



**amazon** Rekognition

**API AWS REKOGNITION  
RUBY ON JETS**



## SUMÁRIO

INTRODUÇÃO.....	07
DETECÇÃO DE FACE.....	08
COMPARAÇÃO DE FACES.....	11
PESQUISA E COMPARAÇÃO DE FACES EM VIDEO.....	14
REFERÊNCIAS.....	17

## 1 INTRODUÇÃO

O *Amazon Rekognition* é um serviço de análise de imagens e vídeos oferecido pela *Amazon Web Services (AWS)*. Ele usa algoritmos de aprendizado de máquina avançados para analisar, detectar e reconhecer objetos, rostos e emoções em imagens e vídeos.

Entre os principais recursos do *Amazon Rekognition* estão a detecção e reconhecimento de rostos, análise de emoções, detecção de texto em imagens, identificação e rastreamento de objetos, análise de conteúdo impróprio e até mesmo reconhecimento de celebridades e análise de vídeos. Seus principais recursos são:

- *Label Detection*:
  - O reconhecimento rotula automaticamente objetos, conceitos, cenas e ações em suas imagens e fornece uma pontuação de confiança;
- *Image Moderation*
  - O reconhecimento detecta automaticamente conteúdo adulto explícito ou sugestivo ou conteúdo violento em suas imagens e fornece pontuações de confiança;
- *Facial Analysis*
  - Faça uma análise completa dos atributos faciais, incluindo pontuações de confiança;
- *Face Comparison*
  - Compare os rostos para ver o quão perto eles combinam com base em uma porcentagem de similaridade;
- *Stored Video Analysis*
  - Analisa vídeos e detecta objetos, atividades, pessoas, celebridades em vídeos armazenados no *Amazon S3*.

Para a API desenvolvida em *Ruby on Rails*, utilizaremos apenas os recursos relacionados a análise e validação de faces, como o *Facial Analysis*, *Face Comparison* e *Stored Video Analysis*.

## 2 DETECÇÃO DE FACES

O recurso de *Facial Analysis* do AWS Rekognition é uma funcionalidade poderosa que permite detectar e reconhecer rostos em imagens e vídeos, além de fornecer informações detalhadas sobre as características faciais de uma pessoa.



### ▼ Results



looks like a face	99.9 %
appears to be female	99.9 %
age range	25 - 35 years old
not smiling	63.5 %
appears to be happy	70 %
wearing glasses	99.9 %
wearing sunglasses	99.8 %
eyes are open	98.7 %
mouth is open	88.3 %
does not have a mustache	98.3 %
does not have a beard	94.4 %

## Detect Faces – Ruby on Jets

Detect Faces		
POST /detect-faces		
Param	Type	Description
photo_image	base64 JPG type	A photo to be detected a face. <i>Required.</i>

### Request:

API Facial Recognition / detect-faces valid

POST ▼ http://127.0.0.1:8888/detect-faces

Params Authorization Headers **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```

1  {}
2  {
    "photo_image": "data:image/jpeg;base64,/9j/4QAYRXhpZgAASUkqAAgAAAAAAAAAAAAAAP/sAB
    +7vyIgaWQ9Ilc1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCI/
    PiA8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIiB4OnhtcHRrPSJBZG9iZS8yTVAgQ
    +IDxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3lu
  
```

### Response:

