

Kakao API – 포즈

1. 이미지 분석

- Rest API로만 사용 가능

Request

URL

```
POST /pose HTTP/1.1
Host: cv-api.kakaobrain.com
Authorization: KakaoAK {REST_API_KEY}
```

Header

Name	Description	Required
Authorization	앱 인증 키 [내 애플리케이션] > [앱 설정] > [앱 키]에서 확인 가능	O

Parameter

Name	Type	Description	Required
image_url	String	이미지 URL 리다이렉트 이미지원	O*
file	Binary	처리할 이미지 파일	O*

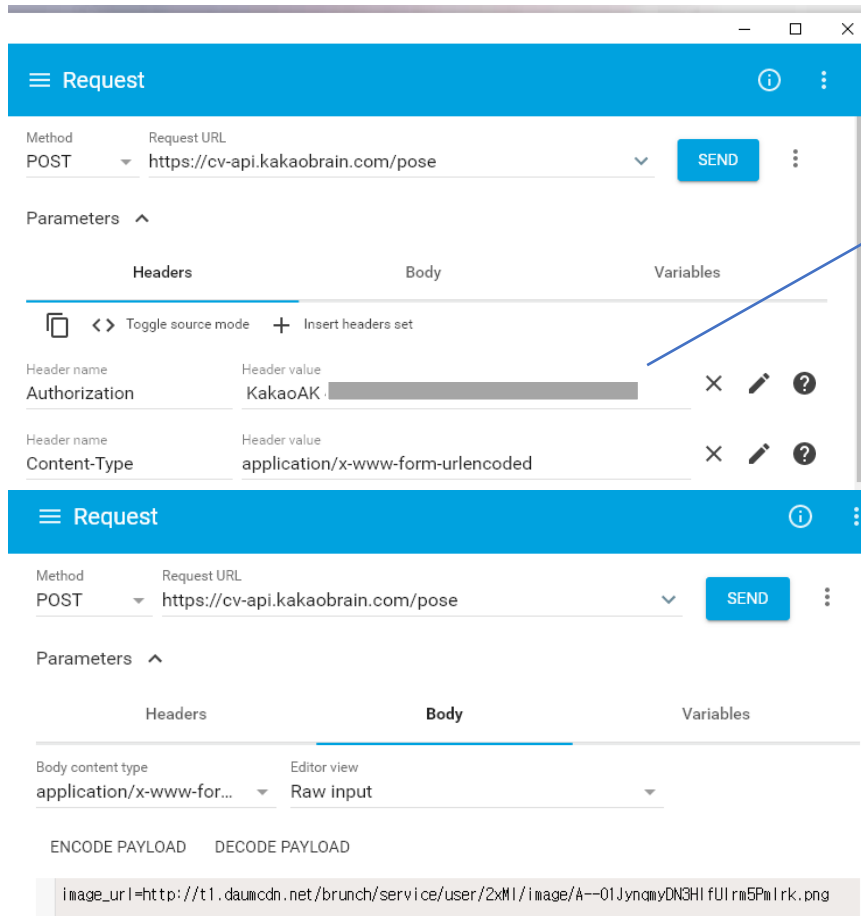
> Response

Person

Name	Type	Description
area	Float	키 포인트를 모두 포함하는 바운딩 박스(bounding box)의 넓이
bbox	Float []	키 포인트 중 가장 위쪽에 있는 키 포인트의 좌표(x,y)와 바운딩 박스의 너비(w)와 높이(h) [x, y, w, h]
category_id	Int	1로 고정 1: Person
keypoints	Float []	이미지에서 검출된 17개의 키 포인트의 좌표(x, y)와 정확도(score) [x_1, y_1, score_1, x_2, y_2, score_2, ..., x_17, y_17, score_17] x, y, score 값 뒤에 표기한 1부터 17까지의 키 포인트 ID는 사람의 각 신체 부위를 나타냄 (아래 keypoints 항목 참고) 예: [코의 x 좌표, 코의 y 좌표, 코의 키 포인트에 대한 신뢰도, 왼쪽 눈의 x 좌표, 왼쪽 눈의 y 좌표, 왼쪽 눈의 키 포인트에 대한 신뢰도, ..., 오른쪽 발목의 x 좌표, 오른쪽 발목의 y 좌표, 오른쪽 발목의 키 포인트에 대한 신뢰도]
score	Float	키 포인트 데이터에 대한 신뢰도, 0부터 1 사이 값

1. 이미지 분석

- 실행 코드



```
{  
  "code": -401,  
  "msg": "Not authorized with provided app key"  
}
```

만약 이러한 에러가 뜬다면
Admin 키를 사용하세요!



image_url=https://t1.daumcdn.net/cfile/tistory/991E6C3B5B03B68519
로 주소 변경되었습니다.

