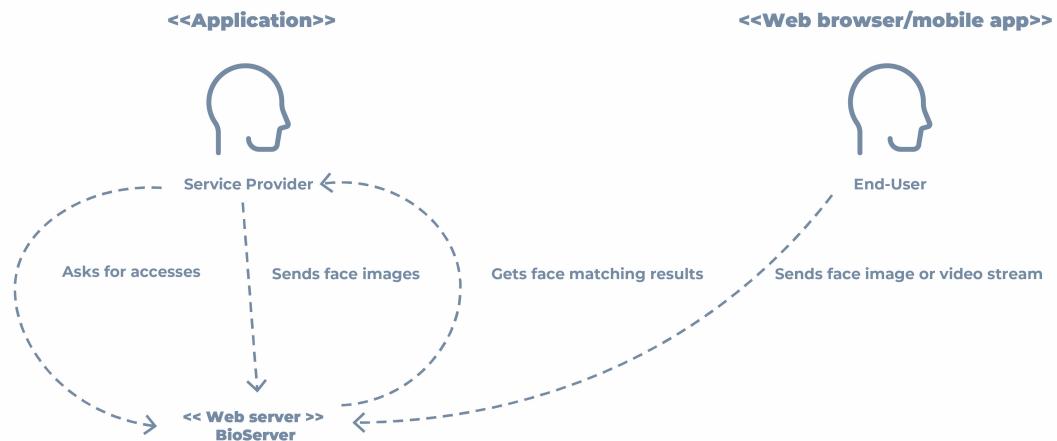


Biometric Services

Overview

Biometric Services is intended to be used by service providers to build identity proofing services for their end-users.

- **Biometric Services** exposes a simple REST API to detect and recognize faces from still images.
- **WebCapture SDK (Face Autocapture & Liveness)** provides face and liveness detection from video streams.



Service Provider

A service provider is an entity that develops apps and use cases on top of the BioServer.

The service provider:

- Owns a client ID and a secret that gives access to Biometric Services for either the service provider itself or a third party
- Can eventually be billed according to the use of Biometric Services (see: *Billing System*)

End-user

The end-user is a person whose face images are sent to Biometric Services in the context of use cases that are implemented by the service provider.

Typical Use Cases

The most common use cases are described below.

Reference Image Verification

This use case flow allows the verification and storage of a user's face image.

Process Steps

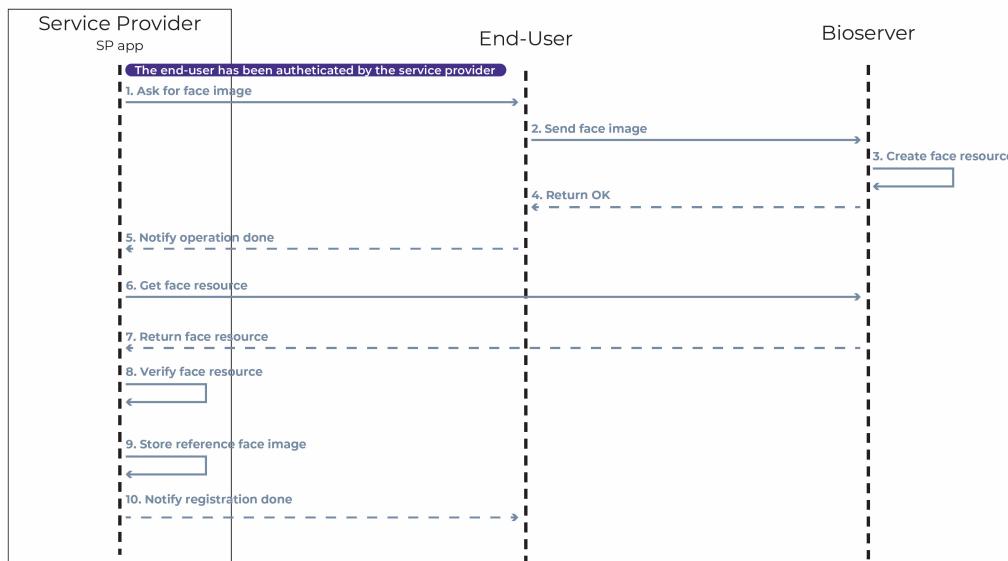
The primary process steps are:

Note: The known end-user has already been authenticated by the Service Provider.

1. The Service Provider (SP) app asks the known end-user for an image of their face,
2. The face image is sent to the BioServer.
3. The BioServer creates the face resource.
4. An "OK" notification is returned to the end-user.
5. The SP app is notified that the operation is done.
6. The SP app gets the face resource from the BioServer.
7. The BioServer returns the face resource to the SP app.
8. The face image is verified by Biometric Services and becomes a reference face image.
9. The back-end stores the user's reference image.
10. The SP app notifies the end-user that registration is complete.

UML Diagram of Workflow

A diagram of the reference image verification process is shown below.



One-to-One Verification

This use case allows one-to-one face image recognition. An end-user is asked for a face biometric authentication. Biometric Services checks it against a reference face (previously stored by the back-end).

Process Steps

The primary process steps are:

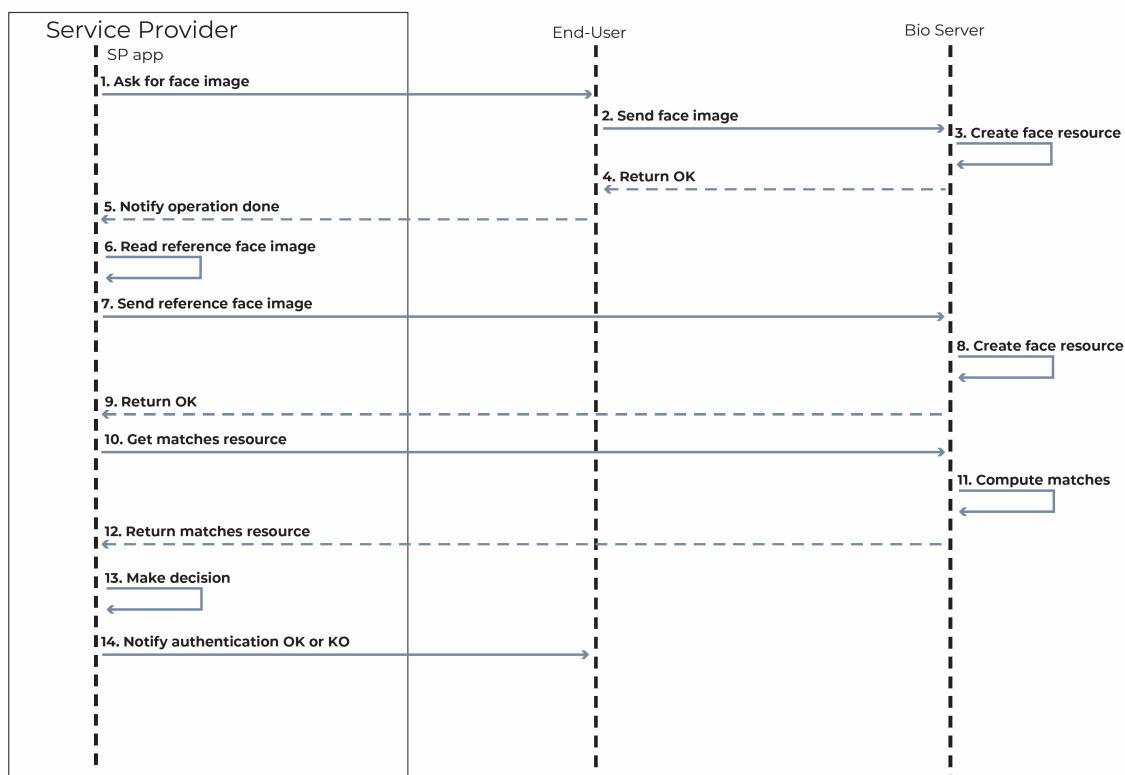
1. The Service Provider (SP) app asks the end-user for a face biometric authentication.
2. The end-user sends their face image to the BioServer.
3. The BioServer creates the face resource.
4. The face resource is returned to the end-user who "OKs" it.

5. The SP app is notified that the operation is complete.
6. The SP app reads the reference face image.
7. The SP app sends the reference face image to the BioServer.
8. Biometric Services checks if a reference face matches another image of the end-user's face, and creates the face resource.
9. "OK" is returned to the SP app.
10. The SP app returns the matches resource to the BioServer.
11. The BioServer computes the matches.
12. The matches resource is returned to the SP app.
13. The SP app makes a decision about the match.
14. The SP app notifies the end-user if the authentication is "OK" or not.

Note: Only authentication via the `APIkey` is required. The Biometric Services session is not used. Images and verification results are not stored in the session.

Process Workflow Diagram

A diagram of the one-to-one verification process is shown below.



One-to-Few Identification

This use-cases allows the service provider to find matches of a reference face among a small number of face images.

