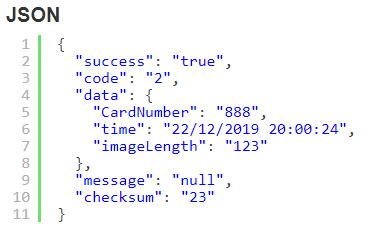
GIAO TIẾP GIỮA BỘ KHUÔN MẶT (LÂM) – QUẢN LÝ THỜI GIAN HỌC (TIỆP)

* Giao tiếp qua TCP/IP
* Bộ KM sẽ làm client, khi có học viên sẽ đẩy thông tin IDHocVien + hình ảnh lên cho server.

+ Các giao tiếp cơ bản:

1. Kết nối
   * Thực hiện kết nối trong mạng Lan, IP máy chủ là: 192.169.150.150:6868
   * Mỗi bộ sẽ phải cấu hình IP cố định, giúp cho server xác định đc client đó là ai thông qua IP đã lưu trên server.
   * Các gói tin giao tiếp gửi dưới dạng chuỗi JSON, bỏ khoảng cách. Vd Frame như sau:



Trong đó:

+ success: trạng thái thành công/thất bại

+ code: mã lệnh

+ data: dữ liệu truyền lên

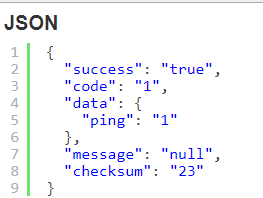
+ message: null || tin nhắn lỗi từ server nếu có.

+ checksum: checksum số kí tự của 3 trường trên. ("success":"true","data":{"CardNumber":"2","time":"22/12/201918:59:00"},"message":"null")

Validate JSON: <https://jsonformatter.curiousconcept.com/>

* + Sau khi kết nối, cứ mỗi 10s, thiết bị sẽ gửi gói tin ping-pong lên cho server.

Ping:



{

  "success": "true",

  "code": "1",

  "data": {

    "ping": "1"

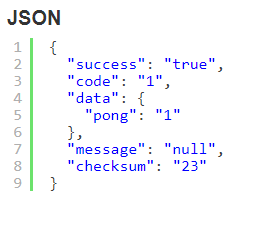
  },

  "message": "null",

  "checksum": "23"

}

Server trả về pong:



{

  "success": "true",

  "code": "1",

  "data": {

    "pong": "1"

  },

  "message": "null",

  "checksum": "23"

}

**2. Đẩy thông tin lên server**

* + - Khi có học viên thực hiện checkin/checkout, bộ KM thực hiện nhận dạng và trả kết quả về cho server gồm thông tin + hình ảnh.

Json = {

  "success": "true",

  "code": 2,

  "data": {

    "CardNumber": maDK của thí sinh,

    "time": "22/12/2019 20:00:24",

    "imageLength": "123"

"imageDir": đường dẫn file ảnh trong lưu trong FTP ( nếu recBy = “FGP” thì sẽ không có ảnh, trường này sẽ trống)

       "recBy": “Face” nếu nhận diện được từ khuôn mặt hoặc “FGP” nếu nhận diện từ FGP

  },

  "message": "null",

  "checksum": "23"

}

Hình ảnh sẽ đẩy qua FTP vào forder: localdata/gate\_shoot/20191222.

- Thông tin vân tay, khuôn mặt được thêm từ client gửi lên server

json = {

            "mac":MAC,

            "code":4,

            "fileName": fileName

        }

Client tải tệp có tên là fileName lên FTP sau đó gửi frame này để báo cho server.

Quy tắc đặt tên của fileName: TTND\_Y\_M\_D\_H\_M\_S.json

Nội dung file là chuỗi json:

{

“ID”: một số kiểu int là ID của thí sinh.

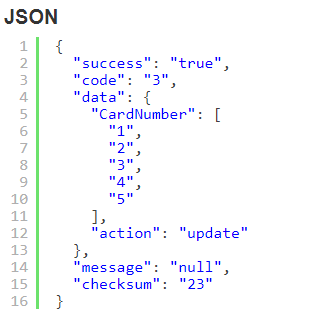
“FaceEncoding”: chuỗi string đặc trưng nhận diện khuôn mặt

“FGPEncoding”:chuỗi string đặc trưng vân tay.

