**Como faço a verificação da biometria de uma pessoa no BioPass ID ABIS?**

 Para realizar a operação de verificação da biometria de um indivíduo é necessário utilizar o método **Verify**. Nesta operação é feita a comparação biométrica 1:1, utilizando-se dos dados biométricos de um indivíduo com alguém já registrado na base. Sendo necessário a biometria facial ou digital, o retorno será uma mensagem confirmando se as biometrias pertencem ou não à mesma pessoa.

Para utilizar essa operação o cliente deve realizar a chamada através da API. A documentação completa está disponível no site<https://docs.biopassid.com/>.

**Campos de entrada:**

* **CandidateId:** Identificador único do candidato conforme registrado na base.
* **Face:** Informações da biometria facial, sendo elas:
	+ **ImageFileName:** Tipo de imagem a ser passada (“WSQ”, “PNG”, etc.).
	+ **ImageBase64:** Imagem da biometria facial codificada em uma string base64.
	+ **HorzResolution:** Tamanho da imagem em PPI na horizontal.
	+ **VertResolution:** Tamanho da imagem em PPI na vertical.
* **Fingers:** Lista de objetos contendo cada biometria digital que será usada, bem como informações adicionais quanto a elas:
	+ **FingerName:** Nome do dedo, ou, caso não haja um nome, usar o mesmo valor de “Position”.
	+ **Position:** Posição do dedo na qual a biometria foi extraída, podendo ser: Desconhecido, PolegarDireito, IndicadorDireito, MedioDireito, AnelarDireito, MinimoDireito, PolegarEsquerdo, IndicadorEsquerdo, MedioEsquerdo, AnelarEsquerdo, MinimoEsquerdo.
	+ **CollectedType:** Tipo de captura de digitais, podendo ser: “Plain” para captura pousada, “Rolled” para captura rolada e “Latent” para captura latente.
	+ **ImageFileName:** Tipo de imagem a ser passada (“WSQ”, “PNG”, etc.).
	+ **ImageBase64:** Imagem da biometria digital codificada em uma string base64.
	+ **HorzResolution:** Tamanho da imagem em PPI na horizontal.
	+ **VertResolution:** Tamanho da imagem em PPI na vertical.
	+ **Anomalydescription:** Informações adicionais para anomalias na impressão digital, caso haja, informando o tipo e o período (permanente, temporário) da anomalia.

Exemplo de como ficaria o corpo da requisição:

{

 "CandidateId": "example-candidate-id",

 "Face": {

 "ImageFileName": "string",

 "ImageBase64": "string",

 "HorzResolution": 0,

 "VertResolution": 0

 },

 "Fingers": {

 "Finger": [

 {

 "FingerName": "string",

 "Position": "string",

 "CollectedType": "string",

 "ImageFileName": "string",

 "ImageBase64": "string",

 "HorzResolution": 0,

 "VertResolution": 0,

 "AnomalyDescription": {

 "AnomalyType": "string",

 "AnomalyTimeFrame": "string"

 }

 }

 ]

 },

 "DoMatchIfFound": true

}

O header da requisição, por sua vez, é onde deve ser inserido o tipo do conteúdo. Abaixo, é apresentado um exemplo de como o header deve ser representado:

{

 'Content-Type': 'application/json'

}

Por fim, na requisição, devem ser passados como parâmetros a especificação de que é um método POST, a URL ‘https://abis.api.biopassid.com/verify/post’, além do header e do corpo descrito anteriormente. Abaixo, apresentamos um exemplo de código em Python que realiza essa requisição. Exemplos em outras linguagens de programação podem ser encontrados em nossa documentação.

import http.client

import json

conn = http.client.HTTPSConnection("abis.api.biopassid.com")

payload = json.dumps({

 "CandidateId": "example-candidate-id",

 "Face": {

 "ImageFileName": "string",

 "ImageBase64": "string",

 "HorzResolution": 0,

 "VertResolution": 0

 },

 "Fingers": {

 "Finger": [

 {

 "FingerName": "string",

 "Position": "string",

 "CollectedType": "string",

 "ImageFileName": "string",

 "ImageBase64": "string",

 "HorzResolution": 0,

 "VertResolution": 0,

 "AnomalyDescription": {

 "AnomalyType": "string",

 "AnomalyTimeFrame": "string"

 }

 }

 ]

 },

 "DoMatchIfFound": True

})

headers = {

 'Content-Type': 'application/json'

}

conn.request("POST", "/verify/post", payload, headers)

res = conn.getresponse()

data = res.read()

print(data.decode("utf-8"))

 A resposta recebida em caso de uma operação bem-sucedida incluirá diversos campos de resposta, como descritos abaixo.

