

8 インチ検温・顔認証端末
API 仕様書
V1.1

目 次

1.	概要	4
1.1	通信プロトコル.....	4
1.2	暗号化	4
1.3	応答形式	4
2.	ログイン	5
2.1	ローカル管理サーバへのログイン.....	5
2.2	初期化バージョン.....	5
2.3	サービスログイン.....	6
2.4	パスワードの変更.....	6
3.	言語管理	7
3.1	言語一覧の取得.....	7
3.2	システム言語一覧の取得.....	8
4	登録者管理	8
4.1	登録者一覧の取得.....	8
4.2	新規登録	10
4.3	更新	11
4.4	削除	11
4.5	登録者データのエクスポート.....	12
5	入退記録（履歴）	12
5.1	入退記録一覧の取得.....	12
5.2	指定削除	14
5.3	一括削除	14
5.4	入退記録のエクスポート.....	14
6	端末管理	15
6.1	設定情報の取得.....	15
6.2	設定変更	19
6.3	検温・顔認証APPのソフトウェア更新.....	20
6.4	システム時間の変更.....	21
6.5	logoファイルのアップロード.....	21
6.6	スクリーンセーバーのアップロード.....	22
7	勤怠管理	22

7.1 勤怠設定情報の取得.....	22
7.2 打刻データの検索.....	23
7.3 シフト変更	24
7.4 打刻データのエクスポート.....	25
7.5 端末情報の取得	26
8. 入退記録のプッシュ (Callback)	26
8.1 プッシュ機能及びURL設定.....	26
8.2 リクエストパラメータ.....	27
8.3 APIインターフェイス認証.....	27
8.4 Jsonデータフォーマット.....	29
8.5 再送処理	29
8.6 Springbootサーバ側設定 (例)	30

1. 概要

この API 仕様書は外部システムが 8 インチ検温・顔認識タブレット端末に内蔵されているローカル管理サーバにアクセスするための通信インターフェース仕様であり、通信プロトコル、データフォーマット、暗号化方法などが含まれます。

1.1 通信プロトコル

- ◆ http または https
- ◆ Content-Type : application / json または Content-Type : x-www-form-urlencoded
- ◆ API 通信ポート : 8080

1.2 暗号化

スタンドアロンバージョンはログイン認証を除いてローカルエリアネットワークを利用するため、他の通信は当面暗号化されません。

1.3 応答形式

接口統一戻り値为：

```
{  
  "success":true,  
  "code":0,  
  "desc":"成功",  
  "data":{}  
}
```

パラメーター	種類	詳細説明
success	Boolean	True : 成功 False : 失敗
code	Int	エラーコード
desc	String	エラーメッセージ
data	Object	◆ オブジェクトまたはデータ ◆ Success の値は false の場合は null

2. ログイン

2.1 ローカル管理サーバへのログイン

POST: /api/admin/login

リクエストパラメータ: x-www-form-urlencoded

パラメータ	タイプ	内容
password	String	管理者パスワード

注: password はソルト処理(暫定案: YT)後に MD5 の暗号化処理を行います。

例: デフォルトパスワード 123456 の場合

- デフォルトパスワード: 123456
- ソルト処理された後のパスワード: 123456YT
- 123456YT に対して MD5 暗号化処理した結果 = 6cffc03a4295e711d0e9af2328f05a44

戻り値:

```
{  
  "msg": "成功",  
  "code": 0,  
  "data": {  
    "loginServer": true, //管理サーバへのログイン状態、false 且つ initType=2、ログイン画面  
    に遷移する  
    "initType": 2 //0 タイプ未選択の場合はスタンドアロン版とクラウド版の選択画面に遷移す  
    る  
  },  
  "success": true  
}
```

2.2 初期化バージョン

POST: /api/admin/initType

リクエストパラメータ: x-www-form-urlencoded

パラメータ	タイプ	内容
initType	Int	1 : スタンドアロン版 2 : クラウド版

2.3 サービスログイン

POST: /api/admin/loginServer

リクエストパラメータ: `application/json`

パラメータ	タイプ	内容
devName	String	端末名
loginName	String	ログインアカウント

戻り値:

```
{  
  "msg": "成功",  
  "code": 0,  
  "success": true  
}
```

注: 成功した場合、端末側検温・顔認証 APP は再起動を行う

2.4 パスワードの変更

POST: /api/admin/changePassword

リクエストパラメータ: `x-www-form-urlencoded`

パラメータ	タイプ	内容
oldPassword	String	現行のパスワード (ソルト & MD5 処理)
newPassword	String	新しいパスワード (暗号化処理せず) (6桁の数字)