```

1  {}
2  {
3    "face_details": [
4      {
5        "bounding_box": {
6          "width": 0.234577938914299,
7          "height": 0.2554052770137787,
8          "left": 0.26640182733535767,
9          "top": 0.1076713353395462
10       },
11       "age_range": {
12         "low": 24,
13         "high": 34
14       },
15       "smile": {
16         "value": true,
17         "confidence": 95.7833023071289
18       },
19       "eyeglasses": {
20         "value": false,
21         "confidence": 93.75161743164062
22       },
23       "sunglasses": {
24         "value": false,
25         "confidence": 99.99553680419922
26       },
27       "gender": {
28         "value": "Male",
29         "confidence": 99.81163787841797
30       },
31       "beard": {
32         "value": true,
33         "confidence": 99.40918731689453
34       }
35     ],
36     "emotions": [
37       {
38         "type": "HAPPY",
39         "confidence": 99.44390106201172
40       },
41       {
42         "type": "SURPRISED",
43         "confidence": 6.352367877960295
44       },
45       {
46         "type": "FEAR",
47         "confidence": 5.896050002500566
48       },
49       {
50         "type": "SAD",
51         "confidence": 2.158233404159546
52       },
53       {
54         "type": "CONFUSED",
55         "confidence": 0.11366324126720428
56       },
57       {
58         "type": "DISGUSTED",
59         "confidence": 0.05947431921958923
60       },
61       {
62         "type": "CALM",
63         "confidence": 0.05574188381433487
64       },
65       {
66         "type": "ANGRY",
67         "confidence": 0.034142542630434036
68       }
69     ]
70   }
71 }
  
```

