

Nabla Vision Platform

Basic Face Recognition API

API リファレンス V1.0.1

概要	1
著作権	1
改訂履歴	1
1. サービス概要	2
2. 認証方法とデータ管理	3
2.1. API URL	3
2.2. 認証方法	3
2.3. データ管理	3
2.4. レスポンスデータのフォーマット	3
2.5. 顔画像の制限	4
3. API エンドポイント	5
3.1. 顔検知	5
3.2. 顔の属性取得	9
3.3. 顔画像の品質確認	13
3.4. 顔の特徴量抽出	17
3.5. 顔画像比較（1 対 1）	19
3.6. 顔の特徴量比較（1 対 1）	22
3.7. 特徴量データベース作成	24
3.8. 特徴量データベース情報取得	26
3.9. 特徴量データベース削除	28
3.10. 特徴量データベース一覧取得	30
3.11. 特徴量データベースに特徴量追加	32
3.12. 特徴量データベースから特徴量取得	34
3.13. 特徴量更新	36
3.14. 特徴量削除	38
3.15. 画像で特徴量検索（1 対 N 認証）	40
4. レスポンスコード	42
4.1. 顔検出のエラーコード	42
4.2. 顔の品質と属性のエラーコード	43
4.3. 1 対 N 顔認識のエラーコード	44

本ドキュメントについて

概要

本ドキュメントは、ナブラワークス株式会社が提供する（以降、当社）Nabla Vision Platform における Basic Face Recognition API（以降、本サービス）の API リファレンスです。本ドキュメントは、開発者の方を対象に作成しています。

著作権

本サービスおよびそのドキュメントに関するすべての著作権およびその他の知的財産権は、当社に帰属します。本サービスおよびそのドキュメントの利用は、使用許諾契約および利用規約に従います。詳細については、該当する契約書および規約をご参照ください。

改訂履歴

ドキュメント Ver.	システム Ver.	日付	改訂内容
1.0.0	1.0.0	2025 年 1 月 31 日	初版
1.0.1	1.0.1	2025 年 2 月 27 日	「3.10 特徴量データベース一覧取得」送信フォーマットとパラメータ修正

1. サービス概要

本サービスは、お客様のニーズに合わせた汎用性の高い顔認証 API を提供しています。このドキュメントでは、API の使用方法について説明します。

2. 認証方法とデータ管理

2.1. API URL

本サービスはクラウドサービスであり、以下の以下の URL ルールを使用してサービスにアクセスできます：

`https://frecognition.nablacloud.com/api/<endpoint>`

2.2. 認証方法

本サービスでは、認証のために HTTPS ヘッダー内で x-app-id と x-app-key を使用します。これらの認証情報は、ナブワークス株式会社との契約締結時に発行されます。「顔検知」の機能の例は以下です。

Item	Sample	Description
API URL	<code>https://frecognition.nablacloud.com/api/face/v2/detect</code>	アクセスする API の URL
Header: Content-Type	<code>multipart/form-data</code>	Http Header の Content-Type ヘッダー
Header: x-app-id	<code>95aef116-5876-4394-b1ad-796a80f701b9</code>	契約時に発行されるアプリケーション ID
Header: x-app-key	<code>2ItS....._ AVU</code>	DeveloperDashboard で発行できるアプリキー
Body: file	<code>/home/nabla/test_man.jpg</code>	顔検知用画像ファイルの保存パス
Body: rotate	<code>false</code>	画像ファイルの回転を有効とするパラメーター

2.3. データ管理

本サービスのすべての API エンドポイントは、安全なデータ交換を保証するために HTTPS を使用します。すべての顔画像ファイルやデータはサーバー側に保存されません。顔認識サーバーは、お客様を除く第三者サービスとのプライバシーデータの通信を行いません。

2.4. レスポンスデータのフォーマット

本サービスでは、すべての HTTP リクエストに対して同じフォーマットの JSON データを返します。JSON データには 3 つの基本フィールドが含まれます。

Field	Type	Description
code	String	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。

msg	String	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	Object	レスポンスの詳細データです。このフィールドは、呼び出した API に応じて内容が異なります。

2.5. 顔画像の制限

本サービスを利用するためにアップロードする画像ファイルにはいくつかの制限があります。

- ・ フォーマットは JPG、JPEG、BMP、PNG、TIFF である必要があります。
- ・ 画像の幅と高さは 8px 以上 5000px 以下である必要があります。
- ・ 画像ファイルのサイズは 5MB 以下である必要があります。
- ・ 顔の最小ピクセル数は 30px です。