1. 이미지 분석

• 실행 결과

(x좌표, y좌표, 신뢰도)

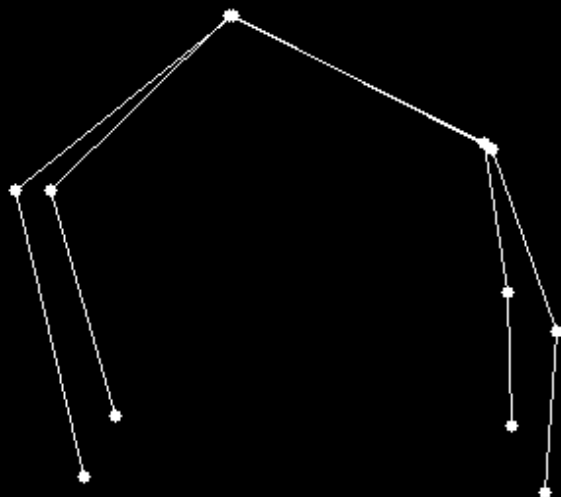
```
[{"area":97263.1423,"bbox": [87.5562, 56.48, 332.1375, 292.84], "category_id":1,
"keypoints": [372.2656, 194.35, 0.0062,
356.6406, 192.05, 0.0025,
356.6406, 189.75, 0.0007,
356.6406, 189.75, 0.1364,
354.2969, 195.1167, 0.0052,
358.2031, 148.35, 0.3711,
354.2969, 145.2833, 0.2546,
391.0156, 241.8833, 0.7333,
366.7969, 221.95, 0.6397,
385.5469, 323.9167, 0.7512,
368.3594, 289.4167, 0.7571,
226.1719, 80.8833, 0.4364,
224.6094, 80.8833, 0.4292,
115.2344, 169.8167, 0.5838,
133.2031, 169.8167, 0.5349,
150.3906, 315.4833, 0.5465,
166.7969, 284.8167, 0.5167], "score":0.3936}]
```

keypoints

키 포인트 ID	신체 부위
1	nose
2	left_eye
3	right_eye
4	left_ear
5	right_ear
6	left_shoulder
7	right_shoulder
8	left_elbow
9	right_elbow
10	left_wrist
11	right_wrist
12	left_hip
13	right_hip
14	left_knee
15	right_knee
16	left_ankle
17	right_ankle

1. 이미지 분석

- 좌표의 이미
- 좌표는
- 얼굴 : 코, 왼 /오른쪽 눈, 왼 /오른쪽 귀
- 몸 : 왼 / 오른쪽의 손목, 어깨, 팔꿈치, 엉덩이, 무릎, 발목을 의미한다.
- 다음 슬라이드는 얼굴 부분을 제외한 몸 부분의 좌표들을 openCV를 활용하여 그린 것이다.