```

79      ],
80      "landmarks": [
81        {
82          "type": "eyeLeft",
83          "x": 0.30912041664123535,
84          "y": 0.19840118288993835
85        },
86        {
87          "type": "eyeRight",
88          "x": 0.40884700417518616,
89          "y": 0.19784659147262573
90        },
91        {
92          "type": "mouthLeft",
93          "x": 0.31800341606140137,
94          "y": 0.2811819016933441
95        },
96        {
97          "type": "mouthRight",
98          "x": 0.40116241574287415,
99          "y": 0.2811044454574585
100      },
101      {
102          "type": "nose",
103          "x": 0.3383001387119293,
104          "y": 0.23950877785682678
105      },
106      {
107          "type": "leftEyeBrowLeft",
108          "x": 0.2786557674407959,
109          "y": 0.18061554431915283
110      },
111      {
112          "type": "leftEyeBrowRight",
113          "x": 0.3300454454436003
213      "x": 0.29954299330711365,
214      "y": 0.29315751791000366
215      },
216      {
217          "type": "chinBottom",
218          "x": 0.362354040145874,
219          "y": 0.33883920311927795
220      },
221      {
222          "type": "midJawlineRight",
223          "x": 0.4762324392795563,
224          "y": 0.2919960021972656
225      },
226      {
227          "type": "upperJawlineRight",
228          "x": 0.49928414821624756,
229          "y": 0.20297737419605255
230      }
231      ],
232      "pose": {
233          "roll": -2.9681801795959473,