3. API エンドポイント

3.1. 顔検知

- API の概要

画像内の顔を検出します。顔の座標点と顔の重要なポイントをマークします。複数の顔検出に対応しています。

- 定義

Invoke Method: **POST**

Request Body Format: **multipart/form-data**

Request URL: **https://<service domain>/api/face/v2/detect**

- リクエストパラメーター

Http Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Http Body:

Parameter	Type	Required	Description
file	file	No	バイナリファイル形式の画像をアップロードしてください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
data	string	No	Base64 エンコードされたバイナリ画像データを入力してください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
rotate	boolean	No	true: 回転する false: 回転しない パラメーターのデフォルト値は false です。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。

data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。
------	---------------	------------------------------

レスポンス詳細:

Parameter	Type	Description
numOfFaces	number	アップロードされた画像で検出された顔の数。
rotation	enum	0: 0 度 1: 90 度 2: 180 度 3: 270 度 rotate パラメーターの値が false の場合、0 が返されます。
context	string	アップロードされた画像の顔検出から得られたデータ。 これは他の API への入力として使用できます。この文字列は暗号化されています。
faces	array[Object]	アップロードされた画像の顔検出から得られた顔のフレームとキーポイントのリスト。
faceId	string	顔 ID 番号。アップロードされた画像に顔が 1 つだけの場合、faceId は 0 です。 アップロードされた画像に複数の顔がある場合、faceId は 0 から始まり、順番に増加します。
rect	object	顔の矩形ボックスの位置。
vertex	object	顔の矩形ボックスの位置は以下のプロパティを含みます。 各プロパティの値は整数です。
top	number	矩形ボックスの左上隅の縦座標（ピクセル単位）。
left	number	矩形ボックスの左上隅の横座標（ピクセル単位）。
right	number	矩形ボックスの右下隅の横座標（ピクセル単位）。
bottom	number	矩形ボックスの右下隅の縦座標（ピクセル単位）。
revisedVertex	object	顔の矩形ボックスの位置は以下のプロパティを含みます。各プロパティの値は整数です rotate パラメーターが「true」として渡される場合、正の回転に合わせた顔の矩形ボックスの位置が返されます。 rotate パラメーターが「false」として渡される場合、このフィールドは返されません。
left	number	矩形ボックスの左上隅の縦座標（ピクセル単位）。
top	number	矩形ボックスの左上隅の横座標（ピクセル単位）。
right	number	矩形ボックスの右下隅の横座標（ピクセル単位）。
bottom	number	矩形ボックスの右下隅の縦座標（ピクセル単位）。
confidence	number	顔の矩形ボックス位置の信頼度。
landmarks	object	顔のキーポイント座標の配列。
points	array[Object]	顔のキーポイントの座標のリスト。以下のプロパティを含み、それぞれは float 型の値です。
x	number	x: キーポイントの横座標。
y	number	y: キーポイントの縦座標。
occlusion	enum	顔のキーポイントが隠れているかどうかの結果を返します。 • 0: 隠れている • 1: 隠れていない
revisedPoints	array[Object]	顔のキーポイントの座標のリスト。以下のプロパティを含み、それぞれは float 型の値です。
x	number	x: キーポイントの横座標。
y	number	y: キーポイントの縦座標。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "numOfFaces": 1,
    "rotation": 0,
    "context": "1f8b0800...583e0b0000",
    "faces": [
      {
        "faceId": "0",
        "rect": {
          "vertex": {
            "left": 667,
            "top": 208,
            "right": 852,
            "bottom": 637
          },
          "revisedVertex": {
            "left": 804,
            "top": 656,
            "right": 771,
            "bottom": 743
          },
          "confidence": 0.8
        },
        "landmarks": {
          "points": [
            {
              "x": 17.28,
              "y": 49.36
            }
          ],
          "occlusion": [
            1
          ],
          "revisedPoints": [
            {
              "x": 30.70,
              "y": 85.59
            }
          ]
        }
      }
    ]
  }
}
```

