Oracle® Hospitality Desktop and Mobile ID Document Scanning Integration

API Scanning Specifications

Release 20.2 and higher.
F43697-02
September 2023

Oracle Hospitality Desktop and Mobile ID Document Scanning Integration API Scanning Specifications Release 20.2 and higher.

F43697-02

Contents

Preface

Purpose

This document outlines a technical implementation to allow OPERA Cloud version 20.2 and higher to interface with third-party scanning hardware devices and invoking cloud scanning solutions. This document covers communication with scanning devices where IFC8 property interface protocol or IP based integration is not used. The solution is agnostic and interfaces with any external hardware device or RESTful API that implements the required approach outlined below. This document outlines a technical implementation that facilitates this requirement.

Audience

Third-party vendors who wish to interface mobile and/or desktop scanning peripheral with OPERA Cloud. Oracle customers interested in mobile and/or desktop ID scanning solution for OPERA Cloud.

Customer Support

To contact Oracle Customer Support, access the Customer Support Portal at the following URL:

https://iccp.custhelp.com

When contacting Customer Support, please provide the following:

- Product version and program/module name
- Functional and technical description of the problem (include business impact)
- Detailed step-by-step instructions to re-create
- Exact error message received
- Screen shots of each step you take

Documentation

Oracle Hospitality product documentation is available on the Oracle Help Center at

http://docs.oracle.com/en/industries/hospitality/

Table 1 Revision History

Date	Description
October 2019	Initial Publication

Date	Description
December 2019	• Removed CORS header requirement for mobile scanning specifications • Added details about AccessToken expiration handling • Added examples for API requests/responses • Unified character case in all API header keys • Separated specs for desktop and mobile scanning APIs
January 2020	• Added “Image” element to Scan API response for passing full ID document image. Applies to both desktop and mobile solutions.
April 2020	• Added “IssueCountry” and “IssueCountryLong” element to Scan API response. Applies to both desktop and mobile solutions.
June 2021	• Updated Scan API method for desktop scanner solution to GET. • Removed "SignImage" and "PersonalNumber" from both desktop and mobile scanner solutions. • Marked fields NationalityLong, CountryLong, and IssueCountryLong as optional for both desktop and mobile scanner solutions. • Updated FaceImage and Image response format to Base64 (JPEG data encoded). • Updated description for Title and Alt Title attributes.
September 2023	• Replaced reference to oracleindustry.com with oraclecloud.com

1 Overview

For the purpose of managing guest data during a guest's stay, it is in some cases required by law or important for business reasons to have the ability to scan guest identification artifacts and store this data as part of the profile or reservation record. This capability should be available whether the user is working on a desktop workstation or a mobile device.

As OPERA Cloud is a web-based application, user workstations do not have Oracle supplied applications installed and running as a part of the typical OPERA Cloud instance. It is necessary to design a generic software bridge between OPERA Cloud' browser application and any third-party software that will enable the use of onboard or integrated hardware to capture images and subsequently identification data.

Supported Versions

This functionality is only available to OPERA Cloud versions 20.2 and higher.