234          "yaw": -15.16724681854248,
235          "pitch": 7.80928373336792
236      },
237      "quality": {
238          "brightness": 93.36430358886719,
239          "sharpness": 92.22801208496094
240      },
241      "confidence": 99.99968719482422
242      }
243      }
244      ]

```

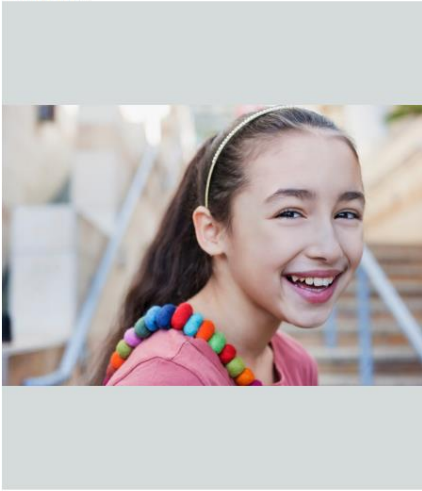
Descrição dos *labels response*:

- *face\_details*:
  - Detalhes encontrados na análise da imagem;
- *bounding\_box*:
  - As coordenadas da caixa delimitadora que circunda o rosto
- *emotions*:
  - Um conjunto de emoções com confiança na análise.
- *landmarks*:
  - Uma matriz de marcos faciais. Para cada ponto de referência (, como olho esquerdo, olho direito e boca ), fornece as coordenadas x-y
- *pose*:
  - Descreve a rotação do rosto dentro da imagem.
- *quality*:
  - Descreve o brilho e nitidez do rosto.
- *confidence*:
  - O nível de confiança que garante ser um rosto.


### 3 COMPARAÇÃO DE FACES

O recurso *Face Comparison* do *AWS Rekognition* é uma funcionalidade avançada que permite comparar duas ou mais imagens de rostos diferentes e determinar se pertencem à mesma pessoa. Ao enviar duas ou mais imagens para o serviço. O sistema comparará as imagens e retornará uma pontuação de confiança, indicando a probabilidade de as imagens pertencerem à mesma pessoa. A pontuação de confiança pode variar de 0 a 100, sendo que uma pontuação mais alta indica uma maior probabilidade de que as imagens sejam da mesma pessoa.


