**TABLE OF CONTENTS**

[1. API Kiểm tra tính hợp lệ của Khuôn mặt (TNI) 2](#_xu922ny4dyx3)

[2. API Đăng kí Khuôn mặt (TNI) 4](#_mfyv3iarstl8)

[3. API Đăng kí Khuôn mặt với nhiều hình (TNI) 6](#_87inabls6mbw)

[4. API Xác thực Khuôn mặt (TNI) 9](#_tujh0tfpxrer)

[5. API Kiểm tra khuôn mặt đã đăng kí (TNI) 10](#_72lt9zn1y6eu)

[6. API Tra cứu Khuôn mặt (ITD) 12](#_m5uo0eyzqd20)

[7. API Danh sách Face ID hợp lệ (TNI) 13](#_3x906u4hezeb)

[8. API Phát hiện Lảng vảng (ITD) 14](#_nyr6lamm2h07)

[9. API Phát hiện Xâm nhập (ITD) 14](#_xf64jh6aj10l)

[10. API Sức khỏe Hệ thống (ITD) 14](#_l55o55np4xm6)

[11. API Đọc cấu hình hệ thống (TNI) 15](#_o0kka8imyrfr)

[12. API Ghi cấu hình hệ thống (TNI) 16](#_d6vs7cx0qxr4)

[13. API Cấu hình số lượng camera (TNI) 17](#_k1r4bj6ydwpk)

[14. API Danh sách camera (TNI) 18](#_sc2dvt1segtv)

[15. API Cấu hình camera (TNI) 19](#_hzjxvakjuj8g)

[16. API Restart hệ thống (TNI) 21](#_7kkvxs1amdwr)

[17. API Danh sách Truy vết (TNI) 22](#_8gp4qc5cq8c6)

[18. API Truy vết theo ID (TNI) 24](#_h2o0wcb3799c)

[19. API Truy vết theo hình (TNI) 27](#_v7b07t1cvgos)

[20. API Phát hiện người (TNI) 29](#_pr6f3ptuv8b7)

[21. API Theo dõi người xâm nhập (ITD) 31](#_kpjt1ftlro5d)

# Face Validation API

- Purpose: Client calls API to check the validity of a person's face photo.

- Support: 1 photo for 1 person.

**Request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Time to send data to the server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | ImageData | string/null | The image contains a base64 encoded face from a .jpg . file |  |

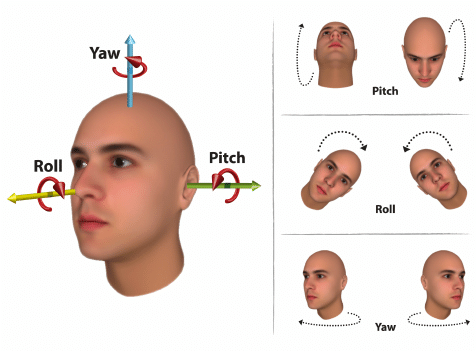
* Example for Request:
* url: <http://localhost:8004/face_validation>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {  "ImageData": "a91h3jyjksi1"  }  } |

**Response: Only after checking process**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Return code:  0: Unsuccess  1: Success |  |
| 2 | Message | string | Detailed description of the status code |  |
| 3 | IsValid | int | Determine if the snapshot is valid: 0/1, 0 is invalid, 1 is valid. |  |

* Enum definition:
* STATUS\_CODE:
  + FAIL = 0
  + SUCCESS = 1
* MESSAGE:
  + WRONG\_FORMAT = "Wrong format request fields"
  + CAPTURE\_INCORRECT = "Image is captured incorrectly"
  + CAPTURE\_CORRECT = "Image is captured correctly"
  + INTERNAL\_SERVER\_ERROR = "Internal Server Error"
* Check the existence, Data Types , Binding Value of the defined fields, if not satisfied, return:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.WRONG\_FORMAT
* IsValid = -1
* If the image is **NOT TAKEN CORRECTLY**, return:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.CAPTURE\_INCORRECT
* IsValid = 0
* If the image is **TAKEN CORRECTLY**, return:
* StatusCode = STATUS\_CODE.SUCCESS
* Message = MESSAGE.CAPTURE\_CORRECT
* IsValid = 1
* If there is an unknown error during processing, return:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.INTERNAL\_SERVER\_ERROR
* IsValid = -1
* **Criteria of face photo for registration:**
* The face must be the main subject in the frame, with a minimum horizontal and vertical dimension of 32 pixels. The software automatically checks and informs if not satisfied.
* Head rotation constraints: roll +-20 degrees, pitch +-30 degrees, yaw +-20 degrees. The software automatically checks and informs if not satisfied.



* The face have to not wear any mask, not obscured by any objects. The software have not yet could automatically check.
* Clear glasses are allowed, dark glasses are not allowed. The software does not automatically check.
* Lighting conditions are sufficient, and clear, not dim, not blurred, or not bright. The software does not automatically check

# Face Registration API

* **Purpose:** Client calls API to register new/update/delete a person's face..
* Support: 1 image for 1 person.