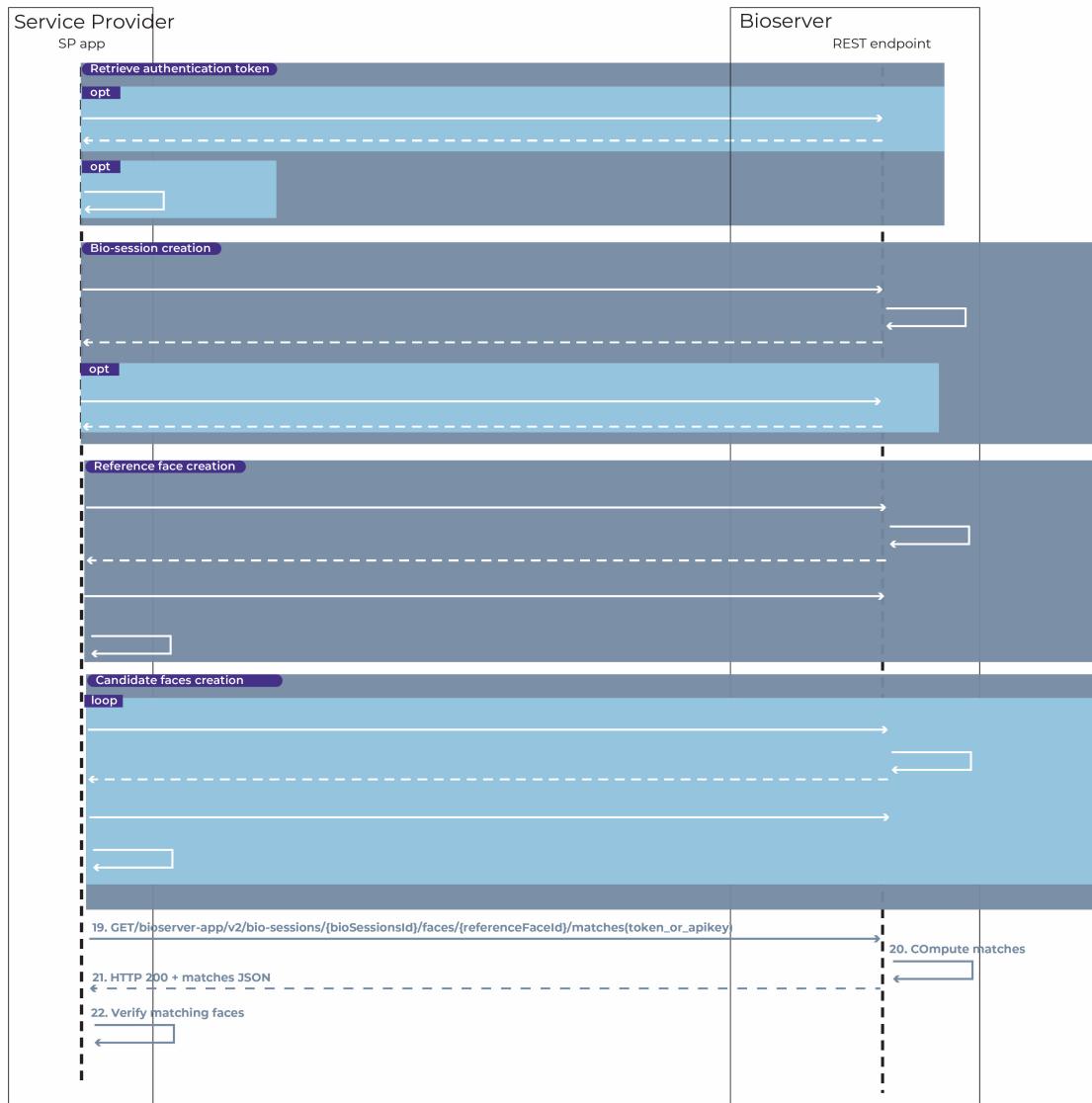
The primary process steps are:

1. The Service Provider (SP) app sends the reference face image of a known end-user to the BioServer.
2. The BioServer creates the face resource.
3. An "OK" notification is returned to the SP app.
4. The face image is sent to the BioServer.
5. The face resource is created.

6. An "OK" response is returned to the SP app.
7. The SP app gets the matches resource from the BioServer.
8. The BioServer computes matches.
9. The BioServer returns the matches resource to the SP app.
10. The SP app verifies that the faces match.

UML Workflow Diagram

A diagram of the one-to-few identification process is shown below.



API Diagrams

This section discusses provides API UML diagrams for biometric services workflows.

One-to-One Verification

This use-cases allows the service provider to match a known end-user's reference face to that specific end-user's face image.

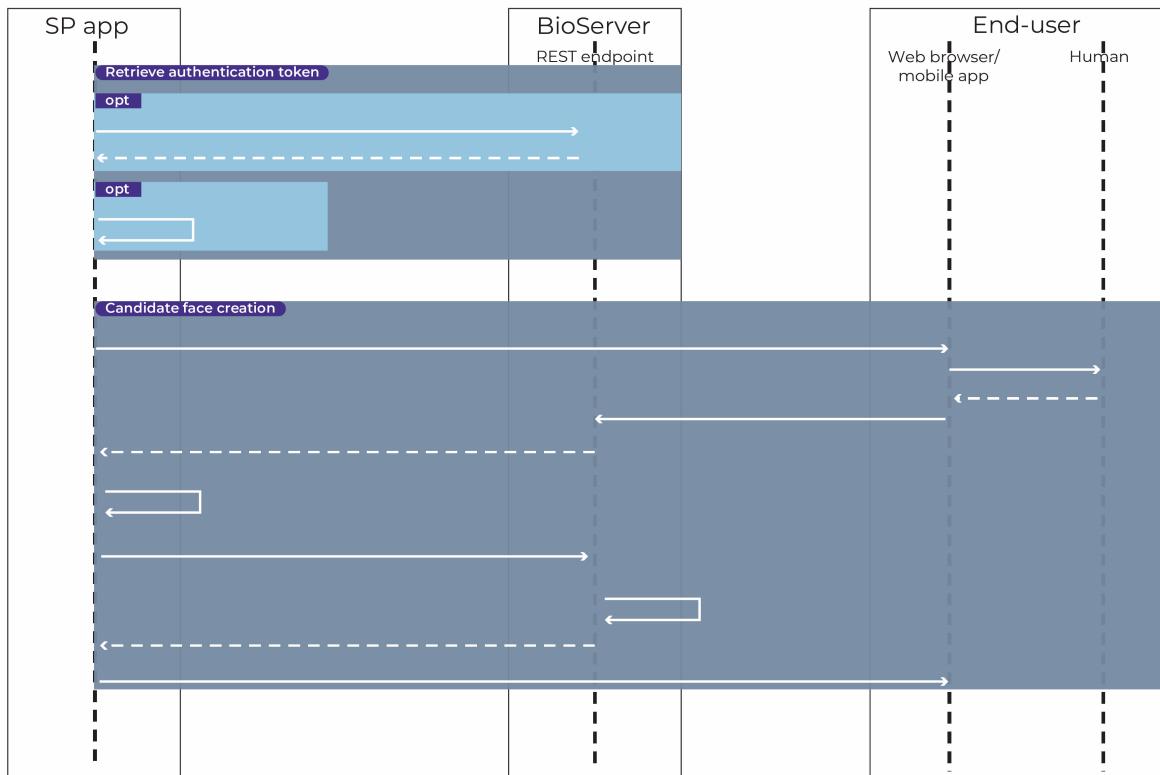
Web Services

The following web services are involved in this scenario:

- *Direct Create Face*: Returns the Face result in JSON format
- *Direct Match Image vs. Template*: Returns the Match result in JSON format

API Workflow Diagram

A diagram of the one-to-one verification process is shown below.



One-to-Few Identification

This use-cases allows the service provider to find matches of a reference face among a small number of face images:

- Biometric Services checks a face from a photo ("selfie") and matches a face extracted from an identity document.
- The result is signed by Biometric Services and can be added to the decision proof.

Note: All face images must be provided. No end-user database is involved.

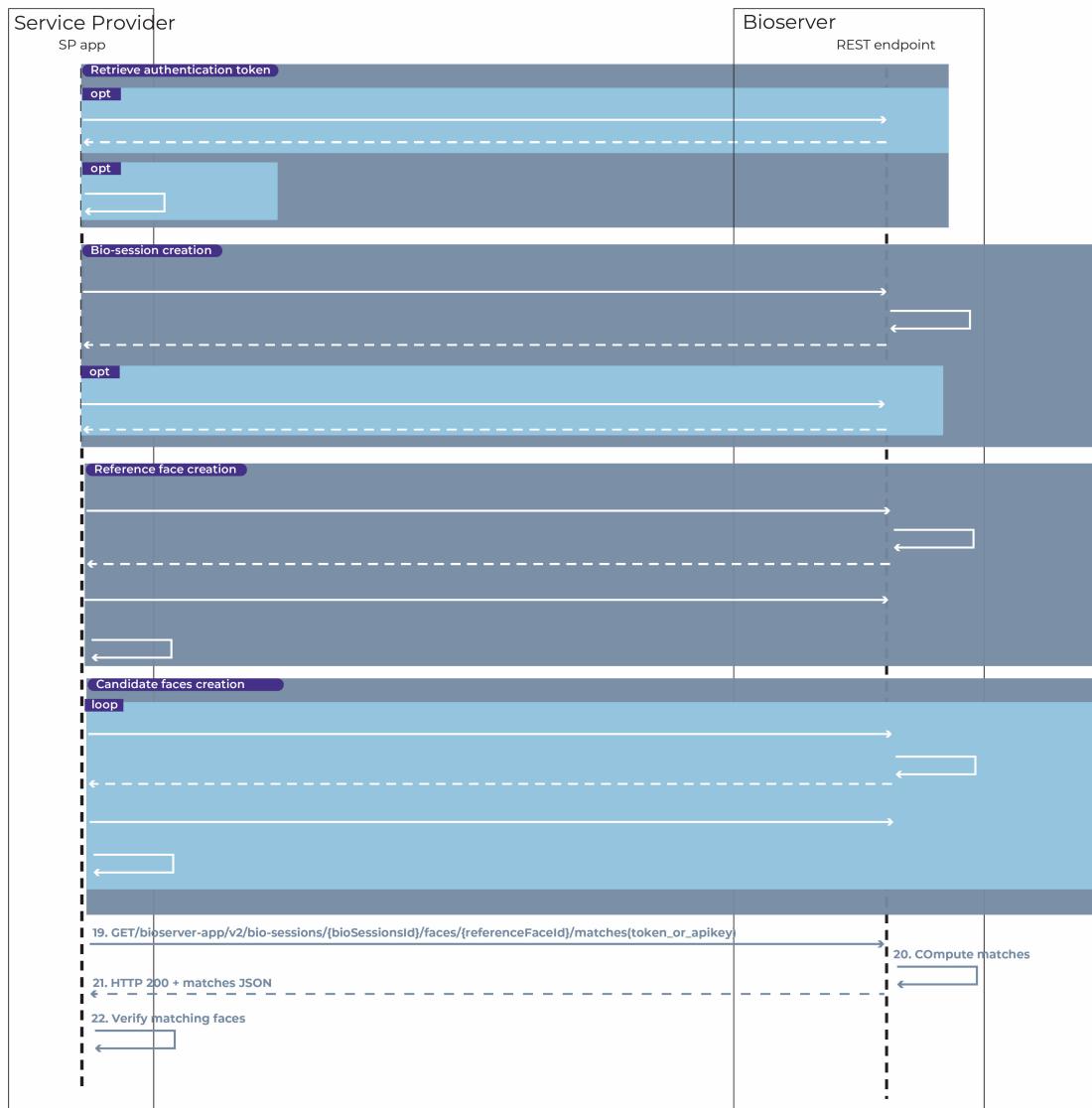
Web Services

The following web services are involved in this scenario:

- `CreateBioSession`: Returns a `SESSION_ID`
- `GetBioSession` (Optional): Retrieves `SESSION` information.
- `CreateFaceFromImage` (Multiple times): Encodes each face image and stores the template within distributed cache. Returns the `faceid` in response header(Location).
- `GetFace` : (Optional): Retrieves each face information.
- `Matches` : Retrieves a list of ordered matches (best scores come first) for a given face.