Reference face




Choose a sample image



Comparison faces



Choose a sample image



▼ Results

	=	
Similarity <div><div></div></div> 99.9 %		
	≠	
	≠	

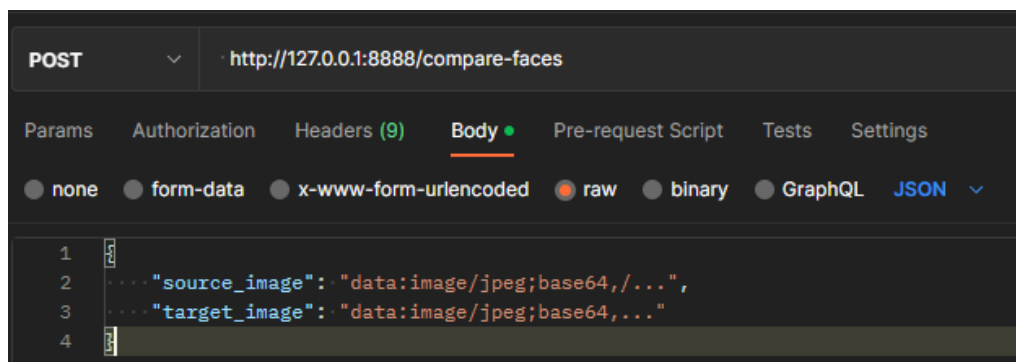
## Compare Faces - Ruby on Jets

🔗 Compare faces

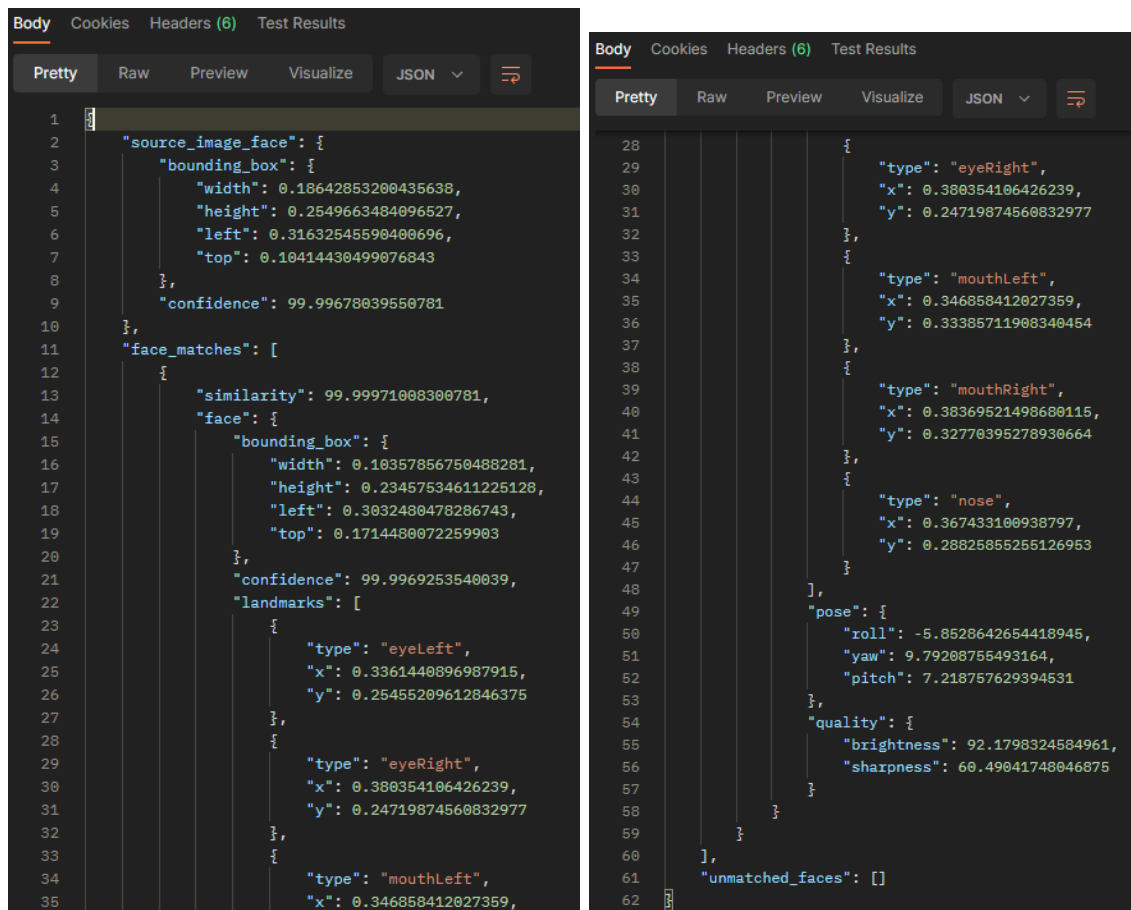
**POST** /compare-faces

Param	Type	Description
source_image	base64 JPG type	A face photo to be used as a source. <i>Required</i>
target_image	base64 JPG type	A face photo to be used as a target. <i>Required</i> .

Request:



Response:





Descrição dos *labels response*:

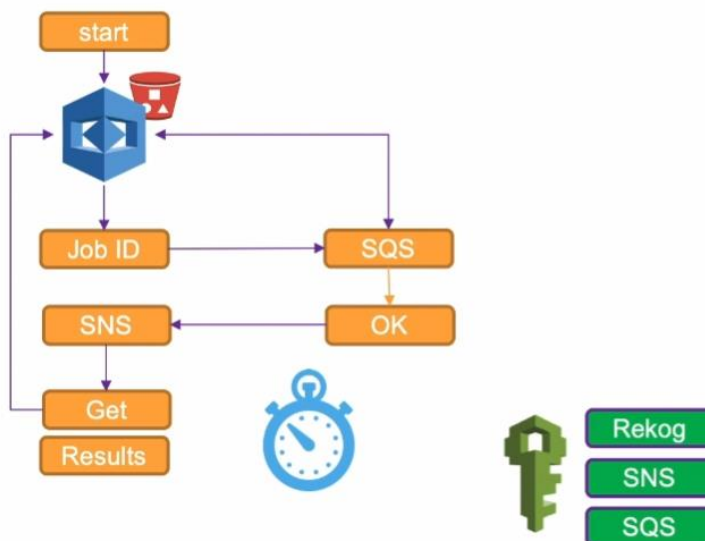
- **source\_image\_face:**
  - O resposta inclui informações sobre o rosto da imagem de origem que foi usado para comparação, incluindo a caixa delimitadora e o valor da confiança.
- **face\_matches:**
  - Contém uma lista de correspondências de rostos que foram encontrados nas imagens comparadas.
- **similarity:**
  - A pontuação de confiança para a correspondência de rosto, que varia de 0 a 100. Quanto maior a pontuação, maior a probabilidade de que as duas imagens sejam do mesmo rosto.
- **face:**
  - Informações detalhadas sobre o rosto encontrado na imagem, incluindo a posição e a orientação do rosto e os recursos faciais detectados.
- **unmatched\_faces:**
  - Contém informações detalhadas sobre cada imagem de rosto que não correspondia a nenhuma outra imagem na comparação.