**Request content:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Time to send data to the server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | ImageData | string/null | The image contains a base64 encoded face from a .jpg . file |  |
| 2 | FaceID | int  -1/>=0 | ID của face trong DB (duy nhất không trùng) | T |
| 3 | OpCode | int  [0, 1, 2] | Mã thao tác:  0: Tạo mới  1: Cập nhập lại face đã đăng kí (ghi đè)  2: Xóa |  |

* Các ràng buộc kết hợp:
* Trong trường hợp OpCode=0, FaceID=-1, ImageData có giá trị.
* Trong trường hợp OpCode=1, FaceID có giá trị, ImageData có giá trị.
* Trong trường hợp OpCode=2, FaceID có giá trị, ImageData =null.
* Ví dụ request:
* url: <http://localhost:8004/face_registration>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {  "ImageData": "a91h3jyjksi1",  "FaceID": -1,  "OpCode": 0  }  } |

**Nội dung response: Chỉ phản hồi khi đã lưu xong data.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | FaceID | int | ID của face trong DB (duy nhất không trùng) |  |

* Định nghĩa Enum:
* STATUS\_CODE:
  + FAIL = 0
  + SUCCESS = 1
* MESSAGE:
  + WRONG\_FORMAT = "Wrong format request fields"
  + CAPTURE\_INCORRECT = "Image is captured incorrectly"
  + REMOVE\_SUCCESS = "Remove face successfully"
  + REMOVE\_FAIL = "Remove face failed, FaceID may not exist"
  + ADD\_NEW\_SUCCESS = "Add new face successfully"
  + UPDATE\_SUCCESS = "Update face successfully"
  + UPDATE\_FAIL = "Update face failed, FaceID may not exist"
  + INTERNAL\_SERVER\_ERROR = "Internal Server Error"
* Kiểm tra sự hiện diện (existence), kiểu dữ liệu (type), ràng buộc giá trị (value) của các trường đã định nghĩa, và [các ràng buộc kết hợp](#gpotq2ss1m9d), nếu không thỏa mãn, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.WRONG\_FORMAT
* FaceID = -1
* Nếu OpCode=2 và FaceID không tồn tại, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.REMOVE\_FAIL
* FaceID = -1
* Nếu OpCode=2 và FaceID tồn tại, xóa record có trùng FaceID với trong request, trả về:
* StatusCode = STATUS\_CODE.SUCCESS
* Message = MESSAGE.REMOVE\_SUCCESS
* FaceID = FaceID
* Nếu OpCode=0 hoặc 1, kiểm tra ảnh có được chụp đúng chuẩn không, nếu không thỏa mãn, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.CAPTURE\_INCORRECT
* FaceID = -1
* Nếu OpCode=0 và hình chụp đúng chuẩn, cấp phát FaceID, ghi 1 record mới vào DB, trả về:
* StatusCode = STATUS\_CODE.SUCCESS
* Message = MESSAGE.ADD\_NEW\_SUCCESS
* FaceID = FaceID

**Lưu ý**, FaceID mới luôn luôn đếm lên và không dùng lại các FaceID đã bị xóa, và có công thức: FaceID=max(FaceIDs) + 1.

* Nếu OpCode=1 và hình chụp đúng chuẩn và FaceID không tồn tại, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.UPDATE\_FAIL
* FaceID = -1
* Nếu OpCode=1 và hình chụp đúng chuẩn và FaceID tồn tại, cập nhập lại record, trả về:
* StatusCode = STATUS\_CODE.SUCCESS
* Message = MESSAGE.UPDATE\_SUCCESS
* FaceID = FaceID
* Nếu có lỗi không xác định xảy ra trong quá trình xử lý, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.INTERNAL\_SERVER\_ERROR
* FaceID = -1

# API Đăng kí Khuôn mặt với nhiều hình (TNI)

* **Mục đích:** Client gọi API để đăng kí mới/cập nhập/xóa face của 1 người.
* Hỗ trợ: Nhiều ảnh cho 1 người, tối đa 5 ảnh.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | ImageData | list of string | - List chứa nhiều string, mỗi string tương ứng 1 ảnh  - Mỗi ảnh chứa khuôn mặt đã được mã hóa base64 từ file .jpg | [  “a91h3jyjksi1”,  “u2n1b3ud91k”,  “u2n13j1d91k”,  “02k1b3ud91k”,  “u2n1b3u29k1”,  ] |
| 2 | FaceID | int  -1/>=0 | ID của face trong DB (duy nhất không trùng) |  |
| 3 | OpCode | int  [0, 1, 2] | Mã thao tác:  0: Tạo mới  1: Cập nhập lại face đã đăng kí (ghi đè)  2: Xóa |  |

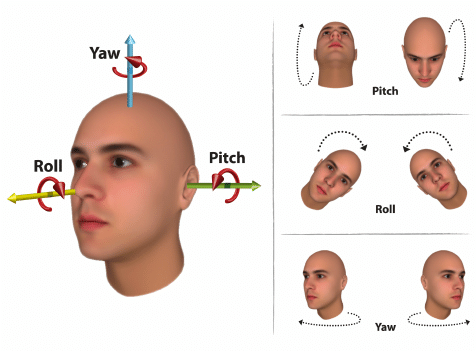
* **Các ràng buộc kết hợp:**
* Trong trường hợp OpCode=0, FaceID=-1, ImageData có giá trị.
* Trong trường hợp OpCode=1, FaceID có giá trị, ImageData có giá trị.
* Trong trường hợp OpCode=2, FaceID có giá trị, ImageData =null.
* Ví dụ request:
* url: <http://localhost:8004/face_list_registration>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {  "ImageData": [  "a91h3jyjksi1",  "u2n1b3ud91k",  "u2n13j1d91k",  "02k1b3ud91k",  "u2n1b3u29k1"  ],  "FaceID": -1,  "OpCode": 0  }  } |

