



Government Data Center and Cloud Service

**Face Verification
API Manual**

National Telecom Public Company Limited

support@gdcc.onde.go.th

Table Content

Table Content	2
1. Face Verifications API Manual	3
1.1 Requirement	3
1.1.1 Post Request	3
1.1.2 Image URL Requirements	3
1.2 API Calls Example	3
1.3 Response	4
1.4 Example Web Request	8
2. Errors	11
2.1 Standard API Errors	11
2.2 Application Specific Errors	11
3. Use case	12
3.1 E-KYC	12
3.2 Control Access To Sensitive Areas	13
3.3 Check-in	14

1. Face Verifications API Manual

ผู้ใช้งานสามารถเริ่มใช้งาน Face Verifications API ได้ผ่าน

Endpoint: <https://faceai.gdcc.onde.go.th/api/v1/inference/verifications>

Face Verification สามารถแยกใบหน้าได้ตาม Biometrics Guideline ของธนาคารแห่งประเทศไทย ที่ False Acceptance Rate (FAR) ที่ 0.36% และ False Rejection Rate (FRR) ที่ 0.05% สามารถนำไปใช้ในการจำแนกใบหน้าในระบบ e-KYC สำหรับ API ใช้สำหรับการเปรียบเทียบรูปสองรูปว่ามีใบหน้าคล้ายคลึง โดยในรูปจะมีได้แค่หนึ่งใบหน้าเท่านั้น ในการนำมาประมวลผล

1.1 Requirement

1.1.1 Post Request

การเรียกใช้งาน Face Verifications API นั้นสามารถทำได้ด้วยการส่ง Post Request ไปที่

URL: <https://faceai.gdcc.onde.go.th/api/v1/inference/verifications> พร้อมกับข้อมูล ซึ่งสามารถดูตัวอย่างการใช้งานได้ใน หัวข้อ 1.2 API Calls Example

ใน Manual เราจะสาธิตวิธีการเรียก API ผ่าน cURL ซึ่งเป็นเครื่องมือในการ รับส่งข้อมูลจาก Server ผ่าน Protocol ต่างๆ โดยปกติแล้วในระบบปฏิบัติการ Linux และ Mac Os จะมี cURL ติดตั้งอยู่แล้วถ้าเกิด ไม่มีสามารถติดตั้งได้ผ่าน Command Line ข้างล่างนี้

สามารถติดตั้ง cURL ใน Mac OS ได้ผ่าน homebrew ด้วยคำสั่ง

```
brew install curl
```

หรือสามารถติดตั้ง cURL ในระบบ Linux ด้วย

```
sudo apt-get install curl
```

หรือสามารถดาวน์โหลด cURL เวอร์ชัน Window ได้ที่

URL: <https://curl.se/windows/>

1.1.2 Image URL Requirements

ในการเรียกใช้งาน Face Verifications นั้นจะเป็นการเปรียบเทียบระหว่างสองรูปโดยที่รูปทั้งสองรูปนั้น ต้องทำการเก็บไว้บน Server ที่เป็น Public Access ให้ตัว API สามารถเรียกดูได้ เช่นบน Amazon s3 เป็นต้น

1.2 API Calls Example

สามารถดูตัวอย่างการเรียกใช้งาน API ผ่าน Curl ได้ดังรูป 1-1 และคำอธิบายค่าที่ต้องใส่ได้ที่ตาราง 1-1

Technical Support | Tel: 02-024-1999 ext. 1| Email: support@gdcc.onde.go.th

```
curl --location --request POST 'https://faceai.gdcc.onde.go.th/api/v1/inference/verifications' \
--header 'Api-Key: XXXXXXXXXXXXXXXXXXXX' \
--header 'Content-Type: application/json' \
--data-raw '{
  "image_url1": "https://XXXXXXXXXXXXXXXXXXXXX.jpg",
  "image_url2": "https://YYYYYYYYYYYYYYYYYYY.jpg"
}'
```

รูป 1-1 ตัวอย่างการเรียกใช้งาน API ผ่าน Curl

Parameter	Description
Api-Key	API Key ที่ได้รับจากระบบ API Management Platform
image_url1	URL ของรูปที่ต้องการนำมาเปรียบเทียบ
image_url2	URL ของรูปที่ต้องการนำมาเปรียบเทียบ