**Campos de resposta:**

* **Success:** Valor booleano indicando o sucesso da operação.
* **Message:** Mensagem adicional referente ao sucesso da operação.
* **Status:** Status da operação (Ex.: Processing, Done, Error).
* **MatchingSubject:** Objeto contendo informações do indivíduo o qual ocorreu o match biométrico.
	+ **ReferenceCandidateId:** Identificador interno do candidato passado na chamada da operação.
	+ **IdentifyCandidateId:** Identificador interno do indivíduo o qual ocorreu o match biométrico.
	+ **Score:** Valor numérico representando o grau de confiabilidade do match biométrico.
	+ **ScoreFace:** Valor numérico representando o grau de confiabilidade do match da biometria facial.
	+ **ScoreFinger:** Valor numérico representando o grau de confiabilidade do match da biometria digital.
* **Faces:** Objeto contendo informações sobre a biometria de face enviada, e o Score de confiabilidade do match.
* **Fingers:** Objeto contendo informações sobre as biometrias de digitais enviadas, e o Score de confiabilidade do match.

A mensagem de resposta para o exemplo fornecido pode ser verificada abaixo:

{

 "Success": true,

 "Message": "string",

 "Status": "string",

 "MatchingSubject": {

 "ReferenceCandidateId": "example-candidate-id",

 "IdentifyCandidateId": "example-candidate-id",

 "ScoreN": 0,

 "Score": 0,

 "ScoreFaceN": 0,

 "ScoreFace": 0,

 "ScoreFingerN": 0,

 "ScoreFinger": 0,

 "Threshold": 0,

 "FacesMatched": [

 {

 "SondaTemplateFaceId": "example-candidate-id",

 "ReferenceTemplateFaceId": "example-candidate-id",

 "FaceSourceMatchedId": 0,

 "MatchedIndex": 0,

 "Score": 0,

 "ScoreN": 0

 }

 ],

 "FingerMatched": [

 {

 "FingerSourcePosition": 0,

 "FingerMatchedId": 0,

 "SondaTemplateFingerId": "example-candidate-id",

 "ReferenceTemplateFingerId": "example-candidate-id",

 "FingerSourceMatchedId": 0,

 "ScoreN": 0,

 "Score": 0,

 "CenterX": 0,

 "CenterY": 0,

 "BiometricType": 0,

 "TranslationX": 0,

 "TranslationY": 0

 }

 ]

 },

 "Faces": {

 "Faces": [

 {

 "Id": "string",

 "CreateDateUtc": "2024-07-02T18:11:33.367Z",

 "PathImageBlob": "string",

 "ImageFileName": "string",

 "ImageBase64": "string",

 "ImagePngBase64": "string",

 "ImageThumbnail120PngBase64": "string"

 }

 ],

 "FacesMatch": [

 {

 "InicialImageGuid": "example-candidate-id",

 "ReceivedImageFileName": "string",

 "Score": 0,

 "Threshold": 0

 }

 ]

 },

 "Fingers": {

 "Fingers": [

 {

 "Id": "string",

 "CreateDateUtc": "2024-07-02T18:11:33.367Z",

 "PathImageBlob": "string",

 "ImageFileName": "string",

 "ImageBase64": "string",

 "ImagePngBase64": "string",

 "ImageThumbnail120PngBase64": "string",

 "Nfiq1": 0,

 "Nfiq2": 0,

 "Nfiq21": 0,

 "Position": 0,

 "PositionName": "string",

 "CollectedType": "string",

 "AnomalyTimeFrame": "string",

 "AnomalyType": 0,

 "AnomalyTypeName": "string"

 }

 ],

 "FingersMatch": [

 {

 "InicialImageGuid": "example-candidate-id",

 "ReceivedImageFileName": "string",

 "Score": 0,

 "Threshold": 0,

 "StatusName": "string",

 "ErrorMessage": "string"

 }

 ]

 }

}

Caso seja retornado um erro, verifique se o conteúdo do corpo e da requisição está corretamente preenchido. Caso receba uma mensagem com o código 200 (OK), significa que sua requisição foi solicitada e realizada com sucesso e você pode prosseguir com as próximas requisições.