**Nội dung response: Chỉ phản hồi khi đã lưu xong data.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | FaceID | int | ID của face trong DB (duy nhất không trùng) |  |

* Định nghĩa Enum và Kiểm tra các ràng buộc: Tương tự như API Đăng kí khuôn mặt
* Quy chuẩn của hình chụp khuôn mặt dùng cho đăng kí:
* Kích thước cả khung hình chụp: từ 720p (cạnh ngắn hơn có độ dài 720 pixels) - 1920p (cạnh dài hơn có độ dài 1920 pixel) → phần mềm check được
* Khuôn mặt phải là vật thể chính trong khung hình, có tỉ lệ diện tích khuôn mặt trong khung hình: 4% - 36%, diễn giải tỉ lệ: → phần mềm check được
* Kích thước ngang dọc của khuôn mặt bằng 1/5 kích thước ngang dọc của khung hình, nên tỉ lệ diện tích sẽ là 1/5\*1/5~4%.
* Kích thước ngang dọc của khuôn mặt bằng 3/5 kích thước ngang dọc của khung hình, nên tỉ lệ diện tích sẽ là 3/5\*3/5~36%.
* Các ràng buộc góc xoay của đầu: roll +-35 độ, pitch +-35 độ, yaw +-35 độ. → phần mềm check được



* Khuôn mặt không đeo khẩu trang, không che khuất bởi bất kì vật thể gì → phần mềm không check được
* Cho phép đeo kính trong suốt, không được đeo kính tối màu→ phần mềm không check được
* Điều kiện ánh sáng đủ, và rõ ràng, không tối mờ, không nhòe mờ, hoặc không sáng chói → phần mềm không check được

# API Xác thực Khuôn mặt (TNI)

* **Mục đích:** Xác thực 1 cặp hình (hình đã đăng kí trước, và hình hiện tại chụp từ camera), liệu có chung của 1 người hay không.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | ImageData | string | Ảnh khuôn mặt (chụp từ camera) đã được mã hóa base64 từ file .jpg | "A91h3jyjksi1…." |
| 2 | FaceID | int  >=0 | ID của face trong DB (duy nhất không trùng) | 0 |

* Ví dụ request:
* url: <http://localhost:8004/face_verification>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {  "ImageData": "A91h3jyjksi1",  "FaceID": 0,  }  } |

**Nội dung response: Chỉ phản hồi khi đã xử lý so sánh xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Similarity | float | Độ tương đồng giữa hình đã đăng kí trước và hình hiện tại chụp từ camera |  |

* Định nghĩa Enum:
* STATUS\_CODE:
  + FAIL = 0
  + SUCCESS = 1
* MESSAGE:
  + WRONG\_FORMAT = "Wrong format request fields"
  + CAPTURE\_INCORRECT = "Image is captured incorrectly"
  + FACE\_NOT\_EXIST = "FaceID is not exist"
  + VERIFY\_SUCCESS = "Verification successfully"
  + INTERNAL\_SERVER\_ERROR = "Internal Server Error"
* Kiểm tra sự hiện diện (existence), kiểu dữ liệu (type), và ràng buộc giá trị (value) của các trường đã định nghĩa, nếu không thỏa mãn, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.WRONG\_FORMAT
* Similarity = -1
* Nếu FaceID không tồn tại, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.FACE\_NOT\_EXIST
* Similarity = -1
* Nếu FaceID tồn tại, kiểm tra ảnh có được chụp đúng chuẩn không, nếu không thỏa mãn, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.CAPTURE\_INCORRECT
* Similarity = -1
* Nếu FaceID tồn tại và ảnh được chụp đúng chuẩn, tính độ tương đồng và trả về:
* StatusCode = STATUS\_CODE.SUCCESS
* Message = MESSAGE.VERIFY\_SUCCESS
* Similarity = Similarity
* Nếu có lỗi không xác định xảy ra trong quá trình xử lý, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.INTERNAL\_SERVER\_ERROR
* Similarity = -1

# API Kiểm tra khuôn mặt đã đăng kí (TNI)

* **Mục đích:** Kiểm tra xem liệu rằng 1 người đã đăng kí khuôn mặt hay chưa.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | ImageData | string | Ảnh khuôn mặt (chụp từ camera) đã được mã hóa base64 từ file .jpg |  |
| 2 | Threshold | float >= 0 | Ngưỡng xác định người quen, range [0,1] |  |

* Ví dụ request:
* url: <http://localhost:8004/face_existence>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {  "ImageData": [  "A91h3jyjksi1",  ],  "Threshold": 0.4,  }  } |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | IsRegistered | int | Xác định đã đăng kí hay chưa: 0/1, 0 là chưa đăng kí, 1 là đã đăng ký |  |