API Workflow Diagram

A diagram of the one-to-few verification process is shown below.



Replay Challenge

This feature allows you to replay data from smartphone Biometric Services that defines the challenge settings for the mobile device.

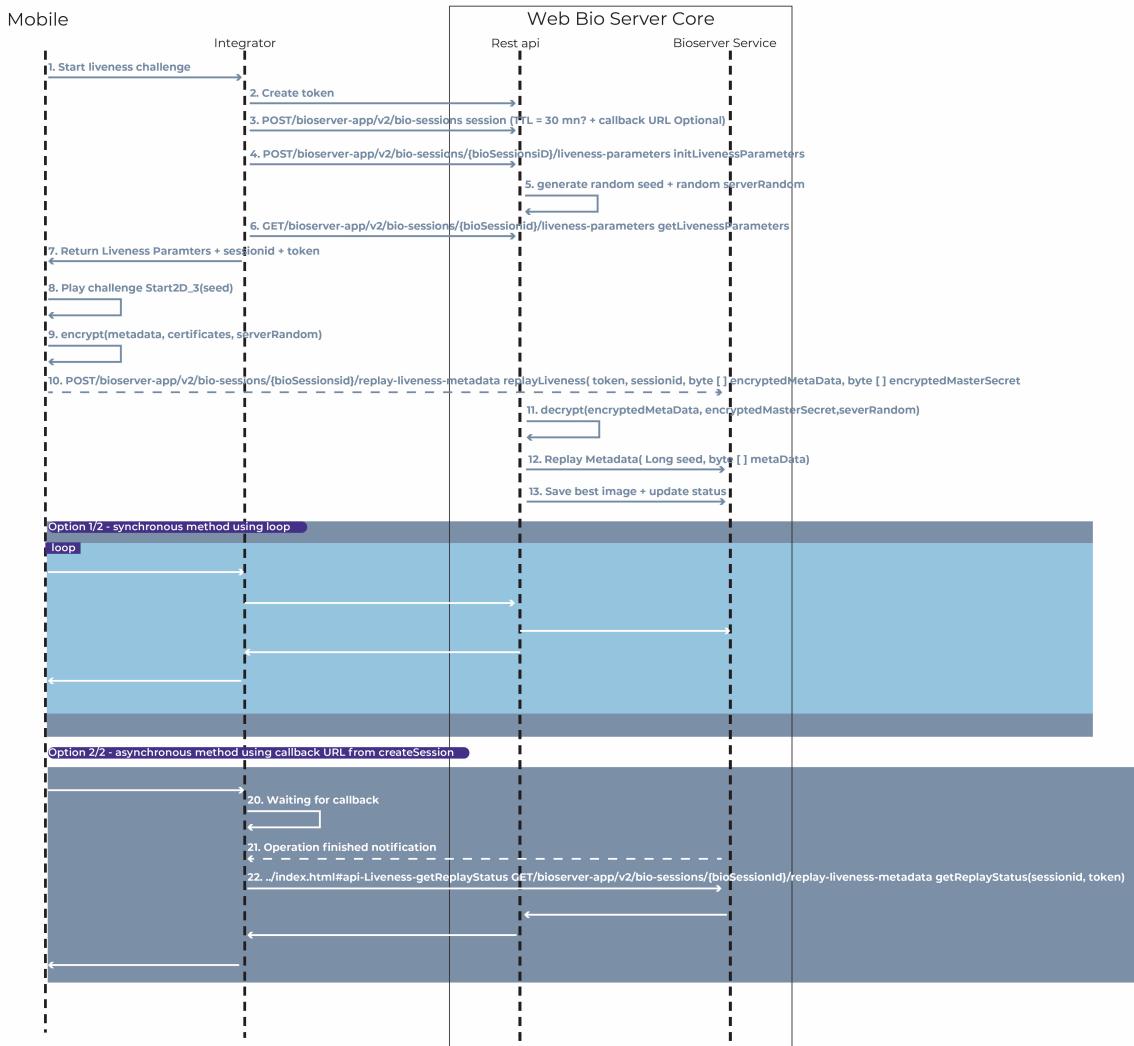
The mobile device uses these settings to:

- Execute the liveness challenge
- Send metadata to Biometric Services
- Replay the liveness challenge from the given metadata via `Faceflow`.

Web Services

The following web services are involved in this scenario:

- `CreateBioSession`: Returns `SESSION_ID`
- `GetBioSession` (Optional): Retrieves `SESSION` information
- `initLivenessParameters`: Posts liveness settings that will be used to run the challenge
- `getLivenessParameters` (Optional): Retrieves liveness settings that will be used to run the challenge on the mobile side



API Errors

Calls to the API may return errors. When this happens, an HTTP error code is returned along with an error object.

Error objects include the following:

- An ID attribute which is used to identify the error
 - An optional message attribute which gives more information about the error

Sample Error Object

The object containing the `id` error and the `message` describing the error are shown in the snippet:

```
{  
  "id": "a52e98b4-b052-4289-b001-b53773ab656b",  
  "message": "morpho.rt.Exception: An operation has failed with error code=8  
  (Buffer has an unknown format)"  
}
```

REST APIs

Overview

This section provides details about the REST APIs used for end-user identify proofing in Biometric Services. These REST APIs accept JSON for the request payload, and send responses to JSON.

Bio-Sessions APIs

This section describes the APIs used to create and retrieve bio-sessions.

A bio-session is a sequence of Biometric Services interactions related to the same end-user. A bio-session is explicitly created, but ends automatically after a configured number of seconds has elapsed.

CreateBioSession API

The `createBioSession` API creates a bio-session.

Method

A sample request to create the bio-session is shown in the snippet:

```
curl -X POST \
  https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions \
  -H 'Content-Type: application/json' \
  -H 'apikey: [APIKEY_VALUE]' \
  -d '[{
    "correlationId",
    "imageStorageEnabled",
    "ttlSeconds"
  }]'
```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields containing meta-data associated with the API request and response fields are shown below.

Field	Description
<code>apikey</code> (optional)	<code>apikey</code> value
<code>Content-Type</code>	Allowed values: <code>application/json</code>

Request Body Fields

The request body is the data that you send to the server when you request the API. It is used to create the resource information returned by the API, and contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>correlationId</code> (optional)	string	This specifies the custom identifier coming from the caller. The default value is <code>null</code> .
<code>imageStorageEnabled</code> (optional)	boolean	This specifies if images need to be stored for this bio-session; if this parameter is set to <code>true</code> , then the image used to create a face resource can be retrieved using <code>GetFaceImage</code> . The default value is <code>false</code> .
<code>ttlSeconds</code> (optional)	number	This specifies a time-to-live value in seconds after which this bio-session is automatically ended. If it is not specified by <code>CreateBioSession</code> , the <code>ttlSeconds</code> will be set to the configured default. An error will occur if <code>ttlSeconds</code> is set higher than the configured max.
<code>callbackURL</code> (optional)	string	This specifies the URL used to notify a service provider that liveness check/replay-metadata results are available.

Response Example

This response sample shows a sample response from the request example.

The requested parameters are returned as specified with the `-d` option shown in the snippet:

```
curl -X POST \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions \
-H 'Content-Type: application/json' \
-H 'apikey: [APIKEY_VALUE]' \
-d '[{
"correlationId": "891a6728-1ac4-11e7-93ae-92361f002671",
"imageStorageEnabled": false,
"ttlSeconds": 300
}]'
```

Success and Error Codes

Success Response 201

The `201` code response indicates that the bio-session was successfully created. It also contains the `Location` header string of the URI for the created face image, as shown below.

Field	Type	Description
<code>Location</code>	string	Header containing the URI of the bio-session created

An example success snippet is shown below:

```
HTTP/1.1 201 Created
get
Location: /v2/bio-sessions/0991cedc-9111-4b9d-9e4e-8d6eb4db488f
```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to create the bio-session

GetBioSession API

The `GetBioSession` API retrieves information about the bio-session.

Method

```
curl -X GET \
```

The `GET` and `curl` commands are used to request the bio-session header from the server.

Endpoint

```
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{bioSessionId} \
```

Request Example

The requested `biosessionID` field and `apikey` header value (specified with the `-H` option) are shown in the snippet:

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{bioSessionId} \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions

The `APIkey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Field

The HTTP header field containing meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code> (optional)	<code>APIkey</code> value

The header data is specified with the `-H` option as shown in the snippet:

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/capabilities \
-H 'Authorization: [APIKEY_VALUE]' \
```

URI Field

The URI field that provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved is shown below.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session resource to retrieve

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created, as shown in the fields below.

Field	Type	Description
<code>id</code>	string	The bio-session unique identifier generated by Biometric Services
<code>correlationId</code> (optional)	string	The custom identifier coming from the caller
<code>imageStorageEnabled</code>	boolean	If images will be stored for this bio-session
<code>created</code>	datetime	The date on which the bio-session is created
<code>expires</code>	datetime	The date after which the bio-session will expire and will be removed from the server
<code>signature</code> (optional)	string	The digital signature (JWS) of the response—authentication and integrity can be verified afterward using the Biometric Services public certificate

Response Example

This response sample returns the requested parameters with the bio-session information as shown in the snippet:

```
{  
  "id": "0991cedc-9111-4b9d-9e4e-8d6eb4db488f",  
  "correlationId": "891a6728-1ac4-11e7-93ae-92361f002671",  
  "imageStorageEnabled": false,  
  "created": "2017-05-18T10:14:39.587Z",  
  "expires": "2017-05-18T10:19:39.587Z",  
  "signature": "eyJhbGciOiJSUzI1NiJ9.eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.  
  }
```

Error Codes

The error response code indicates the specific error type and the error code issue. If an error occurs, one of the 4xx error codes will be generated as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to retrieve the bio-session
404	Unable to find a bio-session for the given identifier

Capabilities APIs

This section describes the `GetCapabilities` API that requests some capabilities of the bio-session, along with the version number and the supported algorithm.

GetCapabilities API

This API retrieves various capabilities of the bio-session, along with the algorithm used, the version used, and the supported algorithms.

Method

The `GET` and `curl` commands are used to make the HTTP request to the URL for the bio-session capabilities.

The header data is specified with the `-H` option as shown in the snippet:

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/capabilities \
-H 'Authorization: [APIKEY_VALUE]' \
```

Header Field

The HTTP header field containing meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code> (optional)	<code>APIkey</code> value

Permissions

The `APIkey` is the API key unique identifier used to authenticate requests and track and control API usage.

