VeriDas

/Phygital

/Integration Biometric Management System (BMS) API

Index

/0. Scope

This document is intended for access control system providers and integrators. It is a supplement to the detailed technical documentation of the Veridas Biometric Management System (BMS) registration API, which is available online at the following web address: https://api.app.das-gate.com/public/docs

/1. Attachments

- *BMS_API_collection.json*: Contains the most commonly used API endpoints explained in section4

- *BMS_API_test_environment.json*: Contains the variables that should be used in a test environment

- *BMS_API_production_environment.json*: Contains the variables that should be used in a production environment

/2. Test and production environments

The Veridas BMS API is available in two environments, a test environment and a production environment. The following table shows the main differences between the two:

Feature	Test	Production
Availability	Monday-Friday 8:00 to 22:00 CET	24x7
Update	They can occur without warning, and can make the environment unavailable for a few minutes	Updates are reported
Cost	Free	Determined by contract
Capacity	Restrictions on the number and size of requests may apply	Determined by contract
URL	https://api.work-srv.das-gate.com/public	https://api.srv.das-gate.com/public

/3. API Authentication

Authentication within the **BMS API** is based on **OpenID Connect (OIDC)**, which is an authentication protocol implemented on top of the OAuth 2.0 authorization framework.

In the following sections, the authentication process for API calls is specified in more detail with examples and the available resources are also described in a detailed way.

3.1 Token generation

Veridas will provide the client with two pieces of information that identify the client and are necessary to obtain the access token: *client_id* and *client_secret*. With these two pieces of information, an access token must be obtained by calling the API endpoint /token as follows:

```
POST /auth/realms/{TENANT_ID}/protocol/openid-connect/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Accept: */*
Host: iam.work-srv.das-gate.com
content-length: XX

grant_type=client_credentials&scope=roles&client_id=CLIENT_ID&client_secret=CLIENT_S
ECRET
```

Veridas will provide for each customer with: ***CLIENT_ID, TENANT_ID*** *and* ***CLIENT_SECRET***, so the integration solution will allow us to configure these parameters for every client. **These variables must be defined in the attached environment file,** in addition to the *API_KEY* variable also provisioned by Veridas.

The obtained response is a JSON with the following format:

```
{'access_token': 'eyJhbGc....',
 'expires_in': 300,
 'token_type': 'bearer',
 'not-before-policy': 0,
 'session_state': '3f9a7a76-cd4a-4082-b3e0-95686bac24ee',
 'scope': 'email profile roles'
}
```

From this response, the *access_token* shall be extracted for use in API requests as follows. The *expires_in* field indicates the validity period of the token in seconds.

3.2 Refreshment of tokens

When the token expires, it is necessary to generate a new one. To do so, it is necessary to carry out the operation indicated in point 3.1 again.

It is also possible to use the **Refresh Token grant**, although it involves sending practically the same as in previous section *"Token generation"*:

```
POST /auth/realms/{TENANT_ID}/protocol/openid-connect/token HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Accept: */*
Host: iam.work-srv.das-gate.com
content-length: XX

grant_type=refresh_token&refresh_token=REFRESH_TOKEN&client_id=CLIENT_ID&client_secr
et=CLIENT_SECRET
```

/4. Endpoints

Below, the most commonly used endpoints of the API are described. The definition of the parameters for these endpoints is provided in the Annex.

4.1 User registration

Operation	Description
POST /singles ***1**	Registration of a single user by sending a selfie and the user's identification data
POST /batches ***2**	Batch registration, allowing the sending of a collection of selfies and data of the users to be registered
GET /singles/{onboarding_id}	Obtaining the result of the registration process, and the Biometric QR when so configured, depending on the Accept header
GET /batches/{batch_id}	Obtaining the result of the batch registration process

1 - Selfie limitations

In order for the biometrics engine to generate an optimal representation for authentication systems, the following minimums are recommended:

- *Minimum selfie size accepted: 3 KB*

- *Distance between eyes, recommended greater than 200 pixels*
- *Images without backlight*
- *Images without excessive noise*
- *Images with the user's frontal face, a natural expression and without obstructing accessories*
- *Warning: Rotated or flipped photographs must not be used*

2 - Batches limitation

**There are limitations on the maximum file size that can be uploaded. Currently, the limits are:*

- *For each entry, selfie limitations apply*
- *Maximum number of onboardings accepted in a batch onboarding process is 1000*
- *Maximum size for zip archive with media content is 1 GB*

4.2 User and account management

Operation	Description
GET /users	Search for registered users
GET /users/{user_id}	Getting the details of a user
DELETE /accounts/{user_id}	Deletion of user account (IAM, evidence, credentials)

4.3 Credential management for biometric authentication

Operation	Description
GET /credentials/{user_id}	Gets the credentials of that user in the system
PUT /credentials/{user_id}/{wanted_state}	Disabling or re-enabling a user's credentials for authentication processes
PUT /credentials/{wanted_state}	Disabling or re-enabling credentials for all users for authentication processes
GET /credentials	Allows searching for credentials by other criteria, such as the source access control system identifier, *acs_id*

POST /credentialresets	Regenerating a credential for a user. It is necessary to send the *user_id* and the *sid* of the credential you want to reset
GET /credentialresets/{user_id}	Getting a user's credential regenerations

/Annex Parameters

- ***user_id***: User identifier in the BMS. Format: UUID
- ***onboarding_id***: Onboarding identifier. Format: UUID
- ***portrayal_type***: Type of registration (“selfie”)
- ***acs_id***: User identifier in the external system
- ***acs_id_schema***: Format of the *acs_id* field. Typically, “string64”
- ***selfie***: User's photo.
  Format: "..."
- ***deliver_by_email***: Boolean value. Send QR code by email. If this value is "true", the email is mandatory
- ***show_username***: Boolean indicating whether the username field will appear on the terminal screen once the user has authenticated. It is an optional field
- ***username***: This field allows you to enter a name that identifies the user
- ***email*:** Email Address
- ***wanted_state*:** Desired credentials state (active/inactive)
- ***sid*:** Single credential ID. Format: UUID