* Định nghĩa Enum:
* STATUS\_CODE:
  + FAIL = 0
  + SUCCESS = 1
* MESSAGE:
  + WRONG\_FORMAT = "Wrong format request fields"
  + CAPTURE\_INCORRECT = "Image is captured incorrectly"
  + VERIFY\_SUCCESS = "Verification successfully"
  + INTERNAL\_SERVER\_ERROR = "Internal Server Error"
* Kiểm tra sự hiện diện (existence), kiểu dữ liệu (type), và ràng buộc giá trị (value) của các trường đã định nghĩa, nếu không thỏa mãn, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.WRONG\_FORMAT
* IsRegistered = -1
* Kiểm tra ảnh có được chụp đúng chuẩn không, nếu không thỏa mãn, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.CAPTURE\_INCORRECT
* IsRegistered = -1
* Nếu ảnh được chụp đúng chuẩn, tính độ tương đồng cao nhất của face đang xét với các face trong DB, so sánh với Threshold, và trả về:
* StatusCode = STATUS\_CODE.SUCCESS
* Message = MESSAGE.VERIFY\_SUCCESS
* IsRegistered = IsRegistered
* Nếu có lỗi không xác định xảy ra trong quá trình xử lý, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.INTERNAL\_SERVER\_ERROR
* IsRegistered = -1

# API Tra cứu Khuôn mặt (ITD)

* **Mục đích:** Tra cứu thông tin khuôn mặt có tồn tại trong Hệ thống Đăng ký khuôn mặt hay chưa.
* Input: Hình ảnh khuôn mặt của 1 người.
* Output: Danh sách N (tối đa 3) FaceId có độ tương đồng gần giống nhất với hình ảnh input trong Hệ thống Đăng ký khuôn mặt nếu có.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | ImageData | string (base64) | Ảnh khuôn mặt (chụp từ camera) đã được mã hóa base64 từ file .jpg |  |

# API Danh sách Face ID hợp lệ (TNI)

**Mục đích:** Tra cứu danh sách Face ID còn hợp lệ trong hệ thống (ID chưa bị remove)

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Data | list | [  ***// face 1***  {  “ID”: Face\_ID\_1,  “image”: image\_url\_1  },  ***// face 2***  {  “ID”: Face\_ID\_2,  “image”: image\_url\_2  }, …  ] |  |

* Ví dụ request:
* url: [http://localhost:8004/valid\_face\_ids](http://localhost:8004/face_list_registration)
* method: GET
* params:

|  |
| --- |
| "TimeSend": "2023-01-06T09:10:49.182" |

* Định nghĩa Enum:
* STATUS\_CODE:
  + FAIL = 0
  + SUCCESS = 1
* MESSAGE:
  + WRONG\_FORMAT = "Wrong format request fields"
  + FACE\_BANK\_EMPTY = "Face bank is empty"
  + GET\_VALID\_FACES\_SUCCESS="Get valid FaceIDs successfully”
  + INTERNAL\_SERVER\_ERROR = "Internal Server Error
* Nếu không tồn tại Face ID valid, trả về:
  + - StatusCode = STATUS\_CODE.SUCCESS
    - Message = MESSAGE.FACE\_BANK\_EMPTY
    - Data = []
* Nếu tồn tại Face ID valid, trả về:
  + - StatusCode = STATUS\_CODE.SUCCESS
    - Message = MESSAGE.GET\_VALID\_FACES\_SUCCESS
    - Data = [{ID:ID\_1, image:URL\_1}, {ID:ID2, image:URL\_2…]
* Nếu có lỗi không xác định xảy ra trong quá trình xử lý, trả về:
* StatusCode = STATUS\_CODE.FAIL
* Message = MESSAGE.INTERNAL\_SERVER\_ERROR
* Data = []

# API Phát hiện Lảng vảng (ITD)

* Định nghĩa lảng vảng (sắp xếp từ trên xuống, theo thứ tự ưu tiên):
* Đứng hoặc di chuyển trong vùng nhận dạng (ROI) lâu hơn một thời gian nhất định, vd 5s.
* Hành trình di chuyển không thẳng, quanh co, lặp đi lặp lại.

**Nội dung request:**

<https://drive.google.com/file/d/1HijnRtsLQEy9ie2O-xqCBxC5W68sBicU/view?usp=share_link>

# API Phát hiện Xâm nhập (ITD)

* Định nghĩa xâm nhập: Phát hiện đối tượng một trong 2 trường hợp sau:
* Đi vào vùng nhận dạng (ROI)
* Tiến qua vạch có xác định hướng (line crossing).
* Đồng thời nhận dạng khuôn mặt (nếu thấy được khuôn mặt), và chỉ ra liệu người này có phải là nhân viên (check trong DB) hay không.