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>version</code>	string	The Biometric Services version.
<code>modes</code>	string	The available modes that can be used for face creation (see Create face).

Sample Code

The sample code response for the version and modes is shown in the snippet:

```
{
  "status": "ok",
  "currentMode": [
    "F5_1_VID60"
  ],
  "acceptedModes": [
    "F5_2_VID75",
    "F5_1_VID60",
    "F5_5_VID60"
  ],
  "version": "3.8.1"
}
```

Success Response 200

The `200` code will be returned with a description in the event of success, as shown below.

Name	Description
<code>200</code>	The instance is up.

Error Response 500

The `500` error code will be returned with an error description if there is an issue, as shown below.

Name	Description
<code>500</code>	The instance is not working properly.

Faces APIs

This section describes the APIs that handle face functions.

CreateFaceFromImage API

The `CreateFaceFromImage` function creates a face resource from a known user's image by processing the face image on the server.

Note: No more than one face can be detected per image.

Image Requirements

- There should be at least 60 pixels between the eyes.

- When an image is compressed using JPEG (.jpeg/.jpg), the image quality should be 80% or higher.

Method

The `POST` and `curl` commands are used to create the face image resource.

The header data is specified with the `-H` option, and the paths to the multi-form images are flagged with the `-F` option, as shown in the snippet:

```
curl -X POST \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{biosessionId}/faces \
-H 'Content-Type: multipart/mixed' \
-H 'apikey: [APIKEY_VALUE]' \
-F 'face=@[ABSOLUTE_LOCAL_PATH_TO_face]' \
-F 'image=@[ABSOLUTE_LOCAL_PATH_TO_image]'
```

Sample Code

A JSON image object for the API is shown in the snippet:

```
{
  "imageType" : "SELFIE",
  "friendlyName" : "Presidential portrait of Barack Obama",
  "imageRotationEnabled":false,
}
```

Permissions

The `APIkey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	The <code>APIkey</code> value
<code>Content-Type</code>	Allowed values: <code>multipart/mixed</code> , <code>multipart/form-data</code>

URI Field

The URI field provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session in which to create the face.

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>face</code>	object	The face JSON object
<code>mode</code> (optional)	string	The biometric algorithm used to create the face biometric template; allowed values: <code>F5_2_VID75</code> , <code>F5_1_VID0</code> , <code>F5_3_VID60</code>
<code>imageType</code>	string	The kind of image; allowed values: <code>ID_DOCUMENT</code> , <code>SELFIE</code> , <code>VIDEO_FRAME</code>
<code>friendlyName</code> (optional)	string	A friendly name for the face; default value: <code>null</code>
<code>imageRotationEnabled</code> (optional)	Boolean	This enables image rotation for coding. Note: Do not use if the image is not rotated. For optimal performance, it's recommended to use images that are already correctly oriented rather than to use this parameter to rotate them; default value: <code>null</code>
<code>image</code>	binary	The face image to process

Success Response 201

If the bio-session was successful, then the response will return code `201` and contain the `Location` header string of the URI for the created face image.

Field	Type	Description
<code>Location</code>	string	Header containing the URI of the created face

Sample Code

The `201 created` response indicates that the bio-session was successfully created. This response also returns the `Location` string of the URI for the created bio-session as shown in the snippet:

```
HTTP/1.1 201 Created
Location: /v2/bio-sessions/0991cedc-9111-4b9d-9e4e-8d6eb4db488f/faces/aa0bd6ca-1206-415b-af94-8d2c18aa9c70
```

Error Responses 4xx

If an error occurs, one of the `4xx` error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to create the face
413	The request was too big

CreateDirectFaceFromImage API

This function creates a face resource by processing a face image on the server. The face resource created is not associated with a session.

Note: No more than one face can be detected per image.

Image Requirements

- There should be at least 60 pixels between the eyes.
- When an image is compressed using JPEG (.jpeg/.jpg), the image quality should be 80% or higher.

Method

This method uses the `POST` command with `curl` to create the image resource on the BioServer.

The header data is specified with the `-H` option and the image paths are flagged with the `-F` option, as shown in the snippet:

```
curl -X POST \
https://[URL_MAIN_PART]/bioserver-app/v2/faces \
-H 'Content-Type: multipart/mixed' \
-H 'apikey: [APIKEY_VALUE]' \
-F 'face=@[ABSOLUTE_LOCAL_PATH_TO_FACE]' \
-F 'image=@[ABSOLUTE_LOCAL_PATH_TO_image]' \
-d '{"correlationId": "891a6728-1ac4-11e7-93ae-92361f002671"}'
```

```
{
  "imageType" : "SELFIE",
  "friendlyName" : "Presidential portrait of Barack Obama",
  "imageRotationEnabled":false
}
```

Permissions

The `APIkey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	APIkey value
<code>Content-Type</code>	Allowed values: <code>multipart/mixed</code> , <code>multipart/form-data</code>

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>face</code>	object	The JSON object for a reference face
<code>mode</code> (optional)	string	The biometric algorithm used to create the face biometric template (currently the algorithm used is the one in the server configuration file) Allowed values: <code>F5_2_VID75</code> , <code>F5_1_VID60</code> , <code>F5_3_VID60</code>
<code>imageType</code>	string	The kind of image. Allowed values: <code>ID_DOCUMENT</code> , <code>SELFIE</code> , <code>VIDEO_FRAME</code>
<code>friendlyName</code> (optional)	string	A friendly name for the face. Default value: <code>null</code>
<code>signatureDisabled</code> (optional)	string	Disable the signature of the result. Default value: <code>false</code>
<code>imageRotationEnabled</code> (optional)	Boolean	To enable image rotation for coding. <i>Do not use if the image is not rotated.</i> For optimal performance, it is recommended to use images that are already correctly oriented rather than to use this parameter to rotate them. Default value: <code>null</code>
<code>image</code>	binary	The face image to process.
<code>correlationId</code> (optional)	string	A custom identifier coming from the caller. Default value: <code>null</code>

Success Response Object 200

These fields are included in the JSON object (payload) when a success code of `200` is returned after the face image has been successfully created.

Field	Type	Description
<code>id</code>	string	The face unique identifier generated by BioServer.
<code>friendlyName</code> (optional)	string	A friendly name for the face
<code>digest</code> (optional)	string	The SHA-256 digest of the image file from which the face has been created for confidentiality and verification purposes
<code>mode</code>	string	The biometric algorithm used to create the face biometric template
<code>imageType</code>	string	The type of image
<code>quality</code> (optional)	number	The biometric template quality—a good quality template is a template with a quality superior to 100; if the quality is negative, then the face needs to be sent again
<code>landmarks</code> (optional)	object	Landmarks detected on the face
<code>eyes</code>	object	Eye detection information
<code>x1</code>	number	The x-coordinate of the first eye
<code>y1</code>	number	The y-coordinate of the first eye
<code>x2</code>	number	The x-coordinate of the second eye
<code>y2</code>	number	The y-coordinate of the second eye
<code>template</code>	string	The biometric template of the face
<code>templateversion</code>	number	The version of the biometric template; <i>newer versions of biometric algorithms don't support the previous version of biometric templates</i>
<code>imageRotationEnabled</code> (optional)	Boolean	Image rotation enabled; default value: <code>null</code>
<code>created</code>	datetime	The date on which the face is created

Sample Response Payload

The following JSON object (with payload) is returned when the face image is successfully created, as shown in the snippet:

```
{
  "id": "3d069016-e66e-498e-af32-b930ba1c5067",
  "friendlyName": "Presidential portrait of Barack Obama",
  "digest": "8b11d3b37fcdb71da6b26d1c601c262a37b4c0f39edfb51978cc46ace122862",
  "mode": "F5_3_VID60",
  "imageType": "SELFIE",
  "quality": 295,
```

```

    "landmarks": {"eyes": {
        "x1": 1220.5284,
        "y1": 1891.722,
        "x2": 1621.1726,
        "y2": 1873.5762
    }},
    "imageRotationEnabled": false,
    "template": "EQAAAJQDAAACAE09Uo_fRI...yJzdwIioij0ZAEAAQDAAC==",
    "templateversion": 113,
    "created": "2018-10-25T14:32:48.863Z"
}

```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to create the face
413	The request was too big

CreateFaceFromJson API

This function creates a face resource from a previous face JSON object by processing the face biometric template on the server.

Method

Note: This method is significantly faster than creating faces from images, but you must be careful as newer versions of biometric algorithms don't support the previous version of biometric templates.

This uses the `POST` and `curl` command to send data to the BioServer to create the face resource.

```

curl -X POST \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{biosessionId}/faces \
-H 'Content-Type: application/json' \
-H 'apikey: [APIKEY_VALUE]' \
-d '{
    "id": "ee9b1742-06f4-4e22-8630-3852d55f314a",
    "friendlyName": "Official portrait of First Lady Michelle Obama in the Green Room of the White House",
    "digest": "20b4839d1da77fba4ed185e5a4649d1c514b101f823299c928dc43356f68b4df",
    "mode": "F5_1_VID60",
    "imageType": "SELFIE",
    "quality": 306,
    "landmarks": {
        "eyes": {
            "x1": 518.71204,
            "y1": 444.68396,
            "x2": 518.71204,
            "y2": 444.68396
        }
    }
}'

```

```

    "x2": 655.408,
    "y2": 446.599
  },
  },
  "template": "EQAAAMwDAAACA&hellip;AAAAAAAAAYMIPAFAA",
  "templateVersion": 92,
  "created": "2017-11-13T13:28:24.098Z",
  "expires": "2017-11-13T13:33:24.108Z",
  "signature":
  "eyJhbGciOiJSUzI1NiJ9.eyJkNmlhbCI6ImFjY29tLmRseU5hbWUoroAE_oxDF_ZtH-E"
}

```

Permissions

The `APIkey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	<code>APIkey</code> value
<code>Content-Type</code>	Allowed values: <code>application/json</code>

URI Field

The URI field that provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved is shown below.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session in which to create the face.

Request Body Field

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>face</code>	object	A face JSON.

Success Response 201

If the bio-session was successfully created, then the response will return code `201` and contain the `Location` header string of the URI for the created face image.

Field	Type	Description
<code>Location</code>	string	Header containing the URI of the bio-session created

Sample Code

The `201 created` response indicates that the bio-session was successfully created. This response also returns the `Location` string of the URI for the created bio-session as shown in the snippet:

```
HTTP/1.1 201 Created
Location: /v2/bio-sessions/0991cedc-9111-4b9d-9e4e-8d6eb4db488f/faces/aa0bd6ca-
1206-415b-af94-8d2c18aa9c70
```

Error Responses 4xx

If an error occurs, one of the `4xx` error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to create the face

GetFace API

This function retrieves a face image.