戻り値:

```
{  
  "msg": "処理成功",  
  "code": 0,  
  "success": true  
}
```

3. 言語管理

3.1 言語一覧の取得

GET: /api/language/support

戻り値:

```
{  
  "msg": "処理成功",  
  "code": 0,  
  "data": {  
    "defLanguage": "zh_CN",  
    "enable": true,  
    "languageLists": [  
      {  
        "language": "zh_CN",  
        "remark": "中国語簡体字",  
        "tabField": "zhCn"  
      },  
      {  
        "language": "zh_HK",  
        "remark": "中国語繁体字",  
        "tabField": "zhHk"  
      },  
      {  
        "language": "en_US",  
        "remark": "English",  
        "tabField": "enUs"  
      },  
      {  
        "language": "ja_JP",  
        "remark": "日本語",  
        "tabField": "jaJp"  
      },  
      {  
        "language": "ko_KR",  
        "remark": "한국어",
```

```
        "tabField": "koKr"
      }
    ]
  },
  "success": true
}
```

3.2 システム言語一覧の取得

GET: /api/language/list

パラメータ		タイプ	内容
lang		String	language

戻り値:

```
{
  "msg": "処理成功",
  "code": 0,
  "data": [
    {
      "remark": "システム言語",
      "source": "系統言語"
    },
    {
      "remark": "メインキーID",
      "source": "主键 ID"
    }
  ],
  "success": true
}
```

4 登録者管理

4.1 登録者一覧の取得

POST: /api/devUser/listData

リクエストパラメータ: `application/json`

パラメータ	タイプ	内容
userName	String	登録者名もしくは ID で検索可能

Header

パラメータ	タイプ	内容
pageNum	int	◆ ページ番号 ◆ デフォルト：1
pageSize	Int	◆ 1 ページ当たりの表示件数 ◆ デフォルト：30

登録者データがある場合の戻り値:

```
{
  "msg": "処理成功",
  "code": 0,
  "data": {
    "data": [
      {
        "deptId": 101,
        "faceEnable": 1,
        "faceToken": "q6pdhiaEohhMpqD3F5jGBg==",
        "faceUrl": "https://app.yecon-
tech.com/api/download/L3Byb2ZpbGUvdXBsb2FkLzIwMjA0MTIvMTYvZmEzNTVhZTUuNzg3Mi00NjIyLTlmNjQtOTZh
ZmVhMmE5ZmM5LmpwZw%3D%3D",
        "id": 234,
        "sex": 0,
        "userCode": "gfff",
        "userName": "xzff",
        "userPhone": "1245"
      }
    ],
    "pageNum": 1,
    "pageSize": 30,
    "total": 1
  }
}
```

```
    },  
    "success": true  
  }  
}
```

注: faceUrl はチェックが必要です。http で始まった場合は顔認証タブレット端末のカメラで撮影した写真で、それ以外は base64 の画像

登録者データがない場合の戻り値:

```
{  
  "msg": "処理成功",  
  "code": 0,  
  "data": {  
    "pageNum": 1,  
    "pageSize": 30,  
    "total": 0  
  },  
  "success": true  
}
```

4.2 新規登録

POST: /api/devUser/add

リクエストパラメータ: `application/json`

パラメータ	タイプ	内容
userName	String	登録者名
userCode	String	登録者番号
userPhone	String	電話番号
faceUrl	String	画像、Base64 String
sex	Int	0=男, 1=女

戻り値:

```
{  
  "msg": "処理成功",  
  "code": 0,  
  "success": true,  
  "data": {  
    "id": 100,  
  }  
}
```

```
}  
}
```