2

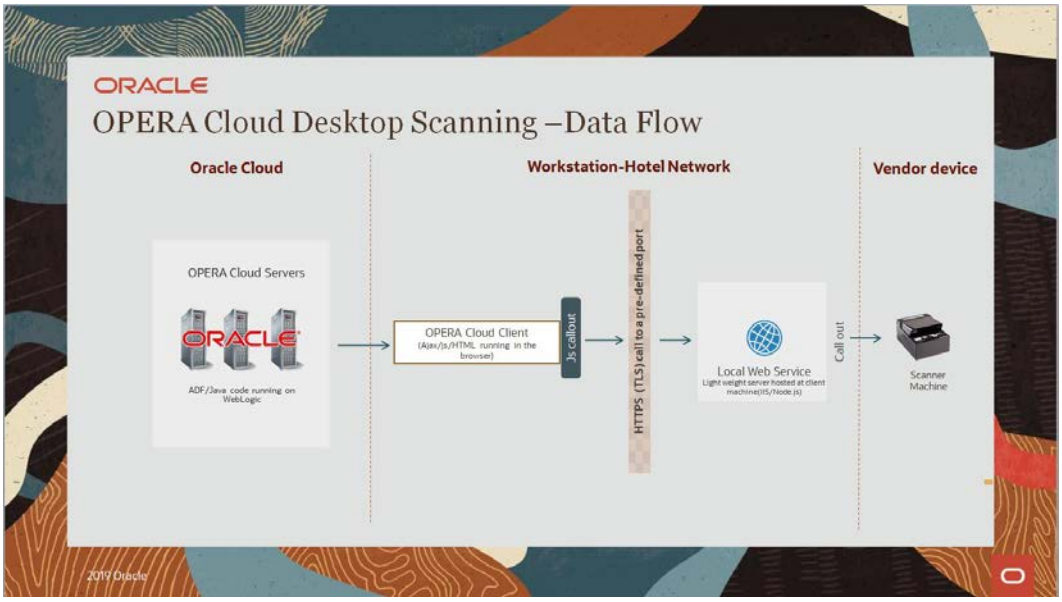
Desktop Scanning

Third-Party Responsibilities

Third parties who wish to interface a ID Document scanning peripheral with OPERA Cloud are required to install/ package and maintain a micro web service that runs on the local workstation that is to interface with the peripheral. This micro web service provides the communication channel that OPERA Cloud requires so it can interact with the third-party's peripheral. All communication and data passed via this micro web service is expected to conform to the standard SSL protocol. The SSL certificate must be a signed trusted certificate by the third-party vendor and reflect the domain name of the URL configured in OPERA Cloud. Self-signed certificates can be supported for these purposes as allowed by the browser installed on the workstation.

The micro web service operation should allow OPERA Cloud to invoke the client's connected device via an AJAX call and respond with a JSON payload (using the standard UTF-8 encoding) containing the data extracted from the ID document in the format specified in the Scan API Response Specifications section below.

Overview



To overcome the CORS and mixed content browser restrictions, the third-party must implement the following points. The third-party's micro web service installer is responsible for appending the client's host file with URL entry that OPERA Cloud can call. It should not simply overwrite the entire host file as clients may have existing custom configurations. This entry should contain the client's local IP address and a corresponding domain name (same as is specified in the SSL certificate mentioned above) that OPERA Cloud will use to issue requests to the micro web service. The host file entry should contain both IPv4 and IPv6 entries.

Example:

iPv4 Entry

127.0.0.1 YourCompanyDomain.com

iPv6 Entry

::1 YourCompanyDomain.com

iPv6 Entry

fe80::1%lo0 YourCompanyDomain.com

The micro web service should support universal CORS access. This can be achieved by including the following two entries in the micro web service's response header:

Access-Control-Allow-Origin: <Client's OPERA Cloud Domain>

Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept

The third-party is also responsible for providing either manual instructions for generating and installing the required SSL certificate on the workstation, or tooling that automates the process. If manual instructions are provided, a sample certificate should also be provided for testing the installation.

The third-party must also provide complete installation instructions of their solution, including technical documentation of their micro web service operations.

A resource from the third-party must be available to partner with the certification team on verifying their solution installation.

The expected responses for the desktop scanner are outlined in the Response Specifications tables in the Scan API section.

Scan API

This operation is expected to use **GET** HTTP method.

Request Specifications

Outlined below are the details of the request that will be issued from OPERA Cloud to the third-party vendor's micro web service.