File Edit Format Run Options Window Help

```
import numpy as np
import cv2
```

```
# color 설정
color = (255, 255, 255)
```

```
# 모두 0으로 되어 있는 빈 Canvas(검정색)
```

```
img = np.zeros((700, 700, 4), np.uint8)
```

```
#img = cv2.line(img, (372,194), (372,194), blue_color, 5) #1
```

```
#img = cv2.line(img, (356,192), (356,192), blue_color, 5) #2
```

```
#img = cv2.line(img, (356,189), (356,189), blue_color, 5) #3
```

```
#img = cv2.line(img, (356,189), (356,189), blue_color, 5) #4
```

```
#img = cv2.line(img, (354,195), (354,195), blue_color, 5) #5
```

```
img = cv2.line(img, (358,148), (358,148), color, 5) #6
```

```
img = cv2.line(img, (354,145), (354,145), color, 5) #7
```

```
img = cv2.line(img, (391,241), (391,241), color, 5) #8
```

```
img = cv2.line(img, (366,221), (366,221), color, 5) #9
```

```
img = cv2.line(img, (385,323), (385,323), color, 5) #10
```

```
img = cv2.line(img, (368,289), (368,289), color, 5) #11
```

```
img = cv2.line(img, (226,80), (226,80), color, 5) #12
```

```
img = cv2.line(img, (224,80), (224,80), color, 5) #13
```

```
img = cv2.line(img, (115,169), (115,169), color, 5) #14
```

```
img = cv2.line(img, (133,169), (133,169), color, 5) #15
```

```
img = cv2.line(img, (150,315), (150,315), color, 5) #16
```

```
img = cv2.line(img, (166,284), (166,284), color, 5) #17
```

```
#왼쪽
```

```
img = cv2.line(img, (358,148), (391,241), color, 1) #6-8 어깨 - 팔꿈치
```

```
img = cv2.line(img, (391,241), (385,323), color, 1) #8-10 팔꿈치 - 손목
```

```
img = cv2.line(img, (358,148), (226,80), color, 1) #6-12 어깨 - 엉덩이
```

```
img = cv2.line(img, (226,80), (115,169), color, 1) #12-14 엉덩이 - 무릎
```

```
img = cv2.line(img, (115,169), (150,315), color, 1) #14-16 무릎 - 발목
```

```
#오른쪽
```

```
img = cv2.line(img, (354,145), (366,221), color, 1) #7-9 어깨 - 팔꿈치
```

```
img = cv2.line(img, (366,221), (368,289), color, 1) #9-11 팔꿈치 - 손목
```

```
img = cv2.line(img, (354,145), (224,80), color, 1) #7-13 어깨 - 엉덩이
```

```
img = cv2.line(img, (224,80), (133,169), color, 1) #13-15 엉덩이 - 무릎
```

```
img = cv2.line(img, (133,169), (166,284), color, 1) #15-17 무릎 - 발목
```

```
cv2.imshow('image', img)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

2-1. 영상 분석

• 필드에 대한 이해

▶ Request

URL

```
POST /pose/job HTTP/1.1
Host: cv-api.kakaobrain.com
Authorization: KakaoAK {REST_API_KEY}
```

Header

Name	Description	Required
Authorization	앱 인증 키 [내 애플리케이션] > [앱 설정] > [앱 키]에서 확인 가능	O

Parameter

Name	Type	Description	Required
video_url	String	처리할 영상의 URL HTTP(80포트)와 HTTPS(443 포트) 지원, 리다이렉트 미지원	O*
file	Binary	처리할 영상 파일	O*
smoothing	Boolean	검출된 프레임 사이의 키 포인트 위치를 smoothing 처리할 것인지 여부, true 또는 false (기본값: true)	X
callback_url	String	요청 처리 결과를 받을 콜백(callback) URL HTTPS(443 포트) 사용 권장, 리다이렉트 미지원 아래 Sample: Callback 과 같은 형태로 한 번 호출되며, 실패할 경우 재시도는 하지 않습니다.	X

* 조건부: video_url / file 중 하나 필수

▶ Response

Key

Name	Type	Description
job_id	String	포즈 분석을 요청한 영상의 Job ID

2-1. 영상 분석

- 실행 코드

Request

Method

POST

Request URL

https://cv-api.kakaobrain.com/pose/job

SEND

Parameters ^

Headers

Body

Variables

Header name

Header value

X

?

Authorization

KakaoAK [REDACTED]

Header name

Header value

X

?

Content-Type

multipart/form-data

Method

POST

Request URL

https://cv-api.kakaobrain.com/pose/job

SEND

Parameters ^

Headers

Body

Variables

Body content type

multipart/form-data

Field name

file

CHOOSE FILE

사람.mp4 (5904015 bytes)

X

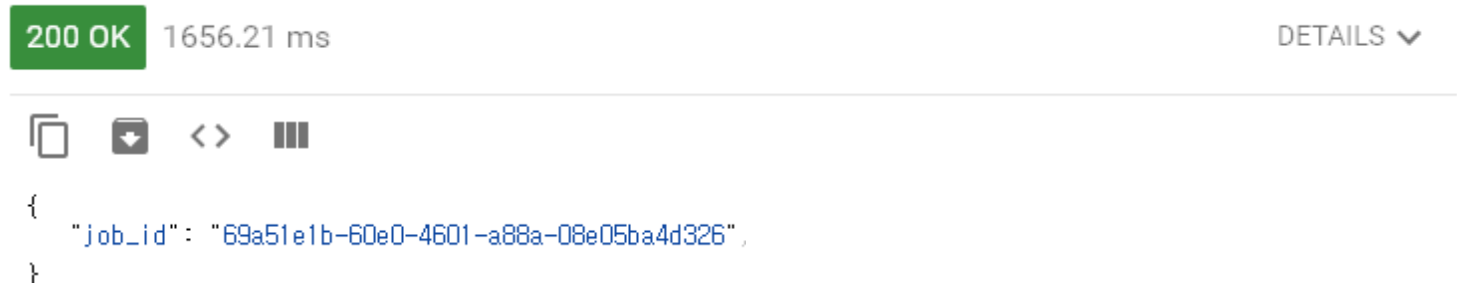
2-1. 영상 분석

- 입력 영상



2-1. 영상 분석

- 실행 결과



- 이 결과를 이용하여 영상 분석 결과 확인하기를 수행함.
- 이 과정을 수행하게 될 경우, 해당 영상은 다중 이미지로 처리되었고, 이를 사용하기 위한 식별자로 job_id를 반환한다.

2-2. 영상 분석 결과 확인하기

- 필드에 대한 이해

> Request

URL

