

Unblock Documentation

Api Verestro | User flow documentation

Implemented endpoints

All active endpoints are exposed under the `user` controller:

- `GET /user`
- `GET /user/documents-required/:externalUserId`
- `GET /user/get-verification-status/:externalUserId`
- `GET /user/validate-user/:externalUserId`
- `POST /user/create`
- `POST /user/upload-document`
- `POST /user/general-document`
- `POST /user/upload-video`

`POST /usr-document/upload-video` is not an active endpoint in the current codebase.

User flow (step-by-step)

1. **Create the user** `POST /user/create`
2. **Check required documents** `GET /user/documents-required/:externalUserId`
3. **Upload required documents** `POST /user/upload-document`
4. **Optionally upload complementary files** `POST /user/general-document` `POST /user/upload-video`
5. **Submit user for verification** `GET /user/validate-user/:externalUserId`
6. **Check verification status** `GET /user/get-verification-status/:externalUserId`

Create user

`POST /user/create`

Purpose: Create and register a user associated with the merchant identified by the API key.

Request body (application/json)

Name	Description	Type	Required
externalUserId	Merchant-side unique user identifier	String	Yes
phone	User phone number including country code	String	Yes
isoCodePhone	ISO 3166-1 alpha-2 country code for phone	String	Yes
name	User first name	String	Yes
lastName	User last name	String	Yes
email	User email	String	Yes
isoCodeCountryResidence	ISO 3166-1 alpha-2 residence country code	String	Yes
address	User residential address	String	Yes
isoCodeCountryNationality	ISO 3166-1 alpha-2 nationality country code	String	Yes
state	State or region	String	Yes
city	City	String	Yes
dateOfBirth	Date of birth in YYYY-MM-DD format	String	Yes
zipCode	Postal code	String	Yes

Notes:

- dateOfBirth must be a valid date in YYYY-MM-DD format.
- The user must be at least 18 years old.

Example request body

```
{
  "externalUserId": "1337",
  "phone": "+573221112233",
  "isoCodePhone": "C0",
  "name": "Jhon",
  "lastName": "Doe",
```

```
"email": "jhondoe@test.com",
"isoCodeCountryResidence": "US",
"address": "123 Main St",
"isoCodeCountryNationality": "US",
"state": "Florida",
"city": "Miami",
"dateOfBirth": "1990-02-15",
"zipCode": "12345"
}
```

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "",
  "serverTime": "2026-01-16T14:38:34.638Z",
  "data": {
    "idUser": 507,
    "externalUserId": "1337",
    "uniqueCode": "USR-000507"
  },
  "success": true,
  "codeError": 0
}
```

Get user

GET /user

Purpose: Retrieve a user by `externalUserId` or by `idUser`.

Query parameters

Name	Description	Type	Required
------	-------------	------	----------

externalUserId	Merchant-side unique user identifier	String	Conditionally
idUser	Internal Verestro user identifier	Number	Conditionally

Rules:

- Send exactly one filter.
- Valid combinations:
 - GET /user?externalUserId=1337
 - GET /user?idUser=507
- Invalid:
 - sending both
 - sending neither

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "Operation successful",
  "serverTime": "2026-01-16T17:20:00.000Z",
  "data": {
    "idUser": 507,
    "externalUserId": "1337",
    "firstName": "Jhon",
    "lastName": "Doe",
    "email": "jhondoe@test.com",
    "phone": "+573221112233",
    "isoCodePhone": "CO",
    "address": "123 Main St",
    "zipCode": "12345",
    "dateOfBirth": "1990-02-15T00:00:00.000Z",
    "verified": false,
    "verifiedStatus": 1,
    "status": true,
    "isBlocked": false,
    "uniqueCode": "USR-000507"
  }
}
```

```
},
"success": true,
"codeError": 0
}
```

Retrieve user required documents

GET /user/documents-required/:externalUserId

Purpose: Retrieve the list of required documents for a specific user, including whether each document is already loaded.

Path parameters

Name	Description	Type	Required
externalUserId	Merchant-side unique user identifier	String	Yes

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "Operation successful",
  "serverTime": "2026-01-16T17:33:49.049Z",
  "data": [
    {
      "docType": "IDENTITY_CARD_FRONT",
      "isLoading": false
    },
    {
      "docType": "IDENTITY_CARD_BACK",
      "isLoading": false
    }
  ]
}
```