}

1. **Truy vấn danh sách học viên, Cập nhật /xóa 1 danh sách nhiều học viên**

* Server đẩy 1 danh sách CardNumber xuống client, yêu cầu cập nhật hoặc xóa. Nếu cập nhật, server đẩy thêm hình ảnh xuống cho client, có tên ảnh là ID của học vien .jpg.



{

  "success": "true",

  "code": "3",

  "data": {

    "CardNumber": [

      "1",

      "2",

      "3",

      "4",

      "5"

    ],

    "action": "update"

  },

  "message": "null",

  "checksum": "23"

}

Action:

+ update: cập nhật thêm/sửa

+ getBy: lấy về danh sách theo ID

+ getAll: lấy tất cả

+ deleteBy: xóa theo danh sách

+ deleteAll : xóa tất cả

- Server yêu cầu tạo client thêm một khóa học mới.

+ frame sẽ yêu cầu client tạo một khóa học mới có thông trong trường course và cập nhật danh sách học viên trong trường cardNumber vào khóa học mới vừa tạo.

course = {

    "IDKhoaThi": 1,

    "TenKhoaThi":"KhoaThi 2/1/2020",

    "NgayTao":"13:32:32 02/01/2020",

    "DuongDanLuuAnh":""

}

json = {

    "success":"True",

    "code":3,

    "data":{

        "CourseInfo":course,

        "CardNumber":lstStudent,

        "action":"newCourse",

    },

    "message":””,

    "checksum":21

}

- Server yêu cầu client lấy tệp cập thông tin vân tay và khuôn mặt:

json = {

    "code":5,

    "data":{

        "FileName": Tên file dữ liệu khuôn mặt và vân tay trên FTP

    },

    "message":””,

    "checksum":21

}

**4. Server yêu cầu kiểm tra dữ liệu của client**

Server gửi xuống một bản tin cho client. Client trả lời bằng một tệp thông tin gửi lên ftp và một bản tin thông báo gửi vào socket

\* Nội dung bản tin server → client

 json = {

"code": 4,

                "target":target,

“detail”:detail,

                "checkSum":""

            }

- target và detail có thể là một trong các giá trị sau:

|  |
| --- |
| target = “listCourse”: server yêu cầu client gửi lên danh sách các khóa học đang lưu.  \* trả lời của client:  - File gửi lên ftp. Quy tắc đặt tên file DSKT\_Y\_M\_D\_H\_M\_S.json. Nội dung là một chuỗi json sau:  json = {                  "MAC":MAC,                  "ListCourse":list các json con mỗi json con chứa thông tin một khóa học.               }  json con = {  “ID”: ID của khóa học,  “Name”: Tên của khóa học,  “Date”: Ngày tạo của khóa học,  }  - Thông báo gửi lên socket:  json = {                  "MAC": MAC,                  "code": 4,                  "target": “listCourse”,                  "fileName":tên file thông tin đã được gửi lên ftp,                  "checkSum":""              } |

|  |
| --- |
| target = “listStudent”, : Dùng khi server yêu cầu client gửi lên tệp chứa danh sách học viên của một khóa học. Với detail = { “course” : ID của khóa học}  \* trả lời của client:  - File gửi lên FTP: Quy tắc đặt tên của file DSTS\_Y\_M\_D\_H\_M\_S.json. NỘi dung của file là chuỗi json.  json= {                      "MAC":MAC,                      "Course": ID khóa học client yêu cầu,                      "ListStudent":list các json con. mỗi json con chứa thông tin của một học viên,           }  json con = {  “ID”: id của thí sinh,.  “Name”: tên của thí sinh,  “FGP” : “T” : đã thêm vân tay Hoặc “F”:chưa thêm vân tay  “Face” = “T” : đã thêm khuôn mặt hoặc “F” : chưa thêm khuôn mặt  }  - Bản tin gửi vào socket:    json  = {                      "MAC":MAC,                      "code":4,                      "target":”listStudent”,                      "fileName":fileName,                      "checkSum":""                  } |

|  |
| --- |
| Target = “deleteCourse”. Client sẽ xóa khóa học và toàn bộ học viên trong khóa. Với detail = {“course”: id của khóa học muốn xóa}. |