ตาราง 1-1 คำอธิบาย Parameter ที่ต้องใส่เมื่อใช้งาน Face Verification API

1.3 Response

Response ที่ได้จาก API หลักๆประกอบไปด้วย status, image_url1, image_url2, error_message และ results ดังรูป 1-2 และสามารถอ่านคำอธิบายของแต่ละฟิลด์ได้ที่ตาราง 1-2 และ 1-3

```
{
  "status": "done",
  "image_url1": "https://XXXXXXXXXXXXXXXXXXXXX.jpg",
  "image_url2": "https://YYYYYYYYYYYYYYYYYYY.jpg",
  "error_message": null,
  "request_useragent": "XXXXXXXXXXXXXXXXXXXXX",
  "requested_at": "2021-XX-XXTXX:XX:XX.XXXXXX",
  "finished_at": "2021-XX-XXTXX:XX:XX.XXXXXX",
  "results": {
    "similarity": 0.3522166609764099,
    "match_level_1": true,
    "match_level_2": false,
    "match_level_3": false,
    "match_level_4": false
  }
}
```

รูป 1-2 ตัวอย่างข้อมูล Response ที่ได้กลับมาจาก API

Field	Description
status	Done - API ทำงานสำเร็จ

	Error - API ทำงานไม่สำเร็จ
image_url1	URL ของรูปที่ต้องการนำมาเปรียบเทียบ
image_url2	URL ของรูปที่ต้องการนำมาเปรียบเทียบ
error_message	Error ที่เกิดขึ้นเมื่อ API ทำงานไม่สำเร็จ
requested_at	เวลาที่เรียกใช้งาน API
finished_at	เวลาที่ API ประมวลผลสำเร็จ
results	ผลลัพธ์ที่ได้จากรูป

ตาราง 1-2 ค่าอธิบายของแต่ละฟิลด์ของ Response ที่ได้จาก API

Results	Description
similarity	ค่าความเหมือนระหว่าง 2 รูป มีค่าระหว่าง 0-1
match_level_1	true เมื่อ similarity > 0.2891
match_level_2	true เมื่อ similarity > 0.3549
match_level_3	true เมื่อ similarity > 0.3960
match_level_4	true เมื่อ similarity > 0.4475

ตาราง 1-3 ค่าอธิบายของฟิลด์ Results

การตอบสนองจะแสดงคะแนนความคล้ายคลึงและความมั่นใจในระดับการจับคู่ ความสัมพันธ์ของคะแนนความคล้ายคลึงกับ FAR (อัตราการยอมรับที่ผิดพลาด) ดังตาราง 1-4 และความสัมพันธ์ระหว่าง FAR และ Match Level ได้ที่ตาราง 1-5

FAR	Similarity
0.1%	0.2891
0.01%	0.3549
0.001%	0.3960
0.0001%	0.4475

ตาราง 1-4 ความสัมพันธ์ระหว่าง FAR และ Similarity

FAR	Match Level	Description
-----	-------------	-------------

0.1%	match_level_1: true	รูปถูกต้องอนุมัติ 999 ใน 1000 ใบหน้า
0.01%	match_level_2: true	รูปถูกต้องอนุมัติ 9,999 ใน 10,000 ใบหน้า
0.001%	match_level_3: true	รูปถูกต้องอนุมัติ 99,999 ใน 100,000 ใบหน้า
0.0001%	match_level_4: true	รูปถูกต้องอนุมัติ 999,999 ใน 1000,000 ใบหน้า