## 4 PESQUISA E COMPARAÇÃO DE FACES EM VIDEO

O recurso *Searching stored videos for faces* do AWS Rekognition é uma funcionalidade busca em uma coleção de rostos que correspondam às faces das pessoas detectadas em um vídeo armazenado ou em um streaming de vídeo.

As faces que você procura devem primeiro ser indexadas em uma coleção usando ***IndexFaces***.

A pesquisa de rosto do Amazon Rekognition Video segue o fluxo de trabalho **assíncrono**, analisam vídeos armazenados em um S3. Para começar a procurar rostos em um vídeo, é utilizado o ***StartFaceSearch*** e forneça o ID da ***collection*** que você quer pesquisar. O Rekognition publica o status de conclusão da análise de vídeo em um Tópico do ***Simple Notification Service (SNS)***. Se a análise de vídeo for bem-sucedida, com o método ***GetFaceSearch*** é possível obter os resultados.



Para esse processo é necessário:

- Vídeo armazenado no S3 (formato MOV ou MP4 máximo 8GB)
- Configurar um tópico no SNS
- Configurar uma Role IAM

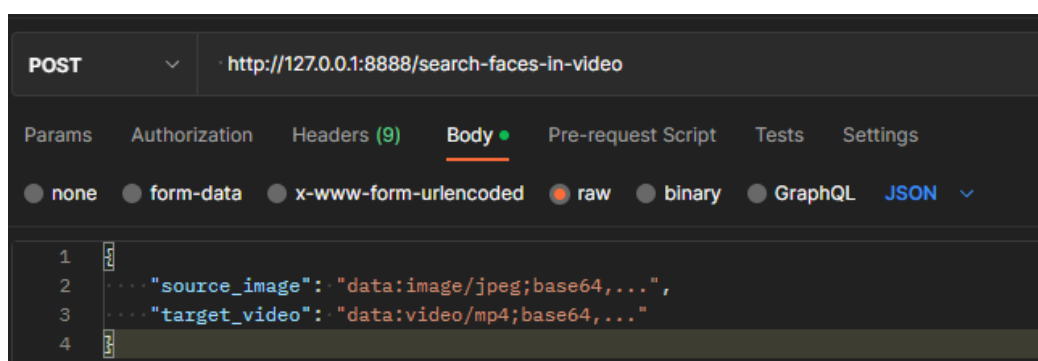
## Start Face Detection – Ruby on Jets

Start Face Detection

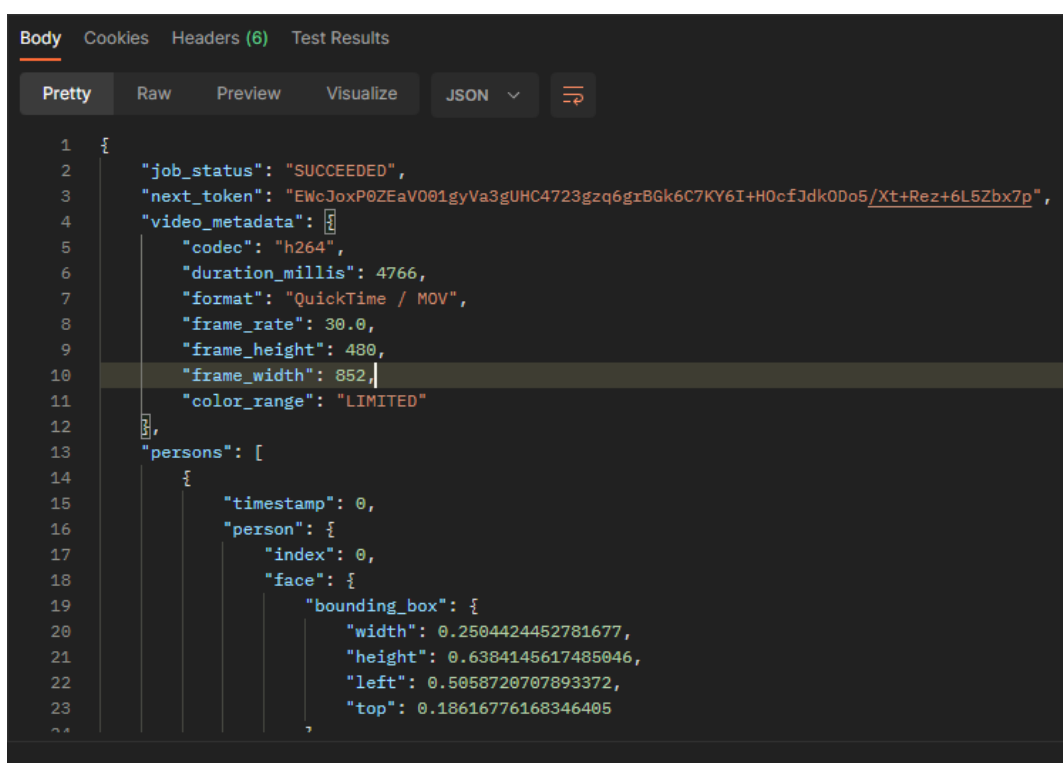
**POST** `/search-faces-in-video`

Param	Type	Description
<code>source_image</code>	base64 JPG type	A face photo to be used as a source. <i>Required</i>
<code>target_video</code>	base64 MP4 type	A video to be used as a target. <i>Required</i> .
<code>async</code>	boolean	Set asynchronous operation. <i>Default is FALSE.</i>

Request:



Response (por padrão **async é false**):