```
}  
  }  
    }  
      }  
        }
```

3.2. 顔の属性取得

- API の概要

顔の属性を認識します。顔の各属性に関する情報を返します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/advanced/v2/attribute

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
file	file	No	バイナリファイル形式の画像をアップロードしてください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
data	string	No	Base64 エンコードされたバイナリ画像データを入力してください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
rotate	boolean	No	true: 回転する false: 回転しない パラメーターのデフォルト値は false です。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
-----------	------	-------------

numOfFaces	number	アップロードされた画像で検出された顔の数。
rotation	enum	アップロードされた画像の正しい角度への回転角度。 0: 0 度 1: 90 度 2: 180 度 3: 270 度
face	array[Object]	顔検出結果。
faceId	string	顔 ID 番号。アップロードされた画像に顔が 1 つだけの場合、faceId は 0 です。 複数の顔がアップロードされた画像に含まれている場合、faceId は 0 から始まり、順番に増加します。
attribute	array[Object]	顔の属性の検出。
key	string	顔の属性キー。
value	string	顔の属性値。
confidence	number	顔の属性の信頼度。 推奨される閾値は 0.8~0.9 です。（実際には、特定のビジネスシナリオに応じて閾値を選択できます）

属性キーと値の関係は次の通りです:

Key	Attribute	Value	Description
1	age	number	推定年齢
2	beard	0	髭なし
		1	髭あり
3	Not supported		Not supported
4	mask	0	マスクなし
		1	マスクあり
5	mask position	0	マスクなし
		1	マスクを着用した際に鼻孔が露出している（通常ではない着用）
		2	マスクを着用した際に口が露出している（通常ではない着用）
6	wearing glasses	0	メガネなし
		1	透明メガネ
		2	サングラス
7	gender	0	女性
		1	男性
8	Not supported		Not supported
9	expression	0	怒り
		1	喜び
		2	悲しみ
		3	冷静
		4	驚き
		5	吐き気
		6	怯え
		15	あくび
10	hat	0	帽子なし
		1 or 2	帽子あり

11	Not supported		Not supported
12	Helmet	0	ヘルメットなし
		1	ヘルメットあり

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "numOfFaces": 1,
    "face": [
      {
        "faceId": "0",
        "attribute": [
          {
            "key": "1",
            "value": "27",
            "confidence": 1.0
          },
          {
            "key": "2",
            "value": "0",
            "confidence": 0.986
          },
          {
            "key": "4",
            "value": "0",
            "confidence": 0.993
          },
          {
            "key": "6",
            "value": "0",
            "confidence": 0.999
          },
          {
            "key": "7",
            "value": "0",
            "confidence": 1.0
          },
          {
            "key": "9",
            "value": "1",

```

```
        "confidence": 0.238
      },
      {
        "key": "10",
        "value": "0",
        "confidence": 0.928
      },
      {
        "key": "12",
        "value": "0",
        "confidence": 1.0
      },
      {
        "key": "5",
        "value": "2",
        "confidence": 1.0
      }
    ]
  },
  "rotation": 0
}
```

3.3. 顔画像の品質確認

- API の概要

顔画像の品質を認識します。照明、顔のポーズ、ブレ、口の状態、目の状態に関する情報を返します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/advanced/v2/quality

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
file	file	No	バイナリファイル形式の画像をアップロードしてください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
data	string	No	または Base64 エンコードされたバイナリ画像データを入力してください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
rotate	boolean	No	true: 回転する false: 回転しない パラメーターのデフォルト値は false です。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
numOfFaces	number	アップロードされた画像で検出された顔の数。
rotation	enum	アップロードされた画像の正しい角度への回転角度。 0: 0 度 1: 90 度 2: 180 度 3: 270 度
pass	boolean	顔の品質が合格か不合格かを示します。 • true: 合格 • false: 不合格
face	array[Object]	顔検出結果。
faceId	string	顔 ID 番号。
quality	object	品質属性。
score	number	品質スコア。
light	object	照明情報。
mean	number	平均輝度。
var	number	照明の分散。
over_dark	number	80 未満のピクセル輝度の割合。
over_glare	number	200 以上のピクセル輝度の割合。
left_eyestate	object	左目の状態情報。
valid	number	信頼度。
open	number	目が開いているかどうか。
right_eyestate	object	右目の状態情報。
valid	number	信頼度。
open	number	目が開いているかどうか。
headpose	object	顔のポーズ情報。
yaw	number	頭を振る角度。
pitch	number	頭を上げる角度。
roll	number	回転角度（平面）。
blur	number	顔のぼかし分析結果。値が大きいほど顔が鮮明です。
mouth	number	口の状態情報。値が大きいほど口が閉じている状態に近づきます。
occlusion	object	キーポイントが隠れているかどうか。
left_eye	number	左目の隠蔽スコア。1 に近いほど隠蔽が少ないです。
right_eye	number	右目の隠蔽スコア。1 に近いほど隠蔽が少ないです。
nose	number	鼻の隠蔽スコア。1 に近いほど隠蔽が少ないです。
mouth	number	口の隠蔽スコア。1 に近いほど隠蔽が少ないです。
right_cheek	number	右頬の隠蔽スコア。1 に近いほど隠蔽が少ないです。
left_cheek	number	左頬の隠蔽スコア。1 に近いほど隠蔽が少ないです。