4.3 更新

POST: /api/devUser/saveEdit

リクエストパラメータ: `application/json`

パラメータ	タイプ	内容
id	long	登録者 ID
userName	String	登録者名
userCode	String	登録者番号
userPhone	String	電話番号
faceUrl	String	画像、Base64 String
sex	Int	0=男, 1=女
faceToken	String	登録者一覧の戻り値
faceEnable	Int	登録者一覧の戻り値

注: 顔写真の変更がなければ (faceUrl = null)、faceUrl のパラメータは送信する必要がない

戻り値:

```
{  
  "msg": "処理成功",  
  "code": 0,  
  "success": true  
}
```

4.4 削除

POST: /api/devUser/delete

リクエストパラメータ: `x-www-form-urlencoded`

パラメータ	タイプ	内容
Id	Long	登録者 ID

戻り値:

```
{
```

```
"msg": "処理成功",  
"code": 0,  
"success": true  
}
```

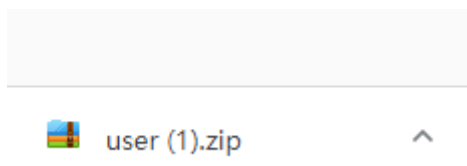
4.5 登録者データのエクスポート

POST: /api/devUser/export

リクエストパラメータ:

パラメータ	タイプ	内容
なし	なし	なし

戻り値: user.zip という圧縮フォルダーが返されます。



5 入退記録 (履歴)

5.1 入退記録一覧の取得

POST: /api/currentRecord/listData

リクエストパラメータ: **application/json**

パラメータ	タイプ	内容
userCode	String	登録者番号指定で検索
startTime	String	yyyy-MM-dd
endTime	String	yyyy-MM-dd

注: startTime と endTime を指定しない場合、デフォルトは直近7日間の記録を返す

Header

パラメータ	タイプ	内容
pageNum	int	◆ ページ番号 ◆ デフォルト：1
pageSize	Int	◆ 1ページ当たりの表示 件数 ◆ デフォルト：30

戻り値:

```
{
  "msg": "操作成功", "処理成功"
  "code": 0,
  "data": {
    "data": [
      {
        "createTime": "2020-12-17 08:59:44",
        "devSn": "YGKJ20204202773",
        "devUserId": 234,
        "faceImageBase64": "通行记录图片 Base64 此处省略", "スナップ写真 Base64 省略",
        "faceToken": "q6pdhiaEohhMpqD3F5jGBg==",
        "id": 9,
        "temp": "36.6", //検温機能オフの場合、ここは null
        "uploadStatus": 1,
        "userName": "xzff",
        "userCode": "xzff",
        "passType": 0, //認証方法 0=顔認証、 1=カード認証、 2=QR コード
        "attendanceOperationType": 0, //打刻区分(1 出勤, 2 退勤, 3 休憩開始, 4 休憩終了,
        5 残業開始, 6 残業終了)
      }
    ],
    "pageNum": 1, //ページ番号
    "pageSize": 30, //1ページ当たりの表示件数
    "total": 9 //トータル件数
  },
}
```

```
    "success": true
}
```

5.2 指定削除

POST: /api/currentRecord/delete

リクエストパラメータ: x-www-form-urlencoded

パラメータ	タイプ	内容
Id	Long	入退記録 ID

戻り値:

```
{
  "msg": "処理成功",
  "code": 0,
  "success": true
}
```

5.3 一括削除

POST: /api/currentRecord/deleteAll

パラメータ	タイプ	内容
なし	なし	なし

戻り値:

```
{
  "msg": "処理成功",
  "code": 0,
  "success": true
}
```

5.4 入退記録のエクスポート

POST: /api/currentRecord/listData

パラメータ: application/json

パラメータ	タイプ	内容
-------	-----	----

userCode	String	登録者番号
startTime	String	yyyy-MM-dd
endTime	String	yyyy-MM-dd

備考：デフォルトは直近7日間分の入退記録

戻り値：user.zip という圧縮フォルダーが返されます。

6 端末管理

6.1 設定情報の取得

GET: /api/settings/loadInfo

リクエストパラメータ:

パラメータ	タイプ	内容
なし	なし	なし

戻り値:

```
{
  "msg": "操作成功", "処理成功"
  "code": 0,
  "data": {
    "binocularLiving": true,
    "deviceVolume": 80,
    "enablePrinting": false,
    "faceAngle": 270,
    "faceMirror": false,
    "featureThreshold": 75,
    "id": 23,
    "livenessThreshold": 65,
    "living": 1,
    "maskDetection": 0,
    "multiPersonRecognition": false,
    "nearFaceWidth": 125,
    "noPictures": true,
```

```

    "printCharset": "GBK",
    "printImage": true,
    "runMode": 1,
    "screenSaverTime": 30,
    "showLogo": false,
    "showScreenSaver": false,
    "strangersPass": 1,
    "tempEnd": 37.3,
    "tempStart": 35.9,
    "tempUnit": 0,
    "temperature": 1,
    "temperatureMeasurementTime": 350,
    "temperatureMeasuringModule": 2,
    "videoAngle": 90,
    "videoMirror": false,
    "voicePromptType": 1
    ... //各パラメータの意味はパラメーター一覧表を参照ください。
},
"success": true
}

```

パラメーター一覧表:

<code>private Long id;</code>	//设备 ID
<code>private String deviceName;</code>	//设备名
<code>private double featureThreshold = 70;</code>	//特征比对阈值 (识别分)
<code>private int deviceVolume = 100;</code>	//设备音量
<code>private int temperature = 1;</code>	//测温开关 (0=关, 1=开)
<code>private int maskDetection = 0;</code>	//口罩检测 (0=关, 1=开)
<code>private int living = 1;</code>	//活体开关 (0=关, 1=开)
<code>private boolean noPictures;</code>	//不保存图片
<code>private boolean binocularLiving = true;</code>	//双目活体
<code>private int livenessThreshold = 70;</code>	//活体分
<code>private int strangersPass = 1;</code>	//陌生人通行 (0=关, 1=开)
<code>private boolean cardPass;</code>	//允许刷卡通行


```

private boolean qrCodePass; //允许刷二维码通行
private boolean multiPersonRecognition = false; //多人识别
private double tempStart = 35.9; //温度
private double tempEnd = 37.3; //温度
private double temperatureCompensation = 0; //温度补偿值
private String adminPassword = "123456"; //管理员密码
private int temperatureMeasuringModule = 1; //测温模块 (1 海康, 2 海曼, 3 高德, 4=高力合)
private long temperatureMeasurementTime = 350; //测温時間 暂定, 海康 350ms, 海曼 700ms 高力合 1050
private int runMode = 2; //运行模式(0=广告+测温+通行记录) 1=测温, 2=广告+测温, 3=宽屏,
4. 考勤模式
private boolean videoMirror = false; //视频流镜像
private boolean faceMirror = false; //人脸镜像
private int videoAngle = 90; //视频流角度
private int faceAngle = 270; //人脸角度
private int voicePromptType = 0; //语音类型(0=音效, 1 语言, 2 自定义 U 盘)
private int nearFaceWidth = 125; //小于这个宽度, 提示请靠近
private long screenSaverTime = 30; //屏保時間, 默认 30 秒
private boolean enablePrinting = false; //启用打印
private boolean printImage = true; //打印同时打印通行图片
private String printCharset = "GBK"; //打印机通讯编码
private int printType = 0; //打印机类型(0=蓝牙, 1=USB)
private int printTemplate = 0; //打印模版类型 0=默认, 1=自定义
private String printTemplateContent; //打印模版内容
private int printFontSize = 0; //字体放大多少倍
private int printTextAlign = 0; //字体居中方式 0=1 居左, 1=居中, 2=居右
private int tempUnit = 0; //0=摄氏度, 1=华氏度
private boolean showLogo = false; //主界面显示 logo
private String logoPath; //主界面 logo 路径
private boolean showScreenSaver = false; //主界面显示自定义屏保
private String screenSaverPath; //屏幕保护路径
private boolean pushRecordEnabled; //是否推送通行记录
private String pushRecordUrl; //推送地址
private String pushRecordKey; //推送密钥