Headers

Key	Description	Format	Sample Value
Content-Type	Content type of the request body	MIME type	application/json
Origin	CORS related header. Specifies the name of the originating OPERA Cloud domain	Alphanumeric	<instance>.oraclecloud.com
X-Requested-With	CORS related header. Specifies the type of AJAX request made	AJAX method	XMLHttpRequest
Accept	CORS related header. Specifies the content type expected to be sent back in the response	MIME type	application/json

Body

Key	Description	Format	Sample Value
Empty			

Example:

GET /scan HTTP/1.1

Host: vendor-domain.com:443

Content-Type: application/json

Origin: <instance>.oraclecloud.com

X-Requested-With: XMLHttpRequest

Accept: application/json

Response Specifications

Outlined Below are the details of the response that OPERA Cloud expects to be sent from the third-party vendor's micro web service

Headers

Key	Description	Format	Sample Value
Content type	Content type of the response body	MIME type	application/json; charset=utf-8
Access-Control-Allow-Origin	CORS related header. Specifies the OPERA Cloud domain(s) where the requests are allowed to be originated from	Alphanumeric	<instance>.oraclecloud.com
Access-Control-Allow-Headers	CORS related header. Specifies which headers should be allowed in the request	Alphanumeric	Origin, X-Requested-With, Content-Type, Accept

Body

Key	Description	Format	Sample Value
IDType	ID document type code	Pre-defined ID Type Codes: • PASSPORT • DRIVER_LICENSE • VISA • ID • UNKNOWN	PASSPORT
FirstName	The given name of the ID holder	Alphanumeric	<FIRSTNAME>
AltFirstName	ID holder's given name in alternate/native language	Alphanumeric	<ALTFIRSTNAME>
MiddleName	The middle name of the ID holder	Alphanumeric	<MIDDLENAME>
LastName	The surname of the ID holder	Alphanumeric	<LASTNAME>
AltLastName	ID holder's surname in alternate/native language	Alphanumeric	<ALTLASTNAME>
Title	Salutation or Honorary/academic title	Alphanumeric	<TITLE>

Key	Description	Format	Sample Value
AltTitle	Salutation or Honorary/academic title in alternative/native language	Alphanumeric	<ALTTITLE>
Gender	Gender of the ID holder	Pre-defined Gender Codes: • M = Male • F = Female • U = Unknown	M
BirthDate	Date of birth in ISO 8601 format	YYYY-MM-DD	<YYYY-YY-YY>
AltLanguage	Language code for alternate name information. Typically based on ISO 639-1 2-letter code	Alpha-2 code	<ALTLANG>
Nationality	ISO 3166-1, 2-letter country code representing the nationality of the ID holder	Alpha-2 code	<NATIONALITYID>
NationalityLong**	Description of the nationality	Alphanumeric	<NATIONALITYFULL>
Ethnicity*	The ethnicity of the ID holder	Alphanumeric	<ETHNICITY>
Province*	Province of the ID holder's address as per ISO 3166-2 standard	Alphanumeric	<STATE>
Address1	Street address line 1	Alphanumeric	<ADDRESS1>
Address2	Street address line 2, if needed	Alphanumeric	<ADDRESS2>
Address3	Street address line 3, if needed	Alphanumeric	<ADDRESS3>
Address4	Street address line 4, if needed	Alphanumeric	<ADDRESS4>
City	Town/city of the ID holder's address	Alphanumeric	<CITY>
County*	The county where the ID holder is residing	Alphanumeric	<COUNTY>
State	ISO 3166-2, 2/3-letter state code of ID holder's address	Alphanumeric	<STATE>
Country	ISO 3166-1, 2-letter country code of the ID holder's address	Alpha-2 code	<COUNTRYID>