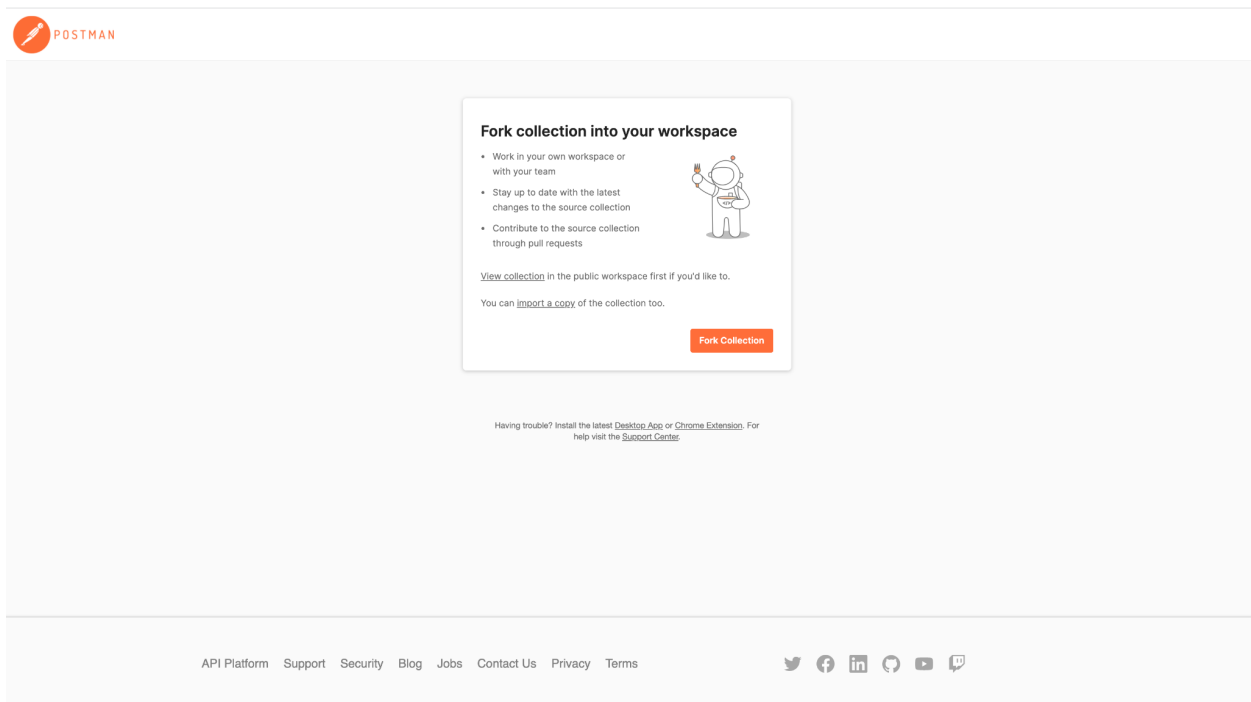
ตาราง 1-5 ความสัมพันธ์ระหว่าง FAR และ Match Level