```

パラメーター	タイプ	内容
Id	Long	端末 ID

featureThreshold	int	顔認証しきい値
deviceVolume	int	スピーカー音量
temperature	int	検温スイッチ (0=オフ, 1=オン)
maskDetection	int	マスク着用チェック (0=オフ, 1=オン)
living	int	生体認証スイッチ (0=オフ, 1=オン)
strangersPass	int	未登録者入場 (0=オフ, 1=オン)
tempStart	double	検温結果の正常範囲下限
tempEnd	double	検温結果の正常範囲上限
temperatureMeasuringModule	int	検温モジュール (1 : HIKVISION, 2:HEIMANN, 3: GUIDE, 4 : GAILIHE)
runMode	int	動作モード (0=広告+検温+入退記録) 1=検温, 2=広告+検温 3=横表示
videoMirror	boolean	動画のミラーリング
faceMirror	boolean	顔画像のミラーリング
videoAngle	int	動画撮影角度: 0 90 180 270
faceAngle	int	顔スナップ写真角度: 0 90 180 270
voicePromptType	int	音声案内タイプ (0=通知音, 1: TTS 音声案内)
nearFaceWidth	int	この幅より小さかったら、「近づいてください」と表示
screenSaverTime	int	◆ スクリーンセーバー待ち時間 ◆ デフォルト: 30 秒
binocularLiving	boolean	生体認証

enablePrinting	boolean	プリントアウト機能有効
livenessThreshold	Int	生体認証スコア
multiPersonRecognition	boolean	複数人同時認証
noPictures	boolean	スナップ写真保存オフ
printCharset	String	プリンター出力文字コード：6種類(GBK、GB2312、GB18030、UTF-8、UTF-16、IOS-8859-1)
printImage	boolean	検温結果合わせて、スナップ写真を同時にプリントアウト
showLogo	boolean	Logo 表示スイッチ
showScreenSaver	boolean	スクリーンセーバーの表示スイッチ
tempUnit	Int	温度単位(0=摂氏、1=華氏)
temperatureMeasurementTime	Int	検温時間
logoPath	String	Logo パス (URL)
screenSaverPath	String	スクリーンセーバーパス

注:

1. 音声案内：現状は 0（通知音）と 1（TTS 音声案内）のみをサポート、2 のカスタマイズは未サポート
2. adminPassword：戻り値は null、パスワード変更は 6.2 を参照ください。
3. LogoPath：Logo ファイルをアップロードしてない場合、logoPath のパラメーターは返されない。
4. screenSaverPath：スクリーンセーバーをアップロードしてない場合、screenSaverPath のパラメーターは返されない。

6.2 設定変更

POST: /api/settings/saveEdit:

リクエストパラメータ: `application/json`

注:

- a. nearFaceWidth 及び screenSaverTime : 変更不可
- b. adminPassword : 返さない、パスワード変更は別の API を使用して下さい
- c. loadInfo パラメーターで戻ってきたすべてのパラメーターを返す必要がある。
- d. 変更成功の場合、APP は再起動を行う。
- e. Logo ファイルをアップロードしてない場合、logoPath のパラメーターは返されない。
- f. スクリーンセーバーをアップロードしてない場合、screenSaverPath のパラメーターは返されない。