Key	Description	Format	Sample Value
CountryLong**	Full country name of the ID holder's address	Alphanumeric	<COUNTRYFULL>
PostalCode	The zip/postal code where the ID holder is residing	Numeric	<ZIPCODE>
PostalCodeExt	Additional zip/postal code extension designating a more specific location	Numeric	<ZIPCODEEXT>
IDNumber	The number issued to the ID to uniquely identify the ID holder	Alphanumeric	<IDNUMBER>
IssueDate	The ID document issue date in ISO 8601 date format	YYYY-MM-DD	<XXXX-XX-XX>
IssueCountry	ISO 3166-1, 2-letter country code of the ID Country of Issue	Alpha-2 code	<COUNTRYID>
IssueCountryLong**	Full country name of the ID Country of Issue	Alphanumeric	<COUNTRYFULL>
PlaceOfIssue	The city/place/town where the ID document was issued	Alphanumeric	<ISSUEPLACE>
PlaceOfBirth	The city/place/town where the ID holder was born	Alphanumeric	<BIRTHPLACE>
ExpirationDate	The ID document expiry date in ISO 8601 date format	YYYY-MM-DD	<XXXX-XX-XX>
FaceImage***	The ID document cropped face image in Base64 format with JPEG image data encoded	Base64 (JPEG data encoded)	<JPEG:BASE64>
Image***	The full ID document image in Base64 format with JPEG image data encoded	Base64 (JPEG data encoded)	<JPEG:BASE64>
ResultCode	Error code specified by the vendor in case of failure while scanning the document	Numeric	<CODE>
ResultDescription	Error message specified by the vendor in case of failure while scanning the document	Alphanumeric	<DESC>

Legend:

* Fields marked with a red asterisk will be available on OPERA Cloud screens in the future.

** Fields marked with two red asterisks are optional. They are not used in OPERA Cloud, but they might be used for support and troubleshooting purposes.

*** Images must be returned as JPEG image type encoded in Base64 format. If the image type or format is invalid, the application will return the following error: "File Content is not recognized by the application."

Example:

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

Access-Control-Allow-Origin: <instance>.oraclecloud.com

Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept

{
"IDType": "PASSPORT",
"FirstName": "FIRSTNAME",
"AltFirstName": "ALTFIRSTNAME",
"MiddleName": "MIDDLENAME",
"LastName": "LASTNAME",
"AltLastName": "ALTLASTNAME",
"Title": "TITLE",
"AltTitle": "ALTTITLE",
"Gender": "M",
"BirthDate": "XXXX-XX-XX",
"AltLanguage": "ALTLANG",
"Nationality": "NATIONALITYID",
"NationalityLong": "NATIONALITYFULL",
"Ethnicity": "ETHNICITY",
"Province": "STATE",
"Address1": "ADDRESS",
"Address2": "ADDRESS2",
"Address3": "ADDRESS3",
"Address4": "ADDRESS4",
"City": "CITY",
"County": "COUNTY",

"State": "STATE",
"Country": "COUNTRYID",
"CountryLong": "COUNRTYFULL",
"PostalCode": "ZIPCODE",
"PostalCodeExt": "ZIPEXT",
"IDNumber": "IDnumber",
"IssueDate": "XXXX-XX-XX",
"IssueCountry": "COUNTRYID",
"IssueCountryLong": "COUNTRYFULL",
"PlaceOfIssue": "ISSUEPLACE",
"PlaceOfBirth": "BIRTHPLACE",
"ExpirationDate": "XXXX-XX-XX",
"FaceImage": "JPEG:BASE64"
"Image": "JPEG:BASE64"
"ResultCode": "CODE",
"ResultDescription":"DESC"
}

