Your APIs for us - Notifications

We can send following information to your API endpoints:

- 3DS OTP code, so you can handle delivery to the user yourself via SMS, Push or other channel.
- Notification about outcome of KYC process.
- Simple notification about transactions.

To make this work, you need to expose an API according to relevant section of this documentation.

Security

To set secured server-server connection between our services Verestro requires a similar connection as in the case of client to Verestro communication based on the x509 certificate, more about this model here.
In the first step, Verestro will send to the client a CSR for the dev and production environments, more details here.
The next step is for the client to sign the CSR and send the certificate back to Verestro along with the base URL for the methods listed below. Verestro will authorize itself with each request with a certificate, which should be checked on the client side.

Additional data encryption & integration

Some requests and responses contain sensitive data, to additionally secure the connection we require JSON Web Encryption (JWE).

	normal	encrypted
Example of request with sensitive data	{ "cardNo" : "5555444455554444" }	{ "payload" : "very long JWE token" }

	normal	encrypted
Example ofresponse with sensitive data	{ "id" : 1125, "type" : "1125" , "cvv" : "123" , "cardNo" : "5555444455554444" , "exp" : "2026-01-31" }	{ "payload" : "very long JWE token" }

Idempotency Key

With some requests additional header X-Idempotency-Key could be send. This header contain unique random id allowing to identify single request.
If client send this header, operation should be triggered only once and for any further request with this key, response should be identical - in most cases, returned from cache.

example headers:

```
X-Idempotency-Key: 20e87975-dbfb-4c95-b239-169516c0b707
```

JWE configuration

connection we need from you enc and alg from JWE parameters. Acceptable values are:

- Algorithm used by Verestro to encipher content of message (enc) - A256GCM,
- Algorithm used by Verestro to encipher encryption key (alg) - RSA-OAEP-256,
- Algorithm needed from you to encipher content of message (enc) - A256GCM,
- Allowed algorithms for key encryption (alg) - RSA-OAEP-256 or RSA-OAEP.

Recommended JWE libraries for various programming languages:

- PHP,
- JAVA.

Request:

To process encrypted message you need to perform a few additional steps on top of standard message processing:

- Verestro add headlines:
  - Public-Key through which you can encrypt response to us (if needed),
  - Encrypted-Request headline confirming message encryption; value true or false,

- Expose endpoint with your Public Key - you will find a similar endpoint in our technical documentation, GET /secure/public_key
- Use Verestro Public Key to create JWE and transfer data in payload,

Response:

When the response contains sensitive data that requires encryption, use Verestro public key encryption available here GET /secure/public_key

Additional information:

- In case of errors (i.e. validation errors) you will receive unencrypted response,
  - ENCRYPTION_REQUIRED,
  - INVALID_PUBLIC_KEY,
  - INVALID_PAYLOAD,
  - CANT_DECRYPT_PAYLOAD.

Example request:

Correct request	Sent request (incorrect)	Received by CMS Antaca (after decipher action with private key)

`{"card_no" : 1337}`	{"payload" : "eyJhbGciOiJSU0EtT0FFUC0yNTYiLCJlbmMiOiJBMjU2R0NNIn0.rdUrW12XCZQgLFDJ-2zAHWYYnaAanctceE1-Y6yJUplX0B2dLu-bvYOEJ83KxxUs-ZjA41R4PmAVilx1cTF4pv-7CZR0_ki85XRATBYF2-MvZdcC81fHy2QPU_ZsAEWAW00a1wKJmuEsgPB2m1aLZ7oK4fC1hciep4PyAtuWQRYHjhNb-UDT41_gDKTbnSGTwheL7S0mAJ_HsKfnZFHYUrM77UcxQGZKnH7Mzqvndf9THiMo0-3MWliYFDAm1bqN2_KTIoBNCprYjFnyIXPCjib73bjWX_P2ip5Ul84cngbQmFVzc7o91JrpJvYou1INS7zL4XKLFcADN4nZ_9ePWsm5_kX5SOMyUyEhOC9gusrLNAJ0MHaIFHni8WqnMAWM3_MC4OQDYetKax5bnHK6x42_5eFaf6ZmzmioKny5aGm-4Vo8TEu691FmPxglhyenWlMhvBvf6ZeVsy58Ofr0mi3TXjwYbAyas7m6sncxZu1FhEJ4da6gtNjmjuKdikOOntu8V71QQ07nczNqfGlUv0RcUc9uKJq5je4b9BEbK9WuQcroxmALqC4HTt1xhICHrVUA0d_t3fglhS2n7wNaKKCFq70ZWIrpdTaBd35kdVQOEjZgCavSjbZOzgOzcEqS6P2Blm7bZ7ZZBmnfk8y8M4m0xWoQNTmLC6nqz9bSbME.UEryKNClDxQZpyWu.6Lw_5CcZ9HiVxHfi_XTAFw.pYbQ6tdmQYe1kiPonm1GhA"}	`{"card_no" : 1337}`