Method

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-
sessions/{bioSessionId}/faces/{faceId} \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Field

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
apikey	APIkey value

URI Fields

The URI field and related field that provides the HTTP address for the `bioSessionID` string identifier and the bio-session face resource are shown below.

Field	Type	Description
<code>bioSessionId</code>	string	The identifier of the bio-session containing the face.
<code>faceId</code>	string	The identifier of the face resource to retrieve.

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>id</code>	string	The face unique identifier generated by BioServer.
<code>friendlyName</code> (optional)	string	A friendly name for the face.
<code>digest</code> (optional)	string	The SHA-256 digest of the image file from which the face has been created for confidentiality and verification purposes.
<code>mode</code>	string	The biometric algorithm used to create the face biometric template.
<code>imageType</code>	string	The kind of image.
<code>quality</code> (optional)	number	The biometric template quality. A good quality template is a template with a quality superior to 100. If the quality is negative, then the face needs to be sent again.
<code>landmarks</code> (optional)	object	Landmarks detected on the face.
<code>eyes</code>	object	Eye detection information.
<code>x1</code>	number	The x-coordinate of the first eye.
<code>y1</code>	number	The y-coordinate of the first eye.
<code>x2</code>	number	The x-coordinate of the second eye.
<code>y2</code>	number	The y-coordinate of the second eye.
<code>template</code>	string	The biometric template of the face.
<code>templateversion</code>	number	The version of the biometric template Note: Newer versions of biometric algorithms don't support the previous version of biometric templates.
<code>imageRotationEnabled</code> (optional)	Boolean	Image rotation enabled. Default value: <code>null</code>
<code>created</code>	datetime	The date on which the face is created.
<code>expires</code>	datetime	The date after which the face will expire and will be removed from the server.
<code>signature</code> (optional)	string	A digital signature (JWS) of the response. Authentication and integrity can be verified afterward using the Biometric Services public certificate.

Sample Response Object 200

A sample success object with payload is shown in the snippet:

```
{  
  "id": "aa0bd6ca-1206-415b-af94-8d2c18aa9c70",  
  "friendlyName": "Presidential portrait of Barack Obama",  
  "digest": "39bd0d9606a772b1e7076401f32f14bdde403b9608e789e0771b90fb79b664a4",  
  "mode": "F5_1_VID60",  
  "imageType": "SELFIE",  
  "quality": 295,  
  "landmarks": {  
    "eyes": {  
      "x1": 1191.4584,  
      "y1": 582.79565,  
      "x2": 1477.8955,  
      "y2": 580.3324  
    }  
  },  
  "template": "EQAAAMwDAAACA&hellip;AAAAAAAAAYMIPAFAA",  
  "templateVersion": 92,  
  "created": "2017-05-18T12:39:48.987Z",  
  "expires": "2017-05-18T12:40:48.988Z",  
  "signature":  
    "eyJhbGciOiJSUzI1NiJ9.eyJ0KogIMFhMGJkNmNhL&hellip;ogIClbumRseU5hbWUoroAE_oxDF_ZtH-E"  
}
```

Error Responses 4xx

If an error occurs, one of the *4xx* error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to retrieve the face
404	Unable to find a bio-session or a face for the given identifier

GetFaceImage API

Note: This function is only possible if image storage has been enabled for the bio-session.

This function retrieves the image that has been used to create a face resource as shown in the snippet.

```
curl -X GET \  
  https://[URL_MAIN_PART]/bioserver-app/v2/bio-  
sessions/{bioSessionId}/faces/{faceId}/image?compression=true \  
  -H 'apikey: [APIKEY_VALUE]' \
```

Permissions

The `APIkey` is the API key unique identifier that is used to authenticate requests and track and control API usage.

Header Field

The HTTP header field that contains meta-data associated with the API request and response fields is shown below.

Name	Description
<code>apikey</code>	The <code>APIkey</code> value

URI Fields

The URI field and related fields that provide the HTTP address for the `biosessionID` string identifier and the bio-session face resource are shown below.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session containing the face.
<code>faceID</code>	string	The identifier of the face resource for which the image needs to be retrieved. ⁰⁹
<code>compression</code> (optional)	Boolean	To enable image jpeg compression. Default value: <code>false</code>

Success Response 200

The success code `200` will generate when the face image is successfully retrieved as shown below.

Name	Description
<code>200</code>	The image has been successfully retrieved.

Sample Code

The 200 success response code is shown in the snippet:

```
HTTP/1.1 200 OK
Content-Type: image/jpeg
(image)
```

Error Response 204

The `204` error code will generate when storage is not enabled for the bio-session as shown below.

Name	Description
<code>204</code>	Storage is not enabled for the bio-session.

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to retrieve the face image
404	Unable to find a bio-session or a face for the given identifier

GetFaces API

This function retrieves the list of faces that were created in a bio-session.

Method

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{bioSessionId}/faces \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions Field

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	<code>apikey</code> value

URI

Field	Type	Description
<code>bioSessionId</code>	string	The identifier of the bio-session.

Response (example):

```
[  
{  
  "id": "3fd2bf77-244a-4563-a6ba-70abdc14b2ee",  
  "friendlyName": "Barack Obama's Columbia University Student ID",  
  "digest": "94d1b6ff2acf368c3e0ccaebe1d8e447ed1ccd7b596dc5cac3c13a4822b256c6",  
  "mode": "F5_1_VID60",  
  "imageType": "ID_DOCUMENT",  
  "quality": 186,  
  "landmarks": {
```

```

"eyes": {
  "x1": 141.83296,
  "y1": 217.47075,
  "x2": 241.09653,
  "y2": 216.0568
},
},
"template": "EQAAAMwDAAACA&hellip;AAAAAAAAAYMIPAFAA",
"templateVersion": 92,
"created": "2017-05-18T12:43:05.373Z",
"expires": "2017-05-18T12:44:08.132Z",
"signature":
"eyJhbGcioiJSUzI1NiJ9.ew0KICAiawQiIDog&hellip;JPKaepToVlY4_InM6HQOm9JzFBMnzajI"
},
{
"id": "aa0bd6ca-1206-415b-af94-8d2c18aa9c70",
"friendlyName": "Presidential portrait of Barack Obama",
"digest": "39bd0d9606a772b1e7076401f32f14bdde403b9608e789e0771b90fb79b664a4",
"mode": "F5_1_VID60",
"imageType": "SELFIE",
"quality": 295,
"landmarks": {
"eyes": {
"x1": 1191.4584,
"y1": 582.79565,
"x2": 1477.8955,
"y2": 580.3324
}
},
"template": "EQAAAMwDAAACA&hellip;AAAAAAAAAYMIPAFAA",
"templateVersion": 92,
"created": "2017-05-18T12:43:06.945Z",
"expires": "2017-05-18T12:44:08.132Z",
"signature":
"eyJhbGcioiJSUzI1NiJ9.ew0KogImFhMGJkNmNhL&hellip;ogIC1bmRseU5hbWUoroAE_oxDF_ZtH-E"
}
]

```

Success 200

Name	Description
200	The faces have been successfully retrieved.

Error 4xx

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to retrieve the face list
404	Unable to find a bio-session for the given identifier

DeleteFace API

This function deletes a face image.

```
curl -X DELETE \
  https://[URL_MAIN_PART]/bioserver-app/v2/bio-
sessions/{biosessionId}/faces/{faceId} \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions Field

`apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Field

The HTTP header field that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	The <code>apikey</code> value

URI

Field	Type	Description
<code>biosessionId</code>	string	The identifier of the bio-session
<code>/</code>	string	The identifier of the face resource to delete

Success Response 204

The `204` code will be returned when the face image has been successfully deleted.

Name	Description
<code>204</code>	The face has been successfully deleted.

Sample Code

The sample response for the `204` code is shown in the snippet:

```
HTTP/1.1 204 No Content
```

Error Responses 4xx

If an error occurs, one of the `4xx` error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to delete the face list
404	Unable to find a bio-session or a face for the given identifier

DeleteFaces API

This function deletes the entire collection of faces in the current bio-session.

Method

```
curl -X DELETE \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{biosessionId}/faces \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Field

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	The <code>apikey</code> value

URI Field

The URI field that provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved is shown below.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session.

Success Response 204

The `204` code will be returned when the face image has been successfully deleted.

Name	Description
<code>204</code>	The face has been successfully deleted.

Sample Code

The sample response for the 204 code is shown in the snippet:

```
HTTP/1.1 204 No Content
```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to delete the face list
404	Unable to find a bio-session for the given identifier

Matches APIs

This section discusses how to perform face image matching.

GetMatches API

This function retrieves a list of ordered matches (best scores come first) for a given face.

The reference face is compared to all the faces created in the bio-session. The result of each comparison is called a "match". Each match is composed of the reference face, a candidate face, a matching score, and a false acceptance rate.

False Acceptance Rate

The threshold you want to avoid using is driven by the expected FAR (False Acceptance Rate).

FAR	Matching threshold
0.0001%	4500
0.001%	4000
0.01%	3500
0.1%	3000
1%	2500

Recommended Threshold

- *Matching Document/Selfie*: The recommended threshold is 2500 or 3000.
- *Matching Selfie/Selfie*: The recommended threshold is 3000, 3500, or higher depending on the use case.

Method

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-
sessions/{bioSessionId}/faces/{referenceFaceId}/matches \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header

The HTTP header field that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	<code>apikey</code> value

URI

Field	Type	Description
<code>bioSessionId</code>	string	The identifier of the bio-session containing the faces.
<code>referenceFaceId</code>	string	The identifier of the reference face.

Success 200

Field	Type	Description
<code>reference</code>	object	The reference face JSON.
<code>candidate</code>	object	A candidate face JSON.
<code>score</code>	number	The matching score.
<code>falseAcceptanceRate</code>	number	The false acceptance rate, or FAR: measure of the likelihood that the Biometric Services will incorrectly return a match when the faces do not actually belong to the same person. For instance, "100" means there is no chance the two faces belong to the same person, "0.000000000028650475" means there is almost no chance Biometric Services can be wrong.
<code>correlationId</code> (optional)	string	A custom identifier coming from the caller and currently associated with the bio-session.
<code>created</code>	datetime	The date on which the match has been created.
<code>expires</code>	datetime	The date after which the match will expire and will be removed from the server.
<code>signature</code> (optional)	string	A digital signature (JWS) of the response. Authentication and integrity can be verified afterward using the Biometric Services public certificate.