3

Mobile Scanning

Overview

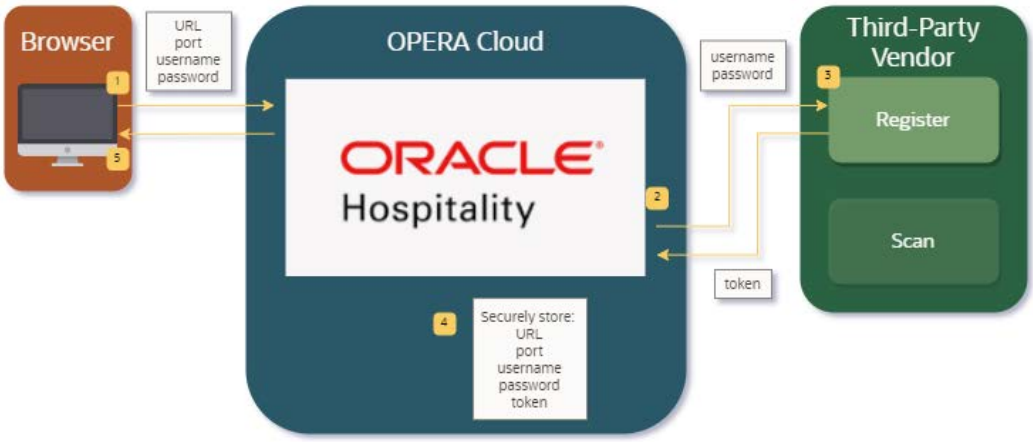
Mobile scanning is a cloud initiative for scanning ID documents with reduced hardware and maintenance costs. In this architecture, no peripheral device is required to be connected to the user's workstation, which means there is no need for a micro web service running on the property's network to interface between the device and OPERA Cloud. Instead, utilizing the built-in device camera, an image of the ID document can be sent to a cloud-hosted RESTful API (provided by the third-party) which processes the image and returns the JSON formatted data according to the specifications outlined in the Scan API Response Specifications section below. All communication and data is expected to conform to the standard SSL protocol. The property is responsible for registering their mobile devices with the third-party vendor. The mobile scanning solution utilizes OPERA Cloud's image capture functionalities to capture a photo of the ID document supported by the third-party and makes a REST call to the third-party API passing the base64 encoded image for processing.

It is the vendor's responsibility to ensure security and privacy as well as local regulation conform processing of the ID Document data.

Third-Party Responsibilities

Third-parties should provide two RESTful APIs:

1. Client Registration API – This API provides a unique token identifying the client to the third-party vendor. This token is used to authenticate calls to the scan API detailed below. To call the client registration API, credentials must be provided which identify the client/property to the third-party vendor. These credentials are provided to the client by the third-party and configured in OPERA Cloud for the purposes of calling this API only. After successful registration on the third-party side, a unique access token is sent in the response which OPERA Cloud will use in the header for all subsequent scan API calls along with the date time (UTC). Combined, these details should be validated by the third-party scan API in order to accept the request and process the given image. All access tokens must expire before a 1-year period from the issue date.



2. Scan API – This API processes the base64 encoded ID document image passed in the request and extracts the personal information from the Machine Readable Zone (MRZ) of the document. The response returns a JSON payload conforming to the specifications outlined below containing the extracted personal information. This API should be available as both third-party cloud-hosted and on premise hosted based on the client's requirements. This API requires the access token, which was obtained from the client registration API detailed above, to be passed in the request headers. If the access token has expired and a call to this API is made using it, the response must return a 401 Unauthorized HTTP error code.

Client Registration API

This operation is expected to use **POST** HTTP method.

Request Specifications

Outlined below are the details of the request that will be issued from OPERA Cloud to the third-party vendor's hosted API.

Headers

Key	Description	Format	Sample Value
AccountId	Username issued to the client by the third-party vendor	Alphanumeric	<USER>
AccessSecret	Password issued to the client by the third-party vendor	Alphanumeric	<PASSWORD>
ApiKey	Unique identification key associated to the client's third-party vendor license to authorize access to this API	Alphanumeric	<KEY>

Body

Key	Description	Format	Sample Value
Empty			

Example:

POST /client-register HTTP/1.1

Host: vendor-domain.com:443

AccountId: <USER>

AccessSecret: **********

ApiKey: <KEY>

Content-Length: 0

Cache-Control: no-cache

Response Specifications

Outlined below are the details of the request that will be issued from OPERA Cloud to the third-party vendor's hosted API.