```
GET /pose/job/{job_id} HTTP/1.1
Host: cv-api.kakaobrain.com
Authorization: KakaoAK {REST_API_KEY}
```

Header

Name	Description	Required
Authorization	앱 인증 키 [내 애플리케이션] > [앱 설정] > [앱 키]에서 확인 가능	O

Parameter

Name	Type	Description	Require
job_id	String	영상 분석하기 API 응답으로 받은 Job ID	O

2-2. 영상 분석 결과 확인하기

• 필드에 대한 이해

▶ Response

Key

Name	Type	Description
annotations	Annotation[]	프레임 수만큼의 크기를 가진 배열로 각 프레임(Frame) 별로 검출한 키 포인트의 좌표와 신뢰도를 담은 객체들의 목록 아래 Annotations 항목 참고 "status: success"인 경우에만 반환
categories	Category[]	키 포인트가 의미하는 정보를 담은 객체 아래 Categories 항목 참고 "status: success"인 경우에만 반환
info	Info	생성된 작업물에 대한 정보(버전, 생성 날짜, URL, 설명 등)를 담은 객체 "status: success"인 경우에만 반환
job_id	String	포즈 분석을 요청한 영상의 Job ID
status	String	요청에 대한 응답 상태: waiting, processing, success, failed, not found 중 하나 - waiting: 내부에서 대기 중 - processing: 처리 중 - success: 정상 - failed: 처리 실패 - not found: 해당 Job ID로 찾을 수 없는 경우나 결과 보관 기간이 7일을 지난 경우
video	Video	요청한 영상의 프레임에 대한 정보(초당 진행되는 프레임의 수, 영상의 전체 프레임 수, 프레임의 가로/세로 길이)를 담은 객체 "status: success"인 경우에만 반환
description	String	status가 "failed"인 경우에만 반환되는 요청 처리 실패 사유 예: "Failed to get video"

Annotation

Name	Type	Description
frame_num	Int	프레임의 번호, 0에서 n-1(n=프레임 수)
objects	Person[]	검출한 키 포인트의 좌표와 신뢰도를 담은 객체 객체에 대한 세부 파라미터는 이미지 분석하기 API의 Person 참고

Category

Name	Type	Description
id	Int	1로 고정 1: person
keypoints	String[]	17개의 키 포인트에 해당하는 신체 부위의 영문명을 담은 배열 ["nose", "left_eye", "right_eye", "left_ear", "right_ear", "left_shoulder", "right_shoulder", "left_elbow", "right_elbow", "left_wrist", "right_wrist", "left_hip", "right_hip", "left_knee", "right_knee", "left_ankle", "right_ankle"]
name	String	"person"으로 고정
skeleton	List<Int[]>	연결된 두 키 포인트 ID들을 담은 목록 예: [1, 2]는 코와 왼쪽 귀를 연결한 선을 의미
supercategory	String	"person"으로 고정

2-2. 영상 분석 결과 확인하기

- 실행 코드 – get 방식으로, job / {아까 응답으로 받은 job_id} 로 요청을 보낸다.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- Request URL:** `https://cv-api.kakaobrain.com/pose/job/69a51e1b-60e0-4601`
- Parameters:** None
- Headers:** Authorization: KakaoAK [redacted]
- Variables:** None

Buttons for 'SEND', 'Toggle source mode', 'Insert headers set', and a help icon are also visible.

2-2. 영상 분석 결과 확인하기

• 실행 결과