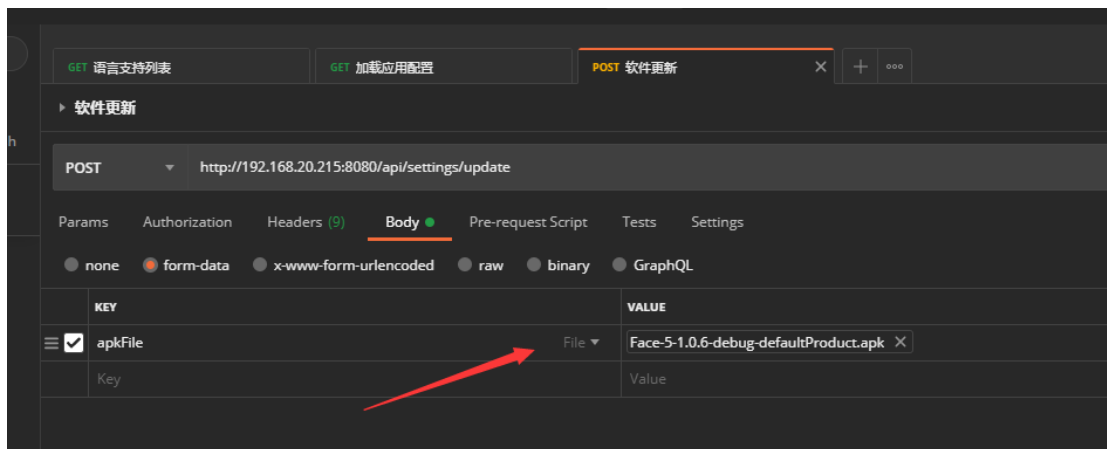
6.3 検温・顔認証 APP のソフトウェア更新

POST: /api/settings/updateApp

リクエストパラメータ: **multipart/form-data**

パラメータ	タイプ	内容
apkFile	File	Apk ファイル

注: apkFile 例は下表を参照ください。



戻り値:

```
{
  "msg": "処理成功",
  "code": 0,
  "success": true
}
```

注: 戻り値は成功の場合、10 秒後に画面更新を行う

6.4 システム時間の変更

POST: /api/settings/updateDateTime

リクエストパラメータ: x-www-form-urlencoded

パラメータ	タイプ	内容
dateTime	String	yyyy-MM-dd HH:mm:ss

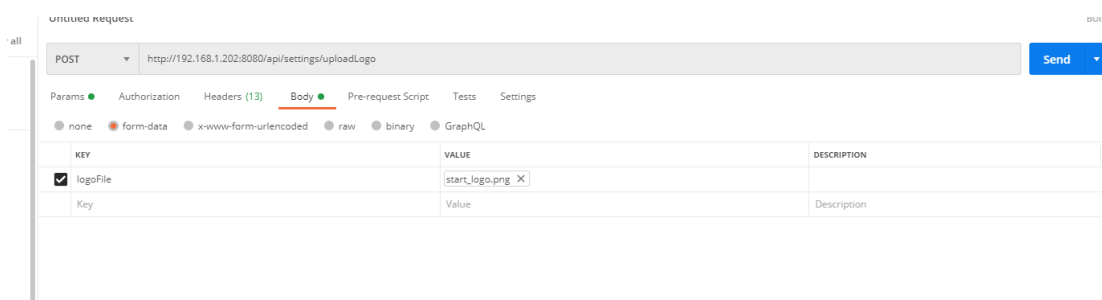
戻り値:

```
{
  "msg": "処理成功",
  "code": 0,
  "success": true
}
```

6.5 logo ファイルのアップロード

POST:/api/settings/uploadLogo

リクエストパラメータ: multipart/form-data



戻り値:

```
{
  "msg": "OK",
  "code": 0,
  "data": "/storage/emulated/0/Android/data/com.ceco.face/files/config/logo.png",
  "success": true
}
```

注: 戻り値 data にある Logo のパス (URL) は logoPath パラメーターに反映する必要があります。また、showLogo パラメーターを true と設定し、/api/settings/saveEdit に保存。

6.6 スクリーンセーバーのアップロード

POST: /api/settings/uploadScreenSaver

リクエストパラメータ: `multipart/form-data`

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> screenSaverFile	screenSaver.png	
Key	Value	Description

戻り値:

```
{
  "msg": "OK",
  "code": 0,
  "data": "/storage/emulated/0/Android/data/com.ceco.face/files/config/screenSaver.png",
  "success": true
}
```

注: 戻り値 data にあるスクリーンセーバーのパス (URL) は screenSaverPath パラメーターに反映する必要があります。また、showScreenSaver パラメーターを true と設定し、/api/settings/saveEdit に保存。

7 勤怠管理

7.1 勤怠設定情報の取得

GET: /api/attendance/loadInfo

戻り値:

```
{
  "code": "",
  "msg": "",
  "data": {
    "workShiftType": 0, //シフトタイプ, 0=1日1交代, 1=2交代, 2=3交代
    "workShifts": [
      {
        "startTime": "09:00", //始業時間

```

```

        "endTime": "18:00" //終業時間
    }
]
},
"success": "true"
}

```

7.2 打刻データの検索

POST : /api/attendance/listData

リクエストパラメータ: [application/json](#)