1.4 Example Web Request

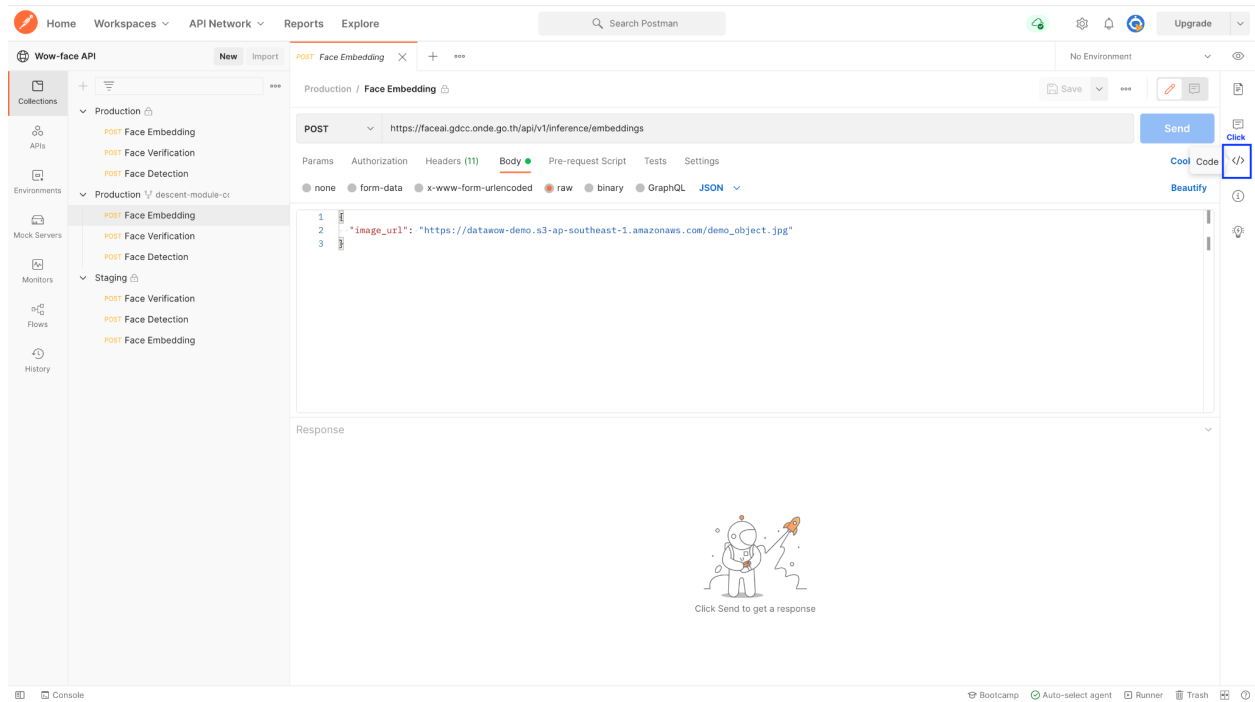
เครื่องมือสำหรับทดสอบ API แบบ Web Based

<https://god.gw.postman.com/run-collection/18153545-654baae7-af92-473e-9518-afa62e8b76b9?action=collection%2Ffork&collection-url=entityId%3D18153545-654baae7-af92-473e-9518-afa62e8b76b9%26entityType%3Dcollection%26workspaceId%3Dcd9d2008-4a1f-4633-9942-de2ff585e92e>

หมายเหตุ: ต้องสมัคร Postman Account ก่อนใช้งาน



1. ให้ทำการกด Fork หรือ View Collection
2. ให้ทำการ Click ที่ปุ่ม Code ด้านขวา



3. จะมี Code Snippet ขึ้นมา ให้เลือกตัวอย่างของการเรียกใช้งานในแต่ละภาษาได้ ตัวอย่างจะเป็น PHP - cURL

```

1  <?php
2
3  $curl = curl_init();
4
5  curl_setopt_array($curl, array(
6      CURLOPT_URL => 'https://faceai.gdcc.
       onde.go.th/api/v1/inference/
       embeddings',
7      CURLOPT_RETURNTRANSFER => true,
8      CURLOPT_ENCODING => '',
9      CURLOPT_MAXRECURSIONS => 10,
10     CURLOPT_TIMEOUT => 0,
11     CURLOPT_FOLLOWLOCATION => true,
12     CURLOPT_HTTP_VERSION =>
        CURL_HTTP_VERSION_1_1,
13     CURLOPT_CUSTOMREQUEST => 'POST',
14     CURLOPT_POSTFIELDS => '{
15         "image_url": "https://datawow-demo.
        s3-ap-southeast-1.amazonaws.com/
        demo_object.jpg"
16     }',
17     CURLOPT_HTTPHEADER => array(
18         'Content-Type: application/json',
19         'Api-Key:
        qYEetNK48pQpAGwRgnA7EHQ77uLMfmJa'
20     ),
21 ));
22
23 $response = curl_exec($curl);
24
25 curl_close($curl);
26 echo $response;
27

```

4. ทำการ copy code ตัวอย่างไปใช้งาน แต่ต้องทำการแก้ไข key และ params ตามที่ Client ต้องการใช้งาน

2. Errors

2.1 Standard API Errors

API protocol จะส่งคืนข้อผิดพลาดที่รู้จักกันทั่วไปในการใช้งาน API สามารถดูรายละเอียดได้ที่ตาราง 2-1

Error	Code	Description
Unauthorized	401	รหัสข้อผิดพลาดนี้บ่งบอกว่าคำขอไม่ได้รับอนุญาต ลูกค้าควรตรวจสอบคีย์ API ที่ให้มา
Method Not Allowed	405	รหัสข้อผิดพลาดนี้บ่งชี้ว่าลูกค้าใช้วิธีการเรียก API ที่ไม่ถูกต้อง
Payload Too Large	413	รหัสข้อผิดพลาดนี้บ่งชี้ว่าเนื้อหาของคำขอมีขนาดใหญ่เกินไป
Internal Server Error	500	รหัสข้อผิดพลาดนี้บ่งชี้ว่ามีข้อผิดพลาดที่ไม่สามารถจัดการได้จากเซิร์ฟเวอร์
Service Unavailable	503	รหัสข้อผิดพลาดนี้บ่งชี้ว่าเซิร์ฟเวอร์ไม่พร้อมใช้งาน
Gateway Timeout	504	รหัสข้อผิดพลาดนี้บ่งชี้ว่าเซิร์ฟเวอร์แบ็กเอนด์ใช้เวลาในการตอบสนองนานเกินไป

ตาราง 2-1 คำอธิบายของ Standard API Errors

2.2 Application Specific Errors

นอกจาก Standard API Errors จะมีข้อผิดพลาดที่เกิดจาก Face Verification API มีด้วยกันสองกรณีตามตาราง 2-2