Headers

Key	Description	Format	Sample Value
Content-Type	Specifies the formatting of the response body	Alphanumeric	application/json; charset=utf-8

Body

Key	Description	Format	Sample Value
AccessToken	Response token which grants access to the scan API for this particular property	Alphanumeric	<TOKENVALUE>
ResultCode	Vendor specific response code to indicate success or errors. Error codes help to debug vendor responses	Alphanumeric	<CODE>
ResultDescription	User friendly description to shown on the screen in OPERA Cloud configuration	Alphanumeric	<DESC>

Example:

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

```
{
   "AccessToken": "TOKENVALUE",
   "ResultCode": "CODE",
   "ResultDescription": "DESC"
}
```

Scan API

This operation is expected to use **POST** HTTP method.

Request Specifications

Outlined below are the details of the request that will be issued from OPERA Cloud to the third-party vendor's hosted API.

Headers

Key	Description	Format	Sample Value
AccessToken	Access token generated by third party through client registration API	Alphanumeric	<TOKENVALUE>
DateTime	Date and time of request in ISO format	ISO 8601 standard date time (YYYY-DD-MM HH:MM:SS)	<XXXX-XX-XX>
Content-Type	Content type of the request body	MIME type	application/json

Body

Key	Description	Format	Sample Value
Image	Captured image in base64 format with JPEG image data encoded	Base64 (JPEG data encoded)	<JPEG:BASE64>

Example:

POST /scan HTTP/1.1

Host: vendor-domain.com:443

Content-Type: application/json

AccessToken: "TOKENVALUE",

DateTime: <XXXXX-XX-XX HH:MM:SS>

{

"Image": " IMGBASE64"

}

Response Specifications

Outlined Below are the details of the response that OPERA Cloud expects to be sent from the third-party vendor's micro web service

Headers

Key	Description	Format	Sample Data
Content-Type	Content type of the response body	MIME type	application/json; charset=utf-8

Body

Key	Description	Format	Sample Data
IDType	ID document type code	Pre-defined ID Type Codes: PASSPORT DRIVER_LICENSE VISA ID UNKNOWN	PASSPORT
FirstName	The given name of the ID holder	Alphanumeric	<FIRSTNAME>
AltFirstName	ID holder's given name in alternate/native language	Alphanumeric	<ALTFIRSTNAME>
MiddleName	The middle name of the ID holder	Alphanumeric	<MIDDLENAME>
LastName	The surname of the ID holder	Alphanumeric	<LASTNAME>
AltLastName	ID holder's surname in alternate/native language	Alphanumeric	<ALTLASTNAME>
Title	Salutation or Honorary/academic title	Alphanumeric	<TITLE>
AltTitle	Salutation or Honorary/academic title in alternative/native language	Alphanumeric	<ALTTITLE>
Gender	Gender of the ID holder	Pre-defined Gender Codes: M = Male F = Female U = Unknown	M

Key	Description	Format	Sample Data
BirthDate	Date of birth in ISO 8601 format	YYYY-MM-DD	<XXXX-XX-XX>
AltLanguage	Language code for alternate name information. Typically based on ISO 639-1 2-letter code	Alpha-2 code	<ALTLAG>
Nationality	ISO 3166-1, 2-letter country code representing the nationality of the ID holder	Alpha-2 code	<NATIONALITYID>
NationalityLong**	Description of the nationality	Alphanumeric	<NATIONALITYFULL>
Ethnicity*	The ethnicity of the ID holder	Alphanumeric	<ETHNICITY>
Province*	Province of the ID holder's address as per ISO 3166-2 standard	Alphanumeric	<STATE>
Address1	Street address line 1	Alphanumeric	<ADDRESS1>
Address2	Street address line 2, if needed	Alphanumeric	<ADDRESS2>
Address3	Street address line 3, if needed	Alphanumeric	<ADDRESS3>
Address4	Street address line 4, if needed	Alphanumeric	<ADDRESS4>
City	Town/city of the ID holder's address	Alphanumeric	<CITY>
County*	The county where the ID holder is residing	Alphanumeric	<COUNTY>
State	ISO 3166-2, 2/3-letter state code of ID holder's address	Alphanumeric	<STATE>
Country	ISO 3166-1, 2-letter country code of the ID holder's address	Alpha-2 code	<COUNTRYID>
CountryLong**	Full country name of the ID holder's address	Alphanumeric	<COUNTRYFULL>