Example response:

Correct response	Sent response by CMS Antaca	Received by you

`{"card_no" : 1338}`	{"payload" : "eyJhbGciOiJSU0EtT0FFUC0yNTYiLCJlbmMiOiJBMjU2R0NNIn0.iPmvEKtMAMrrEiR89vlwsL77ZfqxXrcMiy-bx3z6_7HAo__aQzBpMVDtLyj3kTHYWxen8bhPuVyebXyaIHL20sekFzcIFFzvaGoyQYU6zOK8tPv81tgixQe8SDnEr5v9VWBfiHxtPvqlpQlig2is5ynBkyqjdpQWEagR3MpqpATGl7f-omG82Jq0OwZByWI8I6P89hczwgK37F-MUnQDxcRUM3RagbHKNeIcfmPdJpNeqFZHe45y4wUkTWN0uzW72qydkN_4uM9fy0nrUpgsJNbtJGAVIUVmDz4pIZkiI1zyGbfZX-PT7Wh9UNM06gEUf4i2goZY-m4wPB0n2zXvxzcEdfTH27iPp-aKiJjfJpYb_ZnHyklk__gZlAy9r7W0594dY-eBJ_iUa5aeDsFS2TIfsfjMJsL8NRWY2noiTw5lsneD8dwvr6N_rYcWoFXDyWXHoRitSSd2iYrB80gbeSOBW0wfKtPxNIZrR0uDhkE8FouS5Pk7QBw412kd43GtrEpAijqn3ne7MNUpCtuNfJ8e_NdGDLTR7CSHhC0jfFlchpIvklF42o216NO-OnyJsjdv1w4_w1ugs61fTHDl8lgBalOjOxauKwIvJJOyFdWmpjlXuzJhrray7ov25uh2ibvFv3Gfd2iuGUnLIZzYBOTT8ftGWTCGXTDvVOvzGbs.c3qMNb2Bne-7g0Wz.PInghFM6Q8Gn0p4Tlebig32s-ZrpLqTMqQDlpXLLYx0iq-StrKco_HrjdN4MxondP4CicCgseIjcV8JR29jKYX-nqKdchEYq_vVIzFHcNI_Mx7y1el192QbMyx6b0Gbj5L79wpuB7qCUqTBNhJZ2c07PuyPsewcNwglvnc-OrA-2vL6lJnBi5ZGH8gBH1cZCgmbrMpZGNFPG3oFpOn9JPzmnvQxe9tvSFFj5989A8d_XMHP-ZQ.dJZxnBRxJeMKswDsCA3cXA"}	Check yourself by using Private Key included in the response.

Public Key

This method is used to share your public key for encryption.

```
GET https://server-domain.com/public-key
```

Headers:

```
Content-Type: application/json
```

response:

```
200 OK
{
  "publicKey": "QSBwdWJsaWMga2V5IHNob3VsZCBiZSBoZXJlIGhvd2V2ZXIgaXQgd2FzIHRvbyBsb25nIDoo"
}
```

3DS External OTP Notifier

This document describes API for external OTP notifier handling. Clients that are interested into having OTP notifier on their side must have implement this API to allow communication with Antaca to provide one time password about the transaction to client own users.

API 3DS External OTP Notifier

Below you will find a list of endpoints that you should implement on your server side. Please pay special attention to the appropriate security of our connection, the syntax of requests that you can expect from the Verestro side, idempotency and the exact way in which you should respond to each request.