レスポンスデータサンプル:

```
{
  "code": "0000",
```

```
"msg": "success",
"data": {
  "numOfFaces": 1,
  "face": [
    {
      "faceId": "0",
      "quality": {
        "score": 0.90141719,
        "light": {
          "mean": 146.97772,
          "var": 0.6296765,
          "over_dark": 0.014186088,
          "over_glare": 0.013131
        },
        "left_eyestate": {
          "valid": 0.99738294,
          "open": 0.0021450324
        },
        "right_eyestate": {
          "valid": 0.99738294,
          "open": 0.0021450324
        },
        "headpose": {
          "yaw": -7.955658,
          "pitch": 10.906641,
          "roll": -8.941551
        },
        "blur": 0.9996469,
        "mouth": 1,
        "occlusion": {
          "left_eye": 0.9890352,
          "right_eye": 0.9923968,
          "nose": 0.99636805,
          "mouth": 0.9993784,
          "right_cheek": 0.99348027,
          "left_cheek": 0.9550856
        }
      }
    }
  ],
  "rotation": 0,
  "pass": false
}
```

```
}  
}
```

3.4. 顔の特徴量抽出

- API の概要

単一の画像から顔の特徴量を抽出します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/face/v2/extract

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
file	file	No	バイナリファイル形式の画像をアップロードしてください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
data	string	No	Base64 エンコードされたバイナリ画像データを入力してください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
context	string	No	顔検出 API のレスポンスフィールド。顔検出の処理データ。このフィールドを使用して、顔検出 API の処理を繰り返さないようにします。この値は、顔検出インターフェースの出力内の「context」フィールドです。ファイル、データ、またはコンテキストのいずれかを入力する必要があります。すべて入力された場合、優先順位はファイル > データ > コンテキストの順です。。
index	string	No	-2: 最も大きな顔 -1: すべての顔 n: face id = n の顔 (n は自然数で、入力パラメーターが context の場合にのみサポートされます) このパラメーターが指定されていない場合、デフォルトで -2 が設定されます。
rotate	boolean	No	true: 回転する false: 回転しない このパラメーターのデフォルト値は false です。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
numOfFaces	number	アップロードされた画像で検出された顔の数。
rotation	enum	アップロードされた画像の正しい角度への回転角度。 0: 0 度 1: 90 度 2: 180 度 3: 270 度
faces	array[Object]	アップロードされた画像を検出した顔のリスト。
faceId	string	顔 ID 番号。 アップロードされた画像に顔が 1 つしかない場合、faceId は 0 になります。 アップロードされた画像に複数の顔がある場合、faceId は 0 から始まり、累積されます。
feature	string	暗号化された faceId の特徴量文字列。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "numOfFaces": 1,
    "rotation": 0,
    "faces": [
      {
        "faceId": "0",
        "feature": "eyJ.....ifQ=="
      }
    ]
  }
}
```

3.5. 顔画像比較（1対1）

- API の概要

2つの顔画像を比較し類似度を算出します。

- 定義

Invoke Method: **POST**

Request Body Format: **multipart/form-data**

Request URL: **https://<service domain>/api/face/v2/compareImages**

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
firstFile	file	No	1 番目の画像をバイナリファイルとしてアップロードしてください。 firstFile、firstData、または firstContext のいずれかを入力する必要があります。すべてが入力された場合、firstFile が最優先され、次に firstData、最後に firstContext が優先されます。
firstData	string	No	1 番目の画像の Base64 エンコードされたバイナリ画像データを入力してください。 firstFile、firstData、または firstContext のいずれかを入力する必要があります。すべてが入力された場合、firstFile が最優先され、次に firstData、最後に firstContext が優先されます。