Key	Description	Format	Sample Data
PostalCode	The zip/postal code where the ID holder is residing	Numeric	<ZIPCODE>
PostalCodeExt	Additional zip/postal code extension designating a more specific location	Numeric	<POSTEXT>
IDNumber	The number issued to the ID to uniquely identify the ID holder	Alphanumeric	<IDNUMBER>
IssueDate	The ID document issue date in ISO 8601 date format	YYYY-MM-DD	<XXXX-XX-XX>
IssueCountry	ISO 3166-1, 2-letter country code of the ID Country of Issue	Alpha-2 code	<COUNTRYID>
IssueCountryLong**	Full country name of the ID Country of Issue	Alphanumeric	<COUNTRYFULL>
PlaceOfIssue	The city/place/town where the ID document was issued	Alphanumeric	<ISSUEPLACE>
PlaceOfBirth	The city/place/town where the ID holder was born	Alphanumeric	<BIRTHPLACE>
ExpirationDate	The ID document expiry date in ISO 8601 date format	YYYY-MM-DD	<XXXX-XX-XX>
FaceImage***	The ID document cropped face image in Base64 format with JPEG image data encoded	Base64 (JPEG data encoded)	<JPEG:BASE64>
Image***	The full ID document image in Base64 format with JPEG image data encoded.	Base64 (JPEG data encoded)	<JPEG:BASE64>
ResultCode	Error code specified by the vendor in case of failure while scanning the document	Numeric	<CODE>

Key	Description	Format	Sample Data
ResultDescription	Error message specified by the vendor in case of failure while scanning the document	Alphanumeric	<DESC>

Legend:

* Fields marked with a red asterisk will be available on OPERA Cloud screens in the future.

** Fields marked with two red asterisks are optional. They are not used in OPERA Cloud, but they might be used for support and troubleshooting purposes.

*** Images must be returned as JPEG image type encoded in Base64 format. If the image type or format is invalid, the application will return the following error: "File Content is not recognized by the application."

Example:

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

{
"IDType": "PASSPORT",
"FirstName": "FIRSTNAME",
"AltFirstName": " ALTFIRSTNAME",
"MiddleName": "MIDNAME",
"LastName": "LASTNAME",
"AltLastName": " ALTLASTNAME",
"Title": "TITLE.",
"AltTitle": "ALTTITLE",
"Gender": "M",
"BirthDate": "XXXX-XX-XX",
"AltLanguage": "ALTLANG",
"Nationality": “NATIONALITYID",
"NationalityLong": "NATIONALITYFULL",
"Ethnicity": "ETHNICITY",
"Province": "STATE",
"Address1": "ADDRESS1",

"Address2": "ADDRESS2",
"Address3": "ADDRESS3",
"Address4": "ADDRESS4",
"City": "CITY",
"County": "COUNTY",
"State": "STATE",
"Country": "COUNTRYID",
"CountryLong": "COUNTRYFULL",
"PostalCode": "ZIPCODE",
"PostalCodeExt": "ZIPEXT",
"IDNumber": "IDNUMBER",
"IssueDate": "XXXX-XX-XX",
"IssueCountry": " COUNTRYID ",
"IssueCountryLong": "COUNTRYFULL",
"PlaceOfIssue": "ISSUEPLACE",
"PlaceOfBirth": "BIRTHPLACE",
"ExpirationDate": "XXXX-XX-XX",
"FaceImage": "JPEG:BASE64",
"Image": "JPEG:BASE64",
"ResultCode": "CODE",
"ResultDescription": "DESC"
}