Error	Code	Description	Solution
Face Detection Error	400	API ไม่สามารถตรวจจับใบหน้าได้	อัปโหลดรูปที่มีใบหน้าอีกครั้ง
Image Not Found	404	URL ของรูปภาพไม่ถูกต้อง	อัปโหลด URL ที่ถูกต้อง

ตาราง 2-2 คำอธิบาย Error ที่เกิดจาก Face Verification API

3. Use case

ในการนำ Face Verification ไปใช้งาน จะมีตัวอย่างการใช้งานทั่วไป ตามตัวอย่าง

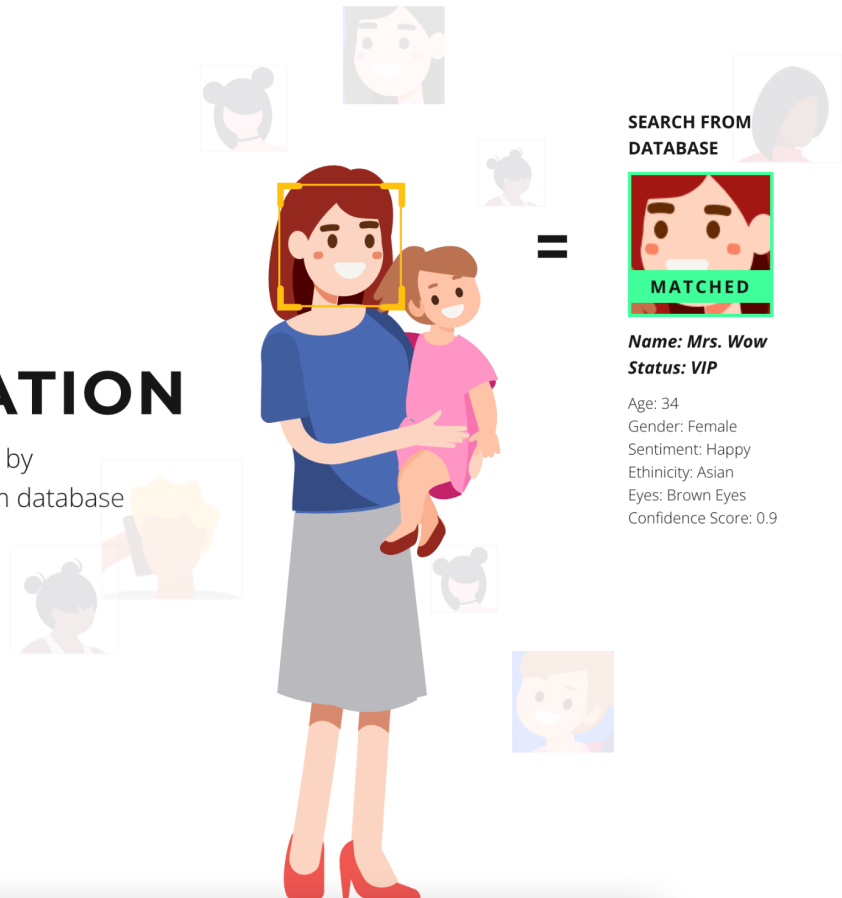
3.1 E-KYC

ใช้สำหรับระบบลงทะเบียนเข้าระบบ แล้วต้องการการยืนยันตัวตน เช่น รูปบัตรประชาชน และ รูปภาพจริง เพื่อให้ Face Verification ช่วยในการตรวจสอบความถูกต้องของบุคคลที่ลงทะเบียน

FACE IDENTIFICATION

Able to identify detected person by comparing number of faces from database

[GET IN TOUCH →](#)



3.2 Control Access To Sensitive Areas

ใช้สำหรับตรวจสอบการเข้าสู่พื้นที่ที่ต้องการกำหนดไว้ เช่น จะมีพนักงานบางคนเท่านั้นที่เข้าบางพื้นที่ได้ ระบบจะเก็บใบหน้าพนักงานคนนั้นไว้ในระบบ และให้ตรวจสอบในหน่วยงานว่าพนักงานในไหนที่สามารถที่จะเข้าพื้นที่นั้นๆได้



3.3 Check-in

ใช้สำหรับตรวจสอบการเข้าทำงานโดยการตรวจจากหน้าผู้ใช้งานในระบบที่มาทำงานที่หน้างาน