パラメータ	タイプ	内容
userCode	String	登録者 ID
startTime	String	開始時間
endTime	String	終了時間

Header

パラメータ	タイプ	内容
pageNum	int	◆ ページ番号 ◆ デフォルト : 1
pageSize	Int	◆ 1 ページ当たりの表示 件数 ◆ デフォルト : 30

戻り値 :

```

{
  "msg": "処理成功",
  "code": 0,
  "data": {
    "data": [
      {
        "createTime": "2021-03-08 18:38:35",
        "devSn": "YGKJ20209240958",
        "devUserId": 297,
        "exportType": 0,
        "id": 1,

```

```

        "used": false,
        "userCode": "006655",
        "userName": "ly"
    }
],
    "pageNum": 1,
    "pageSize": 30,
    "total": 1
},
    "success": true
}

```

7.3 シフト変更

POST : /api/attendance/saveEdit

リクエストパラメータ: `application/json`

パラメータ	タイプ	内容
<code>workShiftType</code>	Int	0= 1 日 1 交代, 1= 2 交代, 2= 3 交代
<code>startTime</code>	String	始業時間
<code>endTime</code>	String	終業時間

body

```

{
    "workShiftType": 0, //シフトタイプ, 0= 1 日 1 交代, 1= 2 交代, 2= 3 交代
    "workShifts": [
        {
            "startTime": "", //始業時間
            "endTime": "" //終業時間
        }
    ]
}

```


戻り値：

```
{  
  "code": "",  
  "msg": "",  
  "data": "",  
  "success": "true"  
}
```

注：

1 交代：workShifts 数は 1

2 交代：workShifts 数は 2

3 交代：workShifts 数は 3

それぞれは対象シフトの始業時間と終業時間を表す。

7.4 打刻データのエクスポート

POST: /api/attendance/export

リクエストパラメータ: `application/json`

パラメータ	タイプ	内容
userCode	String	登録者 ID
startTime	String	開始時間
endTime	String	終了時間
exportType	String	0=生データ, 1=集計表 2=集計表 (手動)

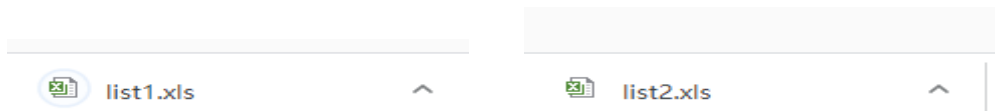
Header

パラメータ	タイプ	内容
pageNum	int	◆ ページ番号 ◆ デフォルト: 1
pageSize	Int	◆ 1 ページ当たりの表示 件数 ◆ デフォルト: 30

戻り値: 会直接戻り値一個ファイル。

ファイル名: list1.xls =生データ、ファイル名: list2.xls=集計表、ファイル名:

list3.xls=集計表 (手動)



7.5 端末情報の取得

GET:/api/aboutDevice/loadInfo

リクエストパラメータ :

パラメータ	タイプ	内容
なし	なし	なし

戻り値 :

```
{
  "msg": "処理成功",
  "code": 0,
  "data": {
    "appVersionName": "release 1.0.7_20210318",
    "devicesSn": "YGKJ20204202773",
    "ethernetMac": "8C:E8:C6:0B:8D:22",
    "faceSdkVersion": "V3.8.2",
    "romVersionName": "Android 7.1.2",
    "usbMode": "USB Device",
    "wifiMac": "C0:84:7D:31:85:62"
  },
  "success": true
}
```

8. 入退記録のプッシュ (Callback)

この章では、3rd Party のシステムに対して入退記録のプッシュ (callback) を行うための API 詳細を説明します。

8.1 プッシュ機能及び URL 設定

- ◆ プッシュ URL : http または https で始まる URL をサポートしています。

- ◆ URL の最後に “/” を入れる必要があります。

例：`http://app.com/api/test/`

- TimeOut 値：30S
- プッシュする件数：1回あたり最大10件

8.2 リクエストパラメータ

HTTP POST application/x-www-form-urlencoded (文字コード：UTF-8)

パラメータ	タイプ	内容
data	String	Json 形式、詳細は 8.4 章を参照ください。
token	String	パラメーター暗号化は MD5 を使用、詳細は 8.3 章を参照ください。
timeMillis	Long	端末の時刻
action	String	テスト時のみに使用。 action=test&data=[]を送信。