**Nội dung request:**

<https://drive.google.com/file/d/1HijnRtsLQEy9ie2O-xqCBxC5W68sBicU/view?usp=share_link>

# API Sức khỏe Hệ thống (ITD)

* Mục đích để phía server biết được các service của AI có đang hoạt động không, và các thông số của hệ thống, bao gồm:
* Mức sử dụng CPU (%)
* Mức sử dụng GPU (%)
* Mức sử dụng RAM(%)
* Mức sử dụng đĩa cứng lưu trữ (%)
* Nhiệt độ CPU (độ C)
* Nhiệt độ GPU (độ C)
* Thời gian sống của máy tính từ lúc khởi động tới hiện tại (Ngày tháng năm)

**Nội dung request:**

* <https://drive.google.com/file/d/1HijnRtsLQEy9ie2O-xqCBxC5W68sBicU/view?usp=share_link>

# 

# API Đọc cấu hình hệ thống (TNI)

* Đọc nội dung file (json) cấu hình hệ thống
* URL: <http://localhost:8000/GetSystemConfig>
* **Nội dung request:**
* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| Rỗng | | | | |

* **Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Data | dict | Nội dung của file (json) cấu hình hệ thống | |

* Nếu không có lỗi, trả về (StatusCode=1, Message=“Success”).
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data={}).
* Ví dụ request:
* url: <http://localhost:8000/GetSystemConfig>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {}  } |

# 

# API Ghi cấu hình hệ thống (TNI)

* Ghi nội dung vào file (json) cấu hình hệ thống.
* URL: <http://localhost:8000/SetSystemConfig>
* **Nội dung request:**
* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | Data | dict | Nội dung cần ghi vào file (json) cấu hình hệ thống |  |

* **Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |

* Nếu không có lỗi, trả về (StatusCode=1, Message=“Success”).
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”).
* Ví dụ request:
* url: [http://localhost:8000/](http://localhost:8000/GetSystemConfig)[SetSystemConfig](http://localhost:8000/SetSystemConfig)
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {  "Data": {  "ais\_api": {},  "ais\_deepstream": {},  ……………  }  }  } |

# API Cấu hình số lượng camera (TNI)

* Mục đích: Cấu hình số lượng tối đa camera dùng cho hệ thống.
* Khi sử dụng API này, danh sách các camera (nếu tồn tại) sẽ bị xóa để tạo lại danh sách mới. Lúc này, người dùng cần sử dụng API Cấu hình camera để thêm thông tin cho từng camera.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | MaxCam | int | Số lượng tối đa camera dùng cho hệ thống | 12 |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |

* Ràng buộc MaxCam (int) > 0, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid MaxCam”).
* Nếu không có lỗi, trả về (StatusCode=1, Message=“Success”).
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data=[]).

# 

# API Danh sách camera (TNI)

* Trả về danh sách các camera.
* **Nội dung request:**
* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| Rỗng | | | | |

* **Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Data | list | [  ***// cam1***  {  “CamID”: STT\_1\_to\_MaxCam,  “DetectStreamURL”: url\_string,  “TopicID”: topic\_string,  “ROI”: [ // list or null  x1, y1,  x2, y2,  …  xn, yn,  ],  “Line”: [ // list or null  line\_x1, line\_y1,  line\_x2, line\_y2,  dir\_x1, dir\_y1,  dir\_x2, dir\_y2,  ],  “Floor”: floor\_int,  “EnableIntrusion”: True/False  “EnableLoitering”: True/False  “EnableReID”: True/False  },  ***// cam2***  ***// cam3***  …  ] | |

* Nếu không có lỗi, trả về (StatusCode=1, Message=“Success”).
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data=[]).

# API Cấu hình camera (TNI)

* Mục đích: Cấu hình các thông số của 1 camera.
* Chú ý, sau khi cấu hình hoàn tất cho tất cả camera, cần phải gọi API Restart hệ thống để hệ thống cập nhập cấu hình mới.
* **Nội dung request:**
* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | CamID | int | Số thứ tự của camera, có giá trị liên tục trong khoảng [1; MaxCam] |  |
| 2 | DetectStreamURL | string | URL stream của camera | rtsp://10.0.8.115:554/0/  profile1/media.smp |
| 3 | TopicID | string | Mã dịnh danh của camera | cam1\_tang1 |
| 4 | ROI | list | Zone biểu diễn bằng polygon. List hoặc null | [  x1, y1,  x2, y2,  …  xn, yn,  ] |
| 5 | Line | list | Vạch biểu diễn bằng line và hướng. Hướng từ 1->2. List hoặc null | [  line\_x1, line\_y1,  line\_x2, line\_y2,  dir\_x1, dir\_y1,  dir\_x2, dir\_y2,  ] |
| 6 | Floor | int | Số của tầng sàn | 1 |
| 7 | EnableIntrusion | bool | Kích hoạt chức năng phát hiện xâm nhập | True/False |
| 8 | EnableLoitering | bool | Kích hoạt chức năng phát hiện lảng vảng | True/False |
| 9 | EnableReID | bool | Kích hoạt chức năng truy vết | True/False |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | OverlayImage | string (basse64) | Hình đã overlay các thông tin của camera như ROI, Line, Floor, EnableIntrusion, EnableLoitering, và EnableReID |  |

* Ràng buộc CamID (int) > 0 và <= MaxCam, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid CamID”).
* Nếu không đọc được frame từ camera URL, trả về (StatusCode=0, Message=“Invalid camera URL”).
* Ràng buộc TopicID không được trùng với các TopicID của những camera khác, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Existed TopicID”).
* Ràng buộc tọa độ của các điểm ROI phải nằm bên trong ảnh, không được ra ngoài, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid ROI”).
* Ràng buộc tọa độ của các điểm Line phải nằm bên trong ảnh, không được ra ngoài, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid Line”).
* Ràng buộc Floor (int) > 0, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid Floor”).
* Nếu đọc được frame từ camera URL, trả về (StatusCode=1, Message=“Success”).
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data=[]).