Response (example):

```
[
{
  "reference": {
    "id": "aa0bd6ca-1206-415b-af94-8d2c18aa9c70",
    "friendlyName": "Presidential portrait of Barack Obama",
    "digest": "39bd0d9606a772b1e7076401f32f14bdde403b9608e789e0771b90fb79b664a4",
    "mode": "F5_1_VID60",
    "imageType": "SELFIE",
    "quality": 295,
    "landmarks": {
      "eyes": {
        "x1": 1191.4584,
        "y1": 582.79565,
        "x2": 1477.8955,
        "y2": 580.3324
      }
    }
  },
  "candidate": {
    "id": "6e1741f1-3715-416a-bfc6-4fc381d228a3",
    "friendlyName": "Barack Obama's Columbia University Student ID",
    "digest": "94d1b6ff2acf368c3e0ccaebe1d8e447ed1ccd7b596dc5cac3c13a4822b256c6",
    "mode": "F5_1_VID60",
    "imageType": "ID_DOCUMENT",
  }
}
```

```

"quality": 186,
"landmarks": {
  "eyes": {
    "x1": 141.83296,
    "y1": 217.47075,
    "x2": 241.09653,
    "y2": 216.0568
  }
},
"score": 7771.43408203125,
"falseAcceptanceRate": 0.00000000028650475616752694,
"correlationId": "891a6728-1ac4-11e7-93ae-92361f002671",
"created": "2017-05-18T12:41:09.58Z",
"expires": "2017-05-18T12:42:00.844Z",
"signature": "eyJhbGciOiJSUzI1NiJ9.eyJ0KICAicm&hellip;ONCiAgICB9DQHSQfU7Q"
}
]

```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to retrieve the matches
404	Unable to find a bio-session or a face for the given identifier

Images Match API

This function serves a direct image match by returning a single matched object (see Get images matches) computed by two images: a reference image and a candidate image.

False Acceptance Rate

The threshold you want to avoid using is driven by the expected FAR (False Acceptance Rate).

FAR	Matching threshold
0.0001%	4500
0.001%	4000
0.01%	3500
0.1%	3000
1%	2500

Thresholds

The recommended thresholds are the following:

- *Matching Document/Selfie*: 2500 or 3000
- *Matching Selfie/Selfie*: 3000, 3500, or higher depending on the use case

Method

```
curl -X POST \
https://[URL_MAIN_PART]/bioserver-app/v2/match/images \
-H 'Content-Type: multipart/form-data' \
-H 'apikey: [APIKEY_VALUE]' \
-F 'referenceImage=@[ABSOLUTE_LOCAL_PATH_TO_referenceImage]' \
-F 'candidateImage=@[ABSOLUTE_LOCAL_PATH_TO_candidateImage]' \
-F 'referenceImageType=@[ABSOLUTE_LOCAL_PATH_TO_referenceImageType]' \
-F 'candidateImageType=@[ABSOLUTE_LOCAL_PATH_TO_candidateImageType]'
```

Permissions:

APIkey - (the API key unique identifier) - API key used to authenticate requests and track and control API usage.

Header

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
apikey	The APIkey value
Content-Type	Allowed values: multipart/mixed, multipart/form-data

Body

Field	Type	Description
referenceImage	binary	A reference face image to process.
candidateImage	binary	A candidate face image to process.
referenceImageType	String	The kind of image for the reference face. Allowed values: ID_DOCUMENT, SELFIE, VIDEO_FRAME
candidateImageType	String	The kind of image for the candidate face. Allowed values: ID_DOCUMENT, SELFIE, VIDEO_FRAME
correlationId (optional)	string	A custom identifier coming from the caller. Default value: null
imageRotationEnabled (optional)	boolean	Enables image rotation for coding. <i>Do not use if the image is not rotated.</i> For best performance, use images that are already correctly oriented instead of using this parameter to rotate incorrectly oriented images. Default value: null

Response (example):

```
{  
    "reference": {  
        "id": "31214b58-9edc-425b-9436-60dcd94fbc3a",  
        "mode": "F5_1_VID60",  
        "imageType": "SELFIE",  
        "quality": 306,  
        "landmarks": {  
            "eyes": {  
                "x1": 518.71204,  
                "y1": 444.68396,  
                "x2": 655.408,  
                "y2": 446.599  
            }  
        }  
    },  
    "candidate": {  
        "id": "7d94bbfd-b779-44a3-a8fa-dfc20324fb2b",  
        "mode": "F5_1_VID60",  
        "imageType": "SELFIE",  
        "quality": 306,  
        "landmarks": {  
            "eyes": {  
                "x1": 518.71204,  
                "y1": 444.68396,  
                "x2": 655.408,  
                "y2": 446.599  
            }  
        }  
    },  
    "score": 10269.438178710938,  
    "falseAcceptanceRate": 0.0000000000000001,  
    "correlationId": "c5a0aec2-1262-41b6-82bd-e9d60da92ef2",  
    "created": "2017-11-14T14:31:42.228Z",  
    "signature":  
        "eyJhbGciOiJSUzI1NiJ9.eyJ0eXAiOiJKV1QiLCJleHBvcnQIjoiMTYwMjEwNjIwOTk4LjIwMDAiLCJpYXQiOjE1NjUyNjUwNjAsImV4cCI6MTU2NTI2NjUwNjAsInR5cGUiOiJFQUNPVCIsImF1ZGUiOnsibmFtZSI6ImF1ZGUiLCJ0eXAiOiJFQUNPVCJ9"  
}
```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to create the face
413	The request was too big

Match API

This function returns a direct match by returning a single match object (see [GetMatches](#)) that was computed from a reference face (i.e., a previous face JSON) and a candidate's face image.

False Acceptance Rate

The threshold you should avoid using is driven by the expected FAR (False Acceptance Rate) as shown in the table.

FAR	Matching threshold
0.0001%	4500
0.001%	4000
0.01%	3500
0.1%	3000
1%	2500

Thresholds

The following thresholds are recommended:

- *Matching Document/Selfie*: The recommended threshold is 2500 or 3000.
- *Matching Selfie/Selfie*: The recommended threshold is 3000, 3500, or higher depending on the use case.

Method

This command uses `POST` and `curl` to....

```
curl -X POST \
https://[URL_MAIN_PART]/bioserver-app/v2/match \
-H 'Content-Type: multipart/form-data' \
-H 'apikey: [APIKEY_VALUE]' \
-F 'candidateImage=@[ABSOLUTE_LOCAL_PATH_TO_candidateImage]' \
-F 'candidateImageType=@[ABSOLUTE_LOCAL_PATH_TO_candidateImageType]' \
-d '[
  {
    "id": "ee9b1742-06f4-4e22-8630-3852d55f314a",
    "friendlyName": "Official portrait of First Lady Michelle Obama in the Green
Room of the White House",
    "digest": "20b4839d1da77fba4ed185e5a4649d1c514b101f823299c928dc43356f68b4df",
    "mode": "F5_1_VID60",
    "imageType": "SELFIE",
    "quality": 306,
    "landmarks": {
      "eyes": {
        "x1": 518.71204,
        "y1": 444.68396,
        "x2": 655.408,
        "y2": 446.599
      }
    },
    "template": "EQAAAMwDAAACA&hellip;AAAAAAAAAAyMIPAFAA"
  }
]
```

```

    "templateVersion": 92,
    "created": "2017-11-13T13:28:24.098Z",
    "expires": "2017-11-13T13:33:24.108Z",
    "signature":
    "eyJhbGciOiJSUzI1NiJ9.ewOKogImFhMGJkNmNhL&hellip;ogIClbumRseU5hbWUoroAE_oxDF_ZtH-
E"
}
]

```

Permissions

The `APIkey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	<code>APIkey</code> value
<code>Content-Type</code>	Allowed values: <code>multipart/mixed</code> , <code>multipart/form-data</code>

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>candidateImage</code>	binary	A candidate face image to process.
<code>candidateImageType</code>	String	The kind of image for the candidate face. Allowed values: <code>ID_DOCUMENT</code> , <code>SELFIE</code> , <code>VIDEO_FRAME</code>
<code>referenceFace</code>	object	A reference face JSON.
<code>imageRotationEnabled</code> (optional)	boolean	To enable image rotation for coding. <i>Do not use if the image is not rotated.</i> For best performance, use images that are already correctly oriented instead of using this parameter to rotate incorrectly oriented images. Default value: <code>null</code>
<code>correlationId</code> (optional)	string	A custom identifier coming from the caller. Default value: <code>null</code>

Response (example):

```
{
  "reference": {
    "id": "31214b58-9edc-425b-9436-60dcd94fbc3a",
    "friendlyName": "Official portrait of First Lady Michelle Obama in the Green Room of the White House",
    "mode": "F5_1_VID60",
    "imageType": "SELFIE"
}
```

```
        },
        "candidate": {
            "id": "7d94bbfd-b779-44a3-a8fa-dfc20324fb2b",
            "mode": "F5_1_VID60",
            "imageType": "SELFIE",
            "quality": 306,
            "imageRotationEnabled": false
            "landmarks": {
                "eyes": {
                    "x1": 518.71204,
                    "y1": 444.68396,
                    "x2": 655.408,
                    "y2": 446.599
                }
            }
        },
        "score": 10269.438178710938,
        "falseAcceptanceRate": 0.0000000000000001,
        "correlationId": "c5a0aec2-1262-41b6-82bd-e9d60da92ef2",
        "created": "2017-11-14T14:31:42.228Z",
        "signature": "eyJhbGciOiJSUzI1NiJ9.eyJ0KICAiicmv&hellip;jQVj1G8qNqhOTbg8cWMQ1k_5F-GQTnco"
    }
}
```

Error 4xx

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Missing permissions to create the face
413	The request was too big

Monitoring APIs

This section describes API monitoring functions.

GetStatus API

This function checks to see if an instance of Biometric Services is up and working properly.

Method

This uses the `GET` and `curl` command to request the resource header.

```
curl -X GET \
  https://[URL_MAIN_PART]/bioserver-app/v2/monitor \
```

Permissions

None

Request Body Fields

The API request body fields containing the body parameter fields are shown below. These response fields return the version and modes.

Field	Type	Description
version	string	The Biometric Services version.
modes	string	The available modes that can be used for face creation.
status	string	"ok" if the instance of Biometric Services is working properly.

Sample Response

The response contains the status, modes, and version number as shown in the snippet:

```
{  
    "status": "ok",  
    "modes": [  
        "F5_2_VID75",  
        "F5_1_VID60"  
    ],  
    "version": "3.8.1"  
}
```

Success 200

The 200 code will be returned with a description in the event of success, as shown below.