JSON パラメーターの説明(大文字、小文字を留意)

createTime	生成時間
devSn	端末 ID
userType	登録者タイプ (0=未登録者, 1=登録者, 2=来訪者)
userName	登録者名(未登録者: "来訪者")
userCode	登録者番号(未登録者: null)
devUserId	登録者 ID(未登録者: null)
faceImageBase64	スナップ写真 base64
id	入退記録 Id
passType	認証タイプ(0=顔認証, 1=カード認証, 2=QR コード)
temp	検温結果 (摂氏、検温オフの場合は null)
attendanceOperationType	打刻区分: 1 出勤, 2 退勤, 3 休憩開始, 4 休憩終了, 5 残業開始, 6 残業終了

8.3 API インターフェイス認証

タブレット端末は記録をプッシュする際に、データの暗号化を行うため、受け取る側ではそれを検証する必要があります。

token = MD5(data + timeMillis + key)

例:

data = []

timeMillis = 1622508961756

key = 123456

暗号化の前 : []1622508961756123456

暗号化の後 : 5B976630F9613B41175D1E4623FB93F9

以下にてオンライン MD5 暗号化の例を示します。

The screenshot shows a web interface for online MD5 encryption. At the top, there are several tabs for different encryption methods: DES, AES 等对称加密解密, MD5 加密 (selected), URL 加密, JS 加/解密, JS 混淆加密压缩, ESCAPE 加/解密, BASE64, 散列/哈希, and 迅雷, 快车, 旋风URL 加解密. Below the tabs, there are two text input fields. The left field contains the input string: []1622508961756123456. The right field contains the resulting MD5 hash: 5B976630F9613B41175D1E4623FB93F9. Between the fields, there is a dropdown menu set to '32位(大)', and two buttons labeled '加密' (Encrypt) and '清空' (Clear).

- ◆ Key は設定されたプッシュキーであります。
- ◆ 受け取る側ではパラメータスプリング後に md5 を行い、リクエストパラメータ token をもって検証を行います。

8.4 Json データフォーマット

```
[
  {
    "createTime":"2021-05-31 17:01:59",
    "devSn":"6F24FAZFE",
    "devUserId":100,
    "userCode":"100",
    "faceImageBase64": "***base64 图片 编码=",
    "id":1,
    "passType":0,
    "temp":"36.8",
    "userName":"访客"
    "userType":0"
  },
  {
    "createTime":"2021-05-31 18:05:59",
    "devSn":"6F24FAZFE",
    "devUserId":101,
```

8.5 再送処理

- ◆ 再送回数：3回
- ◆ 間隔：5分
- ◆ 仕様通りにデータをタブレット端末に返さないと、端末はプッシュ機能をオフにします。

```
{
  "success": true,
  "error": ""
}
```

success true=プッシュ成功

flase=失敗

error : 失敗原因

8.6 Springboot サーバ側設定 (例)

```
@Controller
@RequestMapping("/api/test")
public class ApiTestController {

    protected final Logger logger = LoggerFactory.getLogger(ApiTestController.class);

    @PostMapping
    @ResponseBody
    public JSONObject pushTest(@RequestParam String token, @RequestParam String data, @RequestParam String timeMillis){
        logger.info("Test API: json = {}, token = {}", data, token);

        String key = "123456"; //推送密钥
        String md5 = Md5Utils.hash(s: data + timeMillis + key);

        JSONObject jsonObject = new JSONObject();
        if(md5.equalsIgnoreCase(token)){
            jsonObject.put("success", true);
            //鉴权成功,处理您的业务逻辑。。。
        }else{
            jsonObject.put("success", false);
            jsonObject.put("error", "鉴权失败");
        }
        return jsonObject;
    }
}
```

クライアント側リクエスト (例)

POST http://127.0.0.1:8080/api/test

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> token	5b976630f9613b41175d1e4623fb93f9
<input checked="" type="checkbox"/> timeMillis	1622508961756
<input checked="" type="checkbox"/> action	test
<input checked="" type="checkbox"/> data	[]
Key	Value

Body Cookies (1) Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true
3 }
```