```
{
  "annotations": [
    {
      "frame_num": 0,
      "objects": [
        {
          "area": 779678.9235845946,
          "bbox": [
            30.68, 470.45, 1233.43, 632.12
          ],
          "category_id": 1,
          "keypoints": [
            577.36, 645.85, 0.71,
            685.33, 523.12, 0.83,
            472.38, 535.4, 0.82,
            844.29, 553.81, 0.26,
            349.41, 578.35, 0.22,
            1027.25, 1014.03, 0,
            184.45, 1029.38, 0,
            1132.22, 879.03, 0,
            133.47, 992.56, 0,
            1141.22, 875.97, 0,
            280.43, 1023.24, 0,
            1030.25, 1047.78, 0,
            277.43, 1023.24, 0,
            1159.22, 1026.31, 0,
            262.44, 1032.44, 0,
            391.4, 1044.72, 0,
            277.43, 1023.24, 0
          ],
          "object_id": 0,
          "score": 0.1673
        }
      ]
    }
  ],
  "score": 0.1663
}
```

```
{
  "frame_num": 1,
  "objects": [
    {
      "area": 882873.8916888427,
      "bbox": [
        0, 467.38, 1319.81, 668.94
      ],
      "category_id": 1,
      "keypoints": [
        574.7607058439822, 645.85, 0.71,
        684.0846463034507, 523.12, 0.83,
        469.7853882084444, 535.4, 0.82,
        843.0446463034507, 552.5342512461486, 0.26,
        349.41, 578.35, 0.21,
        1033.25, 1020.17, 0,
        229.44, 1017.1, 0,
        1060.24, 1041.65, 0,
        58.49, 992.56, 0,
        1156.22, 1026.31, 0,
        130.47, 992.56, 0,
        1075.24, 1044.72, 0,
        211.45, 995.62, 0,
        1075.24, 1044.72, 0,
        307.42, 1032.44, 0,
        391.4, 1047.78, 0,
        805.3, 1078.47, 0
      ],
      "object_id": 0,
      "score": 0.1663
    }
  ],
  "score": 0.1663
}
```

중략

```
{
  "frame_num": 219,
  "objects": [
    {
      "id": 1,
      "keypoints": [
        "nose",
        "left_eye",
        "right_eye",
        "left_ear",
        "right_ear",
        "left_shoulder",
        "right_shoulder",
        "left_elbow",
        "right_elbow",
        "left_wrist",
        "right_wrist",
        "left_hip",
        "right_hip",
        "left_knee",
        "right_knee",
        "left_ankle",
        "right_ankle"
      ],
      "name": "person",
      "skeleton": [
        [ 1, 2], [ 1, 3], [ 2, 3], [ 2, 4], [ 3, 5], [ 4, 6], [ 5, 7], [ 6, 7], [ 6, 8], [ 6, 12], [ 7, 9], [ 7, 13], [ 8, 10], [ 9, 11], [ 12, 13], [ 14, 12], [ 15, 13], [ 16, 14], [ 17, 15]
      ],
      "supercategory": "person"
    }
  ],
  "info": {
    "contributor": "Kakao Brain Corp.",
    "date_created": "2020/08/18",
    "description": "Human pose estimation result from Kakao Brain",
    "url": "https://www.kakaobrain.com",
    "version": 2,
    "year": 2020
  },
  "job_id": "69a51e1b-60e0-4601-a88a-08e05ba4d326",
  "status": "success",
  "video": {
    "fps": 23.976023976023978,
    "frames": 220,
    "height": 1080,
    "width": 1920
  }
}
```