# API Restart hệ thống (TNI)

* Mục đích: Sau khi cấu hình xong, cần restart hệ thống để cập nhập cấu hình mới

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| Rỗng | | | | |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |

* Nếu không có lỗi, trả về (StatusCode=1, Message=“Success”), và restart lại hệ thống.
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data=[]).
* Ví dụ request:
* url: <http://localhost:8000/GetSystemConfig>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {}  } |

# 

# API Danh sách Truy vết (TNI)

* Trả về danh sách người trong khoảng thời gian nhất định.
* **Nội dung request:**
* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeStart | string | Thời gian bắt đầu | yyyy-MM-  ddTHH:mm:ss.fff |
| 2 | TimeEnd | string | Thời gian kết thúc | yyyy-MM-  ddTHH:mm:ss.fff |
| 3 | TopicIDs | list | Danh sách TopicID, mặc định là null tương ứng với lấy hết các TopicID | [“cam1\_tang2”, “cam1\_tang3”] |

* **Ví dụ request:**
* url: <http://localhost:8005/reid_list>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-03-29T10:00:00.000",  "Item":  {  "TimeStart": "2023-03-29T00:00:00.000",  "TimeEnd": "2023-03-29T10:00:00.000",  "TopicIDs": null  }  } |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Data | list | [  ***// person1***  {  “ID”: UUID,  “box\_trajectory”: [  (x1\_1, y1\_1, x2\_1, y2\_1),  (x1\_2, y1\_2, x2\_2, y2\_2),  …,  (x1\_n, y1\_n, x2\_n, y2\_n),  ],  “time\_trajectory”: [t1, t2, …, tn],  “camera”: cam\_id,  “image”: image\_url, // cropped  “video”: video\_url, // from t1 to tn  },  ***// person2***  ***// person3***  …  ] | |

* Ràng buộc TimeStart < TimeEnd, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid time, require TimeStart < TimeEnd”, Data=[]).
* Ràng buộc TimeEnd < now, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid time, require TimeEnd < now”, Data=[]).
* Nếu TopicIDs khác null, ràng buộc các phần tử trong TopicIDs phải tồn tại trong config, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid TopicIDs, <TopicID> does not exist”, Data=[]).
* Điều kiện để thêm 1 người trong DB vào danh sách trả về:
* Có thời gian xuất hiện (biến ***t1*** trong trường ***time\_trajectory***) nằm trong khung thời gian (bắt đầu-kết thúc) của request.
* Nếu TopicIDs khác null và là list, chỉ lấy những người xuất hiện trong camera có TopicID được liệt kê trong list TopicIDs đó.
* Nếu danh sách trả về rỗng (vd: Khoảng thời gian không có ai xuất hiện trước camera), trả về (StatusCode=0, Message=“No people”, Data=[]).
* Nếu danh sách trả về không rỗng, trả về (StatusCode=1, Message=“Success”, Data=[...]).
* Danh sách trả về được sắp xếp theo thứ tự thời gian theo biến ***t1*** (thời điểm mà người xuất hiện) trong trường ***time\_trajectory***.
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data=[]).

# API Truy vết theo ID (TNI)

* Chọn 1 ID trong danh sách được trả về từ API Danh sách Truy vết, trả ra thông tin của những người gần giống với người có ID đó.
* Danh sách những người trả về được sắp xếp (sort) theo độ tương đồng từ cao xuống thấp.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeStart | string | Thời gian bắt đầu | yyyy-MM-  ddTHH:mm:ss.fff |
| 2 | TimeEnd | string | Thời gian kết thúc | yyyy-MM-  ddTHH:mm:ss.fff |
| 3 | TopicIDs | list | Danh sách TopicID, mặc định là null tương ứng với lấy hết các TopicID | [“cam1\_tang2”, “cam1\_tang3”] |
| 4 | ID | string | ID của người, lấy từ danh sách trả về bởi API Danh sách Truy Vết | uwj12k2909jfn |
| 5 | MaxPeople | int | Số lượng người gần giống tối đa, kết quả trả về có thể thấp hơn con số này | 8 |
| 6 | Threshold | float | Ngưỡng độ tương đồng để giới hạn số lượng người trong danh sách trả về | 0.4 |

* **Ví dụ request:**
* url: [http://localhost:8005/reid\_list\_by\_id](http://localhost:8005/reid_list)
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-03-29T10:00:00.000",  "Item":  {  "TimeStart": "2023-03-30T00:00:00.000",  "TimeEnd": "2023-03-30T10:00:00.000",  "TopicIDs": null,  "ID": "632ca474-0edc-46bb-bb9a-6524ad19a2fe",  "MaxPeople": 8,  "Threshold": 0.5  }  } |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Data | list | [  ***// person1***  {  “ID”: UUID,  “box\_trajectory”: [  (x1\_1, y1\_1, x2\_1, y2\_1),  (x1\_2, y1\_2, x2\_2, y2\_2),  …,  (x1\_n, y1\_n, x2\_n, y2\_n),  ],  “time\_trajectory”: [t1, t2, …, tn],  “camera”: cam\_id,  “image”: image\_url,  “video”: video\_url,  “similarity”: float\_range\_0\_1,  },  ***// person2***  ***// person3***  …  ] | |