Name	Description
200	The instance is up.

Error 500

Name	Description
500	The instance is not working properly.

Replay Metadata APIs - Deprecated

This section discusses replaying metadata APIs.

initLivenessParameters - Deprecated

Note: This API has been DEPRECATED. Use instead: `Replay Liveness/initLivenessParameters`.

This posts the liveness parameter(s) that will be used to run the challenge.

```
curl -X POST \
```

```

https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{bioSessionId}/liveness-
parameters \
-H 'Content-Type: application/json' \
-H 'apikey: [APIKEY_VALUE]' \
-d '[{
    "type": "CR2D",
    "timeout":300000,
    "securityLevel": "FACEFLOW_SECURITY_MEDIUM",
    "purpose": "FACEFLOW_PURPOSE_ENROLL",
    "nbChallenge": 3,
    "cr2dMode": "FACEFLOW_CR2DMODE_PATH",
    "useAccurateMatch": true,
    "matchThreshold": 2500,
    "enableLighting": true,
    "illuminationMode": "FACEFLOW_ILLUMINATION_MODE_ON"
}]'

```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	<code>apikey</code> value
<code>Content-Type</code>	Allowed values: <code>application/json</code>

URI

Field	Type	Description
<code>bioSessionId</code>	string	The identifier of the bio-session containing the <code>livenessParameter</code> .

Body

Field	Type	Description
<code>livenessParameter</code>	object	A liveness parameter JSON.
<code>type</code>	string	Liveness type. Allowed values: <code>SLAM</code> , <code>CR2D</code>
<code>timeout</code>	number	Timeout before the end of acquisition. (in ms).
<code>securityLevel</code>	string	Specify the level of security. Higher level makes a smaller target. Allowed values: <code>FACEFLOW_SECURITY_VERY_LOW</code> , <code>FACEFLOW_SECURITY_LOW</code> , <code>FACEFLOW_SECURITY_MEDIUM</code> , <code>FACEFLOW_SECURITY_HIGH</code> , <code>FACEFLOW_SECURITY_VERY_HIGH</code>
<code>purpose</code>	string	Purpose of the sequence. Allowed values: <code>FACEFLOW_PURPOSE_TRACK</code> , <code>FACEFLOW_PURPOSE_ENROLL</code>
<code>nbChallenge</code>	number	Number of challenges to ask. Must be one or greater.
<code>cr2dMode</code>	string	CR2D mode to apply for this sequence. Allowed values: <code>FACEFLOW_CR2DMODE_RANDOM</code> , <code>FACEFLOW_CR2DMODE_PATH</code> , <code>FACEFLOW_CR2DMODE_TARGET_FIXED</code> , <code>FACEFLOW_CR2DMODE_TRAINING</code>
<code>useAccurateMatch</code> (optional)	boolean	If true, then apply the Biometric Services matching algorithm on the given liveness metadata images.
<code>matchThreshold</code> (optional)	integer	The matching threshold used when matching the liveness metadata images.
<code>enableLighting</code> (optional)	boolean	If true, then a screen light is emitted during liveness detection.
<code>illuminationMode</code> (optional)	string	Enable/disable liveness illumination check. Default value = <code>ON</code> . Allowed values: <code>FACEFLOW_ILLUMINATION_MODE_ON</code> , <code>FACEFLOW_ILLUMINATION_MODE_OFF</code>

Success Response 204

The `204` code will be returned when the face image has been successfully deleted.

Name	Description
<code>204</code>	The face has been successfully deleted.

Sample Code

The sample response for the `204` code is shown in the snippet:

```
HTTP/1.1 204 No content
```

Error 4xx

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Forbidden
404	Unable to find a bio-session for the given identifier

getLivenessParameters - Deprecated

Note: This API has been DEPRECATED. Use instead: `Replay_liveness/initLiveness`. See Response.

This retrieves the liveness parameter(s) to run the challenge.

```
curl -X GET \
  https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{biosessionId}/liveness-
parameters \
  -H 'apikey: [APIKEY_VALUE]' \
```

Permissions

`APIkey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Field

The HTTP header field that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	APIkey value

URI

The URI field provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session that contains <code>livenessParameter</code> .

Success Response Object 200

These fields are included in the JSON object (payload) when a success code of `200` is returned after the face image has been successfully created.

Field	Type	Description
<code>livenessParameter</code>	object	A liveness parameter JSON created by the init liveness parameters.
<code>seed</code>	number	Seed generated by Biometric Services used to run the challenge.
<code>serverRandom</code>	number	Random 32-byte string generated by Biometric Services used to check the <code>encryptedMetaData</code> .
<code>certificates</code>	String[]	List of Base64-encoded certificates in DER format. To be used to encrypt metadata.
<code>type</code>	string	Liveness type. Allowed values: <code>SLAM</code> , <code>CR2D</code> .
<code>timeout</code>	number	Timeout before the end of acquisition. (in ms)
<code>securityLevel</code>	string	Specify the level of security. Higher level makes a smaller target. Allowed values: <code>FACEFLOW_SECURITY VERY_LOW</code> , <code>FACEFLOW_SECURITY_LOW</code> , <code>FACEFLOW_SECURITY_MEDIUM</code> , <code>FACEFLOW_SECURITY_HIGH</code> , <code>FACEFLOW_SECURITY VERY_HIGH</code> .
<code>purpose</code>	string	Purpose of the sequence. Allowed values: <code>FACEFLOW_PURPOSE_TRACK</code> , <code>FACEFLOW_PURPOSE_ENROLL</code>
<code>nbChallenge</code>	number	Number of challenges to ask; there must be at least one.
<code>cr2dMode</code>	string	CR2D mode to apply for this sequence. Allowed values: <code>FACEFLOW_CR2DMODE_RANDOM</code> , <code>FACEFLOW_CR2DMODE_PATH</code> , <code>FACEFLOW_CR2DMODE_TARGET_FIXED</code> , <code>FACEFLOW_CR2DMODE_TRAINING</code> .
<code>useAccurateMatch</code> (optional)	boolean	If true, then the Biometric Services matching algorithm will be applied on the given liveness metadata images.
<code>matchThreshold</code> (optional)	integer	The matching threshold used when matching the liveness metadata images.
<code>illuminationMode</code> (optional)	string	Liveness illumination check mode.
<code>signature</code>	string	A digital signature (JWS) of the response. Authentication and integrity can be verified afterward using the Biometric Services public certificate.

Sample Code

A sample object with payload will be returned when the 200 success code is generated, as shown in the snippet:

```
{  
  "seed": 1150867590,  
  "serverRandom": "3TnEeJgaP1KWRbvi1Fvf3AHWNJWwwIFT",  
  "type": "CR2D",  
  "timeout": 300000,  
  "securityLevel": "FACEFLOW_SECURITY_MEDIUM",  
  "purpose": "FACEFLOW_PURPOSE_ENROLL",  
  "nbChallenge": 3,  
  "cr2dMode": "FACEFLOW_CR2DMODE_PATH",  
  "useAccurateMatch": true,  
  "illuminationMode": "FACEFLOW_ILLUMINATION_MODE_ON",  
  "matchThreshold": 2500,  
  "certificate":  
    ["eyJhbGciOiJSUzI1NiJ9.eyJ0KogIMhMGJkNmNhogiClbmRseU5hbWUoroAE_oxDF_ZtH-E"],  
  "signature":  
    "eyJhbGciOiJSUzI1NiJ9.eyJ0KogIMhMGJkNmNhL&hellip;ogiClbmRseU5hbWUoroAE_oxDF_ZtH-E"  
}
```

Error Responses 4xx

If an error occurs, one of the *4xx* error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Forbidden
404	Unable to find a bio-session for the given identifier

Replay Liveness APIs

This section discusses APIs for replaying liveness.

initLivenessParameters

This function posts the liveness parameter(s) that will be used to run the challenge.

```

curl -X POST \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{bioSessionId}/init-
liveness-parameters \
-H 'Content-Type: application/json' \
-H 'apikey: [APIKEY_VALUE]' \
-d '[{
  "type": "HIGH",
  "timeout":300000,
  "securityLevel":"MEDIUM",
  "nbChallenge":3
}]'

```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	The <code>apikey</code> value
<code>Content-Type</code>	Allowed values: <code>application/json</code>

URI

Field	Type	Description
<code>bioSessionId</code>	string	The identifier of the bio-session containing the <code>livenessParameter</code> .

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>livenessParameter</code>	object	A liveness parameter JSON.
<code>type</code>	string	Liveness type. Allowed values: <code>MEDIUM</code> , <code>HIGH</code>
<code>timeout</code>	number	Timeout before the end of acquisition. (in ms).
<code>securityLevel</code>	string	Specify the level of security. Higher level makes a smaller target. Allowed values: <code>LOW</code> , <code>MEDIUM</code> , <code>HIGH</code> , <code>VERY_HIGH</code> , <code>VERY_HIGH2</code> , <code>VERY_HIGH3</code> , <code>VERY_HIGH4</code> , <code>VERY_HIGH5</code>
<code>nbChallenge</code>	number	Number of challenges to ask. Must be one or greater.
<code>useAccurateMatch</code> (optional)	boolean	Deprecated. Do not use it.
<code>matchThreshold</code> (optional)	integer	Deprecated. Do not use it.

Sample Response

```
{
  "seed": 1150867590,
  "serverRandom": "3TnEeJgaP1KWRbvi1Fvf3AHWNJWwwIFT",
  "type": "HIGH",
  "timeout": 300000,
  "securityLevel": "MEDIUM",
  "nbChallenge": 3,
  "certificate": [
    "eyJhbGciOiJSUzI1NiJ9.ew0KogImFhMGJkNmNhogiClbmRseU5hbWUoroAE_oxDF_ZtH-E"
  ],
  "signature": "eyJhbGciOiJSUzI1NiJ9.ew0KogImFhMGJkNmNhL&hellip;ogiClbmRseU5hbWUoroAE_oxDF_ZtH-E"
}
```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
<code>400</code>	Something is wrong with the request
<code>401</code>	Authentication is required
<code>403</code>	Forbidden
<code>404</code>	Unable to find a bio-session for the given identifier

getLivenessChallengeResult

This function retrieves the video liveness challenge result as shown in the snippet.

Method

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{bioSessionId}/liveness-
challenge-result/{livenessMode} \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Field

The HTTP header field that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	<code>apikey</code> value

URI Field

The URI field that provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved is shown below.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session that contain <code>livenessParameter</code> .
<code>livenessMode</code>	string	The liveness mode set by the integrator (to avoid any fraud).

