int	操作シリアル番号

例 :

```

{
    "facelog": {
        "opType": 1,
        "seq": 123456
    }
}

```

戻り値の確認

項目	型	説明
cid	string	デバイス ID
type	int	固定値 : 11 ログ操作で返される
time	int	送信時間
facelog	{ }	固定フィールド
errcode	int	0: 成功 1: 入力エラー 2: 失敗
seq	int	配信された seq、もし配信された seq に解決策がなければ、戻り値にはフィールドがありません。



例：

```
{
  "cid": "042082065161",
  "type": 11,
  "time": 1584945022,
  "facelog": {
    "seq": 123456,
    "errcode": 0
  }
}
```

## 11. デバイス再起動

例：

```
{
  "reboot": 1
}
```

## 12. 機器制御の指示

操作指示を出す

項目	型	説明
control	{}	固定フィールド
opType	int	操作タイプ: 1: リレー出力を一度切り替えます。
seq	int	操作シリアル番号

例：

```
{
  "control": {
    "seq": 12123,
    "opType": 1
  }
}
```

## Return confirmation

項目	型	説明
cid	string	デバイス I D
type	int	設定が戻ってきた 固定値 : 12
time	int	送信時間
facelog	{}	固定フィールド
errcode	int	0: 成功 1: 入力エラー 2: 失敗
seq	int	配信された seq、もし配信された seq に解決策がなければ、戻り値にはフィールドがありません。

例 :

```
{  
  "cid": "042082065161",  
  "type": 12,  
  "time": 1584945022,  
  "control": {  
    "seq": 12123,  
    "errcode": 0  
  }  
}
```

## 13. 個人情報の比較・アップロード:

注 : 本契約は ID カードスワイピングデバイスのみ対応しています。

デバイス ID が比較された後、関連するデータがプラットフォームにアップロードされます。プラットフォームは情報を受信し、確認のために 200 OK の HTTP ヘッダを返し、HTTP ボディはデータを運ばない。

項目の説明:

項目	型	説明
ldcard_Match	{}	個人 ID の比較アップロード固定フィールド
cid	String	デバイス I D
result	int	比較結果: 1: 比較成功 0: 比較失敗
userName	String	名前
idCardNum	String	ID 番号
nationality	String	国籍
gender	String	性別、男性/女性

birthdate	String	生年月日: (形式: 1995-03-18)
idCardAddr	String	ID カードアドレス
issuingAuthority	String	発行権限 (ID カード発行権限)
validityPeriod	String	有効期限(ID カードの有効期限 形式: 2012.01.12-2020.01.12)
idCardPhoto	String	ID ポートレート jpg 写真 (base64 エンコード)
caputurePhoto	String	顔の JPG 写真 (base64 エンコード)

例 :

{

```

  "Idcard_Match": {
    "cid": "294300000111",
    "result": 0,
    "userName": "Caocao",
    "idCardNum": "360428198010011110",
    "nationality": "meng",
    "gender": "male",
    "birthdate": "1980-10-01",
    "idCardAddr": "421 Jingyang Avenue, Dongcheng District, Beijing",
    "ssuingAuthority": "Dongcheng District Branch of Beijing Public Security Bureau",
    "issuingAuthority": "2010.01.12-2050.01.12",
    "issuingAuthority": "Dongcheng District Branch of Beijing Public Security Bureau",
    "idCardPhoto": ["base64data ..."],
    "caputurePhoto": ["base64data ..."]
  }
}

```