* Ràng buộc TimeStart < TimeEnd, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid time, require TimeStart < TimeEnd”, Data=[]).
* Ràng buộc TimeEnd < now, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid time, require TimeEnd < now”, Data=[]).
* Nếu TopicIDs khác null, ràng buộc các phần tử trong TopicIDs phải tồn tại trong config, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid TopicIDs, <TopicID> does not exist”, Data=[]).
* Ràng buộc MaxPeople (int) > 0, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid MaxPeople”, Data=[]).
* Ràng buộc Threshold (float), range [0; 1], nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid Threshold”, Data=[]).
* Điều kiện để thêm 1 người trong DB vào danh sách trả về:
* Có thời gian xuất hiện (biến ***t1*** trong trường ***time\_trajectory***) nằm trong khung thời gian (bắt đầu-kết thúc) của request.
* Nếu TopicIDs khác null và là list, chỉ lấy những người xuất hiện trong camera có TopicID được liệt kê trong list TopicIDs đó.
* Có độ tương đồng với người được request: similarity > Threshold.
* Danh sách trả về được sắp xếp theo độ tương đồng (trường ***similarity***) từ cao xuống thấp.
* Danh sách bị giới hạn số lượng không vượt quá MaxPeople.
* Nếu danh sách trả về rỗng, trả về (StatusCode=0, Message=“No people”, Data=[]).
* Nếu danh sách trả về không rỗng, trả về (StatusCode=1, Message=“Success”, Data=[...]).
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data=[]).

# API Truy vết theo hình (TNI)

* Đưa 1 tấm hình (đã crop) của 1 người, trả ra thông tin của những người gần giống với người đó.
* Danh sách những người trả về được sắp xếp (sort) theo độ tương đồng từ cao xuống thấp.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeStart | string | Thời gian bắt đầu | yyyy-MM-  ddTHH:mm:ss.fff |
| 2 | TimeEnd | string | Thời gian kết thúc | yyyy-MM-  ddTHH:mm:ss.fff |
| 3 | TopicIDs | list | Danh sách TopicID, mặc định là null tương ứng với lấy hết các TopicID | [“cam1\_tang2”, “cam1\_tang3”] |
| 4 | ImageData | string (base64) | Ảnh khuôn mặt (chụp từ camera) đã được mã hóa base64 từ file .jpg |  |
| 5 | MaxPeople | int | Số lượng người gần giống tối đa, kết quả trả về có thể thấp hơn con số này | 8 |
| 6 | Threshold | float | Ngưỡng độ tương đồng để giới hạn số lượng người trong danh sách trả về (range [0, 1] | 0.4 |

**Ví dụ request:**

* url: <http://localhost:8005/reid_list_by_image>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-03-29T10:00:00.000",  "Item":  {  "TimeStart": "2023-03-30T00:00:00.000",  "TimeEnd": "2023-03-30T10:00:00.000",  "TopicIDs": null,  "ImageData": "/9j/4QAYRXhpZgAASUkqA",  "MaxPeople": 8,  "Threshold": 0.5  }  } |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Data | list | [  ***// person1***  {  “ID”: UUID,  “box\_trajectory”: [  (x1\_1, y1\_1, x2\_1, y2\_1),  (x1\_2, y1\_2, x2\_2, y2\_2),  …,  (x1\_n, y1\_n, x2\_n, y2\_n),  ],  “time\_trajectory”: [t1, t2, …, tn],  “camera”: cam\_id,  “image”: image\_url,  “video”: video\_url,  “similarity”: float\_range\_0\_1,  },  ***// person2***  ***// person3***  …  ] | |

* Ràng buộc TimeStart < TimeEnd, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid time, require TimeStart < TimeEnd”, Data=[]).
* Ràng buộc TimeEnd < now, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid time, require TimeEnd < now”, Data=[]).
* Nếu TopicIDs khác null, ràng buộc các phần tử trong TopicIDs phải tồn tại trong config, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid TopicIDs, <TopicID> does not exist”, Data=[]).
* Ràng buộc MaxPeople (int) > 0, nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid MaxPeople”, Data=[]).
* Ràng buộc Threshold (float), range [0; 1], nếu không thỏa mãn, trả về (StatusCode=0, Message=“Invalid Threshold”, Data=[]).
* Điều kiện để thêm 1 người trong DB vào danh sách trả về:
* Có thời gian xuất hiện (biến ***t1*** trong trường ***time\_trajectory***) nằm trong khung thời gian (bắt đầu-kết thúc) của request.
* Nếu TopicIDs khác null và là list, chỉ lấy những người xuất hiện trong camera có TopicID được liệt kê trong list TopicIDs đó.
* Có độ tương đồng với người được request: similarity > Threshold.
* Danh sách trả về được sắp xếp theo độ tương đồng (trường ***similarity***) từ cao xuống thấp.
* Danh sách bị giới hạn số lượng không vượt quá MaxPeople.
* Nếu danh sách trả về rỗng, trả về (StatusCode=0, Message=“No people”, Data=[]).
* Nếu danh sách trả về không rỗng, trả về (StatusCode=1, Message=“Success”, Data=[...]).
* Nếu request không có đủ các trường yêu cầu, thì trả về (StatusCode=2, Message=“Wrong request fields”, Data=[]).