> These notifications support sending **Idempotency Key**

Notification OTP

This method is used to transfer a one-time password generated for transactions without a card present in the 3DS standard.

```
POST https://server-domain.com/notifications/otp
```

Headers:

```
Content-Type: application/json
X-Idempotency-Key: 20e87975-dbfb-4c95-b239-169516c0b707
```

request body:

```
{
   "storageCustomerId" => "1337",
   "storageCardId" => "1337",
   "balanceId" => "b334b384-328c-11ed-a261-0242ac120002",
```

```
    "amount" => "1000",
    "currency" => "PLN",
    "merchantName" => "merchant test",
    "otp" => "1111"
}
```

Parameters:

Parameter	Required	Description	Type
storageCustomerId	TRUE	Customer identifier	integer value
storageCardId	TRUE	Card identifier	integer value
balanceId	TRUE	User balance identifier	uuid v4
amount	TRUE	Transaction value in gross (minor value)	integer value
currency	TRUE	Currency 3-letters code in ISO 4217 https://www.iban.com/currency-codes	ISO 4217 3-letter code
merchantName	TRUE	Merchant name	string value
otp	TRUE	One time password	string value

success response:

```
204 No Content
```

error responses:

> **If an error is received, it is not possible to retry the request.**

```
Code 422
{
    "detail": "some specific details provided by server"
}
```

External Verification Notifier

This document describes API for processed KYC verification notifier handling. Clients that are interested into having information about status KYC verification on their side must have implement this API to allow communication with Antaca.

> Notifier provide notifications only with internal KYC status processes

> These notifications support sending **Idempotency Key**

Notification verification In-progress

This method is used to transfer information about changed KYC verification status to 'IN_PROGRESS'.

```
POST https://server-domain.com/notifications/verificationInProgress
```

Headers:

```
Content-Type: application/json
X-Idempotency-Key: 20e87975-dbfb-4c95-b239-169516c0b707
```

request body:

```
{
  "verificationId": "6faaa45a-41f6-4922-95fe-16e316ba7e91",
    "userId": "1337",
"email": "leonbakiewicz@gmail.com",
    "firstName": "Leon",
    "lastName": "Bakiewicz",
    "status": "IN_PROGRESS",
  "reason": null,
}
```

response:

```
204 No Content
```

Notification verification accepted

This method is used to transfer information about changed KYC verification status to 'ACCEPTED'.

```
POST https://server-domain.com/notifications/verificationAccepted
```

Headers:

```
Content-Type: application/json
X-Idempotency-Key: 20e87975-dbfb-4c95-b239-169516c0b707
```

request body:


```
{
  "verificationId": "6faaa45a-41f6-4922-95fe-16e316ba7e91",
    "userId": "1337",
  "email": "leonbakiewicz@gmail.com",
    "firstName": "Leon",
    "lastName": "Bakiewicz",
    "status": "ACCEPTED",
  "reason": null,
}
```


response:

```
204 No Content
```

Notification verification rejected

This method is used to transfer information about changed KYC verification status to 'REJECTED'.

```
POST https://server-domain.com/notifications/verificationRejected
```

Headers:

```
Content-Type: application/json
X-Idempotency-Key: 20e87975-dbfb-4c95-b239-169516c0b707
```

request body:

```
{
  "verificationId": "6faaa45a-41f6-4922-95fe-16e316ba7e91",
    "userId": "1337",
  "email": "leonbakiewicz@gmail.com",
    "firstName": "Leon",
    "lastName": "Bakiewicz",
    "status": "REJECTED",
  "reason": 'INVALID_CUSTOMER_DATA',
}
```


response:

204 No Content

Parameters:

Parameter	Required	Description	Type
verificationId	TRUE	Verification identifier	uuid v4
userId	TRUE	User identifier	integer value
email	TRUE	User's email address	string value
firstName	TRUE	User first name	string value
lastName	TRUE	User last name	string value
status	TRUE	Verification status. Possible values: • REJECTED • IN_PROGRESS • ACCEPTED	string value

Parameter	Required	Description	Type
reason	TRUE	Verification status reason **ACCEPTED**: null **IN_PROGRESS**: null **REJECTED**: • INVALID_CUSTOMER_DATA • BLURRED_DOCUMENT_PHOTO • INVALID_DOCUMENT_PHOTO • BLURRED_SELFIE • INVALID_SELFIE	null/string value

Sensitive data:

This method is used to share your public key for encryption.

```
GET https://server-domain.com/public-key
```

response:

```
200 OK
{
  "publicKey": "QSBwdWJsaWMga2V5IHNob3VsZCBiZSBoZXJlIGhvd2V2ZXIgaXQgd2FzIHRvbyBsb25nIDoo"
}
```

Transactions notifier

To get notifications about transactions use Transaction History Core API

Revision #77
Created 19 September 2023 07:00:51 by Tadeusz Krysa
Updated 3 December 2024 18:10:26 by Barbara Tudruj