```
{,
{
  "docType": "PASSPORT",
  "isLoading": false
},
{
  "docType": "DRIVERS_LICENCE_FRONT",
  "isLoading": false
},
{
  "docType": "DRIVERS_LICENCE_BACK",
  "isLoading": false
},
{
  "docType": "UTILITY_BILL",
  "isLoading": false
},
{
  "docType": "FRONTAL_PHOTO",
  "isLoading": true
},
{
  "docType": "WORK_PERMIT_FRONT",
  "isLoading": false
},
{
  "docType": "WORK_PERMIT_BACK",
  "isLoading": false
}
],
"success": true,
"codeError": 0
}
```

Upload user required document

POST /user/upload-document

Purpose: Upload one of the required verification documents for a user.

Request body (multipart/form-data)

Name	Description	Type	Required
externalUserId	Merchant-side unique user identifier	String	Yes
nameDocument	Display name for the file	String	No
docType	Required document type	Enum	Yes
file	File to upload	File (Binary)	Yes

Allowed docType values

- IDENTITY_CARD_FRONT
- IDENTITY_CARD_BACK
- PASSPORT
- DRIVERS_LICENCE_FRONT
- DRIVERS_LICENCE_BACK
- UTILITY_BILL
- FRONTAL_PHOTO
- WORK_PERMIT_FRONT
- WORK_PERMIT_BACK

Allowed mimetypes for file

- image/jpeg
- image/png
- application/pdf

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "Operation successful",
  "serverTime": "2026-01-16T16:41:29.289Z",
  "data": {},
  "success": true,
```

```
"codeError": 0
}
```

Request user verification

GET /user/validate-user/:externalUserId

Purpose: Validate that a user has uploaded the required documents and mark the account as ready for verification.

Current validation rules in code:

- `FRONTAL_PHOTO` is part of the required-document set when configured as required for the merchant.
- One valid identity set can satisfy the identity-document requirement:
 - `PASSPORT`
 - `IDENTITY_CARD_FRONT` and `IDENTITY_CARD_BACK`
 - `DRIVERS_LICENCE_FRONT` and `DRIVERS_LICENCE_BACK`
 - `WORK_PERMIT_FRONT` and `WORK_PERMIT_BACK`
- Video is not currently required by this endpoint.

Path parameters

Name	Description	Type	Required
externalUserId	Merchant-side unique user identifier	String	Yes

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "Operation successful",
  "serverTime": "2026-01-16T17:58:43.303Z",
  "data": {},
  "success": true,
  "codeError": 0
}
```

```
}
```

Example error response

```
{
  "headerStatus": {
    "code": 400,
    "description": "Error"
  },
  "messages": "Missing documents: IDENTITY_CARD_FRONT, IDENTITY_CARD_BACK, PASSPORT",
  "serverTime": "2026-01-16T17:51:18.570Z",
  "data": {},
  "success": false,
  "codeError": 134
}
```

Upload user general document

POST /user/general-document

Purpose: Upload a general-purpose user document that is not part of the required verification-document catalog.

Request body (multipart/form-data)

Name	Description	Type	Required
externalUserId	Merchant-side unique user identifier	String	Yes
nameDocument	Display name for the file	String	No
file	File to upload	File (Binary)	Yes

Allowed mimetypes for file

- video/mp4
- video/webm
- video/mpeg
- application/pdf
- image/jpeg

- `image/png`
- `image/webp`
- `application/msword`
- `application/vnd.openxmlformats-officedocument.wordprocessingml.document`
- `application/vnd.ms-excel`
- `application/vnd.openxmlformats-officedocument.spreadsheetml.sheet`
- `text/plain`

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "Operation successful",
  "serverTime": "2026-01-16T16:34:05.519Z",
  "data": {},
  "success": true,
  "codeError": 0
}
```

Upload user video

`POST /user/upload-video`

Purpose: Upload or replace the user video used for verification flows that require video evidence.

Request body (`multipart/form-data`)

Name	Description	Type	Required
externalUserId	Merchant-side unique user identifier	String	Yes
fileName	Display name for the file	String	No
video	Video file to upload	File (Binary)	Yes

Allowed mimetypes for `video`

- video/mp4
- video/webm
- video/mpeg

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "Operation successful",
  "serverTime": "2026-01-16T16:34:05.519Z",
  "data": {},
  "success": true,
  "codeError": 0
}
```

Retrieve user verification status

GET /user/get-verification-status/:externalUserId

Purpose: Retrieve the current verification status of a specific user.

Path parameters

Name	Description	Type	Required
externalUserId	Merchant-side unique user identifier	String	Yes

Possible status values

- VERIFIED
- NOT_VERIFIED

Example success response

```
{
  "headerStatus": {
    "code": 200,
    "description": ""
  },
  "messages": "Operation successful",
  "serverTime": "2026-01-16T17:43:27.079Z",
  "data": {
    "status": "NOT_VERIFIED"
  },
  "success": true,
  "codeError": 0
}
```

Revision #2

Created 14 June 2026 08:40:49 by Krzysztof Wereski

Updated 14 June 2026 08:48:10 by Krzysztof Wereski