```

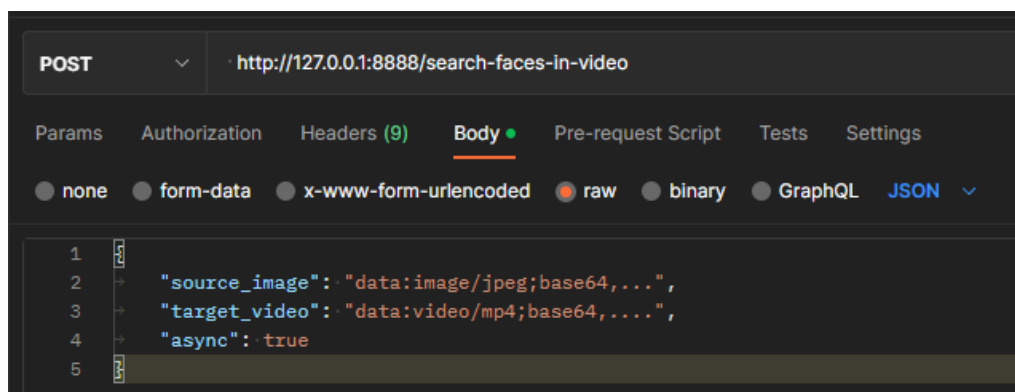
13  "persons": [
14    {
15      "timestamp": 0,
16      "person": {
17        "index": 0,
18        "face": {
19          "bounding_box": {
20            "width": 0.2504424452781677,
21            "height": 0.6384145617485046,
22            "left": 0.5058720707893372,
23            "top": 0.18616776168346405
24          },
25          "landmarks": [
26            {
27              "type": "eyeLeft",
28              "x": 0.6268179416656494,
29              "y": 0.41678816080093384
30            },
31            {
32              "type": "eyeRight",
33              "x": 0.7189028859138489,
34              "y": 0.4241822063922882
35            }
36          ]
37        }
38      }
39    }
40  ],
41  "pose": {
42    "roll": 6.670616149902344,
43    "yaw": 31.001371383666992,
44    "pitch": 15.067915916442871
45  },
46  "quality": {
47    "brightness": 65.55905151367188,
48    "sharpness": 89.85481262207031
49  },
50  "confidence": 99.9998550415039
51  }
52  ]
53  }
54  }
55  }
56  }
57  }
58  }
59  }
60  }
61  }
62  }
63  }
64  "face_matches": [
65    {
66      "similarity": 99.99674987792969,
67      "face": {
68        "face_id": "13dd2502-8d0b-4fa8-b53a-ca53a38e1f29",
69        "bounding_box": {
70          "width": 0.237746000289917,
71          "height": 0.289669007062912,
72          "left": 0.44290998578071594,
73          "top": 0.279558002948761
74        },
75        "image_id": "e548c49f-663a-3ff8-96b9-520991f152a7",
76        "external_image_id": "bea2f2e0-bd15-013b-3f8b-28a159459eb6",
77        "confidence": 99.98380279541016
78      }
79    }
80  ]
81  }
82  ]
83  }

```

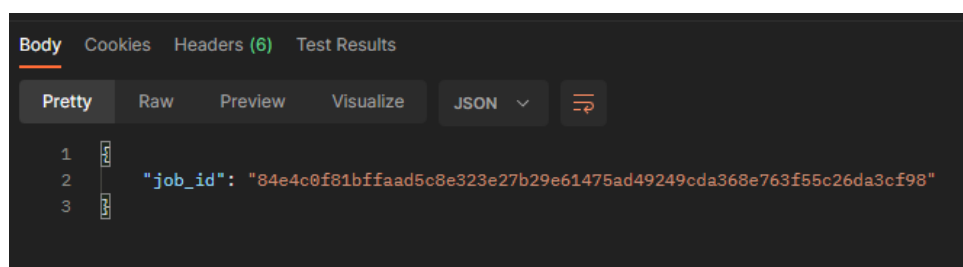
Descrição dos *labels response*:

- **job\_id**: ID do job que foi criado para análise da face em vídeo;
- **job\_status**: Status da execução do job.
- **next\_token**: Caso o response seja dividido em página usar esse campo.
- **video\_metadata**: Metadados do vídeo que foi analisado.
- **Persons**: Dados da pessoa encontrada no vídeo, como: face, quality, landmarks e confidence.
- **face\_matches**: Contém uma lista de correspondências de rostos que foram encontrados na comparação entre imagem e vídeo.
- **Similarity**: A pontuação de confiança para a correspondência de rosto.
- **face\_id**: ID da face que foi indexada na collection e correspondida com a face do vídeo.

Request com **async true**:



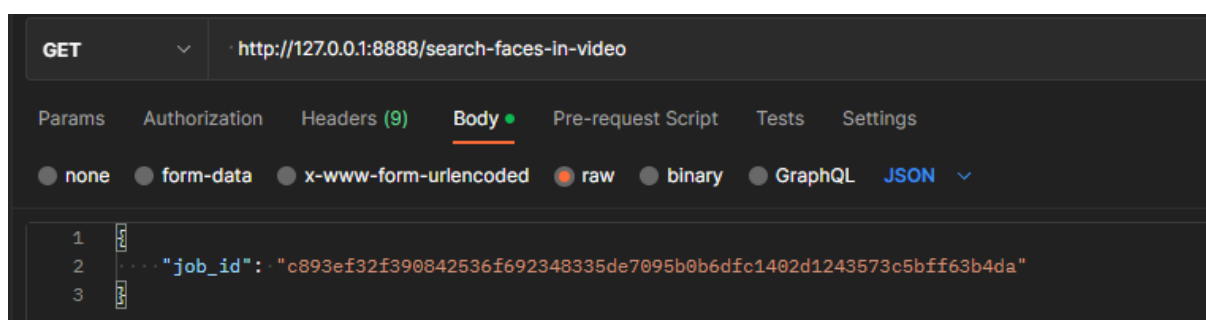
Response **async true**:



Descrição dos *labels response*:

- **job\_id**: ID do job que foi criado para análise da face em vídeo;

Para consultar o job é necessário fazer uma request GET informando o **job\_id** no body:



Response será o mesmo do método `async false`.

## REFERÊNCIAS

- AWS Rekognition:
  - <https://docs.aws.amazon.com/rekognition/latest/APIReference/Welcome.html>
- API\_DetectFaces:
  - [https://docs.aws.amazon.com/rekognition/latest/APIReference/API\\_DetectFaces.html](https://docs.aws.amazon.com/rekognition/latest/APIReference/API_DetectFaces.html)
- API\_CompareFaces:
  - [https://docs.aws.amazon.com/rekognition/latest/APIReference/API\\_CompareFaces.html](https://docs.aws.amazon.com/rekognition/latest/APIReference/API_CompareFaces.html)
- API\_StartFaceDetection:
  - [https://docs.aws.amazon.com/rekognition/latest/APIReference/API\\_StartFaceDetection.html](https://docs.aws.amazon.com/rekognition/latest/APIReference/API_StartFaceDetection.html)