firstContext	string	No	3.1 顔検出 API のレスポンスフィールド。顔検出の処理データ。顔検出 API の処理を繰り返さないようにこのフィールドを使用します。 値は顔検出インターフェースの出力にある「context」フィールドです。 firstFile、firstData、または firstContext のいずれかを入力する必要があります。すべてが入力された場合、firstFile が最優先され、次に firstData、最後に firstContext が優先されます。
firstIndex	string	No	最初の画像から顔を選択し、その顔のプロパティを返します。このパラメーターの値は次の通りです： -2: 最も大きな顔 -1: すべての顔 n: faceId = n の顔（n は自然数で、入力パラメーターが context の場合のみサポートされます） このパラメーターが指定されない場合、デフォルトで -2 が入力されます。
firstRotate	boolean	No	1 番目の画像を回転させるかどうかを指定し、画像の方向結果を返します。

			true: 回転する false: 回転しない 注意: パラメーターのデフォルト値は false です。
secondFile	file	No	2 番目の画像をバイナリファイルとしてアップロードしてください。 secondFile、secondData、または secondContext のいずれかを入力する必要があります。すべてが入力された場合、secondFile が最優先され、次に secondData、最後に secondContext が優先されます。
secondData	string	No	2 番目の画像の Base64 エンコードされたバイナリ画像データを入力してください。 secondFile、secondData、または secondContext のいずれかを入力する必要があります。すべてが入力された場合、secondFile が最優先され、次に secondData、最後に secondContext が優先されます。
secondContext	string	No	3.1 顔検出 API のレスポンスフィールド。顔検出の処理データ。顔検出 API の処理を繰り返さないようにこのフィールドを使用します。 値は顔検出インターフェースの出力にある「context」フィールドです。 secondFile、secondData、または secondContext のいずれかを入力する必要があります。すべてが入力された場合、secondFile が最優先され、次に secondData、最後に secondContext が優先されます。
secondIndex	string	No	2 番目の画像から顔を選択し、その顔のプロパティを返します。このパラメーターの値は次の通りです： -2: 最も大きな顔 -1: すべての顔 n: faceId = n の顔（n は自然数で、入力パラメーターが context の場合のみサポートされます）。 このパラメーターが指定されない場合、デフォルトで -2 が入力されます。
secondRotate	boolean	No	2 番目の画像を回転させるかどうかを指定し、画像の方向結果を返します。 true: 回転する false: 回転しない 注意: パラメーターのデフォルト値は false です。

• レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
firstRotation	enum	1 番目の画像の正しい角度への回転角度。 rotate パラメーターの値が「true」として渡された場合、受け取った画像の回転角度が時計回りに調整されて返されます。 0: 0 度 1: 90 度

		2: 180 度 3: 270 度 firstRotate パラメーターの値が「false」として渡された場合、0 が返されます。
secondRotation	enum	2 番目にアップロードされた画像の正しい角度への回転角度。 rotate パラメーターの値が「true」として渡された場合、受け取った画像の回転角度が時計回りに調整されて返されます。 0: 0 度 1: 90 度 2: 180 度 3: 270 度 secondRotate パラメーターの値が「false」として渡された場合、0 が返されます。
result	array[Object]	比較の結果
firstFaceId	string	1 番目の画像の顔 ID 番号。アップロードされた画像に顔が 1 つだけの場合、faceId は 0 です。 複数の顔がアップロードされた画像に含まれている場合、faceId は 0 から始まり、順番に増加します。
secondFaceId	string	2 番目の画像の顔 ID 番号。アップロードされた画像に顔が 1 つだけの場合、faceId は 0 です。 複数の顔がアップロードされた画像に含まれている場合、faceId は 0 から始まり、順番に増加します。
similarity	number	比較スコア、2 つの特徴量の類似度。
pass	boolean	比較が成功したか失敗したか。 true: 比較に成功。 false: 比較に失敗。

レスポンスデータサンプル:

```

{
  "code": "0000",
  "msg": "success",
  "data": {
    "result": [
      {
        "firstFaceId": "0",
        "secondFaceId": "0",
        "similarity": 0.9312,
        "pass": true
      }
    ],
    "firstRotation": 0,
    "secondRotation": 0
  }
}

```

3.6. 顔の特徴量比較（1対1）

- API の概要

2 つの特徴量を比較し類似度を算出します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/face/v2/verify

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
feature1	string	Yes	3.4 顔の特徴量抽出 API から取得した特徴量 1。
feature2	string	Yes	3.4 顔の特徴量抽出 API から取得した特徴量 2。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
similarity	number	比較スコア、2 つの特徴量の類似度。
pass	boolean	比較が成功したか失敗したか。 true: 比較に成功 false: 比較に失敗

レスポンスデータサンプル:

```
{  
  "code": "0000",  
  "msg": "success",  
  "data": {  
    "similarity": 0.931,  
    "pass": true  
  }  
}
```

3.7. 特徴量データベース作成

- API の概要

名前とストレージ容量の情報に基づいて、新しい特徴量データベースを追加します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/feature/v2/addDb

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
dbName	string	Yes	特徴量データベース名。命名規則では、数字と英語が使用できます。フィールドの最大長は 20 バイトです。数字のみのデータベース名は使用できません。
maxSize	string	Yes	特徴量データベースの最大容量（MySQL モードでは、各データベースの現在の最大容量の合計が認証の最大容量を下回る必要があります）。
dbType	string	No	特徴量データベース内の特徴量タイプ。デフォルト値は「Face」です。（入力する必要はありません）。
nameSpace	string	No	名前空間。（入力する必要はありません）
tempSpace	string	No	一時領域。（入力する必要はありません）。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
dbId	string	特徴量データベースの ID。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "dbId": "TEST13"
  }
}
```

3.8. 特徴量データベース情報取得

- API の概要

特徴量データベース ID を指定して、対象の特徴量データベース情報を取得します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/feature/v2/getDbInfo

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
dbId	string	Yes	特徴量データベース ID 番号

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
dbId	string	特徴量データベースの ID 番号。
dbName	string	特徴量データベースの名前。
dbType	string	特徴量データベースの種類。
maxSize	string	特徴量データベース最大レコード数。
currentSize	number	特徴量データベースの現在レコード数。
normalizeSrc	string	正規化された特徴量の入力値

normalizeDst	string	正規化された特徴量の出力値
--------------	--------	---------------

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "dbId": "TEST13",
    "dbName": "TEST13",
    "dbType": "Face",
    "maxSize": "1000",
    "currentSize": 0,
    "normalizeSrc": "{-1f,0.3f,0.352f,0.385f,0.395f,0.425f,0.46f,1f}",
    "normalizeDst": "{0f,0.4f,0.5f,0.6f,0.7f,0.85f,0.95f,1f}"
  }
}
```

3.9. 特徴量データベース削除

- API の概要

特徴量データベース ID を指定して、特徴量データベースを削除します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/feature/v2/deleteDb

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
dbId	string	Yes	特徴量データベースの ID 番号。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
dbId	string	特徴量データベースの ID 番号。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
```

```
  "data": {  
    "dbId": "TEST13"  
  }  
}
```

3.10. 特徴量データベース一覧取得

- API の概要

特徴量データベースのリスト情報を取得します。

- 定義

Invoke Method: POST

Request Body Format: application/json

Request URL: https://<service domain>/api/feature/v2/getDbList

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
dbId	string	No	特徴量データベースの ID 番号。
dbName	string	No	特徴量データベースの名前。
current	number	Yes	現在取得するのページ番号
size	number	Yes	現在取得するページごとの結果数

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
dbId	string	特徴量データベースの ID 番号。
dbName	string	特徴量データベースの名前。
dbType	string	特徴量データベースの種類。

maxSize	string	特微量データベース最大レコード数。
currentSize	number	特微量データベースの現在レコード数。
normalizeSrc	string	正規化された特微量データの入力値
normalizeDst	string	正規化された特微量データの出力値
creatTs	string	データベースが作成された日時。
lastModTs	string	データベースが最後に変更された日時。

レスポンスデータサンプル:

```
{
  "data": [
    {
      "dbId": "CFJ",
      "dbName": "CFJ",
      "dbType": "face",
      "maxSize": "2000",
      "currentSize": 6,
      "normalizeSrc": "-
1.0f,0.08425623178482056f,0.1538366377353668f,0.2071316689252853f,0.2537092864513397
f,0.2973994612693787f,0.3400076925754547f,1.0f",
      "normalizeDst":
"0.0f,0.4000000059604645f,0.5f,0.6000000238418579f,0.699999988079071f,0.800000011920
929f,0.8999999761581421f,1.0f",
      "creatTs": "2025-01-19 18:33:32",
      "lastModTs": "2025-01-19 18:33:32"
    }
  ],
  "size": 30,
  "total": 1,
  "code": "0000",
  "msg": "success"
}
```

3.11. 特徴量データベースに特徴量追加

- API の概要

画像内にある顔の特徴量を抽出し、データベースに追加します。ひとつの顔のみサポートします。画像に複数の顔がある場合、最も大きな顔が対象になります。

- 定義

Invoke Method: **POST**

Request Body Format: **multipart/form-data**

Request URL: **https://<service domain>/api/feature/v2/insertByImage**

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
file	file	No	バイナリファイル形式の画像をアップロードしてください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
data	string	No	Base64 エンコードされたバイナリ画像データを入力してください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
dbId	string	Yes	特徴量データベースの ID 番号
rotate	boolean	No	true: 回転する false: 回転しない パラメーターのデフォルト値は false です。
externalUuid	string	No	外部システム連携用キー。ビジネス関係の作成に使用され、128 バイトをサポートします。
remark	string	No	拡張メッセージ。ユーザー定義のメッセージ用です。Json 文字列をサポートし、2KB まで対応しています。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。

data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。
------	---------------	------------------------------