Request Body Fields

The request body is used to create the resource information returned by the API. It contains the representation of the resource to be created as shown in the fields below.

Field	Type	Description
<code>livenessResult</code> (optional)	object	
<code>livenessStatus</code>	String	Status of liveness challenge result. Allowed values: <code>SUCCESS</code> , <code>FAILED</code> , <code>SPOOF</code> , <code>SPOOF_JS</code> , <code>ERROR</code> , <code>TIMEOUT</code>
<code>diagnostic</code> (optional)	String	Diagnostic in case of liveness failure.
<code>bestImageId</code> (optional)	String	The ID of the stored best-image in the session.
<code>fakewebcam</code> (optional)	boolean	Set to <code>true</code> if a fake webcam was used during the liveness process (input from 'saveLivenessChallengeResult' method).
<code>livenessMode</code>	String	The liveness mode used during face capture. Allowed values: <code>NO_LIVENESS</code> , <code>LIVENESS_MEDIUM</code> , <code>LIVENESS_HIGH</code> , <code>LIVENESS_HIGH_TRAINING</code> .
<code>signature</code> (optional)	String	A digital signature (JWS) of the response.

Success Sample Code

The sample success object with payload is shown in the snippet:

```
{  
  "livenessStatus": "SUCCESS",  
  "bestImageId": "5597f426-3863-4fa1-b4ff-76a957913f39",  
  "livenessMode": "LIVENESS_HIGH",  
  "numberOfChallenge": 2,  
  "fake webcam": false,  
  "signature":  
    "eyJhbGciOiJSUzI1NiJ9.eyJ0KogIMFhMGJkNmNhL&hellip;ogIClbtmRseU5hbWUoroAE_oxDF_ZtH-  
    E"  
}
```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Forbidden
404	Unable to find a bio-session for the given identifier

replayLivenessMetadata

This function will post replay liveness from the mobile challenge metadata.

Method

This function uses the `POST` and `curl` commands to create the resource on the server.

The header is flagged with `-H`, along with the `-F` flag for the input form data that contains the path to the encrypted data, as shown in the snippet:

```
curl -X POST \
  https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{biosessionId}/replay-
liveness-metadata \
  -H 'Content-Type: multipart/form-data' \
  -H 'apikey: [APIKEY_VALUE]' \
  -F 'encryptedMetaData=@[ABSOLUTE_LOCAL_PATH_TO_encryptedMetaData]' \
  -F 'encryptedMasterSecret=@[ABSOLUTE_LOCAL_PATH_TO_encryptedMasterSecret]' \
  -F 'livenessDeviceInfo=@[ABSOLUTE_LOCAL_PATH_TO_livenessDeviceInfo]'
```

The sample code for the device information is shown in the snippet:

```
{"deviceModel" : "Samsung GS7 SM-G930F", "osType" : "Android", "osVersion": "7.1"}
```

Permissions

The `apikey` is the the API key unique identifier) used to authenticate requests and track and control API usage.

Header Fields

The HTTP header fields that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	The <code>apikey</code> value
<code>Content-Type</code>	Allowed values: <code>multipart/mixed</code> , <code>multipart/form-data</code>

URI Field

The URI field that provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved is shown below.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session in which to create the face.

Request Body Fields

The API request body fields containing the body parameter fields are shown below.

Field	Type	Description
<code>encryptedMetaData</code>	binary	The encrypted challenge metadata binary. Must be different than <code>NULL</code> .
<code>encryptedMasterSecret</code>	binary	The client-side key encrypted with the Biometric Services public key provided with the liveness parameters. This key is used on the client side to encrypt the challenge metadata.
<code>livenessDeviceInfo</code>	object	Mobile information JSON.
<code>osType</code>	string	Mobile OS type (Android, iOS ...).
<code>osVersion</code>	string	Version of phone OS ex: 7.0.
<code>deviceModel</code>	string	The phone model. Example: Samsung Galaxy S7 Edge SM-G935F, iPhone 7 A1778 ...

Success Response 201

If the bio-session was successfully created, then the response will return code `201` and contain the `Location` header string of the URI for the created face image.

Field	Type	Description
<code>Location</code>	string	Header containing the URI of the bio-session created

Sample Code

The `201 created` response indicates that the bio-session was successfully created. This response also returns the `Location` string of the URI for the created bio-session as shown in the snippet:

```
HTTP/1.1 201 No Content
```

Error Responses 4xx

If an error occurs, one of the `4xx` error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
<code>400</code>	Something is wrong with the request
<code>401</code>	Authentication is required
<code>403</code>	Forbidden
<code>404</code>	Unable to find a bio-session for the given identifier
<code>413</code>	The request was too big

getReplayStatus

This function retrieves a challenge replay response.

Method

```
curl -X GET \
https://[URL_MAIN_PART]/bioserver-app/v2/bio-sessions/{bioSessionId}/replay-
liveness-metadata \
-H 'apikey: [APIKEY_VALUE]' \
```

Permissions

The `apikey` is the API key unique identifier used to authenticate requests and track and control API usage.

Header Field

The HTTP header field that contains meta-data associated with the API request and response fields is shown below.

Field	Description
<code>apikey</code>	The <code>apikey</code> value

URI Field

The URI field that provides the HTTP address where the `biosessionID` string identifier for the bio-session resource can be retrieved is shown below.

Field	Type	Description
<code>biosessionID</code>	string	The identifier of the bio-session resource to retrieve.

Success Response Object 200

These fields are included in the JSON object (payload) when a success code of `200` is returned after the face image has been successfully created.

Field	Type	Description
<code>ReplayLivenessResponse</code>	object	
<code>status</code>	string	Liveness challenge response. Allowed values: INITIALIZED, IN_PROGRESS, ERROR, SUCCESS, FAILED, SPOOF, TIMEOUT
<code>signature</code>	string	A digital signature (JWS) of the response with detached payload to avoid duplication of the image.
		Authentication and integrity can be verified afterward using the Biometric Services public certificate JWS = HEADER.PAYLOAD.SIGNATURE. The JWS payload has to be reconstructed from the trimmed response (without signature field).
<code>image</code>	string	Best image from replay liveness (Base64-encoded).

Sample Response Payload

The following JSON payload is returned when the face image is successfully created.

```
{
  "status": "SUCCESS",
  "image": "F401MTb6QMNy8VFDMYDJ0dBQZepKDeHbTTCXRGaL+gFOCAomRITeZeJoQG3+CU2YNT2JXVz
VgcZWD5LQNASqJA==",
  "signature":
  "eyJhbGciOiJSUzI1NiJ9..cioeyJhiJSmRseU5hbwogIClbmRseU5hbwuoroAE_oxDF_ZtH-E"
}
```

Error Responses 4xx

If an error occurs, one of the 4xx error codes will be generated. The error response code indicates the specific error type and the error code issue as shown below.

Name	Description
400	Something is wrong with the request
401	Authentication is required
403	Forbidden
404	Unable to find a bio-session for the given identifier

FAQ

How do I validate a signature from JSON responses ?

Signatures are JWT signatures.

As an example you can use an online tool to verify the signature from JSON response: <https://jwt.io/>

There are two inputs required:

- The content of the signature attribute
- The public key from Biometric service

The screenshot shows the jwt.io debugger interface. At the top, there's a navigation bar with links for Debugger, Libraries, Introduction, Ask, and Get a T-shirt!. To the right, it says "Crafted by Auth0". Below the navigation, there's a dropdown menu set to "RS256".

The main area has two sections: "Encoded" and "Decoded".

Encoded: This section contains a long string of encoded JWT tokens. A portion of the string is highlighted in pink: `eyJhbGciOiJSUzI1NiJ9.eyJpZCI6IjA2ZDM2YzI1LTF1ZGYtNDYzOS05Y2U2LTy4NjUxNmUxZTJ1ZS1sImNvcnJlbGF0aW9uSWQ10iJwb3N0bWFuLXR1c3RzIiwiwA1hZ2VTdG9yYWdlRW5hYmx1ZCI6dHJ1ZSw1Y3J1YXR1ZCI6IjIwMjEtMDEtMTRUMDK6MTc6MTEuMjY0WiIsImV4cGlyZXMi0iIyMDIxLTAxLTE0VDA50jQ3ojExLjI2NFoifQ.iNpdnaALnpSrRiHxXTDMKz-dN0S8x7UgmC7Sca3PqbTS6RBgG32DG5a1NPG5AH-Jq2YvD5ic5gpjCUafHnsgXJdWthnOr6PaoyiMT8I5SFvA-L22gcFMBfVrrqBhM8udKDsiSizyd_i6Rh23IcLCJsDbENSMKEDqCjFViwb_TQV_EiUXQRzk6vX-koJhonQyvf_p3oiEgy0GR05yoLx5MJVP-Xd8S94TDin-UPsG9XrqGWpRLsTt-1LKaA8avrz3Hgle50crL6_ZnDki$FG7rjCqAOqMKqi-YUBaKuscVCzVaZh1XpZZ-sd2RYjvx4m4Azt61aEvESbONzJgsVSeEkyGJM7uwxtc9wjx06UXMVPyWwt88wHt4ZFbwHLzeskDLVjYDLOhQwk0502Uxpbe9r04YVYKRJj1ZJHgb4TZ_d6iJFLNabIoTo_kKa2v-kgfw6Z9bhVXgDpRw5mDZ28He-hknplPh0NxrqpYQm57tk21GIqo2oqU5DnehkdmcxfTjdZ5ap6INXiUpIgkT6qfY-LU-`

Decoded: This section shows the decoded JSON payload and the verification process.

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "RS256"  
}
```

PAYOUT: DATA

```
{  
  "id": "06d36c25-1edf-4639-9ce6-686516e1e2ee",  
  "correlationId": "postman-tests",  
  "imageStorageEnabled": true,  
  "created": "2021-01-14T09:17:11.264Z",  
  "expires": "2021-01-14T09:47:11.264Z"  
}
```

VERIFY SIGNATURE

RSASHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
-----BEGIN PUBLIC KEY-----
MIICijANBgkqhkiG9w0BAQEFAAO
Cg8AMIIICgKCAgEAyiy22/wOCl
iBNP+FEAHD
-----END PUBLIC KEY-----
)

Private Key. Enter it in plain text only if you want to generate a new token. The key never leaves your browser.

SHARE JWT

Signature Verified

SHARE JWT

Note: Also, you can verify the signature of JSON response programmatically according to the technology you use.