# API Phát hiện người (TNI)

* Đưa 1 tấm hình toàn cảnh, trả ra thông tin những người phát hiện được.
* Danh sách những người trả về được sắp xếp (sort) theo độ tin cậy từ cao xuống thấp.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | ImageData | string (base64) | Ảnh đã được mã hóa base64 từ file .jpg |  |

**Ví dụ request:**

* url: <http://localhost:8006/PersonDetection>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-01-06T09:10:49.182",  "Item": {  "ImageData": "a91h3jyjksi1"  }  } |

**Nội dung response: Chỉ phản hồi khi đã xử lý xong.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | StatusCode | int | Mã trả về:  0: Không thành công  1: Thành công  2: Sai format |  |
| 2 | Message | string | Mô tả chi tiết status code |  |
| 3 | Data | list | [  ***// person1***  {  “box”: [int\_x1, int\_y1, int\_x2, int\_y2],  “score”: float\_range\_0\_1,  },  ***// person2***  ***// person3***  …  ] | |

**Kiểm tra Enum và các ràng buộc kết hợp:**

* Nếu danh sách trả về không rỗng, trả về (StatusCode=1, Message=“Success”, Data=[...]).
  + Định nghĩa:
    - box: Vẽ khoanh vùng đối tượng theo dạng hình hộp
    - score: Mức độ giống con người

|  |
| --- |
| {  "Data": [  {  "box": [  60,  11,  218,  153  ],  "score": 0.5144849419593811  }  ],  "Message": "Detect successfully",  "StatusCode": 1  } |

* Nếu request không có đủ các trường yêu cầu, thì trả về
  + Data=[]
  + StatusCode=0
  + Message= MESSAGE.WRONG\_FORMAT

* Nếu ImageData = null:
  + Data = []
  + StatusCode = 0
  + Message = MESSAGE.WRONG\_FORMAT
* Nếu TimeSend = null:
  + Data = []
  + StatusCode = 0
  + Message = MESSAGE.WRONG\_FORMAT

# API Theo dõi người xâm nhập (ITD)

* Khi hệ thống AI phát hiện 1 người xâm nhập (line crossing hoặc zone crossing), hệ thống AI gửi ngay API phát hiện xâm nhập về máy chủ thông qua [API Phát hiện Xâm nhập (ITD)](#_xf64jh6aj10l).
* Đồng thời, hệ thống AI kích hoạt theo dõi người đó (thông qua truy vết), và liên tục cập nhập hành trình di chuyển của người đó về máy chủ.

**Nội dung request:**

* Timesend:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | TimeSend | string | Thời gian gửi dữ liệu lên server | yyyy-MM-  ddTHH:mm:ss.fff |

* Item:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Key** | **Datatype** | **Descriptions** | **Example** |
| 1 | EventID | string (uuid) | ID của event xâm nhập, đã được gửi ngay lúc phát hiện, thông qua [API Phát hiện Xâm nhập (ITD)](#_xf64jh6aj10l) | 3a42cf74-9dba-46a3-8841-a011526e5725 |
| 2 | IsFinish | bool | Cho biết quá trình theo dõi đã kết thúc hay chưa:   * True: đã kết thúc * False: chưa kết thúc | False/True |
| 3 | RetrievedPeople | list | [  ***// person1***  {  “ID”: UUID,  “box\_trajectory”: [  (x1\_1, y1\_1, x2\_1, y2\_1),  (x1\_2, y1\_2, x2\_2, y2\_2),  …,  (x1\_n, y1\_n, x2\_n, y2\_n),  ],  “time\_trajectory”: [t1, t2, …, tn],  “camera”: cam\_id,  “image”: image\_url,  “video”: video\_url,  “similarity”: float\_range\_0\_1,  },  ***// person2***  ***// person3***  …  ] | |

**Ví dụ request:**

* url: <http://localhost:6543/IntrusionTracing>
* method: POST
* json:

|  |
| --- |
| {  "TimeSend": "2023-05-27T14: 59: 28.082083",  "Data": [  {  "EventID": "3a42cf74-9dba-46a3-8841-a011526e5725",  "IsFinish": false,  "RetrievedPeople": [  {  "ID": "a5794814-02fb-4d33-8646-5413e221668d",  "image": "http://10.0.8.121:8012/open/images/object/2023052712/CCTV24/20230527120255507\_CCTV24\_27.jpg",  "video": "rtspt://admin:Cctv2018@10.0.8.40:554/profile4/media.smp",  "camera": "CCTV24",  "similarity": 0.8,  "box\_trajectory": [  1230,  404,  1487,  1150  ],  "time\_trajectory": "2023-05-27T12:02:55.507247"  },  {  "ID": "3a42cf74-9dba-46a3-8841-a011526e5725",  "image": "http://10.0.8.121:8012/open/images/object/2023052712/CCTV24/20230527120255507\_CCTV24\_29.jpg",  "video": "rtspt://admin:Cctv2018@10.0.8.40:554/profile4/media.smp",  "camera": "CCTV24",  "similarity": 0.5,  "box\_trajectory": [  1350,  500,  1200,  1000  ],  "time\_trajectory": "2023-05-27T12:02:59.507247"  }  ]  }  ]  } |

**Nội dung response: Tương tự như** [**API Phát hiện Xâm nhập (ITD)**](#_xf64jh6aj10l)