レスポンス詳細:

Parameter	Type	Description
featureId	string	特徴量 ID 番号。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "featureId": "69e98924-0c82-4da1-bc5e-7c32870a63e5"
  }
}
```

3.12. 特徴量データベースから特徴量取得

- API の概要

特徴量 ID と特徴量データベース ID を使って特徴量情報を取得します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/feature/v2/get

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
featureId	string	Yes	特徴量の ID 番号。
dbId	string	Yes	特徴量データベースの ID 番号。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
featureId	string	特徴量データ ID 番号
dbId	string	特徴量データベースの ID 番号
feature	string	特徴量データ
dbName	string	特徴量データベースの名前

externalUuid	string	外部システム連携用キー。ビジネス関係の作成に使用され、128バイトをサポートします。
remark	string	拡張メッセージ。ユーザー定義のメッセージ用です。Json 文字列をサポートし、2KB まで対応しています。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "featureId": "69e98924-0c82-4da1-bc5e-7c32870a63e5",
    "dbId": "TEST13",
    "feature": "eyJ.....ifQ==",
    "dbName": "TEST13",
    "externalUuid": "9257723f-e122-42f6-a2e6-76d8194dfca3",
    "remark": ""
  }
}
```

3.13. 特徴量更新

- API の概要

画像を使って特徴量データベース内の特徴量を更新します。

- 定義

Invoke Method: **POST**

Request Body Format: **multipart/form-data**

Request URL: **https://<service domain>/api/feature/v2/updateByImage**

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
featureId	string	Yes	特徴量データ ID 番号。
dbId	string	Yes	特徴量データベースの ID 番号。
file	file	No	バイナリファイル形式の画像をアップロードしてください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
data	string	No	Base64 エンコードされたバイナリ画像データを入力してください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
rotate	boolean	No	true: 回転する false: 回転しない パラメーターのデフォルト値は false です。
externalUuid	string	No	外部システム連携用キー。ビジネス関係の作成に使用され、128 バイトをサポートします。
remark	string	No	拡張メッセージ。ユーザー定義のメッセージ用です。Json 文字列をサポートし、2KB まで対応しています。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。

msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
featureId	string	顔の特徴量の ID 番号。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": {
    "featureId": "69e98924-0c82-4da1-bc5e-7c32870a63e5"
  }
}
```

3.14. 特徴量削除

- API の概要

特徴量 ID とデータベース ID を指定して顔の特徴量を削除します。

- 定義

Invoke Method: POST

Request Body Format: multipart/form-data

Request URL: https://<service domain>/api/feature/v2/delete

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
featureIds	string	Yes	特徴量の ID 番号リスト。
dbId	string	Yes	特徴量データベースの ID 番号。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
featureId	string	特徴量の ID 番号。
status	enum	特徴量の操作のステータス 0: 削除成功 1: 削除失敗

レスポンスデータサンプル:

```
{
```

```
"code": "0000",  
"msg": "success",  
"data": [  
  {  
    "featureId": "69e98924-0c82-4da1-bc5e-7c32870a63e5",  
    "status": "0"  
  }  
]  
}
```

3.15. 画像で特徴量検索（1対N認証）

- API の概要

顔画像を使用して特徴量データベースを検索します。最も類似度が高い単一または複数の特徴量とスコアの比較を返します。

- 定義

Invoke Method: **POST**

Request Body Format: **multipart/form-data**

Request URL: **https://<service domain>/api/feature/v2/searchByImage**

- リクエストパラメーター

Headers:

Parameter	Type	Required	Description
x-app-id	string	Yes	認証用のアプリケーション ID。
x-app-key	string	Yes	認証用のアプリケーションキー。

Body:

Parameter	Type	Required	Description
file	file	No	バイナリファイル形式の画像をアップロードしてください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
data	string	No	Base64 エンコードされたバイナリ画像データを入力してください。ファイルまたはデータのいずれかを入力する必要があります。両方が入力された場合、ファイルが優先されます。
dbId	string	Yes	Feature database ID number.
rotate	boolean	No	true: 回転する false: 回転しない パラメーターのデフォルト値は false です。
featureType	enum	No	特徴量データベース内の特徴タイプ。 注: デフォルト値は "Face" です。
count	number	No	最も類似性の高い特徴量の数を返します。デフォルト値は 1 です。
threshold	number	No	比較の閾値は、(0,1] の範囲内で小数を入力することが可能です。

- レスポンス

Parameter	Type	Description
code	string	通常のレスポンスでは「0000」を返します。他の詳細については、レスポンスコード一覧をご参照ください。

msg	string	通常のレスポンスでは「success」を返します。他の詳細については、レスポンスコード一覧をご参照ください。
data	array[Object]	レスポンスの詳細データです。以下「レスポンス詳細」参照。

レスポンス詳細:

Parameter	Type	Description
score	string	比較結果のスコア。
featureId	string	顔特徴の ID 番号。
dbId	enum	特徴量データベースの ID 番号。
dbName	boolean	特徴量データベースの名前。
externalUuid	array[Object]	外部システム連携用プライマリキー、ビジネスリレーションシップの作成に使用。
remark	string	拡張メッセージ。ユーザー定義のメッセージ用です。
type	object	特徴量データベース内の特徴タイプ。

レスポンスデータサンプル:

```
{
  "code": "0000",
  "msg": "success",
  "data": [
    {
      "score": "0.6191986203193665",
      "featureId": "e4ac9958-175d-46b4-a50a-bbf2ad282921",
      "dbId": "TEST13",
      "dbName": "TEST13",
      "externalUuid": "9257723f-e122-42f6-a2e6-76d8194dfca3",
      "remark": "",
      "type": "Face"
    }
  ]
}
```

4. レスポンスコード

4.1. 顔検出のエラーコード

Parameter	Error message	Description
62010301	Call address absent	呼び出し先のアドレスが存在しません。
62010201	**parameter missing	*のパラメーターは必須です (** は特定のパラメーターを示します)。
62010202	Invalid parameter type	無効なパラメータータイプ。
62010203	**Parameter parsing exception	* のパラメーター解析例外 (** は特定のパラメーターを示します)。
62010601	Invalid image type	無効な画像ファイル種類
62010602	Invalid image size	無効な画像ファイルサイズ
62010603	Image parsing error	画像処理エラー
62990101	service error	内部サービスエラー：このようなエラーが繰り返し発生する場合は、ログファイルをコピーし、速やかにナブラワークス株式会社まで連絡してください。

4.2. 顔の品質と属性のエラーコード

Parameter	Error message	Description
64010301	Call address absent	呼び出し先のアドレスが存在しません。
64010201	**parameter missing	*のパラメーターは必須です (** は特定のパラメーターを示します)。
64010202	Invalid parameter type	無効なパラメータータイプ。
64010203	**Parameter parsing exception	* のパラメーター解析例外 (** は特定のパラメーターを示します)。
64010601	Invalid image type	無効な画像ファイル種類
64010602	Invalid image size	無効な画像ファイルサイズ
64010603	Image parsing error	画像処理エラー
64990101	service error	内部サービスエラー：このようなエラーが繰り返し発生する場合は、ログファイルをコピーし、速やかにナブラワークス株式会社まで連絡してください。

4.3. 1 対 N 顔認識のエラーコード

Parameter	Error message	Description
63010301	Call address absent	呼び出し先のアドレスが存在しません。
63010201	**parameter missing	*のパラメーターは必須です (** は特定のパラメーターを示します)
63010202	Invalid parameter type	無効なパラメータータイプ。
63010203	**Parameter parsing exception	*のパラメーター解析例外 (** は特定のパラメーターを示します)
63010601	Invalid image type	無効な画像ファイル種類
63010602	Invalid image size	無効な画像ファイルサイズ
63010603	Image parsing error	画像処理エラー
63040507	** Query failed	*のクエリ失敗 (** は特徴または画像を示します)。
63040503	** rule failed	**検索ルールが満たされていません (DB タイプやその他の検索条件が不正です)
63040501	**existed	*すでに存在します (** は特徴または画像を示します)
63040502	** is null	*存在しません (** は特徴または画像を示します)
63040504	**Add failed	*追加に失敗しました (** は特徴または画像を示します)
63040505	**Update failed	*更新に失敗しました (** は特徴または画像を示します)
63040506	**Delete failed	*削除に失敗しました (** は特徴または画像を示します)
63040508	Inventory quantity exceeds limit	データベースの数が制限を超えています。
63040509	Storage capacity exceeds limit	データベースの特徴の数が制限を超えています。
63990101	service error	内部サービスエラー：このようなエラーが繰り返し発生する場合は、ログファイルをコピーし、速やかにナブラワークス株式会社まで連絡してください。
63020202	static-search error : &&	静的データベース API の例外リクエスト。例：データベースが存在しない、データベースに特徴がない、など。